Higgs Physics In Future Colliders

Eslam Muhammed Ahmed Zenhom
201700788

Supervised by Dr. Ahmed Ali Abdelalim

Submitted in July 2021, in partial fulfillment of
the conditions for the award of a **B.S. degree in Physics of the Earth and the Universe major (High Energy Physics Concentration).**

Physics Department
University of Science and Technology at Zewail City

Abstract

In this thesis, we will simulate the pp collisions in the LHC at the center of mass-energy $s = \sqrt{14}$ TeV, and the FCC at center of mass-energy $s = \sqrt{100}$ TeV. Then we will compare the results of the two. The goal of the collision is to produce the Higgs Boson which will be detected from its decay in the two photons channel.

$$pp \rightarrow h \rightarrow \gamma\gamma$$

The aim of the simulation is to provide insight on how the Future Circular Collider will provide more accurate precision measurements of the Higgs Boson which will provide a gateway into physics beyond the standard model and new discoveries. The simulation will be done using luminosity of 300 fb^{-1} for the LHC, while it will be 3000 fb^{-1} for the FCC.

Contents

List of Figures

Chapter 1

Introduction to the Higgs Boson

In this chapter, I will introduce the standard model of particles, the Higgs Boson, and the mechanisms for the production of the Higgs Boson

1.1 The Standard Model of Particle Physics

The Standard Model of Particle Physics explains the building blocks of our universe and how they interact together to form the known matter. It also explains how matter interacts through the force carriers. It explains three of the four fundamental forces: electroweak force, electromagnetic force, and the strong nuclear force[3], [16], [22]. See the content of the standard model in Figures 1.1 , 1.2

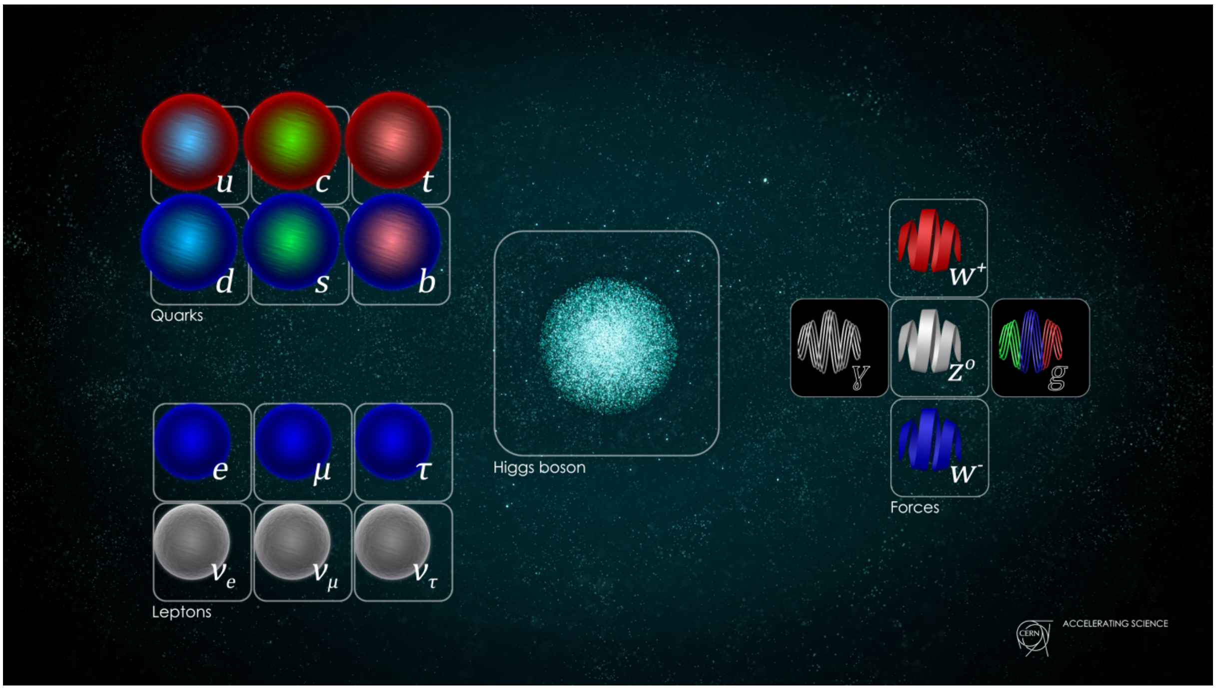

Figure 1.1: The standard model content

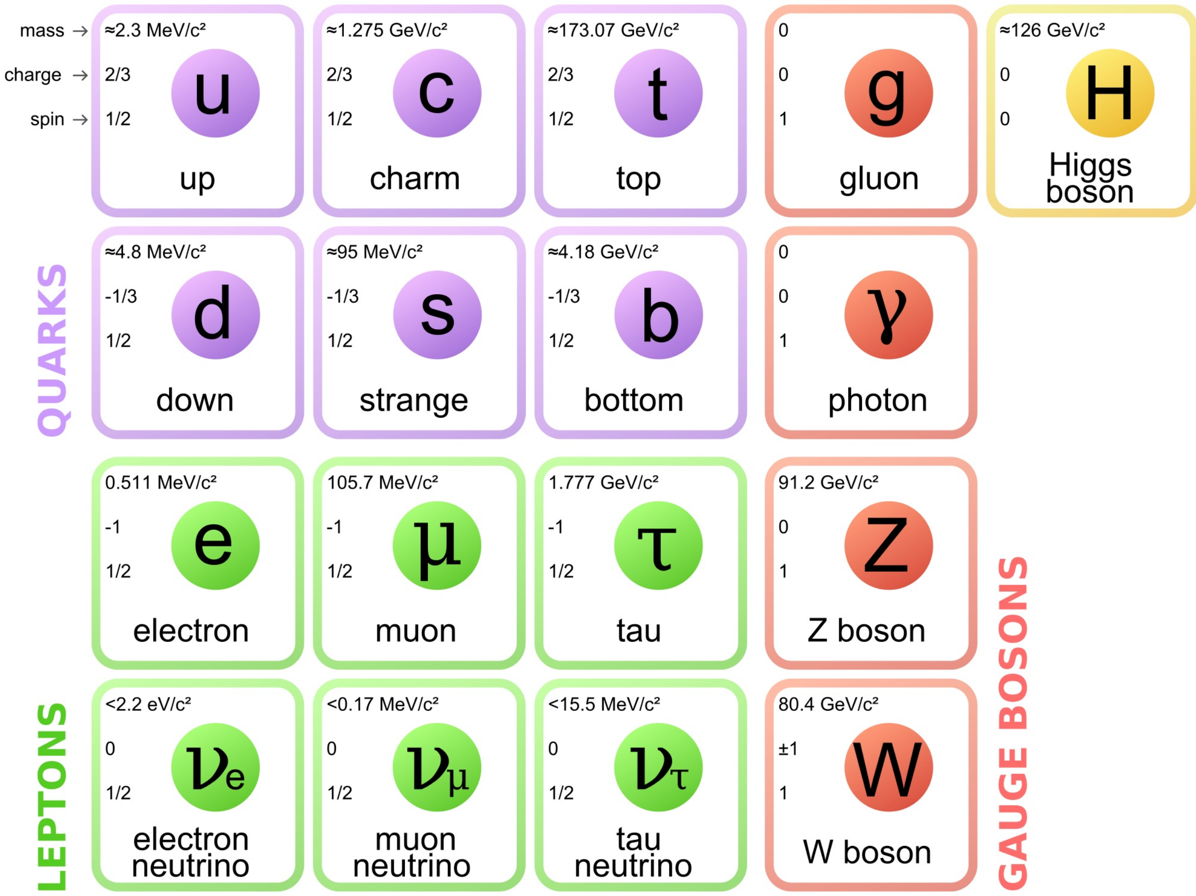

Figure 1.2: The standard model

1.2 The Standard Model Content

1.2.1 Fermions

Fermions are the spin 1/2 particles. The standard model contains 12 different elementary fermions:

- Quarks: they carry a color charge. Therefore, they interact via the strong force. Additionally, they have electric charges and also interact through electromagnetic force. The quarks bound together via the strong force, forming color neutral composites called hadrons. The hadrons are either mesons (quark plus anti-quark) or baryons (formed from three quarks).

  The quarks consist of three generations. Each following generation has a higher mass than the previous one. The lightest generation is the one with (up and down quarks). Due to their low masses, they don't decay, hence all baryonic matter is formed from the first generation quarks.

  The second and third generations decay with very short times.

- Leptons: They are the remaining 6 fermions. They don't have a color charge. Thus, they don't interact via the strong force. They can interact via electromagnetic and electroweak forces. They consist also of three generations: the first one (electron and electron neutrino), second generation (muon and muon neutrino), and the third generation (tau and tau neutrino).

1.2.2 Bosons

They are defined to have integer spin, and they are the carriers of the three fundamental forces: strong, weak, and electromagnetic forces. The standard model explains how particles interact with each other by exchanging the force carriers. The bosons are:

1. Photons: they have spin 1 and they mediate the electromagnetic interactions. Photons are massless

2. W and Z bosons which mediate the electroweak interactions. These bosons are massive.

3. We have eight massless gluons which mediate the strong force interactions.

4. Finally, we have the zero spin Higgs boson. It is a massive boson that was theorized by Peter Higgs in 1964 to give masses to the W and Z bosons. It was first discovered by CERN at the LHC in 2012 with a mass of 125 GeV.

1.3 The Higgs Boson

We know that the weak interaction is mediated by the W and Z bosons. From the content of the standard modern, we know that they have masses of $80.4\mathrm{GeV/c^2}$ and $91.2\mathrm{GeV/c^2}$ respectively. We need our Lagrangian density to stay gauge invariant, thus we can't have the following term : $\frac{1}{8\pi}\left(\frac{mc}{\hbar}\right)^2 A^\mu A_\mu$.
As a result, we can't add the mass term by hand. Therefore, we had to have a mechanism in which gives the W and Z masses and also not spoil the gauge invariant. The theoretical physicists Robert Brout, François Englert and Peter Higgs made a proposal to give the particles mass while staying gauge invariant. The proposal was the Brout-Englert-Higgs mechanism which gives a mass to the W and Z when they interact with an invisible field called the Higgs field [8].

1.4 Production mechanisms of the Higgs boson

We are interested in the production mechanisms as in this thesis we will be interested in calculating the Higgs boson mass. The Higgs production has many channels, but we are interested in the channels related to pp hadron colliders as we will work with the large hadron collider (LHC) and the future circular collider (FCC-hh). There are four main channels ordered from the most dominant to the least dominant [9]:

- **Gluon Fusion channel:** $gg \to H$

  It is the dominant channel in hadron colliders. The protons and anti-protons collide at the hadron colliders. These particles are composed of quarks and gluons. If the gluons collide together, they will form a virtual loop of quarks as shown in Fig 1.3. The coupling to the Higgs is proportional to the mass of the particles. Thus, those virtual quarks are more likely to be the heaviest ones: the top and bottom quarks.

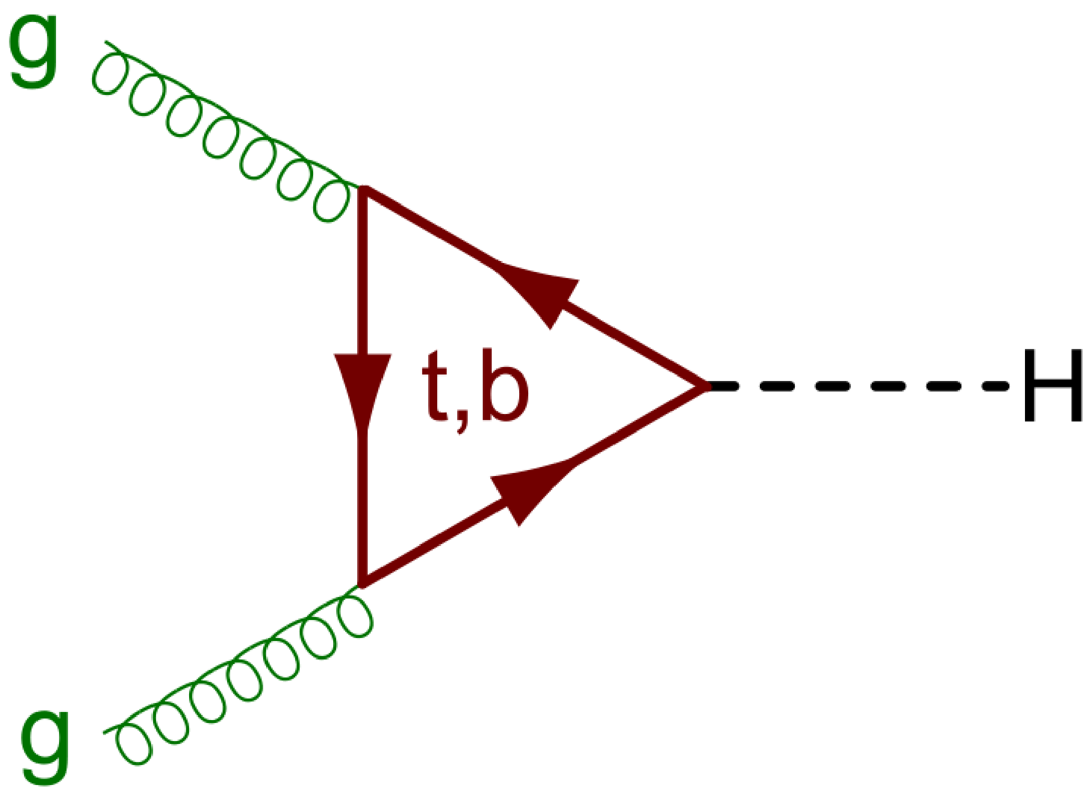

Figure 1.3: The production channel of the gluon fusion

- **Vector Boson Fusion or Weak boson fusion channel**

  It happens when an elementary fermion collides with an anti-fermion like a quark and an antiquark. Then, the two will exchange either virtual W or Z bosons, which will result in the production of Higgs Boson. Note: the fermions might be in different types. The Feynman diagram is depicted in the figure below

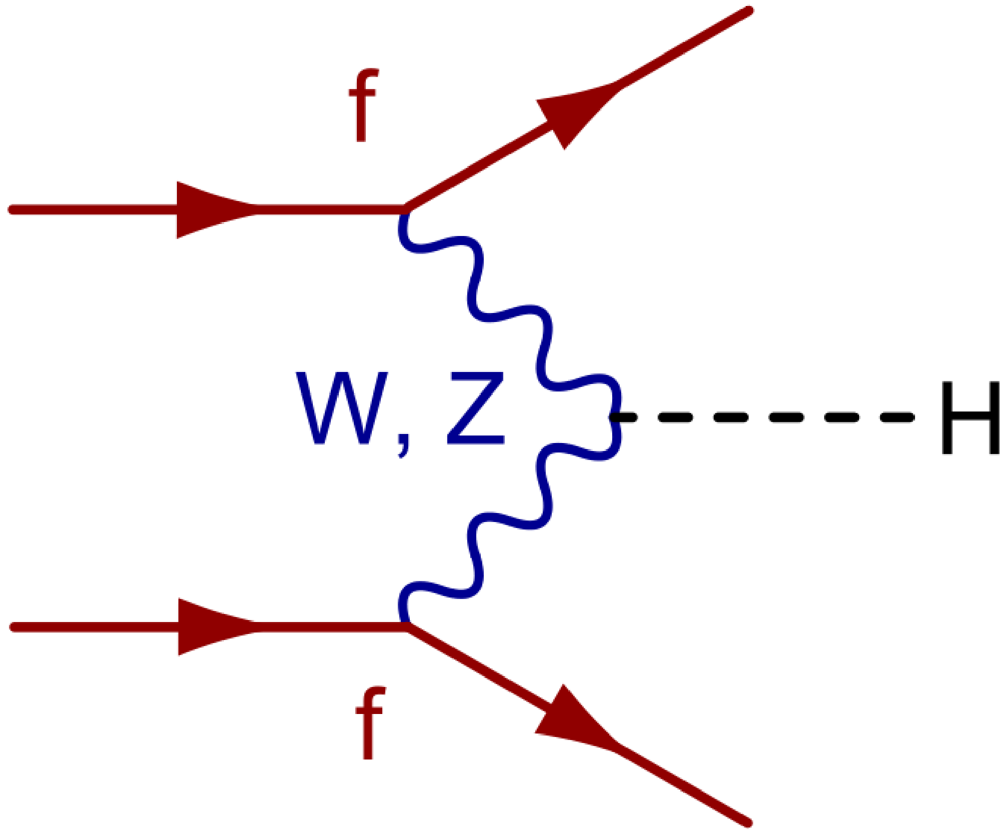

Figure 1.4: The production channel of the vector boson fusion

- **Higgs Strahlung channel**

  It is known as associated production. It happens when an elementary fermion collides with an anti-fermion like a quark with an anti-quark. Those collisions will create virtual W or Z. When they have enough momentum to provide them with enough kinetic energy, they will create a Higgs boson. See the figure below.

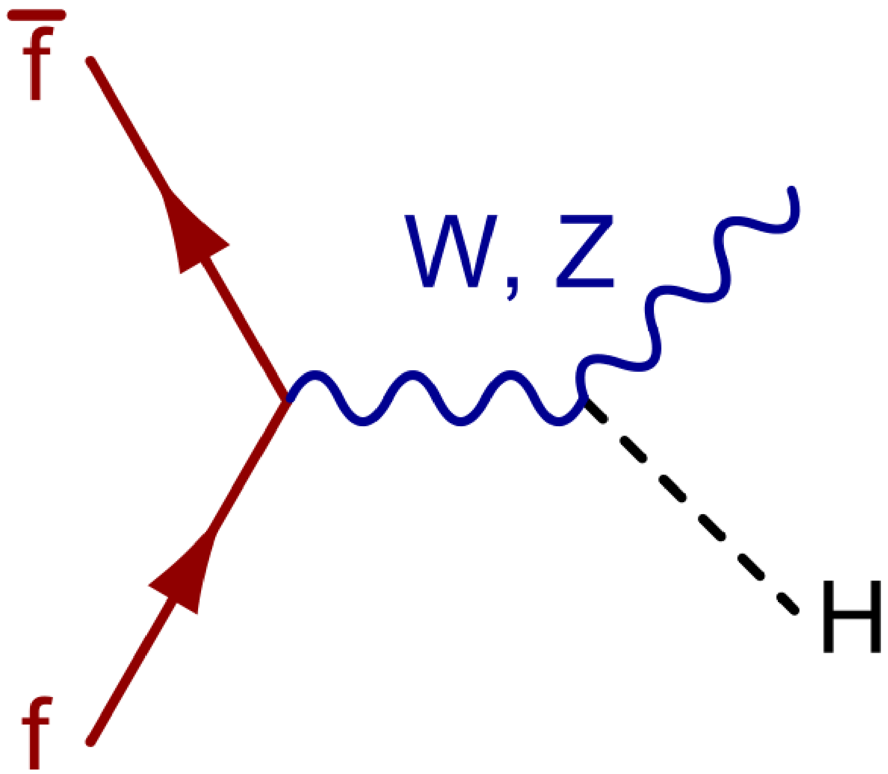

Figure 1.5: The production channel of Higgs Strahlungn

- **Top Fusion channel**

  It happens when two gluons collide and each one decays into a heavy quark and anti-quark pair. Those pairs will interact and form the Higgs as in the figure below.

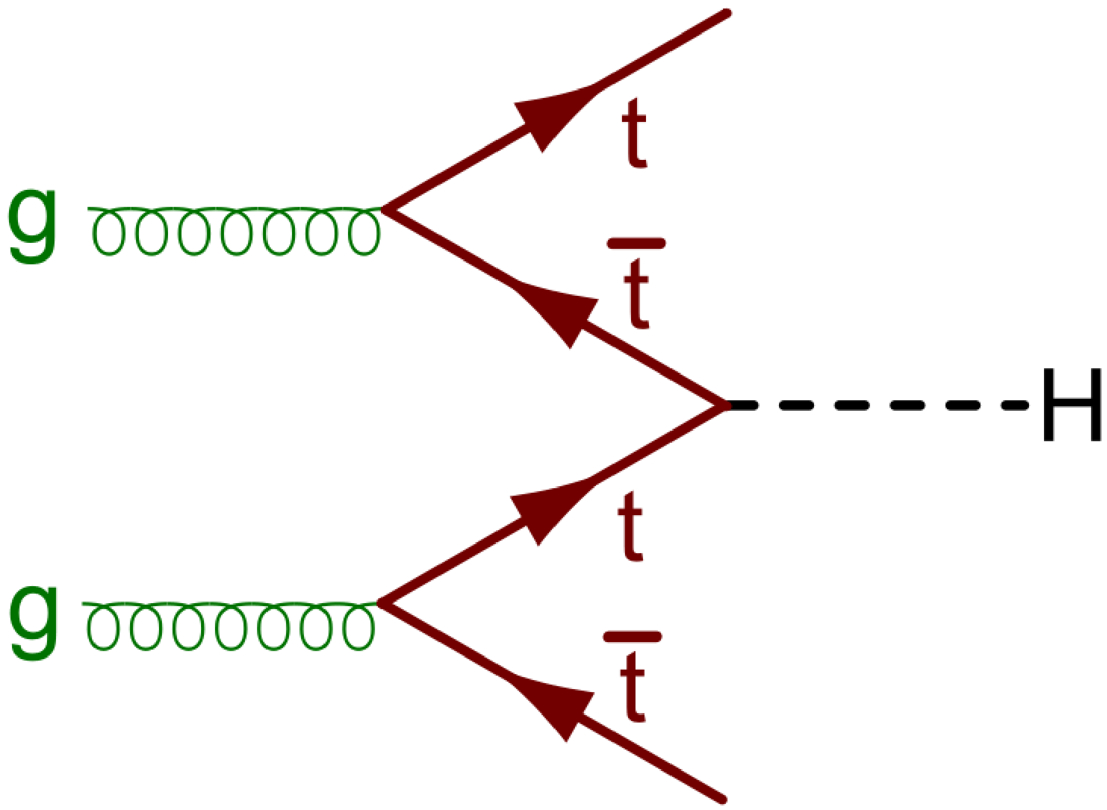

Figure 1.6: The production channel of Top Fusion

1.5 Decay Channels for the Higgs

The Higgs bosons decay into a pair of fermions or bosons [18]. The Higgs boson is estimated to have a mass of 125 GeV which is the second most highest particle mass ever after the top quark. Thus, it decays fast, with a decay time of 1.6×10^{-22}s. Therefore, there is no experimental technology to detect the Higgs before its decay. Thus, we have to study the decay channels in order to have an insight into whether a Higgs boson is produced or not. In this thesis, we will focus on the Higgs decay into two photon (see Figure 1.7).

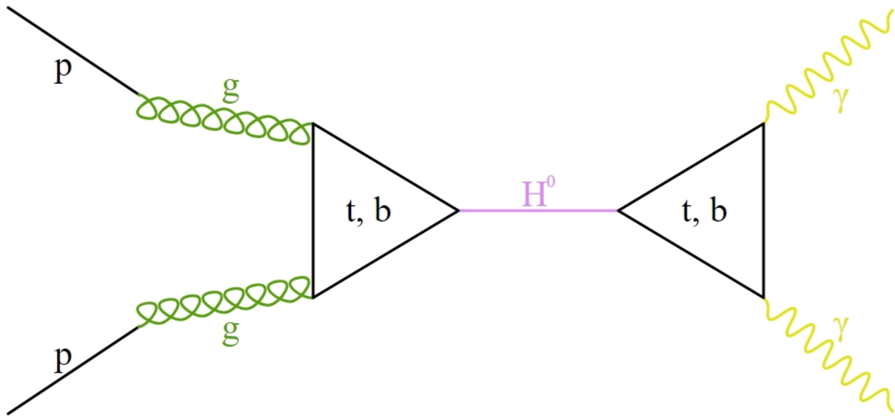

Figure 1.7: The decay of the Higgs into two photons

Figure 1.8 below will illustrate the fermions and bosons to which the Higgs decays. It illustrates the branching ratio, a measure of the ratio of the number of particles that decays into that channel compared to the total number of particles, decaying in all the channels.

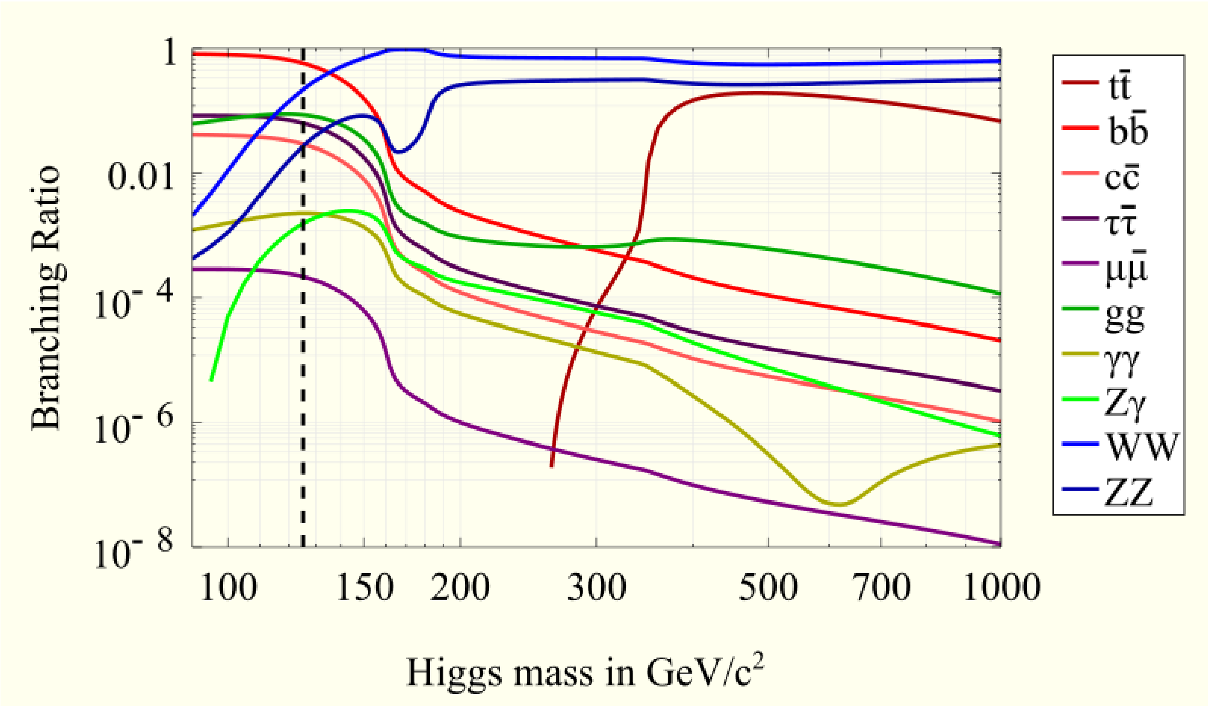

Figure 1.8: BR (branching ratios) for the Higgs Boson

Chapter 2

The Experiment

2.1 Some definitions related to the experiment

2.1.1 Pseudorapidity

It describes the angle of the produced particles relative to the beam. See figure 2.1 and 2.2

$$\eta \equiv -\ln\left[\tan\left(\frac{\theta}{2}\right)\right]$$

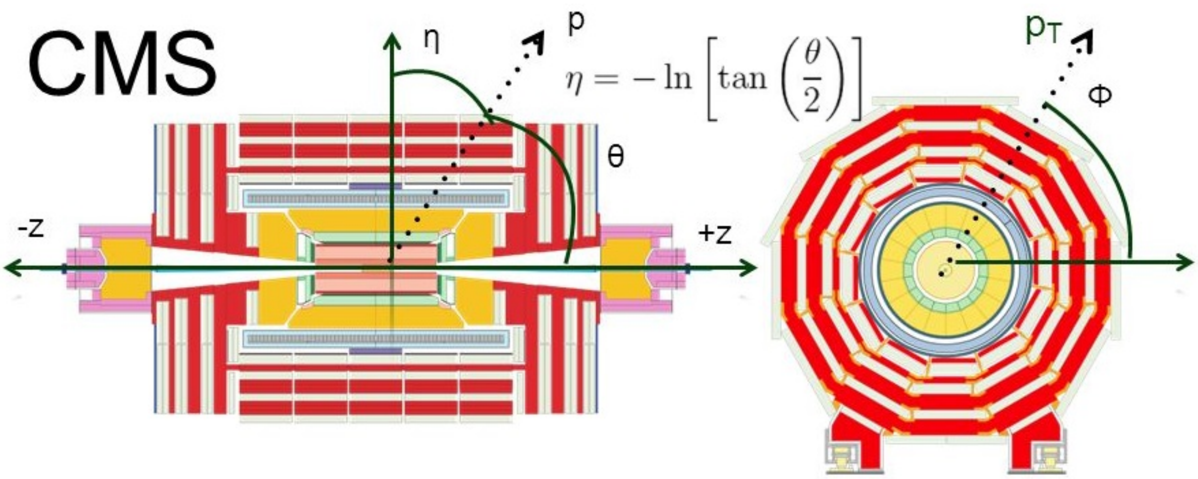

Figure 2.1: Pseudorapidity in CMS detector

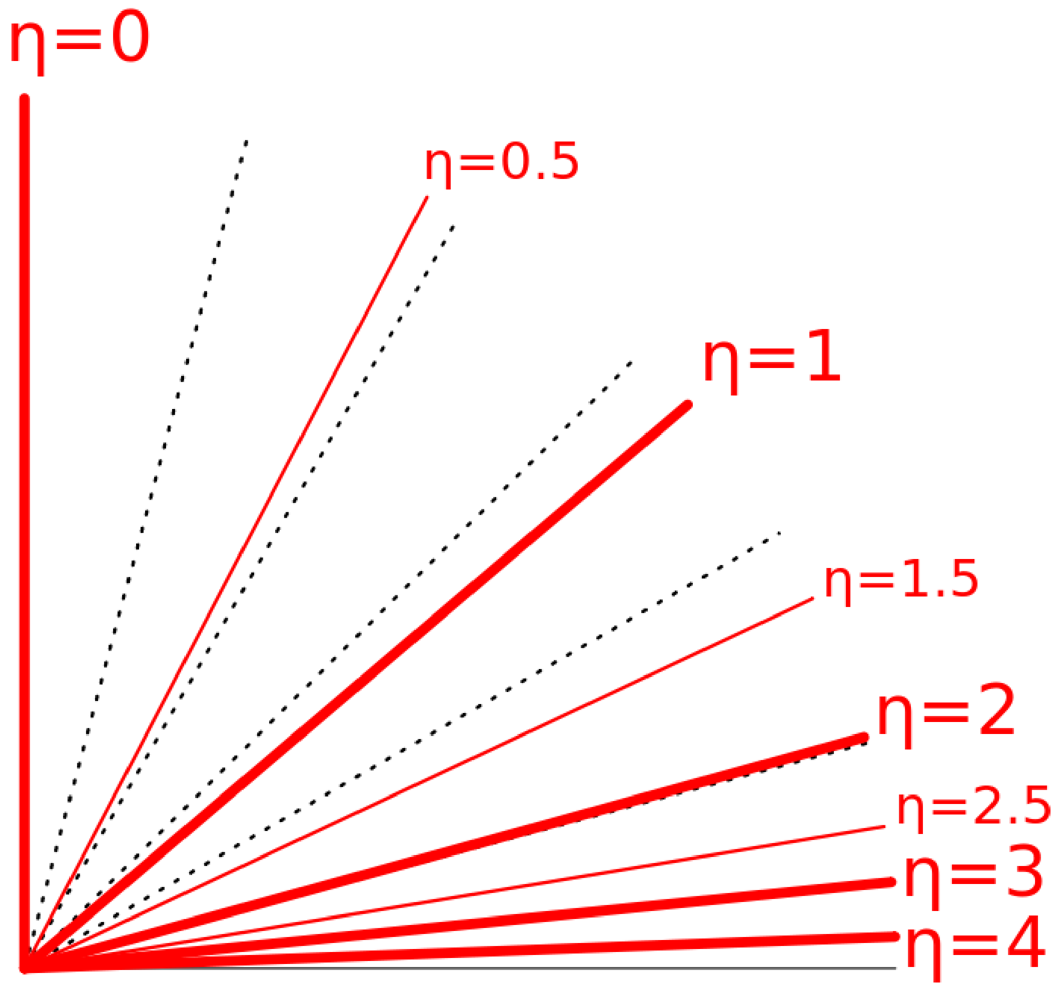

Figure 2.2: Pseudorapidity

2.1.2 Coordinate System

In our thesis and the ROOT analysis, we use the coordinate system which is used in the CMS and ATLAS detectors.

The ϕ angle is the normal azimuthal angle. Here, it is the angle that is made with the x-axis.

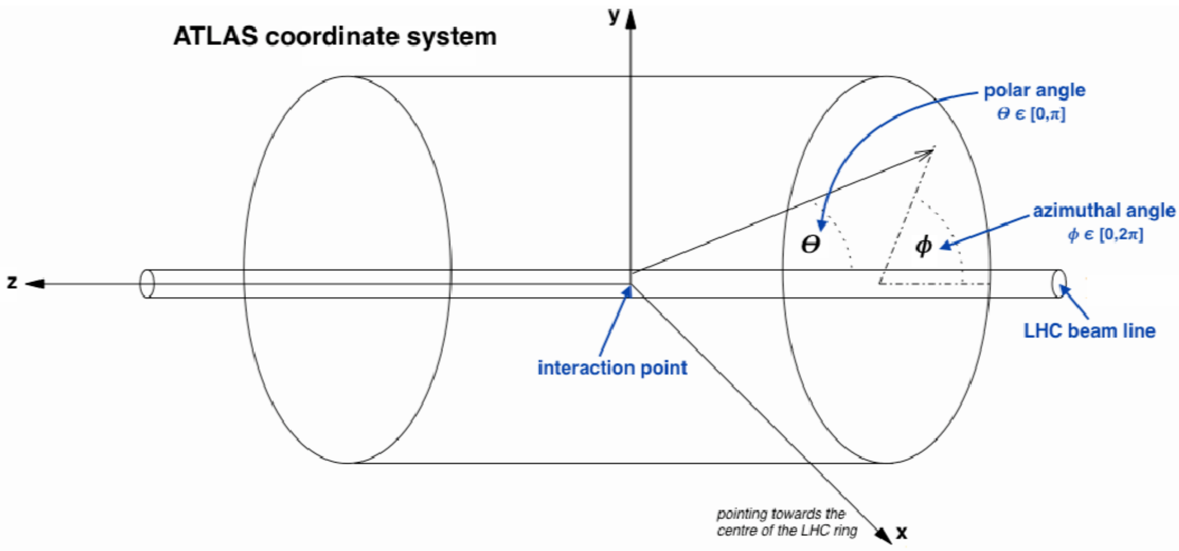

Figure 2.3: Coordinates used in ATLAS [17]

In our system we used a four-vector called Tlorentz vector:

$$p^{\mu} = \begin{pmatrix} p_T \\ \eta \\ \phi \\ M \end{pmatrix}$$

Where ϕ is what we mentioned in the first of the page and η is the Pseudorapidity mentioned in the previous sub-section. p_T is the transverse momentum, which is the momentum perpendicular to the direction of the beam. Finally, M is the invariant mass estimated by:

$$M = \sqrt{E^2 - p^2} \quad (c = 1)$$

2.2 The Large Hadron Collider (LHC)

Figure 2.4: The LHC [11]

The LHC is the largest particle collider that has ever been built in the world. The European Organization for Nuclear Research (CERN) has built it between 1998 and 2008. More than 10,000 scientists and engineers work in collaboration with the LHC. It is 175 meters beneath the ground with a 27 km circumference. The Higgs boson was discovered in 2012 at the LHC by the ATLAS and CMS detectors [11], [10]. In our thesis, we will simulate the conditions of the LHC in order to produce the Higgs boson and measure its mass.

2.2.1 The ATLAS Detector

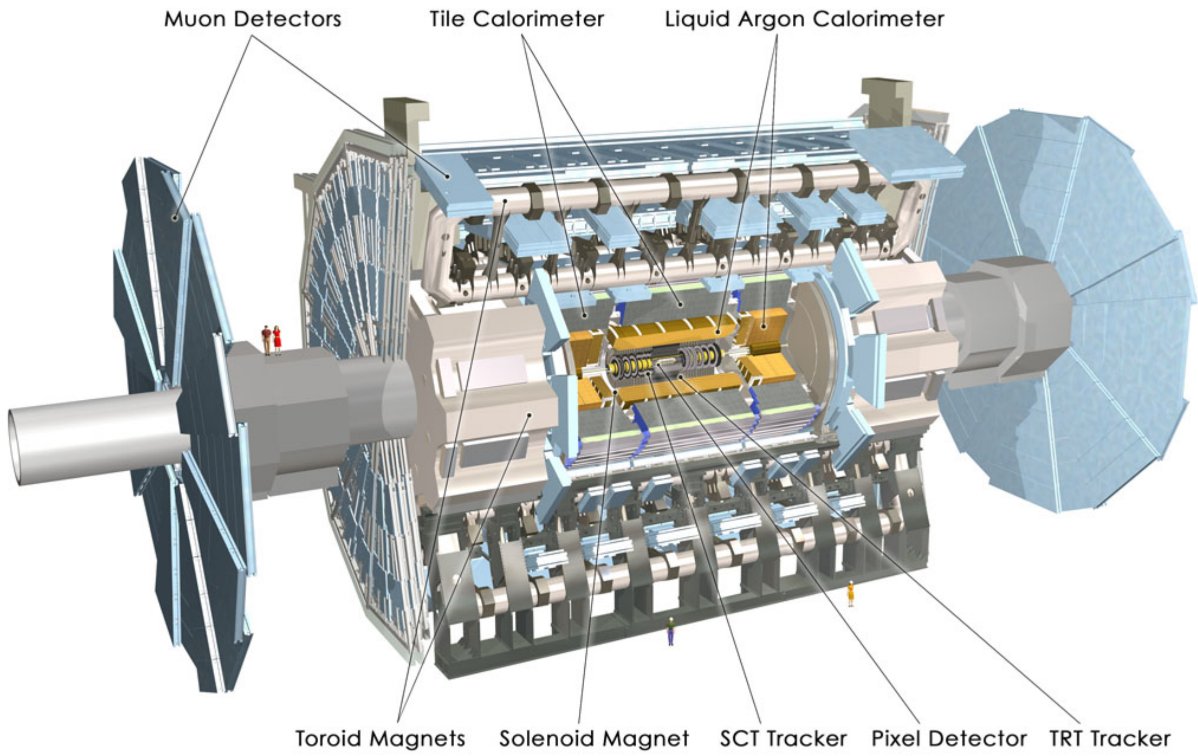

Figure 2.5: The ATLAS Detector [11]

ATLAS was one of the two LHC experiments involved in the discovery of the Higgs boson in 2012. The ATLAS experiment (A Toroidal LHC Apparatus) is a massive 7-ton general purpose particle detector investigating the largest range of physics possible [1]. ATLAS consists of four main components:

- Inner Detector for tracking particles movements
- Calorimeter in order to measure the particles' energies
- The muon detector detects the muons and neutrinos which had high energies enabling them to travel long distances until they reach the far muon detector.
- The trigger system of ATLAS gives the signals and coordinates the data collection of all the sub components.

2.2.2 The CMS Detector

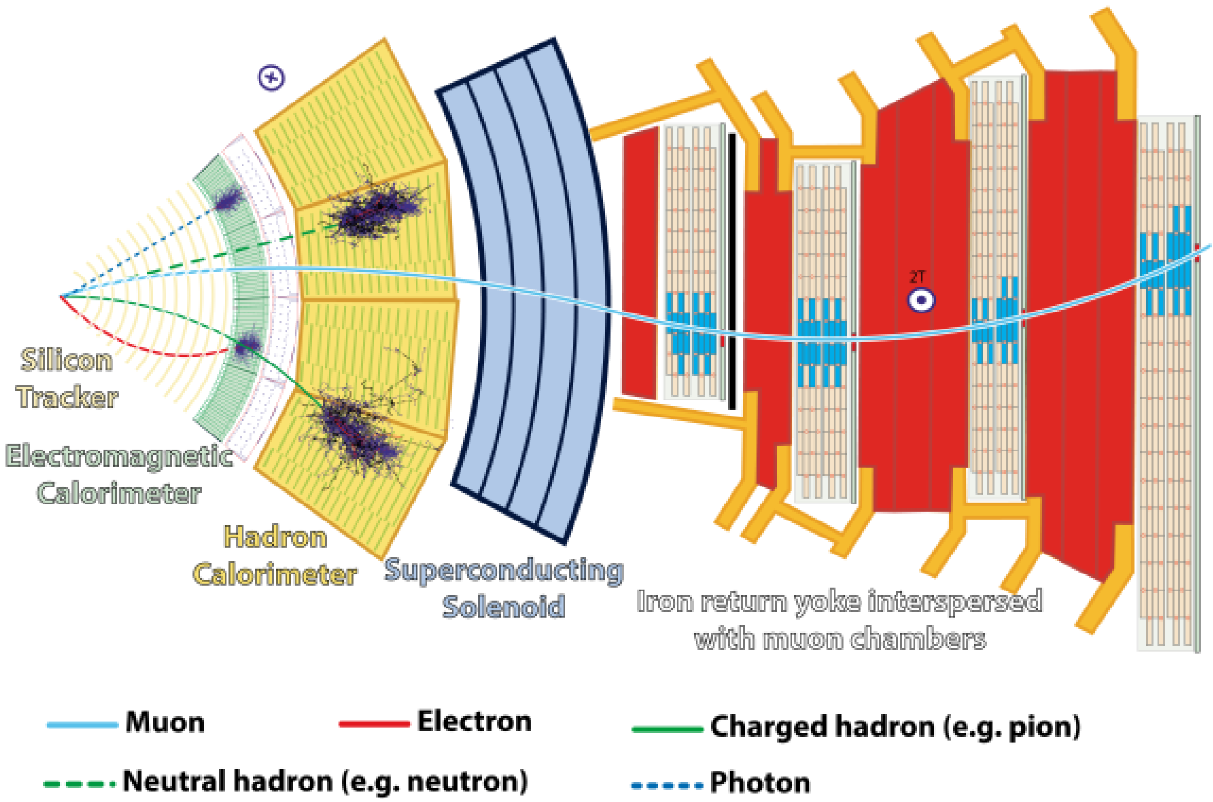

Figure 2.6: The CMS Detector [2]

CMS is similar to ATLAS as being one of the two large experiments of the LHC. It was used also in 2012 to detect the Higgs. The goal of CMS is to study a wide range of physical phenomena like the search for the Higgs boson, extra dimensions, and particles that could make up dark matter [2], [20].

The CMS detector consists of:

- The Silcon tracker is used to calculate the trajectory of particles in a magnetic field. Thus, the momenta can be calculated from the trajectories.
- The electromagnetic calorimeter measures the energies of the photons and electrons via electromagnetic interactions.
- The hadronic calorimeter measures the energies of hadrons. Hadrons are particles consisting of quarks and gluons.

- The muon detector is placed at the end of the CMS detector because muons penetrate all the previous layers.

2.3 The Future Circular Collider (FCC)

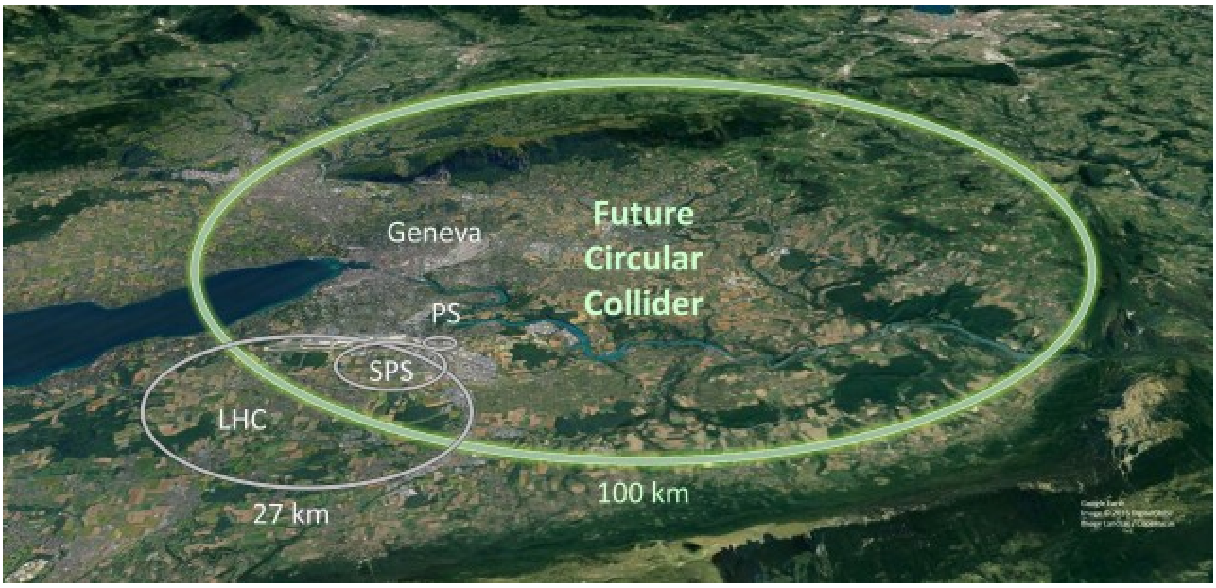

Figure 2.7: The FCC [7]

The FCC is a proposed collider for the future by CERN. It will operate on beams with center of mass energy of 100 TeV. The collider will have circumference of 100 km. It is aimed to increase the energy and intensity of the beams in order to be able to have precision measurements for the Higgs and for future searches for physics beyond the standard model [4]. It has been launched as result of the recommendation made in the 2013 update of the European Strategy for Particle Physics that: "Europe needs to be in a position to propose an ambitious post-LHC accelerator project at CERN by the time of the next Strategy update, when physics results from the LHC running at 14 TeV will be available." [4]

The FCC has 3 type of colliders:

- electron/positron collider (FCC-ee): it is a lepton collider with energies ranging from (90 to 350) GeV.
- High-Energy LHC: it will be in the same tunnel of the LHC, but it will have stronger dipole magnets to increase the center of mass energy by a factor of two (27 TeV). Also, it will provide 3 times the luminosity provided by the high luminosity LHC [5].

- FCC-hh: it is a hadron collider (proton proton collider) with center of mass energy of 100 TeV [6]. In our thesis, we will work with the FCC-hh to get results for the Higgs mass. Note: in the rest of the thesis, I will use FCC to indicate the FCC-hh.

Chapter 3

Methodology for Data Generation, Simulation, and Analysis

The goal of the thesis is to simulate the pp collisions in the CMS experiment at the LHC at the center of mass-energy $s = \sqrt{14}$ TeV, and the FCC at center of mass-energy $s = \sqrt{100}$ TeV then we compare the results of the two. The goal of the collision is to produce the Higgs Boson which will be detected from its decay in the two photons channel.

$$pp \to h \to \gamma\gamma$$

The aim of the simulation is to provide insight on how the Future Circular Collider will provide more accurate precision measurements of the Higgs Boson which will provide a gateway into physics beyond the standard model and new discoveries. The simulation will be done using luminosity of 300 fb^{-1} for the LHC, while it will be 3000 fb^{-1} for the FCC. Other trials with different luminosities will be also included in the results section for more clarification and understanding of the Higgs production and decay.

3.1 The Technical Tools:

3.1.1 MadGraph

MadGraph simulates the pp collision and generates the events: the resulting quarks and gluons. MadGraph5-aMC@NLO (MG5) is a framework that provides all the elements of the standard model and the Higgs effective field theory model (HEFT model). The framework provides a strong tool for the generation of hard events, and computations of the cross-section. The MadGraph has integrated tools which can be installed and run through MadGraph. These tools are Delphes and Pythia8 [14][13][12]. MadGraph5-aMC@NLO is the new version of both MadGraph5 and aMC@NLO that unifies the LO and NLO lines of development of automated tools within the MadGraph family. The version used in the thesis is 2.7.3.

3.1.2 Pythia8

Pythia is a program that deals with the final states of the generated events from MadGraph. Pythia does the job of hadronization, Confining the resulting quarks and gluons to their hadrons. Pythia is the event generator for the showering by simulating the final and initial states of the radiation [23]. Pythia is used as MG5 don't do the showering and hadronization process. Pythia8 is integrated into the MadGraph program and can be turned on to do the showering and the hadronization. The process of how to turn it on will be illustrated in the next section. In short, MG5 just does the general interactions based on Fermi golden rules and the matrix elements calculation while Pythia deals with the more detailed showers of radiation and particles and takes into account the QCD effects.

3.1.3 Delphes

Delphes is a program used to simulate a multipurpose detector for phenomenological studies. The simulation includes a track propagation system embedded in a magnetic field, electromagnetic and hadron calorimeters, and a muon identification system [21]. We will use Delphes in our thesis to work as the detector for the LHC. Also, Delphes has another card for the FCC which will be used to detect the events in the 100 TeV case. Delphes takes the input of the generated data and simulates it as a real detector and then provides us with a ROOT file that can be analyzed and draw the resulting histograms. The used version is Delphes 3.5.0 which installed by the MadGraph itself.

3.1.4 ROOT CERN

ROOT is a framework for data analysis in particle physics created by CERN. It is built and written in C++. It can create histograms and many other plots which can be manipulated easily to be used in scientific papers [19][15]. In our thesis, Delphes will create an output of ROOT file. We will then edit it and write the C++ code for analyzing the data and creating the histograms. The code can be found in the appendix for the case of 14TeV and 100 Tev. The used version is 6.18 as indicated in figure3.1 .

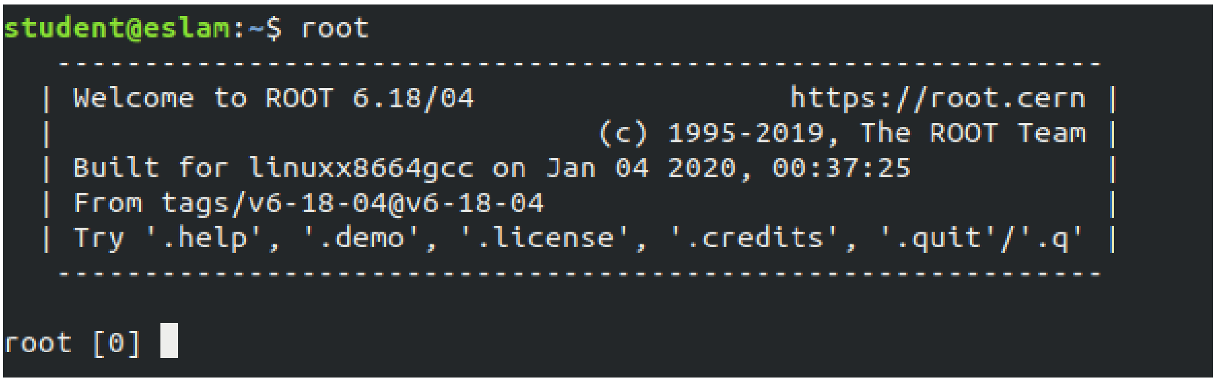

Figure 3.1: ROOT Open Window

3.2 Steps for Events Generation and Simulation

3.2.1 Running MadGraph

First, we will open MadGraph by going to its directory and run the next code:

```
student@eslam:~/mad/MG5_aMC_v2_7_3$ ./bin/mg5_aMC
```

We then will have the welcome window opened:

```
************************************************************
*                                                          *
*                     W E L C O M E  to                    *
*              M A D G R A P H 5 _ a M C @ N L O           *
*                                                          *
*                                                          *
*                 *                       *                *
*                   *        * *        *                  *
*                     * * * * 5 * * * *                    *
*                   *        * *        *                  *
*                 *                       *                *
*                                                          *
*         VERSION 2.7.3                 2020-06-21         *
*                                                          *
*    The MadGraph5_aMC@NLO Development Team - Find us at   *
*    https://server06.fynu.ucl.ac.be/projects/madgraph     *
*                            and                           *
*            http://amcatnlo.web.cern.ch/amcatnlo/         *
*                                                          *
*               Type 'help' for in-line help.              *
*           Type 'tutorial' to learn how MG5 works         *
*    Type 'tutorial aMCatNLO' to learn how aMC@NLO works   *
*    Type 'tutorial MadLoop' to learn how MadLoop works    *
*                                                          *
************************************************************
load MG5 configuration from input/mg5_configuration.txt
```

Now we will import the heft-full model it is an extension for the standard model with the Higgs effective field theory extension. We used this model as our thesis will deal with the Higgs Boson production and decay.

```
MG5_aMC>import model heft-full
```

We will generate the interaction with the MadGraph using the following:

```
MG5_aMC>generate p p > h > a a
```

We can display the interaction as Feynman diagram using MadGraph using:

```
MG5_aMC>display diagrams
```

The Diagram:

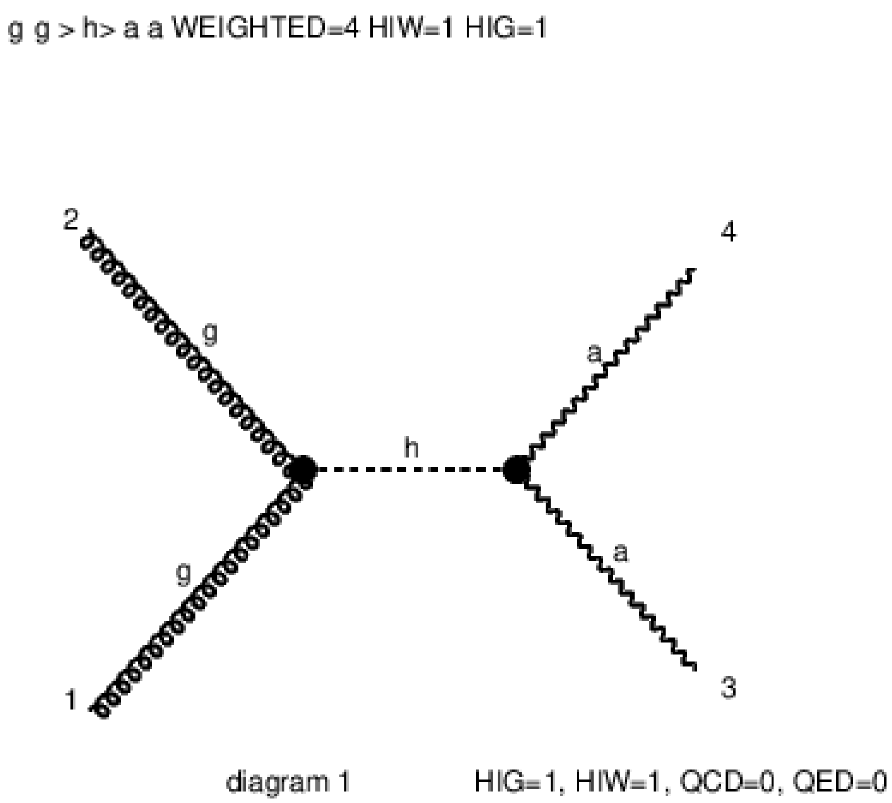

Figure 3.2: The Feynman Diagram for the interaction

Now we have generated the interaction through MG5. We will need to save the output using the following:

```
MG5_aMC>output /home/student/output/ppT0h_hT0aa_14Tev
```

Figure 3.3: Output Command For saving MadGraph file for 14TeV

We will launch the file using:

```
MG5_aMC>launch /home/student/output/ppT0h_hT0aa_14Tev
```

Figure 3.4: Launching command for MadGraph file for 14TeV

We will do the same for the 100 TeV case by just changing each 14 TeV to 100 TeV.

3.2.2 Setting the Parameters for the 14 TeV

Now we will turn on the Pythia and Delphes for the 14 TeV run by:

```
The following switches determine which programs are run:
/==========================================================================\
| 1. Choose the shower/hadronization program     shower = Pythia8          |
| 2. Choose the detector simulation program    detector = Delphes          |
| 3. Choose an analysis package (plot/convert) analysis = ExRoot           |
| 4. Decay onshell particles                    madspin = OFF              |
| 5. Add weights to events for new hypp.       reweight = OFF              |
\==========================================================================/
Either type the switch number (1 to 5) to change its setting,
Set any switch explicitly (e.g. type 'shower=OFF' at the prompt)
```

Figure 3.5: Opening Pythia and Delphes for 14TeV

We will press 0 once we are done with opening Pythia and Delphes. Then, we will have the window for the cards:

```
Do you want to edit a card (press enter to bypass editing)?
/------------------------------------------------------------\
|  1. param   : param_card.dat                               |
|  2. run     : run_card.dat                                 |
|  3. pythia8 : pythia8_card.dat                             |
|  4. delphes : delphes_card.dat                             |
\------------------------------------------------------------/
```

Figure 3.6: Cards used for 14TeV Run

We will press 2 to enter the vim window to change the parameters for the run. We will choose the number of events to be 100000 and the center of mass energy to be 14TeV by choosing each beam energy to be 7000 GeV. See figure 3.7

```
#*********************************************************************
  tag_1     = run_tag ! name of the run
#*********************************************************************
# Number of events and rnd seed                                      *
# Warning: Do not generate more than 1M events in a single run       *
#*********************************************************************
  100000 = nevents ! Number of unweighted events requested
  0   = iseed   ! rnd seed (0=assigned automatically=default))
#*********************************************************************
# Collider type and energy                                           *
# lpp: 0=No PDF, 1=proton, -1=antiproton, 2=photon from proton,      *
#                                         3=photon from electron     *
#*********************************************************************
     1        = lpp1    ! beam 1 type
     1        = lpp2    ! beam 2 type
     7000.0     = ebeam1  ! beam 1 total energy in GeV
     7000.0     = ebeam2  ! beam 2 total energy in GeV
# To see polarised beam options: type "update beam_pol"

#*********************************************************************
```

Figure 3.7: Setting parameters for 14TeV Run

3.2.3 Setting the Parameters for the 100 TeV

We will do the same as before. However, for the Delphes program, we will make it off (See figure 3.8). We did this as if we leave it open it will run using the card for the LHC detector which will provide non-accurate results. We will use Delphes after the run is finished using another command to run using the FCC card which simulates the detector of the FCC. Note: the detector for the FCC is better than LHC as it can detect more events.

```
The following switches determine which programs are run:
/============================================================================\
| 1. Choose the shower/hadronization program     shower = Pythia8            |
| 2. Choose the detector simulation program    detector = OFF                |
| 3. Choose an analysis package (plot/convert) analysis = ExRoot             |
| 4. Decay onshell particles                    madspin = OFF                |
| 5. Add weights to events for new hypp.       reweight = OFF                |
\============================================================================/
```

Figure 3.8: Pythia is open while Delphes is Off for 100 TeV Run

Now we will enter the run card by pressing 2 to set the parameters:

```
Do you want to edit a card (press enter to bypass editing)?
/------------------------------------------------------------\
|  1. param   : param_card.dat                                |
|  2. run     : run_card.dat                                  |
|  3. pythia8 : pythia8_card.dat                              |
\------------------------------------------------------------/
```

Figure 3.9: Cards used for 100TeV Run

We will set the parameters to make it 100 thousand events like the previous case. The energy for each beam will be changed to 50000 GeV to make a total center of mass energy of 100 TeV. See figure 3.10

```
#*********************************************************************
#                      MadGraph5_aMC@NLO                             *
#                                                                    *
#                    run_card.dat MadEvent                           *
#                                                                    *
#  This file is used to set the parameters of the run.               *
#                                                                    *
#  Some notation/conventions:                                        *
#                                                                    *
#   Lines starting with a '# ' are info or comments                  *
#                                                                    *
#   mind the format:   value    = variable     ! comment             *
#                                                                    *
#   To display more options, you can type the command:               *
#      update full_run_card                                          *
#*********************************************************************
#
#*********************************************************************
# Tag name for the run (one word)                                    *
#*********************************************************************
  tag_1     = run_tag ! name of the run
#*********************************************************************
# Number of events and rnd seed                                      *
# Warning: Do not generate more than 1M events in a single run       *
#*********************************************************************
  100000 = nevents ! Number of unweighted events requested
  0   = iseed   ! rnd seed (0=assigned automatically=default))
#*********************************************************************
# Collider type and energy                                           *
# lpp: 0=No PDF, 1=proton, -1=antiproton, 2=photon from proton,      *
#                                         3=photon from electron     *
#*********************************************************************
     1        = lpp1    ! beam 1 type
     1        = lpp2    ! beam 2 type
     50000.0     = ebeam1  ! beam 1 total energy in GeV
     50000.0     = ebeam2  ! beam 2 total energy in GeV
```

Figure 3.10: Setting parameters for 100TeV Run

After this run id done, we have to use Delphes separately to simulate the detector by going into Delphes Directory and writing the following command:

```
./DelphesHepMC cards/FCC/FCChh.tcl /home/student/output/
   ppT0h_hT0aa_100TeV_Cut.root /home/student/output/ppT0h_hT0aa_100Tev/
   Events/run_01/tag_1_pythia8_events.hepmc
```

3.3 The ROOT Analysis

After running the program, we will have a root file that was created from Delphes. This file contains the simulated data from the detector which can be analyzed using ROOT afterward. Also, we will have a file containing the resulting cross-sections which will be used in the ROOT analysis for calculating the luminosity of the generated events. See Figure 3.11 and 3.12

Results in the heft-full for p p > h > a a

Currently Running

Run Name	Tag Name	Cards	Results	Status/Jobs		
				Queued	Running	Done
run_01	tag_1	param_card run_card delphes_card	0.02088 ± 1.252e-05 (pb)	Done		

Available Results

Run	Collider	Banner	Cross section (pb)	Events	Data	Output	Action
run_01	p p 7000.0 x 7000.0 GeV	tag_1	0.02088 ± 1.3e-05	100000	parton madevent	LHE rootfile	remove run launch detector simulation
					pythia8	LOG HEPMC	remove run launch detector simulation
					delphes	LOG rootfile	remove run

Figure 3.11: Cross Section for the 14 TeV Run

Results in the heft-full for p p > h > a a

Currently Running

Run Name	Tag Name	Cards	Results	Status/Jobs		
				Queued	Running	Done
run_01	tag_1	param_card run_card	0.2928 ± 0.0001809 (pb)	Done		

Available Results

Run	Collider	Banner	Cross section (pb)	Events	Data	Output	Action
run_01	p p 50000.0 x 50000.0 GeV	tag_1	0.2928 ± 0.00018	100000	parton madevent	LHE rootfile	remove run launch detector simulation
					pythia8	LOG HEPMC	remove run launch detector simulation

Main Page

Figure 3.12: Cross Section for the 100 TeV Run

While we are in the directory of Delphes, we will create the C++ file and the header file by the following command:

```
Delphes -> MakeClass("analysis")
```

After that, we will have two files: analysis.C and analysis.h

We will type our C++ code in the analysis.C file. This code is written in the appendix with an explanation of each part.

I will explain the important parts of the codes in this section.

3.3.1 Code for the Parameters of 14 TeV Run

We will define our parameters like the cross-section which is retrieved from the MadGraph program. It must be written in units of Femto barn. After that, we will put an approximate luminosity for the actual events at the LHC. It will be 300 fb^{-1}. We have a generated data from Monte-Carlo and we need to relate it to the actual luminosity. Thus, we use a weight function, defined as the ratio of the luminosity of actual data and the luminosity of the generated data from MG5. We got the luminosity of the generated data by dividing our number of events (0) by the cross-section produced from MG5. MCLum $100000/20.88 = 4789.27$ fb^{-1} . The weight value will be $300/4789 = 0.062$

```
float cross_section = 20.88;  // The Cross-Section in fb
float N = 100000;    // Number of events
float DataLum = 300;        //  Luminosity of data in fb^-1
float MCLum = N / cross_section;  // Luminosity of Monte-Carlo  in fb^-1
float weight = DataLum / MCLum;       // Weight of the simulation
    process with respect to data
```

3.3.2 Code for the Parameters of 100 TeV Run

We will do the same definition of parameters like the previous case. However, here we will use a luminosity of 3000 fb^{-1} for the FCC (ten times greater than the luminosity for the LHC). MCLum $100000/292.78 = 341.55$ fb^{-1} . The weight value will be $3000/341.55 = 8.78$

The weight for the FCC divided by the weight of LHC gives $8.78/0.062 = 141$. Thus, we have a favored factor of 141 for the FCC over the LHC which will help in the increase of the number of generated Higgs Bosons in the case of the FCC over the LHC.

```
float cross_section = 292.78;  // The Cross-Section in fb
float N = 100000;    // Number of events
```

```
float DataLum = 3000;       //  Luminosity of data in fb^-1
float MCLum = N / cross_section;  // Luminosity of Monte-Carlo  in fb^-1
float weight = DataLum / MCLum;        // Weight of the simulation
    process with respect to data
```

Chapter 4

Results and Discussion

In this chapter, we will discuss the resulting data from the C++ code in the appendix. We will draw histograms for the following quantities:

1. Number of photons per event in LHC vs FCC
2. Transverse momentum for the leading photon
3. Transverse momentum for the sub-leading photon
4. Pseudorapidity for the leading photon
5. Pseudorapidity for the sub-leading photon
6. Invariant mass distribution for the Higgs Boson
7. Pseudorapidity for the Higgs Boson

4.1 Number of photons per event

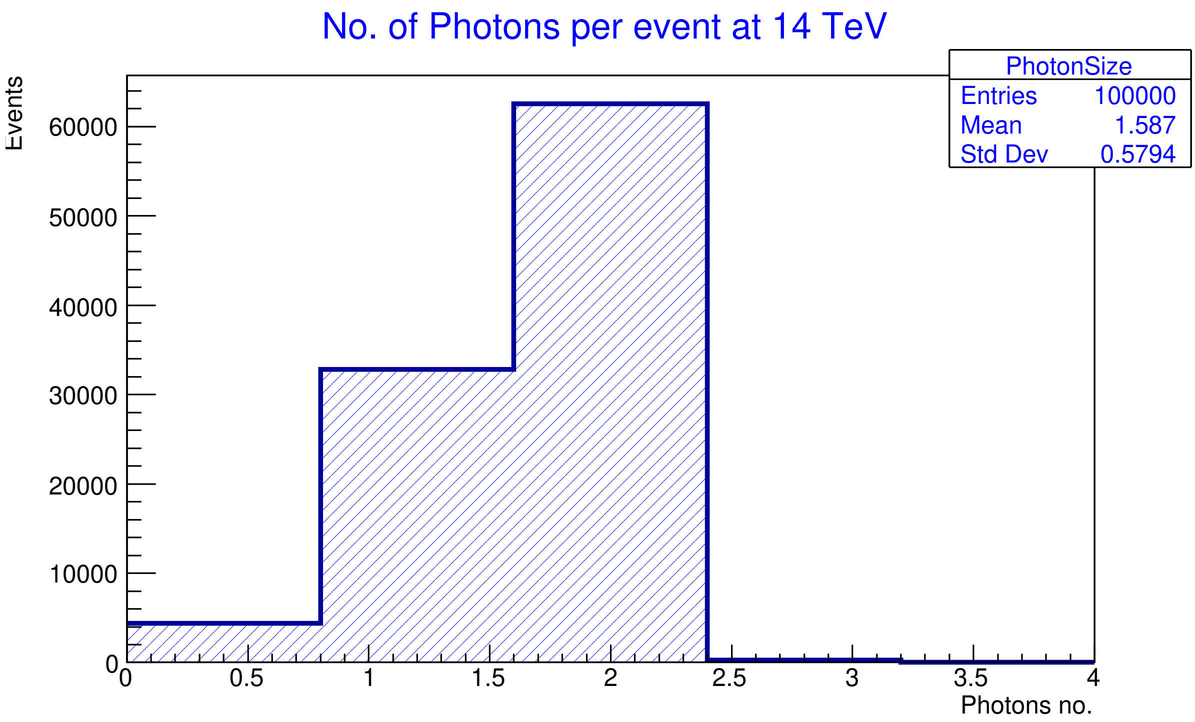

Figure 4.1: Number of photons per event at 14 TeV in the LHC

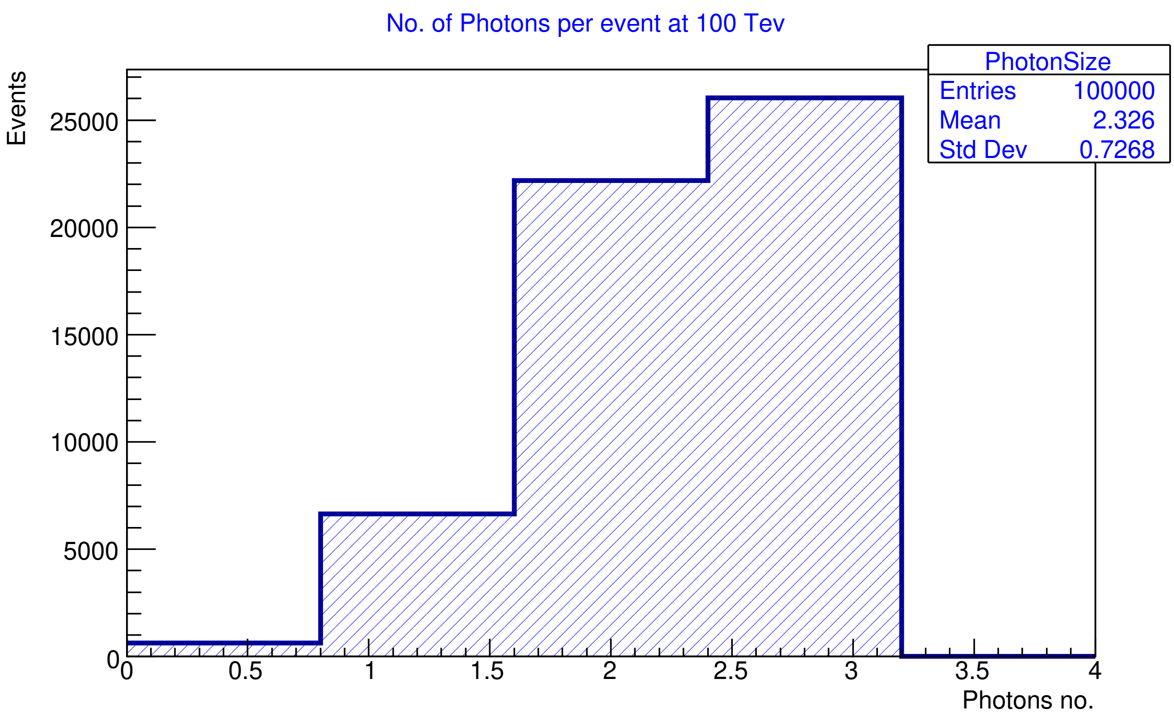

Figure 4.2: Number of photons per event at 100 TeV in the FCC

The above histograms show that the mean value for the number of photons is higher in the FCC case. This indicates higher chances of producing the Higgs in the FCC than LHC. We also should note that those two don't have the weighting function. This means the number of generated photons in the FCC for 3000 Luminosity is much higher than what is in the histograms. Also, the number of generated photons in LHC for luminosity of 300 is much less than what is shown in the histogram because the weight function in that case is 0.06.

By rough estimate, we can predict the number of produced Higgs in each case by multiplying by the weight function. For the LHC case: 60000*0.06 = 3600 which is near the actual value in the results of the Higgs events at the end of this chapter. For the FCC case: 23000*8.78= 210000.

4.2 Transverse momentum for the leading photon

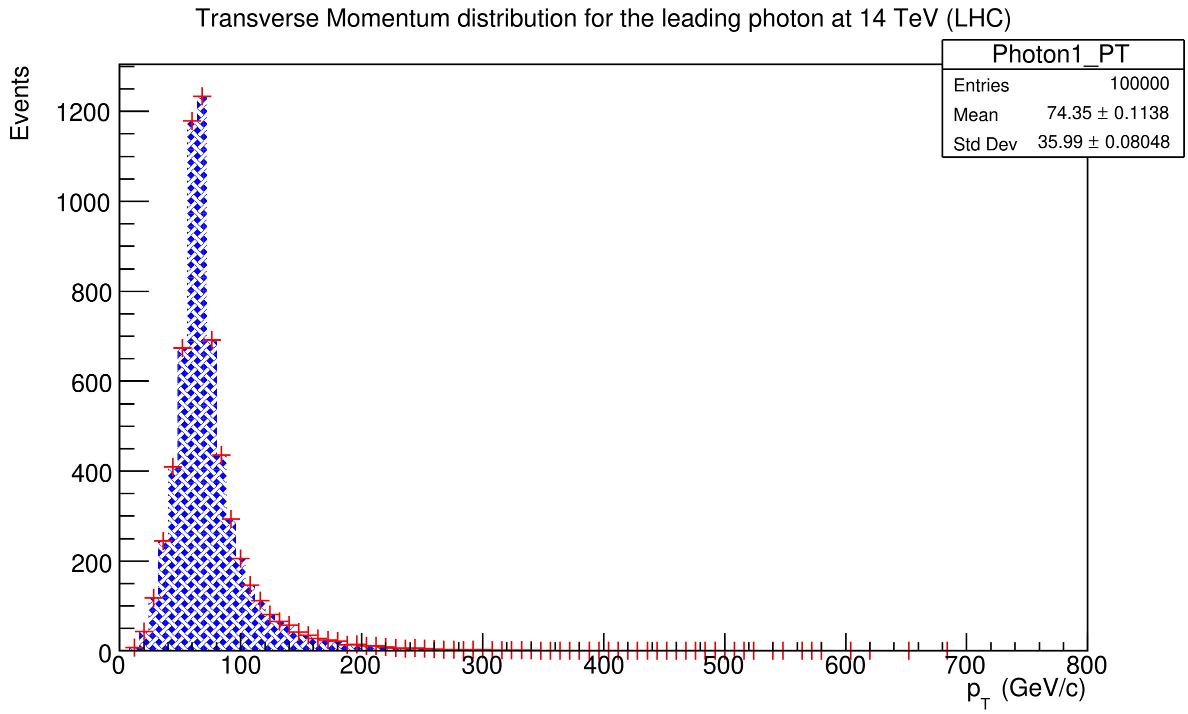

Figure 4.3: Transverse momentum for the leading photon at 14 TeV in the LHC

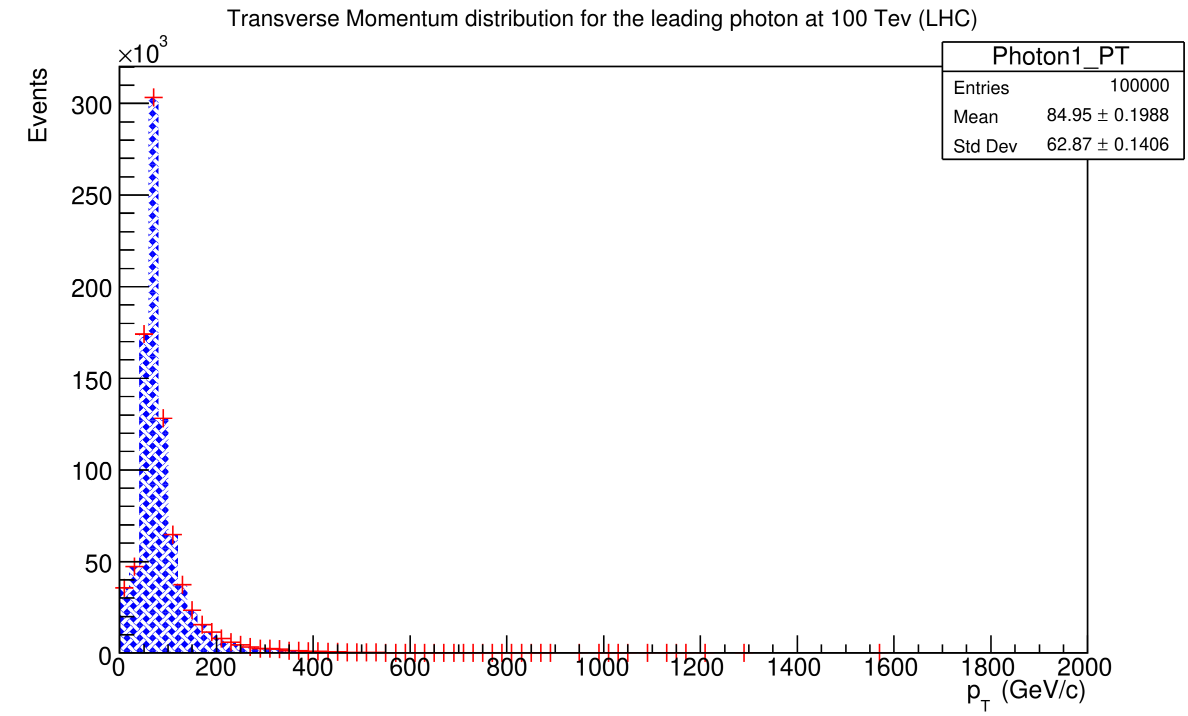

Figure 4.4: Transverse momentum for the leading photon at 100 TeV in the FCC

From the histograms above, we can see that there is higher transverse momentum for the leading photon in the case of FCC reaching 1600 GeV/c. This is helpful for making high momentum cuts in order to decrease the background and noise effects. This will help to distinguish the photons coming from noise and those coming from the Higgs Decay.

Moreover, the mean value of the transverse momentum for the FCC case is higher than the LHC case. This indicates that the higher the center of mass-energy, the higher the transverse momentum of the resulting leading photons.

4.3 Transverse momentum for the sub-leading photon

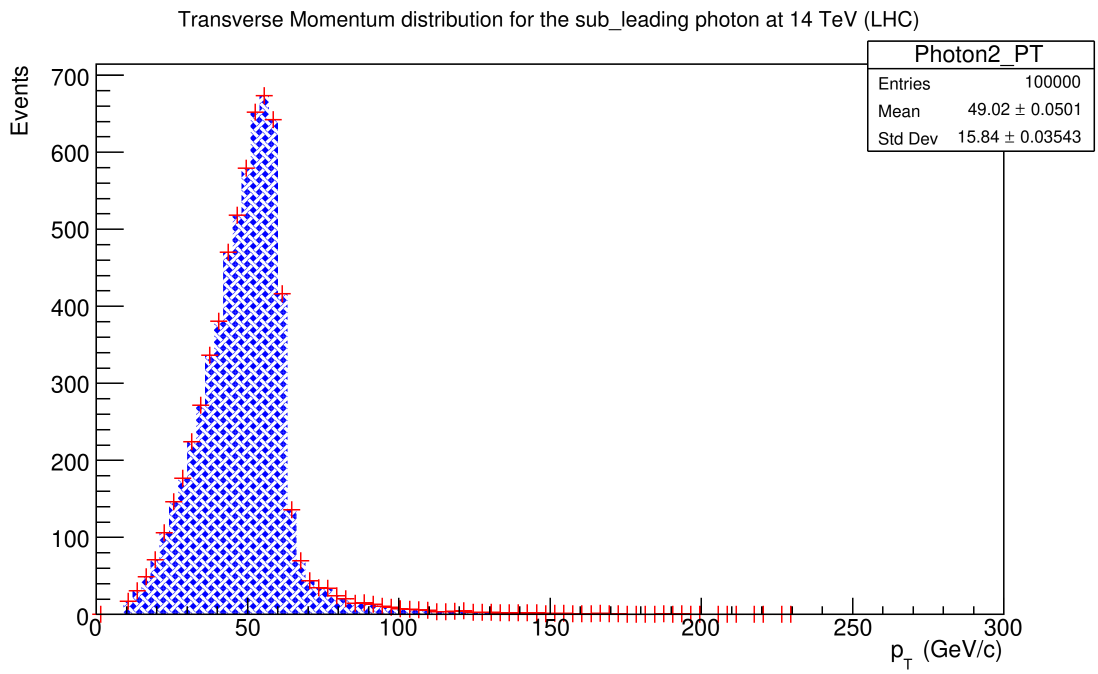

Figure 4.5: Transverse momentum for the sup-leading photon at 14 TeV in the LHC

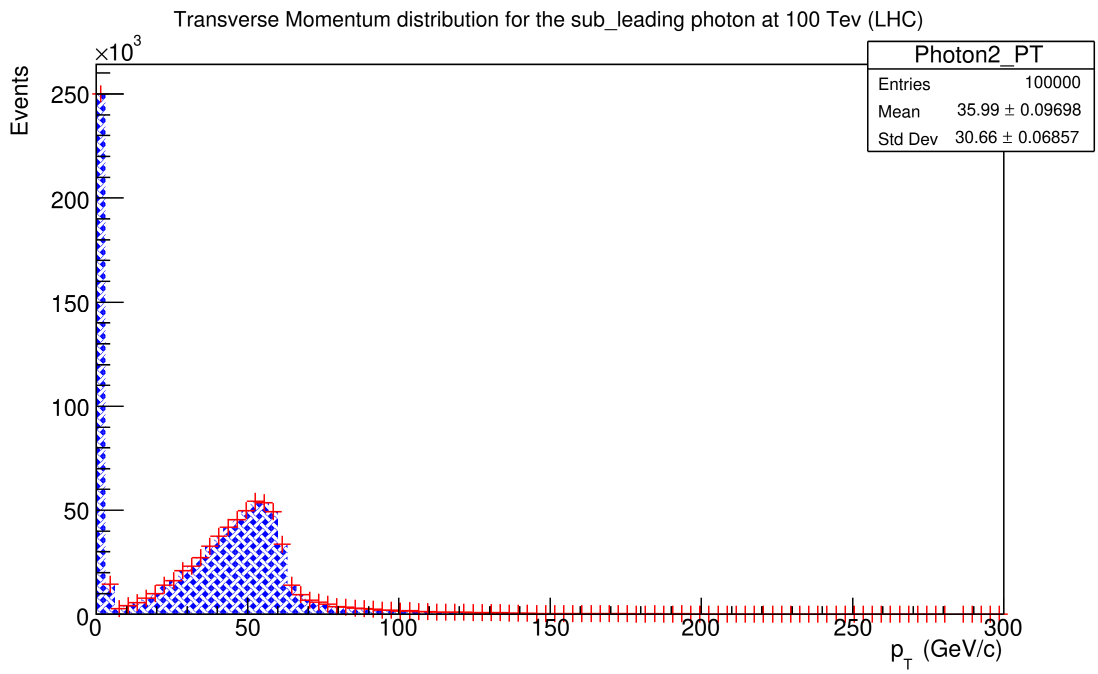

Figure 4.6: Transverse momentum for the sup-leading photon at 100 TeV in the FCC

In the transverse momentum histograms for the sub-leading photons, we note that their momentum is much less than the previous ones for the leading photons. This makes sense as the leading photons are the ones detected with high energies. Thus, they will also have higher transverse momentum.

Additionally, we can see that there is higher transverse momentum for the sup-leading photon in the case of FCC reaching 300 GeV/c. This is helpful for making high momentum cuts in order to decrease the background and noise effects. This will help to distinguish the photons coming from noise and those coming from the Higgs Decay.

4.4 Pseudorapidity for the leading photon

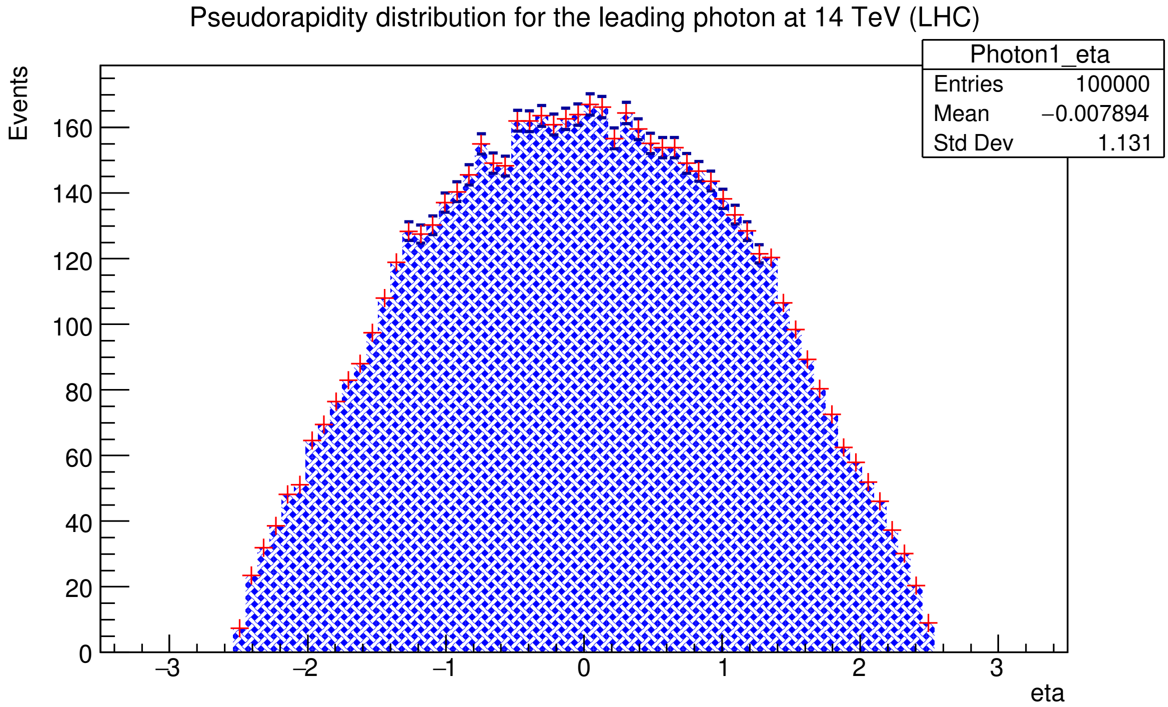

Figure 4.7: Pseudorapidity for the leading photon at 14 TeV in the LHC

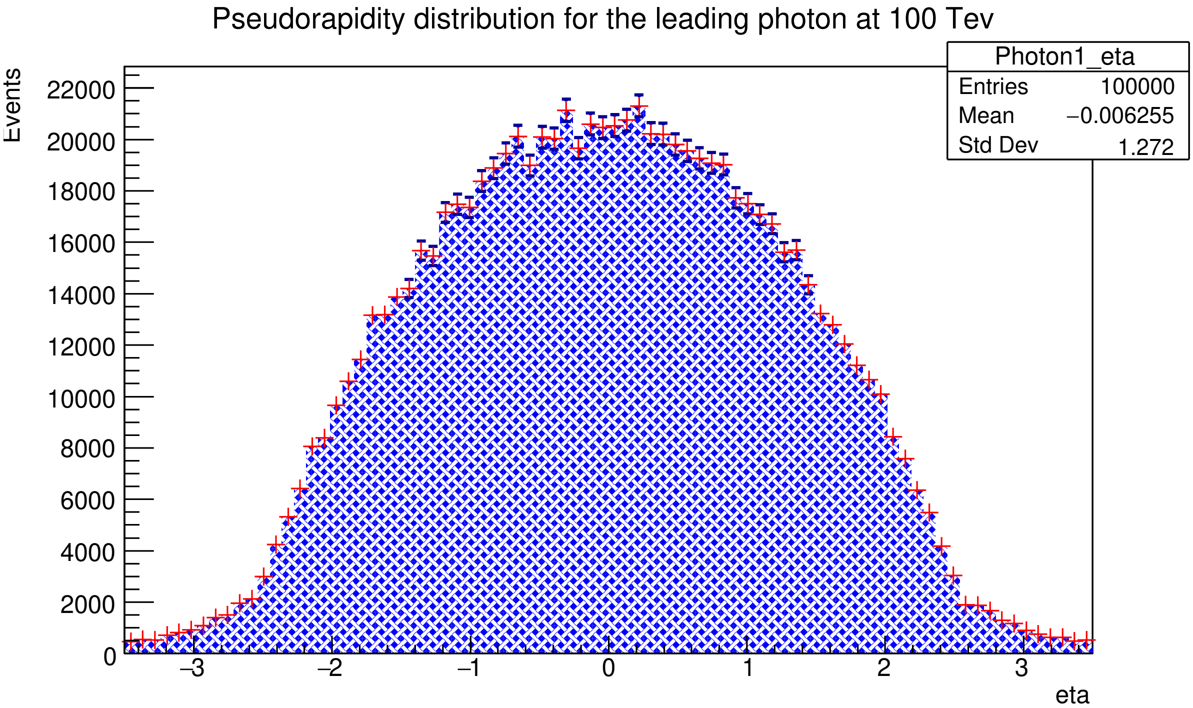

Figure 4.8: Pseudorapidity for the leading photon at 100 TeV in the FCC

From figure 2.7 and 2.8, the Pseudorapidity for the leading photon at LHC $-2.5 < \eta < 2.5$ while at FCC $-3.5 < \eta < 3.5$ This indicates that the spread in angles for the FCC is higher as a result, the FCC requires a better detector in order to be able to have more coverage to detect the new spread of leading photons.

Moreover, the Pseudorapidity is centered around the zero point. This is natural because there is no advantage for the positive axis over the negative axis.

4.5 Pseudorapidity for the sub-leading photon

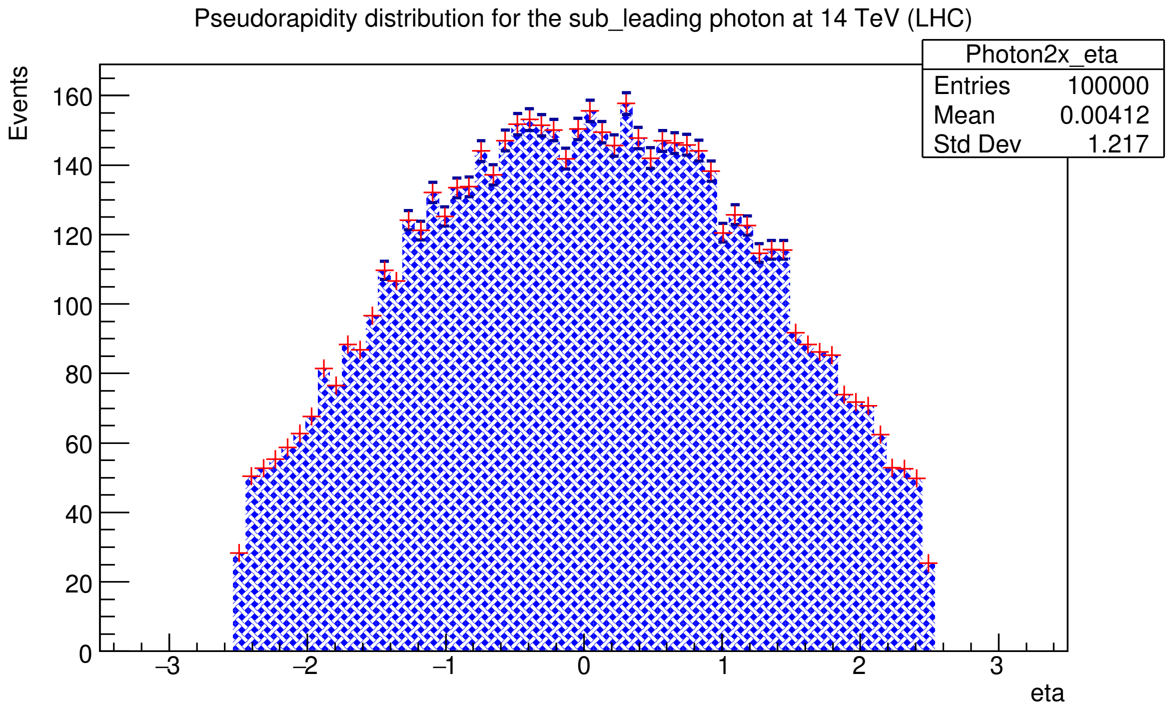

Figure 4.9: Pseudorapidity for the sup-leading photon at 14 TeV in the LHC

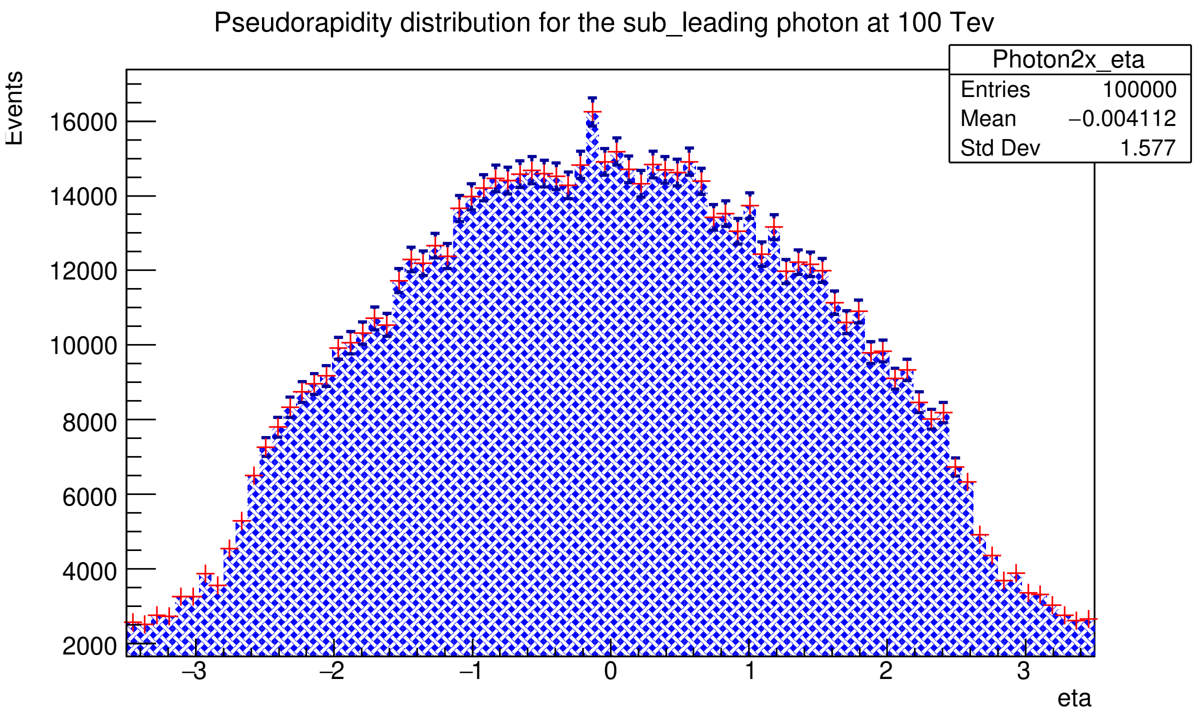

Figure 4.10: Pseudorapidity for the sup-leading photon at 100 TeV in the FCC

4.6 Invariant mass distribution for the Higgs Boson

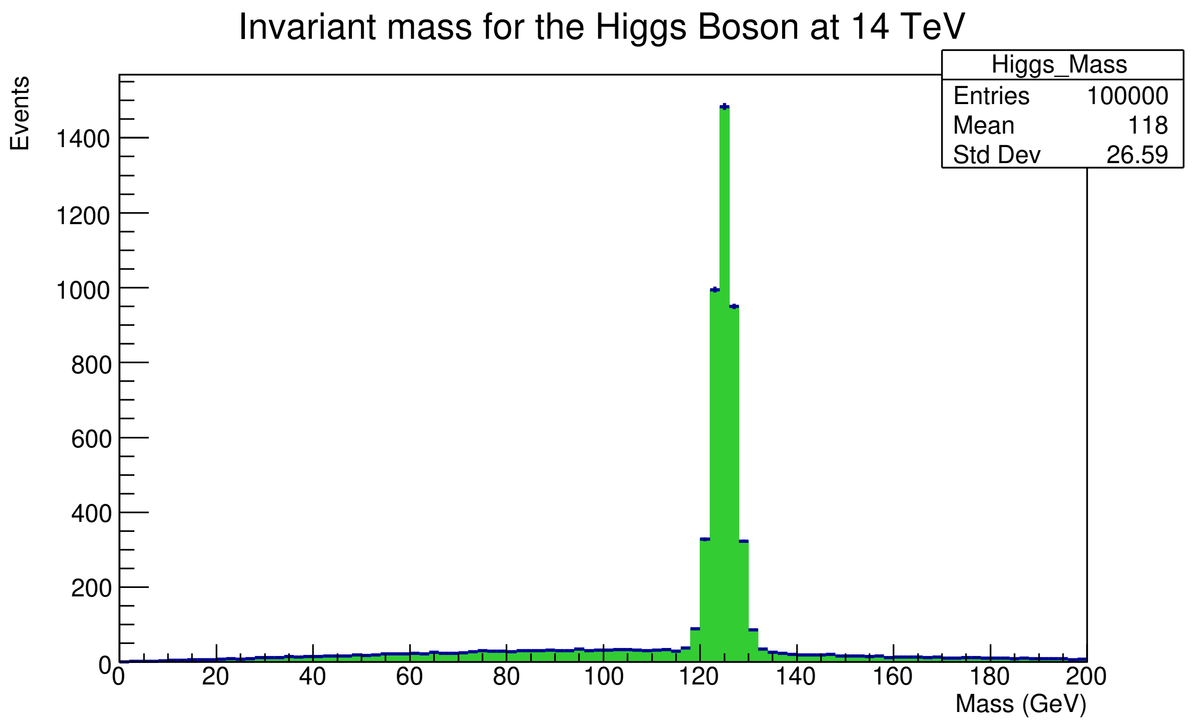

Figure 4.11: Invariant mass distribution for the Higgs Boson at 14 TeV in the LHC

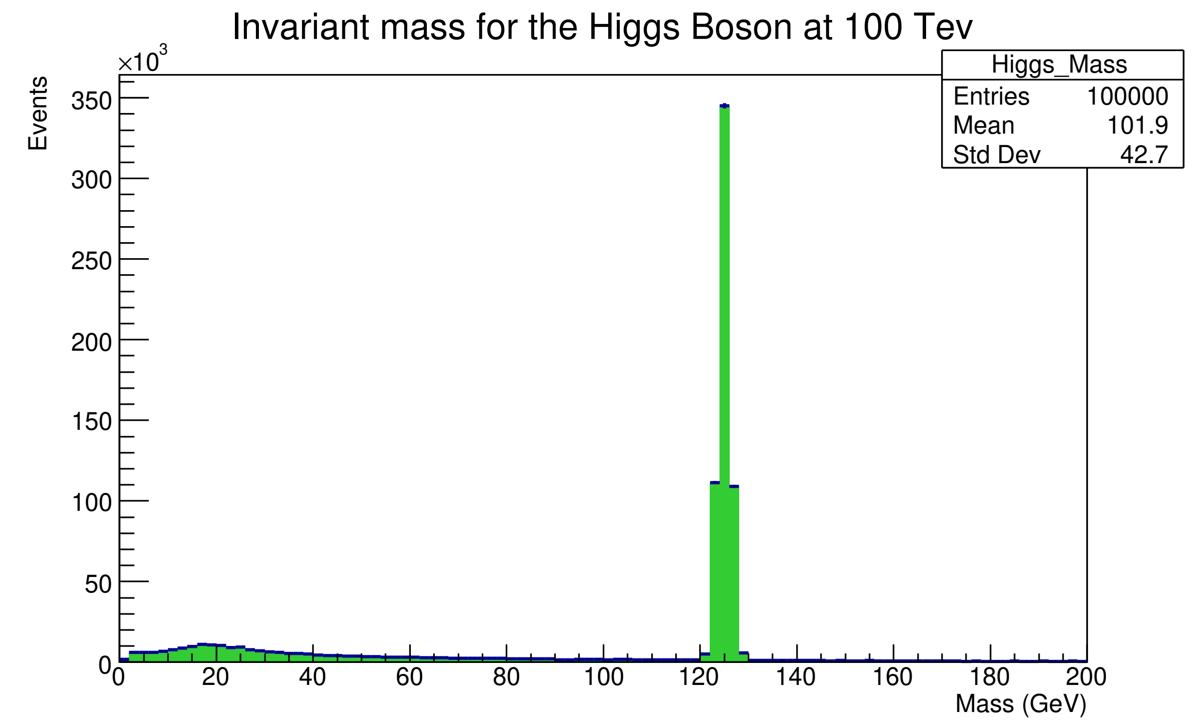

Figure 4.12: Invariant mass distribution for the Higgs Boson at 100 TeV in the FCC

In the two previous figures, the Higgs mass has a peak around 125 GeV as expected.

In the case of FCC, we have an enormous number of produced Higgs (350×10^3) in the range of 125 GeV. The ration between the number for FCC and LHC is $(350 \times 10^3)/1400 = 250$.

Moreover, in the case of FCC, we have less spread in the small range around the 125 GeV peak.

The standard deviation for the 100 TeV case is higher as we have a higher spread over all the 200 GeV x-axis. Which makes sense as we have a higher center of mass energy for the FCC case.

The mean value for the FCC is a little off, thus it requires further investigation. I will draw another graph by extending the x-axis to allow for investigations for energies higher than the 200 GeV to see why the mean is smaller in the FCC case than in the LHC case.

4.7 Pseudorapidity for the Higgs Boson

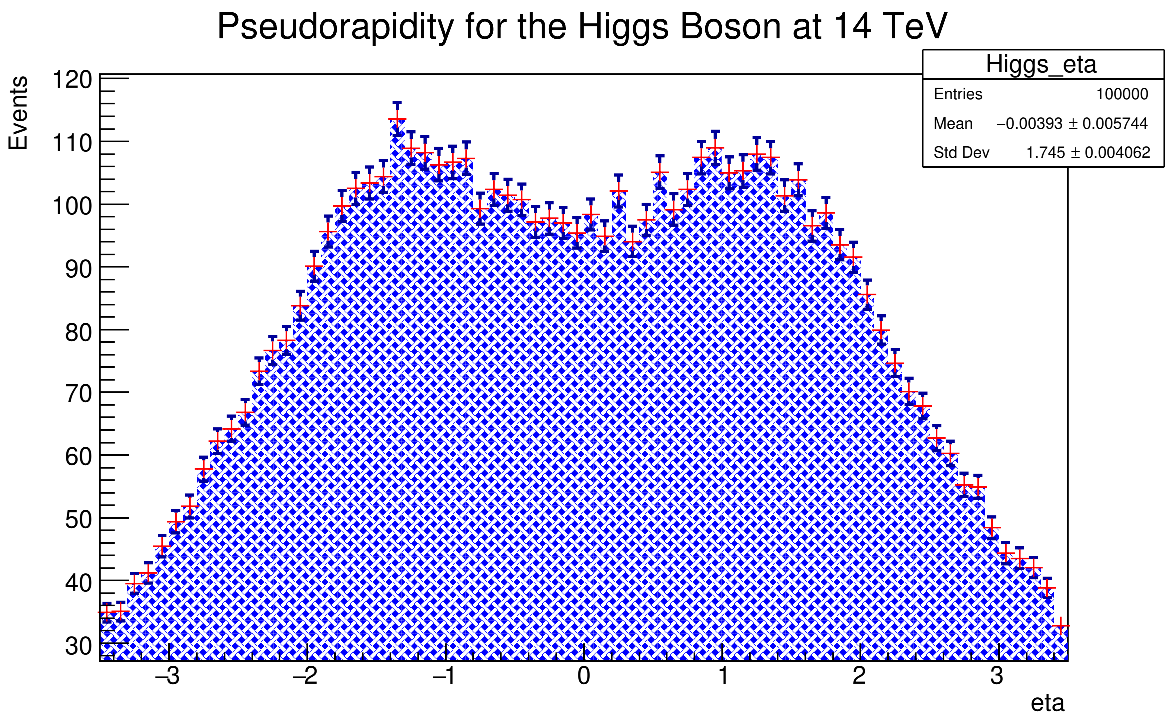

Figure 4.13: Pseudorapidity for the Higgs Boson at 14 TeV in the LHC

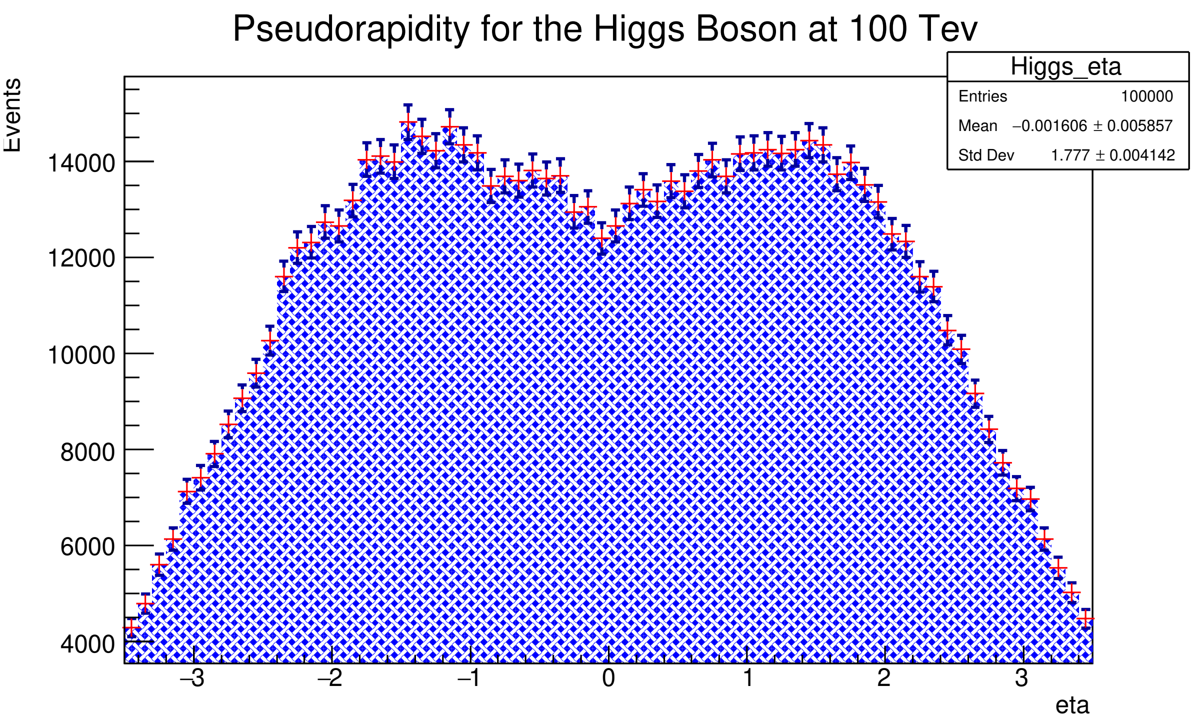

Figure 4.14: Pseudorapidity for the Higgs Boson at 100 TeV in the FCC

4.8 The mass of the Higgs in the FCC with an extended x-axis

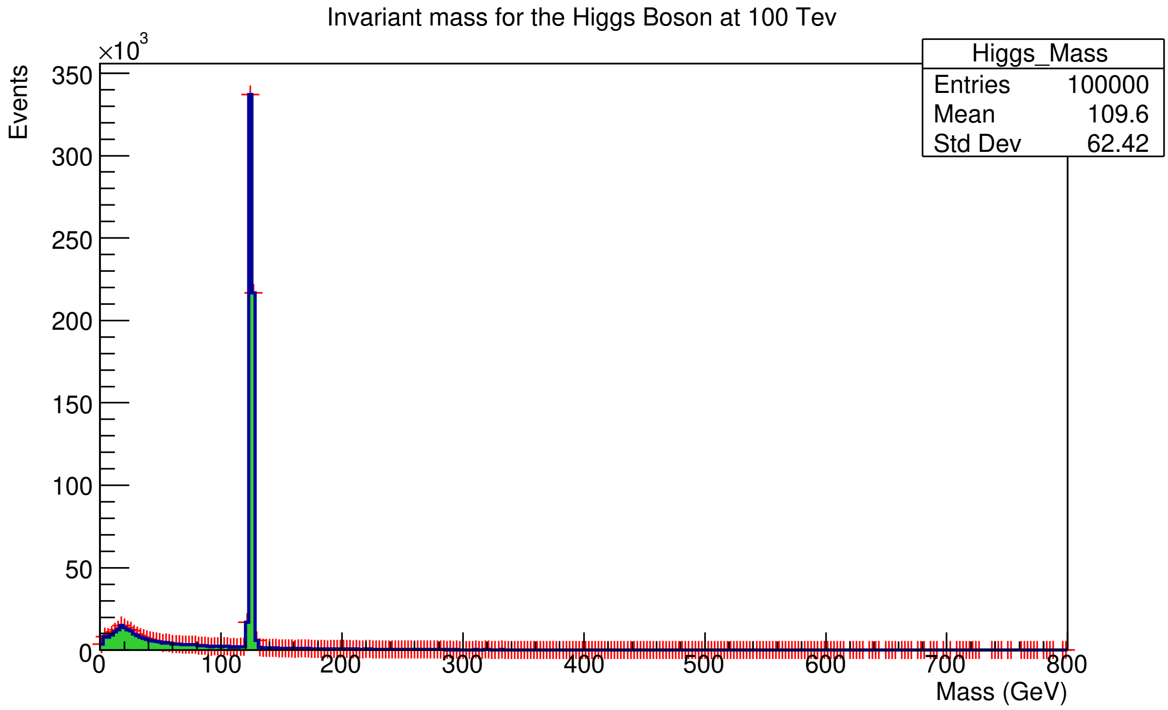

Figure 4.15: The mass of the Higgs in the FCC with an extended x-axis

As compared with the results from section 2.6, we notice that the standard deviation increased in Figure 2.15. This happened as we extended the x-axis for our histograms and we see new particles with masses over the 200 GeV. This might be really physics beyond the standard model and it might a gateway for new discoveries. Also, it might be errors in the reconstruction of the Higgs from photons. The programs use might confused those energetic photons and considered them as coming from the Higgs Decay and they might come from any other processes.

Chapter 5

Conclusion

According to the previous histograms and analysis, it is obvious that the FCC can produce a lot more Higgs Bosons than the LHC, providing a unique opportunity for further studying the Higgs and make precision measurements of the Higgs.

We even found produced particles with Masses higher than the Higgs mass in the FCC (in ranges above 200 GeV), which is an interesting finding to be studied for physics beyond the standard model.

The higher transverse momentum in the FCC can enable us to make higher cuts in order to separate the signal from the background. Thus, FCC provides more accurate measurements for the photons coming from the decay of the Higgs Boson.

The FCC detector is noticed to catch the higher Pseudorapidity in the case of 100 TeV, which provides evidence for its advantage over the LHC detector. The FCC detector has more coverage for the angles of the produced photons.

Finally, the FCC provides a more reliable way to study the Higgs and find new discoveries.

Appendix A

The C++ ROOT Analysis Code

```
          ////*****************////
 ///////The C++ code for the 14 TeV case :///////
        ////*****************////
//////////////

#define analysis_cxx
#include "analysis.h"
#include <TH2.h>
#include <TStyle.h>
#include <TCanvas.h>
#include <TROOT.h>
#include "TH1.h"
#include "TLorentzVector.h"
#include <TMath.h>
#include <iostream>
#include "THStack.h"
#include <TFile.h>
#include <iostream>
#include <iomanip>
#include <fstream>
#include <string>
#include <sstream>
#include <TObject.h>
#include <Math/Vector4D.h>
#include <Math/Vector4Dfwd.h>
#include <iostream>
```

```
#include <algorithm>
#include <cmath>

// these files contents are included while loading the analysis.c code

void analysis::Loop()
{
//   In a ROOT session, you can do:
//      root> .L analysis.C
//      root> analysis t
//      root> t.GetEntry(12); // Fill t data members with entry number
   12
//      root> t.Show();       // Show values of entry 12
//      root> t.Show(16);     // Read and show values of entry 16
//      root> t.Loop();       // Loop on all entries
//

//     This is the loop skeleton where:
//    jentry is the global entry number in the chain
//    ientry is the entry number in the current Tree
//  Note that the argument to GetEntry must be:
//    jentry for TChain::GetEntry
//    ientry for TTree::GetEntry and TBranch::GetEntry
//
//       To read only selected branches, Insert statements like:
// METHOD1:
//    fChain->SetBranchStatus("*",0);  // disable all branches
//    fChain->SetBranchStatus("branchname",1);  // activate branchname
// METHOD2: replace line
//    fChain->GetEntry(jentry);       //read all branches
//by  b_branchname->GetEntry(ientry); //read only this branch
   if (fChain == 0) return;

   Long64_t nentries = fChain->GetEntriesFast();

   TFile *inputfile = new TFile("tag_1_delphes_events.root", "READ");
   TFile *outputfile = new TFile("Output_14Tev.root", "UPDATE" );

```

```
// these two takes input from the tag_1_delphes root file which was created in first place from the delphes program and creates another file with the output of the code with name "Output_14Tev.root"


//""""""""""Now we will define the parametrs for the run and put the weights for the process""""""""

float cross_section = 20.88;  // The Cross-Section in fb
float N = 100000;    // Number of events
float DataLum = 300;        //  Luminosity of data in fb^-1
float MCLum = N / cross_section;  // Luminosity of Monte-Carlo  in fb^-1
float weight = DataLum / MCLum;        // Weight of the simulation process with repsect to data



// **************Now we will create the histograms****************


 //Histogram for the Photon size

TH1I *PhotonSize = new TH1I("PhotonSize", "No. of Photons per event at 14 TeV ", 5, 0, 4);
   PhotonSize->GetXaxis()->SetTitle("Photons no.");
   PhotonSize->GetYaxis()->SetTitle("Events");

//Histograms for the Transverse Momentum PT and the Pseudorapidity eta
// photon 1 will be the leading photon and photon 2 will be the sub-leading photon

TH1D *Photon1_PT = new TH1D("Photon1_PT", "Transverse Momentum distribution for the leading photon at 14 TeV (LHC)", 100, 0., 800);
   Photon1_PT->GetXaxis()->SetTitle("p_{T} (GeV/c)");
   Photon1_PT->GetYaxis()->SetTitle("Events");

   TH1D *Photon1_eta = new TH1D("Photon1_eta", "Pseudorapidity distribution for the leading photon at 14 TeV (LHC)", 80, -3.5, 3.5);
   Photon1_eta->GetXaxis()->SetTitle("eta");
```

```
   Photon1_eta->GetYaxis()->SetTitle("Events");

TH1D *Photon2_PT = new TH1D("Photon2_PT", "Transverse Momentum distribution for the sub_leading photon at 14 TeV (LHC)", 100, 0., 300);
   Photon2_PT->GetXaxis()->SetTitle("p_{T} (GeV/c)");
   Photon2_PT->GetYaxis()->SetTitle("Events");

   TH1D *Photon2_eta = new TH1D("Photon2x_eta", "Pseudorapidity distribution for the sub_leading photon at 14 TeV (LHC)", 80, -3.5, 3.5);
   Photon2_eta->GetXaxis()->SetTitle("eta");
   Photon2_eta->GetYaxis()->SetTitle("Events");


// ****The histograms for the Mass and eta of the Higgs

TH1D *Higgs_Mass = new TH1D("Higgs_Mass", "Invariant mass for the Higgs Boson at 14 TeV", 100, 0., 200.);
   Higgs_Mass->GetXaxis()->SetTitle("Mass (GeV)");
   Higgs_Mass->GetYaxis()->SetTitle("Events");

   TH1D *Higgs_eta = new TH1D("Higgs_eta", "Pseudorapidity for the Higgs Boson at 14 TeV", 70, -3.5, 3.5);
   Higgs_eta->GetXaxis()->SetTitle("eta");
   Higgs_eta->GetYaxis()->SetTitle("Events");




   Long64_t nbytes = 0, nb = 0;
   for (Long64_t jentry=0; jentry<nentries;jentry++) {
      Long64_t ientry = LoadTree(jentry);
      if (ientry < 0) break;
      nb = fChain->GetEntry(jentry);   nbytes += nb;
      // if (Cut(ientry) < 0) continue;

// THe TLorent Vector

```

```
TLorentzVector  photon0, photon1, h;


  photon0.SetPtEtaPhiE(Photon_PT[0], Photon_Eta[0], Photon_Phi[0],
    Photon_E[0]);
  photon1.SetPtEtaPhiE(Photon_PT[1], Photon_Eta[1], Photon_Phi[1],
    Photon_E[1]);



// The Higgs is:

h=photon0 + photon1;


// Now we fill the histograms:

PhotonSize->Fill(Photon_size);
Photon1_PT->Fill(Photon_PT[0], weight);
Photon1_eta->Fill(Photon_Eta[0], weight);
Photon2_PT->Fill(Photon_PT[1], weight);
Photon2_eta->Fill(Photon_Eta[1], weight);
Higgs_Mass->Fill(h.M(), weight);
Higgs_eta->Fill(h.Eta(), weight);

   }
// the for loop is now ended


// Now we write the histograms
PhotonSize->Write();
Photon1_PT->Write();
Photon1_eta->Write();
Photon2_PT->Write();
Photon2_eta->Write();
Higgs_Mass->Write();
Higgs_eta->Write();

// Now we draw the histograms
```

```
PhotonSize->Draw();
Photon1_PT->Draw();
Photon1_eta->Draw();
Photon2_PT->Draw();
Photon2_eta->Draw();
Higgs_Mass->Draw();
Higgs_eta->Draw();


outputfile->Close();

} // end of loop member function






          ////*****************////
  //////The C++ code for the 100 TeV case ://////
         ////*****************////




#define analysis_cxx
#include "analysis.h"
#include <TH2.h>
#include <TStyle.h>
#include <TCanvas.h>
#include <TROOT.h>
#include "TH1.h"
#include "TLorentzVector.h"
#include <TMath.h>
#include <iostream>
#include "THStack.h"
#include <TFile.h>
#include <iostream>
#include <iomanip>
```

```
#include <fstream>
#include <string>
#include <sstream>
#include <TObject.h>
#include <Math/Vector4D.h>
#include <Math/Vector4Dfwd.h>
#include <iostream>
#include <algorithm>
#include <cmath>

// these files contents are included while loading the analysis.c code

void analysis::Loop()
{
//   In a ROOT session, you can do:
//      root> .L analysis.C
//      root> analysis t
//      root> t.GetEntry(12); // Fill t data members with entry number
   12
//      root> t.Show();       // Show values of entry 12
//      root> t.Show(16);     // Read and show values of entry 16
//      root> t.Loop();       // Loop on all entries
//

//     This is the loop skeleton where:
//    jentry is the global entry number in the chain
//    ientry is the entry number in the current Tree
//  Note that the argument to GetEntry must be:
//    jentry for TChain::GetEntry
//    ientry for TTree::GetEntry and TBranch::GetEntry
//
//       To read only selected branches, Insert statements like:
// METHOD1:
//    fChain->SetBranchStatus("*",0);  // disable all branches
//    fChain->SetBranchStatus("branchname",1);  // activate branchname
// METHOD2: replace line
//    fChain->GetEntry(jentry);       //read all branches
//by  b_branchname->GetEntry(ientry); //read only this branch
   if (fChain == 0) return;
```

```
   Long64_t nentries = fChain->GetEntriesFast();

   TFile *inputfile = new TFile("ppT0h_hT0aa_100TeV_Cut.root", "READ");
   TFile *outputfile = new TFile("Output(2)_100Tev.root", "UPDATE" );

// these two takes input from the tag_1_delphes root file which was
    created in first place from the delphes program and creates another
    file with the output of the code with name "Output_100Tev.root"


//""""""""""Now we will define the parametrs for the run and put the
    weights for the process""""""""

float cross_section = 292.78;  // The Cross-Section in fb
float N = 100000;    // Number of events
float DataLum = 3000;      //  Luminosity of data in fb^-1
float MCLum = N / cross_section;  // Luminosity of Monte-Carlo  in fb^-1
float weight = DataLum / MCLum;       // Weight of the simulation
    process with repsect to data



// **************Now we will create the histograms****************


 //Histogram for the Photon size

TH1I *PhotonSize = new TH1I("PhotonSize", "No. of Photons per event at
    100 Tev ", 5, 0, 4);
   PhotonSize->GetXaxis()->SetTitle("Photons no.");
   PhotonSize->GetYaxis()->SetTitle("Events");

//Histograms for the Transverse Momentum PT and the Pseudorapidity eta
// photon 1 will be the leading photon and photon 2 will be the sub-
    leading photon

TH1D *Photon1_PT = new TH1D("Photon1_PT", "Transverse Momentum
    distribution for the leading photon at 100 Tev (LHC)", 100, 0., 2000)
```

```
    ;
    Photon1_PT->GetXaxis()->SetTitle("p_{T} (GeV/c)");
    Photon1_PT->GetYaxis()->SetTitle("Events");

    TH1D *Photon1_eta = new TH1D("Photon1_eta", "Pseudorapidity
    distribution for the leading photon at 100 Tev (LHC)", 80, -3.5, 3.5)
    ;
    Photon1_eta->GetXaxis()->SetTitle("eta");
    Photon1_eta->GetYaxis()->SetTitle("Events");

TH1D *Photon2_PT = new TH1D("Photon2_PT", "Transverse Momentum
    distribution for the sub_leading photon at 100 Tev (LHC)", 100, 0.,
    300);
    Photon2_PT->GetXaxis()->SetTitle("p_{T} (GeV/c)");
    Photon2_PT->GetYaxis()->SetTitle("Events");

    TH1D *Photon2_eta = new TH1D("Photon2x_eta", "Pseudorapidity
    distribution for the sub_leading photon at 100 Tev (LHC)", 80, -3.5,
    3.5);
    Photon2_eta->GetXaxis()->SetTitle("eta");
    Photon2_eta->GetYaxis()->SetTitle("Events");


// ****The histograms for the Mass and eta of the Higgs

TH1D *Higgs_Mass = new TH1D("Higgs_Mass", "Invariant mass for the Higgs
    Boson at 100 Tev", 100, 0., 200.);
    Higgs_Mass->GetXaxis()->SetTitle("Mass (GeV)");
    Higgs_Mass->GetYaxis()->SetTitle("Events");

    TH1D *Higgs_eta = new TH1D("Higgs_eta", "Pseudorapidity for the Higgs
     Boson at 100 Tev", 70, -3.5, 3.5);
    Higgs_eta->GetXaxis()->SetTitle("eta");
    Higgs_eta->GetYaxis()->SetTitle("Events");




    Long64_t nbytes = 0, nb = 0;
```

```
   for (Long64_t jentry=0; jentry<nentries;jentry++) {
      Long64_t ientry = LoadTree(jentry);
      if (ientry < 0) break;
      nb = fChain->GetEntry(jentry);   nbytes += nb;
      // if (Cut(ientry) < 0) continue;

// THe TLorent Vector

TLorentzVector  photon0, photon1, h;


  photon0.SetPtEtaPhiE(Photon_PT[0], Photon_Eta[0], Photon_Phi[0],
   Photon_E[0]);
  photon1.SetPtEtaPhiE(Photon_PT[1], Photon_Eta[1], Photon_Phi[1],
   Photon_E[1]);



// The Higgs is:
h=photon0 + photon1;


// Now we fill the histograms:

PhotonSize->Fill(Photon_size);
Photon1_PT->Fill(Photon_PT[0], weight);
Photon1_eta->Fill(Photon_Eta[0], weight);
Photon2_PT->Fill(Photon_PT[1], weight);
Photon2_eta->Fill(Photon_Eta[1], weight);
Higgs_Mass->Fill(h.M(), weight);
Higgs_eta->Fill(h.Eta(), weight);

   }
// the for loop is now ended


// Now we write the histograms
PhotonSize->Write();
Photon1_PT->Write();
```

```
Photon1_eta->Write();
Photon2_PT->Write();
Photon2_eta->Write();
Higgs_Mass->Write();
Higgs_eta->Write();

// Now we draw the histograms
PhotonSize->Draw();
Photon1_PT->Draw();
Photon1_eta->Draw();
Photon2_PT->Draw();
Photon2_eta->Draw();
Higgs_Mass->Draw();
Higgs_eta->Draw();


outputfile->Close();

} // end of loop member function
```

Bibliography

[1] The atlas detector. https://atlas.cern/discover/detector . Accessed: 2021-06.

[2] The cms detector. https://cms.cern/ . Accessed: 2021-06.

[3] Doe explains...the standard model of particle physics. https://www.energy.gov/science/doe-explainsthe-standard-model-particle-physics. Accessed: 2021-06.

[4] Fcc. https://fcc.web.cern.ch/Pages/About.aspx . Accessed: 2021-06.

[5] The fcc-ee in a few words. https://fcc-ee.web.cern.ch/ . Accessed: 2021-06.

[6] Fcc-hh design study. https://fcc.web.cern.ch/Pages/fcc-hh.aspx . Accessed: 2021-06.

[7] Future circular collider. https://home.cern/science/accelerators/future-circular-collider . Accessed: 2021-06.

[8] The higgs boson. https://home.cern/science/physics/higgs-boson. Accessed: 2021-06.

[9] Higgs boson production. http://opendata.atlas.cern/books/current/get-started/$_book/the-higgs-boson.html. Accessed: 2021-06.$

[10] The large hadron collider. https://home.cern/science/accelerators/large-hadron-collider/ . Accessed: 2021-06.

[11] The lhc. https://people.ece.uw.edu/hauck/LargeHadronCollider/ . Accessed: 2021-06.

[12] The madgraph5-amc@nlo homepage. http://madgraph.phys.ucl.ac.be/. Accessed: 2021-06-29.

[13] Madgraph5-amc@nlo web page. http://amcatnlo.web.cern.ch/amcatnlo/. Accessed: 2021-06-29.

[14] Madgraph5_amc@nlo. https://launchpad.net/mg5amcnlo . Accessed: 2021-06-29.

[15] Root- data analysis framework. https://root.cern/about/. Accessed: 2021-06.

[16] The standard model. https://home.cern/science/physics/standard-model. Accessed: 2021-06.

[17] Unitarization models for vector boson scattering at the lhc. https://www.researchgate.net/figure/Coordinate-system-used-by-the-ATLAS-and-CMS-experiments-at-the-LHC-33$_fig4_3$26689955.$Accessed: 2021 - 06$.

[18] Abdullayev, S. K., Gojayev, M. S., and Saddigh, F. A. Decay channels of the standard Higgs boson. *Moscow Univ. Phys. Bull. 72*, 4 (2017), 329–339.

[19] Brun, R., and Rademakers, F. ROOT: An object oriented data analysis framework. *Nucl. Instrum. Meth. A 389* (1997), 81–86.

[20] Chatrchyan, S., et al. The CMS Experiment at the CERN LHC. *JINST 3* (2008), S08004.

[21] de Favereau, J., Delaere, C., Demin, P., Giammanco, A., Lemaître, V., Mertens, A., and Selvaggi, M. Delphes 3: a modular framework for fast simulation of a generic collider experiment. *Journal of High Energy Physics 2014*, 2 (Feb 2014).

[22] Griffiths, D. J. *Introduction to Elementary Particles.* 2008.

[23] Sjöstrand, T., Ask, S., Christiansen, J. R., Corke, R., Desai, N., Ilten, P., Mrenna, S., Prestel, S., Rasmussen, C. O., and Skands, P. Z. An introduction to pythia 8.2. *Computer Physics Communications 191* (Jun 2015), 159–177.