

From generated to reconstructed particles

Exercise

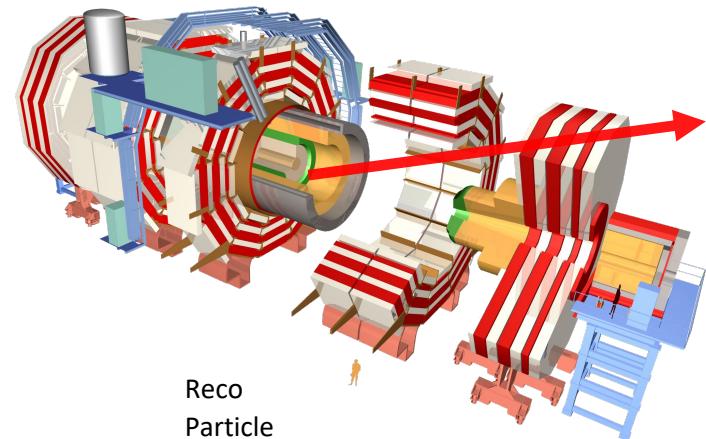
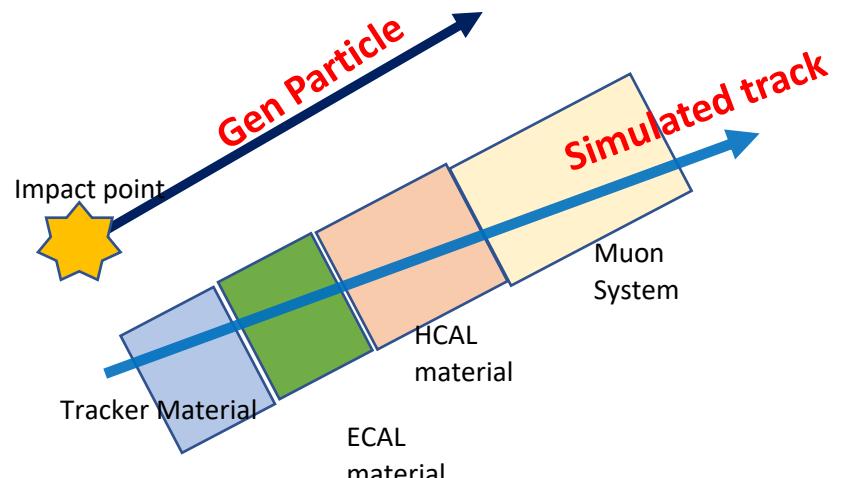
R. Venditti, F. Simone

A.A. 2023-2024

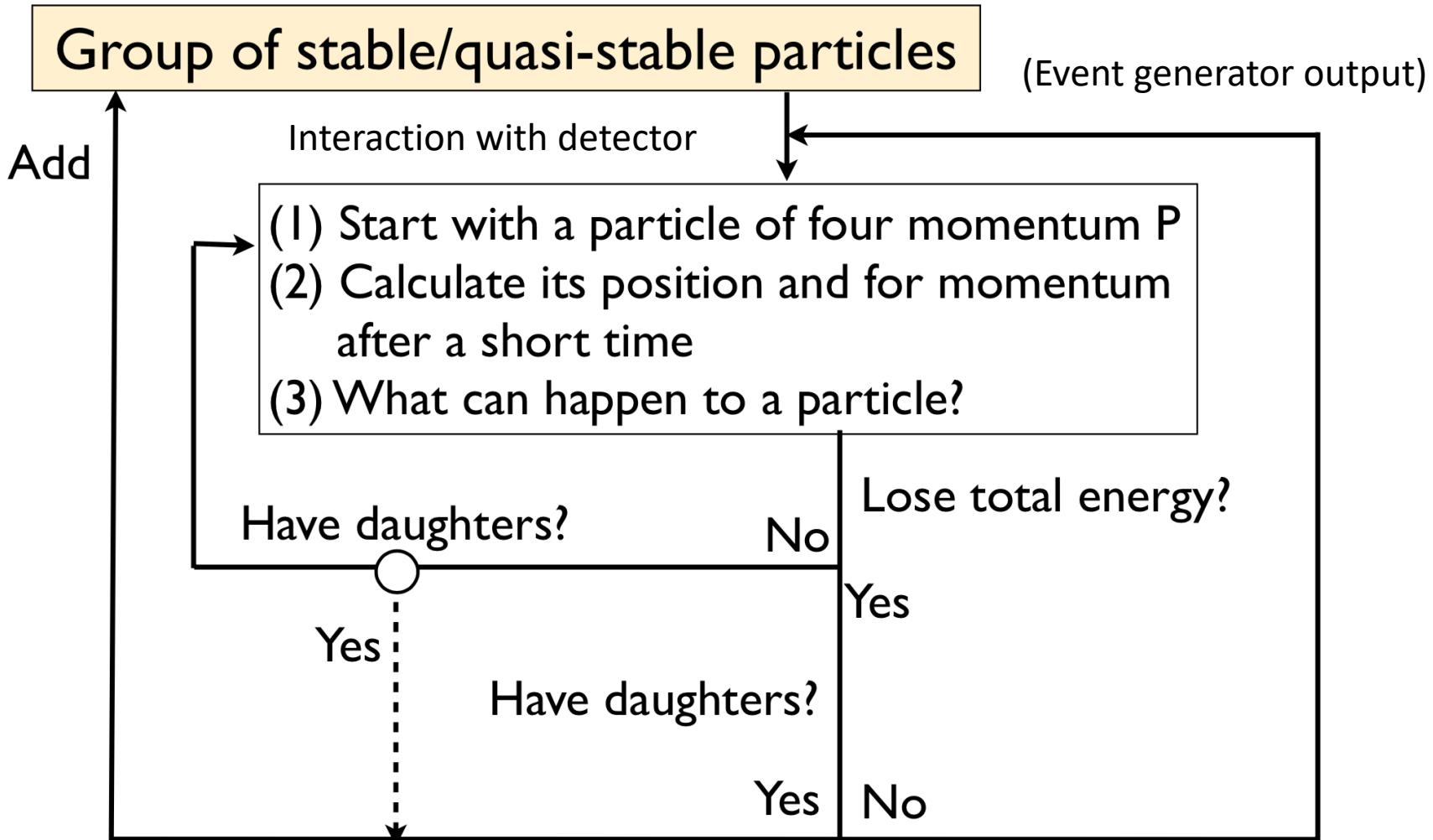


Particle transportation and interaction with matter

- 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 simulated 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 p_1 : ionize the gas, loose some momentum, produce N secondary electrons with momenta P_1, \dots
 - Do nothing with probability $1-p_1$
- Generate random number r in the range $[0,1]$.
- If $r < p_1$, 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

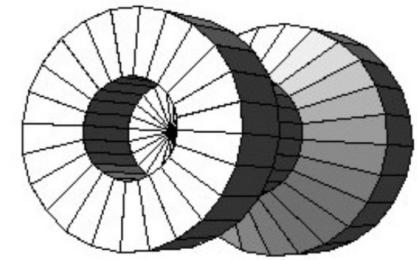
- We (HEP experimentalists) do not write detector simulation programs ourselves
- In CERN there is a dedicated group which does this for us.
- They develop GEANT code for simulation of interaction of radiation with matter.
- GEANT started in 1950's, evolves since GEANT4
- Popular detector simulation
 - 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 some data 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?



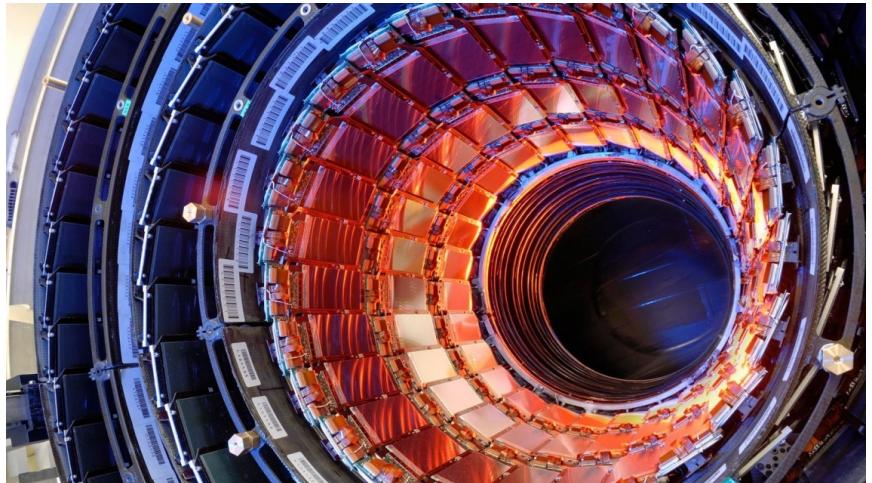
- We need a tool capable to simulate (realistically) particle interactions with matter from the scale
 - of the TeV: primary particles generated in LHC collision
 - of the keV: delta rays from secondary (tertiary...) particles leaving energy for ionization in our detectors
- Correct physics simulation needed for ~ 9 orders of magnitude
- GEANT4 is not a program ... It 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/X_0 is the number of radiation lengths of the given detector
 - **#Hadronic interactions:** λ/λ_0 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 ($qq \rightarrow Hqq$); 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!



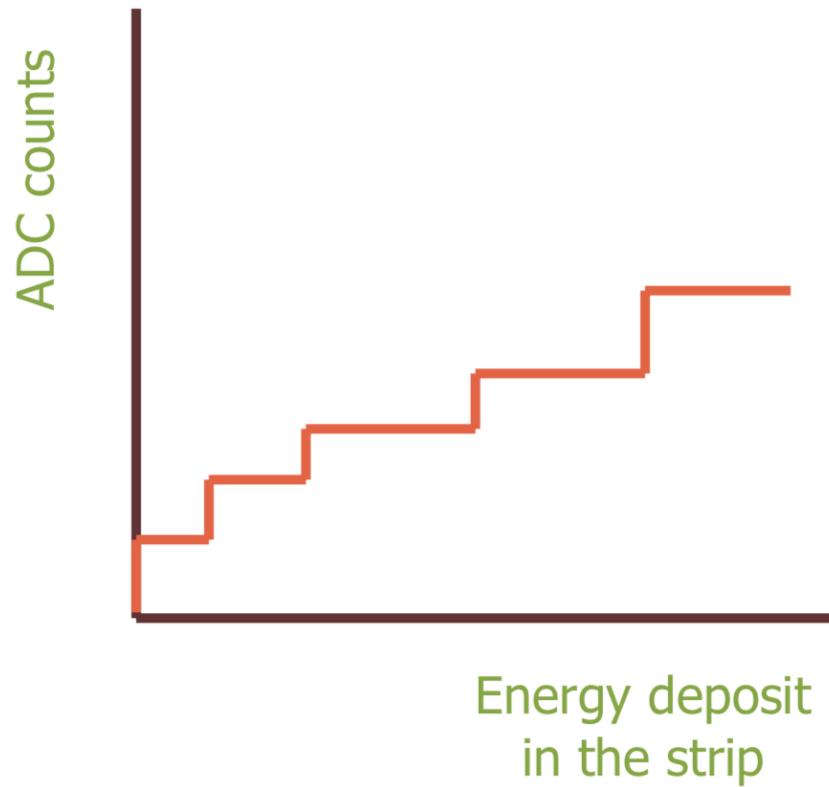
All of these are encoded in G4 library: there are packages, called **physics list**.

- Data driven model
- Parametrized model
- Theory driven model

No perfect physics list!
You need to choose it by yourself.

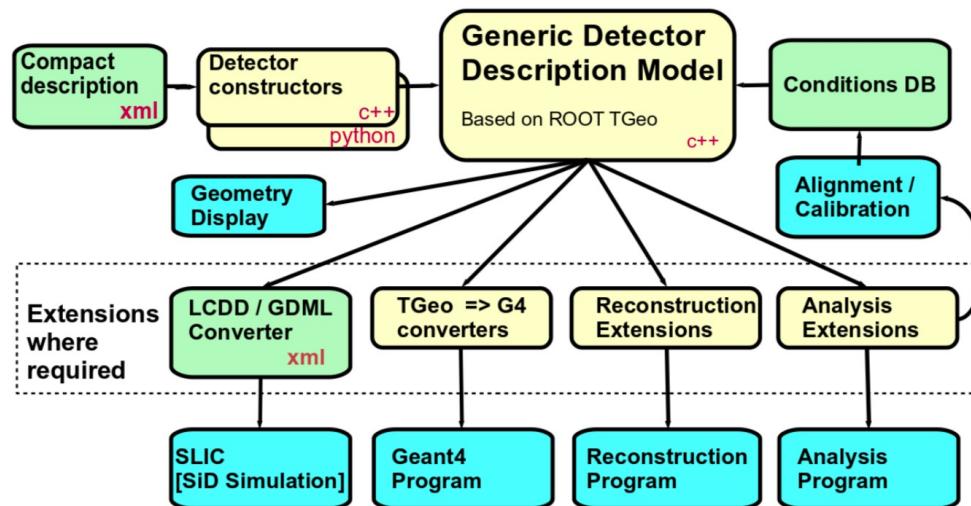
Simulation of the response of a detector

- 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

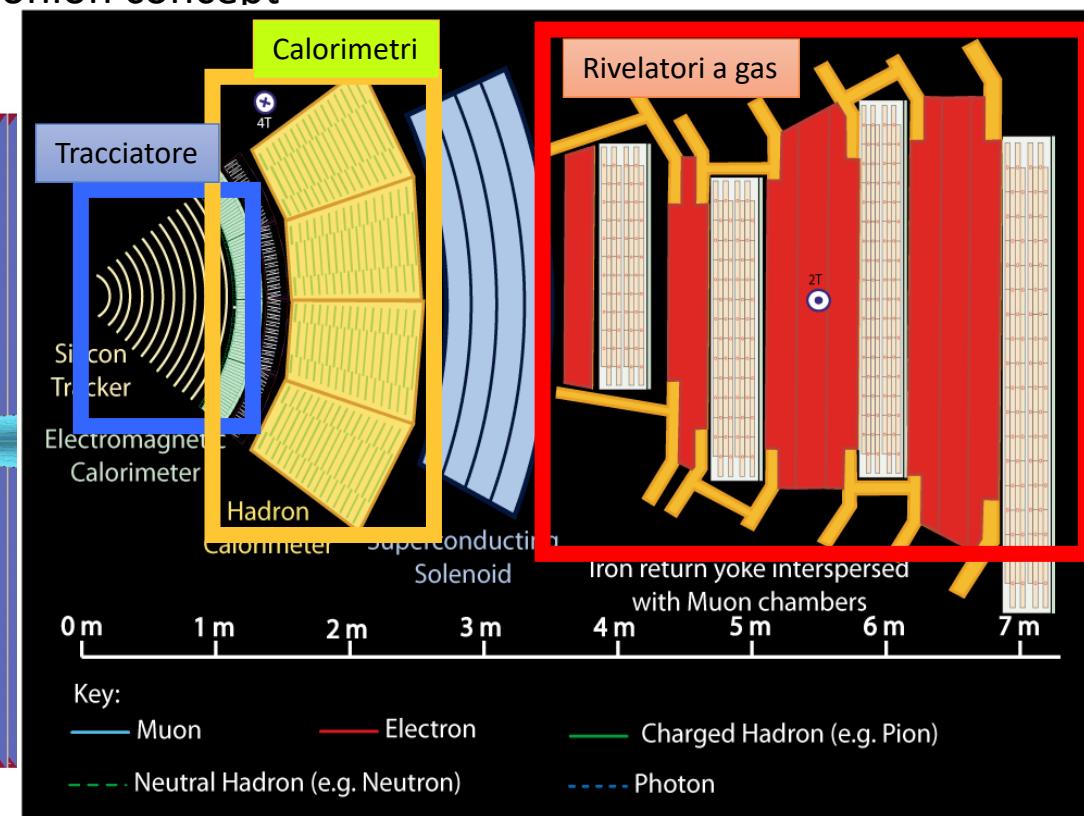
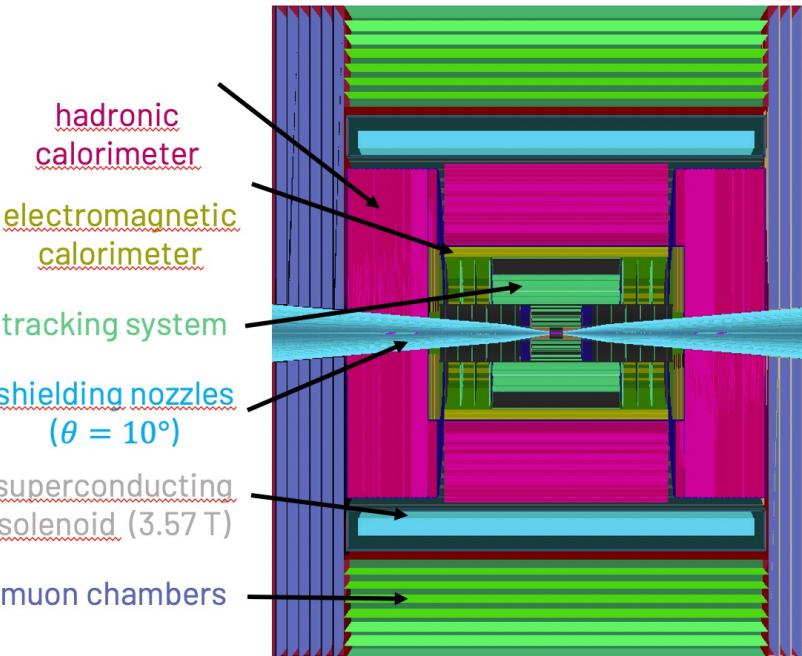
- Identification: Based on combining information coming from several detectors (Particle flow)

Key measurements:

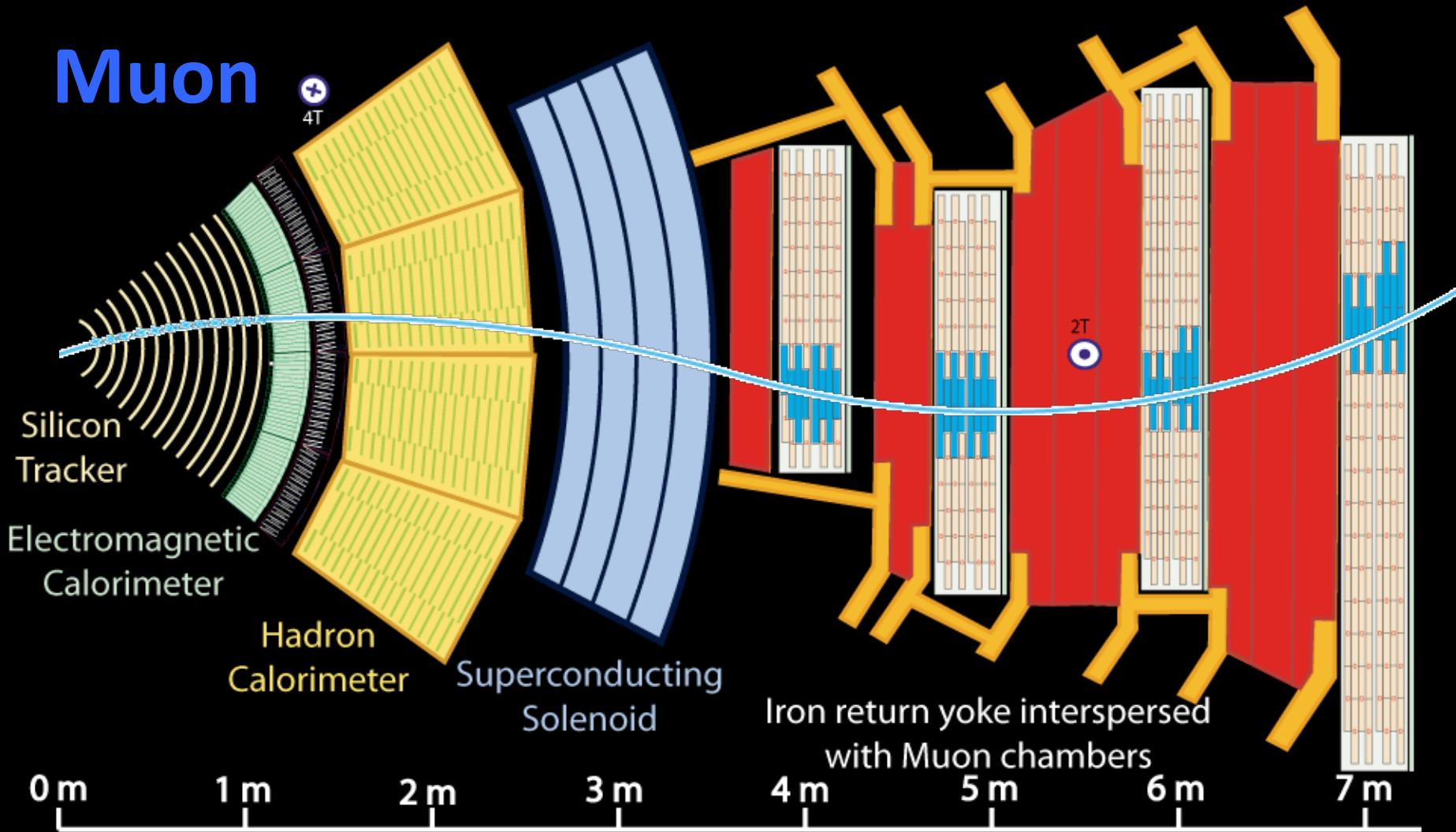
- position (hits in tracker and/or muon systems, cluster in calorimeters)
- Track curvature in B-field (from position interpolation) → momentum
- energy measured in calorimeters

Apparatus design is based on cylindrical onion concept

Muon collider experiment



Muon



Key:

— Muon

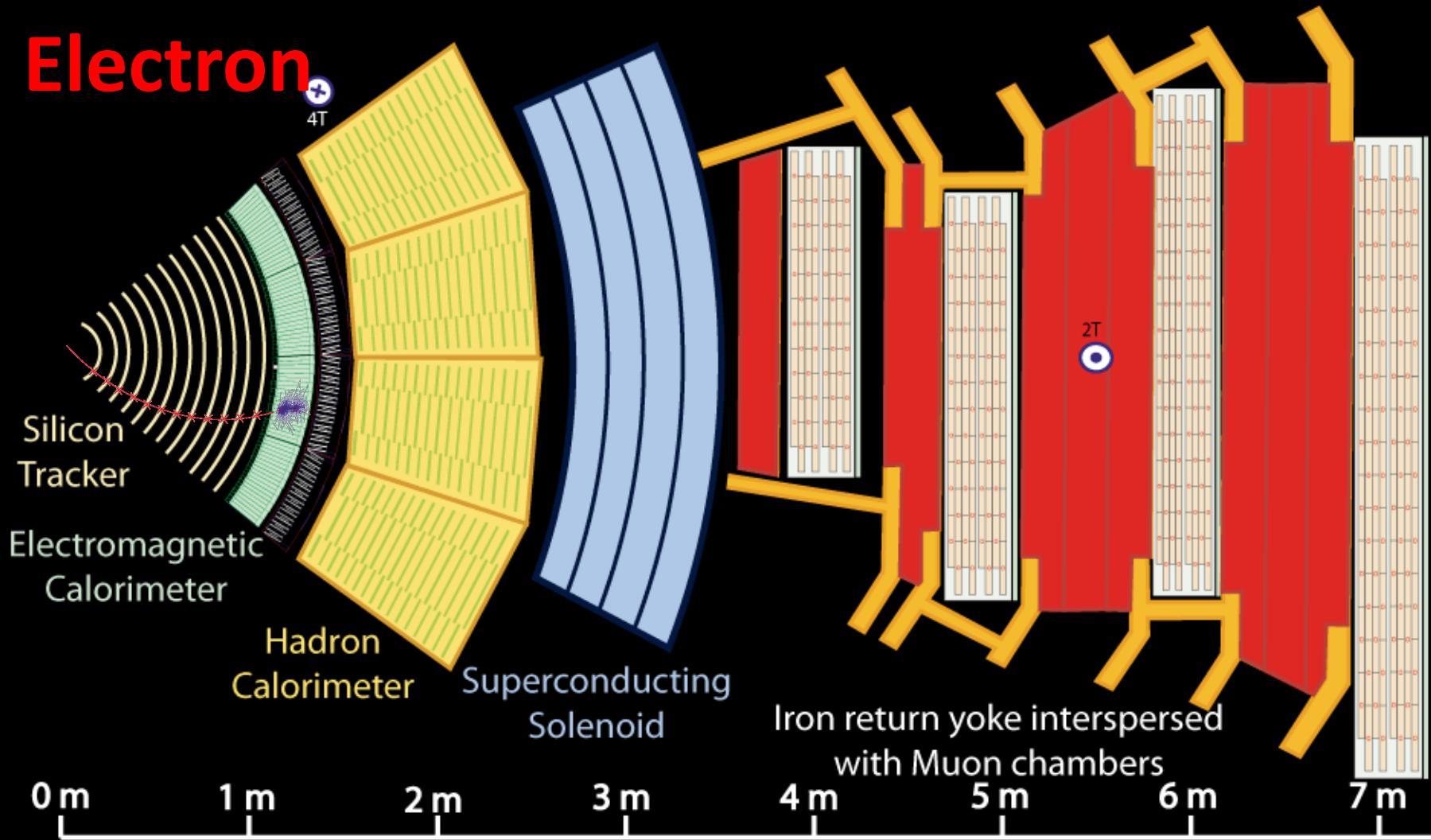
— Electron

— Charged Hadron (e.g. Pion)

- - - Neutral Hadron (e.g. Neutron)

- - - Photon

Electron



Key:

— Muon

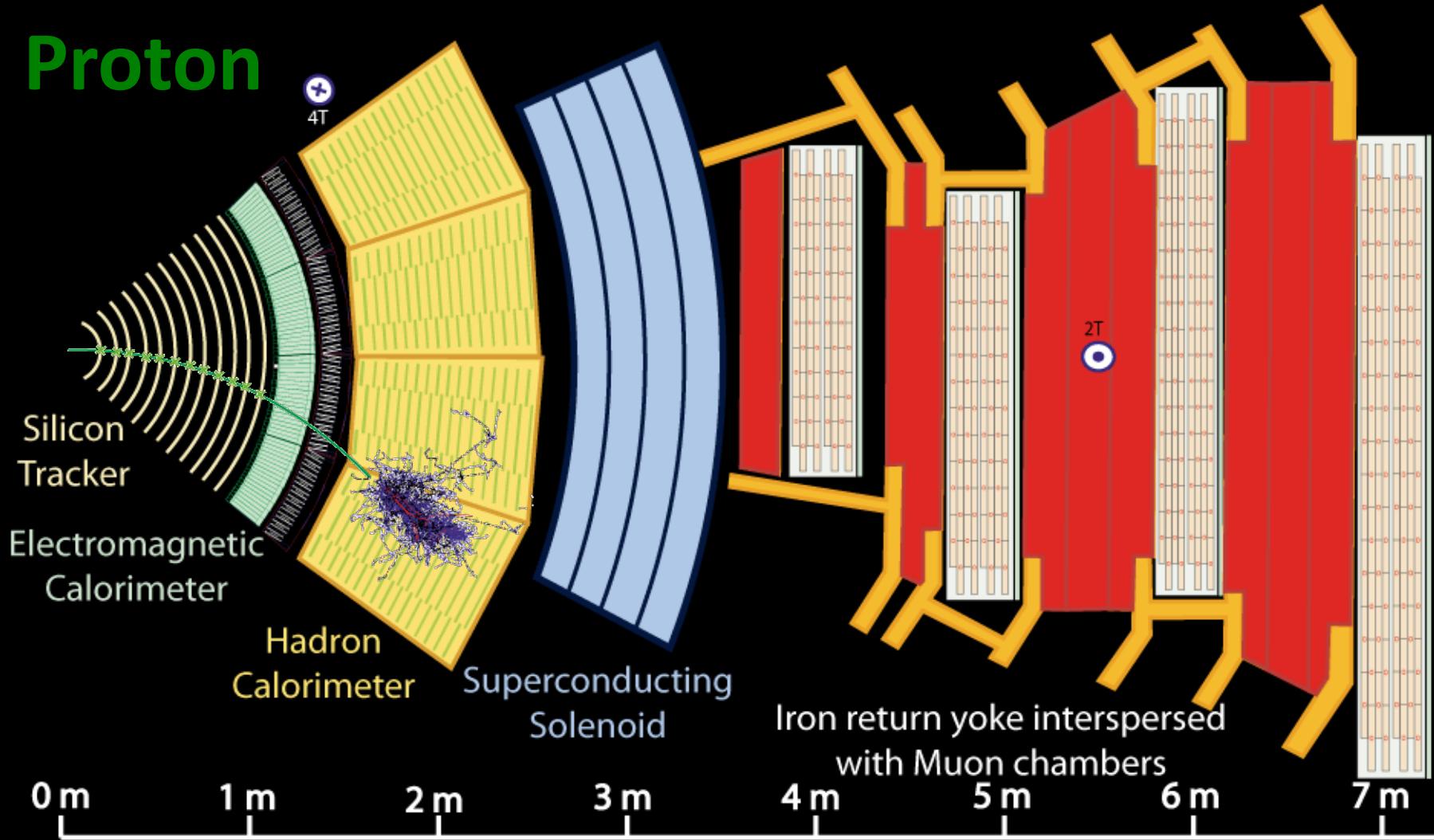
— Electron

- - - Neutral Hadron (e.g. Neutron)

— Charged Hadron (e.g. Pion)

--- Photon

Proton



Key:

— Muon

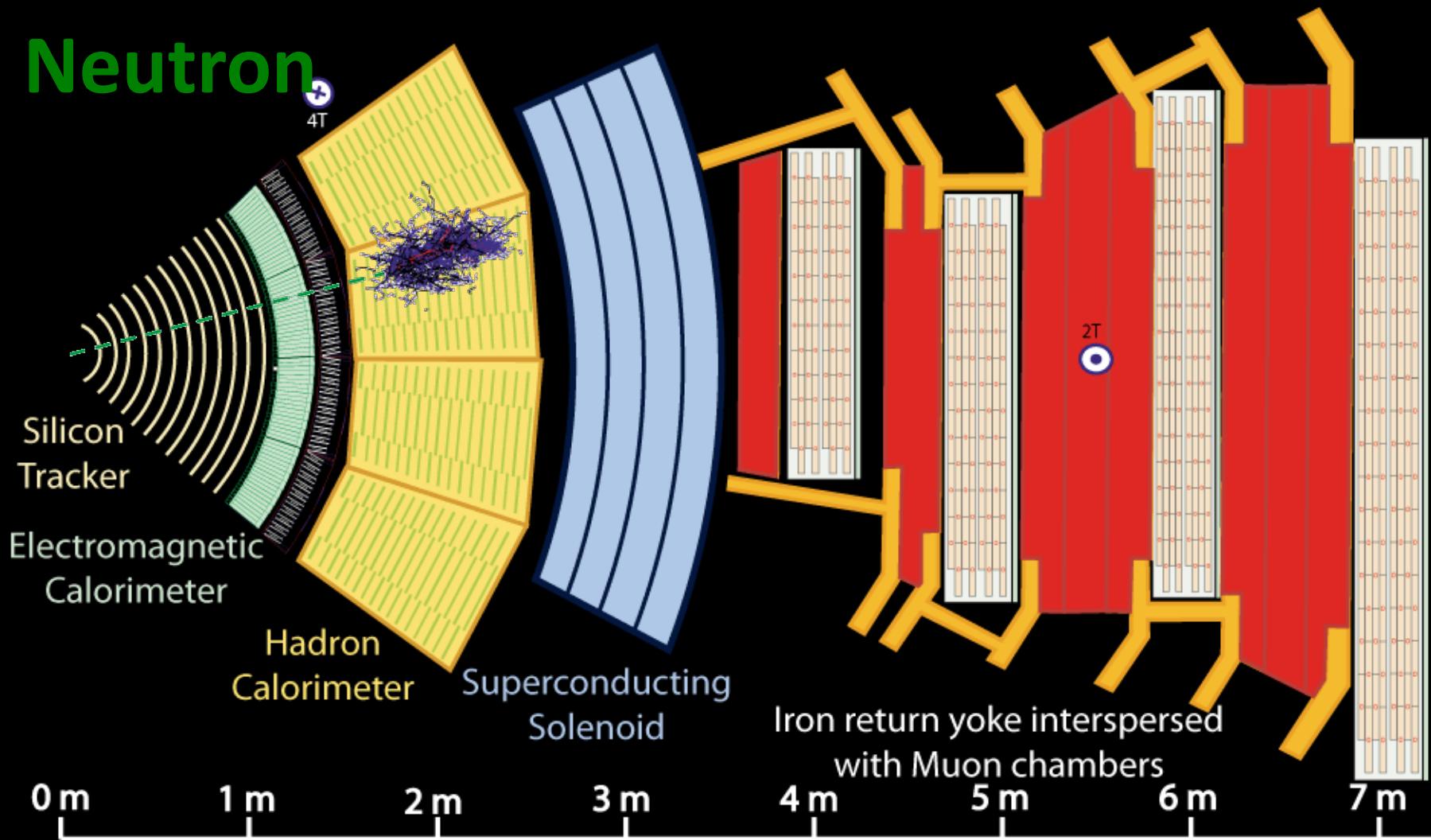
— Electron

— Charged Hadron (e.g. Pion)

- - - Neutral Hadron (e.g. Neutron)

--- Photon

Neutron



Key:

— Muon

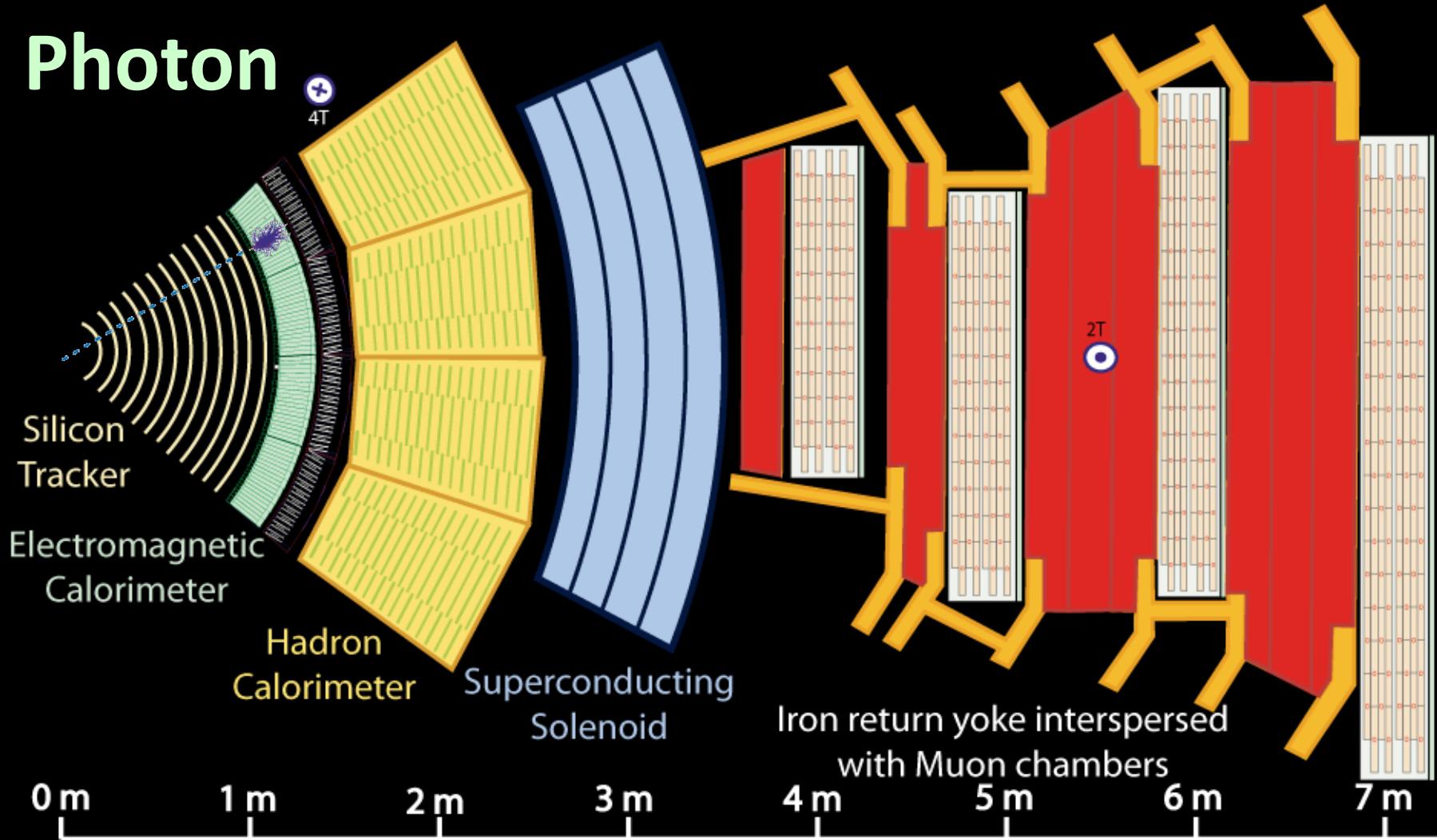
— Electron

— Charged Hadron (e.g. Pion)

- - - Neutral Hadron (e.g. Neutron)

- - - Photon

Photon



Key:

— Muon

— Electron

— Charged Hadron (e.g. Pion)

- - - Neutral Hadron (e.g. Neutron)

--- Photon

Very few words on

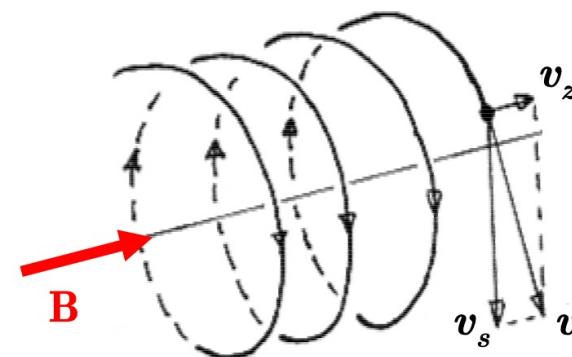
Charged particle momentum measurement

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 homogeneous magnetic field the trajectory is given by an helix of radius R

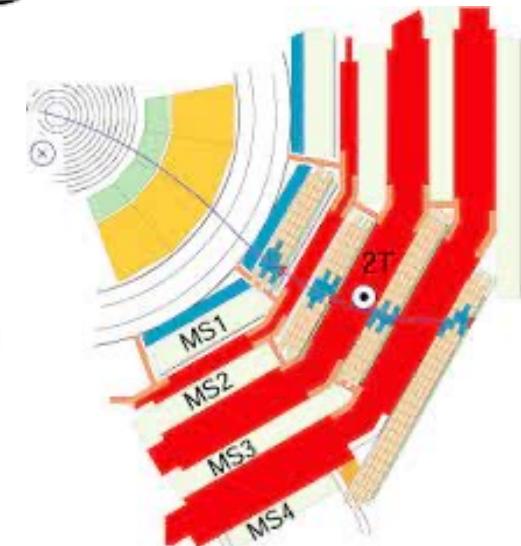
$$R(m) = \frac{p_{\perp}(\text{GeV})}{0.3B(T)}$$



By measuring R one can retrieve the pT

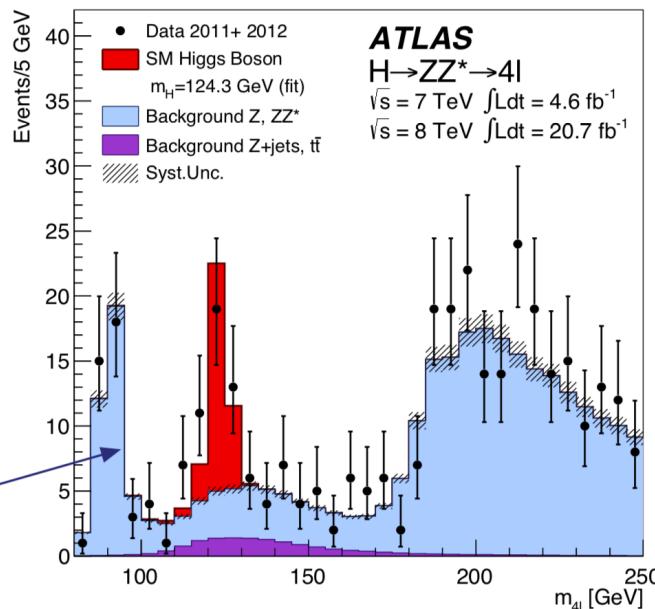
The uncertainty on momentum estimation is given by:

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



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

The screenshot shows the official Miniconda documentation page on the docs.conda.io website. The page has a green header with the Conda logo and a search bar. The main content area is titled "Miniconda" and describes it as a free minimal installer for conda. It includes sections for "System requirements", "Latest Miniconda Installer Links", and "Windows installers". A sidebar on the left lists various Conda-related links.

Miniconda

Miniconda is a free minimal installer for conda. It is a small, bootstrap version of Anaconda that includes only conda, Python, the packages they depend on, and a small number of other useful packages, including pip, zlib and a few others. Use the [`conda install`](#) command to install 720+ additional conda packages from the Anaconda repository.

See if Miniconda is right for you.

System requirements

- License: Free use and redistribution under the terms of the [EULA for Miniconda](#).
- Operating system: Windows 8 or newer, 64-bit macOS 10.13+, or Linux, including Ubuntu, Red Hat, CentOS 7+, and others.
- If your operating system is older than what is currently supported, you can find older versions of the Miniconda installers in our [archive](#) that might work for you.
- System architecture: Windows- 64-bit x86, 32-bit x86; macOS- 64-bit x86 & Apple M1 (ARM64); Linux- 64-bit x86, 64-bit aarch64 (AWS Graviton2), 64-bit IBM Power8/Power9, s390x (Linux on IBM Z & LinuxONE).
- The [`linux-aarch64`](#) Miniconda installer requires [`glibc >=2.26`](#) and thus will not work with CentOS 7, Ubuntu 16.04, or Debian 9 ("stretch").
- Minimum 400 MB disk space to download and install.

On Windows, macOS, and Linux, it is best to install Miniconda for the local user, which does not require administrator permissions and is the most robust type of installation. However, if you need to, you can install Miniconda system wide, which does require administrator permissions.

Latest Miniconda Installer Links

Latest - Conda 23.3.1 Python 3.10.10 released April 24, 2023

Platform	Name	SHA256 hash
Windows	Miniconda3 Windows 64-bit	387194e1f120beb52b83634e89cc67db4f7988bd542254d43d3389ea7fc358
	Miniconda3 Windows 32-bit	4fb64edc9c28880eb16994bfba4829118ex3145baa0d0d5344174ab65d462
macOS	Miniconda3 macOS Intel x86 64-bit bash	5ab70b664b7da9d14ade3130534cc98283hb830cd01baed9cf0d9cc4153f8184
	Miniconda3 macOS Intel x86 64-bit pkg	c5ca18f11c5394f2b739726dc22551c2a19fd1f689c13a275668887u706cfa58
	Miniconda3 macOS Apple M1 64-bit bash	9ed1d12573339c4095b0d5a84af0fff6c32d13c3de1b3d478c8187eb483d64
	Miniconda3 macOS Apple M1 64-bit pkg	699f74725f1f90d772eb776e39974e3e227736c63a77f7083106a336cc7facc075d
Linux	Miniconda3 Linux 64-bit	aef2796bea7767948f16aaad17eb05f6aa9c9487c7c0346f781f4819e5a651
	Miniconda3 Linux-aarch64 64-bit	6950c7b1ff65cc9b87ec1a2d684837771ae7b2ed044e8d9c9d15d1dedc924c
	Miniconda3 Linux-ppc64le 64-bit	b3de53bc0542bc4f5a2f2d2a7338528866e84f8e1a4597573cfe64763e51d16
	Miniconda3 Linux-s390x 64-bit	ed4f51aef987e921ff5721151f567a4c43c4288ac93ec2393c623808c4891de8

Windows installers

Windows

Setup the environment

2) Create a conda environment containing ROOT

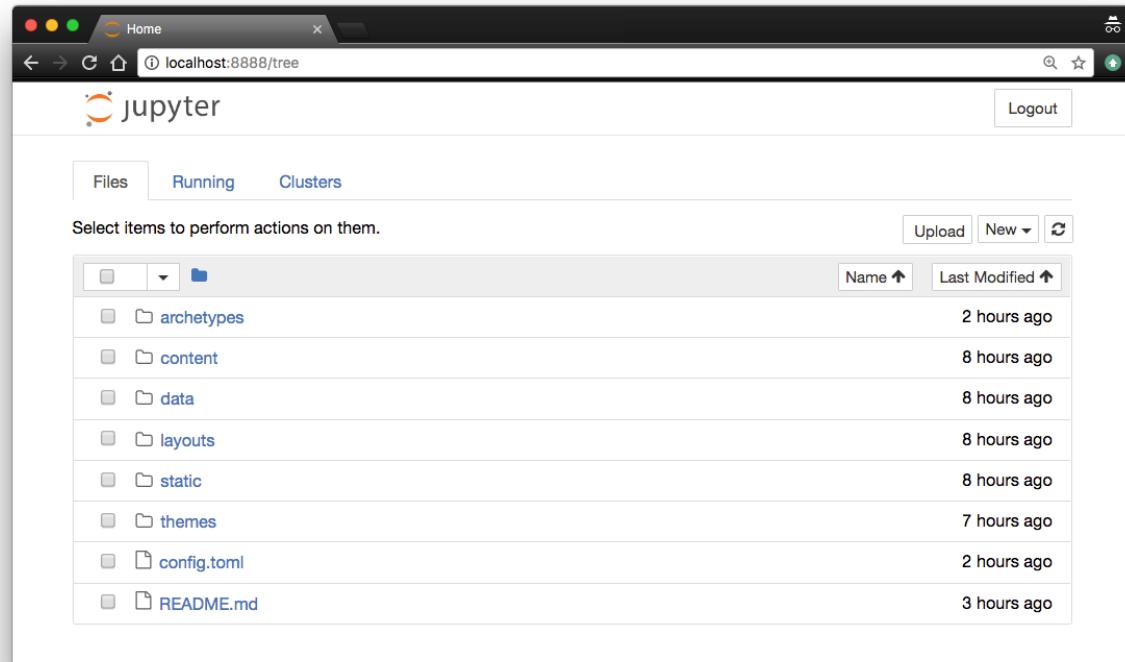
```
$ conda config --set channel_priority strict  
$ conda create -c conda-forge --name <my-environment> root  
$ conda activate <my-environment>
```

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

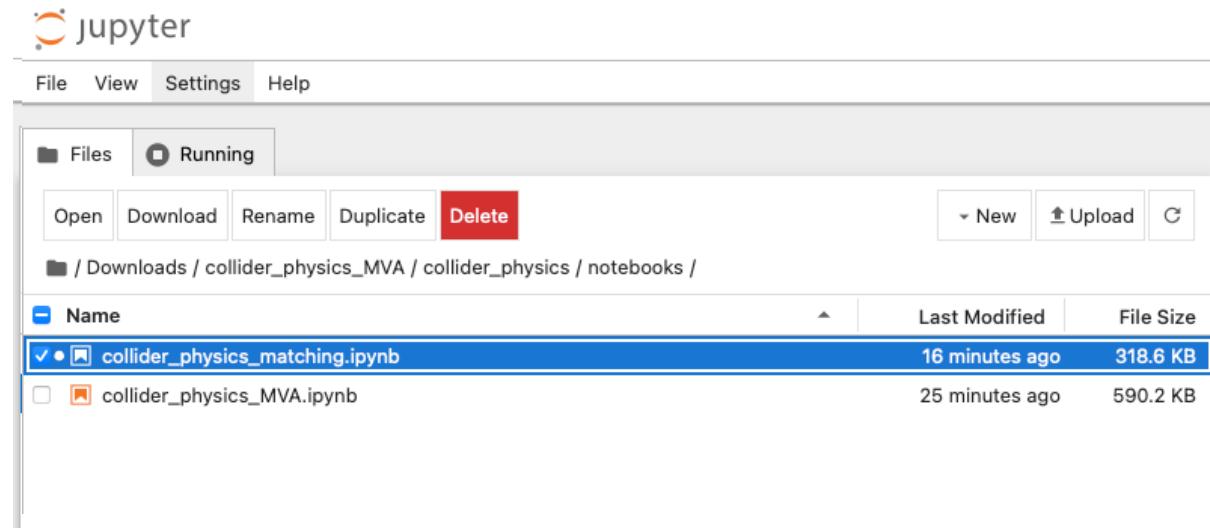
5) Download code

```
$ 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

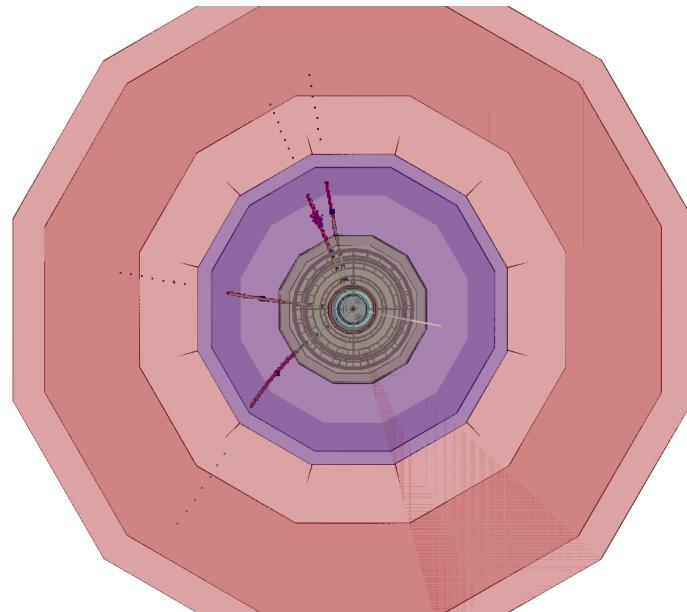
- generated particles: the output of event generator
- Reconstructed particles: the simulation of what the experimental apparatus measures, convolution of several effects:
 - Observables of a generated particle are smeared by the interaction with matter and digitization by readout electronics
 - the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

$\mu\mu \rightarrow Hvv \rightarrow ZZ \rightarrow 4\mu$

Madgraph
Output

```
HepMC::Version 2.06.09
HepMC::IO_GenEvent-START_EVENT_LISTING
E 0 -1 -1.000000000000000e+00 -1.000000000000000e+00 -1.000000000000000e+00 9999 0 5 1 2 0 2 5.409539000000005e-06 5.409539000000004e-09
N 2 "Weight" "Weight_MERGING=0.000"
U GEV MM
C 5.409539000000008e-08 5.409539000000005e-06
V -1 0 0 0 0 1 1 0
P 1 -13 0 0 7.499999255730961e+02 7.500000000000001e+02 1.056600000000000e-01 4 0 0 -1 0
P 3 -13 0 0 7.500000000000000e+02 7.500000000000000e+02 0 21 0 0 -3 0
V -2 0 0 0 0 1 1 0
P 2 13 0 0 -7.499999255730961e+02 7.500000000000001e+02 1.056600000000000e-01 4 0 0 -2 0
P 4 13 0 0 -7.500000000000000e+02 7.500000000000000e+02 0 21 0 0 -3 0
V -3 0 0 0 0 0 3 0
P 5 25 -1.841456176399998e+01 -2.751859515900000e+01 -5.155227643399998e+02 5.3149314574130995e+02 1.249994970100000e+02 22 0 0 -4 0
P 6 14 -6.172836374530293e+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 3 0
P 8 23 -2.1144916457000001e+01 -1.4911177261000001e+01 -3.822972530000001e+02 3.9425097231703830e+02 9.2807267077000006e+01 22 0 0 -5 0
P 9 -13 5.898147327302924e+00 -1.7677473595528799e+01 -1.2785938531898643e+02 1.2921035095068419e+02 1.056600000000000e-01 1 0 0 0 0
P 10 13 -3.1679077085277738e+00 0.6099640245662030e+00 -5.3704258527176298e+00 8.0369653685506357e+00 1.056600000000000e-01 1 0 0 0 0
V -5 0 0 0 0 0 2 0
P 11 -13 -2.4517622542645125e+01 -4.8837158790649319e+01 -3.0370961298568955e+02 3.0858665246222120e+02 1.056600000000000e-01 1 0 0 0 0
P 12 13 3.3727060857451256e+00 3.3925981529649320e+01 -7.8587640015310413e+01 8.5664319855483285e+01 1.056600000000000e-01 1 0 0 0 0
E 1 -1 -1.000000000000000e+00 -1.000000000000000e+00 -1.000000000000000e+00 9999 0 11 1 2 0 2 5.409539000000005e-06 5.409539000000004e-09
N 2 "Weight" "Weight_MERGING=0.000"
```

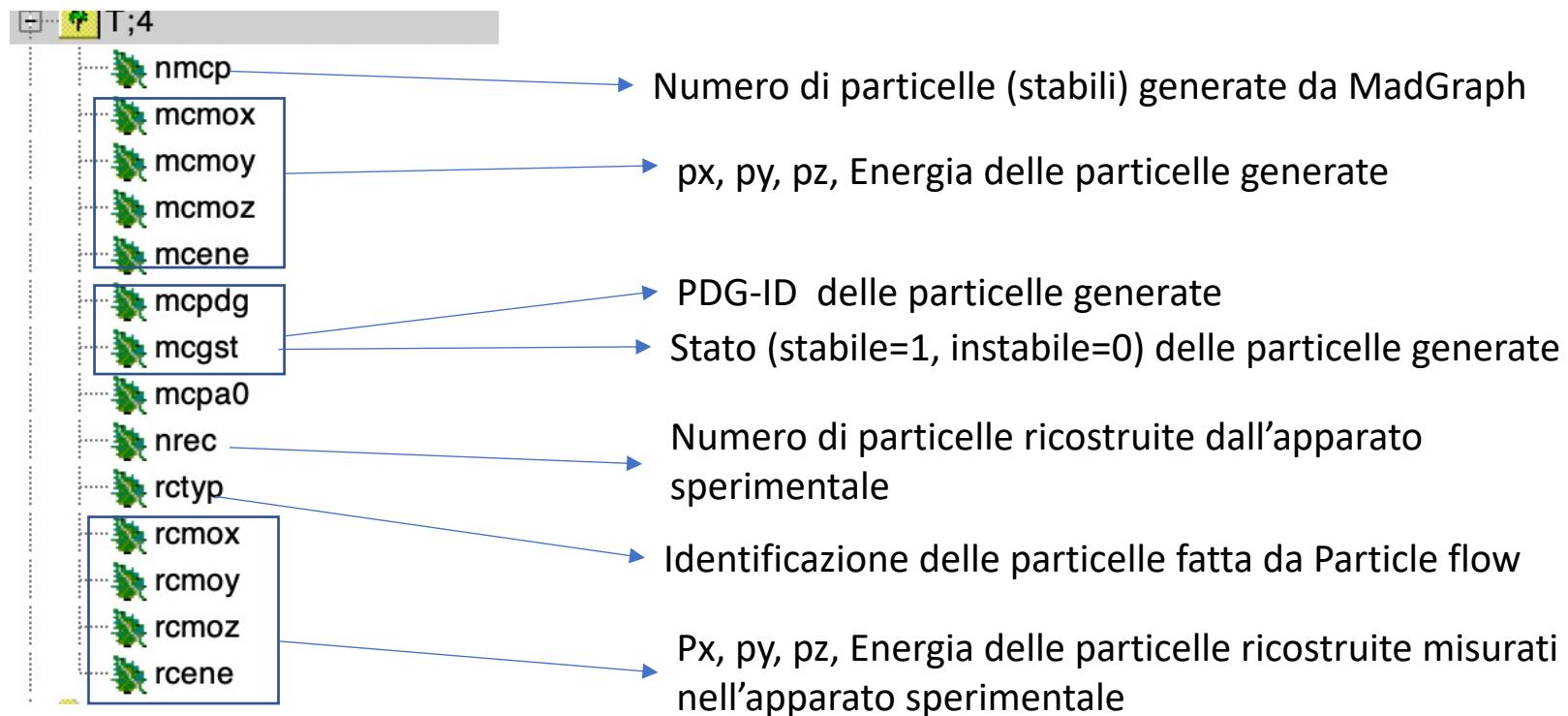
VS



Muon Collider detector
Reconstruction output

Our inputs

```
root -l ntuple_mumuHZZ4mu_final.root  
root [2] TBrowser b
```



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

Exercise 2A

```
[6]: # Define histograms (TH1F) with name and binning
```

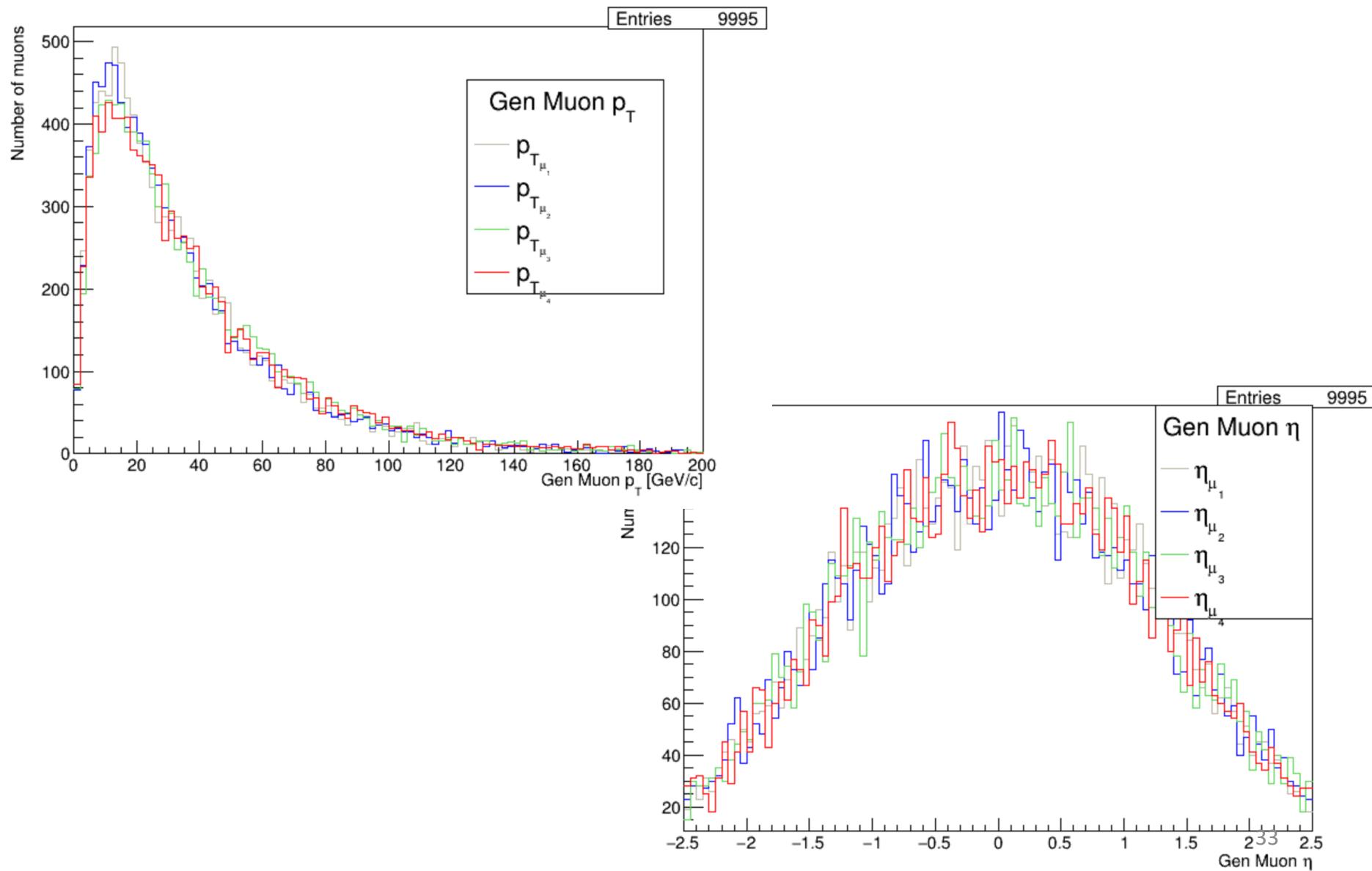
```
hGenMu1Pt = TH1F("hGenMu1Pt","Gen Muon 1 Pt",100,0,200)
hGenMu2Pt = TH1F("hGenMu2Pt","Gen Muon 2 Pt",100,0,200)
hGenMu3Pt = TH1F("hGenMu3Pt","Gen Muon 3 Pt",100,0,200)
hGenMu4Pt = TH1F("hGenMu4Pt","Gen Muon 4 Pt",100,0,200)
```

```
[7]: hRecoMu1Pt = TH1F("hRecoMu1Pt","Reco Muon 1 Pt",100,0,200)
hRecoMu2Pt = TH1F("hRecoMu2Pt","Reco Muon 2 Pt",100,0,200)
hRecoMu3Pt = TH1F("hRecoMu3Pt","Reco Muon 3 Pt",100,0,200)
hRecoMu4Pt = TH1F("hRecoMu4Pt","Reco Muon 4 Pt",100,0,200)
```

```
[9]: hGenMu1Eta = TH1F("hGenMu1Eta","Gen Muon 1 Eta",100,-2.5,2.5)
hGenMu2Eta = TH1F("hGenMu2Eta","Gen Muon 2 Eta",100,-2.5,2.5)
hGenMu3Eta = TH1F("hGenMu3Eta","Gen Muon 3 Eta",100,-2.5,2.5)
hGenMu4Eta = TH1F("hGenMu4Eta","Gen Muon 4 Eta",100,-2.5,2.5)
```

```
[10]: hRecoMu1Eta = TH1F("hRecoMu1Eta","Reco Muon 1 Eta",100,-2.5,2.5)
hRecoMu2Eta = TH1F("hRecoMu2Eta","Reco Muon 2 Eta",100,-2.5,2.5)
hRecoMu3Eta = TH1F("hRecoMu3Eta","Reco Muon 3 Eta",100,-2.5,2.5)
hRecoMu4Eta = TH1F("hRecoMu4Eta","Reco Muon 4 Eta",100,-2.5,2.5)
```

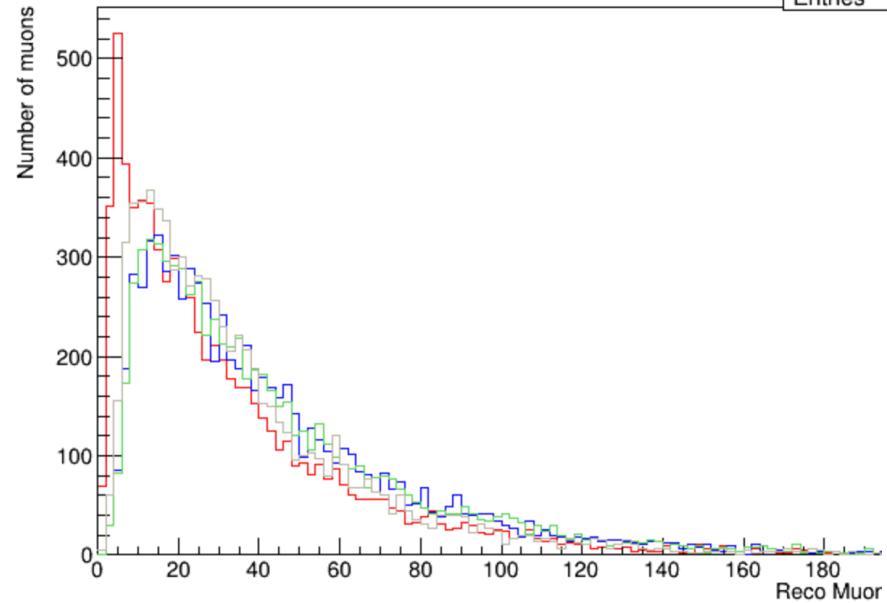
Exercise 2A



Exercise 2A

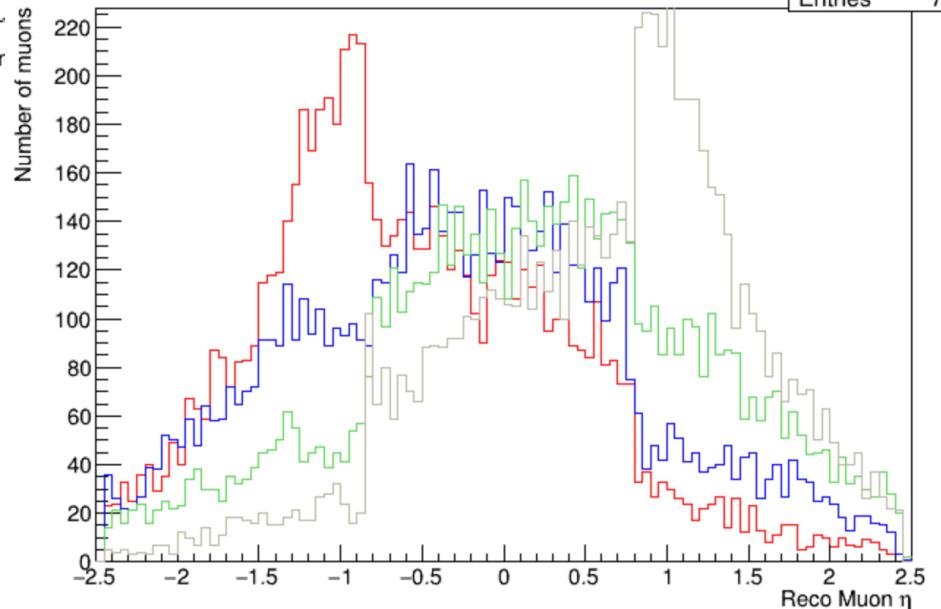
Reco Muon 1 Pt

Entries 7488



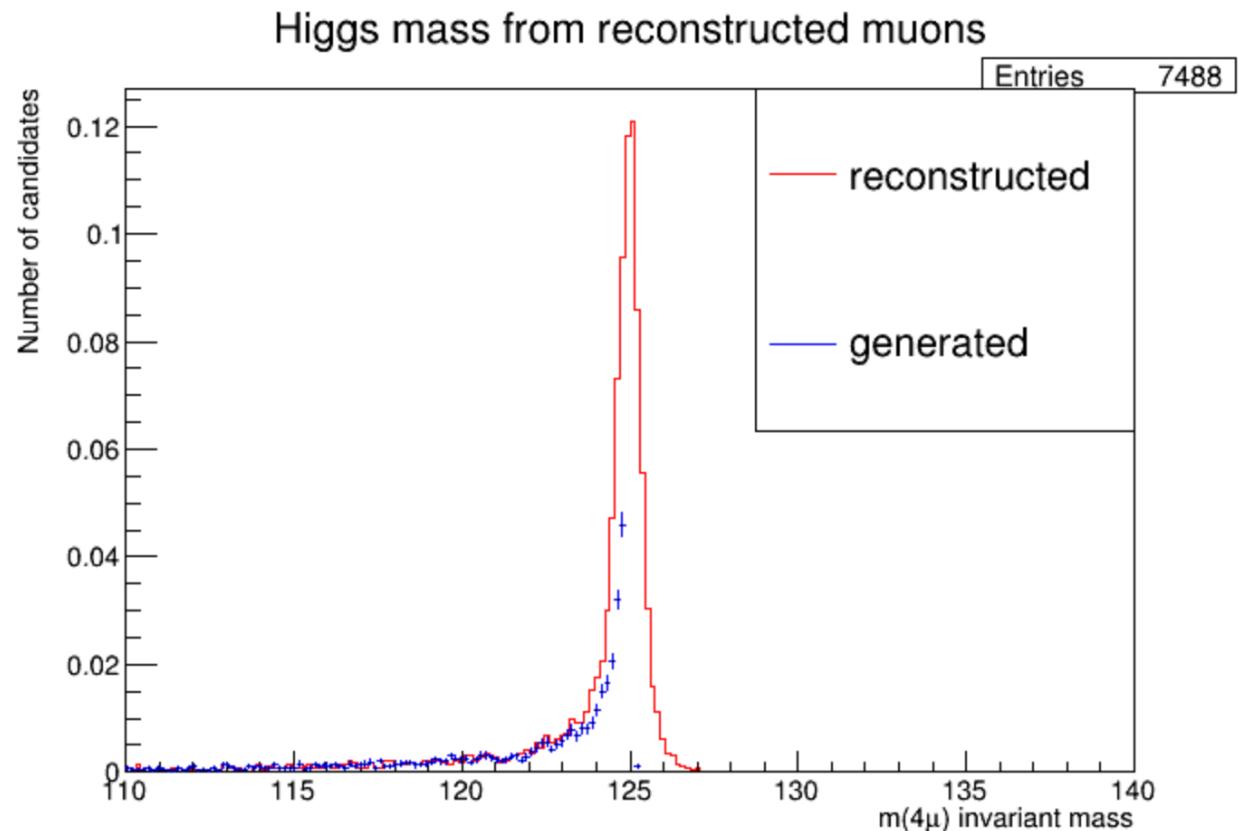
Reco Muon 1 Eta

Entries 7488



Mu1..4 don't really correspond to the generated muons.
Are they shuffled?

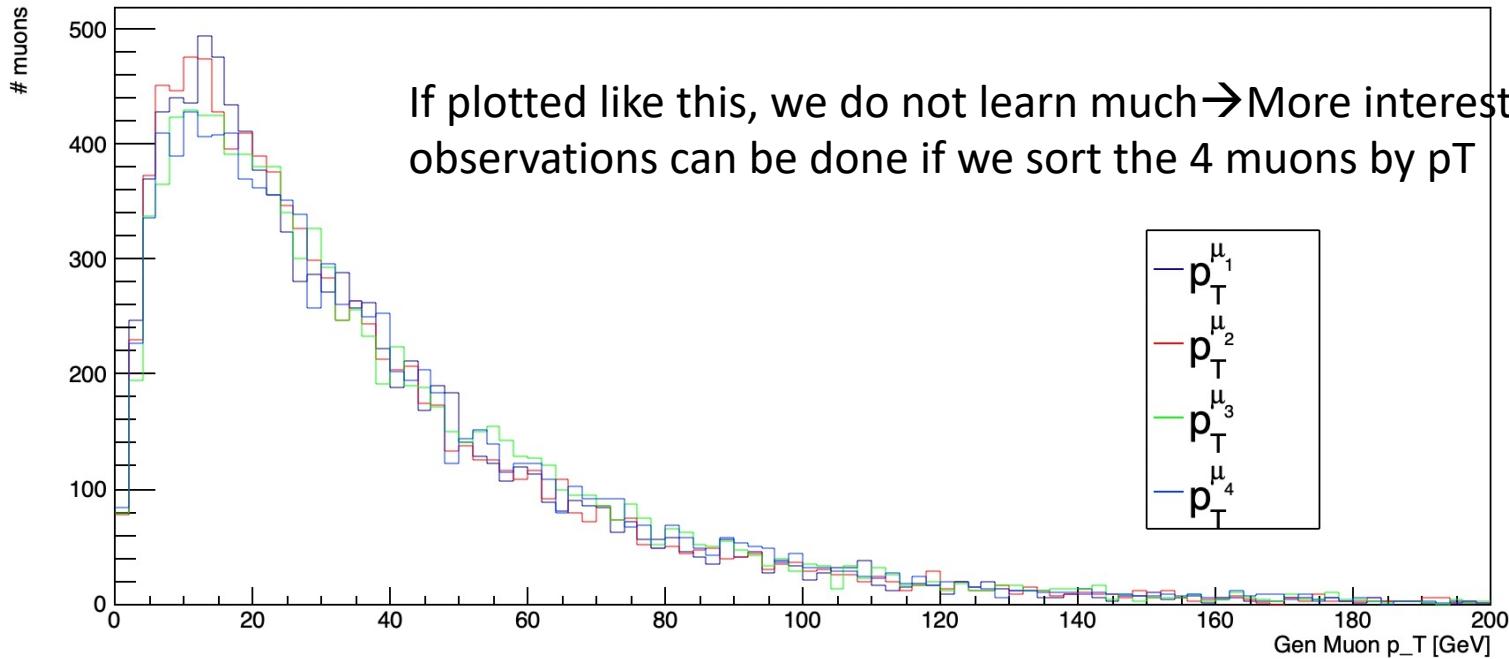
Exercise 2A



Not all the reconstructed muons originate from the simulated process! Need to match the generated and reconstructed muons to reject muons from other sources.

Gen Muon p_T

Gen Muon pT



→ 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()
```

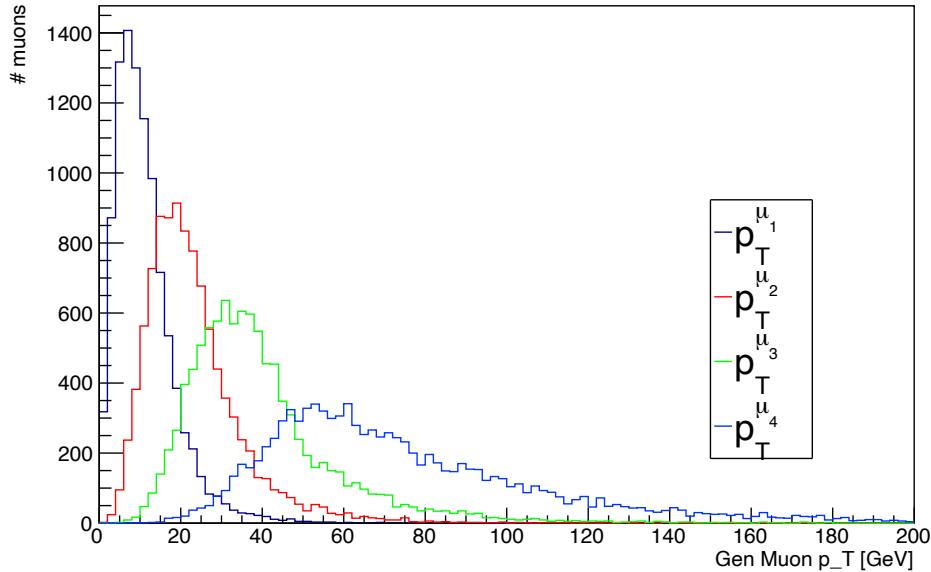
Exercise 2B

- sort muons by pT and plot sorted pT!
- sort muons by eta and plot sorted eta!

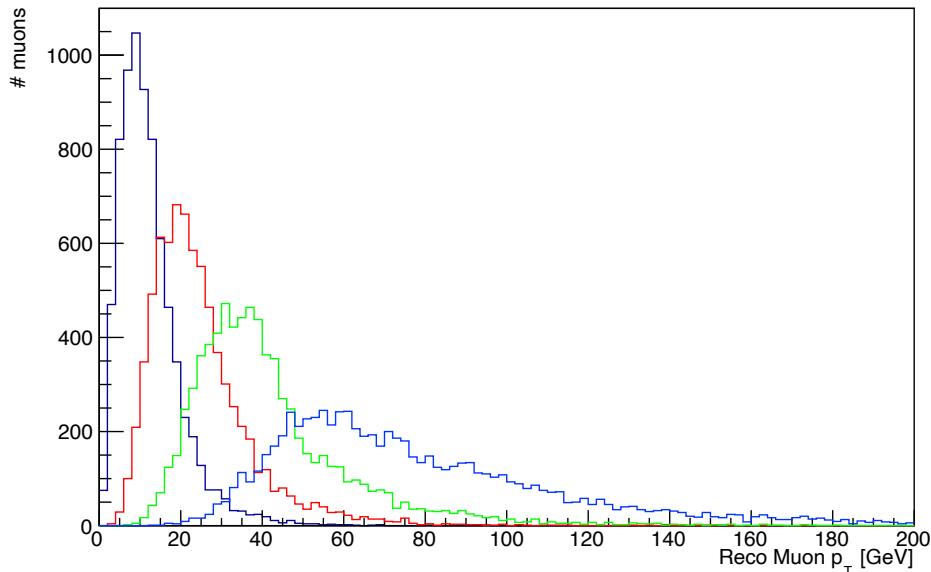
Plot the pt of the 4 sorted muons on same canvas!

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

Gen Muon pT



Reco Muon pT

**Note1**

- The lowest p_T muon has **$pT \sim \text{few GeV}$**
- 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 p_T , are not the same for the same muon

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

Instead, generated muon p_T 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,

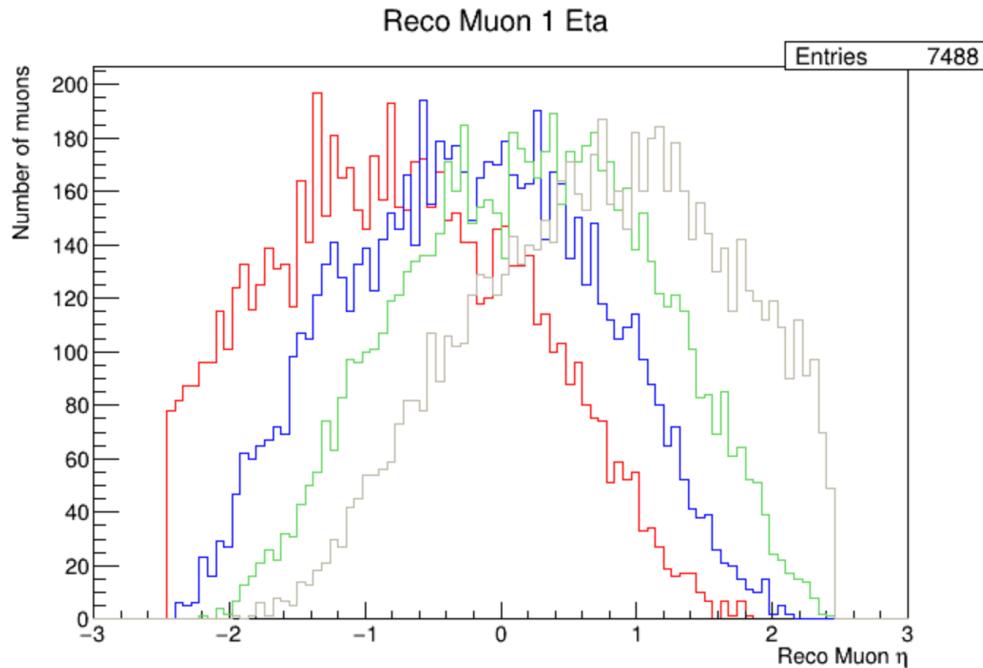
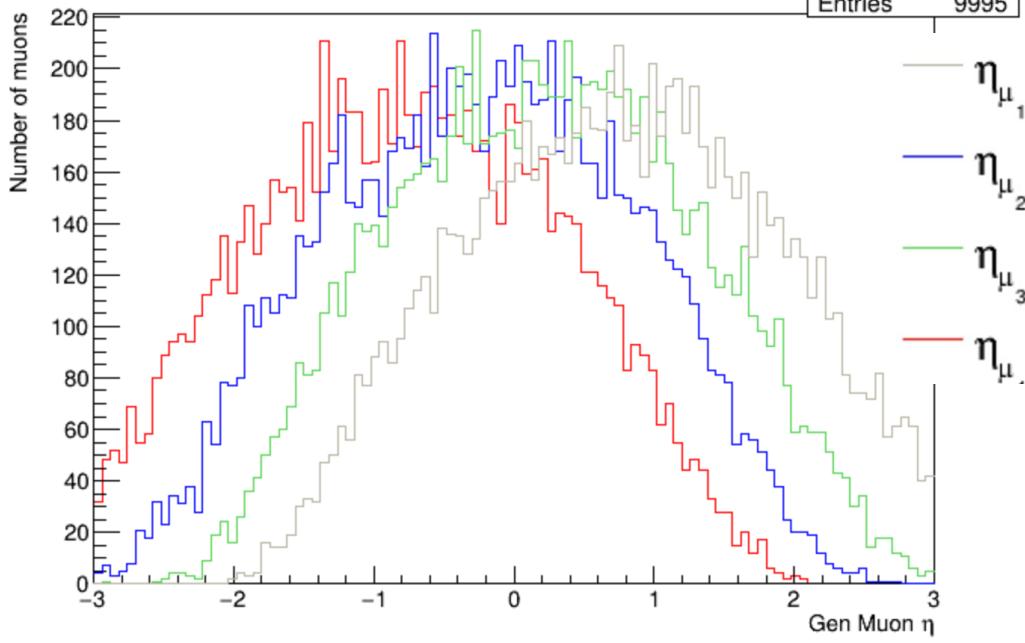
Note2

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



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:

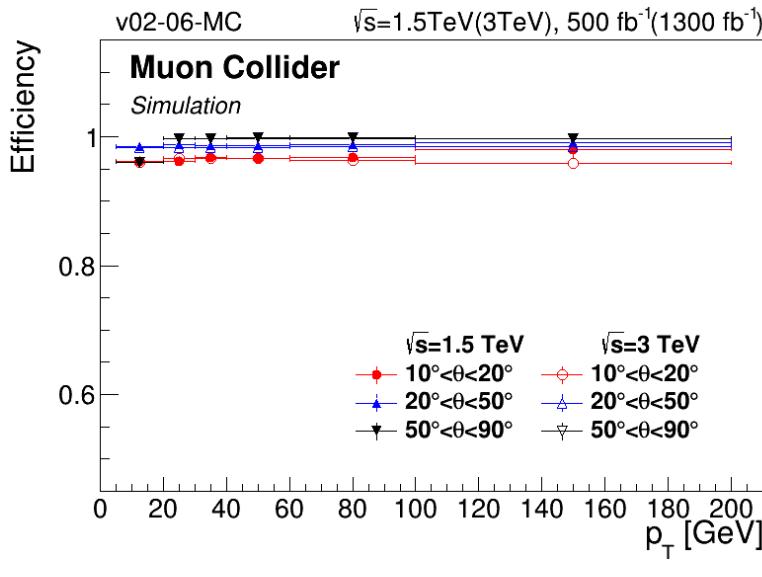
-detector effects are not accounted
 -eta estimation is extrapolated from muon trajectory (uncertainty on R is propagated)

Instead, generated muon eta just come from phase space

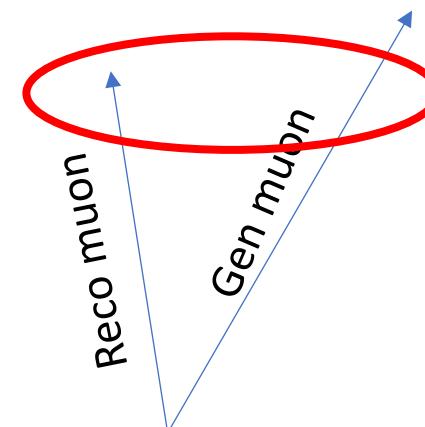
Muon reconstruction performance

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



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

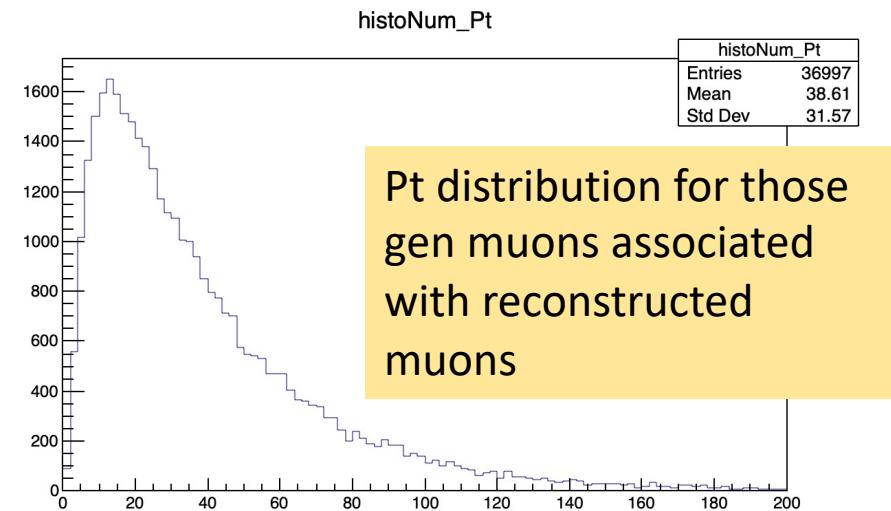
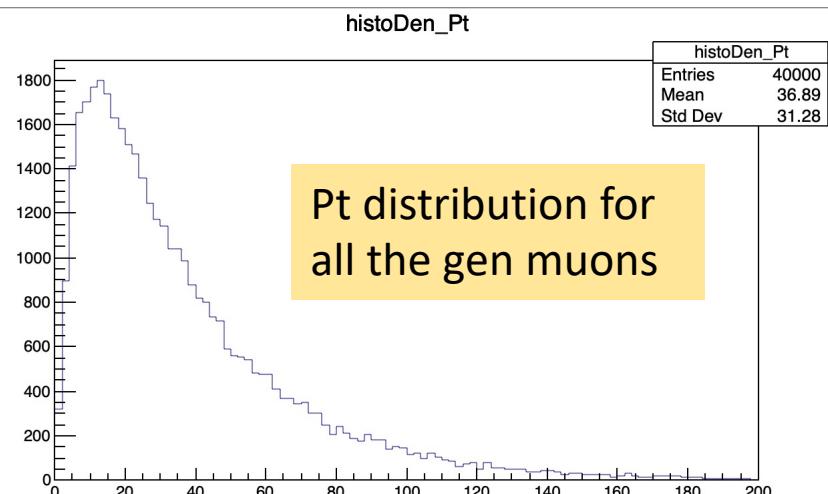
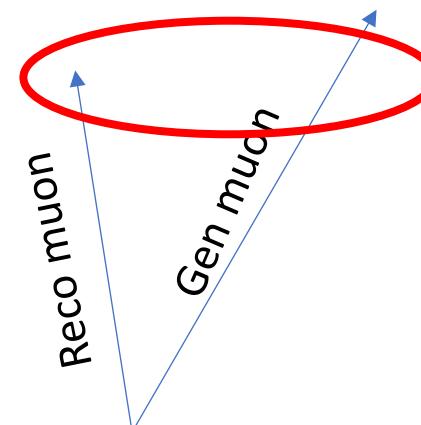
Efficiency

#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



Muon reconstruction performance

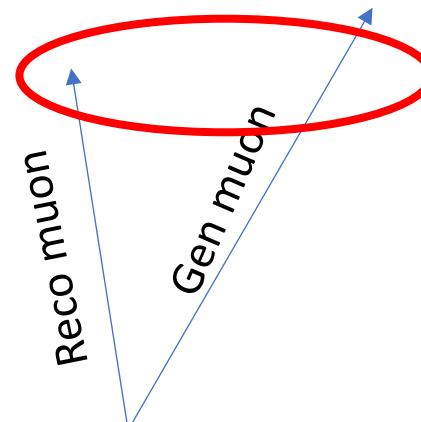
Efficiency

*#Reconstructed muons associated to generated
#Generated muons*

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

Then:

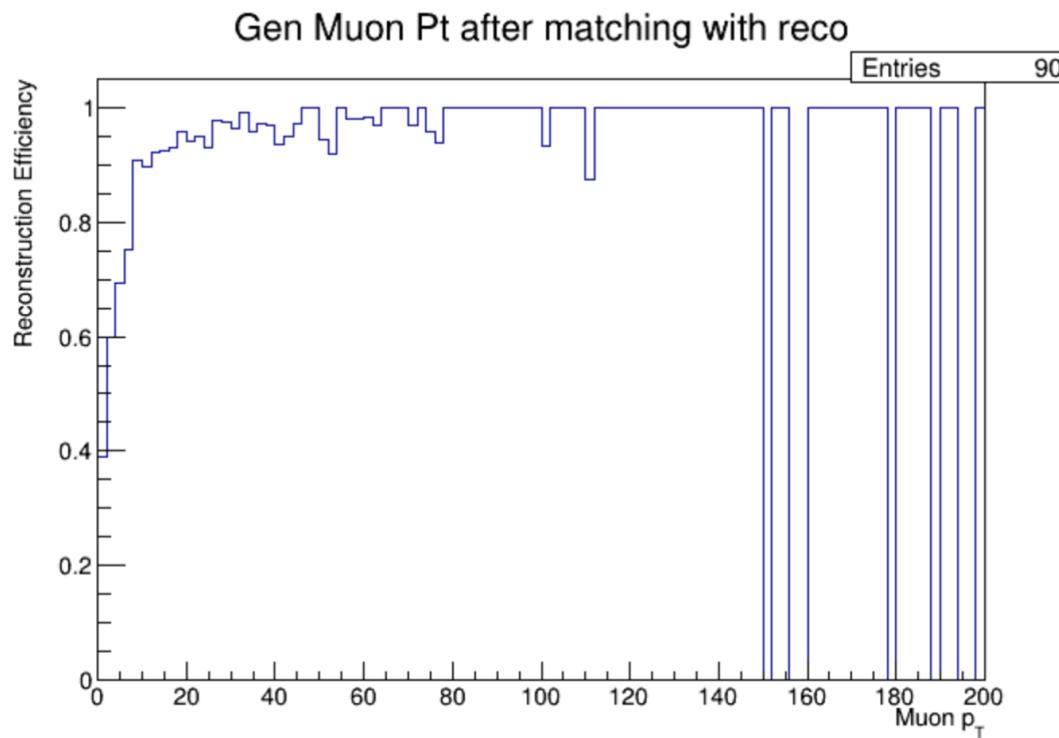
- Plot gen pT
- Plot reco pT of the associated particle
- Do the ratio



```
# GEN-RECO matching based on dR
#print("gen Eta",GenMuonList[0].Eta())
for k in range(len(GenMuonList)):
    histoDen_Pt.Fill(float(GenMuonList[k].Pt()));
    histoDen_Eta.Fill(float(GenMuonList[k].Eta()));
    dR = 1000
    dR_list = []
    for i in range(len(RecoMuonList)):
        thisDR=RecoMuonList[i].DrEtaPhi(GenMuonList[k])
        #print("thisdR=", thisDR)
        if(thisDR<dR):
            dR=thisDR
    #print("DR final", dR)
    if(dR<0.01):
        histoNum_Pt.Fill(float(GenMuonList[k].Pt()))
        histoNum_Eta.Fill(float(GenMuonList[k].Eta()))
```

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 è < 1
soprattutto per muoni
a basso p_T → significa
che non riusciamo a
ricostruire tutti i muoni
con $p_T < 20$ GeV

Perchè l'efficienza è più
bassa a basso p_T ?

Esercizio 2C – efficienza di ricostruzione

Error
Propagation
on ratio

$$y = x_1/x_2$$

$$\begin{aligned} \left(\frac{\partial y}{\partial x_1}\right)_{x_1=p_1} &= \left(\frac{1}{x_2}\right)_{x_1=p_1 \neq 0}, \quad \left(\frac{\partial y}{\partial x_2}\right)_{x_1=p_1} = \left(\frac{-x_1}{x_2^2}\right)_{x_1=p_1 \neq 0} \\ \sigma_y^2 &\approx \left[\frac{1}{x_2} \cdot \frac{1}{x_2} \cdot \nu_{11} + \frac{1}{x_2} \cdot \frac{-x_1}{x_2^2} \nu_{12} + \frac{-x_1}{x_2^2} \cdot \frac{1}{x_2} \nu_{21} + \frac{-x_1}{x_2^2} \cdot \frac{-x_1}{x_2^2} \nu_{22} \right]_{x_1=p_1} \sigma_x^2 \\ &\approx \left[\frac{\sigma_x^2 x_1^2}{x_2^2 x_2^2} - \frac{x_1^2}{x_2^3} \frac{\nu_{12}}{x_2} + \frac{x_1^2}{x_2^4} \frac{\sigma_x^2}{x_2} \right]_{x_1=p_1} \\ &\approx \left[\left(\frac{x_1}{x_2} \right)^2 \left(\frac{\sigma_x^2}{x_1^2} + \frac{\sigma_x^2}{x_2^2} - 2 \frac{\nu_{12}}{x_1 x_2} \right) \right]_{x_1=p_1} \\ \text{Conversione: } &\left[\frac{\sigma_y^2}{y^2} = \frac{\sigma_x^2}{x_2} \right]_{y=p_1 \neq 0} \approx \left[\frac{\sigma_x^2}{x_2^2} + \frac{\sigma_x^2}{x_2^2} - 2 \frac{\nu_{12}}{x_1 x_2} \right]_{x_1=p_1 \neq 0} \\ \text{Se poi } x_1, x_2 \text{ non sono corretti allora:} & \\ \left[\frac{\sigma_y^2}{y^2} = \frac{\sigma_x^2}{x_2} \right]_{y=p_1 \neq 0} &\approx \left[\frac{\sigma_x^2}{x_2^2} + \frac{\sigma_x^2}{x_2^2} \right]_{x_1=p_1} \\ \Rightarrow &\left[\text{Le deviazioni standard relative si sommano in quadratura} \right] \end{aligned}$$

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

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

$$\epsilon = n_{\text{sel}} / n_{\text{tot}}$$

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

$$P_B(x; n, p) = \binom{n}{x} p^x q^{n-x} = \frac{n!}{x!(n-x)!} p^x (1-p)^{n-x} \quad \begin{array}{l} p \rightarrow \epsilon \\ n \rightarrow n_{\text{tot}} \\ x \rightarrow n_{\text{sel}} \end{array}$$

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

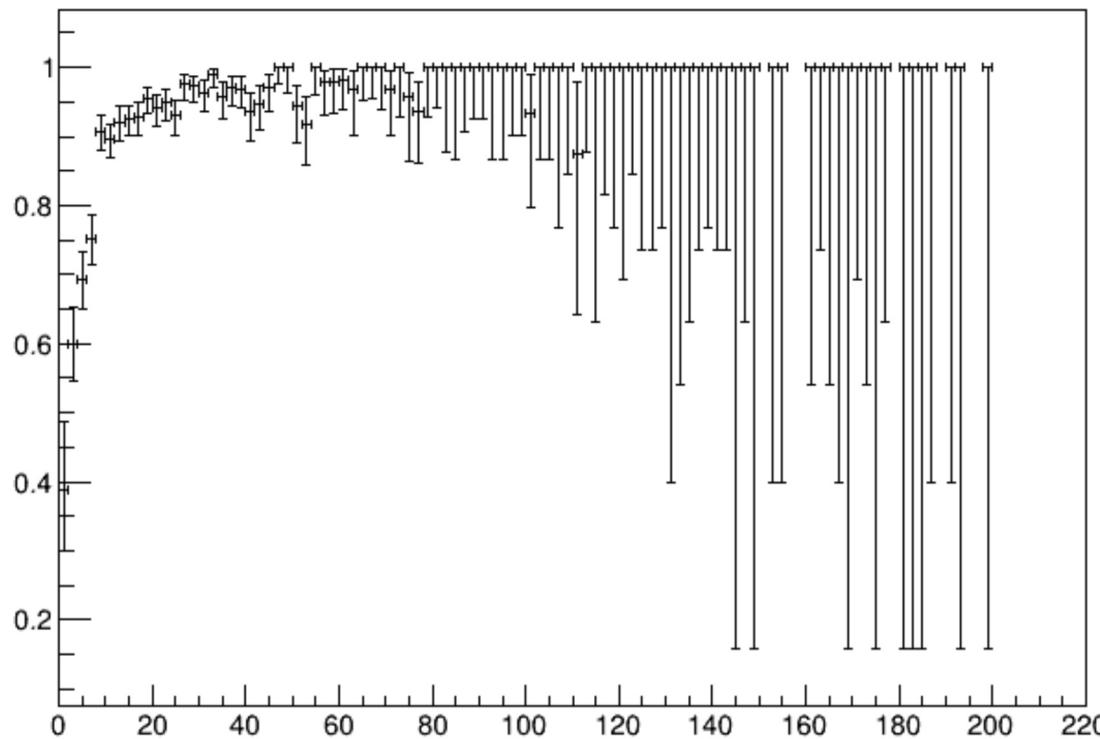
Note: $\sigma_\epsilon = 0$ for $\epsilon = 0, 1 \rightarrow$ sometimes the error on efficiency are computed in other ways

20

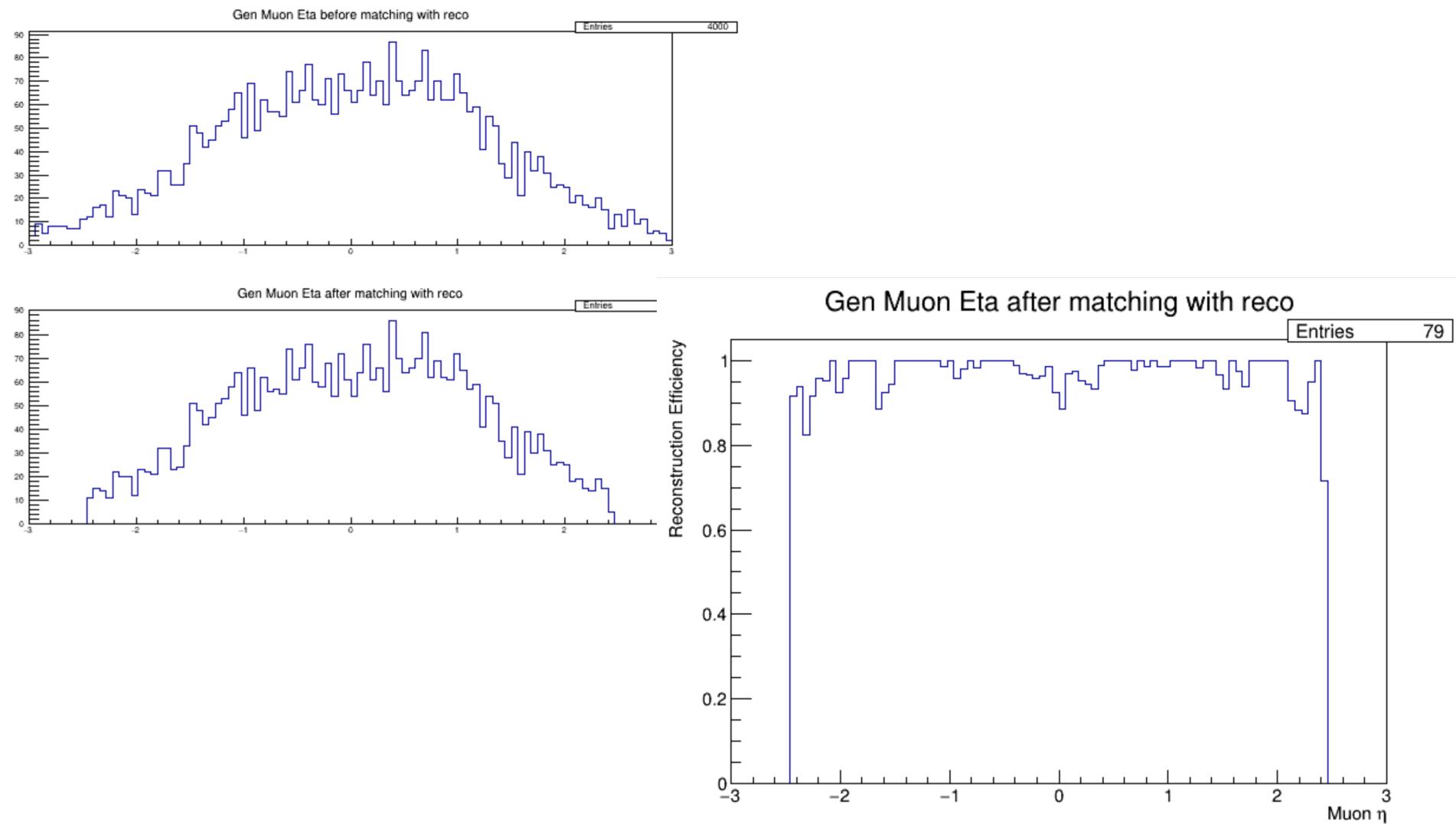
La classe Tefficiency calcola le incertezze sull'efficienza tenendo conto del fatto che l'efficienza è una variabile random che segue una binomiale!

Esercizio 2C – efficienza di ricostruzione

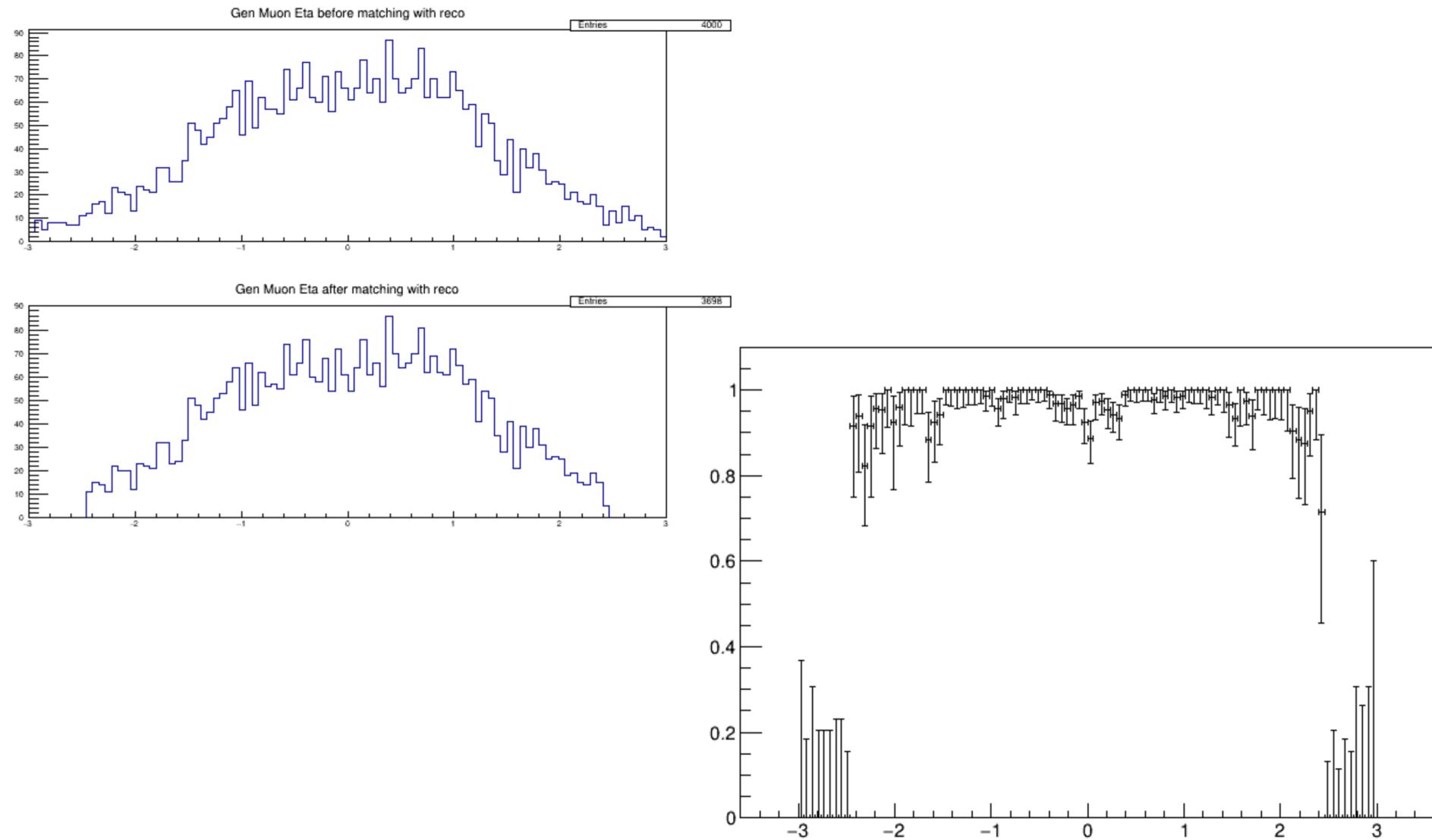
- Decommentate nella macro il calcolo dell'efficienza con Tefficiency
- Cosa notate?
- Plottare l'efficienza vs eta



Esercizio 2C – efficienza di ricostruzione

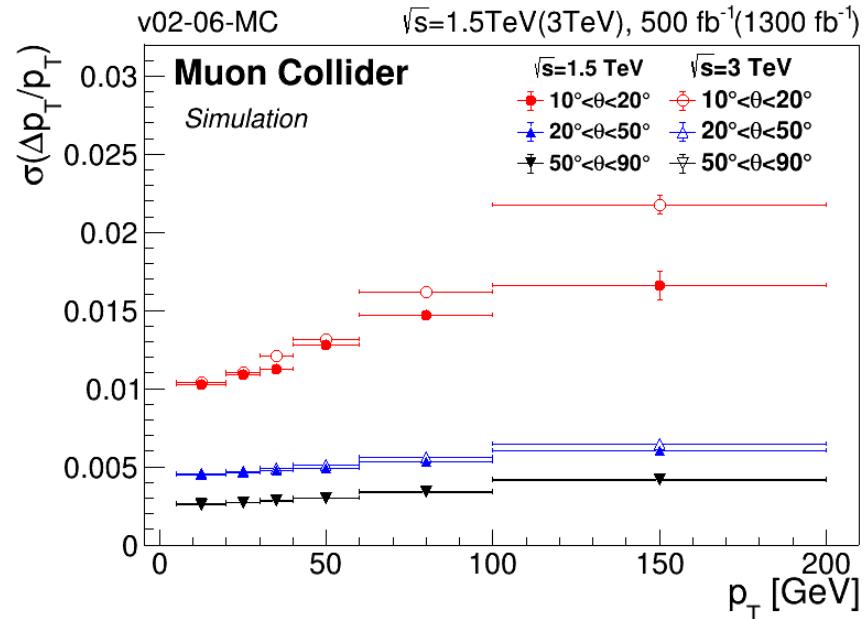
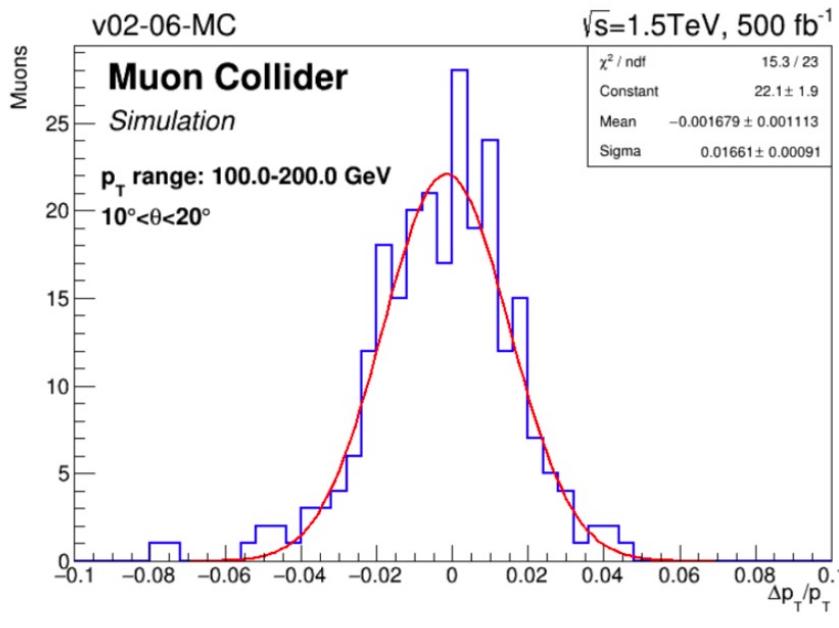


Esercizio 2C – efficienza di ricostruzione



Muon momentum resolution

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

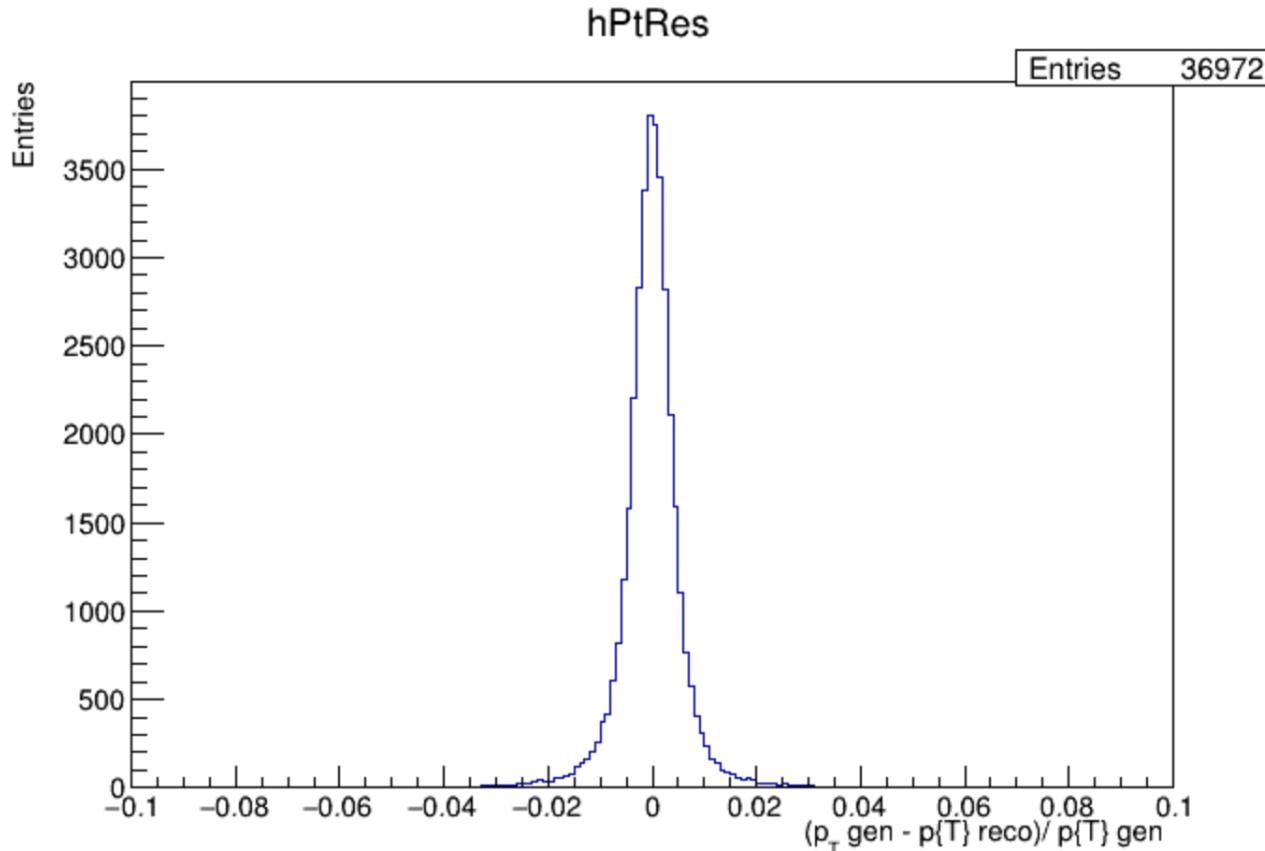
Hint: look at the mean values of Gen and reco Muon p_T distributions

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

Esercizio 3 – muon momentum resolution

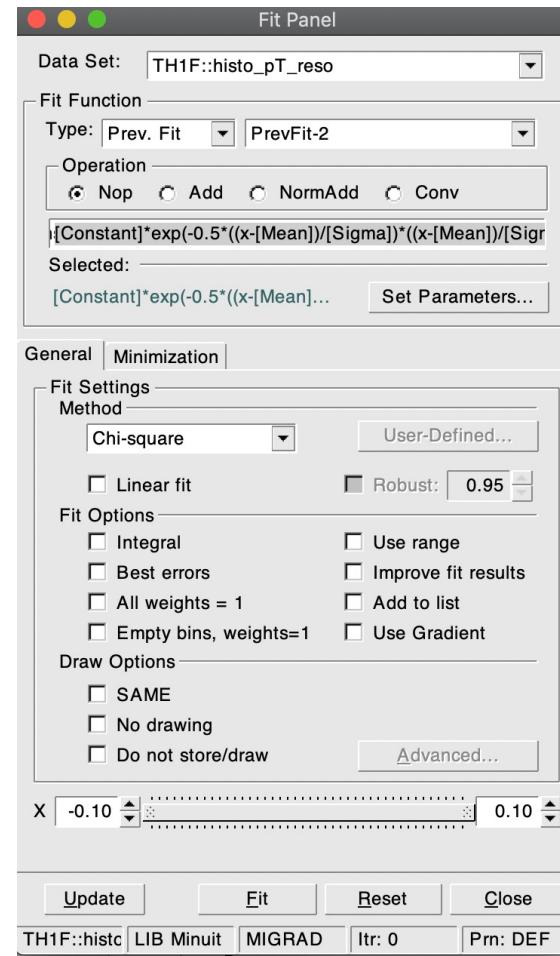
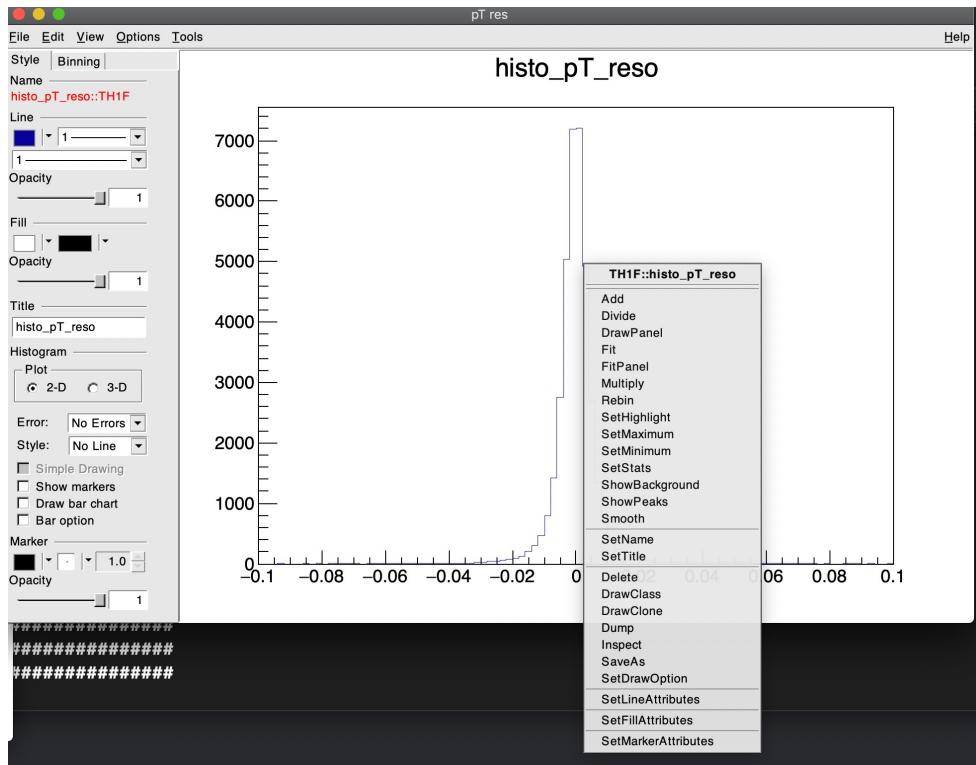
```
c7 = TCanvas( 'c72', 'Muon momentum resolution', 200, 10, 700, 500 )
c7.cd();
hPtRes.GetXaxis().SetTitle("(p_{T} gen - p{T} reco)/ p{T} gen ");
hPtRes.GetYaxis().SetTitle("Entries");
hPtRes.Draw()
c7.Draw()
c7.SaveAs("reso.root")
```

Info in <TCanvas::SaveAs>: ROOT file reso.root has been created

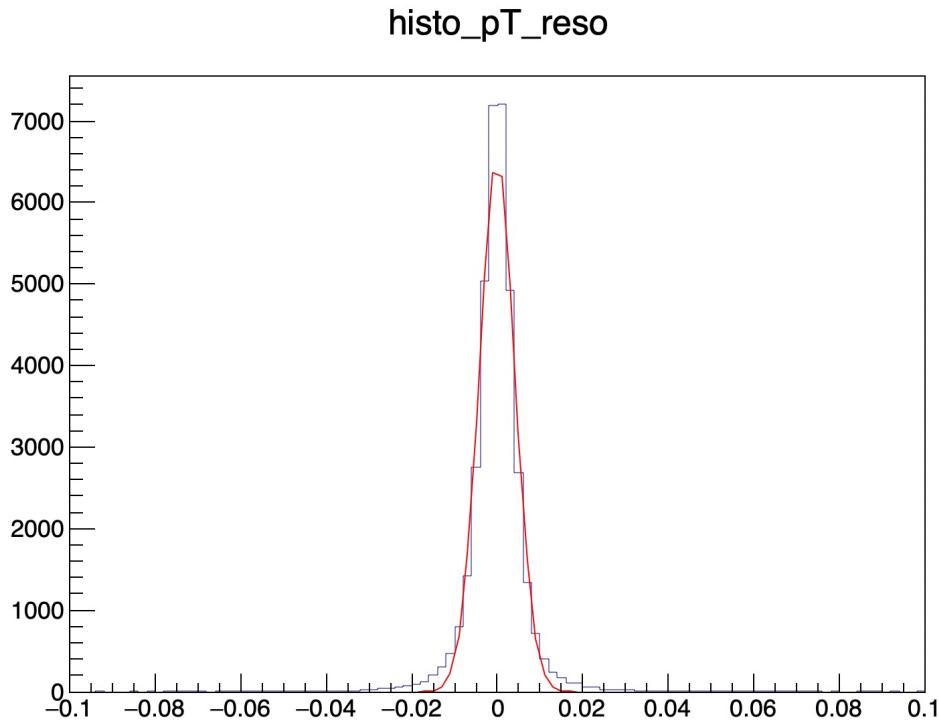


Esercizio 3 – muon momentum resolution

- Fit the histogram with a gaussian using ROOT Browser
 - Click on the histogram line, this will open a menu → choose fit panel
- The muon momentum resolution is given by the sigma of the gaussian



Esercizio 3 – muon momentum resolution



```
root [1] TFitEditor::DoFit - using function PrevFitTMP 0x7fb5fa758770
FCN=2504.75 FROM MIGRAD      STATUS=CONVERGED            98 CALLS          99 TOTAL
                           EDM=3.48579e-10    STRATEGY= 1    ERROR MATRIX ACCURATE
EXT PARAMETER                      STEP                      FIRST
NO.   NAME        VALUE       ERROR        SIZE      DERIVATIVE
 1  Constant     6.53302e+03  5.11924e+01  8.60154e-01 -1.40090e-07
 2  Mean         -6.83109e-05 2.26887e-05  5.54154e-07  2.54186e-02
 3  Sigma         4.20879e-03  2.39430e-05  2.44740e-05 -2.13575e-02
```

Backup

Interlude – collider frame

