Numerical Methods
Lesson 7

Dr. Jose Feliciano Benitez

Universidad de Sonora

Dr. Benitez Homepage: www.jfbenitez.science
Course page: http://jfbenitez.ddns.net:8080/Courses/MetodosNumericos

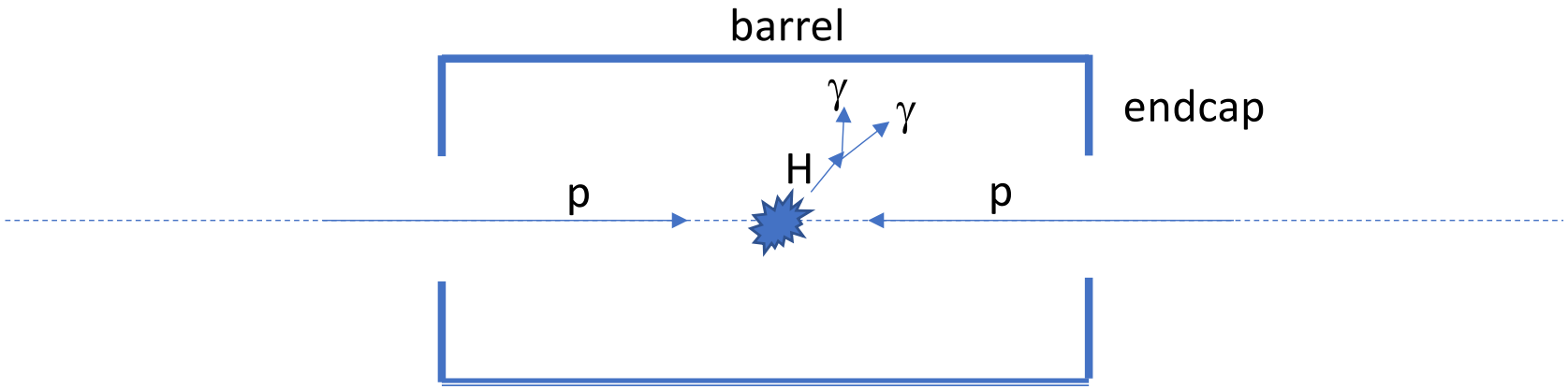
Goals for this Lesson

- Simulate signal process at particle collider
- Simulate detector acceptance and response
- Graph simulated and reconstructed quantities
- Understand detector efficiency and resolution
- Background simulation
- Signal extraction and statistical analysis

PART 1:
Signal Generation

Experimental setup

- Consider production of Higgs bosons at a proton collider (e.g. LHC)



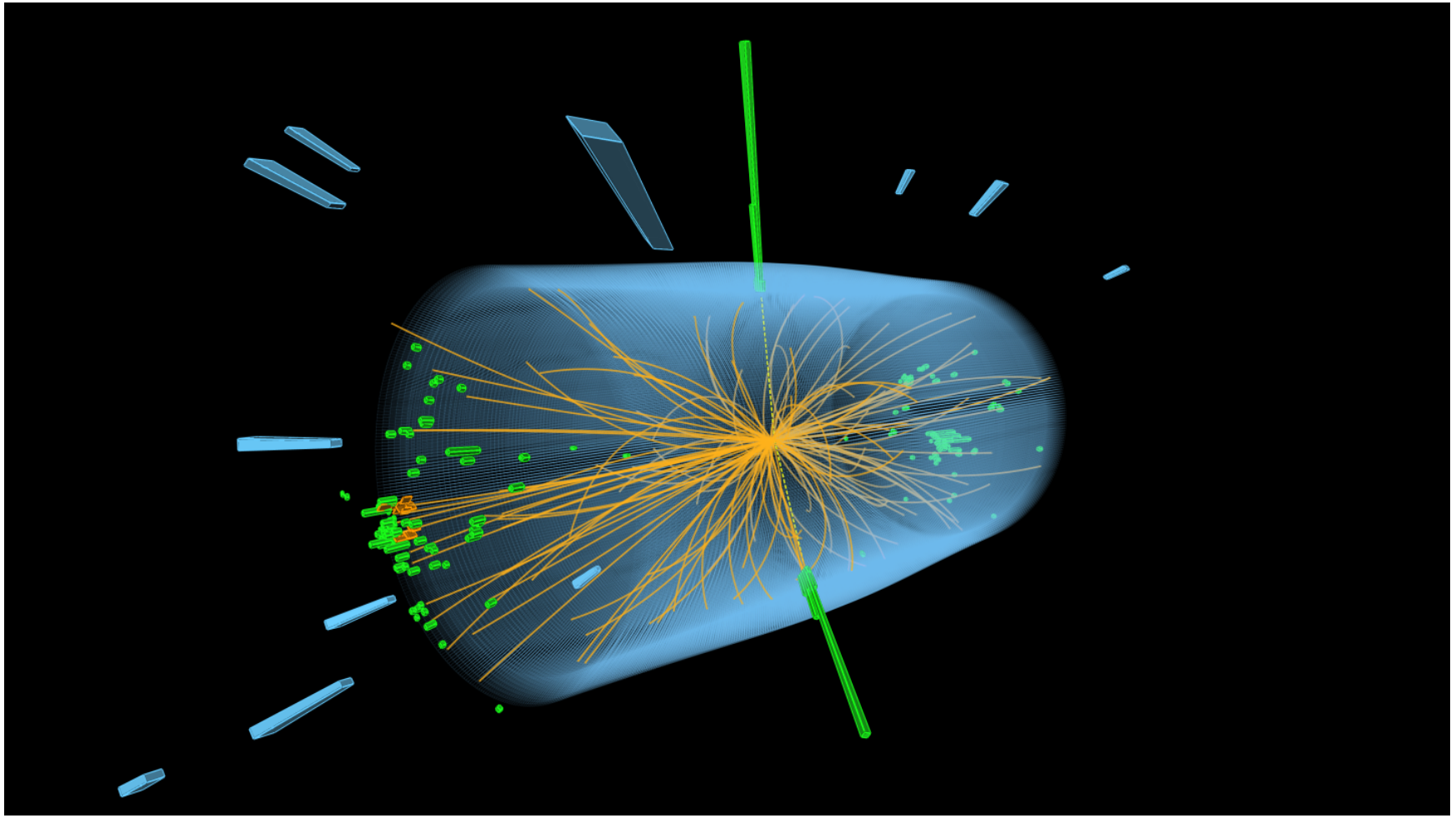
Define pseudorapidity:
$\eta = -\ln[\tan(\theta/2)]$

Barrel length = 4 m
Endcap r1 = 20 cm, r2 = 1 m

Center of mass energy : 13 TeV

Real event display

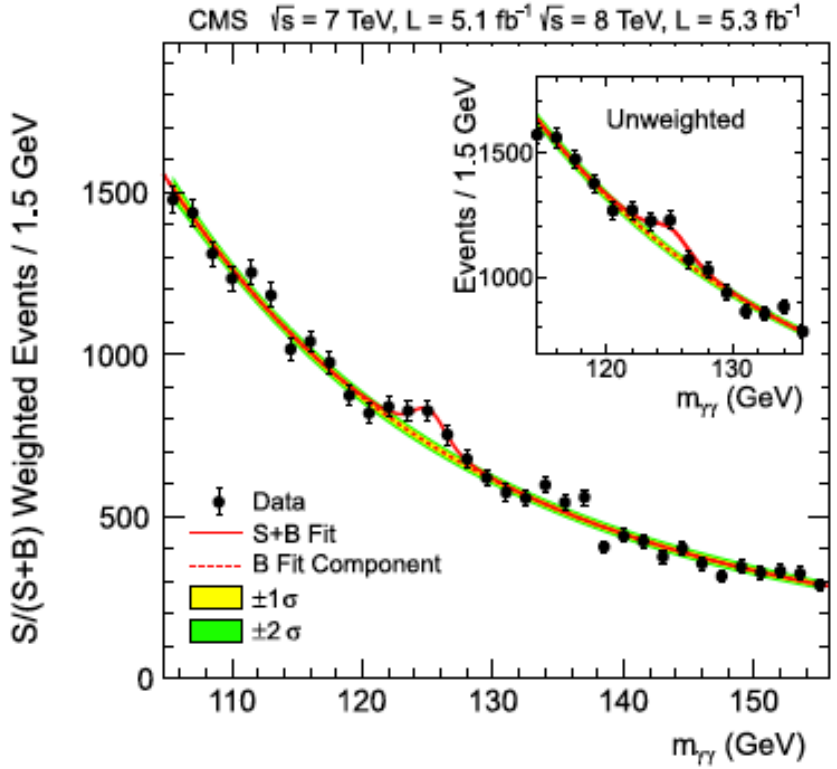
LHC-CMS Experiment



Some interesting questions

- How many Higgs events are produced?
- What is the efficiency to detect the decay products and the Higgs?
- How to reconstruct the signal how does it look like?
- How much background is there and how to estimate it?
- How to measure the signal amount and its uncertainty?
- How to compare to the the theoretical model?

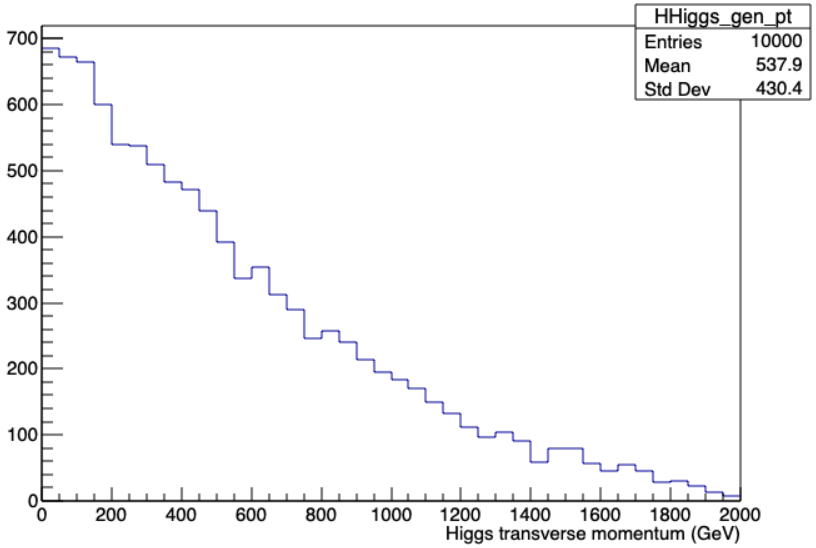
Discovery of the Higgs boson [CMS experiment]



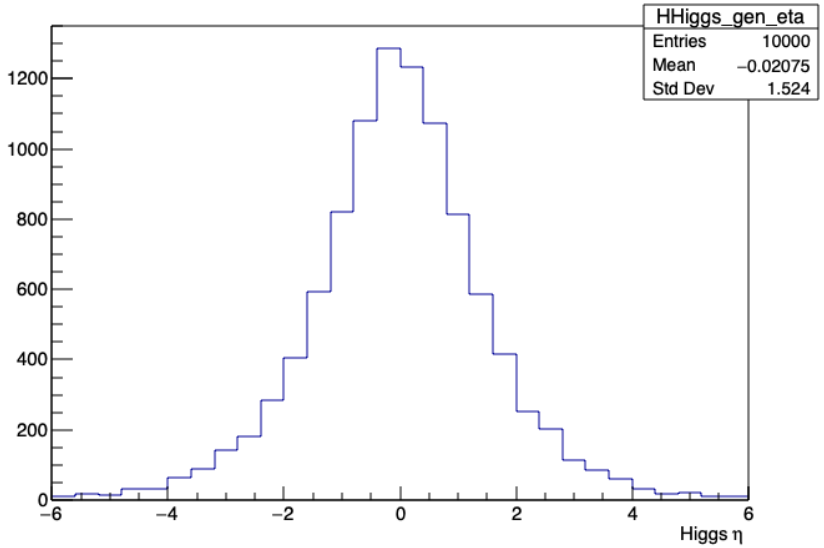
Physics Letters B 716 (2012) 30–61

Simulation Step 1: Higgs production (2D example)

- Generate Higgs particles according to known model predicting kinematic distributions.



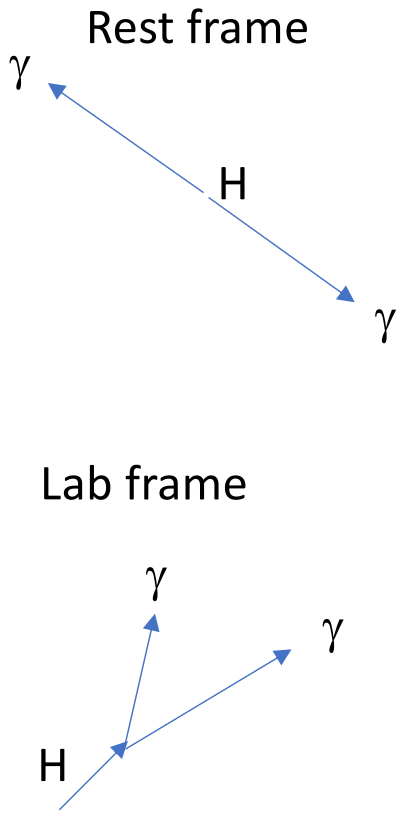
Higgs Transverse momentum



Higgs pseudo-rapidity

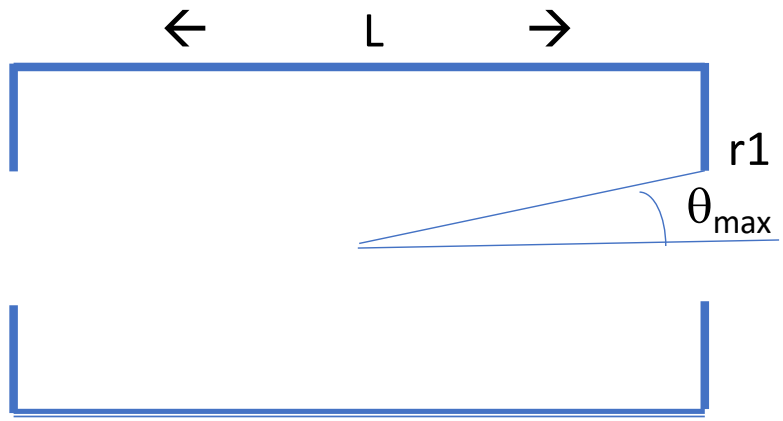
Step 2: Higgs decay

- Higgs is a scalar particle, in this exercise we consider the decay to two photons.
- The decay products in the Higgs rest frame have uniform angular distribution.
- To obtain the momentum vectors of the photons we apply a boost using the Higgs momentum vector.



Step 3: Detector acceptance

- The detector does not cover completely
- We must reject photons outside the detector coverage.
- Photons with $|\eta| > \eta_{max}$ must be rejected, consequently the event must be rejected as the Higgs cannot be reconstructed.



θ_{max} = Pi/2 – arctan(L/(2*r1))

η_{max} = = −ln[tan(θ_{max} /2)]

Step 4: Sensor efficiency

- Even if the photon is inside the detector acceptance, it may not be detected due to sensor inefficiencies:
  - Gaps or dead regions between sensors,
  - Electronics inefficiency due to noise rejection.
- To simulate this effect, we throw a uniform random number between 0 and 1 and reject the photon if this random number is above the known sensor efficiency (here assume 95%)

Step 5: Sensor resolution

- Once we decide the photon is detected we must assign its measured quantities like position and energy
- In this exercise we assume the energy resolution is 1% , usually better than this for energetic photons.
- For spatial resolution, usually the sensor has good granularity leading to very good resolution. In this example we assume 0.02 resolution on the pseudorapidity variable.

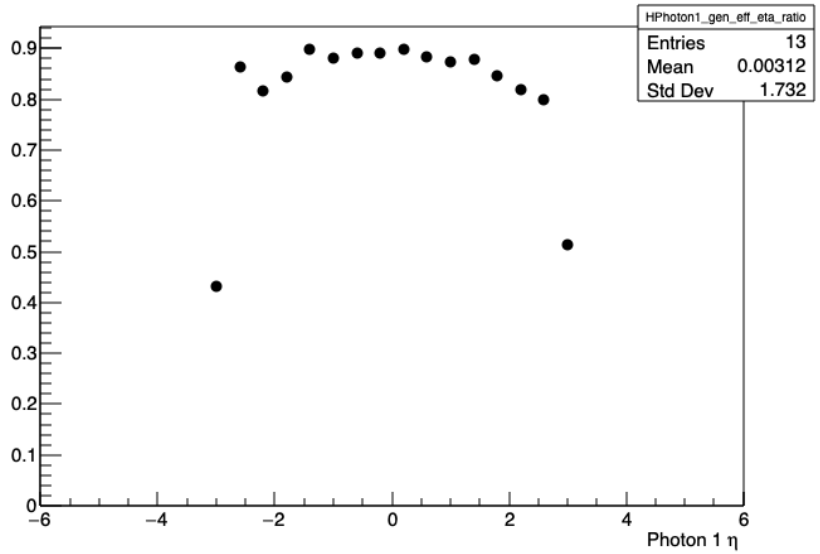
Exercise

- Update the git repository:
  ```
  cd ~/MetodosNumericosCourse
  git pull
  ```
- Read and run the Lesson7 code:
  ```
  cd Lesson7
  root -b -q generate_signal.C\(1000\)
  ```
  this creates a file containing the simulated events.
- Read the data and plot the distributions:
  ```
  root -b -q plot_signal.C
  ```
- Exercise is to complete the reconstruction of the Higgs using the reconstructed photon quantities.
- Publish the graphs in the:

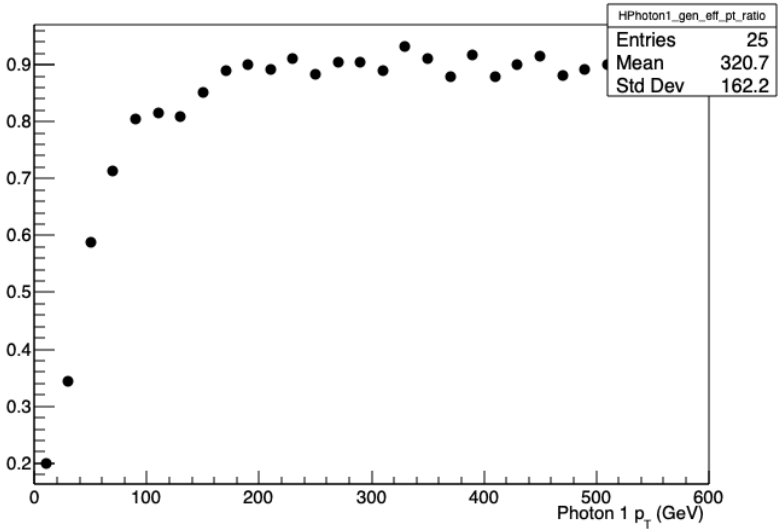
PART 2:
Event reconstruction efficiency and resolution

Photon Efficiency

- Efficiency vs eta

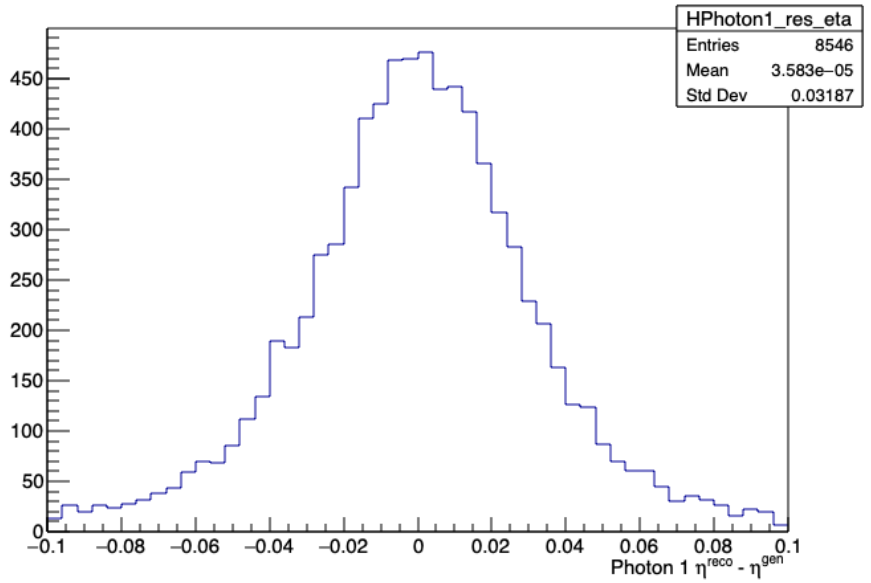


- Efficiency vs pT

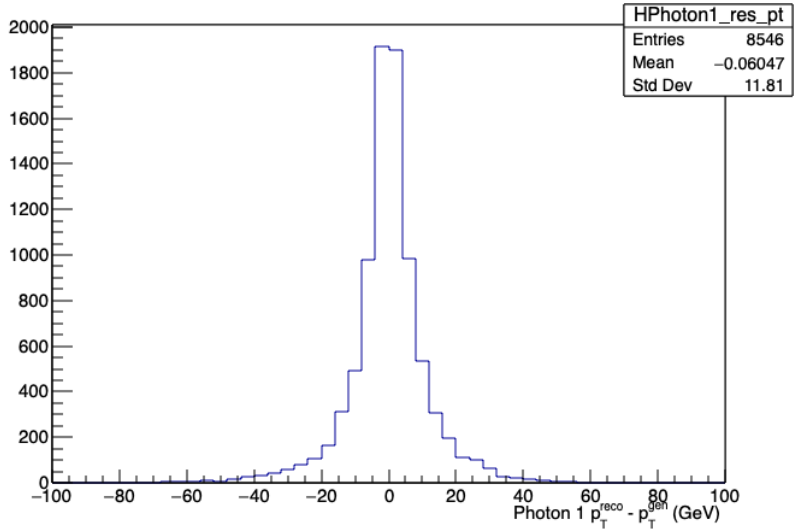


Photon resolution

- Eta resolution



- PT resolution



Higgs reconstruction

- Measured quantities:
  - `Photon 1 : e1, theta1, phi1`
  - `Photon 2 : e2, theta2, phi2`

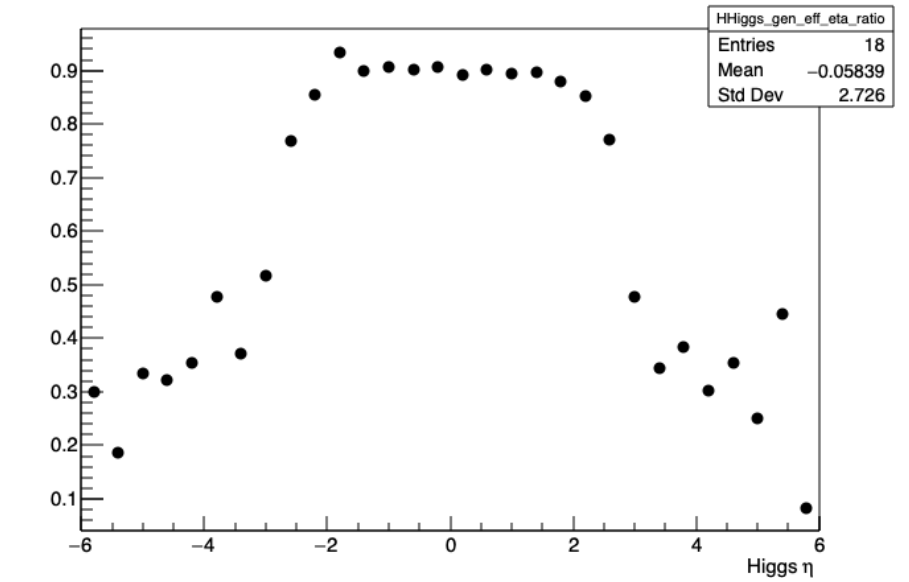
```
TLorentzVector Ph1;
Ph1.SetPx(e1*sin(theta1)*cos(phi1));
Ph1.SetPy(e1*sin(theta1)*sin(phi1));
Ph1.SetPz(e1*cos(theta1));
Ph1.SetE(e1);   //assume massless particle
TLorentzVector Ph2; ….
```

- Higgs:

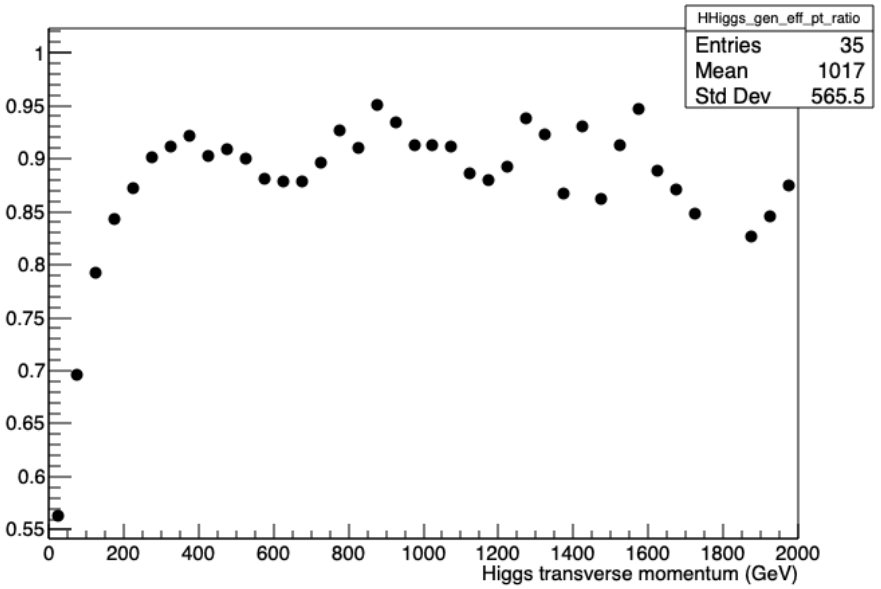
```
TLorentzVector H = Ph1 + Ph2;
```

Higgs Efficiency

- Efficiency vs eta

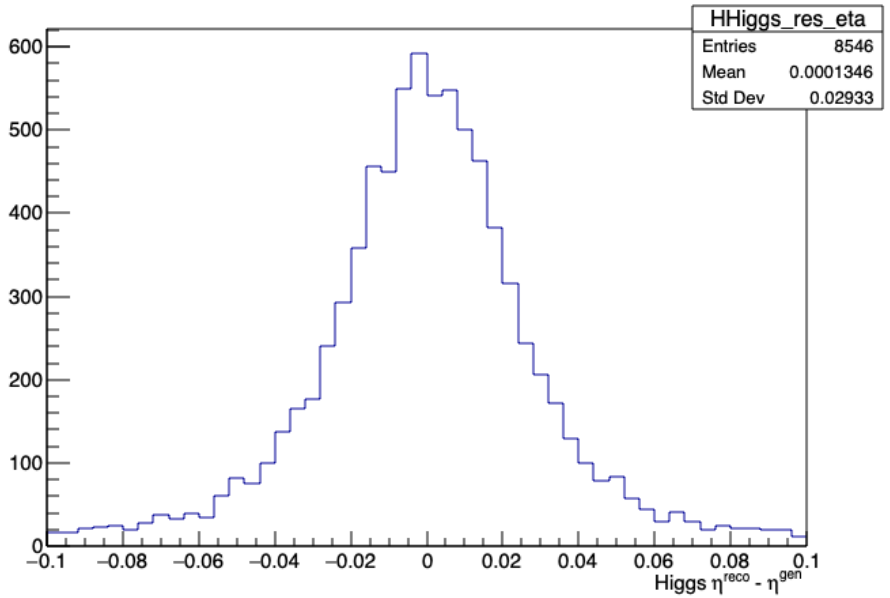


- Efficiency vs pT

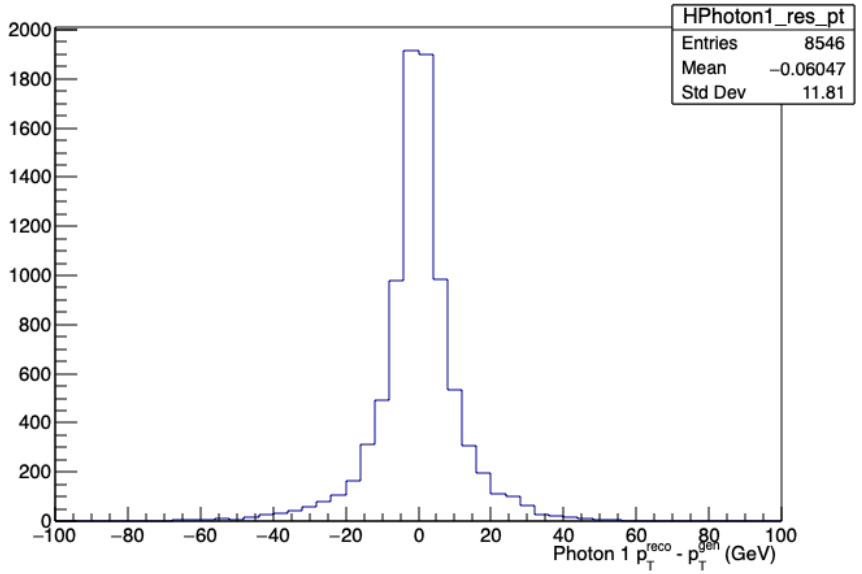


Higgs resolution

- Eta resolution

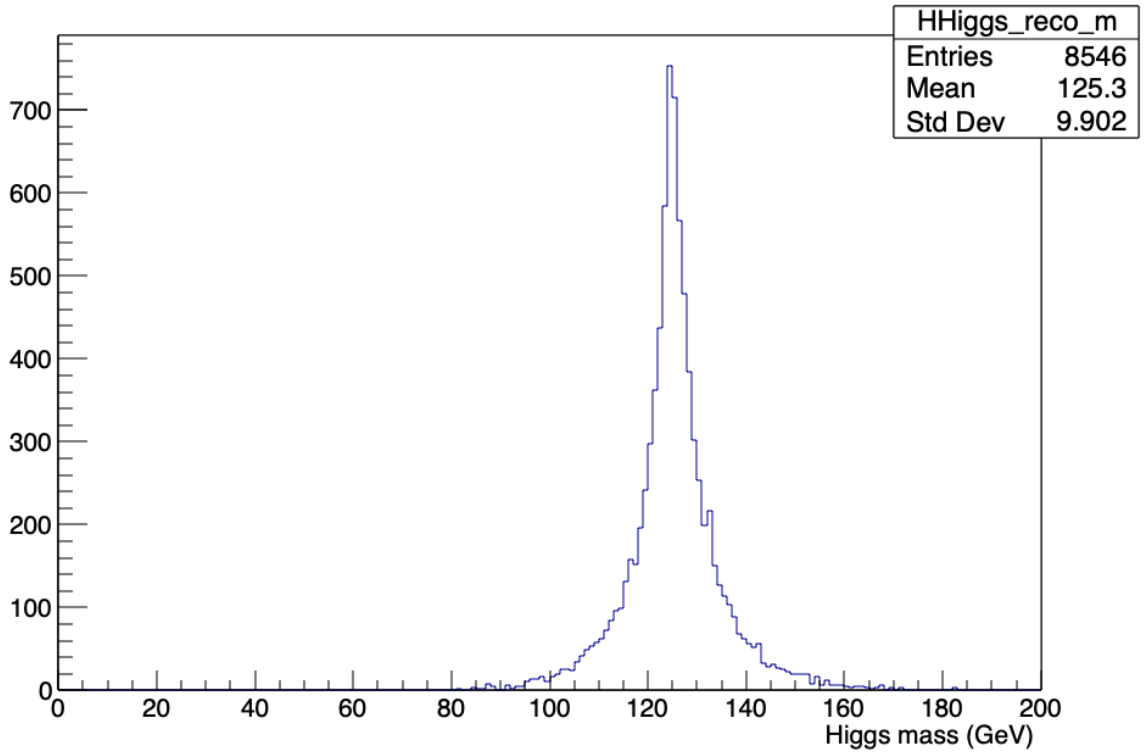


- PT resolution



Higgs mass

- Absolute efficiency
- Resolution



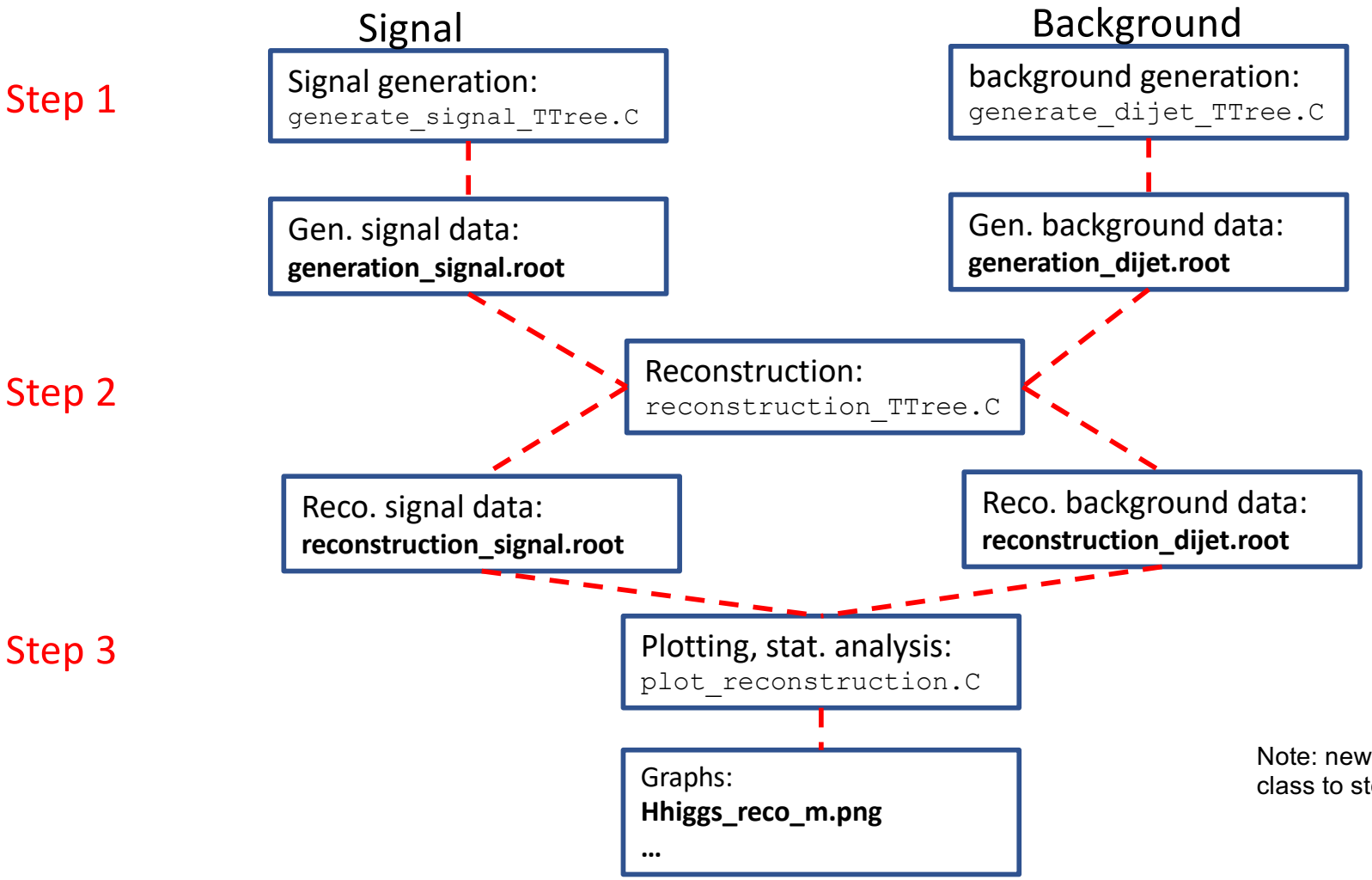
exercise

- Calculate efficiency of photons vs eta and pT using Photon 2.
- Publish graphs in the Lesson7 results folder.

PART 3:
Background simulation

Simulation code organization:

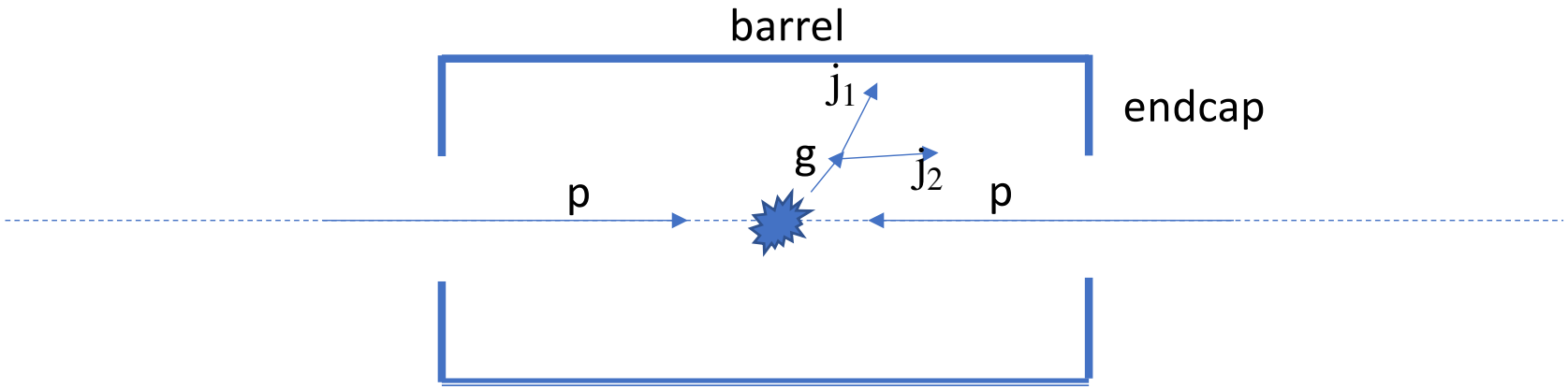
- https://github.com/benitezj/MetodosNumericosCourse , Lesson 7



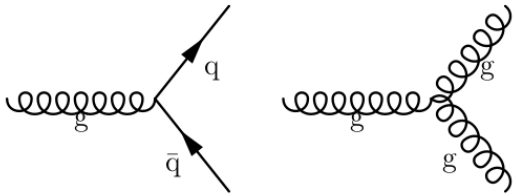
Note: new code uses the TTree class to store gen. events.

Dijet background

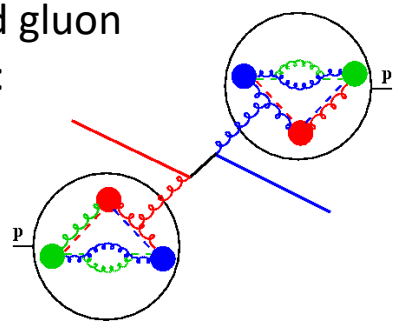
- Main background for the search of Higgs to two photons is dijet production.



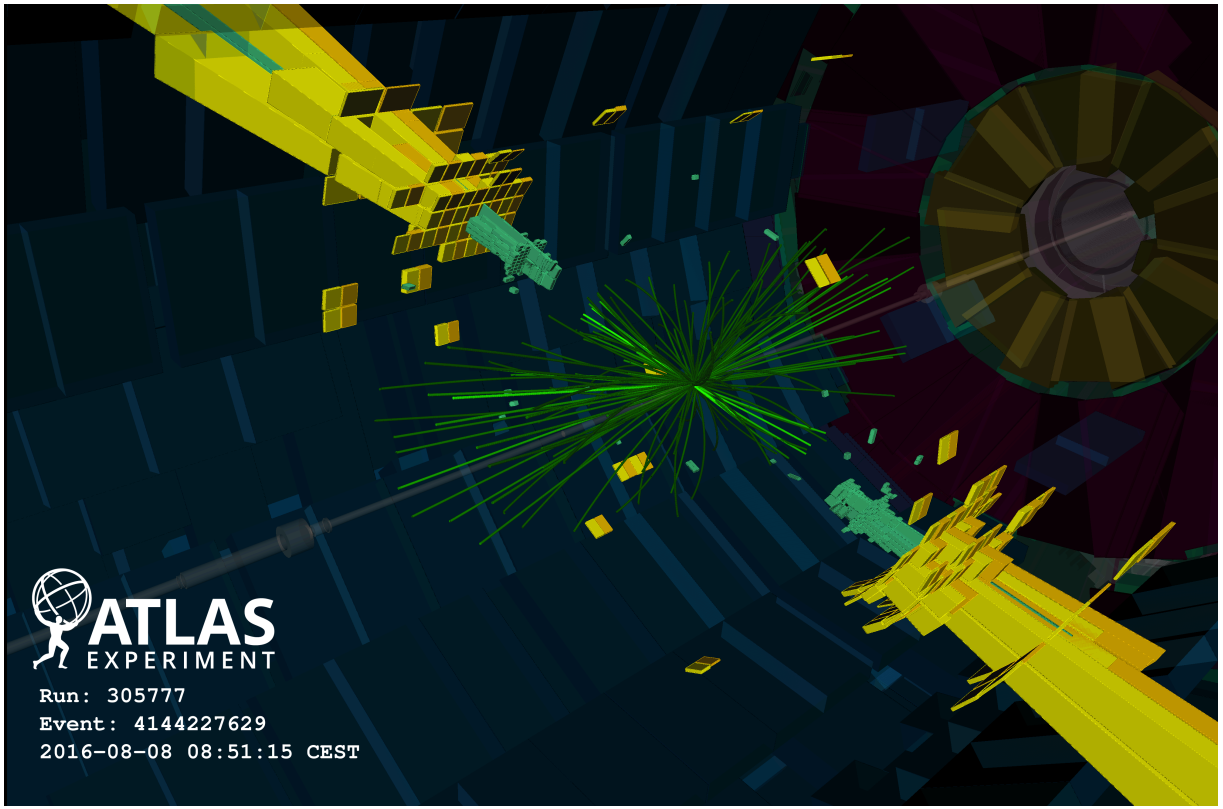
QCD interactions with large cross-section



Proton-proton collision and gluon interaction producing dijet:



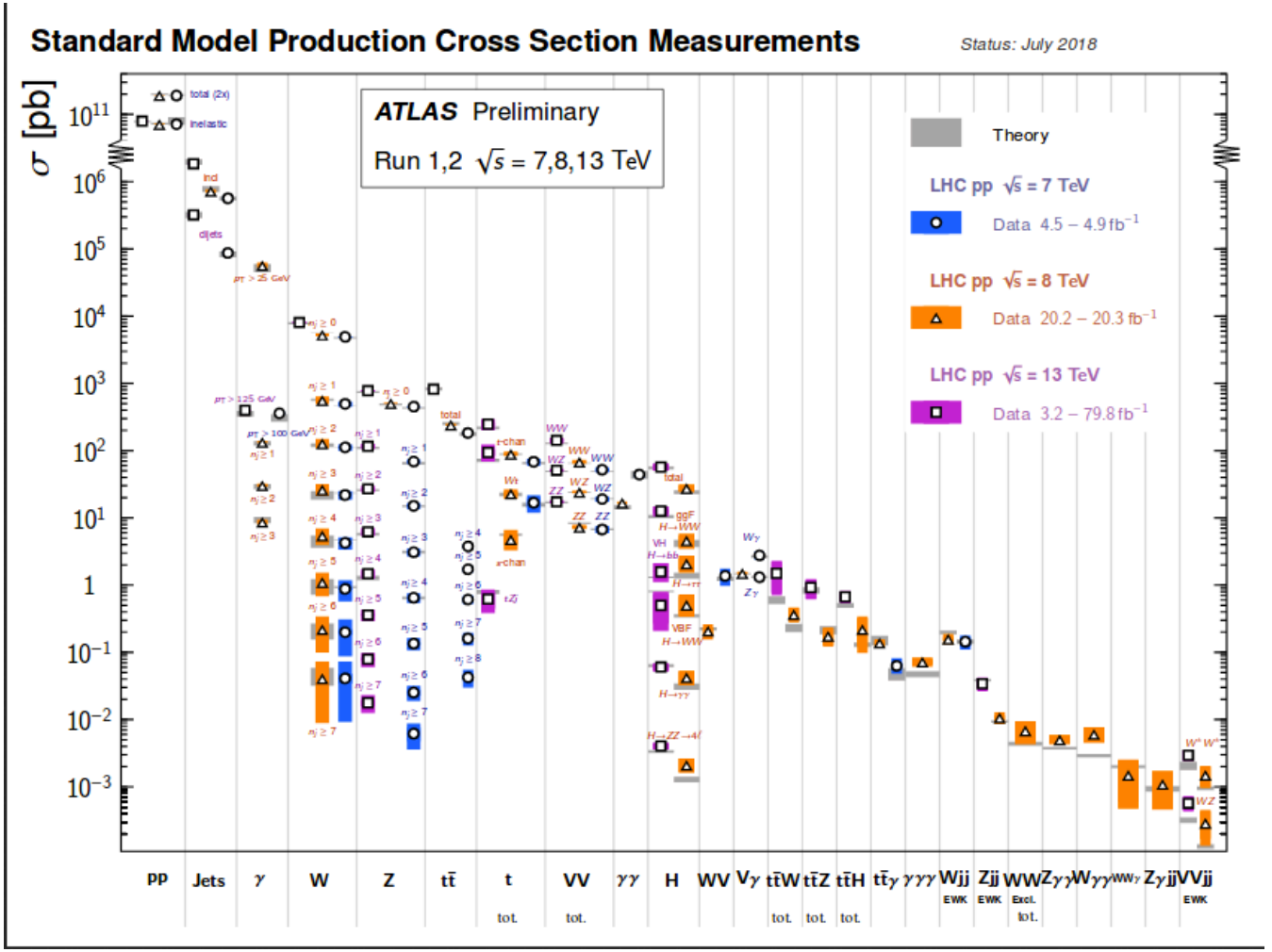
Real event display



- Jets create large energy deposits in the calorimeters. These can be mis-identified as photons due to mis-reconstruction of the isolation (energy around the jet axis).

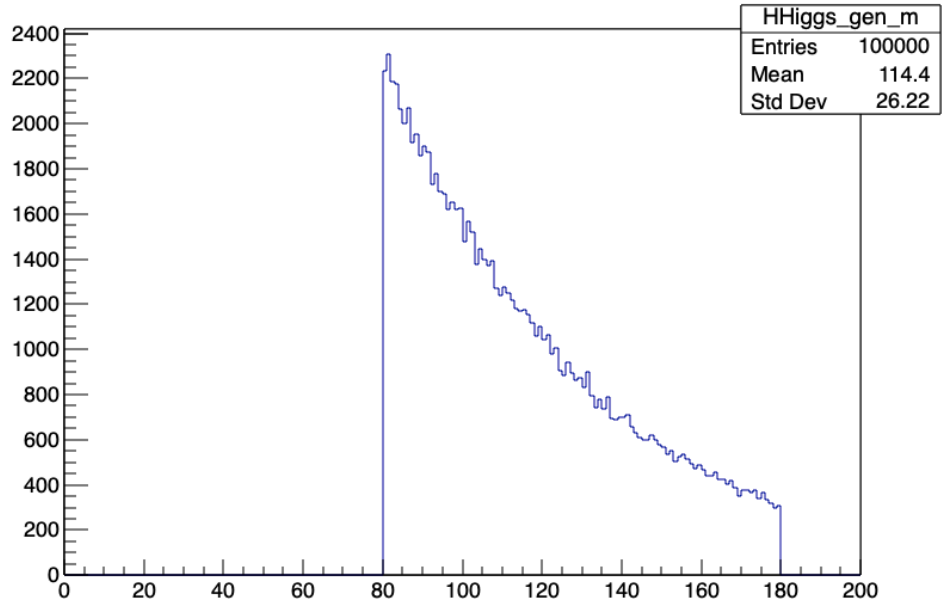
Production crossections

- At the LHC the probability to produce dijet QCD events has a σ_{dijet}**~ 300x10³ pb**
- The Higgs total production probability is σ_H**~ 50 pb** 0.00016 times smaller than dijet.



Virtual gluon mass distribution

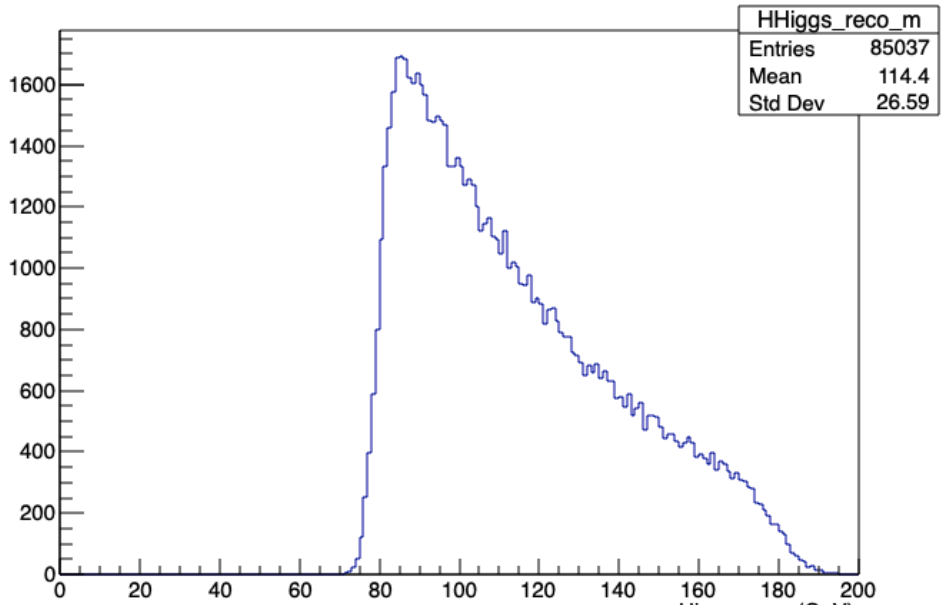
- The gluon is a massless particle, but is virtual particle in this process
- In this toy simulation the gluon mass distribution is simulated using an exponentially decaying distribution.



Distribution is truncated at low and high mass (80 – 180 GeV).

Reconstructed dijet mass distribution

- Apply same event reconstruction as for the signal diphoton events
- Assume same detection efficiency as for photons
- Assume 5% energy resolution for jet.

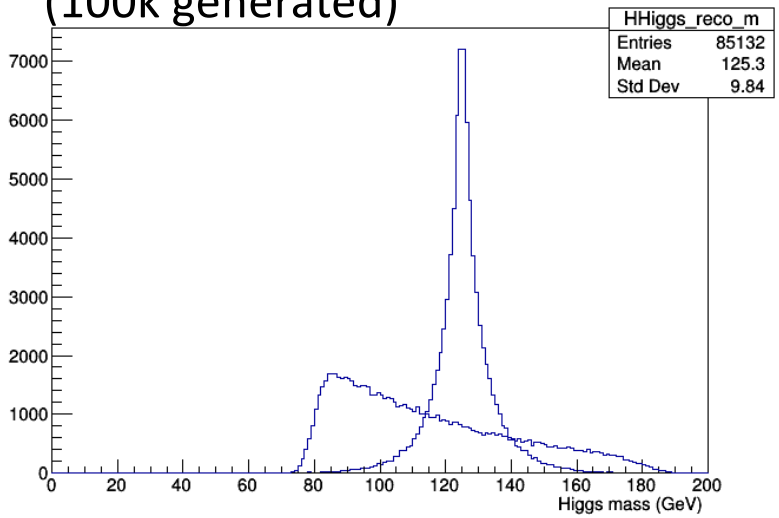


Distribution is truncated at low and high mass.

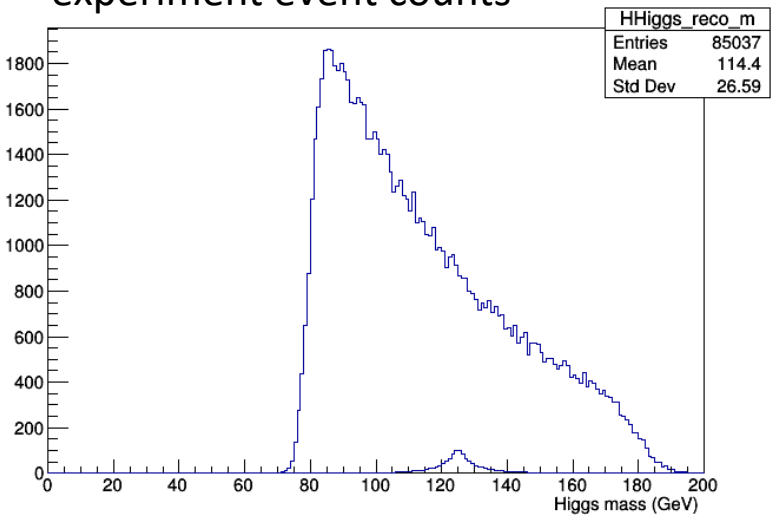
Combining signal and background

- In order to complete the simulation of the experiment we must combine signal and background.

Equal number of events
(100k generated)



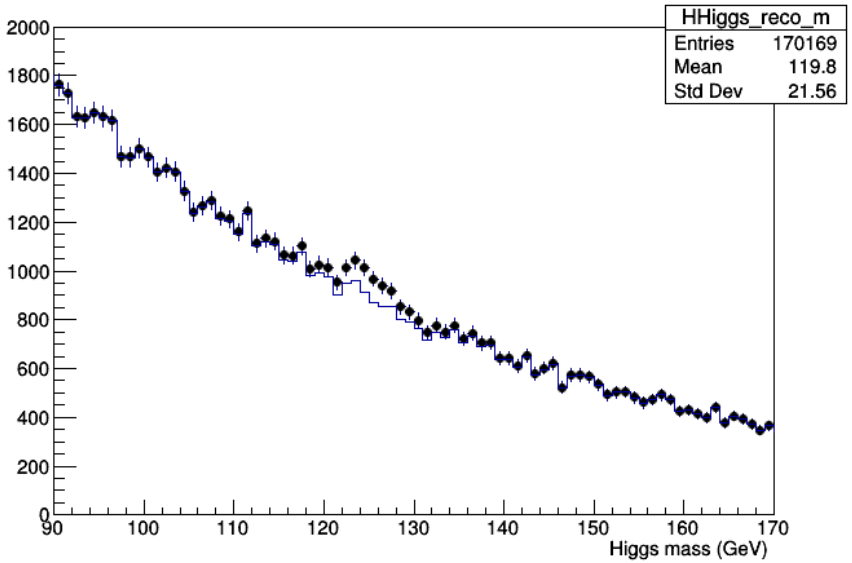
Distributions scaled to match
experiment event counts

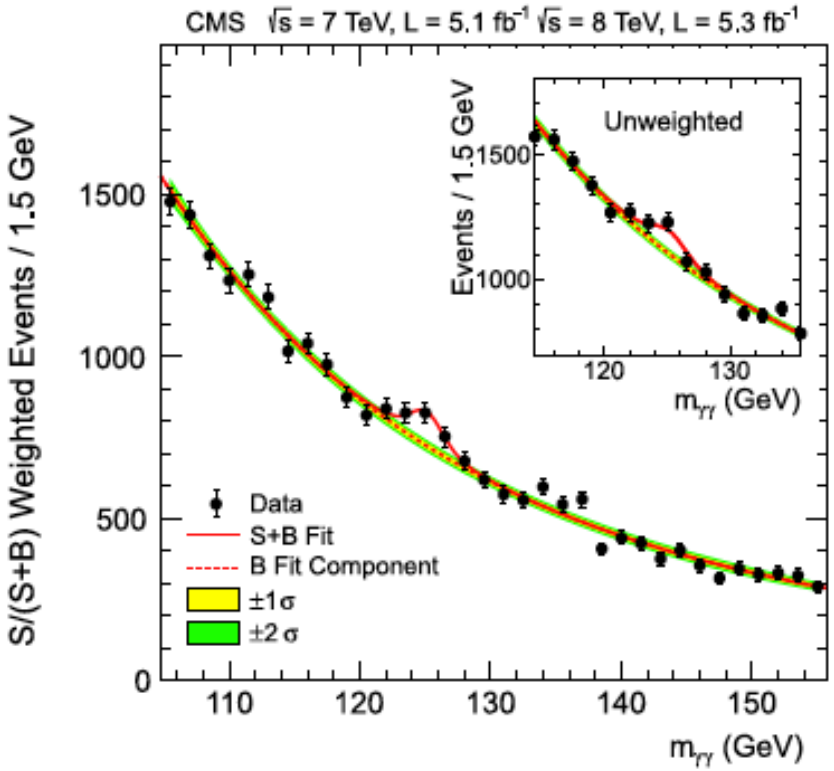


Combining signal and background

- Finally add signal and background distributions

Result from this toy simulation





Next step is to apply a statistical analysis of this distribution, including fitting for the signal.

Exercise

- Run the code and generate your own reconstructed mass distribution as in the previous slide.
- Re-run plotting code using a factor of two larger scaling for signal.
- Publish both graphs in the Lesson7/Results/

APPENDIX:
Event storage and processing with TTree Class

https://root.cern.ch/root/htmldoc/guides/users-guide/Trees.html

How to create a TTree

- Create variables
```
float x;
float y;
```
- Create TTree object with branches:
```
TTree t("tree","test tree");
t.Branch("x",&x,"x/F");//name, link, leaf/type
t.Branch("y",&y,"y/F");
```
- Generate events and fill variables and tree:
```
for(int i=0;i<100;i++){
 x=i%100;
 y=2.*x;
 t.Fill();
}
```
- Save tree to file:
```
TFile  f("tree.root","recreate");
t.Write();
f.Close();
```

How to read a TTree: `Draw()` method

- Access tree from file:
```
TFile  f("tree.root","read");
TTree * t = (TTree*)f.Get("tree");
```
- Explore tree structure:
```
t->Print()
```
- Create graphs using `Draw()`:
```
TCanvas C;
t->Draw("x");//automatic binning
t->Draw("x>>h(100,0,100)");//user defined binning
C.Print("graph_x.png");
```
- Explore correlations:
```
t->Draw("x:y");
```
- Apply selections:
```
t->Drwa("x","y<50");
```

How to read a TTree: Loop method

- Access tree from file:

```
TFile  f("tree.root","read");
TTree * t = (TTree*)f.Get("tree");
```

- Create and link variables :

```
float x;
float y;
t->SetBranchAddress("x",&x);
t->SetBranchAddress("y",&y);
```

- Read events and fill histograms :

```
TH1F Hx("Hx","x",100,0,100);
TH2F Hxy("Hxy","x-y",100,0,100,100,0,200);
for(int i=0;i<t->GetEntries();i++){
 t->GetEntry(i);
 cout<<x<<" "<<y<<endl;
 Hx.Fill(x);
 Hxy.Fill(x,y);
}
```