1-1-2013

Search For Contact Interactions In The Dielectron Channel In P-P Collisions At Sqrt(s) = 8 Tev At Cms

Pramod Lamichhane
Wayne State University,

SEARCH FOR CONTACT INTERACTIONS IN THE DIELECTRON CHANNEL IN P-P COLLISIONS AT $\sqrt{s} = 8$ TeV AT CMS

by

PRAMOD LAMICHHANE

DISSERTATION

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfilment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

2013

MAJOR : PHYSICS

Approved by:

_____

Advisor   Date

_____

_____

_____

DEDICATION

To My parents, Shiva Raj and Sushila Devi,

have no words to thank for your unconditional support and love,

My beautiful loving wife, Santoshi, without you, I would be nothing,

and

My dearest babies Prashna and Samip ,

You, showed me a new window of love, from which I see a beautiful life out there.

ACKNOWLEDGEMENTS

I am thankful to many Professors, fellow students and the administrative staffs at Department of Physics and Astronomy of the Wayne State University. My sincere gratitude goes to our department chair Prof. Ratna Naik for her constant support, encouragement and help. I'm grateful to Dr. Scott Payson and Prof. P.K.Kuo for their ideas and skill in order to run a undergraduate lab class, Delores Cowen and Doris King for helping me in different administrative processes. I would also like to appreciate Graduate advisor Dr. Jogindra Wahdera being so much helpful and being always there when needed. In addition to them, It is my honor to remember my dearest friends Indika Wanniwarachi, Khadije Bazzi, Indermeet Kohli and Pairadeau Mars for their help and friendship during the course work.

Last but not the least, I would like to thank my family for supporting me spiritually throughout my life. I have no words to thank My parents Shiva Raj and Sushila Devi and My dear wife Santoshi who have been supporting me unconditionally in every single decisions I have made. My loving thanks to my sweet children Prashna and Samip who have been very cooperative throughout my research work.

iv

TABLE OF CONTENTS

LIST OF FIGURES

xiv

LIST OF TABLES

Chapter 1

Introduction

In this 21st century, physics has succeeded to give explanation for so many mysteries in our universe. Due to growing technological advancement, high energy particle physics, which deals with the study of how the smallest building blocks of matter interact with each other through forces, has been progressing by probing deeper and deeper into the subatomic scale. With all of these efforts, a theory concerning the electroweak and strong nuclear interactions called the standard model (SM) has been tremendously successful in explaining a wide variety of experimental results, including the SM Higgs boson (possibly discovered at the LHC [1] [2]) that explains the origin of electro-weak spontaneous symmetry breaking necessary to give masses to the observed gauge bosons. However, the standard model falls short of being a complete theory of fundamental interactions, as the SM does not incorporate gravity does not predict many phenomena such as dark energy, neutrino oscillations [3] and the non–zero masses of neutrinos. Additionally, the SM has some unnatural properties within its theoretical framework, leading to puzzles like strong CP [4] and the hierarchy problem. In order to address these limitations, many theories beyond the standard model (BSM) have been developed and are being tested with increasingly more sensitive experiments. One way to model a BSM theory is by introducing a new interaction in a very high energy scale in the form of a 4-fermion contact interaction. This is similar to the approach used by Fermi

to describe nuclear β decay [5], which could explore the large extra spatial dimensions or quark/lepton compositeness for parton interaction well below the actual interaction scale. In order to probe new physics at the TeV scale, the LHC particle accelerator was constructed by the European Organization for Nuclear Research (CERN) with the aim of allowing physicists to test the predictions of different theories of particle physics, particularly the existence of the SM Higgs boson or Higgs bosons in extension to the standard model. One of the four main detectors at the LHC, the CMS detector is designed with the goals of discovering the Higgs boson, looking for evidence of physics beyond the standard model, and studying aspects of heavy ion collisions. The analysis presented in this thesis is based on studying LHC data collected by the CMS detector in 2012 at center of mass energy of 8 TeV. Without knowledge of the exact intermediate particle exchange mechanism, a search for a new physics signature in the form of a 4-fermion contact interaction due to the assumed substructure of quarks and leptons is performed in the high mass region of the Drell-Yan process for pairs of electrons by looking for an excess of events over what is predicted by the standard model. This thesis is organized as follows. Chapter 2 discusses the standard model as well as the motivation for the BSM theory of CI, that is being used in this analysis. At the end, activities in the CMS collaboration in the compositeness search and some previous work on the contact interaction will be explained. In section 3.1, the physics of p-p collisions along with the parton distribution functions and a few generators are discussed. In the beginning of the Chapter 4, the design and detailed structure of the CMS experiment is given. Chapter 5 gives the Pythia implementation in the context of contact interaction with left left isoscalar model. In Chapter 6, a quick summary is given regarding the overview of the method of the CI analysis. Since this thesis is restricted to the CI analysis in the di-electron channel, a brief description of the electron reconstruction and the identifaction in the CMS detector is presented in Chapter 7. The data-driven method for the estimation of electrons from jets and the electron charge misidentification are also explained in this chapter. In Chapter 8, a detailed description of the data and Monte Carlo simulation (MC) used in this analysis is

given. An explanation of higher order correction is given in Chapter 9. In Chapter 10, the theoretical and the experimental systematic uncertainties are discussed. Since the largest source of systematic uncertainty comes from the parton distribution function, a detailed explanation is given for the method used to estimate the PDF uncertainty in this Chapter. In Chapter 11, the flavor of the technical aspects of the data analysis in the CMS collaboration is discussed in the context of the Data-MC spectrum comparison. Chapter 12 gives some discussion of the statistical methods used in high-energy physics and a detailed explanation of the particular method used in this analysis. Finally, in Chapter 13, a summary of the results obtained for this analysis and future plans are discussed.

Chapter 2

Theoretical background

This chapter introduces the standard model (SM) of particle physics, which has successfully explained almost all laboratory results to date. In section 2.2, some of the incompleteness of the SM is discussed followed by the motivation for beyond the SM models. Along with a review of the previous searches, section 2.2.1 introduces the four fermion contact interaction model including possible experimental signatures. Finally, in section 2.2.4, a short description of a previous search using a similar analysis is given.

2.1 The standard Model of particle physics

The standard model is a widely accepted theory which has successfully described the fundamental interactions of nature, within the framework of relativistic quantum field theory (QFT [6]) in which the fields are quantized and the particles represent excitations of quantized fields. The gauge group, $SU(3)_C \times SU(2)_L \times U(1)_Y$ conserves color charge, electric charge, weak isospin and weak hypercharge. Despite its success in explaining other interactions, the standard model does not include gravity.

2.1.1 Fundamental constituents of matters and interactions

All matter in the universe is made up of the fundamental SM particles called quarks and leptons with half integer spins, which are collectively called fermions. The interaction between these fermions, however, are mediated via gauge bosons with integer spins. The leptons are grouped into three families : the electron and electron neutrino (e^-, ν_e), the muon and muon neutrino (μ^-, ν_μ), and the tau and tau neutrino (τ^-, ν_τ). Similarly to leptons, quarks are also categorized in 3 families: up and down quarks (u,d), charm and strange quarks (c,s) and top and bottom quarks (t,b) as well as their antiparticles $(\bar{u}, \bar{d})$, $(\bar{c}, \bar{s})$ and $(\bar{t}, \bar{b})$. However, the quarks possess fractional charge with respect to the leptonic charge. The first families of quarks and leptons form stable matters, in which protons (uud) and neutrons (udd) are bound by the strong interaction in nuclei, whereas electrons are bound to the nucleus via the electromagnetic interaction. Fig. 2.1 shows the quarks, leptons, and bosons of the SM.
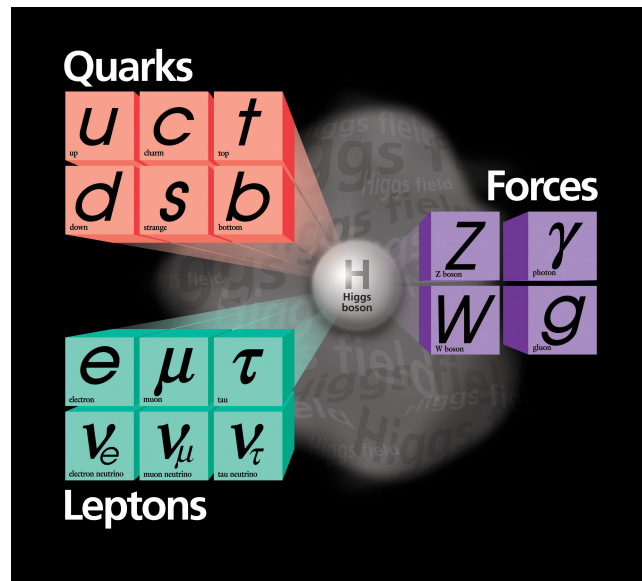
Figure 2.1: The 6 quarks (orange boxes), 6 leptons (green boxes) and 4 bosons (purple boxes) of the Standard Model of elementary particle physics. [7]

The observed physical processes in nature can be explained by four fundamental interactions of which three are incorporated in the Standard Model. The fourth interaction, the

gravitational interaction, however, has yet to be unified with the SM interactions.

The electromagnetic interaction

The electromagnetic interaction is the unified version of the fundamental electric and the magnetic interactions, in which a force is generated between charged or magnetized particles. This phenomenon includes the force between stationary particles and forces between particles with relative motion. This interaction is of infinite range and is described in the standard model by a quantum gauge theory called quantum electrodynamics (QED).

The weak interaction

The weak interaction is a short ranged ($\sim 10^{-24}$ seconds) interaction that affects all fermions and is responsible for β decay of radioactive nuclei. This interaction is caused by the exchange of heavy gauge bosons called the W and Z, the masses of which are explained in terms of spontaneous symmetry breaking by the Higgs mechanism [8, 9]. Both parity and charge parity symmetries are broken within the weak interactions and the unified interaction is referred to as "electroweak".

The Strong interaction

The strong interaction is also a short ranged interaction that affects objects that contains "color" charge (quarks and gluons). It is responsible for binding u and d quarks into nucleus and indirectly, for binding protons and neutrons in the nucleus. The strong interaction gauge group has 8 generators (gluons). Gluons carry both color and charge and hence, unlike photons, create tubes of lines of force. As the tube possesses the constant energy density, quarks need infinite energy to break apart. Due to this reason, it is not possible to obtain free quarks. This phenomenon is commonly known as color confinement [10]. The strong interaction is described in the standard model by a quantum gauge theory called "quantum chromodynamics" (QCD).

The gravitational interaction:

According to classical mechanics, the only interaction in nature that is affected by the

mass of an object is the gravitational interaction. With general relativity, the gravitational interaction is explained in terms of the space time curvature created by massive objects. There have been many attempts to prepare a successful quantum theory of gravity, and one of the hopes is string theory [11], which, unfortunately, has predictions that are outside the range of experimental verification. The gravitational interaction is long ranged and is believed to be mediated by a spin 2 boson called graviton.

2.2   Beyond the standard model

Despite the success of the discovery of the Higgs boson [12], there are a lot of compelling reasons to develop theories to overcome the deficiencies of the standard model. The standard model has failed to explain the mass hierarchy, 3 generations of quarks and leptons, the strong CP problem, and neutrino oscillations. Hence, the standard model is taken as an approximate theory in the low energy regime that unifies the weak and the electromagnetic interactions, but has not succeeded in unifying the strong interaction. In order to bring gravity into the framework of the quantum field theory and to overcome the failure of the standard model to explain various phenomena, there has been much interest in probing theories that go beyond the standard model.

Physics beyond the standard model is very popular area in the particle physics community. Much attention in this area has been focussed on supersymmetry (SUSY) models which propose that all fermionic particles and bosonic particles have superpartners whose spin differs by a half-integer. SUSY theory is believed to resolve the mass hierarchy of the SM, however, so far the LHC has failed to find evidence of SUSY. In order to explain the weakness of gravity, the theory of extra spatial dimension is also a popular beyond the standard model theory. Thus far, no evidence of such extra dimensions has been detected and lower limits have been placed on the production cross-section. Another popular BSM

approach is the idea that quarks and leptons are composite particles in which this thesis is concentrated about. A specific compositeness model called LLIM is discussed later in this Chaper. So, far there is no evidence of the compositeness and a lower limit is set on the compositeness scale in terms of the new interaction energy.

2.2.1 Contact Interaction

As a first attempt to build a weak interaction theory in 1932, Fermi imagined a 4-particle point interaction in space time using the specific example of β decay [5]. for this kind of interactions, where no propagator is involved, the coupling factor is given in terms of the Fermi limit, which is still valid in the low energy regime even after the discovery of the Z, W^+, W^-. For example, the propagation term of a massive boson is $\frac{1}{M_{W/Z}^2 - q^2}$. In the approximation $q^2 \ll M_W$, the scattering amplitudes without a propagator and with a propagator can be written as,

$$f(q^2) \propto \frac{G_F}{\sqrt{2}} \qquad (2.1)$$

$$f(q^2) \propto \frac{g_w^2}{8M_W^2} \qquad (2.2)$$

For $G_F = 1.166 \times 10^{-5}$ GeV^{-2} and $M_W = 80.4$ GeV, this implies the weak coupling factor $g_w^2 = 0.64$. Since, the Fermi's theory is a good approximation in the physics process with extremely massive bosons, if the energy regime is not accessible at the LHC, one can write an effective Lagrangian describing a new vector interaction without knowing the intermediate process. This type of interaction is described as a "contact" interaction. The Lagrangian for the current contact interaction is given by,

$$\mathcal{L} = \frac{g^2}{2\Lambda^2} [\eta_{LL} \bar{\psi}_L \gamma_\mu \psi_L \bar{\psi}_L \gamma^\mu \psi_L + \eta_{RR} \bar{\psi}_R \gamma_\mu \psi_R \bar{\psi}_R \gamma^\mu \psi_R + 2\eta_{LR} \bar{\psi}_L \gamma_\mu \psi_L \bar{\psi}_R \gamma^\mu \psi_R], \qquad (2.3)$$

where g is a coupling constant chosen to obey $\frac{g^2}{4\pi} = 1$, Λ is the contact interaction energy scale and $\psi_{L,R}$ are left and right handed fermion fields.

Since SM Drell-Yan production and the CI process are indistinguishable, assuming the same helicity state, there is a QM interference and the differential cross-section can be written as,

$$\frac{d\sigma(\Lambda)}{dM_{l^+l^-}} = \frac{d\sigma(DY)}{dM_{l^+l^-}} - \eta\frac{I}{\Lambda^2} + \eta^2\frac{C}{\Lambda^4} \tag{2.4}$$

Where, $M_{l^+l^-}$ is the dilepton invariant mass, I corresponds to the product of DY and CI amplitudes and C corresponds to a pure CI term. Destructive and constructive interference correspond to $\eta = +1$ and -1 respectively.

2.2.2   Compositeness of quarks and leptons

The variety of observed quark and lepton flavors suggest that the quarks and leptons may be composite objects of more fundamental particles (often referred to as "preons" [13, 14]). In order to account for the properties of quarks and leptons, a new strong gauge interaction is introduced. The quarks and leptons are bound states of "preons" below some characteristic energy scale Λ and behave as point like particles in that limit. However, above the energy scale Λ, they behave as extended objects. Even below the energy scale, their effect can be seen in the tail of the Drell–Yan [15] mass spectrum by observing an excess of events predicted by the SM. This theoretical framework in high energy particle collisions was described in detail in a paper by E. Eichten, K. Lane and M. Peskin [16] in 1983.

For parton interaction energy values that are much less than the compositeness scale, the metacolor force will manifest itself in the form of a flavor–diagonal contact interaction. If both quarks and leptons share common constituents, it is possible to write the Lagrangian density for contact interaction leading to di-lepton final states as given in equation 2.3. In this thesis, as shown in subsection 2.2.1, in equation 2.3, only the first term of the

Lagrangian, the left-left isoscalar term, is used as the benchmark model (LLIM) for the search for contact interactions.

2.2.3 Searches for compositeness by the CMS Collaboration

Within the CMS Collaboration, there have been a number of searches for quark and lepton substructure. Some of the searches are model dependent, while the others are not. The popular models for the model dependent searches are LLIM (Left Left Isoscalar Model model) [17] and HNC (Helicity Non Conserving Model) in various channels (inclusive jet, dijet, di-lepton, lepton and missing E_T) [18, 19]. In addition to these contact interactions models, the search for quarks substructure is being performed in terms of the search for excited quarks [1].

2.2.4 Review of previous search

Previously, the searches for quark and lepton compositeness have been performed by different experiments at different energies and in various channels. There are limits from different experiments: the Large Electron-Positron Collider(LEP [20–24]), Hadron Electron Ring Accelerator (HERA [25, 26]), the Tevatron [27–32] and A Toroidal LHC ApparatuS(ATLAS [33–36]). In addition to these experiments, there are a number of searches within the CMS experiments as discussed in 2.2.3. The best published limits based on the left left isoscalar model (LLIM [17]) in di-electron channel, which, is the subject of this thesis, is set by the ATLAS experiment [37]. The results described in this thesis represent a significant improvement over the published results.

---

[1]Quarks are fundamental particles according to SM. If quarks are observed in an excited state, that means they are composite objects.

Chapter 3

P-P collision

3.1 Proton-Proton collisions at the LHC

Over the past few decades, there have been a number of discoveries made at collider experiments. The collider experiments have served as the discovery machine of fundamental physics. The Large Hadron Collider has been exploring the physics beyond the SM and the physics at the electroweak scale, delivering a huge number of collisions initially at $\sqrt{s} = 7$ TeV and more recently at 8 TeV at subatomic distance scale, the rule of which is governed by the laws of quantum mechanics. The basic features of proton-proton colliders are following:

(1) Due to the fact that protons are heavy, the LHC can provide huge center of mass energy in head-on collisions.

(2) Since protons participate in the strong interactions, the cross-section is large ~ 100 mb.

(3) Accessibility to many new channels than are possible in lepton colliders at the corresponding energy. These channels open up resonant production for different charge and spin states as well as contributions like initial state WW, ZZ and WZ fusion.

Since, at the LHC, the colliding beams involve protons, the detailed structure of protons needs to be understood in order to accurately calculate the production cross-sections which are the key observables for collider experiments. Compositeness and the strong interactions of protons can be understood in terms of a parton model which was proposed by Richard Feynman [38] in 1969 to explain Bjorken scaling [39] in deep inelastic scattering data. A parton distribution function $f_i(x, Q^2)$ is defined as the probability density for finding a parton of flavor i (quarks or gluons) in the proton carrying a fraction x of the proton momentum with Q being the energy scale of the hard interaction. A precise knowledge of the parton distribution functions (PDF) of the proton is extremely important in order to estimate yields from models that are to be compared with LHC data.

Due to the fact that "sea" quarks arise when valance quarks emit gluons that briefly materialized into quark–antiquark pairs (off-shell quarks), the structure inside the proton is much more complex, as shown in Fig.3.1, than the simple valance quark description (uud) of the proton. Using the QCD factorization theorem, perturbative and non perturbative processes are separated and the convolution of the PDF with the hard scattering (perturbative) cross section yields the total cross section observed. For example, the Drell-Yan production crosssection at LO accuracy, is given by,

$$\sigma(AB \to l^+l^-X) = \sum_a \int dx_A dx_B f_{\frac{a}{A}}(x_A) f_{\frac{\bar{a}}{B}}(x_B) \sigma_{a\bar{a} \to l+l-}(q^2)$$

where the cross-section is summed over all values of a (13 flavors) which include a gluon (0) and quarks and anti quarks (± 1 through ± 6). x_A and x_B are momentum fraction of proton carried by the parton a.

In addition to the hard scattering between two partons, there are other final state particles (denoted by X in the above equation) usually called the underlying event. In addition to the hard interation in such a process, the accelerated colored and/or electrically charged objects in their initial or final state of interaction, emit gluons and/or photons. These processes are referred to ISR and FSR processes respectively. In this particular example of Drell-Yan, the FSR is limited to QED due to the objects with leptonic flavor in final state, however, ISR

can be QED and QCD both. With an increasing energy scale of interactions, the number of sea quarks increase resulting in the enhancement of the probability of the interaction with more than one pair of parton which is known as a "multiparton" interaction. The mulitparton interaction (MPI) can manifest themselves in various ways in high energy hadronic collisions. In particular, large hadronic activity is observed in the soft regime, characterized by small transverse momenta (P_T) of the produced particles. For relatively large P_T values, the observation of MPI will mostly focus on two simultaneous scatterings, i.e. on double parton scattering (DPS). The cross section formula for multiple parton interactions is derived by multiplying the multi-parton distributions for each individual hard scatter which demands good knowledge of double parton distribution functions (DPDFs).
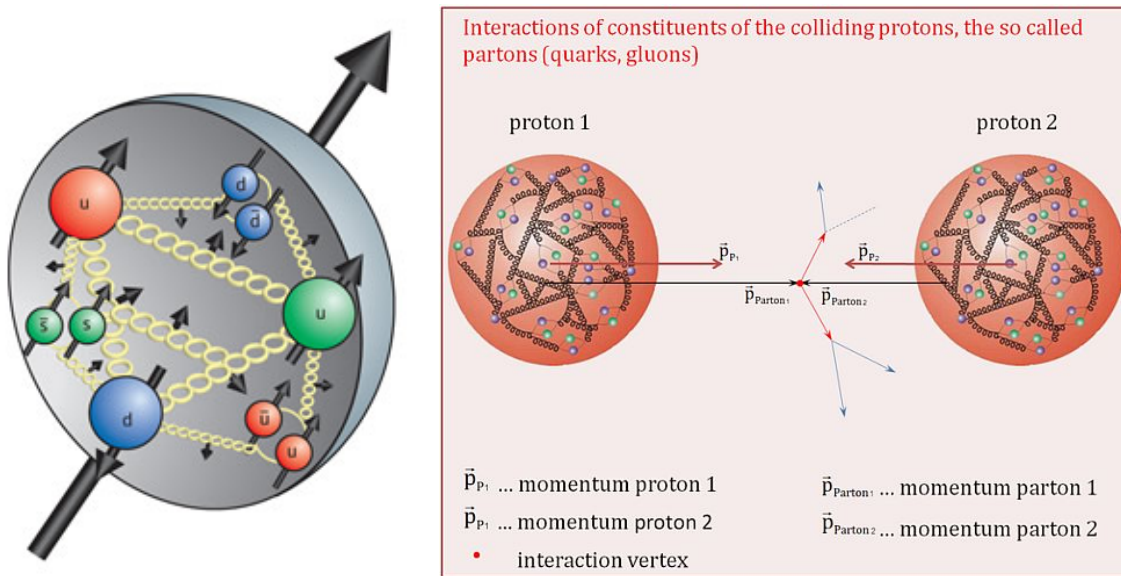
Figure 3.1:   An illustration schematic diagram of the internal structure of a proton with two valance up quarks and one valance down quark which interact via gluons (left). The figure (right) shows a couple of protons about to collide where, a "sea" of quarks and gluons in addition to the valance quarks exist.[40]

The remaining quarks which did not take part in any iteractions are not free due to quark confinement, and eventually hadronize (form stable and semistable particles) complicating the cross section measurement. These processes are approximated by parton showering with the assumption of low transverse momenta. These showers of photons emitted in ISR
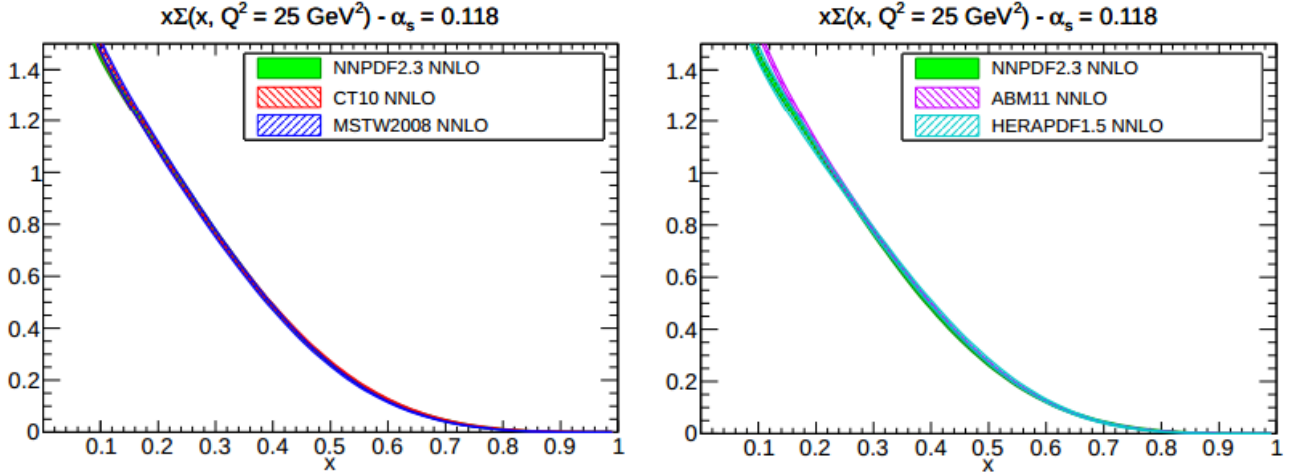
Figure 3.2: The PDFs $x\sum(x, Q^2)$ of the quark singlet at $Q^2 = 25$ GeV^2 plotted versus x on a linear scale with the comparison among NNPDF2.3, CT10 and MSTW08 (left) and NNPDF2.3, HERAPDF1.5 and ABM11 (right). All PDFs are shown for a common value of $\alpha_s = 0.118$ [41]
.

and FSR, the jets produced by these quarks and gluons are collectively called soft QCD processes, more generally "underlying events".

The knowledge of proton PDFs mainly comes from the DIS (deep inelastic scattering [42]) measured by experiments at HERA (Hadron elektron ring anlage), fixed target, tevatorn experiments as well as at the LHC data. Presently the determination of PDFs is carried out mainly by MSTW [43], CTEQ [44], NNPDF [45], HERAPDF [46], AB(K)M [47], GJR [48]. The CTEQ PDFS are obtained by a global analysis of hard scattering data in the framework of general mass perturbation QCD. The CTEQ6 [44] family of PDFs is superseded by newer, more complete, CT10 [49] PDF sets. CT10 also includes the recent HERA I data and more tevatorn data. The PDFs of the quark singlet and gluon at $Q^2 = 25$ GeV^2 plotted versus x with comparison among PDF sets NNPDF2.3, CT10, MSTW08 and NNPDF2.3, HER-APDF1.5 and ABM11, for a common value of $\alpha_s = 0.118$, are shown in Figs. 3.2, 3.3, 3.4 and 3.5. The PDF sets NNPDF, MSTW and CT10 are used to estimate systematic uncertainty due to the knowledge of PDFs in this analysis. The detailed study for the systematic uncertainty due to PDFs is given in Chapter 10.
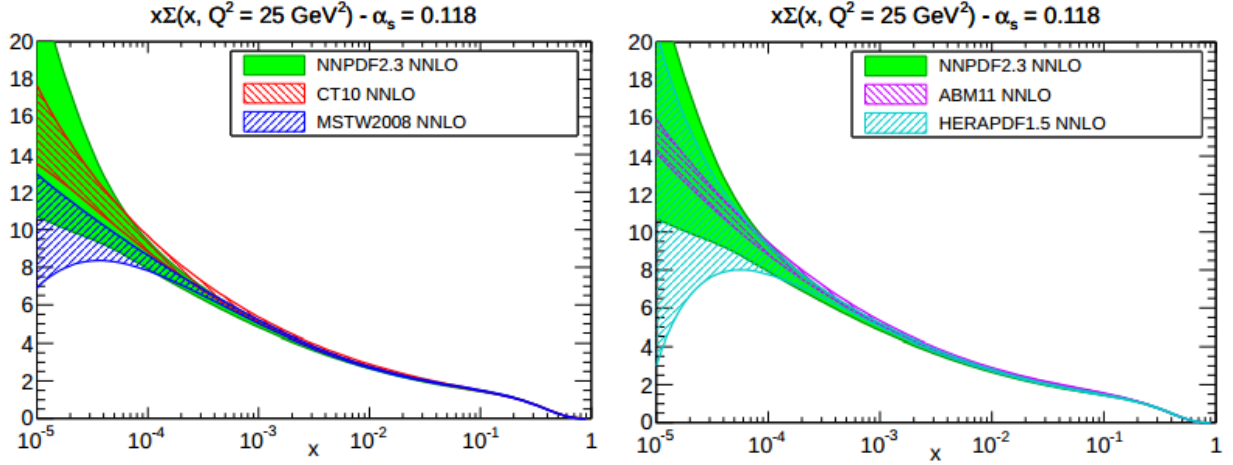
Figure 3.3: The PDFs of the quark singlet at $Q^2 = 25$ GeV^2 plotted versus x on a logarithmic scale with the comparison among NNPDF2.3, CT10 and MSTW08 (left) and NNPDF2.3, HERAPDF1.5 and ABM11 (right). All PDFs are shown for a common value of $\alpha_s = 0.118$ [41]

In CMS, several Monte Carlo event generators are used in order to simulate the backgrounds and signals using different Parton Density Functions (PDFs). Different PDF sets are produced by different groups. The PDF used in this thesis by different event generators comes from the CTEQ group. Some details of the event generators are given below:

- **PYTHIA**

  PYTHIA [50], whose name comes from Greek mythology, is a full event generator in which hard scattering matrix elements are implemented in LO and the effects of higher orders are approximated by adding parton showers to the partons (ISR and FSR) in the hard scattering process. It generates the collisions between leptons, hadrons and gammas following QCD recipes.

- **MC@NLO**

  MC@NLO [51] is a hard event generator using a matrix element at next to leading order (NLO) accuracy which can be interfaced to the shower Monte Carlo HERWIG. MC@NLO includes heavy flavor physics, which is not very common in other MCs.
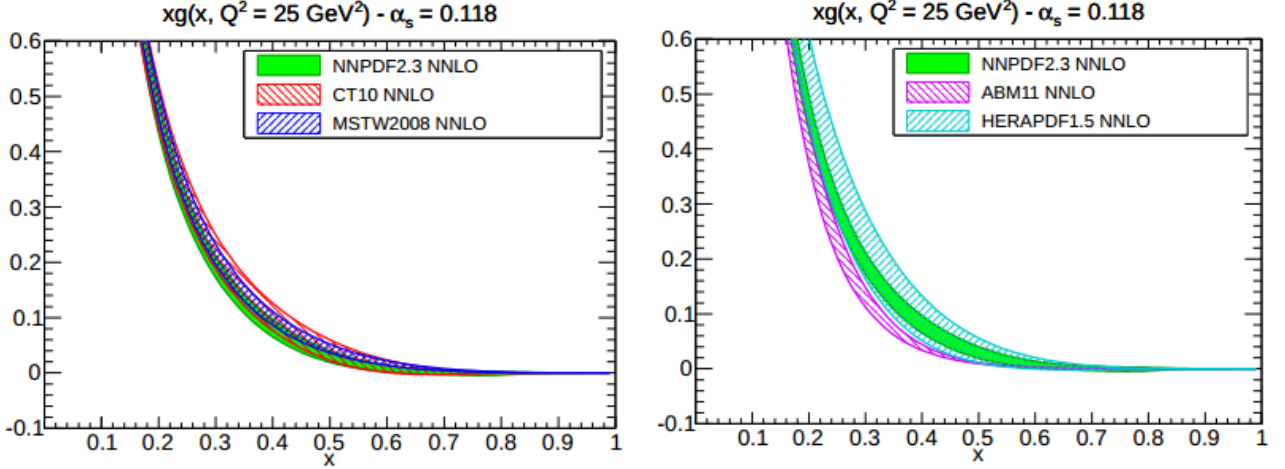
Figure 3.4: The PDFs of the gluons at $Q^2 = 25$ GeV^2 plotted versus x on a linear scale with the comparison among NNPDF2.3, CT10 and MSTW08 (left) and NNPDF2.3, HERAPDF1.5 and ABM11 (right). All PDFs are shown for a common value of $\alpha_s = 0.118$ [41]

- **POWHEG**

  POWHEG [52–55] stands for positive weight hardest emission generator. It is a hard event generator for heavy quark production in hadronic collisions at NLO accuracy which can be interfaced to shower monte carlo programs like HERWIG and PYTHIA in which a shower of LO accuracy and the hard events of NLO accuracy are maintained in the output.

- **HERWIG**

  HERWIG [56] stands for hadron emission reactions with interfering gluons. This is a full event generator in which a matrix element is implemented in LO.

- **HORACE**

  Horace [57–62] is a Monte Carlo event generator for the Drell-Yan process. It includes exact 1-loop electroweak radiative corrections matched with a QED shower. It is widely used for the electroweak NLO correction in hadron collider experiments.
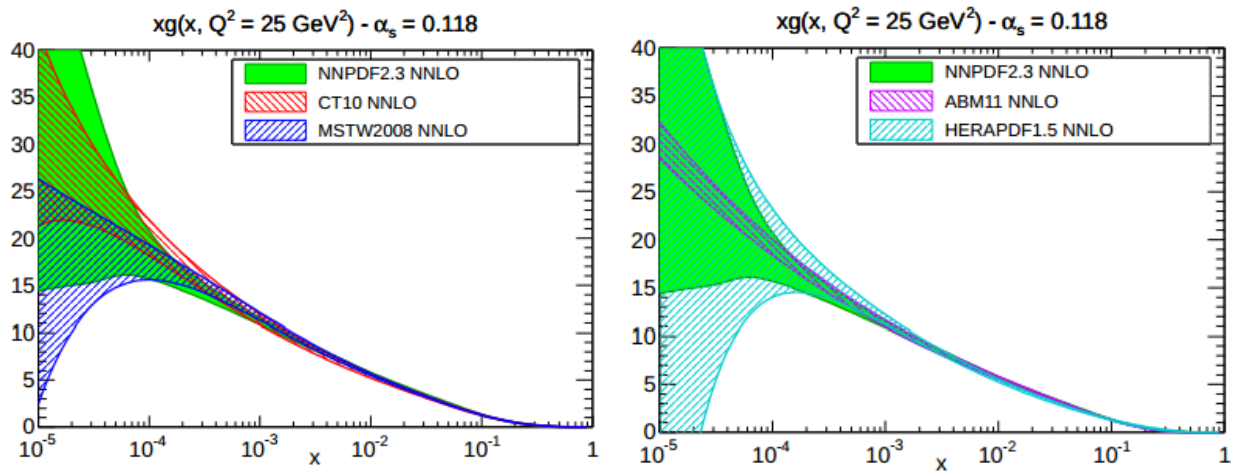
Figure 3.5: The PDFs of the gluons at $Q^2 = 25$ GeV^2 plotted versus x on a logarithmic scale with the comparison among NNPDF2.3, CT10 and MSTW08 (left) and NNPDF2.3, HERAPDF1.5 and ABM11 (right). All PDFs are shown for a common value of $\alpha_s = 0.118$ [41]

Chapter 4

The CMS experiment at the LHC

This chapter starts with a short introduction to the Large Hadron Collider (LHC). In the section 4.1.1, the design and plan of the LHC is given. A detailed description of the detectors and trigger system of the CMS experiment is given in section 4.2.

4.1 The Large Hadron Collider

As suggested by the De Broglie relation, $\lambda = h/|\vec{p}|$, to probe for increasingly smaller constituents, higher and higher energy in collider experiments is required. From 1960 to $\sim$ 2000, the energy of collider experiments had been greatly improved but not enough to study physics beyond the standard model and to discover the Higgs boson. Hence, in order to achieve these goals a new and sophisticated collider with the maximum possible energy was necessary. Regarding the colliding particles, the total energy of e^+e^- colliders can be used in experiments, which can be tuned as required. But, due to considerable energy loss by synchrotron radiation, which is inversely proportional to m^4, enough energy can't be generated, limiting the lepton colliders to precision measurements. This problem could be overcome by using hadron colliders which gave birth to the Large Hadron Collider (LHC).
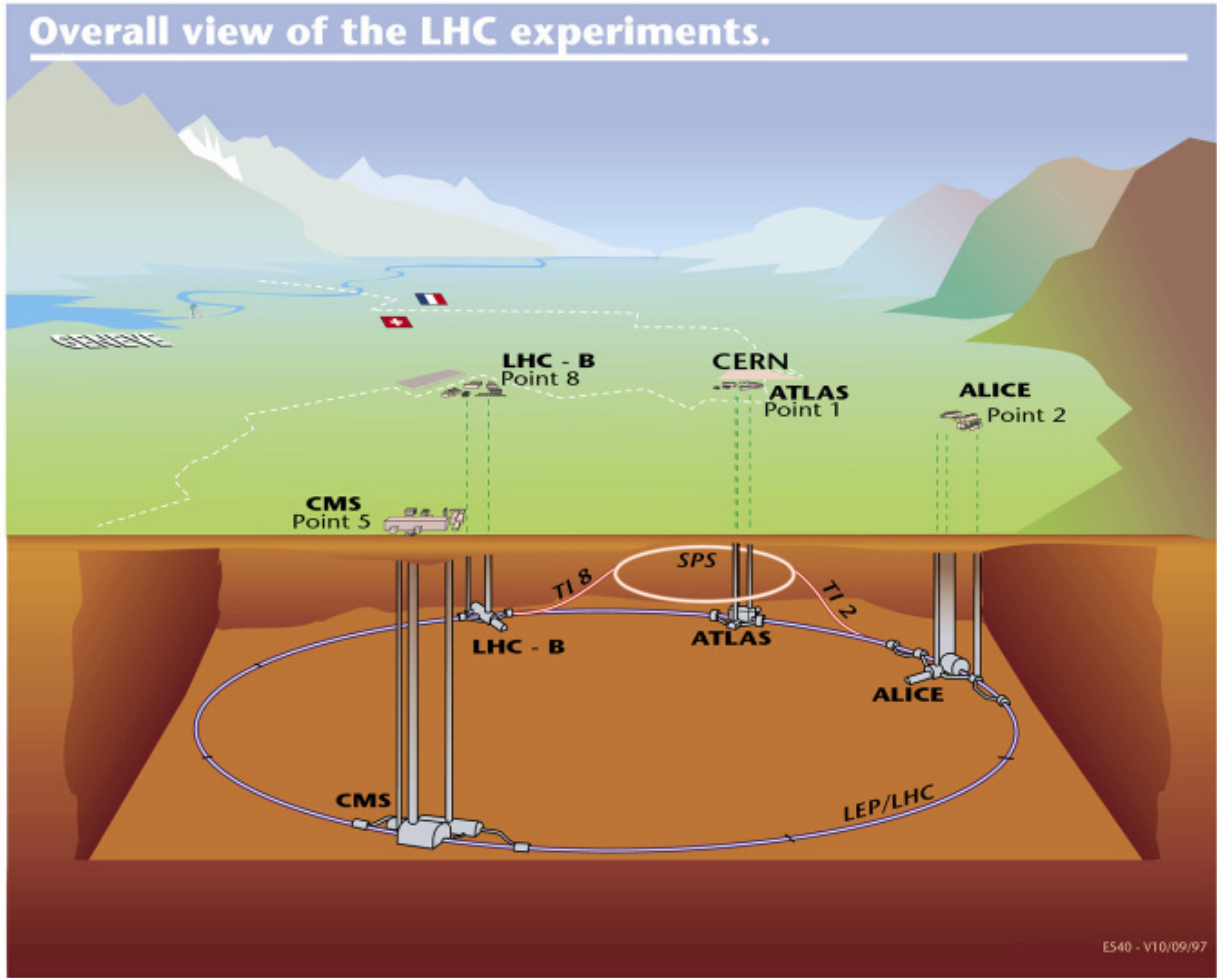
Figure 4.1:   The accelerator complex at CERN [63] [64]

The Large Hadron Collider (LHC) is the world's largest and highest-energy particle accelerator. It was built by the European Organization for Nuclear Research (CERN) from 1998 to 2008 and is located 100 m underground with a circumference of 26.7 km and spans the French-Swiss border near Geneva. The overview of the accelerator complex is shown in Figs. 4.1 and 4.2. It was designed to deliver an instantaneous luminosity of 10^{34}cm^{-2}s^{-1} at a center-of-mass energy of 14 TeV. The LHC also has a heavy ion mode, colliding lead ions with energy of 2.76 TeV at a peak instantaneous luminosity of 10^{27}cm^{-2}s^{-1}.
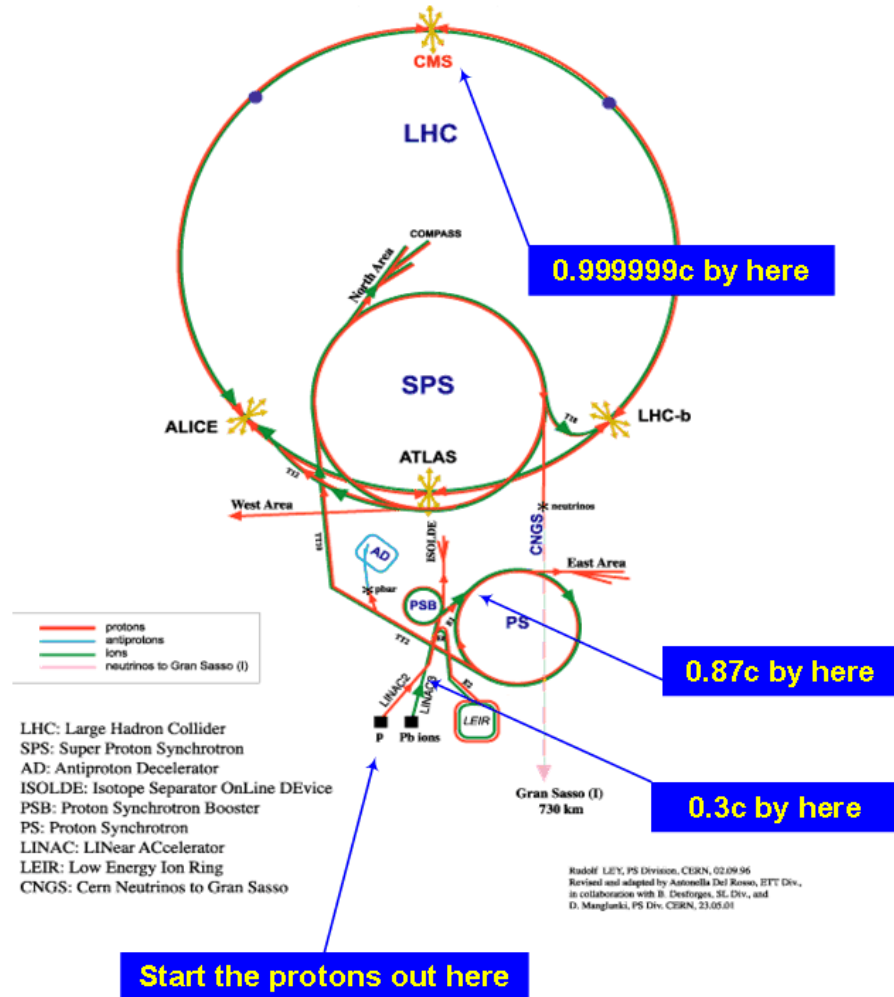
Figure 4.2:   CERN's Rings [63] [64]

4.1.1   LHC design and plans

The LHC was designed to study the standard model and BSM physics. The high luminosity of the collider allows for the possible discovery of low cross-section production with results that are not compromised from statistical fluctuations.

The LHC run starting in September 2008 was short-lived due to a faulty electrical connection causing lots of damage and delaying the operation by 14 months. However, the LHC was successful in its second attempt, in Nov. 20 of 2009 by circulating the beams and after

3 days, achieved on energy of 450 GeV per beam ejected. It caughted world's eye when the first collisions took place on March 30th between two 3.5 TeV beams, setting the world record for energy in a collider.

By Nov 2012, the LHC discovered the Higgs boson. The LHC operated with 4 TeV per beam until the end of 2012. It will resume operation in early 2015 with 6.5 TeV per beam giving the golden chance to detect the new physics lurking around.

The collision rate R in the LHC is proportional to the interaction cross-section

$$R = L\sigma_{int} \tag{4.1}$$

When two bunches, each containing n particles, collide with the frequency f , the luminosity is given by

$$L = f\frac{n^2}{4\pi\sigma_x\sigma_y} \tag{4.2}$$

where, σ_x and σ_y are the beam spread in the horizontal and vertical directions.

During the course of 2012, the LHC showed excellent performance, delivering the impressive integrated luminosity of 23.3 fb^{-1} at a center-of-mass energy of 8 TeV which is $\sim$3 times the integrated luminosity recorded in year 2011 (6.3 fb^{-1}) and way bigger than in year 2010 (44.2 pb^{-1}) at a center-of-mass energy of 7 TeV. The peak instantaneous luminosity was 7.8×10^{33}cm^{-2}s^{-1} in 2012, 3.5×10^{33}cm^{-2}s^{-1} in 2011, and 2×10^{32}cm^{-2}s^{-1} in 2010. The total delivered integrated luminosity and the peak luminosity per day in years 2010, 2011 and 2012 are shown in Figs. 4.3 and 4.4, respectively. Bunches of 10^{11} protons collide every 25 ns with average no. of interactions from $\sim$ 10 – 18. The CERN Council has planned for data-taking up to 2022 assuming good performance of the LHC as well as the detectors. According to the plan, a 16 months shutdown is being carried out this year. After the long shutdown, the LHC will resume at 13 TeV with luminosity $\sim 10^{33}$cm^{-2}s^{-1}. A second shutdown will be in 2015 for 16 months to raise the luminosity with the help

of the new Linac4. The luminosity will be above $10^{34}\mathrm{cm}^{-2}\mathrm{s}^{-1}$. A 19 months shutdown in year 2020 is planned to allow for the LHC high luminosity phase with an expected lumi of $4 \times 10^{34}\mathrm{cm}^{-2}\mathrm{s}^{-1}$. Current planning for the LHC and injector chain foresees a series of three long shutdowns, designated LS1, LS2, and LS3. In LS1 (in the period 2013 – 2014), the CM energy will be increased to 14 TeV (or slightly lower). In the period through LS2 (2018), the injector chain will be improved and upgraded to deliver very bright bunches (high intensity and low emittance) into the LHC. In LS3 (2022), the LHC itself will be upgraded with new low-β triplets and crab-cavities to optimize bunch overlap at the interaction regions.
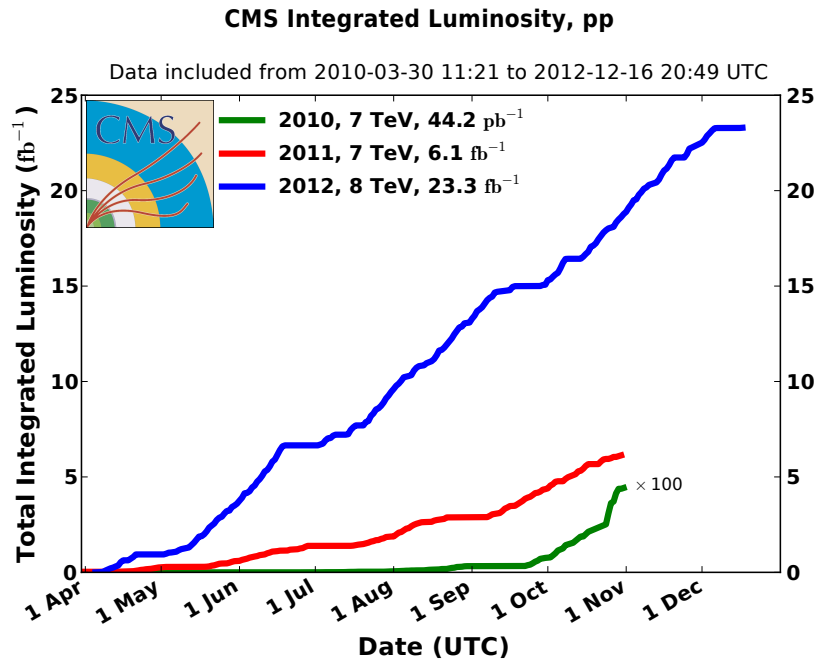
Figure 4.3: Total Integrated luminosity delivered in 2010, 2011 and 2012 in the unit of fb^{-1}

4.2 The CMS experiment

The CMS experiment is located at Point5 on the LHC ring and has been designed to cope with the huge radiation of the LHC while recording data in a resonable precison even in the
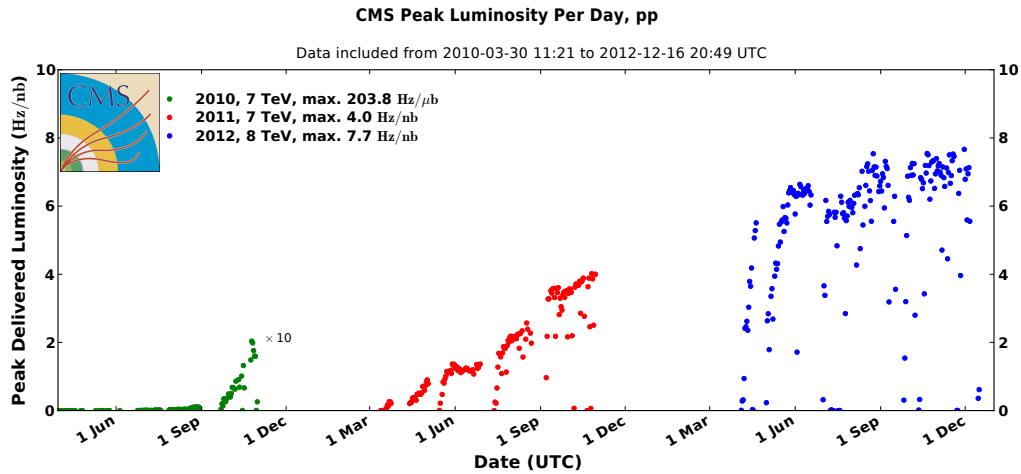
Figure 4.4: CMS peak luminosity per day for years 2010, 2011 and 2012.

huge traffic of the piled up events due to the fast collision rate (10^9s^{-1}). The CMS consists of two main parts: a cylindrical part commonly known as the barrel which is parallel to the beam pipe and two circular parts perpendicular to the beam pipe called the endcaps. These endcaps are located in both ends of the cylinder in order to maximize the detector coverage. A schematic view of the CMS detector showing the various subdetectors is shown in Fig. 4.5. The different subdetectors of the CMS detector are The tracker, ECAL, HCAL and muon system.
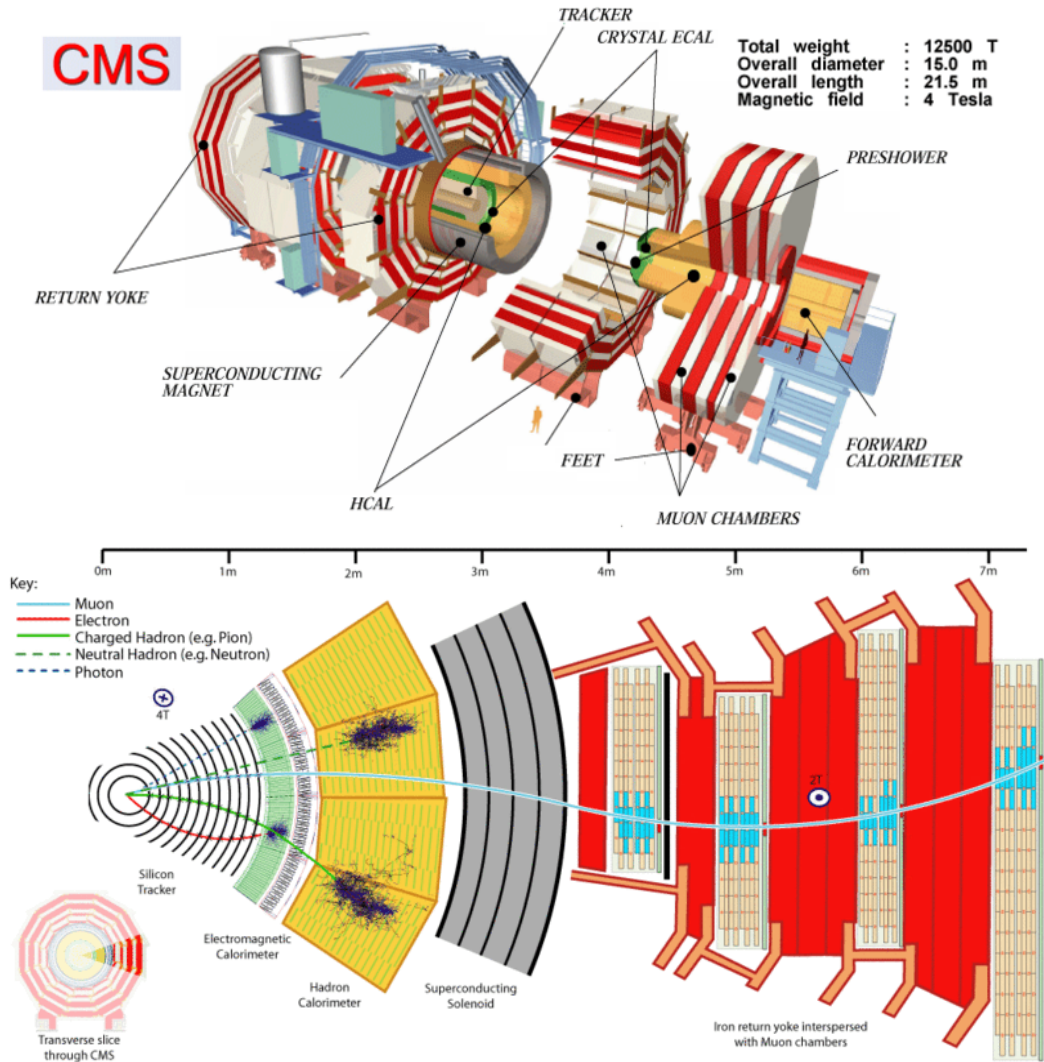
Figure 4.5: Layout of the CMS detector (above) and vertical cross-section showing the interaction of different particles in the detector(below)[65]

4.2.1 The CMS coordinate convention

The coordinate convention is very important in order to locate the hits in the detectors in order to reconstruct the tracks effectively. Since the CMS detector is cylindrical in shape with the origin at the nominal interaction point, a cylindrical coordinate system is a natural choice. A right–handed coordinate system is used in which the x-axis is pointing towards the center of the LHC ring, the z-axis is along the beam (the beam with anti–clockwise

direction, towards the west) and the y-axis is normal to the ground (the upward direction being positive y). The azimuthal angle φ is measured in the x-y plane around the beam axis from the positive x axis and the polar angle ϑ is measured from the positive z axis as shown in Fig 4.6. The $\Delta\eta$ and $\Delta\varphi$ for a pair of particles in CMS are shown in Fig. 4.7. The radius of the polar coordinate system is given by R $= \frac{z}{\cos\vartheta}$. A widely used variable is the pseduorapidity, η_i, which is related to the polar angle ϑ by following relation
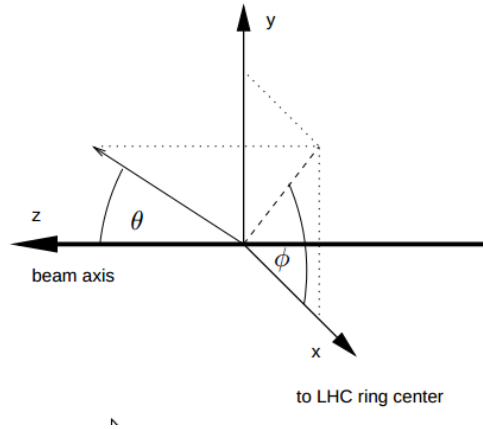
Figure 4.6:   The coordinate system of CMS
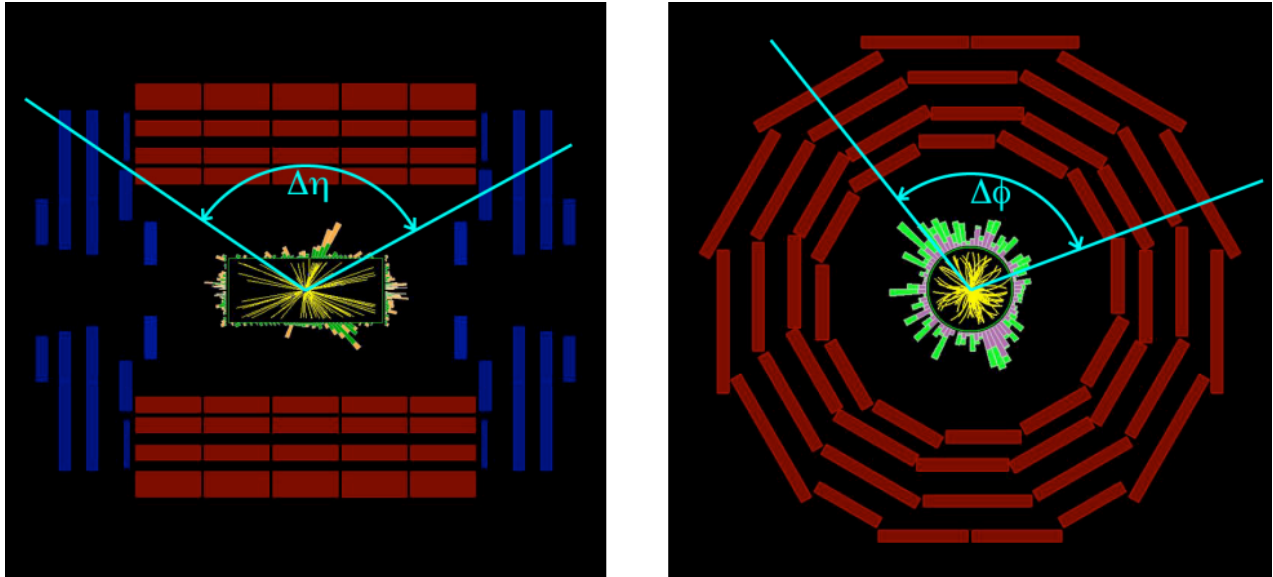
Figure 4.7:   Illustration of $\Delta\eta$ and $\Delta\varphi$ for pair of particles in CMS

$$\eta = -\ln(\tan\frac{\vartheta}{2}) \tag{4.3}$$

A useful variable is the separation in $\eta - \varphi$ space, especially used to find the total transverse momenta within a cone surrounding a track, to measure the deviation of the variables of a simulated particle and after it is reconstructed or to measure the deviation of the values of triggered particle and reconstructed particles in data. These variable dR is given by

$$dR = \sqrt{(\eta_1 - \eta_2)^2 + (\varphi_1 - \varphi_2)^2}. \tag{4.4}$$

4.3   Design

The layout of the CMS detector is shown in Figure 4.5. The 14000 ton detector, situated 100 m underground at the French village of Cessy, measures 21 m in length and 15 m in diameter. At the core of this giant detector, a 13 m long, 6 m inner diameter super conducting magnet provides a uniform magnetic field of 3.8 T. Within this solenoid, the calorimeter and tracker are located. The iron return yoke is equipped with a muon system of 4 stations. The innermost part of the detector in the vicinity of the interaction point is covered by the vertex detector composed of silicon pixel detectors surrounded by tracking detectors of silicon strip sensors. Figure 4.5 shows the transverse cross-section of the CMS detector and indicates the overall size of the inner detectors as well as the general idea of how CMS identifies a particle signature. The details of each subdetector are described next.

4.4   The tracker

The tracking system [66, 67] is designed to reconstruct the trajectories of charged particles, based upon the hits left by the particles, in order to measure their momentum and charge in the magnetic field using the track curvature, benefitting from the huge bending power

of the superconducting magnets. Since it lies to the closest to the beam axis, the tracker has to sustain a hostile radiation environment while giving excellent performance in the track reconstruction. Additionally, a fast response time is extremely important at design luminosity, given the high pileup (multiple interactions per beam crossing). The CMS tracker was designed for a luminosity of 10^{34}cm^{-2}s^{-1} and 25 ns bunch spacing. The CMS tracker can be divided in two distinct subsystems both of which are based on silicon detector technology: the silicon pixel and the silicon strip systems.
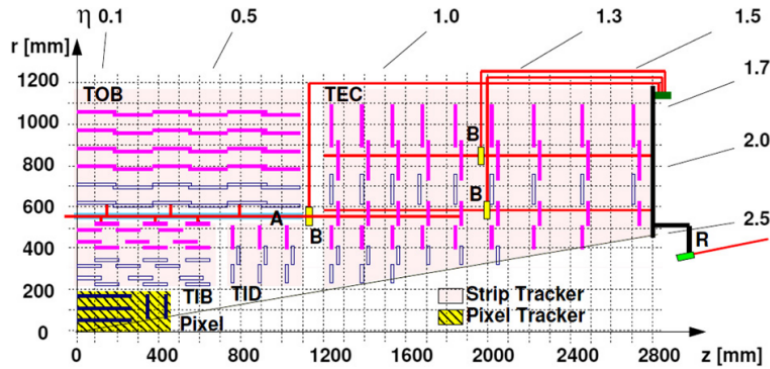
Figure 4.8: A schematic r–z view of the CMS Tracker[66]. The yellow colored region shows the Pixel region, rest being the Strip region.

4.4.1   The silicon pixel system

The pixel detector consists of 3 barrel layers ($|\eta| < 1.560$) with 2 endcap ($|\eta| > 1.560$ and $|\eta| < 2.5$) disks on each side, as shown in Fig. 4.8 The pixel detector is the closest detector to the beam axis where particle flux is $\sim 10^7$s^{-1} at r $\sim$10 cm. A layout of the CMS pixel detector is shown in Fig. 4.9. The three barrel layers are located at radii of 4.4, 7.3 and 10.2 cm and are called BPix (pixel barrel). The endcap disks (Fpix) are located at |z|=34.5 cm and 46.5 cm with radii extending from 6 to 15 cm. The most basic detection element in the tracker is called a module with size of 100×150 μm^2. All together, there are 768 pixel modules in BPix and 672 modules in FPix are used. The modules are tilted by about 20^0

in pixel disks. The pixel layers provide a two-dimensional coordinate of the location of the hit associated with a traversing charged particle.
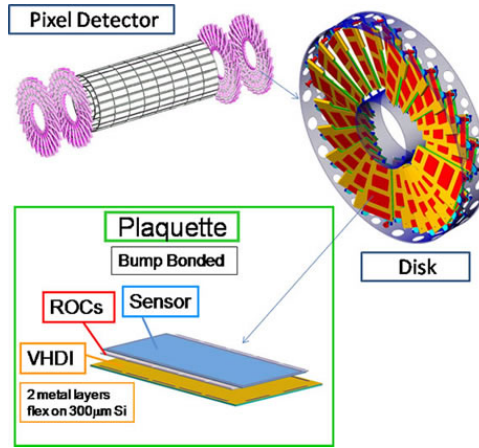
Figure 4.9: Layout of the CMS pixel detector [68]

4.4.2 The silicon strip system

The silicon system consists of a central region and endcap regions. The central region is covered by the Tracker Inner Barrel (TIB) and Tracker Outer Barrel (TOB) while end-cap regions contain the Tracker Inner Disk (TID) and the Tracker End Cap (TEC). The schematic view is shown in Fig 4.8. The TIB consists of four concentric cylinders placed in the region 250 mm < r < 500 mm at 255 mm, 339 mm, 418.5 mm and 498 mm, in the region $|z| < 70$ cm. The two inner layers have double–sided modules whereas the two outer layers have single–sided modules. The TOB is composed of six cylindrical sections that cover the region 50 cm < r <116 cm and $|z|$ <118 cm. The TID is composed of 3 disks on each side, with 3 rings of modules on each disk. The TEC is composed of 9 circular wheels covering the region of 130 cm < $|z|$ <270 cm. The innermost 3 wheels have 7 rings, wheels 4-6 have 6 rings, wheels 7-8 have 5 rings, and the last outermost wheel has 4 rings. The TIB/TID detectors are 320 µm thick silicon microstrip sensors with pitch between 80 and 141 µm. The TOB sensors are 500 µm thick silicon microstrip sensors with pitch ranging from 122 to 183

µm. The CMS strip tracker, with 9.3 million strips, ensures at least 9 particle hits in the full range of $|\eta| < 2.4$ with at least 4 of them being two dimensional measurements. These strip trackers consists of 15148 modules which are mounted on a carbon-fiber structure placed inside a temperature controlled volume.

4.5 Calorimeter

A calorimeter is used to measure the energy of particle when it passes through. Most of the particles entering the calorimeter initiate a shower of particles as shown in Fig. 4.10, which produce light that can be trapped and converted to charge measured by the electronics. Depending upon the nature of the particles, the calorimeter is designed to stop and measure their energy using special materials that are fine tuned to the properties of the interacting particles. The CMS calorimeter consists of two major subsystems which are designed for two different kinds of particles. The ECAL measures those particles which mostly loose their energy via the electromagnetic interaction.The HCAL measures hadrons which can't be stopped by the ECAL.

4.5.1 The Ecal

The CMS electromagnetic calorimeter (ECAL) [69] is composed of two parts, a barrel and endcaps, covering the range $|\eta| < 3.0$. The ECAL detects the electromagnetic showers that are initiated by electrons or photons which hit the scintillating crystal. In order to stop the shower in the smallest possible range, a high density material with very short radiation length as well as small Moliere radius[1] is preferred. Since these properties best match with lead tungstate (density= 8.28 g cm^{-3}, Molière radius = 2.2 cm, radiation length=0.89 cm), 61200 lead tungstate (PbWo4) crystals are used in barrel and 7324 crystals are used in each of the endcaps. In order to detect the light from showering particles and convert the light to an

---

[1]The radius of a cylindrical object that contains on average of 90 % energy deposition of an electromagnetic shower.

analog signal, avalanche photodiodes (APD) are used in the barrel and vacuum phototriode (VPTs) are used in the endcaps. The barrel has inner radius of 129 cm and covers the range $|\eta| < 1.48$ while the endcap covers the region $(1.56 < |\eta| < 3.0)$. The endcaps are placed at $|z| = 314$ cm. The crystals in the barrel have upstream and downstream face dimensions of 22×22 mm^2 and 26×26 mm^2, respectively; the endcap has face dimensions 28.6×28.6 mm^2 and 30×30 mm^2. The length of the crystals in the barrel are 230 mm (25.8 X_0) whereas in endcaps the lenght 220 mm (corresponding to 24.7 X_0). The crystals are arranged in modules and supermodules. A total of 36 supermodules corresponds to the 61200 crystals grouped into the barrel. However, in the endcaps, crystals are gathered in 5×5 arrays called supercrystals, totalling 14648 supercrystals in both endcaps combined. The circular shaped endcap is composed of two structures which are each a half part of the full disk. This "D" shape is commonly known as a Dee. Each Dee holds 3662 crystals. A preshower detector is placed in front of the endcap crystals and covers the range $1.653 < |\eta| < 2.6$. The preshower detector is designed to identify neutral pions that decay to two photons in a very short time.$(8.4 \times 10^{-7}$s$)$.
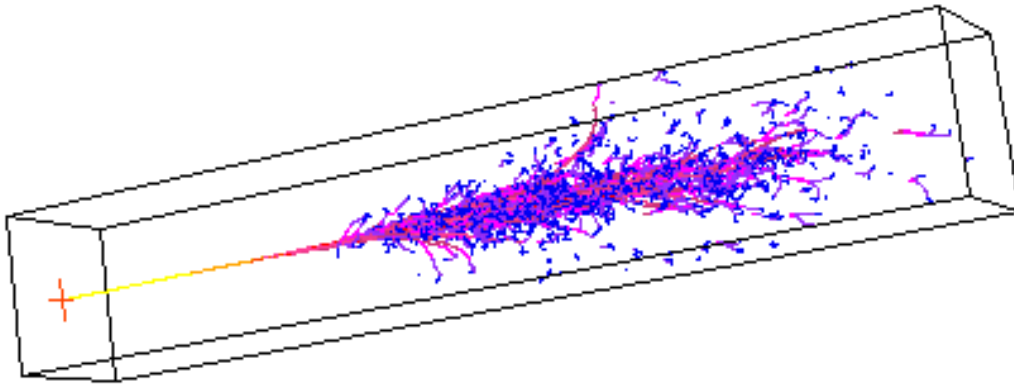
Figure 4.10: Simulation of showering of an electron in a PbWO$_4$ crystal of the CMS detector[70]

4.5.2 The Hcal

The HCAL[71] is a sampling calorimeter designed to measure jets and missing transverse energy the latter of which provides an indirect measurement of the presence of non-interacting charge neutral particles such as neutrinos. Just as the ECAL is designed for electromagnetic showers, the HCAL is designed for hadronic showers which are due to the strong interaction between hadrons and the nuclei of the material of the HCAL. The HCAL consists of layers of brass or steel interspersed with plastic scintillators which emit light at wavelengths between 410 and 425 nm. These photons enter wavelength-shifting fibers connected to hybrid photodiodes that are read out for the analog conversion. In order to detect the hadrons that may penetrate the HCAL due to its limited absorption material, HO is placed outside the solenoid magnet, which is commonly known as "Hadronic outer" or "the tail catcher". The HB covers $|\eta| < 1.4$ and has a polygonal structure with an assembly of 18 wedges that form one half-barrel. The HB is subdivided into HB+ and HB- which cover the pseudorapidity ranges $|\eta| < 1.3$ and $1.3 < |\eta| < 3.0$. The hadron forward calorimeter (HF) covers the pseudorapidity regions between 3.0 and 5.2. The HF is composted of a sandwich of steel and quartz fibers. The HF is located 11 m on either side of the interaction point and employs a technology of steel absorber and quartz fibers for readout. Cerenkov radiation is produced in the quartz fibers which allows good separation of particles in the congested forward region.

4.6 The muon system

Muons are unstable charged particles with very short lifetime (2.2 μs) and are interesting as they are expected to be produced in the decay of exotic particles that could offer a clean signature for new physics. So the CMS experiment is dedicated to detect muons, which can penetrate a few meters into materials as dense as iron and hence impossible to be stopped by the ECAL and the HCAL. So a dedicated system is required for the detection

and reconstruction of the tracks of muons. Hence, for track reconstruction, measurement of momentum and online trigger, three different systems to the very end of the detectors are built, which are collectively called the muon system [72]. There are four stations of muon chambers which are arranged in coaxial cylinders with five wheels in the barrel region called MB. In the endcap, the chambers are installed in concentric rings on each side of the detector called ME.
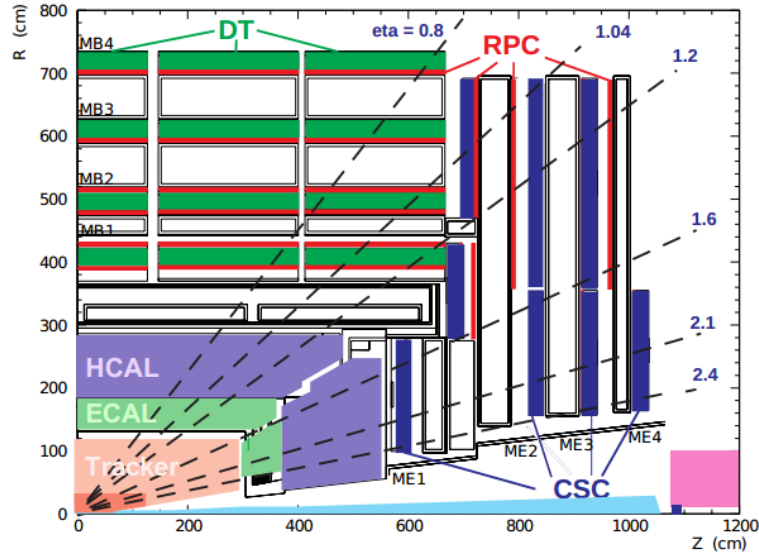
Figure 4.11: A layout of the CMS muon system[73] with the position of DT, RPC and CSC.

As shown in Fig. 4.11, the muon system consists of three different detectors called drift tubes (DT), cathode strip chambers (CSC), and resistive plate chambers (RPC). The DTs and CSCs are used in the barrel and the endcaps respectively, while the RPCs are used in both the barrel and the endcap. All together, there are 1400 muon chambers with 250 DTs, 540 CSCs and 610 RPCs.

4.6.1 Drift tube (DT)

The DT chambers contribute to the p_T measurement in the barrel region where the muon rate is low and magnetic field is relatively uniform, with coverage in the range $|\eta| \leq 1.2$.

Each drift tube is 4 cm wide with a stretched wire (working as anode) within a gas volume. It provides a one dimensional measurement of the hit position by using the drift time of the ionization charge to the wire when a charge particle knocks off an electron from the gas. The barrel DTs are organized into 4 stations. The first 3 stations, containing 8 chambers, measure the muon coordinate in the r-φ plane and the remaining 4 chambers provide the measurement in the z-direction. The last station doesn't measure the z-direction but does provide r-φ measurements.

4.6.2   Cathode strip chambers (CSC)

Four stations of cathode strip chambers (CSC) are used in the endcap disks, where the magnetic field is uneven and the particle rate is high, covering the range $0.9 < |\eta| < 2.4$. CSCs are built of anode wires and copper cathodes strips within a gas volume. When muons pass through the gas volume, the electrons are knocked off the atoms of the gas and are attracted to anode wires creating an avalanche of electrons (as they knock off other electrons in their way). The positive ions induce a charge pulse on the strip as they move towards the cathode. The orientation of strip and wires are perpendicular to each other which gives the two position coordinates for each passing muon.

The CSC is composed of 4 chambers of trapezoidal rings in each endcap with 468 chambers in total. The rings of the chambers are identified by ME±S/R, where, ME stands for Muon Endcap, S for the "station (disk)" and R for the ring number. ME±1/1 and ME±1/2 are smaller and closer to the interaction point whereas ME±2/2, ME±3/2 and ME±4/2 are further away from the interaction point and larger. The CSCs measure the azimuthal coordinates of the muon tracks with high precision, achieved by exploiting the shape of the charge distribution on three consecutive strips. This precision is sufficient to measure the muon momentum for triggering.

4.6.3   Resistive plate chambers (RPC)

RPCs are fast gaseous detectors that provide a muon trigger in parallel with those of the DTs and CSCs. The RPCs are used in both the barrel and endcap regions. All together, 6 layers of RPCs are implemented in the barrel muon system. In the endcap muon system, a plane of RPCs is embedded in each of first three stations. RPCs are made of a couple of parallel plates of high resistivity plastic material working as anode and cathode, which are separated by a gas volume. When a muon passes through the chamber, electrons are knocked off the gas which in turn hit other atoms of gas causing avalanche of electrons. The pattern of hit strips gives a quick measure of the muon momentum which is then used by the trigger to make an immediate decisions about whether the data are worth keeping. RPCs have good spatial resolution (~ 8 mm) and time resolution of just <1.3 ns.

4.7   The Trigger

Since the interaction rate is $\sim 10^9 \text{s}^{-1}$ at the nominal luminosity, the amount of data produced by the CMS is extremely high (~ 1 megabyte per crossing). It is almost impossible to write all this information to tape as it is very unlikely that every event is interesting. Hence, for a huge reduction, while keeping all the interesting events, an automated system is required, commonly known as "the trigger system". In order to make a decision, the trigger system uses a kinematic variable such as transverse energy (E_T) or transverse momentum (P_T). The function of the trigger system is not only to reduce the event rate to write, but also to separate event types. The trigger system works in two steps: Level 1-trigger and High level trigger.

Level 1-trigger

The level 1 trigger (L1) [74] is implemented in custom hardware processors which use information from the calorimeter and muon system to reduce the total event rate from 40

MHz to 100 KHz. This trigger uses the various kinematic variables to decide whether an event should be passed to the HLT, as specified in L1 trigger menu, within ~ 3 μs. Upon a positive decision, the entire detector is read out from a pipelined memory which holds the data. The data is transferred by the data acquisition system (DAQ) to the HLT.

High Level trigger

The High Level Trigger (HLT) [75] is composed of a farm of processors for further reduction of data from 100 KHz to 100 Hz, keeping the interesting events. This trigger performs a more complex calculation and has more time than the L1 trigger. The HLT trigger checks the parameters according to the HLT trigger menu and upon a positive decision the data of the event is transferred to storage.

4.8   Data processing, reconstruction and DQM shifts

After a collision, the signals in the detectors are collected by a data acquisition system consists of a huge numbers of cables and boards. These signals are sent to the detector FrontEnd Boards called FEDs which digitize and process the signals collected from the detector. After the signals are processed by multiple FEDS of each detector, the information is sent to the online processing farm. The full event information is produced by the Builder Units[2] and then sent to the local disk.

Events written by the storage manager (system which writes data to the disk buffer) are analyzed by a CMSSW[3] application as part of Data Quality Monitoring (DQM). The plots are sent to a the twikipage for DQM online shifters to judge whether each detector's performance was satisfactory. The shifter has the option to flag each subdetector as good or

---

[2]A unit of physical system interconnecting data sources with data destinations

[3]The overall collection of software of the CMS experiment is referred to as CMSSW that includes the services needed by the simulation, calibration and alignment, and reconstruction modules that process event data in order to perform analysis.

bad. The responsibility of the online shifters includes identifying problems with the detector performance as well as verifying data integrity based upon the histograms provided by the DQM infrastructure as well as the high level voltage information of subdetectors for each lumisection.

After the online shifter completes the inspection and signs off on a given run, the DQM offline shifter re-evaluates the run based upon the histograms provided by prompt-reconstruction in Tier 0 and CAF (Calibration and Alignment Facility) [76], which generally takes from a few hours to a few days. The CMS data flow summary is given in figure 4.12. After careful evaluation by experts based upon the flags and comments by the DQM shifters, a run is marked either good or bad depending upon the particular channel of interest. The detailed information for each run of given lumi-section is compiled in JSON files by a dedicated group in CMS called pdmv (Physics Data/MC validation group), which can be used by scientists around the world for data analysis.
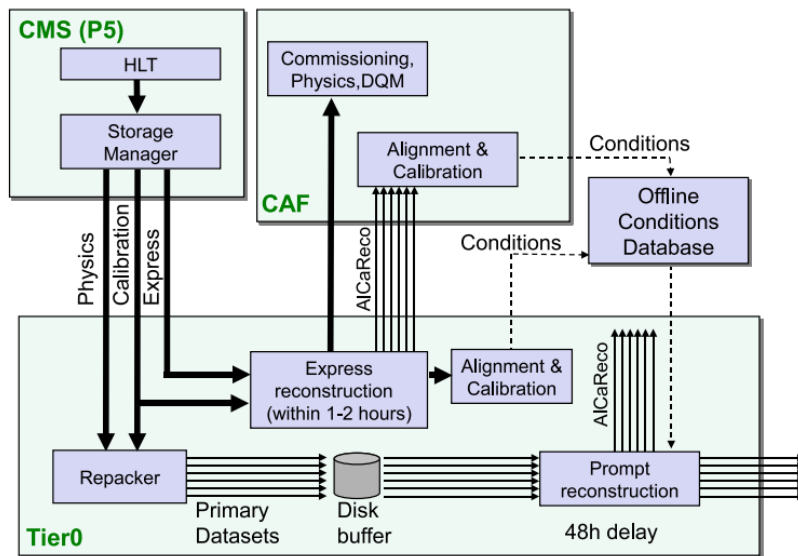
Figure 4.12: A summary of the dataflow for CMS data at CERN, starting from P5, showing the major Tier-0 tasks and PromptCalibration loops, feeding back conditions to the Tier-0

Chapter 5

The Pythia Monte Carlo

This chapter discusses a few models of compositeness implemented in the Pythia Monte Carlo event generator. In Section 5.1, a detailed explanation of contact-interaction models is presented. Since this analysis is based upon the Left Left Isoscalar Model, Section 5.1.2 gives the technical details of the script to simulate the contact interaction process in the context of the LLIM using the Pythia event generator. In Section 5.1.3, the fitting technique of the cross section for LLIM, in order to determine the cross section for a given Λ, is discussed briefly.

5.1 Compositeness models in PYTHIA

In the Pythia generator, a couple of models for the contact interaction process, HNC (Helicity non conserving model) and LLIM (Left-left isoscalar model), are implemented to introduce an anomalous coupling in addition to the Standard Model. In this analysis, the LLIM is tested against the SM under the assumption that all quarks are composite objects, hence a brief description of the PYTHIA architecture for the contact interaction process is given.

The Pythia description of the CI process with both possible interferences (constructive and destructive) with the SM Drell-Yan process, as well as pure SM DY, can be called by using subprocess ISUB 165, which, basically, can be used in three different scenerios i.e pure SM Drell-Yan process, SM DY+ CI (LLIM) and SM DY+CI (Helicity non conserving) by specifying the parameter ITCM(5), with ITCM(5)=0, ITCM(5)=1 or 2, and ITCM(5)= 3 or 4, which will call the respective processes. With ITCM(5)=0, no quarks are assumed to have any substructure, with ITCM(5)=1, only u and d have substructure and with ITCM(5)=2, all quarks have substructure. Similarly, with ITCM(5)=3, only the up quark has substructure whereas with ITCM(5)=4, u, c and t have substructure. All these processes are 2→ 2 processes. To make these processes equivalent to 2→ 1 processes like Z/γ*, MSTP(32) = 4 is used. In order to choose the interference for the cases ITCM(5) > 0, RTCM(42) is used where the value can be either 1 or -1 for destructive and constructive interference respectively with the default being destructive. In order to supply the characteristic energy scale of CI process, Λ, RTCM(41) is used and the value is supplied in GeV with the default being 1000 GeV. In order to keep the event only if the desired particles are in the final state, a card KFPR(165,1) = 11 or 13 is used with 11 for the dielectron and 13 for the dimuon channel.

5.1.1   Simulation of CI samples using Pythia

Based on the options described in Section 5.1 above, SM DY and CI(in both interference scenerio) samples were processed with different minimum mass cut and with different Λ values. For all of the samples, the production level minimum mass cut at 300 GeV, 500 GeV and 800 GeV were applied. $|\eta|$<3.0 , P_T > 30 GeV is applied using the standard Pythia filters in generetor level. For destructive interference Λ of 9, 11, 13, 15 TeV however, for constructive interference, additional samples for Λ of 17, 19 TeV were processed. A program to prepare the CMS executable config files for all of these samples, is given in D.5. The

detail of the processed samples is given in table 8.5.

5.1.2   Script for the CI/DY production

In order to simulate the CI process using PYTHIA 6.4, the following script was used which gives the techincal details for the job submission.

'MSEL = 0 !User defined process',

'MSUB(165) = 1 !CI+g*/Z $\rightarrow$ ee',

'MSTP(32) = 4 !forcing a 2 $\rightarrow$ 2 process to 2 $\rightarrow$ 1 process',

'RTCM(42) =-1 !Constructive Interference',

'RTCM(41) = 13000 !Lambda = 13 TeV',

'ITCM(5) = 2 !LL, all upper quarks composite',

'KFPR(165,1) = 11 !e$^+$e$^-$ final state',

'CKIN(1) = 1000 !Minimum $\sqrt{\hat{s}}$ value in GeV/c^2'),

The meaning of each argument is given below:

(1) MSEL = 0 : This allows the combination of different subproceses.

(2) MSUB(165) = 1 : MSUB(ISUB) specifies the process ISUB. Here, 165 insures the process in which a fermion and anti-fermion anhilate followed by the decay γ^*/Z^0 (3) MSTP(32) = 4 : Forces ISUB 165 to use the $\hat{s}$ value in place of default $P_\perp^2$ of Q^2 scale in parton distributions (forces a 2 $\rightarrow$ 2 process to 2 $\rightarrow$ 1 process)

(4) RTCM(42) = ± 1 : Sign of interferance is +1 for constructive and -1 for destructive

(5) RTCM(41) = Λ : Value of the compositeness energy scale Λ in unit of GeV with default 1000 GeV

(6) ITCM(5) = 2 : Compositeness assumed for all quarks in the initial state

(7) KFPR(165,1) = 11 : Forces the final product e^+e^-

(8) CKIN(1) = Mass$^{\text{min}}$: Minimum $\sqrt{\hat{s}}$ value in GeV/c^2 at the parton level

5.1.3   LLIM and fitting the cross-section distribution of LLIM

This analysis tests the contact interaction model LLIM against CMS data taken in 2012 of 19.6 fb^{-1}. The LLIM corresponds to the first term of equation 2.3 in which all quarks and leptons are assumed to be composite objects. While DY the process can be simulated separately, the CI process cannot be simulated by itself since there is the quantum mechanical interference effect with the DY cross-section needs to be taken into account as depicted by equation 2.4. Hence, the contact interaction process is inseparable from the DY process. In order to specify the terms in the CI process, CI/DY or "signal" will be used throughout this thesis.

Figs. 5.1 and 5.2 show the comparison of dielectron mass spectra for destructive and constructive interference above 100 GeV/c^2 at $\sqrt{s} = 8$ TeV for different Λ values with total lumi of 20 fb^{-1}. For the plots shown here, $|\eta| < 3.0$ and P$_{\text{T}}$ >30 GeV are required to mimic the detector acceptance and the trigger threshold for the transverse momentum. Since multiple samples of 200k events were generated to cover the range 100 GeV to 3000 GeV, the spectra are smooth and continuous and the black color shows the DY distribution. As explained by equation 2.4, it is clearly seen that as Λ increases the dielectron event yield decreases and as $\Lambda \rightarrow \infty$ the spectrum converges to DY. As the CI/DY spectra are relatively flat, they are affected by the reconstruction in the detector. For example, for higher Λ value, it is more probable in DY($\Lambda = \infty$) than CI/DY that an event will be reconstructed with a higher mass than it was generated due to detector resolution smearing and the nature of the spectrum shape. For this reason, it is always better to use detector-simulated predictions to compare the CI process with data.

Despite the large numbers of events in the full-simulation samples, as shown in Table 8.6, these samples are limited to the odd values of Λ. In order to predict the CI cross-section for any given value of Λ and for a given interference sign, the dielectron yield as a function of Λ is fitted using the equation 2.4. The functional form of the event yield is then used to predict the event yield for a given mass bin with a given Λ value. As we can see in Figure 5.3, the fit is very good.
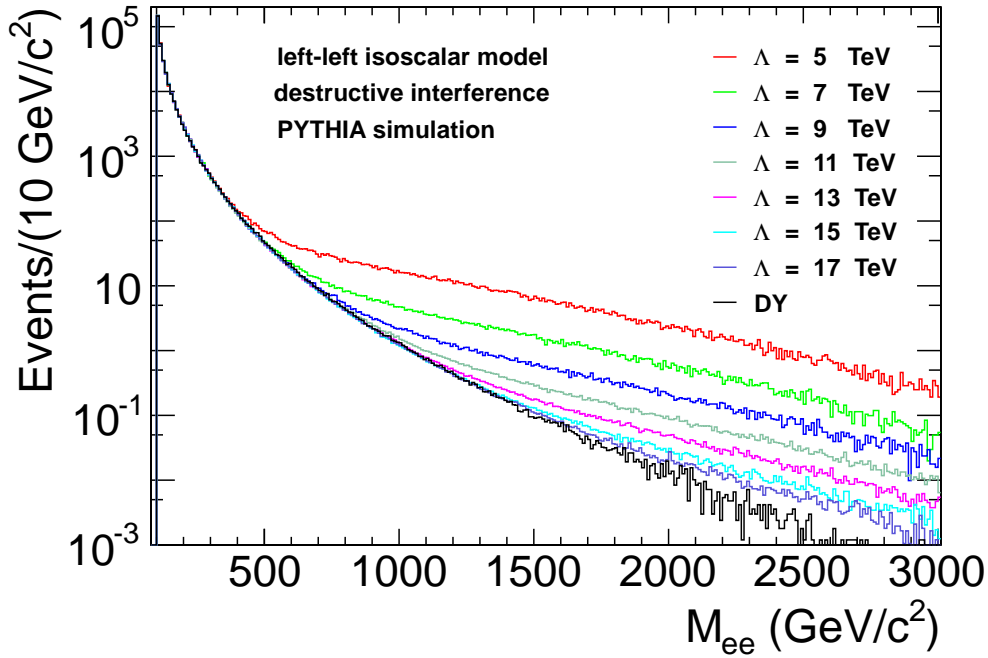
Figure 5.1: Dielectron mass spectra using the LLIM model in the PYTHIA event generator for destructive interference.
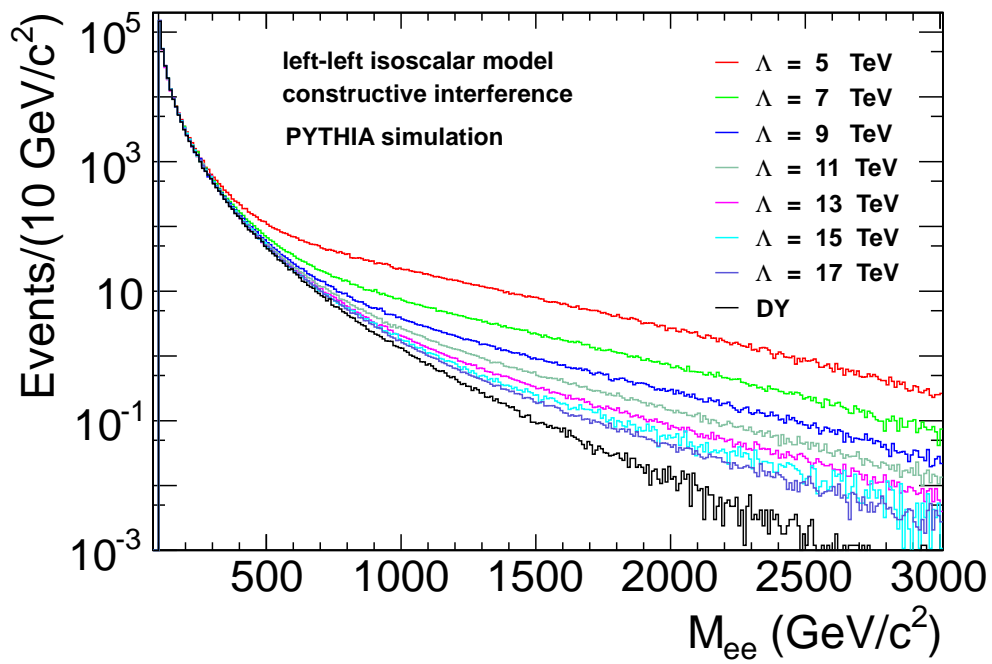
Figure 5.2: Dielectron mass spectra using the LLIM model in the PYTHIA event generator for constructive interference.
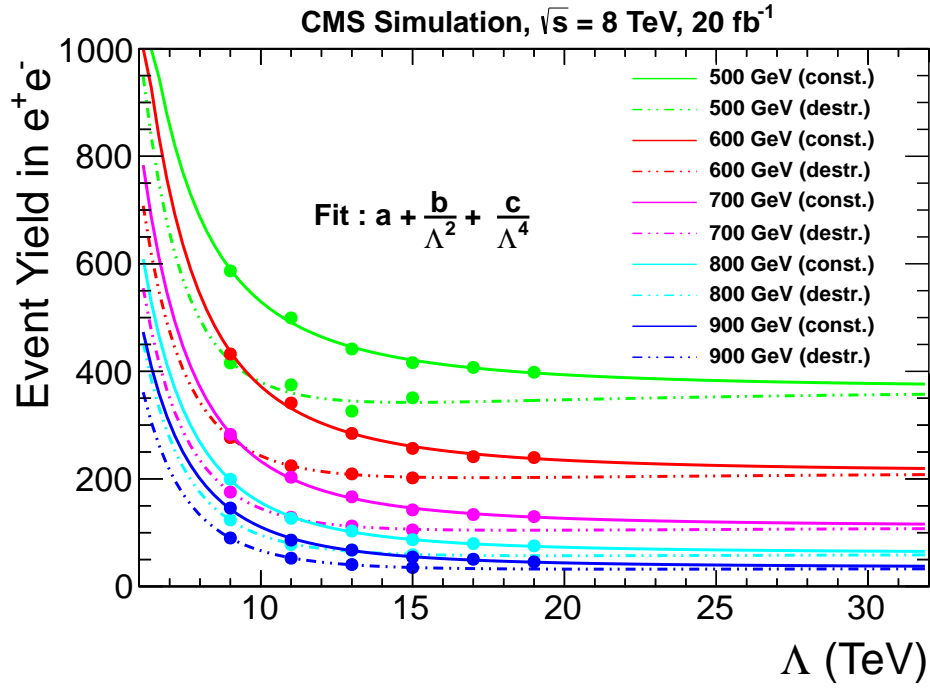
Figure 5.3: Some examples of fits to the predicted CI/DY event yields versus Λ for different values of $\mathrm{Mass}_{\mathrm{min}}$ (minimum dielectron mass). The solid line and dotted line are for constructive and destructive interferences. These curves are fitted using the equation 2.4 for the LLIM model. The colored circles represent the predictions for odd values of Λ.

Chapter 6

Overview of the analysis method

Since the main goal is to explain the 2012 CMS data in the context of the LLIM using the dielectron channel in order to confirm or rule out the compositeness model at some accessible energy range, the SM prediction is very important since it is an irreducible background to the signal. In this chapter, the strategy for the search for the CI process is discussed in brief. An overview of how the data is tested against the predicted CI model is given with a short description of how the signal and backgrounds are predicted. In section 6.3, how data is compared with the SM or CI model is discussed. The limit setting procedure, if the null hypothesis is not accepted, is also given.

6.1 The sensitive region for CI search

In any high energy physics experiment, the search for new phenomena is normally performed by different counting experiments with different observed statistics, background expectations, signal sensitivities and systematic uncertainties. In order to optimize the sensitivity of the search while separating the control region from the discovery region, it is very important to study the selection efficiency, background rate and signal probability for the given

region. In this analysis, the Z-peak region, in which no new physics is expected, provides a large statistical sample to study systematic uncertainties due to the electron reconstruction (using tag and probe), trigger inefficiency and others. The Z-peak can be used to normalize the cross-section from simulation in order to reduce systematic unvertainty. Since previous searches for contact interaction have ruled out $\Lambda \sim 9$ TeV, this analysis is focused on the region beyond 9 TeV, which is considered to be highly sensitive region of search for any CI signal at the given center of mass energy of 8 TeV. With the large statistics of the previous search for CI at $\sqrt{s}=7$ TeV up to 1.5 TeV dimuon mass, the sensitive region for this CI search is beyond 900 GeV/c^2 dilepton mass.

With the knowledge of these sensitive regions, this analysis was performed as a counting experiment with 100 GeV steps of dielectron mass, starting from 300 GeV and up to 2 TeV. This analysis uses data and Monte Carlo simulated events for signal and backgrounds in which various correction factors are applied for better accuracy. While the details of this procedure are given in chapter 8, a brief discussion of the prediction of expected events is given in section 6.2.

6.2   Prediction of observed events

As explained in chapter 2.2.1, the contact interaction signal cannot be generated separately but allowing the natural effect of the quantum mechanical interference with the Standard Model cross-section. This analysis deals with signal + background (called CI/DY + Non DY) and the background (DY + Non DY) hypothesis. Despite the dedicated selection criteria which will be discussed in chapter 7.4, there are a number of different irreducible sources that contaminate the signal. Hence, the prediction is made for all the signal + irreducible background and the irreducible background only using the following formulae

where expected events (SM) and expected events (CI) stand for the background and signal + background respectively. All of these signal and background predictions are obtained from the detector simulated samples using different Monte Carlo programs.

Expected events (SM) = CI/DY($\Lambda = \infty$) $\times$ QCD K-factor $\times$ QED K-factor + Non DY

Expected events (CI) = CI/DY(Λ) $\times$ QCD K-factor $\times$ QED K-factor + Non DY

The QCD and QED K-factors are used to bring the event yields to NLO accuracy by incorporating gluon radiation or quark/gluon loops from QCD interactions as well as photon radiation loops from electroweak interactions. The QCD and QED corrections will be explained in detail in chapter 9. With these expected events for both hypotheses, the observed events from 2012 CMS data at $\sqrt{s} = 8$ TeV can be tested for possible discovery or for ruling out a sensitive region for Λ.

6.3 Consistency check and Limit setting

After evaluation of the predicted event yield and the event yield from data, a statistical test is applied to quantify the consistency of data with either of the hypotheses in terms of a p value. A detailed explanation of this check will be given in chapter 12.1.4. If data are consistent with the SM prediction, which is actually observed in this analysis, a modified frequentist technique, commonly known as the $\mathrm{CL_s}$ method [77] with a profile-likelihood ratio as a test statistic is used to set a lower limit on Λ at 95 % CL. The details of this process are given in chapters 12.1.3 and 12.1.5.

Chapter 7

Electron reconstruction, identification and di-electron mass

This chapter starts with a quick description of electron clustering in 7.1, followed by the electron reconstruction in 7.2. This section gives a short description of selection criteria for electrons in section 7.4 and concludes with the description of the method for "Jet backgrounds" estimation using the "fake rate method" in section 7.4.1. Charge identification of electrons is discussed in 7.4.2. Finally, a short description of di-electron mass is given in 7.4.3.

7.1 Electron clustering

This analysis is based upon the comparison of the invariant mass spectrum beyond 300 GeV, which is composed of high energy di-electrons. The presence of two electrons comes mostly from either DY or CI processes along with a small contribution from the other SM processes. Since the final state involves electrons, electron clustering is described here.

Electromagnetic showers of electrons and photons deposit energy in the crystals of the ECAL and sometimes a small fraction in the material of the HCAL. In order to determine the energy of the incident particle, the total energy deposited in the crystal of the ECAL is reconstructed using the electronic signal from the array of crystals. A detailed description of the procedure for the electron identification and reconstruction can be found in [78, 79]. Different algorithms [80] are used for the energy clustering in the barrel and the endcap regions due to the different geometrical arrangement of the crystals. In the barrel region the "hybrid algorithm" is used where as in the endcap, island algorithm is used. In the hybrid super-clustering algorithm, which is designed for the high-energy electrons in the barrel, the crystals with deposited energy ($E_T > 1$ GeV) are listed, which are called the "seed" crystals. Crystals with the largest energies in the vicinity of the seed crystal are searched. As depicted in Fig. 7.1, depending upon the transverse energy of center crystal of the domino ($E_T > 1$ GeV or $E_T < 1$), 1×5 or 1×3 dominoes are formed with the central crystal of each domino aligns in η with the seed crystal. In this way, construction of dominoes proceeds with a maximum of 10 crystals in φ until the central crystal of the domino reaches the minimum limit ($E_T < 0.1$ GeV). The minimum E_T has to be higher than 0.35 GeV for a subcluster formation. In this way, all clusters are combined to make a supercluster.

Similarly, in the island algorithm, as in the hybrid algorithm, the highest deposited energy crystals are listed with minimum threshold of ($E_T > 0.18$). The list is then ordered in decreasing energy followed by looping over the seeds as shown in Fig 7.2. The algorithm collects the crystals in the φ direction, then moves in η until a rise in energy or a hole is encountered. When one η direction is completed, the algorithm goes back to the seed to search in the other η direction. To pervent double counting, a seed included in one cluster can not seed another cluster.

With the superclusters formed with the algorithms used above, the total energy of the incident particle is calculated from the sum of the energy of the individual crystals in the

supercluster. For matching the track, in the "Ecal–driven" method, the position is obtained from the weighted average of the individual crystals with weight according to each crystals contribution in energy to the supercluster.
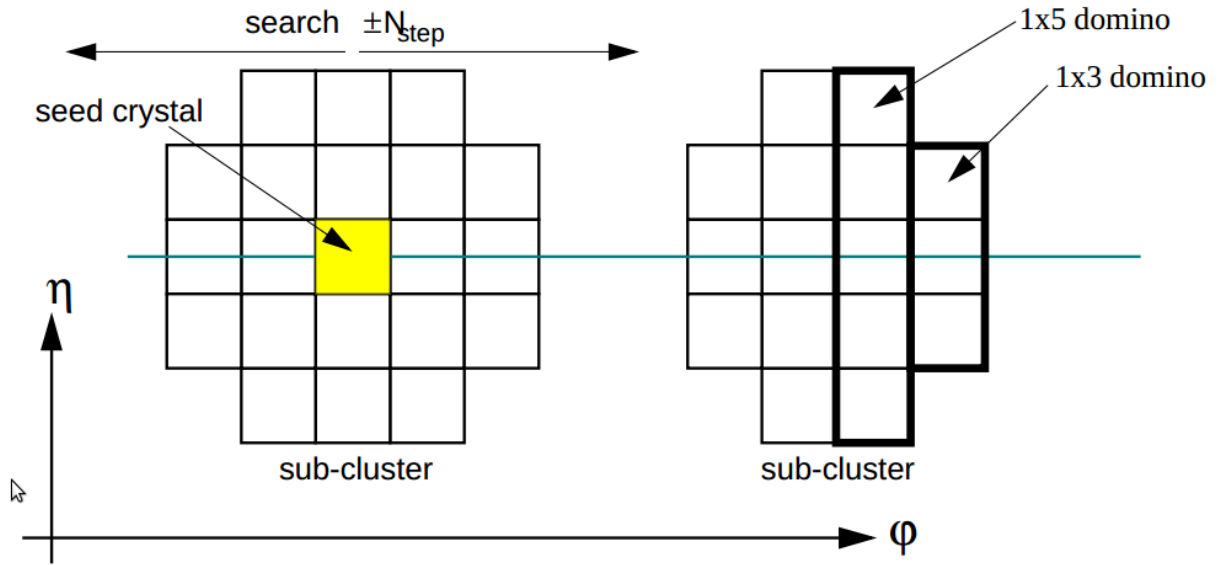
Figure 7.1:   Step of domino construction for "hybrid" algorithm

7.2   Electron reconstruction : GSF method

The electron reconstruction involves track finding in the tracker and the measurement of the position and energy deposited in the ECAL. In order to distinguish an electron from a photon, parameters of the energy in the ECAL and the hits in the tracker are important. The tracker system is based on semiconductor technology and contains significant mass from the electronics, cooling systems and mechanical support. Hence, the effect of this material must be taken into account for the reconstruction of an electron. The reconstruct takes into account Bethe-Heitler distribution of energy loss from bremstrhlung. The track is found using the gaussian-sum filter (GSF [81]) method, which is a non-linear generalization of the Kalman Filter in which weighted sums of gaussian "noise" are used instead of a single
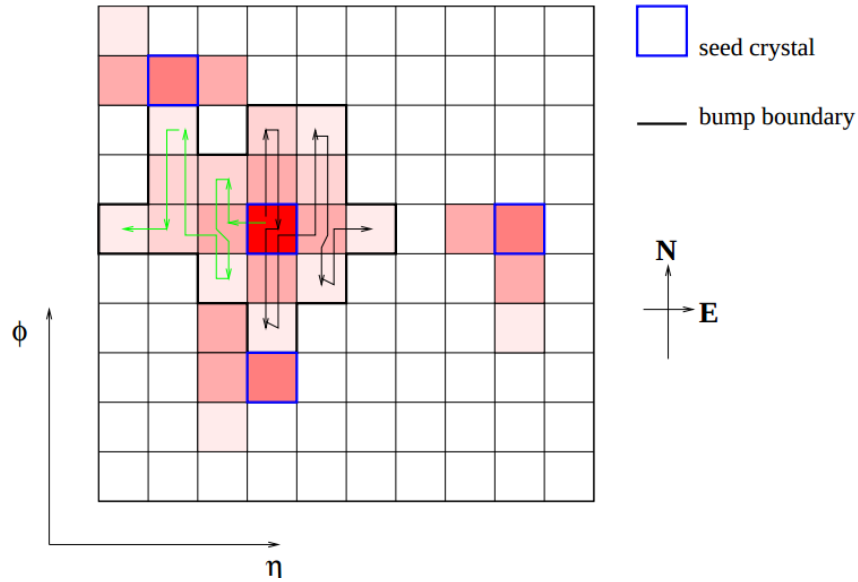
Figure 7.2: Illustration of the "island clustering" in the barrel ECAL

gaussian. In order to reconstruct an electron track, a trajectory is created from a seed. The compatible hits on the silicon layers are searched and then the track is extrapolated until no hit is observed in a couple of successive layers. Using the gaussian sum filter, the track is fitted and the χ^2 of the fitted track is evaluated with the requirement that the number of hits should be at least 5. The track with the minimum χ^2 is assigned to the electron. If energy is deposited in the ECAL with no hit in the tracker, the particle is reconstructed as photon. All the tracks reconstructed using the energy deposited in the ECAL are required to match the tracker for electrons. The track from the tracker is used to assign the charge and momentum of the electron. The energy deposited in the ECAL is used to rescale the momentum assigned using the trackers track.

7.3   Electron identification and isolation

An optimized algorithm has been developed to identify electrons in the CMS detector. The major backgrounds are pair production from a photon, jets faking electrons and hadrons

misreconstructed as electrons. A dedicated group (HEEP) in CMS has established optimized selection criteria. The selection uses the energy deposited in the ECAL and the HCAL, the sum of the transverse momenta close to the track of the electron, energy density of the event and other parameters. Using these selection criteria, commonly known as HEEP v4.1, an electron is isolated with a high efficiency and high background rejection rate.

7.4   HEEP selection criteria

Selection criteria for electrons and events, HEEP (High Energy Electron Pair) selection is followed, in which offline events are required to have two GSF electrons (those electrons tracks of which are obtained using a Gaussian sum filter) with $E_T > 35$ GeV passing the HEEP selection criteria v4.1 (see Table 7.1) with at least one of those electrons incident in the barrel electromagnetic calorimeter (EB). Brief definitions of the parameters of the table are given below.

(1) E_T, the transverse energy of the GSF electron candidate

(2) η_{sc}, the pseudo rapidity of the supercluster based on its position in the calorimeter with respect to 0,0,0.

(3) η_l, the pseudo rapidity of the electron's track measured at the inner layer of the tracker and then extrapolated to the interaction vertex.

(4) $|\Delta\eta_{in}|$, $|\Delta\varphi_{in}|$, the difference in η or φ between the track position as measured in the inner layer, extrapolated to the interaction vertex and then extrapolated to the calorimeter with the η or φ of the supercluster respectively.

(5) ρ, the average energy density in the event caused by PU (pile up [1]).

(6) d_0, minimum distance in (x, y) between track and beamspot.

(7) $\frac{H}{E}$, the ratio of the hadronic energy of the CaloTowers in a cone of radius 0.15 centered

---

[1]multiple events due to additional collisions in the same bunch crossing of the proton beams

on the electron's position in the calorimeter to the electromagnetic energy of the electron's supercluster.

(8) Track Isol, the summed P_T of the cft tracks (using Combinatorial Track Finder algorithm) tracks in a ΔR cone of 0.04–0.3 with $P_T > 0.7$ GeV and z_0 within $\pm$ 0.2 cm of the z_0 of the electron's GsfTrack and $d_0 < 9999$. The variable z_0 is minimum distance in z from the point (0,0,0).

(9) ecal isolation, the transverse EM energy of all the rec-hits with $|E| > 0.08$ GeV ($|Et| > 0.1$ GeV in the endcap) in a cone of radius 0.3 centered on the electron's position in the calorimeter excluding those in an inner cone of radius 3 crystals and eta strip of total width of 3 crystals.

(10) hadronic depth one, the transverse depth 1 hadronic energy of all the HCAL CaloTowers in a cone of radius 0.3 centered on the electron's position in the calorimeter, excluding CaloTowers in a cone of radius 0.15. Depth 1 is defined as All depths Towers 1-17, depth 1 Towers 18-29, depth 2 Towers 27-29.

(11) $\sigma_{i\eta i\eta}$, the measure of the spread in eta in units of crystals of the electron's energy in the 5x5 block centered on the seed crystal.

(12) $\frac{E^{1\times5}}{E^{5\times5}}$ and $\frac{E^{2\times5}}{E^{5\times5}}$, fraction of energy deposited in $\eta \times \varphi$ crystal matrix of size 1×5 (2×5) compared to 5×5 matrix centered on seed crystal

(13) $|dxy|$, transverse impact parameter w.r.t. the first primary vertex

(14) isEcalDriven, the electron reconsturction algorithm, which starts with reconstruction of ecal superclusters of $E_T > 4$ GeV

Reconstruction cuts for baseline selection of events in dielectron channel

(1) To avoid events from beam backgrounds, events are required to have at least 25% of the silicon tracker tracks marked as high-purity.

(2) To reject cosmic-ray muons triggering in empty bunch-crossings, at least one good primary vertex (PV) needs to be found in the offline reconstruction. The vertex must be

Table 7.1:   HEEP selection criteria version 4.1

Variable	Barrel	Endcap
E_T	$> 35\text{GeV}$	$> 35\text{GeV}$
$|\eta|$	$|\eta_{sc}| < 1.442$	$1.56 < |\eta_{sc}| < 2.5$
isEcalDriven	$= 1$	$= 1$
$|\Delta\eta_{in}|$	< 0.005	< 0.007
$|\Delta\varphi_{in}|$	< 0.06	< 0.06
$\frac{H}{E}$	< 0.05	< 0.05
$\sigma_{i\eta i\eta}$	n/a	< 0.03
$\frac{E^{2\times5}}{E^{5\times5}}$	> 0.94 or $\frac{E1\times5}{E5\times5} > 0.83$	n/a
EM + Had Depth 1 Isolation	$< 2 + 0.03 \times E_T + 0.28 \times \rho$	$< 2.5 + 0.28 \times \rho$ for $E_T < 50$
		else $< 2.5 + 0.03 \times (E_T - 50) + 0.28 \times \rho$
Track Isol: Trk Pt	< 5	< 5
Inner Layer Lost Hits	≤ 1	≤ 1
$|dxy|$	< 0.02	< 0.05

associated with four or more tracks, have $|r| < 2$ cm and $|z| < 24$ cm.

(3) Reject events associated with the anomalously high energies given by two EE 5×5 crystal regions ((ix=23, iy=23,iz=-1; ix=48, iy=98,iz=+1)

(4) Reject bad events due to unphysical values of laser calibration constant on some crystals (>3.0 in EB and>8.0 in EE) [for rereco datasets of 2012A and 2012B]

(5) No opposite charge requirement (will be discussed more in 7.4.2)

Reconstruction cuts for individual electrons

(1) The HEEP selection v4.1, shown in Table 7.1, is used for selecting individual electron candidates.

(2) Both electrons must match to HLT_DoubleEle33_CaloIdL_GsfTrkIdVL objects

For events with more than two reconstructed electrons passing all of the cuts, we select the two highest p_T electrons

7.4.1   The fake rate method

The jet background consists of following backgrounds:

(1) QCD multijets background, where two jets each fake an electron.

(2) W + jets, where the W decays into an e ν_e pair and a jet fakes an electron

(3) γ +jets, where the photon is misreconstructed as an electron and a jet fakes an electron

In general, simulation samples are avoided to estimate these backgrounds because of the poor statistics as well as the vary rare chance that a parton hadronise into almost a single particle. Since, in simulation, we are looking at a tail of the distribution where MC may not be accurate, a data-driven technique is used commonly known as "fake rate" in order to estimate the jet background.

The "fake rate" is defined as the ratio of fake leptons passing the tight criteria (HEEP 4.1) over fake leptons passing the loose criteria (Table 7.2). This ratio is determined in the Z' analysis using the SinglePhoton dataset with ratio as a function of the transverse energy of electron with three different eta ranges [82]. The jet background can be estimated by applying the measured fake rate once to a sample of events with 1 HEEP electron and 1 GsfElectron passing the fake rate pre-selection (usually known as the 1 HEEP + 1 GSF method) or twice to a sample of events with 2 GsfElectrons both passing the fake rate pre-selection (usually known as the 2 GSF method). The latter method is used as the former method overestimates the background by double counting the dijet contribution. In order to reduce contamination from $Z/\gamma^* \to$ ee the GsfElectron is required to not pass the full HEEP selection. As a result, the event is further weighted by $\frac{1}{1-\text{fakerate}}$ to correct the lost events. Unlike the 1 HEEP + 1 GSF method, using 2 GsfElectrons on data, the W+jets and γ + jets contributions are not included and hence the MC estimation using the measured fake rate of those are added to the dijet estimate from data in order to estimate the total jet background.

variable	barrel	endcap
$\sigma_{i\eta i\eta}$	< 0.013	< 0.034
$\frac{H}{E}$	< 0.15	< 0.10
no. missing hits	$<= 1$	$<= 1$
$|dxy|$	< 0.02 cm	< 0.05 cm

Table 7.2:   The loose selection requirements for the fake rate calculation

7.4.2   Electron charge misidentification

Since the energy of the electrons is assigned using the ECAL information, the energy resolution for the di-electron channel is generally better than that of the di-muon channel. However, at high E_T, electrons suffer severely from bremsstrahlung, which leads to the possibility of charge misidentification. After bremsstrahlung, the emitted photon can undergo pair production, giving oppositely charged electrons. If this process happens early enough in the tracker, one of the newly created electrons can distort the track of the original electron to the extent that the charge is flipped. Fig 7.3 shows a comparison of same vs. opposite charged leptons observed in the CMS detector. The figure shows that same-sign electrons pairs are in all likelihood Drell-Yan events where on of the charges has been misidentified. For this reason, the opposite charge requirement in the electron pair is dropped in order to retain good reconstruction efficiency. The charge misidentification probability is measured from a Monte Carlo simulation of the Drell-Yan process, which shows that probability per muon is nearly 0.45%, whereas for electron, the probability is 5.5%. In addition to bremsstrahlung, inefficiency in the tracker is also the source for the charge misidentification. In this analysis, there are 9 Drell-Yan events above $M_{ee} = 700$ GeV/c^2 with misidentified charge.
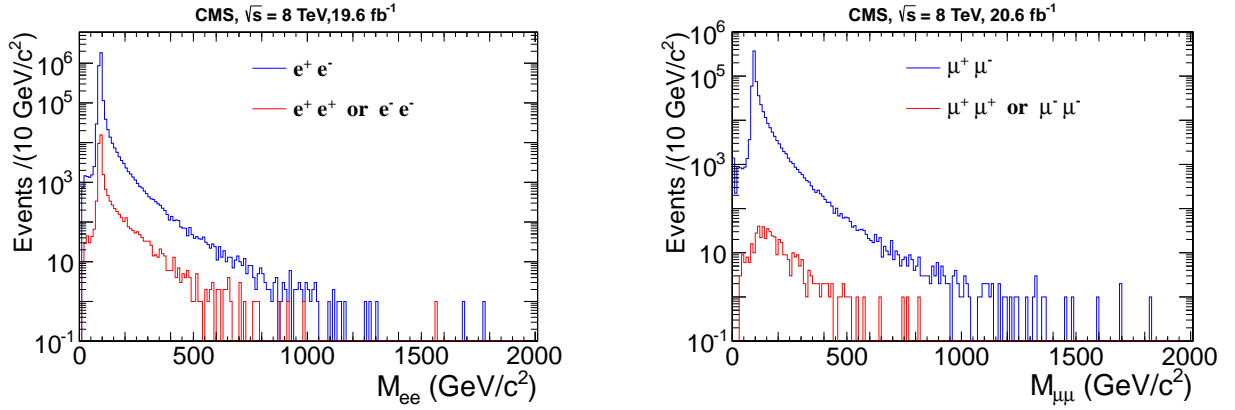
Figure 7.3: The observed events with same charge and opposite charge dielectron(left) and dimuon(right) pair in the CMS detector from the data of 2012.

7.4.3 Di-electron mass

Di-electron invariant mass is reconstructed using the laws of conservation of energy and momentum

$$M_{inv}(e_1 e_2)[GeV/c^2] = \sqrt{(E_{e_1} + E_{e_2})^2 - (\vec{P}_{e_1} + \vec{P}_{e_2})^2} \tag{7.1}$$

where $M_{inv}(e_1 e_2)$ is di-electron invariant mass, E_{e_1}, E_{e_2} are energies and $\vec{P}_{e_1}$, $\vec{P}_{e_2}$ are the momenta of the first and second electrons, respectively.

The main observable in this analysis is the number of events with M_{ee} above a threshold value. The highest invariant mass event in the dielectron sample has $M_{ee} = 1776$ GeV/c^2. Fig. 7.4 shows the reconstructed dilepton mass distributions associated with a particular generator mass window and shows the gradual degrading of mass resolution with increasing dilepton mass. The plots show that electrons have better mass resolution than that of muons in the CMS detector in the high dilepton mass region. However, in low mass region muons have better mass resolution. This is because, electrons lose energy due to bremsstrahlung and this energy is not always recovered at low E_T values. However, at high E_T the energy lost by the bremsstrahlung is measured with great accuracy in the ECAL.
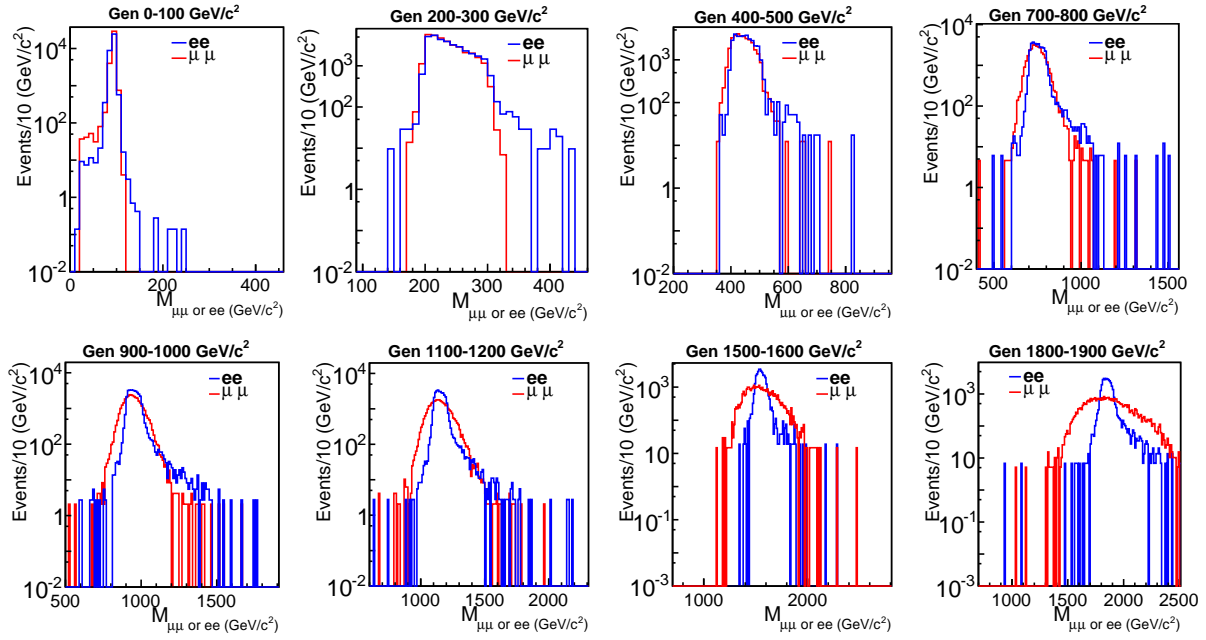
Figure 7.4: Reconstructed dimuon and dielectron distributions associated with a particular generator mass window for some of the standard bins. All the plots are normalized to the event statistics of the 0-100 GeV/c^2 bin. This figure illustrates the gradually degrading mass resolution with increasing dilepton mass.

Chapter 8

Data Sets and MonteCarlo Samples

at $\sqrt{s} = 8$ TeV

This chapter describes the data collected in 2012, the simulated samples for Contact Interaction (CI) and all other background processes. In subsection 8.1.3, the trigger requirement is discussed.

8.1 Samples

The LHC began running with collisions at $\sqrt{s} = 8$ TeV in 2012 at which time the CMS detector collected a total of 19.6 fb^{-1} of data, which is divided into various data taking periods. All of these samples are used and combined in order to compute the observed cross-section in the dielectron channel. In order to predict the expected events for the Contact Interaction model and the Standard Model, the samples were processed using PYTHIA 6.4 using GEANT4 [83] toolkit as a part of the official 8 TeV CMS production campaign in CMSSW_5_3_2_patch4. The baseline alignment scenario of V7A, using the CTEQ6L1 set

Dataset	Trigger	Run Range	Lumi(fb^{-1})
/Photon/Run2012A-13Jul2012-v1/AOD	HLT_DoubleEle33_CaloIdL_GsfTrkIdVL	190456–193621	0.81
/Photon/Run2012A-recover-06Aug2012-v1/AOD	HLT_DoubleEle33_CaloIdL_GsfTrkIdVL	190782–190949	0.08
/DoublePhotonHighPt/Run2012B-13Jul2012-v1/AOD	HLT_DoubleEle33_CaloIdL_GsfTrkIdVL	193833–196531	4.43
/DoublePhotonHighPt/Run2012C-24Aug2012-v2/AOD	HLT_DoubleEle33_CaloIdL_GsfTrkIdVL	198022–198913	0.49
/DoublePhotonHighPt/Run2012C-PromptReco-v2/AOD	HLT_DoubleEle33_CaloIdL_GsfTrkIdVL	198934–203746	6.40
/DoublePhotonHighPt/Run2012D-PromptReco-v1/AOD	HLT_DoubleEle33_CaloIdL_GsfTrkIdVL	203768–205618	7.27

Table 8.1: Datasets used in the analysis.

of parton distribution functions for the incident protons were used. The details of the data and MonteCarlo samples are given in section 8.1.1 and 8.1.2.

8.1.1   2012 Data samples

This analysis uses electron data streams from the 2012 proton-proton collision data with the run range from run 190456 to run 208686.The datasets were reconstructed with CMSSW_5_3_X. Table 8.1 gives a summary of datasets including the triggers, run ranges, and integrated luminosities.

The runs and lumi sections that have been analyzed were selected based on JSON files provided by the PdmV group:

```
Cert_190456-196531_8TeV_13Jul2012ReReco_Collisions12_JSON_v2.txt

Cert_190782-190949_8TeV_06Aug2012ReReco_Collisions12_JSON.txt

Cert_190456-196531_8TeV_13Jul2012ReReco_Collisions12_JSON_v2.txt

Cert_198022-198523_8TeV_24Aug2012ReReco_Collisions12_JSON.txt

Cert_190456-203002_8TeV_PromptReco_Collisions12_JSON_v2.txt

Cert_190456-208686_8TeV_PromptReco_Collisions12_JSON.txt
```

8.1.2   Monte Carlo samples

In order to estimate the DY contribution to the dielectron channel and perform comparisons with the CI process, the samples listed in Tables 8.2 and were used. The samples that were

used to determine the non-DY backgrounds are listed in Table 8.3.

Process	Dataset
	/DYToEE_M-20_CT10_TuneZ2star_8 TeV-powheg-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
	/DYToEE_M-120_CT10_TuneZ2star_8 TeV-powheg-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
	/DYToEE_M-200_CT10_TuneZ2star_8 TeV-powheg-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
	/DYToEE_M-400_CT10_TuneZ2star_8 TeV-powheg-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
DY$\rightarrow$ e$^+$e$^-$	/DYToEE_M-500_CT10_TuneZ2star_8 TeV-powheg-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
	/DYToEE_M-700_CT10_TuneZ2star_8 TeV-powheg-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
	/DYToEE_M-800_CT10_TuneZ2star_8 TeV-powheg-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
	/DYToEE_M-1000_CT10_TuneZ2star_8 TeV-powheg-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
	/DYToEE_M-1500_CT10_TuneZ2star_8 TeV-powheg-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
	/DYToEE_M-2000_CT10_TuneZ2star_8 TeV-powheg-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM

Table 8.2: Central MC samples used to cross check $\Lambda = \infty$ (DY) values.

Process	Dataset
DY$\rightarrow$ $\tau^+\tau^-$	/DYToTauTau_M-20_CT10_TuneZ2star_8 TeV-powheg-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
t$\bar{\text{t}}$	/TTJets_MassiveBinDECAY_TuneZ2star_8 TeV-madgraph-tauola/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
	/TT_CT10_TuneZ2star_8 TeV-powheg-tauola/Summer12_DR53X-PU_S10_START53_V7A-v2/AODSIM
tW	/T_tW-channel-DR_TuneZ2star_8 TeV-powheg-tauola/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
$\bar{\text{t}}$W	/Tbar_tW-channel-DR_TuneZ2star_8 TeV-powheg-tauola/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
WW	/WW_TuneZ2star_8 TeV_pythia6_tauola/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
WZ	/WZ_TuneZ2star_8 TeV_pythia6_tauola/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
ZZ	/ZZ_TuneZ2star_8 TeV_pythia6_tauola/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
W+jets	/WJetsToLNu_TuneZ2Star_8 TeV-madgraph-tarball/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
QCD	/QCD_Pt_20_MuEnrichedPt_15_TuneZ2star_8 TeV_pythia6/Summer12_DR53X-PU_S10_START53_V7A-v3/AODSIM
	/G_Pt-15to30_TuneZ2star_8 TeV_pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
	/G_Pt-30to50_TuneZ2star_8 TeV_pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
	/G_Pt-50to80_TuneZ2star_8 TeV_pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
	/G_Pt-80to120_TuneZ2star_8 TeV_pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
	/G_Pt-120to170_TuneZ2star_8 TeV_pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
γ + jets	/G_Pt-170to300_TuneZ2star_8 TeV_pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
	/G_Pt-300to470_TuneZ2star_8 TeV_pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
	/G_Pt-470to800_TuneZ2star_8 TeV_pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
	/G_Pt-800to1400_TuneZ2star_8 TeV_pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
	/G_Pt-1400to1800_TuneZ2star_8 TeV_pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
	/G_Pt-1800_TuneZ2star_8 TeV_pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM

Table 8.3: MC Samples used in dilepton non-DY background analysis.

Information on the cross sections for the processes in these tables can be found in table 8.4. For the CI prediction, a number of CI config files were submitted for MC sample generation. These samples are summarized in Table 8.5 and the dataset path names are given in Table 8.6.

Process	Generator	Kinematic cuts	Events	σ(pb)	Lumi(pb^{-1})	Order
DY$\to e^+e^-$	POWHEG	$\sqrt{\hat{s}} > 20$GeV	$\sim$3.3M	1915	1721.69	NLO
	POWHEG	$\sqrt{\hat{s}} > 120$GeV	$\sim$100k	12.16	8222.62	NLO
	POWHEG	$\sqrt{\hat{s}} > 200$GeV	$\sim$100k	1.517	65913.65	NLO
	POWHEG	$\sqrt{\hat{s}} > 400$GeV	$\sim$100k	0.1112	899199.64	NLO
	POWHEG	$\sqrt{\hat{s}} > 500$GeV	$\sim$100k	0.04515	2214529.35	NLO
	POWHEG	$\sqrt{\hat{s}} > 700$GeV	$\sim$100k	0.01048	9541030.53	NLO
	POWHEG	$\sqrt{\hat{s}} > 800$GeV	$\sim$100k	0.005615	17807658.06	NLO
	POWHEG	$\sqrt{\hat{s}} > 1000$GeV	$\sim$100k	0.001837	54432226.46	NLO
	POWHEG	$\sqrt{\hat{s}} > 1500$GeV	$\sim$100k	1.74E-04	573388761.47	NLO
	POWHEG	$\sqrt{\hat{s}} > 2000$GeV	$\sim$100k	2.26E-05	4426427622.84	NLO
DY$\to \tau^+\tau^-$	POWHEG	$\sqrt{\hat{s}} > 20$GeV	$\sim$3.3M	1915	1720.75	NNLO
$t\bar{t}$	POWHEG	no cuts	$\sim$21.6M	225.197	96253.37	NLO
tW	POWHEG	no cuts	$\sim$500k	11.18	44513.24	NLO
$\bar{t}$W	POWHEG	no cuts	$\sim$500k	11.18	44137.75	NLO
WW	PYTHIA	no cuts	$\sim$10M	54.8	182489.62	NLO
WZ	PYTHIA	no cuts	$\sim$10M	33.2	301213.34	NLO
ZZ	PYTHIA	no cuts	$\sim$10M	17.7	553667.12	NLO
W+jets	MADGRAPH	no cuts	$\sim$57.7M	36257.2	1591.68	NNLO
γ + jets	PYTHIA	15GeV $< \hat{p}_T < 30$GeV	$\sim$2M	200062	9.85	LO
	PYTHIA	30GeV $< \hat{p}_T < 50$GeV	$\sim$2M	19932	100.01	LO
	PYTHIA	50GeV $< \hat{p}_T < 80$GeV	$\sim$2M	3322.3	600.51	LO
	PYTHIA	80GeV $< \hat{p}_T < 120$GeV	$\sim$2M	558.3	3569.10	LO
	PYTHIA	120GeV $< \hat{p}_T < 170$GeV	$\sim$2M	108	18518.92	LO
	PYTHIA	170GeV $< \hat{p}_T < 300$GeV	$\sim$2M	30.12	66403.35	LO
	PYTHIA	300GeV $< \hat{p}_T < 470$GeV	$\sim$2M	2.139	935077.14	LO
	PYTHIA	470GeV $< \hat{p}_T < 800$GeV	$\sim$2M	0.2119	9321524.30	LO
	PYTHIA	800GeV $< \hat{p}_T < 1400$GeV	$\sim$2M	0.007078	278822266.18	LO
	PYTHIA	1400GeV $< \hat{p}_T < 1800$GeV	$\sim$2M	4.51E-05	44010864745.01	LO
	PYTHIA	$\hat{p}_T > 1800$GeV	$\sim$1.9M	1.87E-06	1036963636363.64	LO

Table 8.4: MC Samples used in dielectron background analysis.

8.1.3 Trigger requirement

All the datasets taken in the different periods of 2012 use double electron triggers, in which the transverse energy of each electron $E_T > 33$ GeV is required for the signal events. In order to mimic the condition of the detector, the same trigger is simulated in the Full simulation samples for CI/DY and other Non–DY background samples. These triggers require two online superclusters with H/E < 0.15 and $\sigma_{i\eta i\eta} < 0.014$ for the electrons in the barrel region and H/E < 0.1 and $\sigma_{i\eta i\eta} < 0.035$ for the electrons in the endcap region. The superclusters are required to have an online pixel match, an online GSF Track matching step satisfying $\Delta\eta_{in} < 0.02$ and $\Delta\varphi_{in} < 0.15$.

A detailed study, performed by the search for high mass resonances decaying into electron pairs [82] using the Tag and Probe method, showed that this trigger is more than 98% efficient over different ranges of dielectron mass which is slightly lower than over different

Table 8.5: Characteristics of simulation samples for the CI/DY process using PYTHIA. For each set of parameters (Λ, M^{min}), the number of events and cross-section σ is shown.

			M^{min} (GeV)					
			300	500	800	300	500	800
Leptons	Λ [TeV]	η	Events			σ (pb)		
	∞		55751	26056	25672	0.2621	0.03562	0.004503
	19		54189	26380	26022	0.2676	0.03821	0.005518
	17		55000	26595	25983	0.2693	0.03915	0.005877
	15		55469	27087	26689	0.2722	0.04034	0.006480
	13		56620	26748	26514	0.2761	0.04289	0.007542
ee	11	-1	55922	25322	28215	0.2852	0.04739	0.009645
	9		57300	26797	26111	0.3041	0.05713	0.014450
	15		56123	26563	26297	0.2554	0.03380	0.004403
	13		55248	33874	26239	0.2542	0.03418	0.004786
	11	+1	56049	26246	25814	0.2536	0.03516	0.005808
	9		55269	26553	26328	0.2555	0.03886	0.008843

Table 8.6:   File names of CI/DY simulation samples for the dielectron final state.

```
CIToEE_ITCM5_M-300_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
CIToEE_ITCM5_M-500_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
CIToEE_ITCM5_M-800_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM

CIToEE_Con_Lambda-9_M-300_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
CIToEE_Con_Lambda-9_M-500_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
CIToEE_Con_Lambda-9_M-800_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
CIToEE_Con_Lambda-11_M-300_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
CIToEE_Con_Lambda-11_M-500_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
CIToEE_Con_Lambda-11_M-800_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
CIToEE_Con_Lambda-13_M-300_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
CIToEE_Con_Lambda-13_M-500_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
CIToEE_Con_Lambda-13_M-800_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
CIToEE_Con_Lambda-15_M-300_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
CIToEE_Con_Lambda-15_M-500_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
CIToEE_Con_Lambda-15_M-800_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
CIToEE_Con_Lambda-17_M-300_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
CIToEE_Con_Lambda-17_M-500_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
CIToEE_Con_Lambda-17_M-800_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
CIToEE_Con_Lambda-19_M-300_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
CIToEE_Con_Lambda-19_M-500_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
CIToEE_Con_Lambda-19_M-800_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM

CIToEE_Des_Lambda-9_M-300_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
CIToEE_Des_Lambda-9_M-500_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
CIToEE_Des_Lambda-9_M-800_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
CIToEE_Des_Lambda-11_M-300_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
CIToEE_Des_Lambda-11_M-500_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
CIToEE_Des_Lambda-11_M-800_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
CIToEE_Des_Lambda-13_M-300_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
CIToEE_Des_Lambda-13_M-500_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
CIToEE_Des_Lambda-13_M-800_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
CIToEE_Des_Lambda-15_M-300_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
CIToEE_Des_Lambda-15_M-500_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
CIToEE_Des_Lambda-15_M-800_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM
```

ranges of E_T for individual electrons as shown in Figs 8.1 and 8.2 .
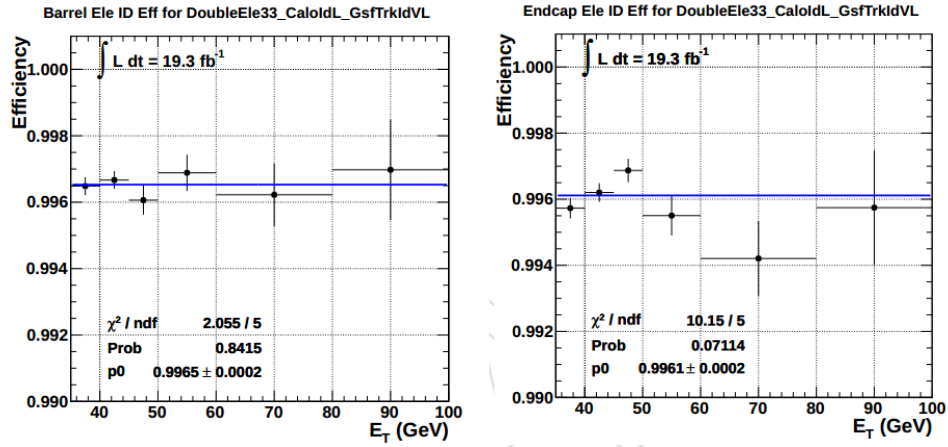
Figure 8.1: The HLT_DoubleEle33_CaloIdL_GsfTrkIdL trigger efficiency for the HEEP electrons as a function of E_T. The plots (left) is for EB or EB-EB electrons and the plots (right) is for EE or EB-EE events
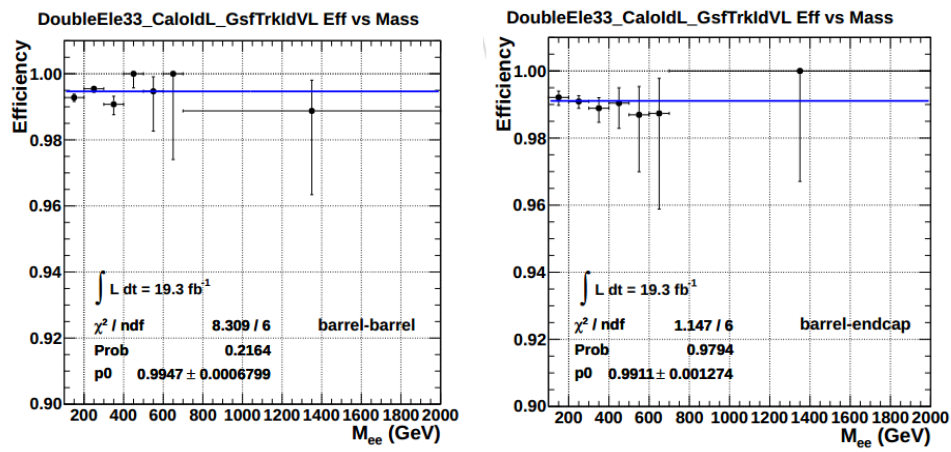
Figure 8.2: The HLT_DoubleEle33_CaloIdL_GsfTrkIdL trigger efficiency for the HEEP electros as a function of M_{ee}. The plots (left) is for EB or EB-EB electrons and the plots (right) is for EE or EB-EE events

Chapter 9

NLO Correction

This chapter gives a detailed description of the QCD and QED K-factor estimation.

9.1 K-factor

It has been observed that a mere leading order (LO) evaluation of QCD and QED cross-sections is inadequate to describe Drell-Yan, heavy quark and jet production. In turn, it can seriously underestimate the real cross-section, which could potentially undermine the possibility of observing new physics. These inadequacies can be overcome by the evaluation of the cross-sections using matrix elements with next to leading order (NLO) or NNLO accuracy. However, the theoretical model for contact interaction on which this analysis is based is implemented only in the PYTHIA event generator. In PYTHIA, the hard scattering matrix elements are implemented in LO and the effects of higher orders are approximated by adding parton showers to the partons (ISR and FSR) in the hard scattering process. In order to predict the theoretical cross-section of the signal, as well as the backgrounds that contaminate the signal, to NLO accuracy, the LO cross-section is boosted by a number calculated from the ratio of the cross-section at NLO accuracy to the LO accuracy, or sometimes from the ratio of the cross-section of data to the theory. This is generally known

as the K-FACTOR. In this analysis, we use QCD and QED k-factors for NLO accuracy of the signal and the background prediction.

$$\text{K-factor (NLO correction)} = \frac{\sigma^{\text{NLO}}}{\sigma^{\text{LO}}}$$

9.2 QCD K-factor

In this analysis, in order to estimate QCD K-FACTOR, [MC@NLO 3.4 + HERWIG 6] and PYTHIA 6 are used, which are event generators at NLO and LO accuracy, respectively. Here, HERWIG 6 is used for the showering and hadronization of the hard interaction events generated by MC@NLO. The pdf sets used are CTEQL1 (PYTHIA) and CTEQ6M (MC@NLO). The region of interest for k-factor calculation in dilepon mass scale (in GeV) is: 300 and up, 400 and up, and so forth up to 2000 and up. In order to obtain sufficient statistics, 200k events are generated for each MC for every single mass step. In essence, minimum mass is 300 GeV, 400 GeV, 500 GeV and so on, so that there are 34 total samples (17 for MC@NLO and 17 for PYTHIA). In order to calculate k-factors, the total number of dilepton events in the region of interest that are survived after the generator level cuts are considered rather than the full cross section. The generator level cuts are the following:
For both electrons:
$E_T > 35$ and $|\eta_e| < 3.0$

As explained earlier, K-factors were calculated for minimum di-electron mass (in GeV) of 300 GeV, 400 GeV and so on up to 2000 GeV. For the normalization, the luminosity of each sample was re-adjusted for the bad events (those events with final products other than dilepton events. In the case of PYTHIA, this is almost negligible: i.e. 1 to 2 events in 200k). In the case of MC@NLO, some events are generated with negative events because of the fact

that exact quantum mechanical computations feature interference phenomena, whose contributions don't have a definite sign, which prevents us having only +1 weights (if weights are > 0, +1 is assigned and if < 0, -1 weight is assigned). The negative events are necessary in order to obtain the exact NLO results for total rates and for differential distributions.

In MC@NLO, following the MC@NLO manual [84], negative weights are taken care of by weighting the related histogram of the physical observables (dielectron mass in this case) with -1 weight, which basically means that subtracting your interesting events with +1 weights by the number of events with -1 weights. In a similar fashion, to assign the luminosity to the sample, all negative events must be subtracted from the total number of events that were asked to be generated.

In the MCatNLO.inputs file for the dielectron channel, IPROC=-1351 was used for the Drell–Yan process. For a minimum mass cut, a negative number is assigned to V1GAMMAX, then V1MASSINF is assigned the desired value of mass in GeV (for minimum mass of dileption) and V1MASSSUP is assigned some ultra high values (for ∞) or any desired value to supply a mass window. In this analysis, 14000 GeV is used for V1MASSSUP and 300, 400, and so on up to 2000 were used for V1MASSINF.

In PYTHIA, MSUB(1) = 1 and MSTP(43) = 3 are used for the Drell–Yan process and CKIN(1) is used for a minimum mass of dielectrons (like above) and no upper limits for mass cut are applied. MDME(182,1) =1 is used for dielectrons in order to select appropriate final states. No filters are applied at generator level. FSR (QED, since the final products are with leptonic flavor only) is not implemented in HERWIG, our shower MC to the hard events generated by MC@NLO, for a good comparison of NLO vs LO, MSTJ(41)=1 is called in PYTHIA in order to turn off FSR deliberately for both channels.

Table 9.1 shows the K-factor results at $\sqrt{s}$=8 TeV as well as the comparison of the total cross-section ratio and the event ratio of ($\frac{\text{MC@NLO}}{\text{PYTHIA}}$) for a given $\text{Mass}_{\text{ee}}^{\text{min}}$.

Mass$_{min}$	PYTHIA$_{\sigma(pb)}$(B)	MC@NLO$_{\sigma(pb)}$(A)	$\frac{A}{B}$	K-factor
300	0.261700	0.338300	1.293	1.2968 ± 0.0046
400	0.086900	0.113000	1.300	1.3012 ± 0.0045
500	0.035560	0.046255	1.301	1.3005 ± 0.0045
600	0.016500	0.021451	1.300	1.3015 ± 0.0044
700	0.008362	0.010854	1.298	1.299 ± 0.0044
800	0.004508	0.005827	1.293	1.2933 ± 0.0044
900	0.002545	0.003279	1.288	1.2901 ± 0.0043
1000	0.001491	0.001911	1.282	1.2832 ± 0.0043
1100	0.000901	0.001148	1.275	1.2757 ± 0.0043
1200	0.000555	0.000705	1.272	1.2713 ± 0.0042
1300	0.000349	0.000442	1.266	1.2664 ± 0.0042
1400	0.000223	0.000282	1.265	1.265 ± 0.0042
1500	0.000145	0.000182	1.259	1.2589 ± 0.0042
1600	0.000094	0.000119	1.261	1.2602 ± 0.0042
1700	0.000062	0.000078	1.257	1.2558 ± 0.0042
1800	0.000042	0.000052	1.257	1.2565 ± 0.0042
1900	0.000028	0.000035	1.259	1.2587 ± 0.0042
2000	0.000019	0.000024	1.264	1.2644 ± 0.0042

Table 9.1: K-Factor and Errors at $\sqrt{s} = 8$ TeV in di-electron channel

These results are summarized in Fig. 9.1. The Fig. 9.2 shows the comparison of PYTHIA $\times$ k – factor vs Powheg, in which a good agreement is seen on both dimuon and di-electron channels. K-factors start to decrease around 400 GeV dilepton mass, and keeps decreasing until 1700 GeV. The K-factors appear to rise after 1700 GeV which could be due to the poor understanding of PDF around 2 TeV. The size of the deviation of the calculated k-factor from widely accepted flat 1.3 is applied as the systematic in order to address this unexpected behavior.

9.3 QED K-factor

Due to the fact that, in higher order QED diagrams, a virtual photon can be emitted and re-absorbed either by fermions or across the vertex, and a fermion and antifermion pair can be produced and annihilate, as a result, the matrix element is suppressed by $\frac{1}{137}$. Hence, these types of higher order corrections are also needed, along with the QCD corrections for the NLO accuracy. In order to estimate the QED K-factor, HORACE 3.1 [57–62] is used by the extra dimension search group in CMS (CMS-PAS-EXO-12-031) and QED K-factor
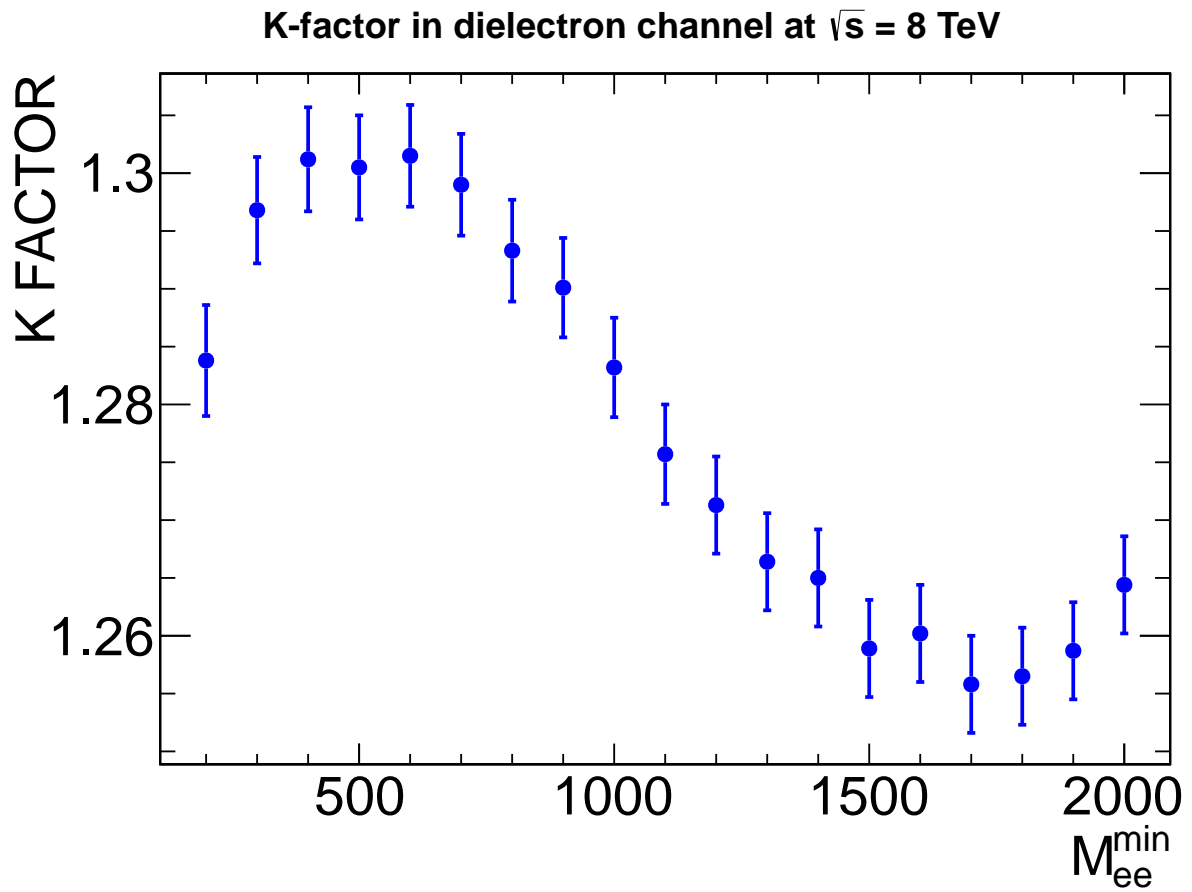
Figure 9.1: The QCD K-factors in dielectron channel at $\sqrt{s}$=8 TeV
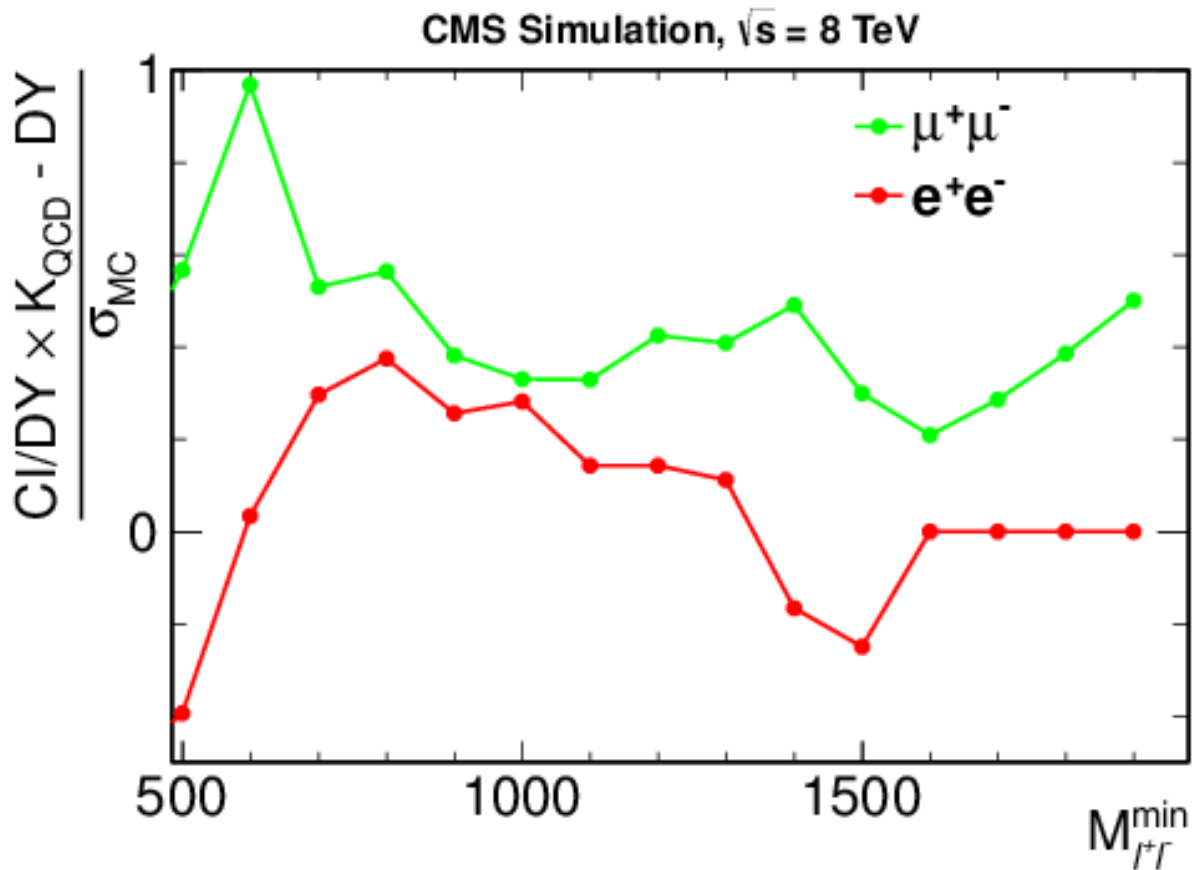
Figure 9.2: Normalized difference in predicted DY yields between PYTHIA with k-factor correction and POWHEG versus M_{min} in units of GeV. The difference is normalized by its statistical uncertainty. The integrated luminosities correspond to the data.

result for dielectron channel is presented in the CMS public document [85] which is used in this analysis.

Chapter 10

Systematic uncertainty

This chapter gives a brief description of theoretical and experimental systematic uncertainties. The PDF uncertainty is discussed in detail in section 10.1.1.

10.1 Theoretical sources

10.1.1 PDF uncertainty

The Les Houches Accord PDF (LHAPDF) [86] interface package, which is specially designed to study PDFs and uncertainties, was used to estimate the PDF uncertainty in this analysis. There are two ways to calculate PDF uncertainty: Brute Force and PDF weight technique. In the Brute force method, one must generate as many MC samples as there are PDF sets. This required a lot of computing resources, especially for the reconstruction and the detector simulation processes. However, in the PDF weight technique, only one MC is generated but weights for each PDF set are stored in an event-by-event basis followed by the observable weighing. In this analysis, the PDF weight technique is used to estimate the uncertainty of cross-section on dielectron mass spectrum due to PDFs.

Hessian Method

In order to propagate the uncertainties through the observables, experimental constraints need to be incorporated. The Hessian Method [87] both constructs a N eigenvector basis of PDFs and provides a method from which uncertainties on observables can be calculated. Hessian method, which is used for MSTW2008 and CT10, works in two steps. The first step is to fit the data using N free parameters, where the global χ^2 of the fit is minimized, which yields the best fit parameter set. In the second step, the global χ^2 is increased to form the Hessian error matrix.

$$\Delta\chi^2 = \sum_{i=1}^{N}\sum_{j=1}^{N} H_{ij}(a_i - a_i^0)(a_j - a_j^0)$$

This matrix can then be diagonalized yielding N (22 for CTEQ66 and 26 for CT10 and CT10W, 20 for MRST2008) eigenvectors. These eigenvectors probe a direction in PDF parameter space; the highest one is taken as the best eigenvalue and the lowest one is taken as the worst eigen value. Each N eigenvector direction is then varied up and down within tolerance to obtain 2N new parameter sets (hence 44 for CTEQ66, 52 for CT10 and CT10W, 40 for MSTW2008), $S_i^{\pm}(i = 1, ..,N)$.

$Mass_{min}$	$\Delta X^+(\%)$	$\Delta X^-(\%)$	Average (%)
300	6.88	4.31	5.6
400	7.33	4.32	5.8
500	7.77	4.63	6.2
600	8.12	4.98	6.6
700	8.31	5.18	6.7
800	8.36	5.25	6.8
900	8.61	5.53	7.1
1000	9.58	6.15	7.9
1100	10.29	6.57	8.4
1200	11.04	7.01	9.0
1300	11.84	7.50	9.7
1400	12.69	7.95	10.3
1500	13.74	8.58	11.2
1600	15.00	9.12	12.1
1700	16.10	9.79	12.9
1800	16.51	10.60	13.6
1900	18.97	11.53	15.2
2000	19.35	11.77	15.6

Table 10.1: PDF uncertainties from the envelope of CT10, MSTW08 and NNPDF2.1 at 68% CL.

PDF WEIGHTS

Since it is not virtually possible to generate the huge numbers of MC samples, the PDF
Weight Technique is preferred for this analysis. In this method, only one MC is generated
but weights for each PDF set are stored in an event-by-event basis followed by the observ-
able weighing. In order to do this, for each event generated using the central PDF from the
set, a PDF weight is calculated and stored in that event.

Technical details for PDF weight calculation using the Hessian method from LHAPDF pack-
age is given in A.2 and in A.3. The weights are calculated for the maximum and minimum
fluctuation using the Modified Tolerence Method (Master's equation). The technical detail
of this calculation is given in B.1. In order to estimate the PDF uncertainty for a given
set, the relative weights are used but to estimate the PDF uncertainty with different PDFs,
the absolute error sets are used so that the relative uncertainty from a common set can be
estimated.

MASTER'S EQUATION

Even if the variations applied in the eigenvector directions are symmetrical by construc-
tion, when propagated through to an observable, this may not always be true. The 2N+1
(1 being central) members of the PDF set provide 2N+1 results for any observable of in-
terest. In general, the well constrained directions (low eigenvector numbers) tend to have
symmetrical positive and negative deviations on either side of the central value of the ob-
servable. These sets of eigenvectors can be used in a Master's formula to approximate the
PDF uncertainty. There are several Master's formulae, however many of them have some
faults.

$$\Delta X_1 = \tfrac{1}{2}\sqrt{\sum_{i=1}^{N}(X_i^+ - X_i^-)^2}$$
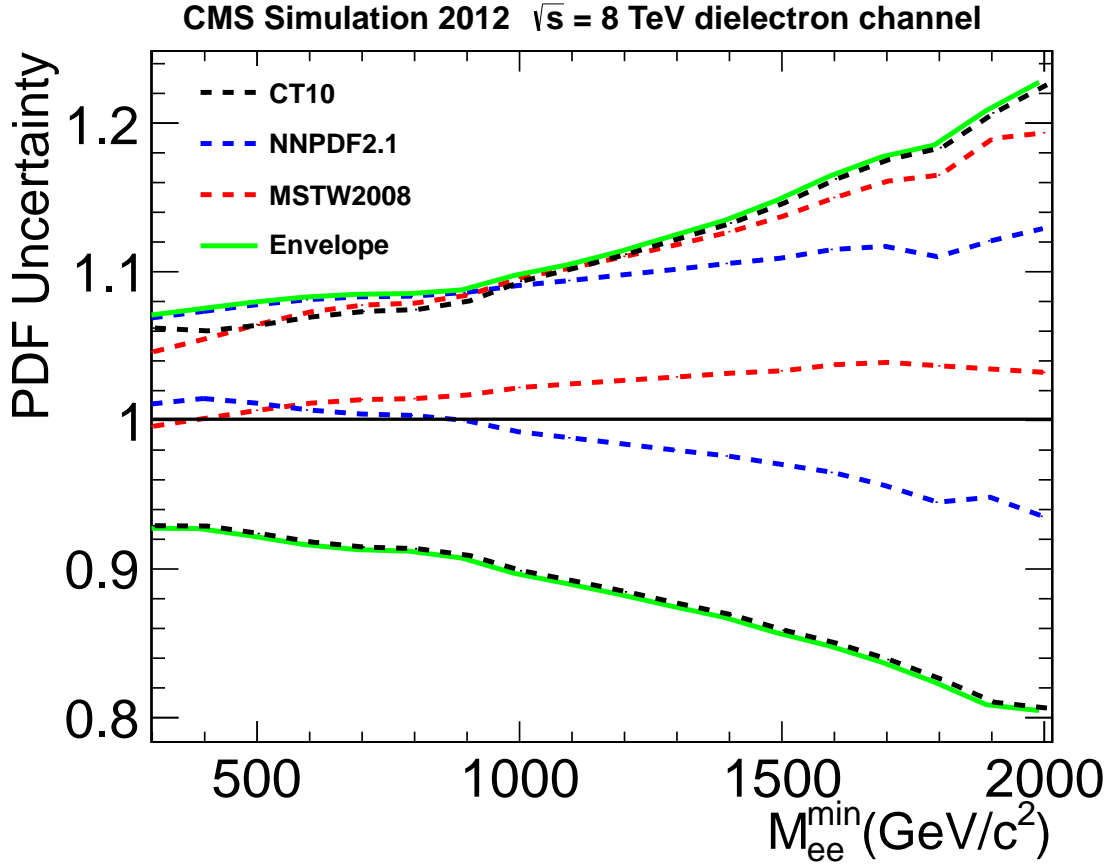
Figure 10.1: PDF uncertainty (shown in 68% CL) from CT10, MSTW08 and NNPDF21. The uncertainties for MSTW08 and for NNPDF21 are shown with respect to the central value of CT10. The envelope shows the worst fluctuation in the three different sets.

This original CTEQ Master's equation only works if X_i^- and X_i^+ are symmetrical. Otherwise, it underestimates the uncertainty.

$$\Delta X^+ = \tfrac{1}{2}\sqrt{\sum_{i=1}^{N}(X_i^+ - X_i^0)^2}$$

$$\Delta X^- = \tfrac{1}{2}\sqrt{\sum_{i=1}^{N}(X_i^- - X_i^0)^2}$$

Since the positive and negative directions defined in the PDF eigenvector space are not

always related to positive and negative variations on an observable, these equations cannot be interpreted as positive and negative errors in the general case.

The Master equation which is called "Modified Tolerence Method" is used in this analysis, as it is considered to give the best performance even in the case where observables are not symmetrical around the central values.

$$\Delta_{\max}^+ = \sqrt{\sum_{i=1}^{N} [\max(X_i^+ - X_0, X_i^- - X_0, 0)]^2}$$

$$\Delta_{\max}^- = \sqrt{\sum_{i=1}^{N} [\max(X_0 - X_i^+, X_0 - X_i^-, 0)]^2}$$

Where $\Delta_{\max}^+$ and $\Delta_{\max}^-$ are positive and negative maximum fluctuations of the observables. X_i^+ is the ith weight above the central and X_i^- is the ith weight below the central (in the case where fluctuation of the observable is symmetric, however it doesn't always happen). The Master's equation, normally, returns the maximum and minimum fluctuation for symmetric case, but for a case in which the fluctuation is not symmetrical, it simply returns 0 for that set. However, one needs to keep getting the fluctuation for all PDF sets and add them in Quadrature. Finally, the total deviations from the unity in either directions are the final weights of that event. These weights are used to weigh the histogram for dimuon or di-electron events. Finally, for a particular region of interest, all dilepon events are counted i.e. with UP weighted, DOWN weighted and CENTRAL weighted (it doesn't matter if weighed or not, as weight =1). The % difference of the UP events from CENTRAL events gives the relative uncertainty in an upward direction (maximum positive fluctuation). Similarly, % difference of the DOWN events from CENTRAL events gives the relative uncertainty in a downward direction (maximum negative fluctuation).

The same calculation can be repeated using the unnormalized values of the weights (central value will not be 1 in this case) which won't effect the result to estimate the uncertainty for a given set. However, if the uncertainties are estimated from multiple sets with the result

relative to each other, unnormalized weights are used in which the values are compared from the central value of one fixed set. Following the PDF4LHC recommendation, the absolute uncertainty from 3 different PDF sets, with respect to the central values of each sets, are taken and the uncertainty are then measured from the common central values. Out of these 3 different uncertainty sets, the worst condition is taken in either direction, called the "envelope". The average of the maximum and the minimum fluctuations of this envelope is used for the systematic uncertainty due to the PDF uncertainty in this analysis.

For this purpose, the PDF sets CT10 (α variation : 0.116 – 0.120), MSTW2008 (α variation : 0.110 – 0.130) and NNPDF2.1 (α variation of 0.114–0.124) were taken using the PDF Weight Technique in which only one MC is generated but weights for each PDF set are stored in an event-by-event basis followed by the observable weighting. The number of error sets are 52 for CT10, 40 for MSTW2008 and 100 for NNPDF2.1. The modified tolerance method is used for the first two PDFs, but, for NNPDF2.1, that works with the Monte Carlo sampling technique using data replica, in which a separate PDF fit is performed to each replica data set, average and standard deviation (σ) is calculated from 100 PDF sets, which is equivalent to the Hessian method for $\Delta\chi^2 = 1$. The PDF uncertainty bands (68% CL) with respect to the central set of CT10, from CT10, MSTW08 and NNPDF2.1 and the corresponding band of the envelope, is shown in Fig 10.1.

10.1.2   QCD and QED K-factors

The CI signal is estimated using the PYTHIA Monte Carlo, a leading order generator with CTEQ6.6L1 PDF set. In order to make higher order QCD and QED corrections, the QCD and QED k-factors are used in this analysis, which is discussed briefly in chapter 9. The statistical errors were negligible (~ 0.4 %) because of the huge sample (~ 200k events) in each of the 100 GeV steps, however the systematic error for QCD K-factor is assigned by 3% which is the size of the NNLO vs. NLO correction. For QED correction, size of the

correction itself is assigned as systematic in order to account for the maximum difference due to the possible unknown factors of the QED between the DY and the CI process.

10.2   Experimental sources

In addition to the theoretical sources, there are 3 different important experimental sources. These are:

(1) Uncertainty due to the luminosity measurement

(2) Uncertainty due to energy scale calibration

(3) Uncertainty due to reconstruction efficiency

The uncertainty in the luminosity is the biggest uncertainty in the experimental sources, and is measured to be 4.4%, as described in [88]. The electron reconstruction and identification uncertainty is measured for $P_T > 25 \text{GeV/c}$ using the tag-and-probe method with $Z \rightarrow$ ee events in which reconstruction uncertainty is 1.9% and identification uncertainty is found to be 1.5% [89]. The total of 5% is assigned in order to cover these two separate uncertainties. In order to measure the energy scale uncertainty, isolated electrons from W-boson decays are used to validate response corrections over a period of time [90].

10.3   Summary of systematic uncertainties

Table 10.2 summarizes all the systematic uncertainties in this analysis for the dielectron mass of 1.5 TeV/c^2, which is the approximate point to set the limit in both interferences. Among these, PDF and higher order corrections depend on the dielectron mass. The total error is combined in quadrature sum in terms of dielectron event yield, in which PDF uncertainty has the largest contribution, followed by the electron reconstruction. These

systematic uncertainties are included as the "nuisance parameters" in the limit setting procedure, where the signal is forced to fluctuate with the value randomly picked from the "log-normal" distribution with the mean.

Source	Rel. Uncert. (%)
PDF	12.1
Electron reconstruction and identification	5.0
Energy scale	1.0
QED k-factor	6.0
QCD k-factor	2.0
Luminosity	4.4

Table 10.2: Systematic uncertainties on dielectron yields above M_{ee}. The PDF and QED k-factor uncertainties are quoted at $M_{ee} = 1500$ GeV; all other uncertainties are independent of the minimum mass.

Chapter 11

Samples processing and Data-MC spectra comparison

In this chapter, the technical processes of the data analysis in the CMS are discussed in section 11.1 and section 11.2. In section 11.4, a comparison of CI models, the SM with the data from CMS is explained.

11.1   Job submission using CRAB

Since the millions of events need to be processed in order to select the events of interest and extract the parameter of interest, di-electron mass, in this analysis, local computing via the interactive method is not feasible, especially for the background estimation. Hence, a grid computing technique [91] is used to process the data and MC samples using CRAB (Cms Remote Analysis Builder)[92], the official CMS analysis software that helps to avoid the complexity in submitting, checking and retrieving the jobs, as well as to publish the output in the desired storage in which the user has permission to write. In this analysis,

T3_US_FNALLPC is used as the storage element for all CRAB submissions using an auto-mated python script interfaced with CRAB system given in A.1.

Even if CRAB can be used both as a standalone and with a server, the latter is more suitable for a huge task. An example would be generating the large number of detector-simulated Monte Carlo samples. In analysis work, the standalone mode was used, in which jobs are directly sent to the scheduler and the user has the responsibility of the jobs, which may need resubmission if the output is corrupted.

11.2   Ntuplization and processing

In order to test the Contact Interaction model against the Standard Model, the cross-section in terms of the number of events as a function of dielectron mass needs to be estimated for both models. In order to compute this information, the data set and Monte Carlo samples that are stored in various storage facilities of the CMS collaboration, need to be processed. The information storage in the CMS is based on ROOT, in which most of the datasets and MC samples are available in AOD format, specially chosen to satisfy the needs of a large fraction of analysis studies, which is a proper subset of RECO format. AOD format samples were used, which were processed with all the major pieces of information kept in "Ntuple" format. These Ntuples were stored in ROOT files using the programs given in A.2 and A.3. In order to select events of interest, a dedicated selection was applied in addition to the trigger requirement. The histograms of various observables were prepared using the programs given in B.1 and B.2 by processing the stored Ntuples. Finally, various pieces of information such as η, φ distributions of the dielectron mass, efficiency $\times$ acceptance of the selection criteria, trigger efficiency, etc. were calculated using the stored information in order to estimate the systematic uncertainty and to estimate the backgrounds from data. However,

most importantly, the di-electron mass was used to compare the various hypotheses, which can be calculated using the program given in D.4.

11.3 Re-binning and combining histograms from various samples

In order to compare visually with the good statistics, optimized binned histograms are preferred for a comparison of data to the SM. For that, the histograms of the dielectron mass from data are re-binned using the program given in C.1.1. Since, for the different MC processes, especially for the Drell-Yan process, the information comes from different samples for different regions, and mass dependent K-factors need to be used to re-scale for NLO or NNLO accuracy, programs given in C.1.3 and C.1.4 are used. For cumulative distribution, the program given in C.1.2 is used for both data and Monte Carlo. With these combined histograms, cumulative and variable binned differential Data-Mc comparison plots are prepared in which errors are separately calculated in order to combine the statistical and systematic uncertainties. The ROOT script for the Data-Mc spectra comparison plot can be found in C.1.6.

11.4 Data-Mc spectra comparison

Following the procedures outlined in sections 11.1, 11.2, 11.3, a Data-Mc spectra comparison plot was created in which the SM, data and the CI model are compared in Fig. 11.1 as a function of M_{ee} and Fig. 11.2 as a function of M_{ee}^{min}. Except for the backgrounds that involve at least one jet (in e^+e^- channel, as shown in these plots, which are estimated using fake rate method using data) all other SM dilepton sources are estimated from Monte Carlo Simulation. The estimated backgrounds are stacked, normalized to the integrated

luminosity of the data and superimposed on the distribution from data in a variable binned histogram beyond the dilepton mass of 300 GeV. The Quantitative comparison of the event yields of these plots are shown in Table 12.1, Tables 12.2 and 12.3. A detailed table for background yields is given in 11.1. The error bars shown in the data distribution is Poisson bar at 68% CL, which don't include the errors from Monte Carlo whereas the error bars on the pedestal plots include the total errors (stat + systematic). It appears that in Fig 11.2, a slight deviation in event yields is observed around 1 TeV, however the ratio of data to SM prediction plot in pedestal of Fig. 11.1 explains the fact that those deviations are not statistically significant and are consistent within 1 σ fluctuation. The highest di-electron mass observed in the data is 1776 GeV/c^2.

M_{ee}^{min}	DY	$t\bar{t}$	jets	Diboson	Other
300	3469.72	557.53	242.45	218.83	55.49
400	1190.80	138.61	76.78	76.03	17.52
500	516.59	41.24	31.63	28.80	6.21
600	246.24	15.76	13.44	11.56	2.10
700	126.49	3.89	5.91	5.20	1.16
800	68.11	1.30	2.92	2.85	1.07
900	39.23	0.65	1.20	1.98	1.01
1000	23.12	0.22	0.57	1.40	0.56
1100	13.82	0.45	0.28	0.68	0.09
1200	8.55	0.00	0.17	0.43	0.06
1300	5.39	0.00	0.10	0.32	0.06
1400	3.45	0.00	0.06	0.18	0.06
1500	2.22	0.00	0.04	0.14	0.00
1600	1.43	0.00	0.02	0.14	0.00
1700	0.94	0.00	0.01	0.00	0.00
1800	0.61	0.00	0.01	0.00	0.00
1900	0.42	0.00	0.01	0.00	0.00
2000	0.28	0.00	0.00	0.00	0.00

Table 11.1: DY and non-DY Event Yields in dielectron channel at 20 fb^{-1}.
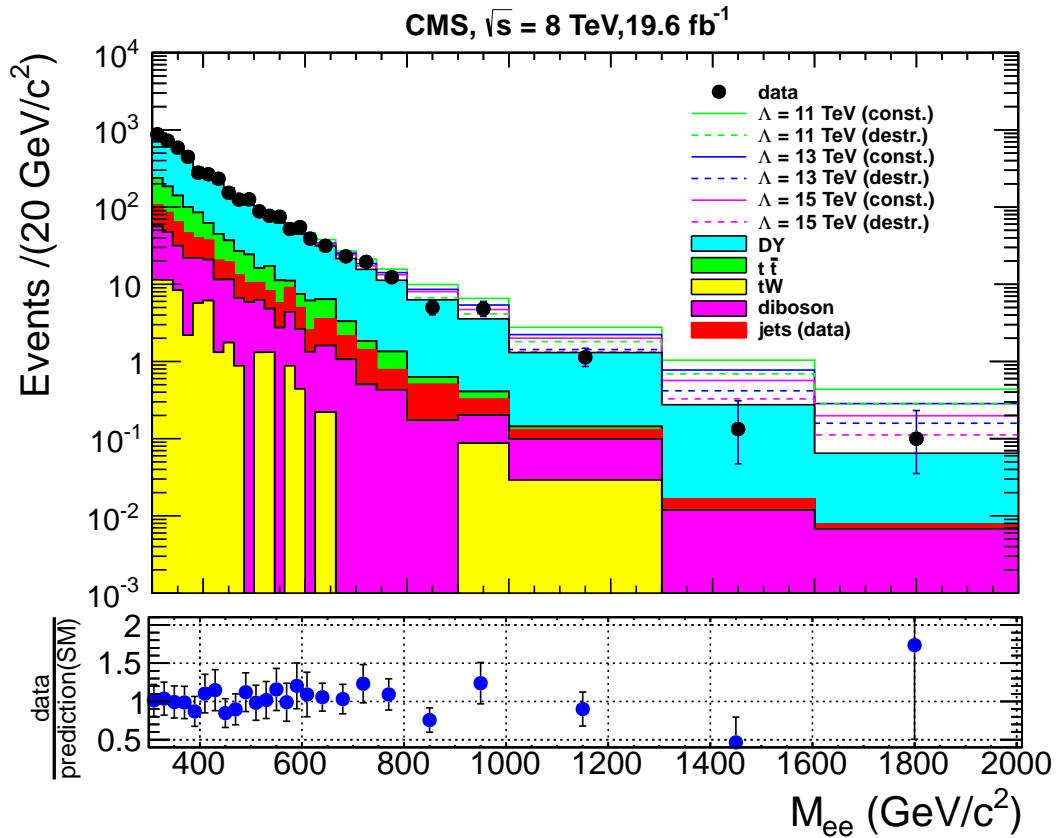
Figure 11.1: The dielectron mass spectrum at 19.6 fb^{-1} shown with MC predictions and with a variable binning. The distribution with error bars (Poisson 68 % CL) represent the data in which other dimuon sources from SM $(Z/\gamma^*, t\bar{t}, \text{diboson}, tW, \text{jets})$ are stacked, normalized and superimposed. The dielectron contribution from multi-jet backgrounds is estimated using fake rate method.
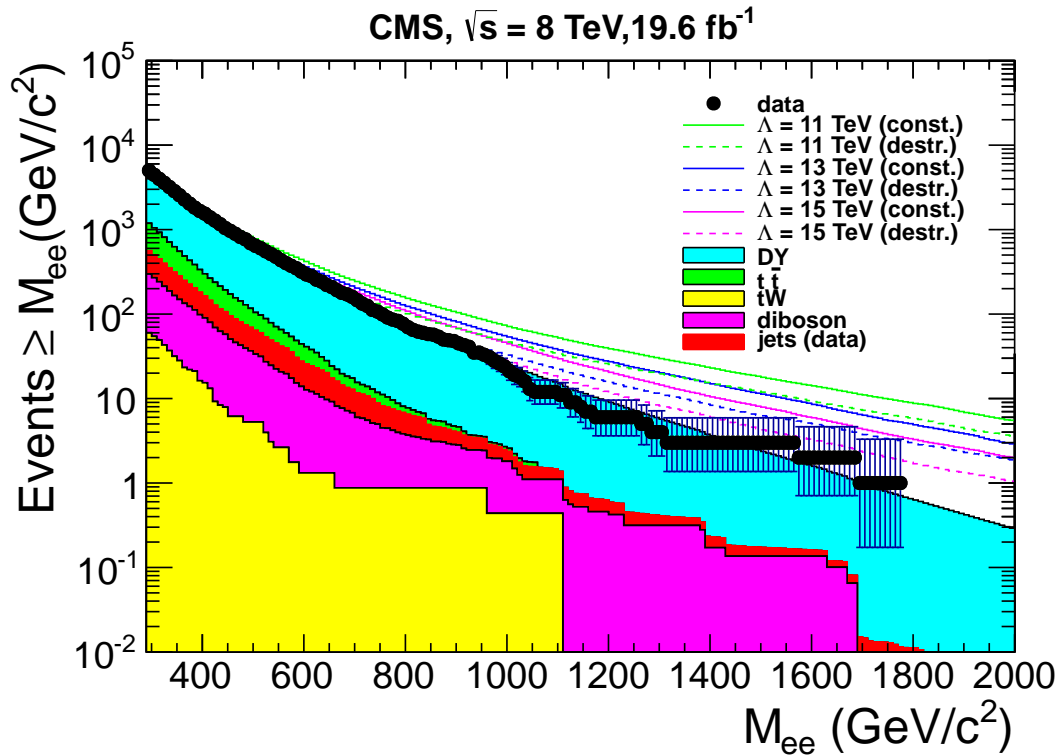
Figure 11.2: The dielectron mass spectrum at 19.6 fb^{-1} shown with MC predictions and for masses above M$_{ee}$. The distribution with error bars (Poisson 68 % CL) represent the data in which other dimuon sources from SM (Z/γ^*, t$\bar{t}$, diboson, tW, jets) are stacked, normalized and superimposed. The dielectron contribution from multi-jet backgrounds is estimated using fake rate method.

Chapter 12

Limit Setting

After all the dedicated isolation requirements, the dielectron mass spectra beyond 300 GeV was analyzed and a search for the contact interaction was performed. This chapter introduces the Frequentist and Bayesian statistics. In section 12.1.3, a brief explanation of CLs is given, which is the method used to set the lower limit of the Contact Interaction energy scale (Λ) as data are found to be consistent with the prediction made by the Standard Model as will be seen in 12.1.4.

12.1 Statistical method

Since experimental particle physics research involves extremely complex and expensive detector facilities as well as effort from a huge number of physicists, it is extremely important to optimize the data by using good statistical practices so that the maximum possible information can be extracted from the data. There are several statistical procedures have to be carried out by particle physicists in order to get the results, such as separating signal from background, comparing the hypotheses, etc. The heart of the analysis in the particle physics is the limit setting procedure to the parameter of interest if the signal is absent or is the decision making by checking the discovery potential if the signal is present. For

this purpose, the two different statistical approaches that are commonly used are called the Bayesian and the Frequentist methods. Basically, in both the Frequestist and Bayesian techniques, the search results can be formulated in terms of the hypothesis test. In this analysis, the null hypothesis is that the CI signal is absent, which means the data can be explained by the prediction from the SM. The alternate hypothesis is that, it exists, which is the discovery of the CI process. These tasks can be formally defined and quantified by checking which of the hypotheses favors the data.

12.1.1 Frequentist method

In the Frequentist method, a convenient test statistic q_μ is constructed, which is used to distinguish the null hypothesis (background-like) and the alternative hypothesis (signal + background like). By the construction of Neyman-Pearson lemma, the ratio of likelihoods is considered the most powerful discriminator.

$$q_\mu = -2\ln\frac{\mathcal{L}(\text{data}|s(\mu) + b)}{\mathcal{L}(\text{data}|b)} \tag{12.1}$$

where, μ refers to the parameter of interest. The values from data and the predictions give the baseline for the estimation of total probability for both hypotheses. The probability density function is constructed using a large number of pseudo data, using the Poisson fluctuation, and by using the "toy Monte Carlo method". Now, suppose the value of q_{obs} obtained from the data is given as in the figure 12.1 with q_μ. Using the constructed PDFs, p_{s+b}, the p-value for signal + background hypothesis and p_b, corresponding value for background only hypothesis can be obtained using following relation:

$$p_{s+b} = P(q \geq q_{obs}|s(\mu) + b) = \int_{q_{obs}}^{\infty} f(q|s(\mu) + b)dq \tag{12.2}$$

$$p_b = P(q \leq_{obs} |b) = \int_{-\infty}^{q_{obs}} f(q|b)dq \qquad (12.3)$$

In the method called CL_{s+b}, The alternative hypothesis (signal+ background) is supposed to be excluded at 95% confidence interval for the given parameter μ if, p_{s+b} in equation 12.2, satisfies the condition, $p_{s+b} \leq 0.05$. This method is expected to give a better result only if the probability density function of the different hypothesis are well separated and the cross-section between the two are comparable. Unfortunately, this does not happen normally in search of physics beyond the Standard Model. A modified version of CL_{s+b} is very popular and will be discussed in detail in 12.1.3.
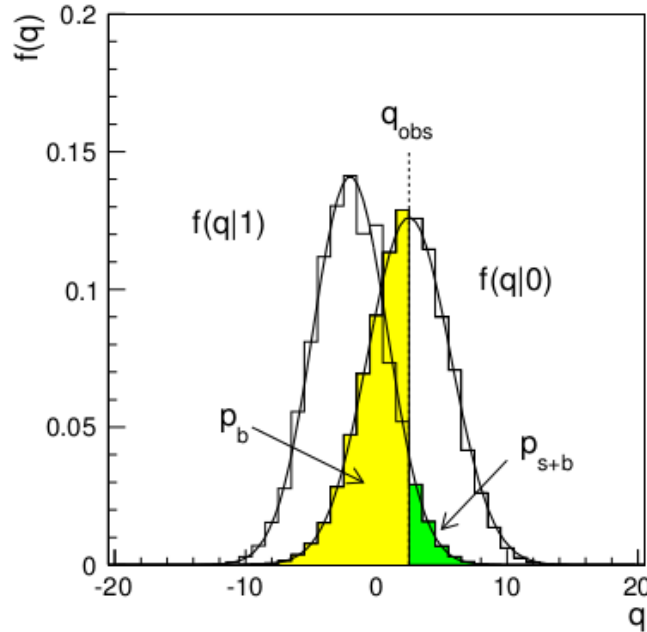
Figure 12.1: The distribution of the variable q of equation 12.1 under null and alternative hypothesis. [93]

12.1.2 Bayesian method

In Bayesian method, subjective probability is used which allows to use the information extracted by previous experiments that could be very important to accept or reject the

hypothesis in a more robust way. However, this is subjective which can be argued as a trade-off of the possible bias in the conclusion. It may not be a great tool, especially if a conservative conclusion is desired. For the posterior probability, the formula established using the Bayesian approach of probability is used.

$$P(H/\vec{x}) = \frac{P(\vec{x}/H)\pi(H)}{\int P(\vec{x}/H)\pi(H)dH} \tag{12.4}$$

Where $P(\vec{x}/H)$ is the probability of observing the data under hypothesis H, $\pi(H)$ is the prior probability, $\int P(\vec{x}/H)\pi(H)dH$ is the normalization factor by summing over all the available hypotheses, and $P(H/\vec{x})$ is the posterior probability. For example, in order to search for contact interaction in a simple counting experiment, say, s is the number of events from CI, which is represented by μ and b is the background. Now, the parameter μ can be expressed in terms of signal (s), background (b) and "nuisance" parameters (ν) which are any parameters that are not under investigation but still have influence on the prediction.

$$\mu = s(\Lambda, \nu) + b(\nu) \tag{12.5}$$

The likelihood of observing the set of n events for a given invariant mass bin with total bins of N_{bins} is given by the product of Poisson probabilities:

$$\mathcal{L}(n/\Lambda, \nu) = \prod_{k=1}^{N_b ins} \frac{\mu_k^{n_k} e^{-\mu_k}}{n_k!} \tag{12.6}$$

Now, using equation 12.4, The posterior probability for Λ for given n observed events can be calculated using the best prior probability. Normally, the prior probability is taken to be flat to make it free from the bias of previous experiments. And, finally, in order to set a limit with 95% Bayesian credibility interval limit,

$$\int_0^{\Lambda_{limit}} P(\Lambda'/n)d\Lambda' = 0.95 \tag{12.7}$$

that can be solved for Λ_{limit}.

12.1.3 The modified Frequentist (CLs) method

As mentioned in section 12.1.1, the $\text{CL}_{\text{s+b}}$ procedure fails to perform well when signal is negligible with respect to background. If the observed events have downward fluctuation, given that signal is very small, then there will be a huge overlap in the probability density functions between the two hypotheses, causing the signal to be excluded at 95% CL. In order to correct this false exclusion, the $\text{CL}_{\text{s+b}}$ method is modified to make some trade off between the p-value of the signal+background hypothesis with the background hypothesis with higher the exclusion potential, the lower the false exclusion rate (p-value of the signal+background hypothesis). This modified version is commonly known as CL_{s} method.

The CL_{s} method [77, 94], the value of CL_{s} is given in the following equation:

$$\text{CL}_{\text{s}} = \frac{\text{CL}_{\text{s+b}}}{1 - \text{CL}_{\text{b}}} = \frac{P(q \geq q_{\text{obs}}|s(\mu) + b)}{1 - P(q \leq_{\text{obs}} |b)} = \frac{\int_{q_{\text{obs}}}^{\infty} f(q|s(\mu) + b)dq}{1 - \int_{-\infty}^{q_{\text{obs}}} f(q|b)dq} \tag{12.8}$$

Where the p-value of the signal hypothesis depends upon the p-value of background hypothesis, making the CL_{s} always greater than $p_{\text{s+b}}$, reducing the false exclusion rate. The penalty depends upon how well these distributions are separated. Similar to $\text{CL}_{\text{s+b}}$, in order to exclude the signal + background hypothesis at 95% CL, the value of CL_{s} is required to be less than or equal to 0.05. As an example of how these variables are related to the probability distribution of the q value, the value of $\text{CL}_{\text{s+b}}$ and CL_{b} are shown in Fig. 12.2 with the probability taken from the LEP experiment in search of the Higgs. In the "LHC-style" CL_{s} method uses the profile-likelihood ratio [95] as the test statistic as shown in the following equation.

$$\lambda(\mu) = \frac{\mathcal{L}(\mu, \hat{\hat{\vartheta}})}{\mathcal{L}(\hat{\mu}, \hat{\vartheta})} \tag{12.9}$$

Where ϑ is the collection of the nuisance parameters, μ can be 0 or 1 depending upon the

signal or background hypothesis. $\hat{\hat{\vartheta}}$ is the conditional maximum likelihood estimator of ϑ which is the function of μ. $\hat{\mu}$ and $\hat{\vartheta}$ are the true maximum likelihood estimators. By this construction, the profile likelihood ratio shows the compatibility between the data and the hypothesis when μ coincides or deviates with $\hat{\mu}$.
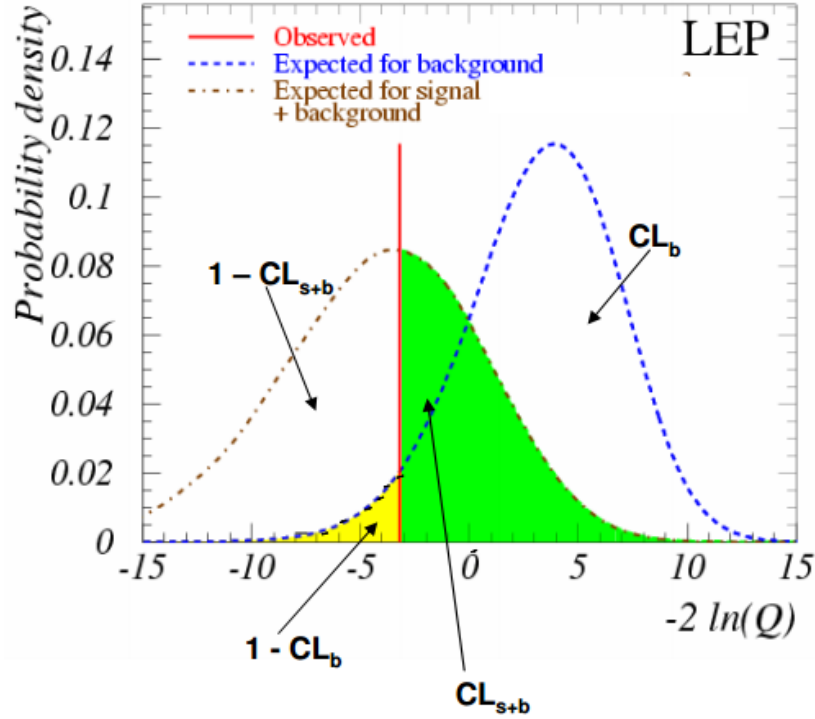
Figure 12.2: The graphical representation of the terms CL_s, CL_{s+b}, in equation 12.8 from the Higgs search of mass 115 GeV/c^2 [96].

12.1.4 Consistency check of data with the SM

In order to check the consistency between data and the null hypothesis, a comparison is performed in the distributions of the dielectron mass spectra from the data and the SM prediction for a given minimum mass as shown in Fig. 11.1, which is normalized to the luminosity of data at 19.6 fb^{-1}. In order to make the visual check better, a cumulative distribution as a function of M_{ee} is shown in Fig. 11.2. In the pedestal plot of the Fig. 11.1,

a ratio of the data to the SM prediction is shown, including the total error of the measurement, which clearly shows agreement within the statistical fluctuation. The quantitative description of the plots is given in table 12.1.

In addition to the visual inspection, the agreement of the data with the null hypothesis is confirmed by measuring "the p-value". In order to measure the p-value, the following test statistics is used by optimizing the nuisance parameters.

$$q_\mu = -2\ln\frac{\mathcal{L}(\text{data}|\text{b})}{\mathcal{L}(\text{data}|\text{s}(\mu) + \text{b})} \tag{12.10}$$

As explained in section 12.1.3, the probability density function is constructed by using "toy mc method", using the Poisson probability functions followed by the calculation of cumulative probability, using equation 12.3. The p-value is expected to be higher than 5 % in order to be consistent. In this analysis, with the 2012 dataset, a p-value of 41% was found above 1 TeV of M_{ee}, which shows a strong consistency with the SM.

Table 12.1: Data and predicted event yields in the dielectron channel.

$M_{ee}^{\min}(\text{GeV/c}^2)$	Data	SM prediction (DY + Non DY)
300	4459 ± 66.78	4456.9 ± 76.3
400	1544 ± 39.29	1469.2 ± 18.7
500	641 ± 25.32	611.7 ± 10.4
600	294 ± 17.15	282.8 ± 3.8
700	146 ± 12.08	139.7 ± 2.4
800	70 ± 8.37	74.6 ± 1.7
900	45 ± 6.71	43.1 ± 1.3
1000	21 ± 4.58	25.3 ± 1.0
1100	11 ± 3.32	15.4 ± 0.8
1200	6 ± 2.45	8.97 ± 0.53
1300	4 ± 2	5.70 ± 0.42
1400	3 ± 1.73	3.62 ± 0.32
1500	3 ± 1.73	2.36 ± 0.25
1600	2 ± 1.41	1.56 ± 0.22
1700	1 ± 1	0.93 ± 0.18
1800	0 ± 0	0.61 ± 0.15
1900	0 ± 0	0.422 ± 0.003
2000	0 ± 0	0.276 ± 0.003

12.1.5   Limit setting

With the agreement between the data and the SM prediction, the possibility of CI process is ruled out. In order to set the 95 % CL lower bound on the CI energy scale Λ, the modified Frequentist method as described in section 12.1.3 was used with the single binned counting experiment. The official CMS package, which is certified by the CMS statistical committee, a CL_s program written in C++ and interfaced with the ROOSTAT in the framework of ROOT version 5.32.00 or higher is used. Using the various functions such as pseudo-random generation from the ROOSTAT, this CL_s takes a number of inputs and returns the median, a $1-\sigma$ and $2-\sigma$ cross-section for the CI/DY process that can be excluded at 95 % CL. This macro is integrated with a program using PYROOT in order to take the arguments and convert these cross-sections in terms of the sensible CI energy scale. This is done by using the event yields in the CI model from the Full-simulated CI samples. The program is given in D.1.

The input taken by the CL_s program is as follows:

(1) Nominal integrated luminosity (pb^{-1})

(2) Absolute error on the integrated luminosity

(3) Nominal value of the efficiency $\times$ acceptance (in range 0 to 1)

(4) Absolute error on the efficiency $\times$ acceptance

(5) Nominal value of the background estimate

(6) Absolute error on the background

(7) Number of observed events (not used for the expected limit)

(8) Number of pseudo experiments to perform for the expected limit calculation

(9) Statistics choice (True = gauss or False = Poisson)

(10) Nuisance Model - distribution function which is used in integration over nuisance parameters : (0 – Gaussian, 1–lognormal, 2 – gamma)

(11) Method of statistical inference ("Bayesian", "mcmc", "cls", "fc", "workspace")

(12) plotFileName

(13) Seed for random number generation

The integrated luminosity for this analysis is 19616 pb^{-1} and 4.4 % error corresponding to the luminosity is 862.4 pb^{-1}. The value of efficiency $\times$ acceptance is set to 1 and the absolute error is set to 0. The nominal value of background estimate (DY + Non DY) is given in Table 12.1. The absolute error on the background is estimated by combining the statistical error with the different systematics described in Chapter 10, in which the % effect on the event yield is given in Table 10.3. The number of observed events are shown in Table 12.1 given as a function of minimum dielectron mass. The statistics choice is used as the "Poisson" and the nuisance model is taken as "lognormal". The last 3 arguments are not very interesting but are simply the mechanical parts of the process.

With the above arguments, the median of the lower limit on the cross-sections for CI signals are returned as a function of M_{ee}^{low} corresponding to the 95% CL along with $1 - \sigma$ and $2 - \sigma$ fluctuations on that limit due to the nuisance parameters. These cross-sections can be interpreted in terms of the event yields using simple multiplication of the luminosity followed by mapping in the interval of the Λ values from Tables 12.2 and 12.3. A simple interpolation is done in order to find the exact Λ value for the limit for a given minimum mass.

The observed and the expected lower limits on Λ as a function of M_{ee}^{low} at 95% CL are shown in Figs 12.3 and 12.4. Due to the fact that expectations of dielectron event yield slightly exceed the data around 1 TeV (Fig. 11.2), though it is not statistically significant, the observed limit peaks around 1 TeV (exceeding slightly above the $1 - \sigma$ fluctuation band of the expected limit). The limit is taken at $M_{ee}^{low} = 1.4$ TeV for destructive and 1.2 TeV for constructive interference. The observed (expected) limits are 13.1 TeV (12.9 TeV) for destructive interference and 18.3 TeV (16.7 TeV) for constructive interference.

Table 12.2: Observed and predicted number of dielectron events using the SM and LLIM over a range of values of M_{ee}. For LLIM, predictions are shown for a range of Λ values, for destructive interference. Both the SM and CI predictions include small contributions from non-DY backgrounds. The integrated luminosity is 19.6 fb^{-1}.

M_{min} (GeV/c^2)	500	600	700	800	900	1000	1100	1200	1300	1400	1500
Source	Number of Events										
data	641	294	146	70	45	21	11	6	4	3	3
SM pred.	611.7	282.8	139.7	74.6	43.1	25.3	15.4	9.0	5.7	3.6	2.4
σ(SM pred.)	55.5	26.3	13.4	7.4	4.4	2.8	1.9	1.2	0.8	0.6	0.4
Λ (TeV)											
20	561.6	261.0	130.3	71.1	41.3	24.9	15.3	9.1	5.9	3.8	2.5
19	559.9	259.7	129.6	71.0	41.4	25.1	15.5	9.4	6.1	4.0	2.6
18	558.3	258.4	129.1	71.0	41.7	25.5	15.9	9.7	6.4	4.3	2.8
17	556.8	257.3	128.8	71.3	42.2	26.0	16.4	10.2	6.8	4.6	3.2
16	555.6	256.5	128.8	71.9	43.0	26.9	17.2	10.9	7.4	5.1	3.6
15	555.1	256.4	129.4	73.0	44.3	28.1	18.4	11.9	8.3	5.9	4.2
14	555.6	257.5	131.1	75.1	46.3	30.1	20.1	13.5	9.6	7.0	5.1
13	558.3	260.6	134.5	78.7	49.7	33.1	22.8	15.8	11.6	8.6	6.5
12	564.8	267.5	141.0	84.7	55.0	37.8	26.9	19.3	14.6	11.1	8.6
11	578.2	281.0	152.7	95.1	63.9	45.5	33.5	24.9	19.3	15.0	11.9
10	604.4	307.0	174.2	113.0	78.8	58.2	44.3	34.0	27.1	21.4	17.2
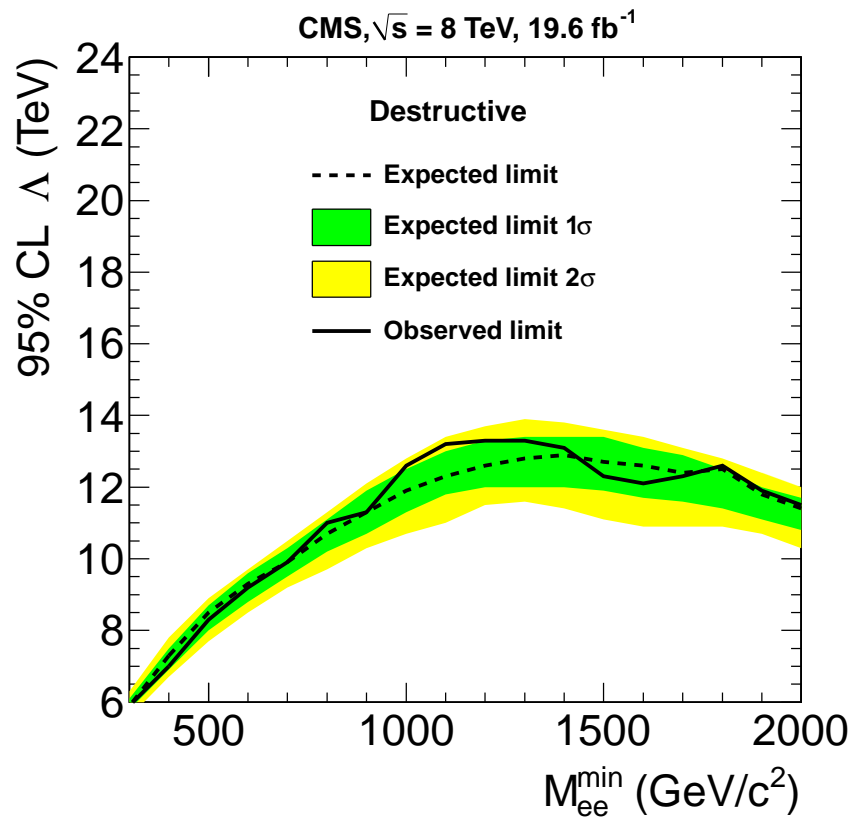
Figure 12.3: Observed and expected limits as a function of M_{ee} for destructive interference.
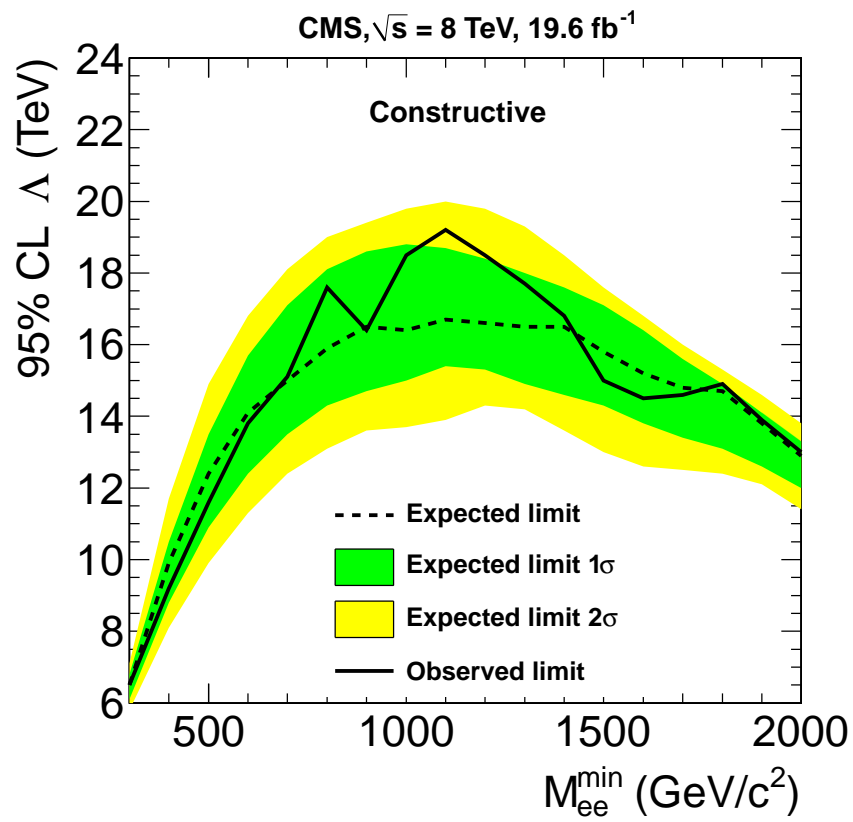
Figure 12.4: Observed and expected limits as a function of M_{ee} for constructive interference.

Table 12.3: Observed and expected number of events for the range of M_{min} in which the 95% CL limit on Λ is established. The expected yields for finite Λ values are shown for constructive interference. Both the SM and CI predictions include small contributions from non-DY backgrounds. The values correspond to an integrated luminosity of 19.6fb^{-1}.

M_{min} (GeV/c^2)	500	600	700	800	900	1000	1100	1200	1300	1400	1500
Source	Number of Events										
data	641	294	146	70	45	21	11	6	4	3	3
SM pred.	611.7	282.8	139.7	74.6	43.1	25.3	15.4	9.0	5.7	3.6	2.4
σ(SM pred.)	55.5	26.3	13.4	7.4	4.4	2.8	1.9	1.2	0.8	0.6	0.4
Λ (TeV)											
20	622.8	306.2	159.2	89.8	53.9	33.4	21.6	13.8	9.4	6.5	4.6
19	627.4	309.4	161.6	91.6	55.3	34.5	22.4	14.5	10.0	7.0	5.0
18	633.0	313.4	164.6	93.9	57.1	35.8	23.5	15.4	10.7	7.5	5.5
17	639.8	318.3	168.3	96.8	59.4	37.5	24.9	16.5	11.6	8.3	6.1
16	648.2	324.6	173.0	100.5	62.4	39.8	26.8	18.0	12.9	9.3	6.9
15	659.0	332.7	179.2	105.4	66.3	42.8	29.3	20.0	14.5	10.6	7.9
14	672.8	343.3	187.4	112.0	71.6	46.9	32.7	22.8	16.7	12.3	9.3
13	691.1	357.6	198.7	121.1	78.9	52.8	37.5	26.6	19.8	14.8	11.4
12	716.0	377.5	214.4	134.0	89.3	61.2	44.4	32.2	24.2	18.4	14.2
11	751.0	405.9	237.4	152.9	104.6	73.7	54.6	40.4	30.8	23.8	18.6
10	802.1	448.5	272.0	181.9	128.1	93.2	70.6	53.2	41.1	32.1	25.3

Chapter 13

Summary and discussion

13.1   Summary and discussion

The analysis presented in this thesis is performed in the framework of the CMS experiment at $\sqrt{s} = 8$ TeV in 2012 at the LHC, in which the integrated luminosity of 19.6 fb^{-1} CMS data was collected and tested against the Standard Model in the context of the Left-Left isoscalar model of the contact interaction. The huge center of mass energy and the instantaneous luminosity as well as the excellent detector resolution (with the dedicated isolation criteria established by significant efforts on the part of various institutions all over the globe) provided the promising sensitivity in this search.

For the prediction, the Monte Carlo Method is utilized in many cases. The dedicated software, computing resources and the huge effort of the theorist made the prediction possible with a high level of accuracy. The data-driven technique, developed for the prediction of jet faking leptons in order to address the inefficiency of the Monte Carlo model, adds to the sensitivity in the results presented in this thesis.

With the sincere efforts mentioned above, the dielectron mass spectrum above 300 GeV

was analyzed where no significant deviation from the SM prediction was observed. At 95% CL, lower limits on the energy scale of llqq contact interactions Λ in the Left-Left isoscalar model were placed in destructive (13.1 TeV) and constructive (18.3 TeV) interference. These limits are the most stringent to date.

As shown in the equation 2.4, as Λ goes higher, the whole spectrum converges to the DY spectrum, which degrades the sensitivity of the search. Although the sensitivity of the search increases with the center of mass energy, since the limit is already set up to 18 TeV in the dielectron channel in this work, and up to 26 TeV in LEP (in the reverse process compared to the process this analysis is based upon, instead of the LLIM) a dijet final state or helicity non conserving model could be more useful. There are several searches that are being performed in the CMS experiment in which the search for the excited quarks is a popular model for the compositeness search. Nevertheless, it is very interesting to observe the data from the CMS experiment at 13 TeV in 2015, which has a great potential to discover or rule out several models, including the Contact Interaction, at least in the accessible energy range of today's technology.

In addition to the CMS experiment at 13 TeV, muon-muon colliders in the future will be very interesting in that these results can be tested with great resolution. Any collider experiments in the future with higher energies and better resolution could reveal the substructure of quarks and leptons.

APPENDIX A

A.1 Module for job submission via CRAB

####################

Module Name : CRAB.py

Author : Pramod Lamichhane

Date : Aug 2013

Last modified : Feb 2013

Purpose : This program is used to submit crab jobs (automatically) in grids, which uses the programs given in A.2 and A.3

to generate Ntuples for Full Sim MC and DATA from CMS detector

####################

```
samples=[ ]
import os
pl_SAMPLE=open("SAMPLES.LOG","w")
mass_cut=[300,500,800]
option=["Des","Con"]
lambda_values=[9,11,13,15,17,19]
count=0
```

```
for i in range(0,len(mass_cut)):

    samples.append("/CIToEE_ITCM5_M-'+str(mass_cut[i])+'_TuneZ2star_8TeV-pythia6/Summer12_DR53X-

PU_S10_START53_V7A-v1/AODSIM")

    for j in range(0,len(option)):

        for k in range(0,len(lambda_values)):

            if not ((lambda_values[k]==17 or lambda_values[k]==19 ) and (option[j]=="Des")):

                samples.append("/CIToEE_"+option[j]+"_Lambda-"+str(lambda_values[k])+"_M-"+

str(mass_cut[i])+"_TuneZ2star_8TeV-pythia6/Summer12_DR53X-PU_S10_START53_V7A-v1/AODSIM")

for i in range(0,len(samples)):

    CI_DY=samples[i].strip("/").split("_")

    print CI_DY[0],CI_DY[1],CI_DY[2],CI_DY[3]


    mycrab="""

    [CMSSW]

total_number_of_events=-1

events_per_job =60000

pset =ci_dielectron_cfg.py

datasetpath =%s

output_file = NTUPLE.root


    [USER]

return_data = 1

ui_working_dir=%s


    [CRAB]

jobtype = cmssw

scheduler = condor
```

[GRID]

rb = CERN

proxy_server = myproxy.cern.ch

"'%(samples[i],CI_DY[0]+'_'+CI_DY[1]+"_"+CI_DY[2]+"_"+CI_DY[3])

   pl_=open(CI_DY[0]+"_"+CI_DY[1]+"_"+CI_DY[2]+"_"+CI_DY[3]+".cfg","w")

   pl_.writelines(mycrab)

   pl_.close()

   pl_SAMPLE.writelines(samples[i]+"n")

   os.system("crab -create -submit-cfg"+CI_DY[0]+"_"

+CI_DY[1]+"_"+CI_DY[2]+"_"+CI_DY[3]+".cfg>>crab_info_for_noofevents.log")

pl_SAMPLE.close()

A.2   Python module

####################

Module Name : CI_DIELECTRON.py

Author : Pramod Lamichhane

Date : Aug 2013

Last modified : Feb 2013

Purpose : This program is integrated to C++ source code given in A.3

to generate Ntuples for Full Sim MC and DATA from CMS detector

####################

import FWCore.ParameterSet.Config as cms

import os

USE_MC=1#1 means MC AND 0 MEANS DATA

PDF_WORK=True

```
process = cms.Process("Demo")

process.load("FWCore.MessageService.MessageLogger_cfi")

process.load("TrackingTools/TransientTrack/TransientTrackBuilder_cfi")

process.load("Configuration/StandardSequences/MagneticField_cff")

process.load("Configuration.Geometry.GeometryIdeal_cff")

process.load("Configuration/StandardSequences/FrontierConditions_GlobalTag_cff")

##########RHO CALCULATION ######

process.load("RecoJets.JetProducers.kt4PFJets_cfi")

process.kt6PFJets = process.kt4PFJets.clone( rParam = 0.6, doRhoFastjet = True )

process.kt6PFJets.Rho_EtaMax = cms.double(2.5)

###################

process.otherStuff = cms.Sequence( process.kt6PFJets )


process.load("RecoMET.METFilters.eeBadScFilter_cfi")

if USE_MC==1:

    process.GlobalTag.globaltag = "START53_V11::All"

elif USE_MC==0:

    process.GlobalTag.globaltag = "GR_R_53_V8::All"


else:

    print "UNSPECIFIED OPTION"

    exit(1)

process.maxEvents = cms.untracked.PSet( input = cms.untracked.int32(5000) )

process.source = cms.Source("PoolSource",

noEventSort = cms.untracked.bool(True),

duplicateCheckMode = cms.untracked.string("noDuplicateCheck"),

fileNames = cms.untracked.vstring(
```

```
'/store/mc/Summer12/DYToEE_M_200_TuneZ2star_8TeV_pythia6/AODSIM/PU_S7_START52_V9-
v1/0000/FCC11233-7F9E-E111-92DD-002354EF3BE0.root',
)
)


process.primaryVertexFilter = cms.EDFilter("GoodVertexFilter",
vertexCollection = cms.InputTag("offlinePrimaryVertices"),
minimumNDOF = cms.uint32 (4),
maxAbsZ = cms.double (24),
maxd0 = cms.double (2)
)
#process.primaryVertexPath = cms.Path(process.primaryVertexFilter)


process.noscraping = cms.EDFilter("FilterOutScraping",
applyfilter = cms.untracked.bool(True),
debugOn = cms.untracked.bool(False),
numtrack = cms.untracked.uint32(10),
thresh = cms.untracked.double(0.25)
)
process.Pramod_mc = cms.EDFilter("HLTHighLevel",
TriggerResultsTag = cms.InputTag("TriggerResults","","HLT"),
HLTPaths = cms.vstring("HLT_DoubleEle33_CaloIdL_GsfTrkIdVL_v*"),
eventSetupPathsKey = cms.string(""),
andOr = cms.bool(True),
throw = cms.bool(True)
)
process.Pramod_data = cms.EDFilter("HLTHighLevel",
```

TriggerResultsTag = cms.InputTag("TriggerResults","","HLT"),

HLTPaths = cms.vstring("HLT_DoubleEle33_CaloIdL_v5"),

eventSetupPathsKey = cms.string(""), # not empty => use read paths from AlCaRecoTrigger-

BitsRcd via this key

andOr = cms.bool(True), # how to deal with multiple triggers: True (OR) accept if ANY is true,

False (AND) accept if ALL are true

throw = cms.bool(True) # throw exception on unknown path names

)

process.TRIGGER_EVENT = cms.EDAnalyzer("WAYNE_STATE_CI_DIELECTRON',

MC_OR_DATA=cms.double(USE_MC),

triggerSummaryLabel = cms.InputTag("hltTriggerSummaryAOD","","HLT"),#"REDIGI311X")

pdfSet = cms.string("CT10.LHgrid"),

PDF_WORK_GARNE_KI_NO=cms.bool(PDF_WORK)

)

process.TFileService = cms.Service("TFileService",

fileName = cms.string("NTUPLE.root")

)

if process.TRIGGER_EVENT.MC_OR_DATA==1:

    process.p = cms.Path(process.otherStuff*process.eeBadScFilter*process.primaryVertexFilter*

    process.noscraping*process.Pramod_mc*process.TRIGGER_EVENT)

    process.NOT_TRIGGER_EVENT = process.TRIGGER_EVENT.clone()

    process.p1 = cms.Path(process.Pramod_mc*process.NOT_TRIGGER_EVENT)

elif process.TRIGGER_EVENT.MC_OR_DATA==0:

    process.p = cms.Path(process.otherStuff*process.eeBadScFilter*process.primaryVertexFilter*

    process.noscraping*process.Pramod_data*process.TRIGGER_EVENT)

    process.NOT_TRIGGER_EVENT = process.TRIGGER_EVENT.clone()

process.p1 = cms.Path(process.Pramod_data*process.NOT_TRIGGER_EVENT)

A.3 C++ source code

####################

Module Name : WAYNE_STATE_CI_DIELECTRON.cc

Author : Pramod Lamichhane

Date : Aug 2013

Last modified : Feb 2013

Purpose : This program is integrated to Python script in A.2 and CMS BuildFile

to generate Ntuples for Full Sim MC and DATA from CMS detector

####################

// Package: WAYNE_STATE_CI_DIELECTRON

// Class: WAYNE_STATE_CI_DIELECTRON

// AUTHOR : PRAMOD LAMICHHANE

//const int PDF_SET_KATI_OTA=100;

const int PDF_SET_KATI_OTA=100;

#include "TTree.h"

#include "TBranch.h"

#include < memory >

#include "FWCore/Framework/interface/Frameworkfwd.h"

#include "FWCore/Framework/interface/EDAnalyzer.h"

#include "FWCore/Framework/interface/Event.h"

#include "FWCore/Framework/interface/MakerMacros.h"

```cpp
#include "FWCore/ParameterSet/interface/ParameterSet.h"

#include "FWCore/Framework/interface/Event.h"

#include "FWCore/Framework/interface/Run.h"

#include "TH1.h"

#include "TH1F.h"

#include "TH2F.h"

#include "TProfile.h"

#include < TROOT.h >

#include "TF1.h"

#include "TMath.h"

#include < TSystem.h >

#include "TFile.h"

#include < TCanvas.h >

#include < cmath >

#include < iostream >

#include < fstream >

#include < vector >

#include < functional >

#include < Math/VectorUtil.h >

#include "DataFormats/Common/interface/Handle.h"

#include "DataFormats/HepMCCandidate/interface/GenParticle.h"

#include "DataFormats/HepMCCandidate/interface/GenParticleFwd.h"

#include "DataFormats/Candidate/interface/CompositePtrCandidate.h"

#include "DataFormats/Candidate/interface/Candidate.h"

#include "DataFormats/Candidate/interface/CandAssociation.h"

#include "FWCore/ServiceRegistry/interface/Service.h"

#include "CommonTools/UtilAlgos/interface/TFileService.h"
```

//muons

#include "DataFormats/GsfTrackReco/interface/GsfTrack.h"

#include "DataFormats/EgammaCandidates/interface/GsfElectron.h"

#include "DataFormats/EgammaReco/interface/BasicClusterFwd.h"

#include "DataFormats/EgammaReco/interface/SuperClusterFwd.h"

#include "DataFormats/EgammaReco/interface/ElectronSeed.h"

#include "DataFormats/EgammaReco/interface/ElectronSeedFwd.h"

#include "DataFormats/JetReco/interface/CaloJetCollection.h"

#include "DataFormats/EcalDetId/interface/EcalSubdetector.h"

#include "DataFormats/BeamSpot/interface/BeamSpot.h"

#include "DataFormats/Common/interface/TriggerResults.h"

#include "DataFormats/MuonReco/interface/MuonFwd.h"

#include "DataFormats/MuonReco/interface/Muon.h"

#include "DataFormats/RecoCandidate/interface/RecoCandidate.h"

#include "DataFormats/MuonReco/interface/MuonChamberMatch.h"

#include "DataFormats/MuonReco/interface/MuonIsolation.h"

#include "DataFormats/MuonReco/interface/MuonEnergy.h"

#include "DataFormats/MuonReco/interface/MuonTime.h"

#include "DataFormats/MuonReco/interface/MuonQuality.h"

#include "DataFormats/MuonReco/interface/MuonSelectors.h"

#include "DataFormats/RecoCandidate/interface/IsoDeposit.h"

#include "DataFormats/RecoCandidate/interface/IsoDepositFwd.h"

#include "DataFormats/TrackReco/interface/Track.h"

#include "DataFormats/TrackReco/interface/TrackFwd.h"

#include "DataFormats/Common/interface/ValueMap.h"

#include "FWCore/Utilities/interface/InputTag.h"

#include "DataFormats/BeamSpot/interface/BeamSpot.h"

#include "DataFormats/Common/interface/Handle.h"

#include "DataFormats/HepMCCandidate/interface/GenParticle.h"

#include "DataFormats/HepMCCandidate/interface/GenParticleFwd.h"

#include "DataFormats/Candidate/interface/CompositePtrCandidate.h"

//trigger

#include "DataFormats/L1GlobalMuonTrigger/interface/L1MuRegionalCand.h"

#include "DataFormats/L1GlobalMuonTrigger/interface/L1MuGMTReadoutCollection.h"

#include "DataFormats/L1GlobalTrigger/interface/L1GlobalTriggerReadoutRecord.h"

#include "DataFormats/L1GlobalTrigger/interface/L1GtPsbWord.h"

#include "FWCore/Common/interface/TriggerNames.h"

#include "DataFormats/Common/interface/TriggerResults.h"

#include "DataFormats/HLTReco/interface/TriggerObject.h"

#include "DataFormats/HLTReco/interface/TriggerEvent.h"

#include "DataFormats/HLTReco/interface/TriggerTypeDefs.h"

#include "HLTrigger/HLTcore/interface/HLTConfigProvider.h"

#include "DataFormats/TrackReco/interface/Track.h"

#include "DataFormats/TrackReco/interface/TrackFwd.h"

#include "DataFormats/VertexReco/interface/Vertex.h"

#include "DataFormats/VertexReco/interface/VertexFwd.h"

#include "DataFormats/Math/interface/deltaR.h"

#include "TTree.h"

#include "RecoVertex/KalmanVertexFit/interface/KalmanVertexFitter.h"

#include "RecoVertex/VertexPrimitives/interface/TransientVertex.h"

#include "TrackingTools/Records/interface/TransientTrackRecord.h"

#include "TrackingTools/TransientTrack/interface/TransientTrack.h"

#include "TrackingTools/TransientTrack/interface/TransientTrackBuilder.h"

#include "TrackingTools/Records/interface/TrackingComponentsRecord.h"

#include "TrackPropagation/SteppingHelixPropagator/interface/SteppingHelixPropagator.h"

#include "TrackPropagation/RungeKutta/interface/RKTestPropagator.h"

#include "TrackingTools/GeomPropagators/interface/SmartPropagator.h"

#include "TrackingTools/GeomPropagators/interface/BeamHaloPropagator.h"

#include "FWCore/Common/interface/TriggerNames.h"

#include "DataFormats/Common/interface/TriggerResults.h"

#include "DataFormats/HLTReco/interface/TriggerObject.h"

#include "DataFormats/HLTReco/interface/TriggerEvent.h"

#include "DataFormats/HLTReco/interface/TriggerTypeDefs.h"

#include "HLTrigger/HLTcore/interface/HLTConfigProvider.h"

#include "SimDataFormats/PileupSummaryInfo/interface/PileupSummaryInfo.h"

#include "FWCore/Framework/interface/Event.h"

#include "DataFormats/EgammaReco/interface/ElectronSeedFwd.h"

#include "DataFormats/EgammaReco/interface/ElectronSeed.h"

#include "DataFormats/EgammaCandidates/interface/GsfElectron.h"

#include "DataFormats/GsfTrackReco/interface/GsfTrack.h"

#include "DataFormats/EgammaReco/interface/BasicCluster.h"

#include "DataFormats/EgammaReco/interface/BasicClusterFwd.h"

#include < memory >

#include "SimDataFormats/GeneratorProducts/interface/LHEEventProduct.h"

#include "SimDataFormats/GeneratorProducts/interface/LHERunInfoProduct.h"

#include "DataFormats/BeamSpot/interface/BeamSpot.h"

// user include files

#include "FWCore/Framework/interface/Frameworkfwd.h"

#include "FWCore/Framework/interface/EDProducer.h"

#include "FWCore/Framework/interface/Event.h"

#include "FWCore/Framework/interface/MakerMacros.h"

```
#include "FWCore/ParameterSet/interface/ParameterSet.h"

#include "SimDataFormats/GeneratorProducts/interface/GenEventInfoProduct.h"

namespace LHAPDF {

void initPDFSet(int nset, const std::string& filename, int member=0);

int numberPDF(int nset);

void usePDFMember(int nset, int member);

double xfx(int nset, double x, double Q, int fl);

double getXmin(int nset, int member);

double getXmax(int nset, int member);

double getQ2min(int nset, int member);

double getQ2max(int nset, int member);

void extrapolate(bool extrapolate=true);

}

std::string pdfSet_; /// lhapdf string

unsigned int Tree_index=0;

double inv_mass_gen,inv_mass_reco;

int event,run,lumi,bxnumber,realdata,primary_vertex;

struct wholething_reco

{ public: double pt_value_reco;

double eta_value_reco;

double phi_value_reco;

double x_mom_reco;

double y_mom_reco;

double z_mom_reco;

double E_energy_reco;

double charge_value_reco;

double ET_RECO;
```

```
double ETA_SC;

double DELTA_ETA_IN;

double DELTA_PHI_IN;

double H_OVER_E;

double SIGMA_IETA_IETA;

double ECAL_ISOLATION;

double HADRONIC_DEPTH_ONE_ISOLATION;

double HADRONIC_DEPTH_TWO_ISOLATION;

double TRACK_PT_ISOLATION;

double E_2_5;

double E_5_5;

double E_1_5;

int IS_ECAL_DRIVEN_;

double NUMBER_OF_LOST_HITS_;

int IS_ENDCAP;

int IS_BARREL;

math::XYZTLorentzVector CHECK_GARNA_KO_LAGI_RECO;

double Recoelectron_dxy;

};

struct wholething_gen

{ public:

double pdgid_value_gen;

double pt_value_gen;

double eta_value_gen;

double phi_value_gen;

double x_mom_gen;

double y_mom_gen;
```

```cpp
double z_mom_gen;

double E_energy_gen;

math::XYZTLorentzVector CHECK_GARNA_KO_LAGI_GEN;

double ET_GEN; };

bool PDF_WORK_GARNE_KI_NO=false;

bool compare_momenta_gen ( const wholething_gen& x, const wholething_gen& y)

{

return (x.pt_value_gen) > (y.pt_value_gen);

}

bool compare_momenta_reco ( const wholething_reco & x, const wholething_reco& y)

{

return (x.pt_value_reco) > (y.pt_value_reco);

}

class TH1F;

class TH2F;

class TStyle;

class TTree;

using namespace edm;

using namespace reco;

using namespace std;

using namespace ROOT::Math::VectorUtil;

using namespace HepMC;

class WAYNE_STATE_CI_DIELECTRON : public edm::EDAnalyzer {

public:

explicit WAYNE_STATE_CI_DIELECTRON(const edm::ParameterSet&);

WAYNE_STATE_CI_DIELECTRON();

private:
```

```
virtual void beginJob();

virtual void analyze(const edm::Event&, const edm::EventSetup&);

virtual void endJob() ;

std::string outputFilename;

edm::InputTag triggerSummaryLabel_;

edm::Handle < double> rho_;

edm::Service < TFileService> fs;

TTree * myTree;

TFile * myFile;

double MC_OR_DATA_;

double PDF_UNCERTAINTY_WEIGHT[PDF_SET_KATI_OTA+1];

double DIELECTRON_MASS_GEN,DIELECTRON_MASS_RECO;

double Recoelectron_px[50], Recoelectron_py[50], Recoelectron_pz[50],Recoelectron_dxy[50];

double Recoelectron_pt[50], Recoelectron_eta[50], Recoelectron_phi[50],Recoelectron_energy[50];

double Recoelectron_charge[50];

double ET_RECO[50],ET_GEN_[50],ETA_SC[50],DELTA_ETA_IN[50],DELTA_PHI_IN[50],H_OVER_E[50],

SIGMA_IETA_IETA[50],ECAL_ISOLATION[50],HADRONIC_DEPTH_ONE_ISOLATION[50],

HADRONIC_DEPTH_TWO_ISOLATION[50],TRACK_PT_ISOLATION[50]; double E_2_5[50];

double E_5_5[50];

double E_1_5[50],NUMBER_OF_LOST_HITS[50];

int IS_ECAL_DRIVEN__[50],IS_ENDCAP__[50],

IS_BARREL__[50];

double STANDARD_DIELECTRON_MASS_RECO;

double STANDARD_DIELECTRON_MASS_GEN;

double RHO;

//Gen

double Genelectron_pt[50],Genelectron_px[50],Genelectron_py[50],Genelectron_pz[50],
```

Genelectron_eta[50],Genelectron_phi[50],Genelectron_charge[50],Genelectron_energy[50];

};

WAYNE_STATE_CI_DIELECTRON::WAYNE_STATE_CI_DIELECTRON(const edm::ParameterSet&

iConfig)

{

pdfSet_=iConfig.getParameter < std::string> ("pdfSet");

PDF_WORK_GARNE_KI_NO=iConfig.getParameter < bool> ("PDF_WORK_GARNE_KI_NO");

if ((pdfSet_ != "")&&(PDF_WORK_GARNE_KI_NO))

{ LHAPDF::initPDFSet(1, pdfSet_.c_str());}

triggerSummaryLabel_ = iConfig.getParameter < edm::InputTag>("triggerSummaryLabel");

MC_OR_DATA_= iConfig.getParameter < double>("MC_OR_DATA");

}

WAYNE_STATE_CI_DIELECTRON:: WAYNE_STATE_CI_DIELECTRON() { }

void WAYNE_STATE_CI_DIELECTRON::analyze(const edm::Event& iEvent, const edm::EventSetup&

iSetup) { for(int jj=0;jj < PDF_SET_KATI_OTA+1;++jj) { PDF_UNCERTAINTY_WEIGHT[jj]=0.0;

}

if(PDF_WORK_GARNE_KI_NO)

{

std::auto_ptr < std::vector < double> > pdf_weights(new std::vector < double>());

if (! iEvent.isRealData() && pdfSet_ !="") {

edm::Handle < GenEventInfoProduct> lamichhane_pl;

if (iEvent.getByLabel("generator", lamichhane_pl)) {

LHAPDF::usePDFMember(1,0);

float q = lamichhane_pl – >pdf() – >scalePDF;

int id1 = lamichhane_pl – >pdf() – >id.first;

double x1 = lamichhane_pl – >pdf() – >x.first;

int id2 = lamichhane_pl – >pdf() – >id.second;

```
double x2 = lamichhane_pl − >pdf() − >x.second;

double xpdf1 = LHAPDF::xfx(1, x1, q, id1);

double xpdf2 = LHAPDF::xfx(1, x2, q, id2);

for(int i=0; i < =PDF_SET_KATI_OTA; ++i){

LHAPDF::usePDFMember(1,i);

float q_ = lamichhane_pl − >pdf() − >scalePDF;

int id1_ = lamichhane_pl − >pdf() − >id.first;

double x1_ = lamichhane_pl − >pdf() − >x.first;

int id2_ = lamichhane_pl − >pdf() − >id.second;

double x2_ = lamichhane_pl − >pdf() − >x.second;

double xpdf1_new = LHAPDF::xfx(1, x1_, q_, id1_);

double xpdf2_new = LHAPDF::xfx(1, x2_, q_, id2_);

double weight = xpdf1_new * xpdf2_new;

pdf_weights − >push_back(weight);

}

}

}

for(size_t plji=0;plji < pdf_weights − >size();plji++)

{

PDF_UNCERTAINTY_WEIGHT[plji]=(*pdf_weights)[plji];

}

}

using namespace edm;

edm::Handle < BeamSpot> beamSpotHandle;

iEvent.getByLabel(InputTag("offlineBeamSpot"),beamSpotHandle);

Handle < GenParticleCollection>gen;

if (MC_OR_DATA_==1)
```

```
{

iEvent.getByLabel( "genParticles",gen);

} wholething_gen record_gen; wholething_reco record_reco;

Handle < reco::GsfElectronCollection>mero_electrons;

iEvent.getByLabel( "gsfElectrons",mero_electrons);


Tree_index=5;

vector < wholething_gen> Information_gen;

vector < wholething_reco> Information_reco;

unsigned int GEN_SEGMENT=0;

unsigned int GEN_NUMBER=1;

unsigned int RECO_SEGMENT=0;

unsigned int RECO_NUMBER=1;

DIELECTRON_MASS_GEN=DIELECTRON_MASS_RECO=0.0;

RHO=0.0;

for(int jj=0;jj < 50;++jj )

{

Genelectron_pt[jj]=0.0; Genelectron_px[jj]=0.0; Genelectron_py[jj]=0.0; Genelectron_pz[jj]=0.0;

Genelectron_eta[jj]=0.0; Genelectron_phi[jj]=0.0; Genelectron_charge[jj]=0.0; Genelectron_energy[jj]=0.0;

Recoelectron_dxy[jj]=0.0; Recoelectron_px[jj]=0.0; Recoelectron_py[jj]=0.0; Recoelectron_pz[jj]=0.0;

Recoelectron_pt[jj]=0.0; Recoelectron_eta[jj]=0.0; Recoelectron_phi[jj]=0.0; ET_RECO[jj]=0.0;

ET_GEN_[jj]=0.0; ETA_SC[jj]=0.0; DELTA_ETA_IN[jj]=0.0; DELTA_PHI_IN[jj]=0.0; H_OVER_E[jj]=0.0;

SIGMA_IETA_IETA[jj]=0.0; ECAL_ISOLATION[jj]=0.0; HADRONIC_DEPTH_ONE_ISOLATION[jj]=0.0;

HADRONIC_DEPTH_TWO_ISOLATION[jj]=0.0; TRACK_PT_ISOLATION[jj]=0.0;

E_2_5[jj]=0.0; E_5_5[jj]=0.0; E_1_5[jj]=0.0; IS_ECAL_DRIVEN__[jj]=0;

Recoelectron_charge[jj]=0.0; NUMBER_OF_LOST_HITS[jj]=20.0; IS_ENDCAP__[jj]=0; IS_BARREL__[jj]=0;

STANDARD_DIELECTRON_MASS_RECO=0.0; STANDARD_DIELECTRON_MASS_GEN=0.0;
```

```
}
```

using namespace edm;

using namespace reco; using namespace std; bool isrho;

isrho = iEvent.getByLabel(edm::InputTag("kt6PFJets:rho"),rho_);

if(isrho) RHO =*rho_; edm::Handle < reco::VertexCollection>PVCollection; bool hasPVs = false;

primary_vertex=0;

if (iEvent.getByLabel(InputTag("offlinePrimaryVertices"), PVCollection)) {

for (reco::VertexCollection::const_iterator pv = PVCollection − >begin(); pv != PVCollection − >end(); ++pv)

{ if (pv − >isFake() || pv − >tracksSize()==0) continue; else { primary_vertex++; hasPVs = true; } } }

Handle < trigger::TriggerEvent> triggerObj;

iEvent.getByLabel(triggerSummaryLabel_,triggerObj);

std::vector < reco::Particle> HLTMuMatched_pl;

for (size_t ia = 0; ia < triggerObj − >sizeFilters(); ++ ia) {

std::string fullname = triggerObj − >filterTag(ia).encode();

std::string name;

size_t p = fullname.find_first_of(':');

if (p != std::string::npos) { name = fullname.substr(0, p);

{cout < < "name of trigger : " < < name < < endl; } } else { name = fullname; } }

if (MC_OR_DATA_==1) { for (reco::GenParticleCollection::const_iterator it = gen − >begin(),ge = gen − >end(); it != ge;++it)

{ if((abs((it) − >pgdId())==11) && ((it) − >status()==1))// &&(abs((it) − >eta()) < eta-cut_gen) &&(abs((it) − >pt())> ptcut_gen))

{

record_gen.pt_value_gen = (it) − >pt(); record_gen.eta_value_gen = (it) − >eta();

```
record_gen.phi_value_gen = (it) − >phi();

record_gen.pdgid_value_gen = (it) − >pdgId();

record_gen.x_mom_gen =(it) − >px(); record_gen.y_mom_gen =(it) − >py();

record_gen.z_mom_gen =(it) − >pz();

record_gen.E_energy_gen =(it) − >energy();

record_gen.CHECK_GARNA_KO_LAGI_GEN=(it) − >p4();

record_gen.ET_GEN=it − >energy()*sin(it − >theta()) ;

Information_gen.push_back(record_gen); } }

} event = iEvent.id().event(); run = iEvent.id().run(); lumi = iEvent.luminosityBlock();

bxnumber = iEvent.bunchCrossing();

realdata = iEvent.isRealData();

if(Information_gen.size()>1)

{ std::sort(Information_gen.begin(),Information_gen.end(),compare_momenta_gen);


math::XYZTLorentzVector v1_no_charge((Information_gen[0].x_mom_gen+Information_gen[1].x_mom_gen),

(Information_gen[0].y_mom_gen+Information_gen[1].y_mom_gen),(Information_gen[0].z_mom_gen

+Information_gen[1].z_mom_gen),

(Information_gen[0].E_energy_gen+Information_gen[1].E_energy_gen)) ;

DIELECTRON_MASS_GEN=v1_no_charge.M();

GEN_SEGMENT=7;

STANDARD_DIELECTRON_MASS_GEN=0.0;


if(GEN_SEGMENT ==7)

{


STANDARD_DIELECTRON_MASS_GEN=(Information_gen[0].CHECK_GARNA_KO_LAGI_GEN

+Information_gen[GEN_NUMBER].CHECK_GARNA_KO_LAGI_GEN).mag();
```

```
for (unsigned int i=0;i < Information_gen.size();i++)
{

Genelectron_pt[i] =(Information_gen[i].pt_value_gen);
Genelectron_px[i] =(Information_gen[i].x_mom_gen);
Genelectron_py[i] =(Information_gen[i].y_mom_gen);
Genelectron_pz[i] =(Information_gen[i].z_mom_gen);
Genelectron_eta[i] =(Information_gen[i].eta_value_gen);
Genelectron_phi[i] =(Information_gen[i].phi_value_gen);

Genelectron_charge[i] =(Information_gen[i].pdgid_value_gen);
Genelectron_energy[i] =(Information_gen[i].E_energy_gen);
ET_GEN_[i] =(Information_gen[i].ET_GEN);
}

}

}
edm::Handle < reco::BeamSpot> theBeamSpot;
iEvent.getByType(theBeamSpot);
edm::Handle < reco::BeamSpot> pBeamSpot;
iEvent.getByLabel("verticesCollection", pBeamSpot);
Handle < reco::VertexCollection> primaryVertexColl;
iEvent.getByLabel("offlinePrimaryVertices",primaryVertexColl);
const reco::VertexCollection* pvcoll = primaryVertexColl.product();
math::XYZPoint firstpvertex(0.,0.,0.);
```

```
if(pvcoll – >size() > 0) {

reco::VertexCollection::const_iterator firstpv = pvcoll – >begin();

firstpvertex.SetXYZ(firstpv – >x(),firstpv – >y(),firstpv – >z());

}


for( reco::GsfElectronCollection::const_iterator pramod=mero_electrons – >begin(); pramod!=mero_electrons
– >end(); ++pramod )

{ static const double mass = 0.000510998928;//0.10566;


record_reco.pt_value_reco = (pramod) – >pt();

record_reco.charge_value_reco = (pramod) – >charge();

record_reco.eta_value_reco = (pramod) – >eta();

record_reco.phi_value_reco = (pramod) – >phi();

record_reco.x_mom_reco = (pramod) – >px();

record_reco.y_mom_reco = (pramod) – >py();

record_reco.z_mom_reco = (pramod) – >pz();

record_reco.E_energy_reco =pramod – >superCluster() – >energy();

record_reco.CHECK_GARNA_KO_LAGI_RECO=pramod – >p4()* (pramod – >caloEnergy() /
pramod – >energy());

record_reco.ET_RECO=pramod – >caloEnergy()*sin(pramod – >p4().theta());

cout < < "ENERGY FROM SUPER CALORIMETER :" < < pramod – >caloEnergy()*sin(pramod
– >p4().theta()) < < endl;

cout < < "ENERGY FROM SUPER CLUSTER :" < < pramod – >superCluster() – >energy()*sin(pramod
– >p4().theta()) < < endl;


record_reco.ETA_SC=pramod – >caloPosition().eta();

record_reco.DELTA_ETA_IN=pramod – >deltaEtaSuperClusterTrackAtVtx();
```

```
record_reco.DELTA_PHI_IN=pramod − >deltaPhiSuperClusterTrackAtVtx();

record_reco.H_OVER_E=pramod − >hadronicOverEm();

record_reco.SIGMA_IETA_IETA=pramod − >sigmaIetaIeta();

record_reco.ECAL_ISOLATION=pramod − >dr03EcalRecHitSumEt();

record_reco.HADRONIC_DEPTH_ONE_ISOLATION=pramod − >dr03HcalDepth1TowerSumEt();

record_reco.HADRONIC_DEPTH_TWO_ISOLATION=pramod − >dr03HcalDepth2TowerSumEt();

record_reco.TRACK_PT_ISOLATION=pramod − >dr03TkSumPt();

record_reco.E_2_5=pramod − >e2x5Max();

record_reco.E_5_5=pramod − >e5x5();

record_reco.E_1_5=pramod − >e1x5();

record_reco.IS_ECAL_DRIVEN_=(pramod − >ecalDriven());

record_reco.IS_BARREL=pramod − >isEB();

record_reco.IS_ENDCAP=pramod − >isEE();

record_reco.Recoelectron_dxy=fabs(pramod − >gsfTrack() − >dxy(firstpvertex));

record_reco.NUMBER_OF_LOST_HITS_=pramod − >gsfTrack()

− >trackerExpectedHitsInner().numberOfLostHits();

Information_reco.push_back(record_reco); }

if((Information_reco.size()>1))

{ std::sort(Information_reco.begin(),Information_reco.end(),compare_momenta_reco); math::XYZTLorentzVector

v2_NO_CHARGE_CHECK(

(Information_reco[0].x_mom_reco+Information_reco[1].x_mom_reco),(Information_reco[0].y_mom_reco

+Information_reco[1].y_mom_reco),(Information_reco[0].z_mom_reco+Information_reco[1].z_mom_reco),

(Information_reco[0].E_energy_reco+Information_reco[1].E_energy_reco) ) ;


DIELECTRON_MASS_RECO=v2_NO_CHARGE_CHECK.M();

RECO_SEGMENT=7;

STANDARD_DIELECTRON_MASS_RECO=0.0;
```

```
if(RECO_SEGMENT ==7)


{ for(unsigned int i=0;i < Information_reco.size();i++)


{
IS_BARREL__[i] =Information_reco[i].IS_BARREL;
IS_ENDCAP__[i] =Information_reco[i].IS_ENDCAP;
Recoelectron_dxy[i] =(Information_reco[i]. Recoelectron_dxy);


Recoelectron_pt[i] =(Information_reco[i].pt_value_reco);
Recoelectron_px[i] =(Information_reco[i].x_mom_reco);
Recoelectron_py[i] =(Information_reco[i].y_mom_reco);
Recoelectron_pz[i] =(Information_reco[i].z_mom_reco);
Recoelectron_eta[i] =(Information_reco[i].eta_value_reco);
Recoelectron_phi[i] =(Information_reco[i].phi_value_reco);
Recoelectron_charge[i] =(Information_reco[i].charge_value_reco);
ET_RECO[i] =(Information_reco[i].ET_RECO);
Recoelectron_energy[i] =(Information_reco[i].E_energy_reco );
IS_ECAL_DRIVEN__[i] =(Information_reco[i].IS_ECAL_DRIVEN_);
ETA_SC[i] =(Information_reco[i].ETA_SC);
DELTA_ETA_IN[i] =(Information_reco[i].DELTA_ETA_IN);
DELTA_PHI_IN[i] =(Information_reco[i].DELTA_PHI_IN);
H_OVER_E[i] =(Information_reco[i].H_OVER_E);
SIGMA_IETA_IETA[i] =(Information_reco[i].SIGMA_IETA_IETA);
ECAL_ISOLATION[i] =(Information_reco[i].ECAL_ISOLATION);
HADRONIC_DEPTH_ONE_ISOLATION[i] =(Information_reco[i].HADRONIC_DEPTH_ONE_ISOLATION);
HADRONIC_DEPTH_TWO_ISOLATION[i] =(Information_reco[i].HADRONIC_DEPTH_TWO_ISOLATION);
```

```
TRACK_PT_ISOLATION[i] =(Information_reco[i].TRACK_PT_ISOLATION);

E_2_5[i] =(Information_reco[i].E_2_5);

E_5_5[i] =(Information_reco[i].E_5_5);

E_1_5[i] =(Information_reco[i].E_1_5);

NUMBER_OF_LOST_HITS[i] =(Information_reco[i].NUMBER_OF_LOST_HITS_); } } }

myTree − >Fill(); }

void WAYNE_STATE_CI_DIELECTRON::beginJob() { TFileDirectory TestDir = fs − >mkdir("demo");

myTree = new TTree("WAYNE_STATE_CI_DIELECTRON","WAYNE_STATE_CI_DIELECTRON_justtitle");

myTree − >Branch("IS_BARREL",IS_BARREL__,"IS_BARREL__[50]/I");

myTree − >Branch("IS_ENDCAP",IS_ENDCAP__,"IS_ENDCAP__[50]/I");

myTree − >Branch("RHO",&RHO,"RHO/D");

myTree − >Branch("E_2_5",E_2_5,"E_2_5[50]/D");

myTree − >Branch("E_5_5",E_5_5,"E_5_5[50]/D");

myTree − >Branch("E_1_5",E_1_5,"E_1_5[50]/D");

myTree − >Branch("IS_ECAL_DRIVEN__",IS_ECAL_DRIVEN__,"IS_ECAL_DRIVEN__[50]/I");

myTree − >Branch("NUMBER_OF_LOST_HITS",NUMBER_OF_LOST_HITS,

"NUMBER_OF_LOST_HITS[50]/D");

myTree − >Branch("STANDARD_DIELECTRON_MASS_RECO",

&STANDARD_DIELECTRON_MASS_RECO,

"STANDARD_DIELECTRON_MASS_RECO/D");

myTree − >Branch("STANDARD_DIELECTRON_MASS_GEN",

&STANDARD_DIELECTRON_MASS_GEN,

"STANDARD_DIELECTRON_MASS_GEN/D");

myTree − >Branch("whaterver", &Tree_index,"Tree_index/I");

myTree − >Branch("PRIMARY_VERTEX", &primary_vertex,"primary_vertex/I");

myTree − >Branch("DIELECTRON_MASS_GEN",&DIELECTRON_MASS_GEN,

"DIELECTRON_MASS_GEN/D");
```

```
myTree − >Branch("DIELECTRON_MASS_RECO",&DIELECTRON_MASS_RECO,
"DIELECTRON_MASS_RECO/D");

myTree − >Branch("ET_RECO",ET_RECO,"ET_RECO[50]/D");

myTree − >Branch("ETA_SC",ETA_SC,"ETA_SC[50]/D");

myTree − >Branch("DELTA_ETA_IN",DELTA_ETA_IN,"DELTA_ETA_IN[50]/D");

myTree − >Branch("DELTA_PHI_IN",DELTA_PHI_IN,"DELTA_PHI_IN[50]/D");

myTree − >Branch("H_OVER_E",H_OVER_E,"H_OVER_E[50]/D");

myTree − >Branch("SIGMA_IETA_IETA",SIGMA_IETA_IETA,"SIGMA_IETA_IETA[50]/D");

myTree − >Branch("ECAL_ISOLATION",ECAL_ISOLATION,"ECAL_ISOLATION[50]/D");

myTree − >Branch("HADRONIC_DEPTH_ONE_ISOLATION",HADRONIC_DEPTH_ONE_ISOLATION,
"HADRONIC_DEPTH_ONE_ISOLATION[50]/D");

myTree − >Branch("HADRONIC_DEPTH_TWO_ISOLATION",HADRONIC_DEPTH_TWO_ISOLATION,
"HADRONIC_DEPTH_TWO_ISOLATION[50]/D");

myTree − >Branch("TRACK_PT_ISOLATION",TRACK_PT_ISOLATION,"TRACK_PT_ISOLATION[50]/D");

myTree − >Branch("EVENT_NUM",&event,"event/I");

myTree − >Branch("RUN_NUM",&run,"run/I");

myTree − >Branch("LUMI_BLOCK",&lumi,"lumi/I");

myTree − >Branch("BXNUMBER",bxnumber,"bxnumber/I");

myTree − >Branch("REAL_DATA",realdata,"realdata/I");

myTree − >Branch("ET_GEN",ET_GEN_,"ET_GEN_[50]/D");

myTree − >Branch("Recoelectron_pt",Recoelectron_pt,"Recoelectron_pt[50]/D");

myTree − >Branch("Recoelectron_px",Recoelectron_px,"Recoelectron_px[50]/D");

myTree − >Branch("Recoelectron_py",Recoelectron_py,"Recoelectron_py[50]/D");

myTree − >Branch("Recoelectron_pz",Recoelectron_pz,"Recoelectron_pz[50]/D");

myTree − >Branch("Recoelectron_eta",Recoelectron_eta,"Recoelectron_eta[50]/D");

myTree − >Branch("Recoelectron_phi",Recoelectron_phi,"Recoelectron_phi[50]/D");

myTree − >Branch("Recoelectron_charge",Recoelectron_charge,"Recoelectron_charge[50]/D");
```

```
myTree – >Branch("Recoelectron_energy",Recoelectron_energy,"Recoelectron_energy[50]/D");

myTree – >Branch("Genelectron_test_pt",Genelectron_pt,"Genelectron_pt[50]/D");

myTree – >Branch("Genelectron_test_px",Genelectron_px,"Genelectron_px[50]/D");

myTree – >Branch("Genelectron_test_py",Genelectron_py,"Genelectron_py[50]/D");

myTree – >Branch("Genelectron_test_pz",Genelectron_pz,"Genelectron_pz[50]/D");

myTree – >Branch("Genelectron_test_eta",Genelectron_eta,"Genelectron_eta[50]/D");

myTree – >Branch("Genelectron_test_phi",Genelectron_phi,"Genelectron_phi[50]/D");

myTree – >Branch("PDF_UNCERTAINTY_WEIGHT",PDF_UNCERTAINTY_WEIGHT,
"PDF_UNCERTAINTY_WEIGHT[101]/D");

myTree – >Branch("Recoelectron_dxy",Recoelectron_dxy,"Recoelectron_dxy[50]/D");

myTree – >Branch("Gen_test_charge",Genelectron_charge,"Genelectron_charge[50]/D");

myTree – >Branch("Gen_test_energy",Genelectron_energy,"Genelectron_energy[50]/D"); }

void WAYNE_STATE_CI_DIELECTRON::endJob()

{ myTree – >Print(); }

DEFINE_FWK_MODULE(WAYNE_STATE_CI_DIELECTRON);
```

APPENDIX B

B.1   C++ program for Ntuple processing

####################

\# Module Name : NTPL_p.cpp

\# Author : Pramod Lamichhane

\# Date : Aug 2013

\# Last modified : Feb 2013

\# Purpose : This program reads the Ntuple prepared by programs in A

\# and generates ROOT files with histograms of various information

\# This module is used by external program B.2 to execute the task

####################

#include "/uscms/home/bikashji/all_c_files/PRAMOD_HEADER_FILE_EL.h"

#include < vector >

#include "TMath.h"

using namespace std;

const float MYTOTAL_LUMI_AILESAMMA=19482.4;

const int USE_MC=0; TString DATA_ROOT_FILE_ADDRESS("/uscms_data/d2/bikashji/all_c_files/");

TString MC_ROOT_FILE_ADDRESS(DATA_ROOT_FILE_ADDRESS);

TString ntp_dir("/uscms_data/d2/bikashji/NTUPLES_VAULT/ALL_DIELECTRON_NTUPLES/");

```
TString NAME_OF_NTUPLE("/WAYNE_STATE_CI_DIELECTRON");
TString DATA_KO_HISTO_NAME("PRIMARY_VERTEX_DISTRIBUTION");
TString MC_KO_HISTO_NAME("Primary_vertex_changed_");
const int PDF_KATI_OTA=45;


//############TRIGGER TURN ON KO STUFF######
bool prashna=false;
float laserTurnOnClusEt(float et);
float noLaserTurnOnClusEt(float et);
float barrelTurnOnClusEt(float et);
float totalClusEtTurnOn(float et,float detEta,float totLumi)
{
if(fabs(detEta) < 1.5) return barrelTurnOnClusEt(et);
else{
float fracPostLaser=(totLumi-325.)/totLumi;
return fracPostLaser*laserTurnOnClusEt(et) + (1-fracPostLaser)*noLaserTurnOnClusEt(et);
}}
float barrelTurnOnClusEt(float et)
{
float maxEff = 0.999;
maxEff=1.;
float midPoint = 34.26;
float turnOn=0.6852;
return 0.5*maxEff*(1+TMath::Erf((et-midPoint)/(sqrt(2)*turnOn)));
}


float laserTurnOnClusEt(float et)
{
```

```
float maxEff = 0.9976;

maxEff=1.;

float midPoint = 33.62;

float turnOn=1.576;

return 0.5*maxEff*(1+TMath::Erf((et-midPoint)/(sqrt(2)*turnOn)));}

float noLaserTurnOnClusEt(float et)

{ float maxEff = 0.9681;

maxEff=1.;

float midPoint = 37.47;

float turnOn=2.323;

return 0.5*maxEff*(1+TMath::Erf((et-midPoint)/(sqrt(2)*turnOn)));}

int pl[13],pl_gen[3],good_gen,good_reco;const double gen_pt_threshold=0.0; const double reco_pt_threshold=0.0;

const double gen_ET_threshold=30.0;

bool GEN_CUT_CHECK(double GEN_ET_[50],double ETA[50],double PDG[50])

{ int GEN_CUT=0;for (int j = 0; j < 2; ++j) {

if ((GEN_ET_[j] > gen_ET_threshold) &&(fabs(ETA[j]) < 2.6)){GEN_CUT++;} }

if((GEN_CUT > 1)&&((PDG[0]*PDG[1])==-121)) {return true;} else{return false;} }

bool RECO_CUT_CHECK( int END[50],int BAR[50],double BARREL_KI_ENDCAP[50],double

ET[50],double PT[50],int ECAL_DRIVEN[50],

double DELTA_ETA_IN[50],double DELTA_PHI_IN[50],double H_OVER_E[50],double SIGMA_IETA_IETA[50],

double E25[50],double E55[50],double E15[50],

double DEPTH_ONE[50],double ECAL_ISO[50],double TRACK_PT[50],double LOST_HIT[50],double

CHARGE[50],double RHO_,double RECOELECTRON__dxy[50])

{ int RECO_CUT=0;

for (int j=0;j < 2;j++)

{

if (BAR[j]==1)
```

```
{
if( (ET[j] > 35) &&
(fabs(BARREL_KI_ENDCAP[j]) < 1.442) &&
(ECAL_DRIVEN[j]==1) &&
(std::abs(DELTA_ETA_IN[j]) < 0.005) &&
(std::abs(DELTA_PHI_IN[j]) < 0.06) &&
(H_OVER_E[j] < 0.05) &&
(((E25[j]/E55[j]) > 0.94) ||((E15[j]/E55[j]) > 0.83) ) &&
((DEPTH_ONE[j])+(ECAL_ISO[j])) < (2+0.03*ET[j]+0.28*RHO_) &&
(TRACK_PT[j] < 5) &&
(LOST_HIT[j] < =1) &&
(std::abs(RECOELECTRON_dxy[j]) < 0.02)
)
{ RECO_CUT++;}
}
if(END[j]==1)
{ bool had_one_cut=false;
double had_cut=0.0;

if (ET[j] < 50){had_one_cut=true;}
if(!had_one_cut){had_cut=2.5+0.03*(ET[j]-50)+0.28*RHO_;}
else{had_cut=2.5+0.28*RHO_;}

if(((fabs(BARREL_KI_ENDCAP[j]) > 1.560)&&(fabs(BARREL_KI_ENDCAP[j]) < 2.5)) &&
(ET[j] > 35) &&
// (PT[j] > reco_pt_threshold) &&
(ECAL_DRIVEN[j]==1) &&
(std::abs(DELTA_ETA_IN[j]) < 0.007) &&
```

```
(std::abs(DELTA_PHI_IN[j]) < 0.06) &&

(H_OVER_E[j] < 0.05) &&

(SIGMA_IETA_IETA[j] < 0.03) &&

//((E25_OVER_E55[j] > 0.94) ||(E15_OVER_E55[j] > 0.83) ) &&

((DEPTH_ONE[j]+(ECAL_ISO[j])) < had_cut) &&

(TRACK_PT[j] < 5) &&

(LOST_HIT[j] < =1) &&

(std::abs(RECOELECTRON_dxy[j]) < 0.05)

)


{RECO_CUT++;}



}


}


bool barrel=false;

for (int a=0;a < 2;a++){if(BAR[a]==1){barrel=true;}}

if((RECO_CUT > 1)&&(1)&&(barrel))

{ return true; } else {return false;}

}

bool GEN_CUT_CHECK_(double GEN_ET_[50],double ETA[50],double PDG[50])

{ int gCUT1=0; int gCUT2=0; int gCUT3=0; int gCUT4=0;


for (int j = 0; j < 2; ++j)

{

if ((GEN_ET_[j]) < =gen_ET_threshold) continue; gCUT1++;
```

```
if (fabs(ETA[j]) > = 2.6) continue; gCUT2++;

}


if((PDG[0]*PDG[1]==-121)){ gCUT3++;
//cout << "pdg values:" << PDG[0] << "" << PDG[1] << endl;
}


if( gCUT1 > 1){pl_gen[0]++;}
if( gCUT2 > 1){pl_gen[1]++;}
if((gCUT1 > 1)&&(gCUT2 > 1)&&( gCUT3==1)){pl_gen[2]++;}


return true;


}



bool RECO_CUT_CHECK_( int END[50],int BAR[50],double BARREL_KI_ENDCAP[50],double
ET[50],double PT[50],int ECAL_DRIVEN[50],
double DELTA_ETA_IN[50],double DELTA_PHI_IN[50],double H_OVER_E[50],
double SIGMA_IETA_IETA[50],double E25[50],double E55[50],double E15[50],
double DEPTH_ONE[50],double ECAL_ISO[50],double TRACK_PT[50],double LOST_HIT[50],double
CHARGE[50],double RHO_)
{


int rCUT1=0; int rCUT2=0;int rCUT3=0;int rCUT4=0;int rCUT5=0;int rCUT6=0;int rCUT7=0;int
rCUT8=0;int rCUT9=0;int
rCUT10=0;int rCUT11=0;int rCUT12=0;int rCUT13=0;
```

```
for (int j=0;j < 2;j++)
{
if(BAR[j]==1)


{
if (ET[j] < =35) continue; rCUT1++;
if (PT[j] < =reco_pt_threshold) continue; rCUT2++;
if (ECAL_DRIVEN[j]!=1) continue; rCUT3++;
if (DELTA_ETA_IN[j] > =0.005) continue; rCUT4++;
if (DELTA_PHI_IN[j] > =0.06) continue; rCUT5++;
if ((H_OVER_E[j]) > =0.05) continue; rCUT6++;rCUT7++;
if (((E25[j]/E55[j]) < =0.94) &&((E15[j]/E55[j]) < =0.83) ) continue; rCUT8++;
if (((DEPTH_ONE[j])+(ECAL_ISO[j])) > =(2+0.03*ET[j]+0.28*RHO_)) continue; rCUT9++;
if (TRACK_PT[j] > =5) continue; rCUT10++;
if (!(LOST_HIT[j] > 1)) continue; rCUT11++;
if ((CHARGE[0]*CHARGE[1])!=-1) continue; rCUT12++;


if(! (fabs(BARREL_KI_ENDCAP[j]) < 1.442)) continue; rCUT13++;



}


else if(END[j]==1)

{ bool had_one_cut=false;
double had_cut=0.0;
```

```
if (ET[j] < 50){had_one_cut=true;}
if(!had_one_cut){had_cut=2.5+0.03*(ET[j]-50)+0.28*RHO_;}
else{had_cut=2.5+0.28*RHO_;}


if (ET[j] < =40) continue; rCUT1++;
if (PT[j] < =reco_pt_threshold) continue; rCUT2++;
if (ECAL_DRIVEN[j]!=1) continue; rCUT3++;
if (DELTA_ETA_IN[j] > =0.007) continue; rCUT4++;
if (DELTA_PHI_IN[j] > =0.06) continue; rCUT5++;
if (H_OVER_E[j] > =0.05) continue; rCUT6++;
if (SIGMA_IETA_IETA[j] > =0.03) continue; rCUT7++;
rCUT8++;
if ((DEPTH_ONE[j]+(ECAL_ISO[j])) > =had_cut) continue; rCUT9++;
if (TRACK_PT[j] > =5) continue; rCUT10++;
if (!(LOST_HIT[j] > 1)) continue; rCUT11++;
if ((CHARGE[0]*CHARGE[1])!=-1) continue; rCUT12++;


if(!((fabs(BARREL_KI_ENDCAP[j]) > 1.560)&&(fabs(BARREL_KI_ENDCAP[j]) < 2.5)))continue;
rCUT13++;
} }



if(rCUT1 > 1) {++pl[0]; }
if(rCUT2 > 1) {++pl[1]; }
if(rCUT3 > 1) {++pl[2]; }
if(rCUT4 > 1) {++pl[3]; }
```

```
if(rCUT5 > 1) {++pl[4]; }

if(rCUT6 > 1) {++pl[5]; }

if(rCUT7 > 1) {++pl[6]; }

if(rCUT8 > 1) {++pl[7]; }

if(rCUT9 > 1) {++pl[8]; }

if(rCUT10 > 1) {++pl[9]; }

if(rCUT11 > 1) {++pl[10];}

if(rCUT12 > 1) {++pl[11];}

if(rCUT13 > 1) {++pl[12];}

return true;


}
void UPDATED_PRASHNA_DATA(TString TRIGGER_KI_HOINA,TString KUN_DATASET ,TString
which_data_histo,TString which_mc_histo)
{
std::vector < int > event_;
std::vector < double > mass_;
std::vector < int > RUN_;
if(USE_MC==1) { cout << "REMINDER MC IS BEING PROCESSED USE_MC VALUE IS 1
NOW, CHANGE TO 0 FOR DATA" << endl; }
if(USE_MC==0) { cout << "REMINDER DATA IS BEING PROCESSED USE_MC VALUE IS
0 NOW, CHANGE TO 1 FOR MC" << endl; }
int pramod1,pramod2,pramod3;
std::string string_common="migration_bin_";std::string W_string_common="W_migration_bin_";
TH1F* DiELECTRON_reco; TH1F* W_DiELECTRON_reco; TH1F* DiELECTRON__gen; TH1F*
W_DiELECTRON__gen;
map < string,TH1F* > nam;map < string,TH1F* > W_nam;
for (int a=1;a < 22;a++)
```

```
{
std::ostringstream temporary; temporary << string_common << a;
nam[temporary.str().c_str()]=new TH1F(temporary.str().c_str(),temporary.str().c_str(),400,0.0,4000.0);
}
map < string, TH1F* > ::const_iterator iter;
for (int a=1;a < 22;a++)
{
std::ostringstream temporary;
temporary << W_string_common << a;
W_nam[temporary.str().c_str()]=new TH1F(temporary.str().c_str(),temporary.str().c_str(),400,0.0,4000.0);
}
for (int a=1;a < 22;a++)
{
std::ostringstream temporary;
temporary << W_string_common << a;
W_nam[temporary.str().c_str()]-> Sumw2(); }


DiELECTRON__reco =new TH1F("DiELECTRON__reco","DiELECTRON__reco",400,0,4000);
W_DiELECTRON__reco =new TH1F("W_DiELECTRON__reco","W_DiELECTRON__reco",400,0,4000);
DiELECTRON__gen =new TH1F("DiELECTRON__gen","DiELECTRON__gen",400,0,4000);
W_DiELECTRON__gen =new TH1F("W_DiELECTRON__gen","DiELECTRON__gen",400,0,4000);
W_DiELECTRON__reco_2 =new TH1F("DiELECTRON__reco_2","DiELECTRON__reco_2",60,0,120);
Reco_pt =new TH1F("Reco_pt","Reco_pt",200,0,500);
Reco_px =new TH1F("Reco_px","Reco_px",200,0,500);
Reco_py =new TH1F("Reco_py","Reco_py",200,0,500);
Reco_pz =new TH1F("Reco_pz","Reco_pz",200,0,500);
Reco_eta =new TH1F("Reco_eta","Reco_eta",200,-5,5);
Reco_phi =new TH1F("Reco_phi","Reco_phi",160,-5,5);
```

```
Gen_pt =new TH1F("Gen_pt","Gen_pt",200,0,500);

Gen_px =new TH1F("Gen_px","Gen_px",200,0,500);

Gen_py =new TH1F("Gen_py","Gen_py",200,0,500);

Gen_pz =new TH1F("Gen_pz","Gen_pz",200,0,500);

Gen_eta =new TH1F("Gen_eta","Gen_eta",200,-5,5);

Gen_phi =new TH1F("Gen_phi","Gen_phi",160,-5,5);

Reco_DELTA_ETA =new TH1F("Reco_DELTA_ETA","Reco_DELTA_ETA",2000,-5.0,5.0);

Reco_DELTA_PHI =new TH1F("Reco_DELTA_PHI","Reco_DELTA_PHI",2000,-5.0,5.0);

Reco_H_OVER_E =new TH1F("Reco_H_OVER_E","Reco_H_OVER_E",2000,0,1.0);

Reco_SIGMA_IETA_IETA =new TH1F("Reco_SIGMA_IETA_IETA","Reco_SIGMA_IETA_IETA",2000,0,1.0);

had_1iso =new TH1F("HADRONIC_DEPTH_ONE_ISOLATION",

"HADRONIC_DEPTH_ONE_ISOLATION",2000,0,100);

had_2iso =new TH1F("had_2iso",

"had_2iso",2000,0,100);

Reco_TRACK_PT_ISOLATION =new TH1F("Reco_TRACK_PT_ISOLATION",

"Reco_TRACK_PT_ISOLATION",2000,0,150);

Reco_ETA_SC =new TH1F("Reco_ETA_SC","Reco_ETA_SC",200,-5,5);

Reco_E_2_5 =new TH1F("Reco_E_2_5","Reco_E_2_5",2000,0,50);

Reco_E_5_5 =new TH1F("Reco_E_5_5","Reco_E_5_5",2000,0,50);

Reco_E_1_5 =new TH1F("Reco_E_1_5","Reco_E_1_5",2000,0,50);

Reco_LOST_LAYER_HIT =new TH1F("NO_OF_LOST_LAYER","NO_OF_LOST_LAYER",2000,0,50);

W_DiELECTRON_reco->Sumw2();W_DiELECTRON_gen- > Sumw2();

TChain myTree(TRIGGER_KI_HOINA+NAME_OF_NTUPLE);

cout << TRIGGER_KI_HOINA+NAME_OF_NTUPLE << endl;

cout << ntp_dir+KUN_DATASET+"/res/*.root" << endl;

myTree.Add(ntp_dir+KUN_DATASET+"/res/*.root");

//myTree.Print();

PDF_ERR_PLUS =new TH1D("Dimuon_reco_PLUS","Dimuon_reco_PLUS",400,0,4000);
```

```
PDF_ERR_CENTRAL =new TH1D("Dimuon_reco_CENTRAL","Dimuon_reco_CENTRAL",400,0,4000);

PDF_ERR_MINUS =new TH1D("Dimuon_reco_MINUS","Dimuon_reco_MINUS",400,0,4000);

PDF_ERR_PLUS– >Sumw2();PDF_ERR_CENTRAL– > Sumw2();PDF_ERR_MINUS– > Sumw2();

TH1D* PRIMARY_VERTEX_DISTRIBUTION;

PRIMARY_VERTEX_DISTRIBUTION =new TH1D("PRIMARY_VERTEX_DISTRIBUTION",

"PRIMARY_VERTEX_DISTRIBUTION",60,0,60);

int Tree_index;

int IS_ENDCAP[50],IS_BARREL[50];

int PRIMARY_VERTEX,EVENT_NUM,RUN_NUM,LUMI_BLOCK,BXNUMBER,REAL_DATA;

double DIELECTRON_MASS_GEN,DIELECTRON_MASS_RECO,E_2_5[50],E_5_5[50],E_1_5[50];

int IS_ECAL_DRIVEN[50];

double NUMBER_OF_LOST_HITS[50],ELECTRON_ET_RECO[50],

ELECTRON_ETA_SC[50],ELECTRON_DELTA_ETA_IN[50],ELECTRON_DELTA_PHI_IN[50]

,ELECTRON_H_OVER_E[50], ELECTRON_SIGMA_IETA_IETA[50],ELECTRON_ECAL_ISOLATION[50],

ELECTRON_HADRONIC_DEPTH_ONE_ISOLATION[50],

ELECTRON_HADRONIC_DEPTH_TWO_ISOLATION[50],

ELECTRON_TRACK_PT_ISOLATION[50],

RECO_ELECTRON__pt[50],RECO_ELECTRON__px[50],

RECO_ELECTRON__py[50],RECO_ELECTRON__pz[50],RECO_ELECTRON__eta[50]

,RECO_ELECTRON__phi[50],RECO_ELECTRON__energy[50],

RECO_ELECTRON__charge[50],GEN_ELECTRON__pt[50],

GEN_ELECTRON__px[50],GEN_ELECTRON__py[50],GEN_ELECTRON__pz[50]

,GEN_ELECTRON__eta[50],GEN_ELECTRON__phi[50],GEN_ELECTRON__pdgid[50]

,GEN_ELECTRON__energy[50];

double data=0.0;double mc=0.0;

double MY_ELE_GEN_ET[50];

double STANDARD_DIELECTRON_MASS_RECO;

double STANDARD_DIELECTRON_MASS_GEN;
```

```
double RHO;

double pdfun[PDF_KATI_OTA];

double RECO_ELECTRON__dxy[50];

myTree.SetBranchAddress("STANDARD_DIELECTRON_MASS_RECO",

&STANDARD_DIELECTRON_MASS_RECO);

myTree.SetBranchAddress("STANDARD_DIELECTRON_MASS_GEN"

, &STANDARD_DIELECTRON_MASS_GEN);

myTree.SetBranchAddress("whaterver", &Tree_index);

myTree.SetBranchAddress("IS_ENDCAP",IS_ENDCAP);

myTree.SetBranchAddress("IS_BARREL",IS_BARREL);

myTree.SetBranchAddress("RHO",&RHO);

myTree.SetBranchAddress("PRIMARY_VERTEX",&PRIMARY_VERTEX);

myTree.SetBranchAddress("EVENT_NUM",&EVENT_NUM);

myTree.SetBranchAddress("RUN_NUM",&RUN_NUM);

myTree.SetBranchAddress("LUMI_BLOCK",&LUMI_BLOCK);

myTree.SetBranchAddress("BXNUMBER",&BXNUMBER);

myTree.SetBranchAddress("REAL_DATA",&REAL_DATA);

myTree.SetBranchAddress("DIELECTRON_MASS_GEN"

,&DIELECTRON_MASS_GEN);

myTree.SetBranchAddress("DIELECTRON_MASS_RECO",&DIELECTRON_MASS_RECO);

myTree.SetBranchAddress("pdfun"

,pdfun);

myTree.SetBranchAddress("E_2_5",E_2_5);

myTree.SetBranchAddress("E_5_5",E_5_5);

myTree.SetBranchAddress("E_1_5",E_1_5);

myTree.SetBranchAddress("IS_ECAL_DRIVEN_",IS_ECAL_DRIVEN);

myTree.SetBranchAddress("NUMBER_OF_LOST_HITS"

,NUMBER_OF_LOST_HITS);
```

```
myTree.SetBranchAddress("ET_RECO",ELECTRON_ET_RECO);

myTree.SetBranchAddress("ET_GEN",MY_ELE_GEN_ET);

myTree.SetBranchAddress("ETA_SC",ELECTRON_ETA_SC);

myTree.SetBranchAddress("DELTA_ETA_IN",ELECTRON_DELTA_ETA_IN);

myTree.SetBranchAddress("DELTA_PHI_IN"

,ELECTRON_DELTA_PHI_IN);

myTree.SetBranchAddress("H_OVER_E",ELECTRON_H_OVER_E);

myTree.SetBranchAddress("SIGMA_IETA_IETA"

,ELECTRON_SIGMA_IETA_IETA);

myTree.SetBranchAddress("ECAL_ISOLATION",ELECTRON_ECAL_ISOLATION);

myTree.SetBranchAddress("HADRONIC_DEPTH_ONE_ISOLATION"

,ELECTRON_HADRONIC_DEPTH_ONE_ISOLATION);

myTree.SetBranchAddress("HADRONIC_DEPTH_TWO_ISOLATION"

,ELECTRON_HADRONIC_DEPTH_TWO_ISOLATION);

myTree.SetBranchAddress("TRACK_PT_ISOLATION"

,ELECTRON_TRACK_PT_ISOLATION);

myTree.SetBranchAddress("Recoelectron_pt",RECO_ELECTRON__pt);

myTree.SetBranchAddress("Recoelectron_px",RECO_ELECTRON__px);

myTree.SetBranchAddress("Recoelectron_py",RECO_ELECTRON__py);

myTree.SetBranchAddress("Recoelectron_pz",RECO_ELECTRON__pz);

myTree.SetBranchAddress("Recoelectron_eta",RECO_ELECTRON__eta);

myTree.SetBranchAddress("Recoelectron_phi",RECO_ELECTRON__phi);

myTree.SetBranchAddress("Recoelectron_energy",RECO_ELECTRON__energy);

myTree.SetBranchAddress("Recoelectron_charge",RECO_ELECTRON__charge);

myTree.SetBranchAddress("Recoelectron_dxy",RECO_ELECTRON__dxy);

myTree.SetBranchAddress("Genelectron_test_pt",GEN_ELECTRON__pt);

myTree.SetBranchAddress("Genelectron_test_px",GEN_ELECTRON__px);

myTree.SetBranchAddress("Genelectron_test_py",GEN_ELECTRON__py);
```

```
myTree.SetBranchAddress("Genelectron_test_pz",GEN_ELECTRON__pz);

myTree.SetBranchAddress("Genelectron_test_eta",GEN_ELECTRON__eta);

myTree.SetBranchAddress("Genelectron_test_phi",GEN_ELECTRON__phi);

myTree.SetBranchAddress("Gen_test_charge",GEN_ELECTRON__pdgid);

myTree.SetBranchAddress("Gen_test_energy",GEN_ELECTRON__energy);

Int_t nevent = myTree.GetEntries();

int vertex_value=0;

cout << "TOTAL EVENTS:" << nevent << endl;

good_gen=good_reco=0; double pramod_gen,pramod_reco;

pramod_gen=pramod_reco=0.0;

TFile f(TRIGGER_KI_HOINA+"_"+KUN_DATASET+".root","recreate");

for (Int_t iev=0;iev < nevent;iev++) //nevent

{

myTree.GetEntry(iev);

if(DIELECTRON_MASS_GEN > 0.0){good_gen++;}

if(DIELECTRON_MASS_RECO > 0.0){good_reco++;}

double Gen_px_0,Gen_py_0,Gen_pz_0,Gen_E_0,Gen_px_1,Gen_py_1,Gen_pz_1,Gen_E_1,px_0,py_0,pz_0

,E_0,px_1,py_1,pz_1,E_1,Gen_diELECTRON__mass,Reco_diELECTRON__mass;

Gen_px_0=Gen_py_0=Gen_pz_0=Gen_E_0=Gen_px_1=Gen_py_1=Gen_pz_1=Gen_E_1=px_0=py_0=pz_0

=E_0=px_1=py_1=pz_1=E_1=Gen_diELECTRON__mass=Reco_diELECTRON__mass=0.0;

data=mc=0.0; double weight_factor=0;

bool CUT_OF_ISOLATION_REQUIREMENT_GEN=GEN_CUT_CHECK(MY_ELE_GEN_ET,

GEN_ELECTRON__eta,GEN_ELECTRON__pdgid);

bool CUT_OF_ISOLATION_REQUIREMENT_RECO=RECO_CUT_CHECK(

IS_ENDCAP,IS_BARREL,ELECTRON_ETA_SC,ELECTRON_ET_RECO,RECO_ELECTRON__pt,

IS_ECAL_DRIVEN,ELECTRON_DELTA_ETA_IN,

ELECTRON_DELTA_PHI_IN,ELECTRON_H_OVER_E,ELECTRON_SIGMA_IETA_IETA,E_2_5,

E_5_5,E_1_5,ELECTRON_HADRONIC_DEPTH_ONE_ISOLATION,
```

```
ELECTRON_ECAL_ISOLATION,ELECTRON_TRACK_PT_ISOLATION,NUMBER_OF_LOST_HITS,

RECO_ELECTRON__charge,RHO,RECO_ELECTRON__dxy);

GEN_CUT_CHECK_(MY_ELE_GEN_ET,GEN_ELECTRON__eta,GEN_ELECTRON__pdgid);

RECO_CUT_CHECK_(IS_ENDCAP,IS_BARREL,ELECTRON_ETA_SC

,ELECTRON_ET_RECO,RECO_ELECTRON__pt,

IS_ECAL_DRIVEN,ELECTRON_DELTA_ETA_IN,

ELECTRON_DELTA_PHI_IN,ELECTRON_H_OVER_E,

ELECTRON_SIGMA_IETA_IETA,E_2_5,E_5_5,E_1_5,

ELECTRON_HADRONIC_DEPTH_ONE_ISOLATION,ELECTRON_ECAL_ISOLATION,

ELECTRON_TRACK_PT_ISOLATION,NUMBER_OF_LOST_HITS,RECO_ELECTRON__charge,RHO);

if(CUT_OF_ISOLATION_REQUIREMENT_GEN)


{ pramod_gen++;


Gen_px_0 = GEN_ELECTRON__px[0];

Gen_py_0 = GEN_ELECTRON__py[0];

Gen_pz_0 = GEN_ELECTRON__pz[0];

Gen_E_0 = GEN_ELECTRON__energy[0];


Gen_px_1 = GEN_ELECTRON__px[1];

Gen_py_1 = GEN_ELECTRON__py[1];

Gen_pz_1 = GEN_ELECTRON__pz[1];

Gen_E_1 = GEN_ELECTRON__energy[1];

TLorentzVector v3((Gen_px_0+Gen_px_1),(Gen_py_0+Gen_py_1),(Gen_pz_0+Gen_pz_1),(Gen_E_0+Gen_E_1));

Gen_diELECTRON__mass=v3.M();

if(CUT_OF_ISOLATION_REQUIREMENT_GEN){Gen_diELECTRON__mass

=STANDARD_DIELECTRON_MASS_GEN;}

if(Gen_diELECTRON__mass > 0.0)
```

```
{

DiELECTRON__gen- > Fill(Gen_diELECTRON__mass);

W_DiELECTRON__gen - > Fill(Gen_diELECTRON__mass,weight_factor);

for(int i=0;i < 2;++i)

{

Gen_pt -> Fill(GEN_ELECTRON__pt[i]);

Gen_px -> Fill(GEN_ELECTRON__px[i]);

Gen_py -> Fill(GEN_ELECTRON__py[i]);

Gen_pz -> Fill(GEN_ELECTRON__pz[i]);

Gen_eta -> Fill(GEN_ELECTRON__eta[i]);

Gen_phi -> Fill(GEN_ELECTRON__phi[i]);

} }}

if(CUT_OF_ISOLATION_REQUIREMENT_RECO)

{pramod_reco++;


px_0 = RECO_ELECTRON__px[0];

py_0 = RECO_ELECTRON__py[0];

pz_0 = RECO_ELECTRON__pz[0];

E_0 = RECO_ELECTRON__energy[0];

px_1 = RECO_ELECTRON__px[1];

py_1 = RECO_ELECTRON__py[1];

pz_1 = RECO_ELECTRON__pz[1];

E_1 = RECO_ELECTRON__energy[1];



TLorentzVector v2((px_0+px_1),(py_0+py_1),(pz_0+pz_1),(E_0+E_1));

Reco_diELECTRON__mass=v2.M();

Reco_diELECTRON__mass=STANDARD_DIELECTRON_MASS_RECO;
```

```
}


if (CUT_OF_ISOLATION_REQUIREMENT_RECO)

{

float W1=totalClusEtTurnOn(ELECTRON_ET_RECO[0],ELECTRON_ETA_SC[0],

MYTOTAL_LUMI_AILESAMMA);

float W2=totalClusEtTurnOn(ELECTRON_ET_RECO[1],ELECTRON_ETA_SC[1],

MYTOTAL_LUMI_AILESAMMA);

float weight_factor_=W1*W2;

if(USE_MC==1)//means monte carlo using and trig turn on ko weight halnuparchha


{

DiELECTRON_reco-> Fill(Reco_diELECTRON_mass);

W_DiELECTRON_reco-> Fill(Reco_diELECTRON_mass,weight_factor_);

W_DiELECTRON_reco_2-> Fill(Reco_diELECTRON_mass,weight_factor_);

}

if(USE_MC==0)//means data, no weight is required


{

string pramod_lamichhane_string("");

string pa,pb,pc,pd;

if (RECO_ELECTRON_charge[0]==1) pa="+";

if (RECO_ELECTRON_charge[1]==1) pb="+";

if (RECO_ELECTRON_charge[0]==-1) pa="-";

if (RECO_ELECTRON_charge[1]==-1) pb="-";


pramod_lamichhane_string="["+pa+" , "+pb+"]";
```

```
if((Reco_diELECTRON__mass > 700.))

{


cout << "EVENT" << EVENT_NUM << "||" << "RUN" << RUN_NUM << "||" << "LUMIX"
<< LUMI_BLOCK << "||" << "MASS" << Reco_diELECTRON__mass << "" << pramod_lamichhane_string
<< endl;



}


DiELECTRON__reco-> Fill(Reco_diELECTRON__mass);

W_DiELECTRON__reco-> Fill(Reco_diELECTRON__mass);

W_DiELECTRON__reco_2-> Fill(Reco_diELECTRON__mass);

}
for(int i=0;i < 2;++i)

{

Reco_phi -> Fill(RECO_ELECTRON__phi[i]);

Reco_pt-> Fill(RECO_ELECTRON__pt[i]);

Reco_px-> Fill(RECO_ELECTRON__px[i]);

Reco_py-> Fill(RECO_ELECTRON__py[i]);

Reco_pz-> Fill(RECO_ELECTRON__pz[i]);

Reco_eta-> Fill(RECO_ELECTRON__eta[i]);

Reco_ETA_SC-> Fill(ELECTRON_ETA_SC[i]);

Reco_DELTA_ETA-> Fill(ELECTRON_DELTA_ETA_IN[i]);

Reco_DELTA_PHI-> Fill(ELECTRON_DELTA_PHI_IN[i]);

Reco_H_OVER_E-> Fill(ELECTRON_H_OVER_E[i]);

Reco_SIGMA_IETA_IETA-> Fill(ELECTRON_SIGMA_IETA_IETA[i]);
```

```
had_1iso-> Fill(ELECTRON_HADRONIC_DEPTH_ONE_ISOLATION[i]);

had_2iso-> Fill(ELECTRON_HADRONIC_DEPTH_TWO_ISOLATION[i]);

Reco_TRACK_PT_ISOLATION-> Fill(ELECTRON_TRACK_PT_ISOLATION[i]);

Reco_E_2_5-> Fill(E_2_5[i]);

Reco_E_5_5-> Fill(E_5_5[i]);

Reco_E_1_5-> Fill(E_1_5[i]);

Reco_LOST_LAYER_HIT-> Fill(NUMBER_OF_LOST_HITS[i]);

}

PRIMARY_VERTEX_DISTRIBUTION-> Fill(PRIMARY_VERTEX);

if(Reco_diELECTRON__mass > 60){pramod1++;}

if(Reco_diELECTRON__mass > 120){pramod2++;}

if(Reco_diELECTRON__mass > 200){pramod3++;}

if((Reco_diELECTRON__mass > 700)||(prashna))

{

event_.push_back(EVENT_NUM);

mass_.push_back(Reco_diELECTRON__mass);

RUN_.push_back(RUN_NUM);

}

for(int g=0;g < 20;g++)

{

std::ostringstream temporary;

temporary << string_common << g+1;

if((Gen_diELECTRON__mass > g*100)&&(Gen_diELECTRON__mass < =(g+1)*100))

{nam[temporary.str().c_str()]-> Fill(Reco_diELECTRON__mass);}

}

int g=20;

std::ostringstream temporary;

temporary << string_common << g+1;
```

```cpp
if((Gen_diELECTRON_mass > 2000)&&(Gen_diELECTRON_mass
< =3000))
{nam[temporary.str().c_str()]-> Fill(Reco_diELECTRON_mass);}
for(int g_n=0;g_n < 20;g_n++)
{
std::ostringstream temporary;
temporary << W_string_common << g_n+1;
if((Gen_diELECTRON_mass > g_n*100)
&&(Gen_diELECTRON_mass < =(g_n+1)*100))
{W_nam[temporary.str().c_str()]-> Fill(Reco_diELECTRON_mass,weight_factor);}
}
int g_n=20;
std::ostringstream temporary__;
temporary__ << W_string_common << g_n+1;

if((Gen_diELECTRON_mass > 2000)&&(Gen_diELECTRON_mass < =3000))
{W_nam[temporary__.str().c_str()]-> Fill(Reco_diELECTRON_mass,weight_factor);}
int size_val =sizeof(pdfun)/sizeof(double);
double PLUS_TOT_SQR=0.0;
double MINUS_TOT_SQR=0.0;
for( int j = 1; j < size_val ; j+=2) {
double WPLUS=0.0; double WPLUS_SQR=0.0 ; double WMINUS=0.0;
double WMINUS_SQR=0.0 ; WPLUS = max((pdfun[j]-pdfun[0]),(pdfun[j+1]-pdfun[0])) ;
WPLUS = max(WPLUS, 0.0);
WMINUS = max((pdfun[0]-pdfun[j]),(pdfun[0]-pdfun[j+1])) ;
WMINUS = max(WMINUS, 0.0);
WPLUS_SQR=pow(WPLUS,2); WMINUS_SQR=pow(WMINUS,2); PLUS_TOT_SQR+=WPLUS_SQR;
MINUS_TOT_SQR+=WMINUS_SQR; }
```

```
if(PLUS_TOT_SQR > 0)PLUS_TOT_SQR=sqrt(PLUS_TOT_SQR);

if(MINUS_TOT_SQR > 0)MINUS_TOT_SQR=sqrt(MINUS_TOT_SQR);

PDF_ERR_PLUS-> Fill(Reco_diELECTRON_mass,1+PLUS_TOT_SQR);

PDF_ERR_CENTRAL-> Fill(Reco_diELECTRON_mass,pdfun[0]);

PDF_ERR_MINUS-> Fill(Reco_diELECTRON_mass,1-MINUS_TOT_SQR); } }

PDF_ERR_PLUS-> Write(); PDF_ERR_CENTRAL-> Write();PDF_ERR_MINUS-> Write(); PRI-
MARY_VERTEX_DISTRIBUTION-> Write();

DiELECTRON_reco-> Write();W_DiELECTRON_reco-> Write();DiELECTRON__gen-> Write();

W_DiELECTRON__gen-> Write();

Reco_pt-> Write(); Reco_px-> Write(); Reco_py-> Write(); Reco_pz-> Write();

Reco_eta-> Write(); Reco_phi-> Write(); Gen_pt-> Write(); Gen_px-> Write();

Gen_py-> Write(); Gen_pz-> Write(); Gen_eta-> Write(); Gen_phi-> Write();

Reco_DELTA_ETA-> Write(); Reco_DELTA_PHI-> Write(); Reco_H_OVER_E-> Write();

Reco_SIGMA_IETA_IETA-> Write(); had_1iso-> Write();

had_2iso-> Write(); Reco_TRACK_PT_ISOLATION-> Write();

Reco_ETA_SC-> Write(); Reco_E_2_5-> Write(); Reco_E_5_5-> Write(); Reco_E_1_5-> Write();

delete DiELECTRON_reco;delete W_DiELECTRON_reco;delete DiELECTRON__gen;

delete W_DiELECTRON__gen;

delete PDF_ERR_PLUS;delete PDF_ERR_CENTRAL;delete PDF_ERR_MINUS;

delete PRIMARY_VERTEX_DISTRIBUTION;

delete Reco_pt; delete Reco_px; delete Reco_py; delete Reco_pz;

delete Reco_eta; delete Reco_phi; delete Gen_pt; delete Gen_px;

delete Gen_py; delete Gen_pz;delete Gen_eta;delete Gen_phi;

delete Reco_DELTA_ETA; delete Reco_DELTA_PHI; delete Reco_H_OVER_E;

delete Reco_SIGMA_IETA_IETA; delete had_1iso;

delete had_2iso; delete Reco_TRACK_PT_ISOLATION;

delete Reco_ETA_SC; delete Reco_E_2_5; delete Reco_E_5_5;

delete Reco_E_1_5;
```

{delete p–> R_second;}W_nam.clear();

B.2   C++ program for Ntuple processing

####################

\# Module Name : process_all.cpp

\# Author : Pramod Lamichhane

\# Date : Aug 2013

\# Last modified : Feb 2013

\# Purpose : This program reads the Ntuple prepared by programs in A

\# and generates ROOT files with histograms of various information

\# This module uses the program B.1

####################

#include "NTPL_p.cpp"

#include<iostream>

#include<vector>

#include<string>

#include <fstream>

#include "TMath.h"

#include "vector"

#include<TString.h>

#include <sstream>

#include <iostream>

#include<vector>

#include<string>

#include <fstream>

#include "TMath.h"

```cpp
#include "vector"
#include<TString.h>
#include <sstream>
int process_all_dyci()
{
using namespace std;
std::vector<std::string>* pramod_dir_names;
std::vector<std::string>pramod_dir_name;
pramod_dir_names=&pramod_dir_name;
TString string_common("CIToEE_");
int mass_cut[3]={300,500,800};
string option[2]={"Des","Con"};
int lambda_values[6]={9,11,13,15,17,19};
for(size_t i=0;i< sizeof(mass_cut)/sizeof(int);i++)
{ std::ostringstream temporary;
temporary<<string_common<<"ITCM5_M-"<<mass_cut[i]<<"_TuneZ2star";
cout<<temporary.str().c_str()<<endl;
pramod_dir_names->push_back(temporary.str().c_str());
for(size_t j=0;j< sizeof(option)/sizeof(string);j++)
{
for(size_t k=0;k< sizeof(lambda_values)/sizeof(int);k++)
{
std::ostringstream temporary1;
if((lambda_values[k]>15 && option[j]=="Des"))
{cout<<endl;}
else
{temporary1<<string_common<<option[j]<<"_"<<"Lambda-"<<lambda_values[k]<<"_M-"<<mass_cut[i];
```

```
cout<<temporary1.str().c_str()<<endl;

pramod_dir_names->push_back(temporary1.str().c_str());

} } }


for (size_t i=0;i<pramod_dir_names->size();i++)

{NTPL_p("TRIGGER_EVENT",(*pramod_dir_names)[i],"VERTEX_DATA_2012","VERTEX_MC_2012");}

return 0;

}
```

APPENDIX C

C.1 Python module to chop histograms and combine to variable histograms

C.1.1 Rebinning for data

####################

Module Name : REBIN_DATA.py

Author : Pramod Lamichhane

Date : Aug 2013

Last modified : Feb 2013

Purpose : This program is used to rebin the combined data histogram for a given variable bin with differential information

####################

```
import os
from ROOT import *
from array import array
ROOT_=TFile("TOTAL_ELECTRON_SAME_SIGN_19P6166_ABS_RHO_FIXED.root")
h1=ROOT_.Get("DiELECTRON_reco")
```

my_array=[ ]

for i in range(1,31):

    my_array.append(float(i))

REST_bins=[660,700,740,800,900,1000,1300,1600,2000]

for i in range(0,len(REST_bins)):

    my_array.append(float(i))

h1 = TH1F("h1","h1",len(my_array),0.,2000.)

VAR_LO_=array("f",my_array)

hnew = h1.Rebin(len(my_array)-1,"hnew",VAR_LO_)

hnew.Draw()

C.1.2　For cumulative distribution

####################

Module Name : CUMU.py

Author : Pramod Lamichhane

Date : Aug 2013

Last modified : Feb 2013

Purpose : This program is used to produce cumulative histograms using differential MC and DATA histograms

####################

import sys, os

from array import array

from ROOT import *

from ROOT import TH1F

import string

import math

```
def cumu(type_,ROOT_NAME,dine_name,kun,THAPNE):

    HAMRO_AFNAI_NAME=TFile(ROOT_NAME)

    h=HAMRO_AFNAI_NAME.Get(dine_name)

    nb = h.GetNbinsX()

    hc = TH1F(h.GetName()+'_NEW',h.GetTitle()+

'_NEW',nb, h.GetXaxis().GetXmin(), h.GetXaxis().GetXmax())

    G = TFile ('CUMU_BIN_'+THAPNE+ROOT_NAME,"recreate")

    hc.Sumw2()

    if type_ == 'ge':

        first, last, step = nb+1, 0, -1

    elif type_ == 'le':

        first, last, step = 0, nb+1, 1

    else:

        raise ValueError('type      for i in xrange(first, last, step):

        prev = 0 if i == first else hc.GetBinContent(i-step)

        c = h.GetBinContent(i) + prev

        hc.Fill(i*kun-1,c)

    hc.Write()

    G.Close()

    return 1
cumu('ge','ttbar_SCALED.root','SCALED_Dimuon_reco',10,'')
cumu('ge','tw_SCALED.root','SCALED_Dimuon_reco',10,'')
cumu('ge','tbarw_SCALED.root','SCALED_Dimuon_reco',10,'')
cumu('ge','ww_SCALED.root','SCALED_Dimuon_reco',10,'')
cumu('ge','wz_SCALED.root','SCALED_Dimuon_reco',10,'')
cumu('ge','zz_SCALED.root','SCALED_Dimuon_reco',10,'')
cumu('ge','wjets_SCALED.root','SCALED_Dimuon_reco',10,'')
cumu('ge','dytautau_SCALED.root','SCALED_Dimuon_reco',10,'')
```

```
cumu('ge','LAMBDA-9.root','SCALED_Dimuon_reco',10,'')

cumu('ge','LAMBDA-11.root','SCALED_Dimuon_reco',10,'')

cumu('ge','LAMBDA-13.root','SCALED_Dimuon_reco',10,'')

cumu('ge','LAMBDA-15.root','SCALED_Dimuon_reco',10,'')

cumu('ge','LAMBDA-17.root','SCALED_Dimuon_reco',10,'')

cumu('ge','LAMBDA-19.root','SCALED_Dimuon_reco',10,'')

cumu('ge','LAMBDA-9.root','SCALED_Dimuon_reco',10,'')

cumu('ge','LAMBDA-11.root','SCALED_Dimuon_reco',10,'')

cumu('ge','LAMBDA-13.root','SCALED_Dimuon_reco',10,'')

cumu('ge','LAMBDA-15.root','SCALED_Dimuon_reco',10,'')

cumu('ge','DRELL_YAN.root','SCALED_Dimuon_reco',10,'')

cumu('ge','LAMBDA-INFINITY.root','SCALED_Dimuon_reco',10,'')

cumu('ge','DATA.root','Dimuon_reco',10,'')
```

C.1.3   Chopping and rescaling for MC

```
####################
# Module Name : CHOP.py
# Author : Pramod Lamichhane
# Date : Aug 2013
# Last modified : Feb 2013
# Purpose : This program is used to rescale the histograms with mass dependent k_factors by
# chopping into pieces and prepare a combined histogram which is taken from the relevent
# sample of interest to optimize the statistics
####################
```

```python
import string

import math

import time

LUMI_TO_SCALE_TO=19616.0

NNLO_FACTOR=1.024

QCD_k_factor=[1.2962,1.2992,1.3029,1.2970,1.3071,1.2979,1.3033,1.2902,

1.2845,1.2870,1.2678,1.2730,1.2622,1.2727,1.2481,1.2497,1.2543,1.2590]

ACCEPTANCE=[1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.]

#QED_k_factor=[1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.]

#QED_k_factor=[0.997, 0.993, 0.989, 0.985, 0.981, 0.976, 0.972, 0.968,

0.964, 0.960 ,0.955, 0.951, 0.947, 0.943, 0.939, 0.934, 0.930, 0.926]

QED_k_factor=[ ]

for i in range(0,len(QCD_k_factor)):QED_k_factor.append(1)

LOCATION='/uscms/home/bikashji/'

def CHOPPING(KUN_ROOT_FILE,CROSS_SECTION,EVENTS,FILTER_EFF):

    print KUN_ROOT_FILE

    KUN_HISTO_KO_BANAUNE='W_DiELECTRON__reco'

    SUPPLIED_VALUE_BEGINNING, SUPPLIED_VALUE_END=0,0

    if '300' in KUN_ROOT_FILE:

        SUPPLIED_VALUE_BEGINNING=31

    SUPPLIED_VALUE_END=61

    if '500' in KUN_ROOT_FILE:

        SUPPLIED_VALUE_BEGINNING=61

        SUPPLIED_VALUE_END=101

    if '800' in KUN_ROOT_FILE:

        SUPPLIED_VALUE_BEGINNING=101

        SUPPLIED_VALUE_END=402
```

```
scale_thing_ORIGINAL=(LUMI_TO_SCALE_TO/(EVENTS/(FILTER_EFF
CROSS_SECTION)))*NNLO_FACTOR
    scale_thing=scale_thing_ORIGINAL


    SUPPLIED_ROOT_FILE=TFile(LOCATION+'TRIGGER_EVENT_'+KUN_ROOT_FILE+'.root')
    h=SUPPLIED_ROOT_FILE.Get(KUN_HISTO_KO_BANAUNE)
    nb =h.GetNbinsX()
    bin_width=(h.GetXaxis().GetXmax()-h.GetXaxis().GetXmin())/nb
    MY_NAME=h.GetName()+'_'+str(int((SUPPLIED_VALUE_BEGINNING-1)*bin_width))+
    '_'+str(int((SUPPLIED_VALUE_END-1)*bin_width))
    hc=TH1F('ele_reco','ele_reco',
    nb,h.GetXaxis().GetXmin(),h.GetXaxis().GetXmax())
    kun=h.GetName().split('_')
    print MY_NAME,MY_NAME,nb,h.GetXaxis().GetXmin(),h.GetXaxis().GetXmax()
    G =TFile (KUN_ROOT_FILE+'_SCALED.root',"recreate")


    print KUN_ROOT_FILE.strip('.root')+kun[1]+'_SCALED.root'
    for u in range(SUPPLIED_VALUE_BEGINNING,SUPPLIED_VALUE_END):
    scale_thing=scale_thing_ORIGINAL
    if u*bin_width>300 and u*bin_width<=400:
        scale_thing=scale_thing*QCD_k_factor[0]*QED_k_factor[0]*ACCEPTANCE[0]


    if u*bin_width>400 and u*bin_width<=500:
        scale_thing=scale_thing*QCD_k_factor[1]*QED_k_factor[1]*ACCEPTANCE[1]


    if u*bin_width>500 and u*bin_width<=600:
        scale_thing=scale_thing*QCD_k_factor[2]*QED_k_factor[2]*ACCEPTANCE[2]
```

```
if u*bin_width>600 and u*bin_width<=700:

    scale_thing=scale_thing*QCD_k_factor[3]*QED_k_factor[3]*ACCEPTANCE[3]


if u*bin_width>700 and u*bin_width<=800:

    scale_thing=scale_thing*QCD_k_factor[4]*QED_k_factor[4]*ACCEPTANCE[4]


if u*bin_width>800 and u*bin_width<=900:

    scale_thing=scale_thing*QCD_k_factor[5]*QED_k_factor[5]*ACCEPTANCE[5]


if u*bin_width>900 and u*bin_width<=1000:

    scale_thing=scale_thing*QCD_k_factor[6]*QED_k_factor[6]*ACCEPTANCE[6]


if u*bin_width>1000 and u*bin_width<=1100:

    scale_thing=scale_thing*QCD_k_factor[7]*QED_k_factor[7]*ACCEPTANCE[7]


if u*bin_width>1100 and u*bin_width<=1200:

    scale_thing=scale_thing*QCD_k_factor[8]*QED_k_factor[8]*ACCEPTANCE[8]


if u*bin_width>1200 and u*bin_width<=1300:

    scale_thing=scale_thing*QCD_k_factor[9]*QED_k_factor[9]*ACCEPTANCE[9]


if u*bin_width>1300 and u*bin_width<=1400:

    scale_thing=scale_thing*QCD_k_factor[10]*QED_k_factor[10]*ACCEPTANCE[10]


if u*bin_width>1400 and u*bin_width<=1500:

    scale_thing=scale_thing*QCD_k_factor[11]*QED_k_factor[11]*ACCEPTANCE[11]


if u*bin_width>1500 and u*bin_width<=1600:
```

```
scale_thing=scale_thing*QCD_k_factor[12]*QED_k_factor[12]*ACCEPTANCE[12]


if u*bin_width>1600 and u*bin_width<=1700:

    scale_thing=scale_thing*QCD_k_factor[13]*QED_k_factor[13]*ACCEPTANCE[13]
if u*bin_width>1700 and u*bin_width<=1800:

    scale_thing=scale_thing*QCD_k_factor[14]*QED_k_factor[14]*ACCEPTANCE[14]
if u*bin_width>1800 and u*bin_width<=1900:

    scale_thing=scale_thing*QCD_k_factor[15]*QED_k_factor[15]*ACCEPTANCE[15]


if u*bin_width>1900 and u*bin_width<=2000:

    scale_thing=scale_thing*QCD_k_factor[16]*QED_k_factor[16]*ACCEPTANCE[16]


if u*bin_width>2000 and u*bin_width<=5000:

    scale_thing=scale_thing*QCD_k_factor[17]*QED_k_factor[17]*ACCEPTANCE[17]
hc.Fill((u*bin_width)-1,h.GetBinContent(u)*scale_thing)
hc.Write()
G.Close()
CHOPPING('CIToEE_ITCM5_M-300_TuneZ2star', 0.2621 , 55751 , 0.778 )
CHOPPING('CIToEE_ITCM5_M-500_TuneZ2star', 0.03562 , 26056 , 0.886 )
CHOPPING('CIToEE_ITCM5_M-800_TuneZ2star', 0.004503 , 25672 , 0.929 )
CHOPPING('CIToEE_Con_Lambda-9_M-300', 0.3041 , 57300 , 0.76 )
CHOPPING('CIToEE_Con_Lambda-9_M-500', 0.05713 , 26797 , 0.866 )
CHOPPING('CIToEE_Con_Lambda-9_M-800', 0.01445 , 26111 , 0.918 )
CHOPPING('CIToEE_Con_Lambda-11_M-300', 0.2852 , 55922 , 0.778 )
CHOPPING('CIToEE_Con_Lambda-11_M-500', 0.04739 , 25322 , 0.918 )
CHOPPING('CIToEE_Con_Lambda-11_M-800', 0.009645 , 28215 , 0.849 )
CHOPPING('CIToEE_Con_Lambda-13_M-300', 0.2761 , 56620 , 0.764 )
CHOPPING('CIToEE_Con_Lambda-13_M-500', 0.04289 , 26748 , 0.868 )
```

CHOPPING('CIToEE_Con_Lambda-13_M-800', 0.007542 , 26514 , 0.9)

CHOPPING('CIToEE_Con_Lambda-15_M-300', 0.2722 , 55469 , 0.782)

CHOPPING('CIToEE_Con_Lambda-15_M-500', 0.04034 , 27087 , 0.856)

CHOPPING('CIToEE_Con_Lambda-15_M-800', 0.00648 , 26689 , 0.897)

CHOPPING('CIToEE_Con_Lambda-17_M-300', 0.2693 , 55000 , 0.79)

CHOPPING('CIToEE_Con_Lambda-17_M-500', 0.03915 , 26595 , 0.87)

CHOPPING('CIToEE_Con_Lambda-17_M-800', 0.005877 , 25983 , 0.919)

CHOPPING('CIToEE_Con_Lambda-19_M-300', 0.2676 , 54189 , 0.799)

CHOPPING('CIToEE_Con_Lambda-19_M-500', 0.03821 , 26380 , 0.875)

CHOPPING('CIToEE_Con_Lambda-19_M-800', 0.005518 , 26022 , 0.916)

CHOPPING('CIToEE_Des_Lambda-9_M-300', 0.2555 , 55269 , 0.791)

CHOPPING('CIToEE_Des_Lambda-9_M-500', 0.03886 , 26553 , 0.882)

CHOPPING('CIToEE_Des_Lambda-9_M-800', 0.008843 , 26328 , 0.917)

CHOPPING('CIToEE_Des_Lambda-11_M-300', 0.2536 , 56049 , 0.778)

CHOPPING('CIToEE_Des_Lambda-11_M-500', 0.03516 , 26246 , 0.891)

CHOPPING('CIToEE_Des_Lambda-11_M-800', 0.005808 , 25814 , 0.933)

CHOPPING('CIToEE_Des_Lambda-13_M-300', 0.2542 , 55248 , 0.789)

CHOPPING('CIToEE_Des_Lambda-13_M-500', 0.03418 , 33874 , 0.685)

CHOPPING('CIToEE_Des_Lambda-13_M-800', 0.004786 , 26239 , 0.916)

CHOPPING('CIToEE_Des_Lambda-15_M-300', 0.2554 , 56123 , 0.774)

CHOPPING('CIToEE_Des_Lambda-15_M-500', 0.0338 , 26563 , 0.871)

CHOPPING('CIToEE_Des_Lambda-15_M-800', 0.004403 , 26297 , 0.91)

C.1.4 For variable histograms

```
####################
# Module Name : VARIABLE.py
# Author : Pramod Lamichhane
# Date : Aug 2013
# Last modified : Feb 2013
# Purpose : This program is used to combine histograms prepared by C.1.3 and prepare
# a variable binned histogram
####################

#!/usr/bin/env python
location ='/uscms/home/bikashji/'
import sys, os
from array import array
from ROOT import *
from ROOT import TH1F
import string
import math
VAR_LO=[ ]
for i in range(1,31):
    VAR_LO.append(float(i*20))
VAR_LO.append(float (620))
VAR_LO.append(float (660))
VAR_LO.append(float (700))
VAR_LO.append(float (740))
VAR_LO.append(float (800))
VAR_LO.append(float (900))
```

```
VAR_LO.append(float (1000))

VAR_LO.append(float (1300))

VAR_LO.append(float (1600))

VAR_LO.append(float (2000))

VAR_LO_=array('f',VAR_LO)

def VAR_H(KUN_ROOT_FILE,KUN_HISTO_KO_BANAUNE,THAPNE):

    SUPPLIED_ROOT_FILE=TFile(location+KUN_ROOT_FILE)

    h=SUPPLIED_ROOT_FILE.Get(KUN_HISTO_KO_BANAUNE)

    nb =h.GetNbinsX()

    number_of_bins=h.GetNbinsX()

    bin_width_cal=(h.GetXaxis().GetXmax()-h.GetXaxis().GetXmin())/number_of_bins

    kun=bin_width_cal

    print VAR_LO

    print nb

    print 'bin width',kun

    print h.GetName(),h.GetTitle()

    hc =TH1F(h.GetName()+'_NEW',h.GetTitle()+

    '_NEW',len(VAR_LO)-1,VAR_LO_)

    G =TFile ('VAR_BIN_'+THAPNE+KUN_ROOT_FILE,"recreate")

    #R hc.Sumw2()

    for i in range(1,nb):

        c=h.GetBinContent(i)

        if not THAPNE=='FOR_POISSON':

            if i*kun-1<=620 :

                hc.Fill(i*kun-1,c)

            if (i*kun-1>620) and (i*kun-1<=660):

                hc.Fill(i*kun-1,c/2 )

            if (i*kun-1>660) and (i*kun-1<=700):
```

```
        hc.Fill(i*kun-1,c/2)
      if (i*kun-1>700) and (i*kun-1<=740):
        hc.Fill(i*kun-1,c/2)
      if (i*kun-1>740) and (i*kun-1<=800):
        hc.Fill(i*kun-1,c/3)
      if (i*kun-1>800) and (i*kun-1<=900):
        hc.Fill(i*kun-1,c/5)
      if (i*kun-1>900) and (i*kun-1<=1000):
        hc.Fill(i*kun-1,c/5)
      if (i*kun-1>1000) and (i*kun-1<=1300):
        hc.Fill(i*kun-1,c/15)
      if (i*kun-1>1300) and (i*kun-1<=1600):
        hc.Fill(i*kun-1,c/15)
      if (i*kun-1>1600) and (i*kun-1<=2000):
        hc.Fill(i*kun-1,c/20)
    if THAPNE=='FOR_POISSON':
        hc.Fill(i*kun-1,c)
  hc.Write()
  G.Close()
  return hc


VAR_H('LAMBDA-INFINITY.root','ele_reco','')
VAR_H('LAMBDA-9.root','ele_reco','')
VAR_H('LAMBDA-11.root','ele_reco','')
VAR_H('LAMBDA-13.root','ele_reco','')
VAR_H('LAMBDA-15.root','ele_reco','')
VAR_H('LAMBDA-17.root','ele_reco','')
VAR_H('LAMBDA-19.root','ele_reco','')
```

VAR_H('LAMBDA-9.root','ele_reco','')

VAR_H('LAMBDA-11.root','ele_reco','')

VAR_H('LAMBDA-13.root','ele_reco','')

VAR_H('LAMBDA-15.root','ele_reco','')

VAR_H('data.root','DiELECTRON_reco','')

VAR_H('DY.root','ele_reco','')

VAR_H('dytautau_SCALED.root','ele_reco','')

VAR_H('ttbar_SCALED.root','ele_reco','')

VAR_H('tw_SCALED.root','ele_reco','')

VAR_H('tbarw_SCALED.root','ele_reco','')

VAR_H('ww_SCALED.root','ele_reco','')

VAR_H('wz_SCALED.root','ele_reco','')

VAR_H('zz_SCALED.root','ele_reco','')

VAR_H('wjets_SCALED.root','ele_reco','')

VAR_H('G15_30_SCALED.root','ele_reco','')

VAR_H('G30_50_SCALED.root','ele_reco','')

VAR_H('G50_80_SCALED.root','ele_reco','')

VAR_H('G80_120_SCALED.root','ele_reco','')

VAR_H('G120_170_SCALED.root','ele_reco','')

VAR_H('G170_300_SCALED.root','ele_reco','')

VAR_H('G300_470_SCALED.root','ele_reco','')

VAR_H('G470_800_SCALED.root','ele_reco','')

VAR_H('G800_1400_SCALED.root','ele_reco','')

VAR_H('G1400_1800_SCALED.root','ele_reco','')

VAR_H('G1800_SCALED.root','ele_reco','')

VAR_H('main_jet_1heep_1gsfmethod.root','JET_BKG','')

VAR_H('total_jet_root_2gsf_method_new.root','JET_BKG','')

C.1.5 Root script to fit the CI cross-sections

##################

Module Name : fit_CI_cross_section.C

Author : Pramod Lamichhane

Date : Aug 2013

Last modified : Feb 2013

Purpose : This program uses the function "$a + \frac{b}{\Lambda^2} + \frac{c}{\Lambda^4}$"

to fit the CI cross-sections that are calculated by D.4

####################

Double_t myFitFunction(Double_t *x, Double_t *par)

{

Double_t contact= par[0]+par[1]/(x[0]*x[0])+par[2]/(x[0]*x[0]*x[0]*x[0]);

return (contact);

}

void fit_CI_cross_section()

{

TCanvas *C4 = new TCanvas("C4","");

C4–>cd();

gStyle–>SetOptFit(1);

gStyle–>SetOptStat(0);

gStyle–>SetOptTitle(0);

C4–>SetFillColor(0);

C4–>SetBorderMode(0);

C4–>SetBorderSize(2);

C4–>SetTickx(1);

```
C4->SetTicky(1);

C4->SetLeftMargin(0.19);

C4->SetRightMargin(0.11);

C4->SetTopMargin(0.15);

C4->SetBottomMargin(0.15);

C4->SetFrameFillStyle(0);

C4->SetFrameBorderMode(0);

C4->SetFrameFillStyle(0);

C4->SetFrameBorderMode(0);

TFile *fc500=new TFile("c500.root");

TH1D *hisc500=fc500->Get("c500MY_HISTO");

TFile *fd500=new TFile("d500.root");

TH1D *hisd500=fd500->Get("d500MY_HISTO");

TFile *fc600=new TFile("c600.root");

TH1D *hisc600=fc600->Get("c600MY_HISTO");

TFile *fd600=new TFile("d600.root");

TH1D *hisd600=fd600->Get("d600MY_HISTO");

TFile *fc700=new TFile("c700.root");

TH1D *hisc700=fc700->Get("c700MY_HISTO");

TFile *fd700=new TFile("d700.root");

TH1D *hisd700=fd700->Get("d700MY_HISTO");

TFile *fc800=new TFile("c800.root");

TH1D *hisc800=fc800->Get("c800MY_HISTO");

TFile *fd800=new TFile("d800.root");

TH1D *hisd800=fd800->Get("d800MY_HISTO");

TFile *fc900=new TFile("c900.root");

TH1D *hisc900=fc900->Get("c900MY_HISTO");

TFile *fd900=new TFile("d900.root");
```

```
TH1D *hisd900=fd900–>Get("d900MY_HISTO");

TF1 *fSpectrumc500 = new TF1("fSpectrumc500",myFitFunction,0.,100.,3);

fSpectrumc500–>SetParameters(367.62,45,10);

fSpectrumc500–>SetLineColor(3);

hisc500–>Fit("fSpectrumc500");

hisc500–>GetXaxis()–>SetRangeUser(7,30);

hisc500–>GetYaxis()–>SetRangeUser(0,1000);

TF1 *fSpectrumd500 = new TF1("fSpectrumd500",myFitFunction,0.,100.,3);

fSpectrumd500–>SetParameters(367.62,45,10);

fSpectrumd500–>SetLineColor(3);

fSpectrumd500–>SetLineStyle(8);

fSpectrumd500–>SetMarkerStyle(20);

hisd500–>Fit("fSpectrumd500");

hisd500–>GetXaxis()–>SetRangeUser(5,30);

hisd500–>GetYaxis()–>SetRangeUser(0,3000);

TF1 *fSpectrumc600 = new TF1("fSpectrumc600",myFitFunction,0.,100.,3);

fSpectrumc600–>SetParameters(213.66,45,10);

fSpectrumc600–>SetLineColor(2);

hisc600–>Fit("fSpectrumc600");

hisc600–>GetXaxis()–>SetRangeUser(5,30);

hisc600–>GetYaxis()–>SetRangeUser(0,3000);

TF1 *fSpectrumd600 = new TF1("fSpectrumd600",myFitFunction,0.,100.,3);

fSpectrumd600–>SetParameters(213.66,45,10);

fSpectrumd600–>SetLineColor(2);

fSpectrumd600–>SetLineStyle(8);

hisd600–>Fit("fSpectrumd600");

hisd600–>GetXaxis()–>SetRangeUser(5,30);

hisd600–>GetYaxis()–>SetRangeUser(0,3000);
```

```
TF1 *fSpectrumc700 = new TF1("fSpectrumc700",myFitFunction,0.,100.,3);

fSpectrumc700->SetParameters(2216.06,45,10);

fSpectrumc700->SetLineColor(6);

hisc700->Fit("fSpectrumc700");

hisc700->GetXaxis()->SetRangeUser(5,30);

hisc700->GetYaxis()->SetRangeUser(0,3000);

TF1 *fSpectrumd700 = new TF1("fSpectrumd700",myFitFunction,0.,100.,3);

fSpectrumd700->SetParameters(111.6,45,10);

fSpectrumd700->SetLineColor(6);

fSpectrumd700->SetLineStyle(8);

hisd700->Fit("fSpectrumd700");

hisd700->GetXaxis()->SetRangeUser(5,30);

hisd700->GetYaxis()->SetRangeUser(0,3000);

TF1 *fSpectrumc800 = new TF1("fSpectrumc800",myFitFunction,0.,100.,3);

fSpectrumc800->SetParameters(61.32,45,10);

fSpectrumc800->SetLineColor(7);

hisc800->Fit("fSpectrumc800");

hisc800->GetXaxis()->SetRangeUser(5,30);

hisc800->GetYaxis()->SetRangeUser(0,3000);

TF1 *fSpectrumd800 = new TF1("fSpectrumc800",myFitFunction,0.,100.,3);

fSpectrumd800->SetParameters(61.32,45,10);

fSpectrumd800->SetLineColor(7);

fSpectrumd800->SetLineStyle(8);

hisd800->Fit("fSpectrumc800");

hisd800->GetXaxis()->SetRangeUser(5,30);

hisd800->GetYaxis()->SetRangeUser(0,3000);

TF1 *fSpectrumc900 = new TF1("fSpectrumc900",myFitFunction,0.,100.,3);

fSpectrumc900->SetParameters(34.39,45,10);
```

```
fSpectrumc900–>SetLineColor(4);

hisc900–>Fit("fSpectrumc900");

hisc900–>GetXaxis()–>SetRangeUser(7,30);

hisc900–>GetYaxis()–>SetRangeUser(0,3000);

TF1 *fSpectrumd900 = new TF1("fSpectrumc900",myFitFunction,0.,100.,3);

fSpectrumd900–>SetParameters(34.39,45,10);

fSpectrumd900–>SetLineColor(4);

fSpectrumd900–>SetLineStyle(8);

hisd900–>Fit("fSpectrumc900");

hisd900–>GetXaxis()–>SetRangeUser(7,30);

hisd900–>GetYaxis()–>SetRangeUser(0,975);

hisc500–>SetMarkerStyle(20);

hisc500–>SetMarkerColor(3);

hisc500–>SetLineColor(3);

//hisc500–>GetXaxis()–>SetRangeUser(300,3000);

//hisc500–>GetYaxis()–>SetRangeUser(0.01,1e5);

hisc500–>GetXaxis()–>SetNdivisions(511);

hisc500–>GetXaxis()–>SetMoreLogLabels(10);

hisc500–>GetXaxis()–>SetMoreLogLabels();

//hisc500–>GetXaxis()–>SetNdivisions(511);

hisc500–>GetYaxis()–>SetNdivisions(505);

hisc500–>GetXaxis()–>SetLabelFont(42);

hisc500–>GetXaxis()–>SetLabelSize(0.047935);

hisc500–>GetXaxis()–>SetTitleSize(0.06);

hisc500–>GetXaxis()–>SetTitleOffset(1.025);

hisc500–>GetXaxis()–>SetTitleFont(42);

hisc500–>GetYaxis()–>SetTitle("Event Yield in e{+}e{-} ");

hisc500–>GetYaxis()–>SetLabelFont(42);
```

```
hisc500->GetYaxis()->SetLabelSize(0.047535);

hisc500->GetYaxis()->SetTitleSize(0.06);

hisc500->GetYaxis()->SetTitleFont(42);

hisc500->SetXTitle("#Lambda (TeV)");

gStyle->SetOptStat(0);

gStyle->SetOptFit(0000);

hisc500->Draw("P");

hisd500->SetMarkerStyle(20);

hisd500->SetMarkerColor(3);

hisd500->SetLineColor(3);

hisd500->SetLineStyle(8);

hisd500->Draw("PSAME");

hisc600->SetMarkerStyle(20);

hisc600->SetMarkerColor(2);

hisc600->SetLineColor(2);

hisc600->Draw("PSAME");

hisd600->SetLineStyle(8);

hisd600->SetMarkerStyle(20);

hisd600->SetMarkerColor(2);

hisd600->SetLineColor(2);

hisd600->Draw("PSAME");

hisc700->SetLineColor(6);

hisc700->SetMarkerStyle(20);

hisc700->SetMarkerColor(6);

hisc700->Draw("PSAME");

hisd700->SetLineStyle(8);

hisd700->SetLineColor(6);

hisd700->SetMarkerStyle(20);
```

```
hisd700->SetMarkerColor(6);

hisd700->Draw("PSAME");

hisc800->SetLineColor(7);

hisc800->SetMarkerStyle(20);

hisc800->SetMarkerColor(7);

hisc800->Draw("PSAME");

hisd800->SetLineColor(7);

hisd800->SetLineStyle(8);

hisd800->SetMarkerStyle(20);

hisd800->SetMarkerColor(7);

hisd800->Draw("PSAME");

hisc900->SetLineColor(4);

hisc900->SetMarkerStyle(20);

hisc900->SetMarkerColor(4);

hisc900->Draw("PSAME");

hisd900->SetLineColor(4);

hisd900->SetLineStyle(8);

hisd900->SetMarkerStyle(20);

hisd900->SetMarkerColor(4);

hisd900->Draw("PSAME");

TLegend *leg = new TLegend(0.6149425,0.4367089,0.941092,0.8396624,NULL,"brNDC");

leg->SetBorderSize(1);

leg->SetTextFont(62);

leg->SetTextSize(0.04);

leg->SetLineColor(0);

leg->SetLineStyle(1);

leg->SetLineWidth(1);

leg->SetFillColor(0);
```

```
leg->SetFillStyle(1001);

leg->SetTextSize(0.03);

leg->AddEntry(hisc500,"500 GeV (const.)","L");

leg->AddEntry(hisd500,"500 GeV (destr.)","L");

leg->AddEntry(hisc600,"600 GeV (const.)","L");

leg->AddEntry(hisd600,"600 GeV (destr.)","L");

leg->AddEntry(hisc700,"700 GeV (const.)","L");

leg->AddEntry(hisd700,"700 GeV (destr.)","L");

leg->AddEntry(hisc800,"800 GeV (const.)","L");

leg->AddEntry(hisd800,"800 GeV (destr.)","L");

leg->AddEntry(hisc900,"900 GeV (const.)","L");

leg->AddEntry(hisd900,"900 GeV (destr.)","L");

leg->Draw("nopave");

C4->Update();

gPad->RedrawAxis();

TPaveText *pt = new TPaveText(0.3204023,0.8544304,0.5862069,0.9198312,"brNDC");

pt->SetFillColor(0);

pt->SetLineColor(0);

pt->SetTextAlign(12);

pt->SetTextSize(0.04);

TText *text = pt->AddText("CMS Simulation, #sqrt{s} = 8 TeV, 20 fb^{-1}");

pt->Draw("nopave");

TPaveText *pt1 = new TPaveText(0.3318966,0.5949367,0.5977011,0.6603376,"brNDC");

pt1->SetFillColor(0);

pt1->SetLineColor(0);

pt1->SetTextAlign(12);

pt1->SetTextSize(0.04);

text = pt1->AddText("Fit : a + #frac{b}{#Lambda^{2}} + #frac{c}{#Lambda^{4}}");
```

```
pt1–>Draw("no pave");
```

}

C.1.6 ROOT script of the DATA-MC comparison plot

```
####################
# Module Name : Spectra.C
# Author : Pramod Lamichhane
# Date : Aug 2013
# Last modified : Feb 2013
# Purpose : This program is used to plot the histograms prepared by C.1.4
####################
{
TString location("/uscms/home/bikashji/");
TCanvas *c1 = new TCanvas("c1", "",66,52,700,502);c1–>Divide(1,2);
pad = c1–>cd(1);
pad–>SetPad(0,0.1687764,0.9971264,0.9957806);
pad = c1–>cd(2);
pad–>SetPad(0,0.004219409,0.9827586,0.335443);
c1–>cd(1);
c1–>cd(1);
gStyle–>SetOptFit(1);
gStyle–>SetOptStat(0);
gStyle–>SetOptTitle(0);
c1_1–>Range(-158.2304,-4.36755,2269.342,4.718543);
c1_1–>SetFillColor(0);
c1_1–>SetBorderMode(0);
```

```
c1_1->SetBorderSize(2);

c1_1->SetLogy();

c1_1->SetTickx(1);

c1_1->SetTicky(1);

c1_1->SetLeftMargin(0.1887608);

c1_1->SetRightMargin(0.110951);

c1_1->SetTopMargin(0.07908163);

c1_1->SetBottomMargin(0.1505102);

c1_1->SetFrameFillStyle(0);

c1_1->SetFrameBorderMode(0);

c1_1->SetFrameFillStyle(0);

c1_1->SetFrameBorderMode(0);

c1_1->SetLogy();

TFile* f2d =TFile::Open(location+"VDRELL_YAN_CHOOPED.root");

TH1F* DY = (TH1F*)f2d->Get("SCALED_DiELECTRON_reco_NEW");

TFile* f2d =TFile::Open(location+"DATA.root");

TH1F* Data = (TH1F*)f2d->Get("DiELECTRON_reco_NEW");

TFile* f2d =TFile::Open(location+"DATA_POISSON.root");

TH1F* poissionData = (TH1F*)f2d->Get("DiELECTRON_reco_NEW");

TFile* f2d =TFile::Open(location+"LAMBDA-9.root");

TH1F* con_lam9 = (TH1F*)f2d->Get("SCALED_DiELECTRON_reco_NEW");

TFile* f2d =TFile::Open(location+"LAMBDA-11.root");

TH1F* con_lam11 = (TH1F*)f2d->Get("SCALED_DiELECTRON_reco_NEW");

TFile* f2d =TFile::Open(location+"LAMBDA-13.root");

TH1F* con_lam13 = (TH1F*)f2d->Get("SCALED_DiELECTRON_reco_NEW");

TFile* f2d =TFile::Open(location+"LAMBDA-15.root");

TH1F* con_lam15 = (TH1F*)f2d->Get("SCALED_DiELECTRON_reco_NEW");

TFile* f2d =TFile::Open(location+"LAMBDA-17.root");
```

```
TH1F* con_lam17 = (TH1F*)f2d->Get("SCALED_DiELECTRON_reco_NEW");

TFile* f2d =TFile::Open(location+"LAMBDA-19.root");

TH1F* con_lam19 = (TH1F*)f2d->Get("SCALED_DiELECTRON_reco_NEW");

TFile* f2d =TFile::Open(location+"LAMBDA-9.root");

TH1F* des_lam9 = (TH1F*)f2d->Get("SCALED_DiELECTRON_reco_NEW");

TFile* f2d =TFile::Open(location+"LAMBDA-11.root");

TH1F* des_lam11 = (TH1F*)f2d->Get("SCALED_DiELECTRON_reco_NEW");

TFile* f2d =TFile::Open(location+"LAMBDA-13.root");

TH1F* des_lam13 = (TH1F*)f2d->Get("SCALED_DiELECTRON_reco_NEW");

TFile* f2d =TFile::Open(location+"LAMBDA-15.root");

TH1F* des_lam15 = (TH1F*)f2d->Get("SCALED_DiELECTRON_reco_NEW");

TFile* f2d =TFile::Open(location+"VAR_BIN_MULTIJETS_SCALED.root");

TH1F* MULTIJETS = (TH1F*)f2d->Get("SCALED_DiELECTRON_reco_NEW");

TFile* f2d =TFile::Open(location+"VAR_BIN_DIBOSON_SCALED.root");

TH1F* DIBOSON = (TH1F*)f2d->Get("SCALED_DiELECTRON_reco_NEW");

TFile* f2d =TFile::Open(location+"VAR_BIN_TW_TBARW_SCALED.root");

TH1F* TW_TBARW = (TH1F*)f2d->Get("SCALED_DiELECTRON_reco_NEW");

TFile* f2d =TFile::Open(location+"VAR_BIN_dytautau_SCALED.root");

TH1F* DYTAUTAU = (TH1F*)f2d->Get("SCALED_DiELECTRON_reco_NEW");

TFile* f2d =TFile::Open(location+"VAR_BIN_wjets_SCALED.root");

TH1F* WJETS = (TH1F*)f2d->Get("SCALED_DiELECTRON_reco_NEW");

TFile* f2d =TFile::Open(location+"VAR_BIN_ttbar_SCALED.root");

TH1F* TTBAR = (TH1F*)f2d->Get("SCALED_DiELECTRON_reco_NEW");

TFile* f2d =TFile::Open(location+"jet_root_new.root");

TH1F* JETS_DATA = (TH1F*)f2d->Get("JET_BKG_NEW");

Data->SetMarkerStyle(20);

JETS_DATA->SetLineColor(2);

DIBOSON->SetLineColor(6);
```

```
TW_TBARW->SetLineColor(5);

DYTAUTAU->SetLineColor(12);

TTBAR->SetLineColor(3);

JETS_DATA->SetFillColor(2);

DIBOSON->SetFillColor(6);

TW_TBARW->SetFillColor(5);

DYTAUTAU->SetFillColor(12);

TTBAR->SetFillColor(3);

DY->SetLineColor(kBlack);

JETS_DATA->SetLineColor(2);

DIBOSON->SetLineColor(kBlack);

TW_TBARW->SetLineColor(kBlack);

DYTAUTAU->SetLineColor(kBlack);

TTBAR->SetLineColor(1);

con_lam9->SetLineColor(2);

des_lam9->SetLineColor(2);

con_lam11->SetLineColor(3);

des_lam11->SetLineColor(3);

con_lam13->SetLineColor(4);

des_lam13->SetLineColor(4);

con_lam15->SetLineColor(6);

des_lam15->SetLineColor(6);

DY->GetXaxis()->SetRangeUser(300,2000);

DY->GetYaxis()->SetRangeUser(0.001,1e4);

DY->GetXaxis()->SetNdivisions(511);

DY->GetXaxis()->SetMoreLogLabels(10);

DY->GetXaxis()->SetTitle("M_{ee} (GeV/c^{2})");

DY->GetXaxis()->SetRange(15,39);
```

```
DY->GetXaxis()->SetMoreLogLabels();

DY->GetXaxis()->SetNdivisions(511);

DY->GetXaxis()->SetLabelFont(42);

DY->GetXaxis()->SetLabelSize(0.047935);

DY->GetXaxis()->SetTitleSize(0.06);

DY->GetXaxis()->SetTitleOffset(1.025);

DY->GetXaxis()->SetTitleFont(42);

DY->GetYaxis()->SetTitle("Events /20 GeV c^{-2}");

DY->GetYaxis()->SetLabelFont(42);

DY->GetYaxis()->SetLabelSize(0.06);

DY->GetYaxis()->SetTitleSize(0.06);

DY->GetYaxis()->SetTitleFont(42);

DY->GetXaxis()->SetRangeUser(300,2000);

DY->GetYaxis()->SetRangeUser(0.001,1e4);

DY->GetXaxis()->SetNdivisions(511);

DY->GetXaxis()->SetMoreLogLabels(10);

DY->GetXaxis()->SetTitle("M_{ee} (GeV/c^{2})");

DY->GetXaxis()->SetRange(15,39);

DY->GetXaxis()->SetMoreLogLabels();

DY->GetXaxis()->SetNdivisions(511);

DY->GetXaxis()->SetLabelFont(42);

DY->GetXaxis()->SetLabelSize(0.047935);

DY->GetXaxis()->SetTitleSize(0.);

DY->GetXaxis()->SetTitleOffset(1.025);

DY->GetXaxis()->SetTitleFont(42);

DY->GetYaxis()->SetTitle("Events /20 GeV c^{-2}");

DY->GetYaxis()->SetLabelFont(42);

DY->GetYaxis()->SetLabelSize(0.047535);
```

```
DY->GetYaxis()->SetTitleSize(0.06);

DY->GetYaxis()->SetTitleFont(42);

DY->SetXTitle("M_{ee} (GeV/c^{2})");

DY->SetYTitle("Events /(20 GeV/c^{2})");

DY->GetYaxis()->SetNdivisions(9,9,9,kFALSE);

des_lam9->SetLineStyle(2);

des_lam11->SetLineStyle(2);

des_lam13->SetLineStyle(2);

des_lam15->SetLineStyle(2);

DY->SetFillColor(kCyan);

THStack *hs = new THStack("hs","Stacked 1D histograms");

con_lam9->Add(JETS_DATA); con_lam11->Add(JETS_DATA); con_lam13->Add(JETS_DATA);

con_lam15->Add(JETS_DATA);

con_lam9->Add(DIBOSON); con_lam11->Add(DIBOSON); con_lam13->Add(DIBOSON);

con_lam15->Add(DIBOSON);

con_lam9->Add(TW_TBARW); con_lam11->Add(TW_TBARW); con_lam13->Add(TW_TBARW);

con_lam15->Add(TW_TBARW);


con_lam9->Add(TTBAR); con_lam11->Add(TTBAR); con_lam13->Add(TTBAR);

con_lam15->Add(TTBAR);

con_lam9->Add(DYTAUTAU); con_lam11->Add(DYTAUTAU); con_lam13->Add(DYTAUTAU);

con_lam15->Add(DYTAUTAU);

des_lam9->Add(JETS_DATA); des_lam11->Add(JETS_DATA); des_lam13->Add(JETS_DATA);

des_lam15->Add(JETS_DATA);

des_lam9->Add(DIBOSON); des_lam11->Add(DIBOSON); des_lam13->Add(DIBOSON);

des_lam15->Add(DIBOSON);

des_lam9->Add(TW_TBARW); des_lam11->Add(TW_TBARW); des_lam13->Add(TW_TBARW);

des_lam15->Add(TW_TBARW);
```

```
des_lam9–>Add(TTBAR); des_lam11–>Add(TTBAR); des_lam13–>Add(TTBAR);

des_lam15–>Add(TTBAR);

des_lam9–>Add(DYTAUTAU); des_lam11–>Add(DYTAUTAU); des_lam13–>Add(DYTAUTAU);

des_lam15–>Add(DYTAUTAU);

DY–>Draw("hist");

DY–>GetXaxis()–>SetLabelSize(0);

con_lam11–>Draw("histsame"); con_lam13–>Draw("histsame"); con_lam15–>Draw("histsame");

des_lam11–>Draw("histsame"); des_lam13–>Draw("histsame"); des_lam15–>Draw("histsame");

hs–>Add(TW_TBARW);

hs–>Add(DIBOSON);

hs–>Add(JETS_DATA);

hs–>Add(TTBAR);

hs–>Add(DY);

hs–>Draw("histsame ");

float b, err;

for(Int_t s=1;s¡=400;s++)

{

b = Data–>GetBinContent(s);

err = sqrt(b);

Data–>SetBinError(s,0);

}

const double alpha = 1 - 0.6827;

TGraphAsymmErrors * g = new TGraphAsymmErrors(Data);

for (int i = 0; i ¡ g–>GetN(); ++i) {

double width=(poissionData–>GetBinWidth(i+1))/20.0;

if(width==0){width=12345678;}

float N = g–>GetY()[i]*width;
```

```
double L = (N==0) ? 0 : (ROOT::Math::gamma_quantile(alpha/2,N,1.));

double U = (N==0) ? ( ROOT::Math::gamma_quantile_c(alpha,N+1,1) ) :

( ROOT::Math::gamma_quantile_c(alpha/2,N+1,1) );

g->SetPointEYlow(i, (N-L)/width);

g->SetPointEYhigh(i, (U-N)/width);

g->SetPointEXlow(i, 0);

g->SetPointEXhigh(i, 0);

}

g->Draw("PSAME");

Data->Draw("PSAME");

c1_1->RedrawAxis();

TPaveText *pt = new TPaveText(0.3717579,0.9387755,0.8069164,0.9897959,"brNDC");

pt->SetFillColor(0);

pt->SetLineColor(0);

pt->SetTextAlign(12);

pt->SetTextSize(0.04321678);

TText *text = pt->AddText("cMS, #sqrt{s} = 8 TeV,19.6 fb^{-1}");

pt->Draw("nopave");

TPaveText *pt = new TPaveText(0.1836242,0.715035,0.4496644,0.7797203,"brNDC");

pt->SetFillColor(kWhite);

pt->SetLineColor(kWhite);

pt->SetShadowColor(0);

pt->SetTextSize(0.04);

pt->SetTextAlign(12);

text=pt->AddText("cMS, #sqrt{s} = 7 TeV, 5.3 fb^{-1}");

pt = new TPaveText(0.3060345,0.6313559,0.4382184,0.7012712,"brNDC");

pt->SetFillColor(kWhite);

pt->SetLineColor(kWhite);
```

```
pt->SetShadowColor(0);

pt->SetTextSize(0.03);

pt->SetTextAlign(12);

text=pt->AddText("5.3 fb{-1}");

pt = new TPaveText(0.3017241,0.7733051,0.4612069,0.8347458,"brNDC");

pt->SetFillColor(kWhite);

pt->SetLineColor(kWhite);

pt->SetShadowColor(0);

pt->SetTextSize(0.03);

pt->SetTextAlign(12);

text=pt->AddText("cMS");

TLegend *leg = new TLegend(0.625,0.4682203,0.9252874,0.8389831,NULL,"brNDC");

leg->SetBorderSize(1);

leg->SetTextFont(62);

leg->SetTextSize(0.029);

leg->SetLineColor(0);

leg->SetLineStyle(1);

leg->SetLineWidth(1);

leg->SetFillColor(0);

leg->SetFillStyle(1001);

leg->AddEntry(Data,"data","P");

leg->AddEntry(con_lam11,"#Lambda = 11 TeV (const.)","L");

leg->AddEntry(des_lam11,"#Lambda = 11 TeV (destr.)","L");

leg->AddEntry(con_lam13,"#Lambda = 13 TeV (const.)","L");

leg->AddEntry(des_lam13,"#Lambda = 13 TeV (destr.)","L");

leg->AddEntry(con_lam15,"#Lambda = 15 TeV (const.)","L");

leg->AddEntry(des_lam15,"#Lambda = 15 TeV (destr.)","L");

leg->AddEntry(DY,"DY","f");
```

```
leg–>AddEntry(TTBAR,"t #bar{t}","f");

leg–>AddEntry(TW_TBARW,"tW","f");

leg–>AddEntry(DIBOSON,"diboson","f");

leg–>AddEntry(JETS_DATA,"jets (data)","f");

leg–>Draw("nopave");

const Int_t n = 26;


Float_t x[n] = {310, 330, 350, 370, 390, 410, 430, 450, 470, 490, 510, 530, 550, 570, 590, 610,

640, 680,

720, 770, 850, 950, 1150,

1450,

1800,2000};

Float_t y[n] = { 1.013, 1.037, 0.993, 0.987, 0.87, 1.103, 1.148, 0.848, 0.897, 1.121, 0.983, 1.019,

1.157, 0.99, 1.204, 1.091, 1.056, 1.031,

1.232, 1.092, 0.757, 1.239, 0.901,

0.465, 1.736,0};

Float_t ey[n] = {0.209, 0.216, 0.209, 0.211, 0.195, 0.252, 0.268, 0.188, 0.202, 0.253, 0.229, 0.242,

0.275, 0.248, 0.299, 0.291, 0.182, 0.193, 0.25,

0.204, 0.159, 0.27, 0.222,

0.33, 1.23,0};

Float_t ex[n] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};

c1–>cd(2);

TGraphErrors *gr = new TGraphErrors(n,x,y,ex,ey);

gr–>SetTitle("TGraphErrors Example");

gr–>SetMarkerColor(4);

gr–>SetMarkerStyle(20);

//gr–>Draw("AEP");

TMultiGraph *mg = new TMultiGraph();
```

```
mg->Add(gr,"p");

mg->Draw("ap");

mg->GetXaxis()->SetRangeUser(300,2000);

mg->GetYaxis()->SetRangeUser(0.4,2.1);

mg->GetXaxis()->SetLabelFont(42);

mg->GetXaxis()->SetLabelSize(0.2);

mg->GetXaxis()->SetTitleSize(0.16);

mg->GetXaxis()->SetTitleOffset(0.1);

mg->GetXaxis()->SetTitleFont(42);

mg->GetYaxis()->SetNdivisions(506);

mg->GetYaxis()->SetLabelFont(42);

mg->GetYaxis()->SetLabelSize(0.3);

mg->GetYaxis()->SetTitleSize(0.3);

mg->GetYaxis()->SetTitleOffset(0.1);

mg->GetYaxis()->SetTitleFont(42);

c1_2->SetFillColor(0);

c1_2->SetBorderMode(0);

c1_2->SetBorderSize(2);

c1_2->SetTickx(1);

c1_2->SetTicky(1);

c1_2->SetTopMargin(0.15);

c1_2->SetBottomMargin(0.15);

c1_2->SetFrameFillStyle(0);

c1_2->SetFrameBorderMode(0);

c1_2->SetFrameFillStyle(0);

c1_2->SetFrameBorderMode(0);

c1_2->SetLeftMargin(0.1887608);

//c1_2->SetRightMargin(0.110951);
```

//here is new

//c1_2–>Range(-144.3232,-0.6441558,2247.97,2.414286);

c1_2–>SetFillColor(0);

c1_2–>SetBorderMode(0);

c1_2–>SetBorderSize(2);

c1_2–>SetTickx(1);

c1_2–>SetTicky(1);

//c1_2–>SetLeftMargin(0.1856725);

c1_2–>SetTopMargin(0.0700637);

c1_2–>SetBottomMargin(0.4394904);

c1_2–>SetFrameFillStyle(0);

c1_2–>SetFrameBorderMode(0);

c1_2–>SetFrameBorderMode(0);

mg–>GetXaxis()–>SetTitle("M_{ee} (GeV/c$\hat{}$2})");

mg–>GetXaxis()–>SetRange(5,96);

mg–>GetXaxis()–>SetLabelFont(42);

mg–>GetXaxis()–>SetLabelSize(0.15);

mg–>GetXaxis()–>SetTitleSize(0.2);

mg–>GetXaxis()–>SetTitleOffset(0.9);

mg–>GetXaxis()–>SetTitleFont(42);

mg–>GetYaxis()–>SetTitle("#frac{data}{prediction(SM)}");

mg–>GetYaxis()–>SetNdivisions(506);

mg–>GetYaxis()–>SetLabelFont(42);

mg–>GetYaxis()–>SetLabelSize(0.15);

mg–>GetYaxis()–>SetTitleSize(0.12);

mg–>GetYaxis()–>SetTitleOffset(0.4);

mg–>GetYaxis()–>SetTitleFont(42);

```
c1_2->Modified();

c1_2->SetGrid();
c1->Update();
gPad->RedrawAxis();
gPad->RedrawAxis();


}
```

APPENDIX D

D.1   Python script for CI Limit calculation

```
####################
# Module Name : Limit.py
# Author : Pramod Lamichhane
# Date : Aug 2013
# Last modified : Feb 2013
# Purpose : This program uses the official CL95 macro and reads the cross-section from the file
prepared by the cross-section (event) calculator,
# calculates limits, and makes the plots for both interference cases
####################

import string
import os
from array import array
from ROOT import TCanvas,TGraph
from ROOT import gROOT
from math import sin
from array import array
```

```
from ROOT import TCanvas, TMultiGraph,TGraphErrors, TPaveLabel

os.system('rm RESULTS_OFlt_vS.txt')

LUMI=19482.4

qcd_k_NLO=[1.2962, 1.3014, 1.3006, 1.3014, 1.299, 1.2932, 1.2901, 1.2833, 1.2757, 1.2713, 1.2664,
1.265, 1.2589, 1.2603, 1.2558, 1.2565, 1.2585, 1.2643]

nnlo_factor=1.024 #THIS IS NNLO FACTOR

qcd_k=[ A*nnlo_factor for A in qcd_k_NLO]

print qcd_k_NLO

print qcd_k

Data_Lambda_Const_,Lambda_Const_max_2s_,Lambda_Const_max_1s_,Lambda_Const_,
Lambda_Const_min_1s_,Lambda_Const_min_2s_,Data_Lambda_Dest_,Lambda_Dest_max_2s_,
Lambda_Dest_max_1s_,Lambda_Dest_,Lambda_Dest_min_1s_,Lambda_Dest_min_2s_ =[ ],[ ],[ ],[ ],[
],[ ],[ ],[ ],[ ],[ ],[ ],[ ]

myfile=open("RESULTS_OFlt_vS.txt",'a') # THIS FILE WRITES ALL LIMITS DETAIL IN A
SINGLE FILE

reso_nu=5

number_step=((len(qcd_k)+2)*100-300)/100+1 # THIS NUMBER IS JUST TO READ THE IN-
PUT FILES AS LONG AS OUR MINIMUM MASS IS 300, IT IS GOOD

nondy=[800.8532756,207.5654896,67.9838348,26.2914988,8.9424216,4.6173288,
3.3704552,1.996946,0.9546376,0.3019772, 0.194824,0.1656004,0.0681884,0.0681884, 0,0,0,0]

QED_K_FACTOR=[0.997, 0.993, 0.989, 0.985 ,0.981 ,0.976, 0.972, 0.968,
0.964 ,0.960 ,0.955 ,0.951 ,0.947 ,0.943 ,0.939 ,0.934 ,0.930 ,0.926]

col_wr,col_atl,F_relt=[ ],[ ],[ ]

MASS_RANGE_NUMBER=len(qcd_k)

def delete_these():

del Data_Lambda_Dest_[:]

del Lambda_Dest_max_2s_[:]

del Lambda_Dest_max_1s_[:]
```

```
del Lambda_Dest_[:]

del Lambda_Dest_min_1s_[:]

del Lambda_Dest_min_2s_[:]

def LIMIT_CALCULATOR_(X1,Y1,X2,Y2,Y):

return (X2-X1)/(Y2-Y1)*(Y-Y1)+X1

def findlt_v(val_supply,search_which,sp_n=6):

ns=[ ]

for ab in range(0,len(search_which)):

if val_supply.SIG_PLUS_BKG¡search_which[ab].NUMBER:

#print 'len:^',len(search_which)

break

ns.append(round(search_which[ab].NUMBER,reso_nu))

ns.append(round(search_which[ab-1].NUMBER,reso_nu))

ns.append(search_which[ab].lambda_)

ns.append(search_which[ab-1].lambda_)

ns.append(round(LIMIT_CALCULATOR_(float(search_which[ab].lambda_),search_which[ab].NUMBER,

float(search_which[ab-1].lambda_),search_which[ab-1].NUMBER,val_supply.SIG_PLUS_BKG),reso_nu))

return ns


def WRITE_FINAL_VAL(START,KUN_RANGE,DESTRUCTIVE):

    fmt1="%10.2f%10.2f%4s%4s%10.2f%4s"

        if DESTRUCTIVE:

    else:

    myfile.writelines('FOR'+'^'+KUN_RANGE+': OBS EXP +1SIG +2SIG -1SIG -2SIG RESPEC-
TIVELY:"\n")

    for pl in range(START,len(qcd_k)*6,len(qcd_k)):

        myfile.writelines(fmt1%(F_relt[pl][0],F_relt[pl][1],F_relt[pl][2],F_relt[pl][3],F_relt[pl][4],'"\n"))
```

```python
myfile.writelines("'\n")
def FOR_PLOT_STUFF(DESTRUCTIVE):
    for pl in range(0,MASS_RANGE_NUMBER):
        if not DESTRUCTIVE:

            Data_Lambda_Const_.append(float((F_relt[pl][4])))

            Lambda_Const_.append(float((F_relt[pl+number_step][4])))

            Lambda_Const_max_1s_.append(float((F_relt[pl+number_step*2][4])))

            Lambda_Const_max_2s_.append(float((F_relt[pl+number_step*3][4])))

            Lambda_Const_min_1s_.append(float((F_relt[pl+number_step*4][4])))

            Lambda_Const_min_2s_.append(float((F_relt[pl+number_step*5][4])))
        else:


            Data_Lambda_Dest_.append(float((F_relt[pl][4])))

            Lambda_Dest_.append(float((F_relt[pl+number_step][4])))

            Lambda_Dest_max_1s_.append(float((F_relt[pl+number_step*2][4])))

            Lambda_Dest_max_2s_.append(float((F_relt[pl+number_step*3][4])))

            Lambda_Dest_min_1s_.append(float((F_relt[pl+number_step*4][4])))

            Lambda_Dest_min_2s_.append(float((F_relt[pl+number_step*5][4])))
class FINALlt_v(object):
    def __init__(self,OBS_CROSS_SECTION,OBS_SIGNAL,SIG_PLUS_BKG,_95lt_v):
        self.OBS_CROSS_SECTION=OBS_CROSS_SECTION
        self.OBS_SIGNAL=OBS_SIGNAL
        self.SIG_PLUS_BKG=SIG_PLUS_BKG
        self._95lt_v=_95lt_v
class CMS_TABLE(object):
    def __init__(self,lambda_,dil_rang):
        self.lambda_ = lambda_
        self.dil_rang= dil_rang
```

```
class CMS_TABLE_CLS(object):

    def __init__(self,LowMass,OBSlt_v,EXPlt_v,EXPlt_v_P_1,EXPlt_v_P_2,EXPlt_v_M_1,EXPlt_v_M_2):

        self.LowMass =LowMass

        self.OBSlt_v = OBSlt_v

        self.EXPlt_v = EXPlt_v

        self.EXPlt_v_P_1 = EXPlt_v_P_1

        self.EXPlt_v_P_2 = EXPlt_v_P_2

        self.EXPlt_v_M_1 = EXPlt_v_M_1

        self.EXPlt_v_M_2 = EXPlt_v_M_2

class FINAL_NO_TO_CHECK(object):

    def __init__(self,lambda_,NUMBER):

        self.lambda_=lambda_

        self.NUMBER=NUMBER

def populate(my_supply_array,mysupply_val):

    for a in range(1,len(col_atl)):

        f_vaclass=FINAL_NO_TO_CHECK(0,0.0)

        f_vaclass.lambda_=col_atl[a].lambda_

        f_vaclass.NUMBER=col_atl[a].dil_rang[mysupply_val]

        my_supply_array.append(f_vaclass)

pramod=input('Press 2 for 2 step and 1 for 1 step lambda:')

if pramod==1:

    des_file_name="NEW_DY_CI_DES_lambda_step1.txt"

    con_file_name="NEW_DY_CI_CON_lambda_step1.txt"

elif pramod==2:

    des_file_name="NEW_DY_CI_DES_lambda_step2.txt"

    con_file_name="NEW_DY_CI_CON_lambda_step2.txt"

else:

    raise ValueError('you typed wrong input, exiting')
```

```
def WHOLE_THING(DESTRUCTIVE):
    del col_atl[:]
    del col_wr[:]
    del F_relt[:]
    if DESTRUCTIVE:
    Atlas_style=open(des_file_name).read().split("\n")
    else :
    Atlas_style=open(con_file_name).read().split("\n")
    sp_n=len(Atlas_style)-3
    lines_atlas,lines_wrapper=[ ],[ ]


    INPUT_FROM_CLS=open("MYWRAPPER.txt").read().split("\n")


    for stuff in INPUT_FROM_CLS:
    lines_wrapper.append(stuff.strip().split(','))
    for stuff in Atlas_style:
    lines_atlas.append(stuff.split())
    for i in range(1,len(lines_atlas)-1):
    ATLAS_TABLE=CMS_TABLE('',[ ])
    for j in range(0,len(qcd_k)+1):
    if j==0:
    ATLAS_TABLE.lambda_=lines_atlas[i][j]
    else:
    ATLAS_TABLE.dil_rang.append(float(lines_atlas[i][j])*qcd_k[j-1]*QED_K_FACTOR[j-1]+nondy[j-1])
    col_atl.append(ATLAS_TABLE)


    for i in range(0,len(lines_wrapper)-1):
```

```python
W_cls=CMS_TABLE_CLS(0.0,0.0,0.0,0.0,0.0,0.0,0.0)

W_cls.LowMass =lines_wrapper[i][0]

W_cls.OBSlt_v =lines_wrapper[i][7]

W_cls.EXPlt_v =lines_wrapper[i][8]

W_cls.EXPlt_v_M_1 =lines_wrapper[i][9]

W_cls.EXPlt_v_P_1 =lines_wrapper[i][10]

W_cls.EXPlt_v_M_2 =lines_wrapper[i][11]

W_cls.EXPlt_v_P_2 =lines_wrapper[i][12]

col_wr.append(W_cls)

for stuff in col_atl:

    if stuff.lambda_=='9':

        print fmt1%(str(stuff.lambda_)+' '),

else:

        print fmt1%(str(stuff.lambda_)+''),

for i in range(0, len(stuff.dil_rang)):

        length=' '* (7-(len(str(round(stuff.dil_rang[i],2)))))

        print str(round(stuff.dil_rang[i],2))+length,

print

for i in range(0,len(qcd_k)):

    print round(col_atl[0].dil_rang[i],2)

tot_stor=[ ]

for val_ in range(0,len(qcd_k)):

    tot_stor_TEMP=[ ]

    F_lt=FINALlt_v(0.0,0.0,0.0,0.0)

    F_lt.OBS_CROSS_SECTION=float(col_wr[val_].OBSlt_v)

    F_lt.OBS_SIGNAL=float(col_wr[val_].OBSlt_v)*LUMI

    F_lt.SIG_PLUS_BKG=float(col_wr[val_].OBSlt_v)*LUMI+col_atl[0].dil_rang[val_]

    tot_stor_TEMP.append(F_lt)
```

```
F_lt=FINALlt_v(0.0,0.0,0.0,0.0)

F_lt.OBS_CROSS_SECTION=float(col_wr[val_].EXPlt_v)

F_lt.OBS_SIGNAL=float(col_wr[val_].EXPlt_v)*LUMI

_FINALlt_v.SIG_PLUS_BKG=float(col_wr[val_].EXPlt_v)*LUMI+col_atl[0].dil_rang[val_]

tot_stor_TEMP.append(F_lt)

F_lt=FINALlt_v(0.0,0.0,0.0,0.0)

F_lt.OBS_CROSS_SECTION=float(col_wr[val_].EXPlt_v_P_1)

F_lt.OBS_SIGNAL=float(col_wr[val_].EXPlt_v_P_1)*LUMI

F_lt.SIG_PLUS_BKG=float(col_wr[val_].EXPlt_v_P_1)*LUMI+col_atl[0].dil_rang[val_]

tot_stor_TEMP.append(F_lt)

F_lt=FINALlt_v(0.0,0.0,0.0,0.0)

F_lt.OBS_CROSS_SECTION=float(col_wr[val_].EXPlt_v_P_2)

F_lt.OBS_SIGNAL=float(col_wr[val_].EXPlt_v_P_2)*LUMI

F_lt.SIG_PLUS_BKG=float(col_wr[val_].EXPlt_v_P_2)*LUMI+col_atl[0].dil_rang[val_]

tot_stor_TEMP.append(F_lt)

F_lt=FINALlt_v(0.0,0.0,0.0,0.0)

F_lt.OBS_CROSS_SECTION=float(col_wr[val_].EXPlt_v_M_1)

F_lt.OBS_SIGNAL=float(col_wr[val_].EXPlt_v_M_1)*LUMI

F_lt.SIG_PLUS_BKG=float(col_wr[val_].EXPlt_v_M_1)*LUMI+col_atl[0].dil_rang[val_]

tot_stor_TEMP.append(F_lt)

F_lt=FINALlt_v(0.0,0.0,0.0,0.0)

F_lt.OBS_CROSS_SECTION=float(col_wr[val_].EXPlt_v_M_2)

F_lt.OBS_SIGNAL=float(col_wr[val_].EXPlt_v_M_2)*LUMI

F_lt.SIG_PLUS_BKG=float(col_wr[val_].EXPlt_v_M_2)*LUMI+col_atl[0].dil_rang[val_]

tot_stor_TEMP.append(F_lt)

tot_stor.append(tot_stor_TEMP)
f_va300,f_va400,f_va500,f_va600,f_va700,f_va800,
f_va900,f_va1000,f_va1100,f_va1200,f_va1300,f_va1400,f_va1500,f_va1600,f_va1700,
```

```
f_va1800,f_va1900,f_va2000=[ ],[ ],[ ],[ ],[ ],[ ],[ ],[ ],[ ],[ ],[ ],[ ],[ ],[ ],[ ],[ ],[ ],[ ]

populate(f_va300,0)

populate(f_va400,1)

populate(f_va500,2)

populate(f_va600,3)

populate(f_va700,4)

populate(f_va800,5)

populate(f_va900,6)

populate(f_va1000,7)

populate(f_va1100,8)

populate(f_va1200,9)

populate(f_va1300,10)

populate(f_va1400,11)

populate(f_va1500,12)

populate(f_va1600,13)

populate(f_va1700,14)        populate(f_va1800,15)

populate(f_va1900,16)

populate(f_va2000,17)

for a in range(0,6):

    F_relt.append(findlt_v(tot_stor[0][a],f_va300,sp_n,) )

    F_relt.append(findlt_v(tot_stor[1][a],f_va400,sp_n,) )

    F_relt.append(findlt_v(tot_stor[2][a],f_va500,sp_n,) )

    F_relt.append(findlt_v(tot_stor[3][a],f_va600,sp_n,) )

    F_relt.append(findlt_v(tot_stor[4][a],f_va700,sp_n,) )

    F_relt.append(findlt_v(tot_stor[5][a],f_va800,sp_n,) )

    F_relt.append(findlt_v(tot_stor[6][a],f_va900,sp_n,) )

    F_relt.append(findlt_v(tot_stor[7][a],f_va1000,sp_n,) )

    F_relt.append(findlt_v(tot_stor[8][a],f_va1100,sp_n,) )
```

```
        F_relt.append(findlt_v(tot_stor[9][a],f_va1200,sp_n,) )

        F_relt.append(findlt_v(tot_stor[10][a],f_va1300,sp_n,) )

        F_relt.append(findlt_v(tot_stor[11][a],f_va1400,sp_n,) )

        F_relt.append(findlt_v(tot_stor[12][a],f_va1500,sp_n,) )

        F_relt.append(findlt_v(tot_stor[13][a],f_va1600,sp_n,) )

        F_relt.append(findlt_v(tot_stor[14][a],f_va1700,sp_n,) )

        F_relt.append(findlt_v(tot_stor[15][a],f_va1800,sp_n,) )

        F_relt.append(findlt_v(tot_stor[16][a],f_va1900,sp_n,) )

        F_relt.append(findlt_v(tot_stor[17][a],f_va2000,sp_n,) )

    for i in range(0,len(qcd_k)):

        WRITE_FINAL_VAL(i,str((i+3)*100),DESTRUCTIVE)

    FOR_PLOT_STUFF(DESTRUCTIVE)

### MAKE PLUS POINT#################################


# THIS IS TO PUT THE PLUS SIGN IN LIMIT PLOT

ci_dilepton_dest_p=[12.6, 0]

mass_ci_dilepton_dest= [1300, 1400]

ci_dilepton_const_p=[17, 0]

mass_ci_dilepton_const= [1100, 1200]

ci_dilepton_dest_p_array= array( 'f',ci_dilepton_dest_p )

mass_ci_dilepton_dest_array= array( 'f',mass_ci_dilepton_dest )

ci_dilepton_const_p_array= array( 'f',ci_dilepton_const_p )

mass_ci_dilepton_const_array= array( 'f',mass_ci_dilepton_const )

thoplo_ci_dilepton_dest =TGraph(1, mass_ci_dilepton_dest_array, ci_dilepton_dest_p_array)

thoplo_ci_dilepton_dest.SetMarkerStyle(34)

thoplo_ci_dilepton_dest.SetMarkerColor(kRed)

thoplo_ci_dilepton_dest.SetMarkerSize(1.3)

thoplo_ci_dilepton_const =TGraph(1, mass_ci_dilepton_const_array, ci_dilepton_const_p_array)
```

thoplo_ci_dilepton_const.SetMarkerStyle(34)

thoplo_ci_dilepton_const.SetMarkerColor(kRed)

thoplo_ci_dilepton_const.SetMarkerSize(1.3)

int_lumi=LUMI/1000.

t = ROOT.TPaveLabel(0.18, 0.85, 0.86, 0.99, 'CMS, #sqrts = 8 TeV, %0.1f fb$\hat{}$-1}'% round(int_lumi,1),'brNDC')

t.SetTextSize(0.35)

t.SetBorderSize(0)

t.SetFillColor(0)

t.SetFillStyle(0)

MyStyle =TStyle("MyStyle","My Style")

MyStyle.SetCanvasBorderMode(0)

MyStyle.SetCanvasColor(10)

MyStyle.SetPadBorderMode(0)

MyStyle.SetPadColor(10)

MyStyle.SetPadBottomMargin(0.15)

MyStyle.SetPadLeftMargin(0.15)

MyStyle.SetPaperSize(18,24)

MyStyle.SetLabelSize(0.06,"XY")

MyStyle.SetLabelOffset(0.01,"XY")

MyStyle.SetTitleOffset(1.1,"XY")

MyStyle.SetTitleSize(0.057,"XY")

gROOT.SetStyle("MyStyle")

c1 = TCanvas('c1',"")

c2 = TCanvas('c2',"")

def LAST_ABA_PLOT(t):

    gROOT.Reset()

    c1.cd()

    gStyle.SetOptFit(1)

```
gStyle.SetOptStat(0)

gStyle.SetOptTitle(0)

c1.Range(200,4,2000,16)

c1.SetFillColor(0)

c1.SetBorderMode(0)

c1.SetBorderSize(2)

c1.SetTickx(1)

c1.SetTicky(1)

c1.SetLeftMargin(0.13)

c1.SetRightMargin(0.15)

c1.SetTopMargin(0.15)

c1.SetBottomMargin(0.15)

c1.SetFrameFillStyle(0)

c1.SetFrameBorderMode(0)

c1.SetFrameFillStyle(0)

c1.SetFrameBorderMode(0)

hr1=c1.DrawFrame(600,5,2000,16)

hr1.SetYTitle(" 95% C.L. #Lambda [TeV] ")

hr1.SetXTitle("M_eeL̂ow (GeV/ĉ2)")

hr1.GetXaxis().SetNdivisions(506)

c1.GetFrame().SetFillColor(21)

c1.GetFrame().SetBorderSize(12)

t.Draw('no pave')

mg1 =TMultiGraph()

pt = TPaveText(0.3104027,0.8618881,0.7466443,0.9055944,"brNDC")

pt.SetFillColor(kWhite)

pt.SetTextSize(0.14321678)

pt.Draw("no pave")
```

```
pt1 =TPaveText(0.3137584,0.7622378,0.75,0.8059441,"brNDC")
pt1.SetLineColor(kBlack)
pt1.SetFillColor(kWhite)
pt1.SetTextAlign(12)
pt1.SetTextSize(0.04020979)
text = pt1.AddText("destructive interference")
pt1.Draw("no pave")
x1=[ ]
for i in range(0,number_step):
    x1.append((i+3)*100)
print x1
n=number_step
Low_Mass = array( 'f',x1 )
Lambda_Dest_min_1s= array( 'f',Lambda_Dest_min_1s_ )
Lambda_Const_min_1s= array( 'f',Lambda_Const_min_1s_ )
Lambda_Dest_min_2s= array( 'f',Lambda_Dest_min_2s_ )
Lambda_Const_min_2s= array( 'f',Lambda_Const_min_2s_ )
Lambda_Dest_max_1s= array( 'f',Lambda_Dest_max_1s_ )
Lambda_Const_max_1s= array( 'f',Lambda_Const_max_1s_ )
Lambda_Dest_max_2s= array( 'f',Lambda_Dest_max_2s_ )
Lambda_Const_max_2s= array( 'f',Lambda_Const_max_2s_ )
Lambda_Const= array( 'f',Lambda_Const_ )
Lambda_Dest= array( 'f',Lambda_Dest_ )
Data_Lambda_Const= array( 'f',Data_Lambda_Const_ )
Data_Lambda_Dest= array( 'f',Data_Lambda_Dest_ )
grmin_1s_Dest = TGraph(n, Low_Mass, Lambda_Dest_min_1s)
grmin_1s_Const = TGraph(n, Low_Mass, Lambda_Const_min_1s)
grmin_2s_Dest = TGraph(n, Low_Mass, Lambda_Dest_min_2s)
```

```
grmin_2s_Const = TGraph(n, Low_Mass, Lambda_Const_min_2s)

grmax_1s_Dest = TGraph(n, Low_Mass, Lambda_Dest_max_1s)

grmax_1s_Const = TGraph(n, Low_Mass, Lambda_Const_max_1s)

grmax_2s_Dest = TGraph(n, Low_Mass, Lambda_Dest_max_2s)

grmax_2s_Const = TGraph(n, Low_Mass, Lambda_Const_max_2s)

gr_Dest =TGraph(n, Low_Mass, Lambda_Dest)

gr_Dest.SetLineStyle(7)

gr_Dest.SetLineColor(kBlue)

gr_Dest.SetLineWidth(3)

gr_Const= TGraph(n, Low_Mass, Lambda_Const)

gr_Const.SetLineStyle(7)

gr_Const.SetLineColor(kBlue)

gr_Const.SetLineWidth(3)

grshade_1s_Dest =TGraph(2*n)

grshade_1s_Const =TGraph(2*n)

grshade_2s_Dest = TGraph(2*n)

grshade_2s_Const =TGraph(2*n)

gr3_Dest =TGraph(n, Low_Mass, Data_Lambda_Dest)

gr3_Const =TGraph(n, Low_Mass, Data_Lambda_Const)

gr3_Const.SetLineColor(kBlack)

for i in range(0,n):

    grshade_1s_Dest.SetPoint(i,Low_Mass[i],Lambda_Dest_max_1s[i])

    grshade_1s_Dest.SetPoint(n+i,Low_Mass[n-i-1],Lambda_Dest_min_1s[n-i-1])

    grshade_1s_Const.SetPoint(i,Low_Mass[i],Lambda_Const_max_1s[i])

    grshade_1s_Const.SetPoint(n+i,Low_Mass[n-i-1],Lambda_Const_min_1s[n-i-1])

    grshade_2s_Dest.SetPoint(i,Low_Mass[i],Lambda_Dest_max_2s[i])

    grshade_2s_Dest.SetPoint(n+i,Low_Mass[n-i-1],Lambda_Dest_min_2s[n-i-1])

    grshade_2s_Const.SetPoint(i,Low_Mass[i],Lambda_Const_max_2s[i]);
```

```
        grshade_2s_Const.SetPoint(n+i,Low_Mass[n-i-1],Lambda_Const_min_2s[n-i-1])

        grshade_2s_Dest.SetFillColor(kGreen);

grshade_2s_Dest.Draw("f")

grshade_1s_Dest.SetFillColor(kYellow);

grshade_1s_Dest.Draw("f")

gr3_Dest.Draw("L")

gr_Dest.Draw("same")

thoplo_ci_dilepton_dest.Draw('P same')

line = TLine(2.0,0.95,7.95297,0.95)

line.SetLineStyle(3)

line.Draw("same")

line1 = TLine(7.95297,0.00,7.95297,0.95)

line1.SetLineStyle(3)

line1.Draw("same")

leg = TLegend(0.3912752,0.1562937,0.7402685,0.3818182)

leg.SetFillColor(kWhite)

leg.SetTextSize(0.03)

leg.AddEntry(gr_Dest,"Expected limit","L")

leg.AddEntry(grshade_1s_Dest,"Expected limit 1#sigma","f")

leg.AddEntry(grshade_2s_Dest,"Expected limit 2#sigma","f")

leg.AddEntry(gr3_Dest,"Observed limit","L")

leg.AddEntry(thoplo_ci_dilepton_dest, "best expected limit", "P");

leg.Draw("same")

pt = TPaveText(0.35,0.94,0.786,0.97)

pt.SetLineColor(0)

pt.SetFillColor(kWhite)

pt.SetTextAlign(12)

pt.SetTextSize(0.03496503)
```

```
text = pt.AddText("CMS Preliminary DESTRUCTIVE")

pt.Draw("nopave")

c2.cd()

gPad.DrawFrame(600,6,2000.0,16)

gStyle.SetOptFit(1)

gStyle.SetOptStat(0)

gStyle.SetOptTitle(0)

c2.Range(200,4,2000,16)

c2.SetFillColor(0)

c2.SetBorderMode(0)

c2.SetBorderSize(2)

c2.SetTickx(1)

c2.SetTicky(1)

c2.SetLeftMargin(0.19)

c2.SetRightMargin(0.11)

c2.SetTopMargin(0.15)

c2.SetBottomMargin(0.15)

c2.SetFrameFillStyle(0)

c2.SetFrameBorderMode(0)

c2.SetFrameFillStyle(0)

c2.SetFrameBorderMode(0)

hr2=c2.DrawFrame(300,5,2000,25)

hr2.SetYTitle(" 95% C.L. #Lambda [TeV]")

hr2.SetXTitle("M_ee{Low} (GeV/c2̂)")

hr2.GetXaxis().SetNdivisions(506)

hr2.GetYaxis().SetNdivisions(508)

new_pt = TPaveText(0.3104027,0.8618881,0.7466443,0.9055944,"brNDC")

new_pt.SetLineColor(0)
```

new_pt.SetFillColor(kWhite)

new_pt.SetTextAlign(12)

new_pt.SetTextSize(0.04321678)

new_pt.Draw("nopave")

new_pt1 =TPaveText(0.4137584,0.7622378,0.75,0.8059441,"brNDC")

new_pt1.SetLineColor(kBlack)

new_pt1.SetFillColor(kWhite)

new_pt1.SetTextAlign(12)

new_pt1.SetTextSize(0.04020979)

text = new_pt1.AddText("constructive interference")

new_pt1.Draw("nopave")

c2.GetFrame().SetFillColor(21)

c2.GetFrame().SetBorderSize(12)

grshade_2s_Const.GetXaxis().SetTitle("M_ee{Low} (GeV/c{2})")

mg2 = TMultiGraph()

grshade_2s_Const.SetFillColor(kGreen)

grshade_2s_Const.Draw("f")

grshade_1s_Const.SetFillColor(kYellow)

grshade_1s_Const.Draw("f")

thoplo_ci_dilepton_const.Draw('P')

r3_Const.Draw("L")

gr_Const.Draw('L')

gPad.Modified()

line2 = TLine(10.0054,0.00,10.0054,0.95)

line2.SetLineStyle(3)

line2.Draw("SAME")

line3 = TLine(2.0,0.95,10.0054,0.95)

line3.SetLineStyle(3)

```
line3.Draw("SAME")

leg1 = TLegend(0.3912752,0.1562937,0.7402685,0.3818182)

leg1.SetFillColor(kWhite)

leg1.SetTextSize(0.03)

leg1.AddEntry(gr_Const,"Expected limit","L")

leg1.AddEntry(grshade_1s_Const,"Expected limit 1#sigma","f")

leg1.AddEntry(grshade_2s_Const,"Expected limit 2#sigma","f")

leg1.AddEntry(gr3_Const,"Observed limit","L")

leg1.AddEntry(thoplo_ci_dilepton_const, "best expected limit", "P");

leg1.Draw("same")

pt_1 = TPaveText(0.3,0.8576224,0.736,0.9013287,"brNDC")

pt_1.SetLineColor(0)

pt_1.SetFillColor(kWhite)

pt_1.SetTextAlign(12)

pt_1.SetTextSize(0.03321678)

pt_1.Draw("nopave")

t.Draw('no pave')

pt_new = TPaveText(0.35,0.94,0.786,0.97,"brNDC")

pt_new.SetLineColor(0)

pt_new.SetFillColor(kWhite)

pt_new.SetTextAlign(12)

pt_new.SetTextSize(0.03496503)

text2 = pt_new.AddText("CMS Preliminary CONSTRUCTIVE")

text2.Draw('no pave')

c2.Update()

c1.Update()

c1.SaveAs('des_electron_19p5ifblt_v.pdf')

c2.SaveAs('con_electron_19p5ifblt_v.pdf')
```

WHOLE_THING(True)

myfile.writelines("\n")

WHOLE_THING(False)

myfile.close()

LAST_ABA_PLOT(t)

D.2 Python script for K-factor calculation

```
####################
# Module Name : kfactor.py
# Author : Pramod Lamichhane
# Date : Aug 2013
# Last modified : Feb 2013
# Purpose : This program calculates k-factor as a function of minimum di-lepton mass starting
# from 200 GeV to 2 TeV using the root files prepared by programs given in B
####################

from ROOT import *
from glob import glob
from ROOT import TChain
import math
from ROOT import TCanvas, TFormula, TF1,TFile,TH1F
from ROOT import gROOT, gObjectTable
import ROOT
import math
from math import *
import os
```

```python
from array import array

ROOT.gROOT.Reset()

class EVENT_INFORMATION(object):

    def __init__(self,evn,evr):

        self.evn=evn

        self.evr=evr

pth=[ ]

mcat=[ ]

def make_bin(ROOT_FILE,sig,ev,WHICH_MC,START_MASS_SAMPLE):

    pramod=EVENT_INFORMATION(0.0,0.0)

    TARGET_LUMI_TO_NORMALIZE_ALL_SAMPLES=143687.4

    f=TFile(ROOT_FILE)

    GEN_HIS=f.Get('INVMASS_RECOPTETA')

    LUMI_OF_THIS_SAMPLE=ev/sig # IN PICO BARN INVERSE

    SCALE_FACTOR=TARGET_LUMI_TO_NORMALIZE_ALL_SAMPLES/LUMI_OF_THIS_SAMPLE

    EVENT_COUNT=0.0

    evr=0.0

    for i in range((START_MASS_SAMPLE/10)+1,(4000/10)+2):

        EVENT_COUNT+=GEN_HIS.GetBinContent(i)

        evr+=pow(GEN_HIS.GetBinError(i),2)

    if evr>0.0:

        evr=math.sqrt(evr)

    pramod.evn=EVENT_COUNT*SCALE_FACTOR

    pramod.evr=evr*SCALE_FACTOR

    if WHICH_MC=='pth':

        pth.append(pramod)

    if WHICH_MC=='mcat':

        mcat.append(pramod)
```

```
fmt="

def CALCULATION_PART(pth,mcat):

    for i in range(0,len(pth)):

        to_PY_ENTRY,to_PY_ERROR,to_mcn_ENTRY,

    to_mcn_ERROR,k_factor,k_factor_err=0.0,0.0,0.0,0.0,0.0,0.0

        to_PY_ENTRY=pth[i].evn

        to_PY_ERROR=pth[i].evr

        to_mcn_ENTRY=mcat[i].evn

        to_mcn_ERROR=mcat[i].evr

        k_factor=to_mcn_ENTRY/to_PY_ENTRY

        k_factor_err=k_factor

    *math.sqrt((pow(to_PY_ERROR/to_PY_ENTRY,2)+pow(to_mcn_ERROR/to_mcn_ENTRY,2)))

        fmt="make_bin('mcn_200.root', 1.500 , 177287 , 'mcat',200)

make_bin('mcn_300.root', 0.3383 , 179591 , 'mcat',300)

make_bin('mcn_400.root', 0.1130 , 180655 , 'mcat',400)

make_bin('mcn_500.root', 4.6255E-02, 181476 , 'mcat',500)

make_bin('mcn_600.root', 2.1451E-02, 182044 , 'mcat',600)

make_bin('mcn_700.root', 1.0854E-02, 182880 , 'mcat',700)

make_bin('mcn_800.root', 5.8270E-03, 183259 , 'mcat',800)

make_bin('mcn_900.root', 3.2788E-03, 183737 , 'mcat',900)

make_bin('mcn_1000.root', 1.9114E-03, 184054 , 'mcat',1000)

make_bin('mcn_1100.root', 1.1484E-03, 184581 , 'mcat',1100)

make_bin('mcn_1200.root', 7.0549E-04, 184862 , 'mcat',1200)

make_bin('mcn_1300.root', 4.4213E-04, 185211 , 'mcat',1300)

make_bin('mcn_1400.root', 2.8160E-04, 185494 , 'mcat',1400)

make_bin('mcn_1500.root', 1.8186E-04, 185760 , 'mcat',1500)

make_bin('mcn_1600.root', 1.1885E-04, 186023 , 'mcat',1600)
```

```
make_bin('mcn_1700.root', 7.8392E-05, 186076 , 'mcat',1700)

make_bin('mcn_1800.root', 5.2184E-05, 186189 , 'mcat',1800)

make_bin('mcn_1900.root', 3.5070E-05, 186676 , 'mcat',1900)

make_bin('mcn_2000.root', 2.3689E-05, 186916 , 'mcat',2000)

make_bin('PY_200_INF.root', 1.17700000,199994,'pth',200)

make_bin('PY_300_INF.root', 0.26170000,199992,'pth',300)

make_bin('PY_400_INF.root', 0.08690000,199998,'pth',400)

make_bin('PY_500_INF.root', 0.03556000,199996,'pth',500)

make_bin('PY_600_INF.root', 0.01650000,199999,'pth',600)

make_bin('PY_700_INF.root', 0.00836200,199996,'pth',700)

make_bin('PY_800_INF.root', 0.00450800,199998,'pth',800)

make_bin('PY_900_INF.root', 0.00254500,200000,'pth',900)

make_bin('PY_1000_INF.root',0.00149100,200000,'pth',1000)

make_bin('PY_1100_INF.root',0.00090080,200000,'pth',1100)

make_bin('PY_1200_INF.root',0.00055450,200000,'pth',1200)

make_bin('PY_1300_INF.root',0.00034910,199999,'pth',1300)

make_bin('PY_1400_INF.root',0.00022260,199998,'pth',1400)

make_bin('PY_1500_INF.root',0.00014450,200000,'pth',1500)

make_bin('PY_1600_INF.root',0.00009428,199999,'pth',1600)

make_bin('PY_1700_INF.root',0.00006236,200000,'pth',1700)

make_bin('PY_1800_INF.root',0.00004152,200000,'pth',1800)

make_bin('PY_1900_INF.root',0.00002785,200000,'pth',1900)

make_bin('PY_2000_INF.root',0.00001874,200000,'pth',2000)

CALCULATION_PART(pth,mcat)
```

D.3 Python script for PDF uncertainty calculation

```
####################
# Module Name : PDF.py
# Author : Pramod Lamichhane
# Date : Aug 2013
# Last modified : Feb 2013
# Purpose : This program calculates PDF uncertainty as a function of minimum di-lepton
# mass starting from 200 GeV to 2 TeV using the root files prepared by programs given in B
####################


from ROOT import TCanvas, TGraph
from ROOT import gROOT
from math import sin
from ROOT import*
import ROOT
from array import array
from ROOT import TCanvas, TGraphErrors
pramod_lamichhane=[ ]
scale_to68=1.64485 # Supply 1 if the chosen PDFSET gives at 68%
COMMON_LUMI=5252 # in picobarns
CROSS_SECTION=[12.16,1.517,1.517,1.517,0.1112, 0.4515 , 0.4515, 0.01048,0.01048,0.005615,0.005615,
0.005615,0.001837, 0.001837 , 0.001837 , 0.001837 ,1.744e-4,1.744e-4,1.744e-4]
ROOTFILES=['tr_evdy120','tr_evdy200','tr_evdy200','tr_evdy200',
'tr_evdy400','tr_evdy500','tr_evdy500','tr_evdy700',
'tr_evdy700','tr_evdy800','tr_evdy800','tr_evdy800','tr_evdy1000','tr_evdy1000','tr_evdy1000',
'tr_evdy1000','tr_evdy1500','tr_evdy1500','tr_evdy1500']
EVENTS_NUMBER=[99987,99991,99991,99991,99986,99986,99986,
```

```
99990,99990,99990,99990,99990,99992,99992,99992,99992,99999,99999,99999]

pdf_un_v_UP=[ ]

pdf_un_v_DOWN=[ ]

ALL_100GEV_INFORMATION=[ ]

def pdf_uncertainty_calc(ROOT_FILE_NAME,SURU_MASS,ANTIM_MASS,a):

    pl1=ROOT.TFile(ROOT_FILE_NAME+'.root')

    PLUS=pl1.Get('Dimuon_reco_PLUS')

    MINUS=pl1.Get('Dimuon_reco_MINUS')

    CENTRAL=pl1.Get('Dimuon_reco_CENTRAL')

    PLUS_,CENTRAL_,MINUS_=0.0,0.0,0.0

    for i in range((SURU_MASS/10)+1,(ANTIM_MASS/10)+1):

    PLUS_+=PLUS.GetBinContent(i)

    CENTRAL_+=CENTRAL.GetBinContent(i)

    MINUS_+=MINUS.GetBinContent(i)

    PLUS_=PLUS_*COMMON_LUMI/(EVENTS_NUMBER[a]/CROSS_SECTION[a])

    CENTRAL_=CENTRAL_*COMMON_LUMI/(EVENTS_NUMBER[a]/CROSS_SECTION[a])

    MINUS_=MINUS_*COMMON_LUMI/(EVENTS_NUMBER[a]/CROSS_SECTION[a])

    ALL_100GEV_INFORMATION.append([PLUS_,CENTRAL_,MINUS_])

for i in range(0,len(ROOTFILES)-1):

    pdf_uncertainty_calc(ROOTFILES[i],(i+2)*100,(i+3)*100,i)

pdf_uncertainty_calc(ROOTFILES[18],2000,4001,18)

fmt="%s%i%s%10.1f%s%10.1f \t %10.5f%10.5f%10.5f"

for i in range(0,len(ALL_100GEV_INFORMATION)):

    up_events,central,down_events=0.0,0.0,0.0

    for a in range(i,len(ALL_100GEV_INFORMATION)):

        up_events+=ALL_100GEV_INFORMATION[a][0]

        central+=ALL_100GEV_INFORMATION[a][1]

        down_events+=ALL_100GEV_INFORMATION[a][2]
```

pdf_un_v_UP.append(round((up_events-central)/central*100/scale_to68,5))

pdf_un_v_DOWN.append(round((down_events-central)/central*100/scale_to68,5))

print fmt%('MINMASS \t ',(i+2)*100,' \t UP % \t ',

round((up_events-central)/central*100/scale_to68,1),

' \t DOWN% \t ',round((-down_events+central)/central*100/scale_to68,1),

round(up_events,5),round(central,5),round(down_events,5))

D.4   Python script to count the events as a function of minimum di-electron mass

```
#####################
# Module Name : COUNER.py
# Author : Pramod Lamichhane
# Date : Aug 2013
# Last modified : Feb 2013
# Purpose : This program calculates cross-section and errors as a function of
# minimum di-lepton mass starting from 300 GeV to 2.5 TeV using the root files prepared
# by programs given in B
#####################
from ROOT import *
from glob import glob
from ROOT import TChain
import math
from ROOT import TCanvas, TFormula, TF1,TFile,TH1F
from ROOT import gROOT, gObjectTable
```

```python
import ROOT
import math
from math import *
import os
from array import array
ci,ci_MUON,ci_ELECTRON=[ ],[ ],[ ]
ELECTRON_KI_MUON=input('PRESS 1 FOR ELECTRON AND 2 FOR MUON:\t \t')
cl_y=0.0
if ELECTRON_KI_MUON==2:
cl_y=20000. #lumi o f the latest json release
if ELECTRON_KI_MUON==1:
cl_y=20000. #lumi o f the latest json release
cl_y_CORRECT_VALUE=input('IF THIS LUMI IS NOT CORRECT, SUPPLY CORRECT LUMI
IN PICOBARNS (IF CORRECT,JUST TYPE 0 TO SKIP):\t \t')
if not cl_y_CORRECT_VALUE==0:
cl_y=cl_y_CORRECT_VALUE
M300_RANGE=[300,600]
M500_RANGE=[600,1000]
M800_RANGE=[1000,4001] #INCLUDING OVER FLOW
WHOLE_THING=[ ]
ROOT.gROOT.Reset()
class INFORMATION(object):
    def __init__(self,RECO_EVENT,re_r):
        self.RECO_EVENT=RECO_EVENT
        self.re_r=re_r
fmt_str1="%10.2f"
fmt_str="%s%10.f%10.f%10.f%10.f%10.f%10.f%10.f%10.f%10.f%10.f%10.f%10.f%10.f%10.f%10.f%10.f%10.f%10.f
class sample(object):
```

```python
    def __init__(self, name,cross_section,events,filter_eff):
        self.name = name
        self.cross_section = cross_section
        self.events= events
        self.filter_eff=filter_eff

ci_MUON=[
sample('CIToMuMu_ITCM5_M-300_TuneZ2star', 0.2619, 50970 , 0.766 ),
sample('CIToMuMu_ITCM5_M-500_TuneZ2star', 0.03549, 24832 , 0.873 ),
sample('CIToMuMu_ITCM5_M-800_TuneZ2star', 0.004512, 25343 , 0.902 ),
sample('CIToMuMu_Con_Lambda-9_M-300', 0.3026, 49833 , 0.786 ),
sample('CIToMuMu_Con_Lambda-9_M-500', 0.0568 , 25281 , 0.861 ),
sample('CIToMuMu_Con_Lambda-9_M-800', 0.01454, 25052 , 0.917 ),
sample('CIToMuMu_Con_Lambda-11_M-300', 0.2847 , 52138 , 0.748 ),
sample('CIToMuMu_Con_Lambda-11_M-500', 0.04717 , 23564 , 0.923 ),
sample('CIToMuMu_Con_Lambda-11_M-800', 0.009625, 26822 , 0.855 ),
sample('CIToMuMu_Con_Lambda-13_M-300', 0.2779 , 48959 , 0.766 ),
sample('CIToMuMu_Con_Lambda-13_M-500', 0.04288, 25099 , 0.863 ),
sample('CIToMuMu_Con_Lambda-13_M-800', 0.007523 , 14312 , 0.917 ),
sample('CIToMuMu_Con_Lambda-15_M-300', 0.272 , 49943 , 0.779 ),
sample('CIToMuMu_Con_Lambda-15_M-500', 0.04031 , 24997 , 0.869 ),
sample('CIToMuMu_Con_Lambda-15_M-800', 0.006455 , 25714 , 0.89 ),
sample('CIToMuMu_Con_Lambda-17_M-300', 0.2697, 49647 , 0.79 ),
sample('CIToMuMu_Con_Lambda-17_M-500', 0.03913, 25528 , 0.849 ),
sample('CIToMuMu_Con_Lambda-17_M-800', 0.005868 , 24659 , 0.926 ),
sample('CIToMuMu_Con_Lambda-19_M-300', 0.2676, 49758 , 0.786 ),
sample('CIToMuMu_Con_Lambda-19_M-500', 0.03817, 24931 , 0.866 ),
sample('CIToMuMu_Con_Lambda-19_M-800', 0.005514, 25154 , 0.911 ),
sample('CIToMuMu_ITCM5_M-300_TuneZ2star', 0.2619 ,50970 , 0.766 ),
```

```
sample('CIToMuMu_ITCM5_M-500_TuneZ2star', 0.03549 ,24832 , 0.873 ),

sample('CIToMuMu_ITCM5_M-800_TuneZ2star', 0.004512 ,25343 , 0.902 ),

sample('CIToMuMu_Des_Lambda-9_M-300', 0.2559 , 49393 , 0.802 ),

sample('CIToMuMu_Des_Lambda-9_M-500', 0.0391 , 25620 , 0.865 ),

sample('CIToMuMu_Des_Lambda-9_M-800', 0.008847 , 25167 , 0.93 ),

sample('CIToMuMu_Des_Lambda-11_M-300', 0.2539 , 50532 , 0.78 ),

sample('CIToMuMu_Des_Lambda-11_M-500', 0.03531 , 25531 , 0.861 ),

sample('CIToMuMu_Des_Lambda-11_M-800', 0.005811 , 24955 , 0.933 ),

sample('CIToMuMu_Des_Lambda-13_M-300', 0.2543 , 50248 , 0.778 ),

sample('CIToMuMu_Des_Lambda-13_M-500', 0.03399 , 24100 , 0.868 ),

sample('CIToMuMu_Des_Lambda-13_M-800', 0.004776, 24990 , 0.926 ),

sample('CIToMuMu_Des_Lambda-15_M-300', 0.2565 , 49444 , 0.791 ),

sample('CIToMuMu_Des_Lambda-15_M-500', 0.03391 , 25890 , 0.844 ),

sample('CIToMuMu_Des_Lambda-15_M-800', 0.004429 , 25040 , 0.922 )]

ci_ELECTRON=[

sample('CIToEE_ITCM5_M-300_TuneZ2star', 0.2621 , 55751 , 0.778 ),

sample('CIToEE_ITCM5_M-500_TuneZ2star', 0.03562 , 26056 , 0.886 ),

sample('CIToEE_ITCM5_M-800_TuneZ2star', 0.004503 , 25672 , 0.929 ),

sample('CIToEE_Con_Lambda-9_M-300', 0.3041 , 57300 , 0.76 ),

sample('CIToEE_Con_Lambda-9_M-500', 0.05713 , 26797 , 0.866 ),

sample('CIToEE_Con_Lambda-9_M-800', 0.01445 , 26111 , 0.918 ),

sample('CIToEE_Con_Lambda-11_M-300', 0.2852 , 55922 , 0.778 ),

sample('CIToEE_Con_Lambda-11_M-500', 0.04739 , 25322 , 0.918 ),

sample('CIToEE_Con_Lambda-11_M-800', 0.009645 , 28215 , 0.849 ),

sample('CIToEE_Con_Lambda-13_M-300', 0.2761 , 56620 , 0.764 ),

sample('CIToEE_Con_Lambda-13_M-500', 0.04289 , 26748 , 0.868 ),

sample('CIToEE_Con_Lambda-13_M-800', 0.007542 , 26514 , 0.9 ),

sample('CIToEE_Con_Lambda-15_M-300', 0.2722 , 55469 , 0.782 ),
```

```
sample('CIToEE_Con_Lambda-15_M-500', 0.04034 , 27087 , 0.856 ),
sample('CIToEE_Con_Lambda-15_M-800', 0.00648 , 26689 , 0.897 ),
sample('CIToEE_Con_Lambda-17_M-300', 0.2693 , 55000 , 0.79 ),
sample('CIToEE_Con_Lambda-17_M-500', 0.03915 , 26595 , 0.87 ),
sample('CIToEE_Con_Lambda-17_M-800', 0.005877 , 25983 , 0.919 ),
sample('CIToEE_Con_Lambda-19_M-300', 0.2676 , 54189 , 0.799 ),
sample('CIToEE_Con_Lambda-19_M-500', 0.03821 , 26380 , 0.875 ),
sample('CIToEE_Con_Lambda-19_M-800', 0.005518 , 26022 , 0.916 ),
sample('CIToEE_ITCM5_M-300_TuneZ2star', 0.2621 , 55751 , 0.778 ),
sample('CIToEE_ITCM5_M-500_TuneZ2star', 0.03562 , 26056 , 0.886 ),
sample('CIToEE_ITCM5_M-800_TuneZ2star', 0.004503 , 25672 , 0.929 ),
sample('CIToEE_Des_Lambda-9_M-300', 0.2555 , 55269 , 0.791 ),
sample('CIToEE_Des_Lambda-9_M-500', 0.03886 , 26553 , 0.882 ),
sample('CIToEE_Des_Lambda-9_M-800', 0.008843 , 26328 , 0.917 ),
sample('CIToEE_Des_Lambda-11_M-300', 0.2536 , 56049 , 0.778 ),
sample('CIToEE_Des_Lambda-11_M-500', 0.03516 , 26246 , 0.891 ),
sample('CIToEE_Des_Lambda-11_M-800', 0.005808 , 25814 , 0.933 ),
sample('CIToEE_Des_Lambda-13_M-300', 0.2542 , 55248 , 0.789 ),
sample('CIToEE_Des_Lambda-13_M-500', 0.03418 , 33874 , 0.685 ),
sample('CIToEE_Des_Lambda-13_M-800', 0.004786 , 26239 , 0.916 ),
sample('CIToEE_Des_Lambda-15_M-300', 0.2554 , 56123 , 0.774 ),
sample('CIToEE_Des_Lambda-15_M-500', 0.0338 , 26563 , 0.871 ),
sample('CIToEE_Des_Lambda-15_M-800', 0.004403 , 26297 , 0.91 )]


file_event=open('ci_EVENT.txt','a')
file_error=open('ci_ERROR.txt','a')
if ELECTRON_KI_MUON ==1:
    HISTO_NAME_RECO='W_DiELECTRON__reco'
```

```
        ci=ci_ELECTRON
    else:
        HISTO_NAME_RECO='Dimuon_reco'
        ci=ci_MUON
    for i in range(0,len(ci),3):
        h300=TFile('TRIGGER_EVENT_'+ci[i].name+'.root')
        h500=TFile('TRIGGER_EVENT_'+ci[i+1].name+'.root')
        h800=TFile('TRIGGER_EVENT_'+ci[i+2].name+'.root')
        his_300=h300.Get(HISTO_NAME_RECO)
        his_500=h500.Get(HISTO_NAME_RECO)
        his_800=h800.Get(HISTO_NAME_RECO)
        MY_100_GEV_VALUES=[ ]
        for a in range(3,26):
            re300,re_r_300,re500,re_r_500,re800,re_r_800=0.0,0.0,0.0,0.0,0.0,0.0
            pramod_300=INFORMATION(0.0,0.0)
            pramod_500=INFORMATION(0.0,0.0)
            pramod_800=INFORMATION(0.0,0.0)
            last_number=(a+1)*10+1
            if a==25:last_number=402
            SCALE_300,SCALE_500,SCALE_800=0.0,0.0,0.0
            for ii in range(a*10+1,last_number):
                re300+=his_300.GetBinContent(ii)
                re_r_300+=pow(his_300.GetBinError(ii),2)
                re500+=his_500.GetBinContent(ii)
                re_r_500+=pow(his_500.GetBinError(ii),2)
                re800+=his_800.GetBinContent(ii)
                re_r_800+=pow(his_800.GetBinError(ii),2)
            if re_r_300>0.0:
```

217

```python
        re_r_300=math.sqrt(re_r_300)
    if re_r_500>0.0:
        re_r_500=math.sqrt(re_r_500)
    if re_r_800>0.0:
        re_r_800=math.sqrt(re_r_800)
    SCALE_300=(ci[i].filter_eff*cl_y*ci[i].cross_section)/(ci[i].events)
    SCALE_500=(ci[i+1].filter_eff*cl_y*ci[i+1].cross_section)/(ci[i+1].events)
    SCALE_800=(ci[i+2].filter_eff*cl_y*ci[i+2].cross_section)/(ci[i+2].events)
    pramod_300.RECO_EVENT=re300*SCALE_300
    pramod_300.re_r=re_r_300*SCALE_300
    pramod_500.RECO_EVENT=re500*SCALE_500
    pramod_500.re_r=re_r_500*SCALE_500
    pramod_800.RECO_EVENT=re800*SCALE_800
    pramod_800.re_r=re_r_800*SCALE_800
    MY_100_GEV_VALUES.append([pramod_300,pramod_500,pramod_800])
    WHOLE_THING.append([MY_100_GEV_VALUES,ci[i].name.strip('M-300').strip('TuneZ2star')])
file_event.writelines(fmt_str%('SAMPLE NAME -> Min Mass :',300,400,500,600,700,800,900,1000,

1100,1200,1300,1400,1500,1600,1700,1800,1900,2000,2100,2200,2300,2400,2500))
file_error.writelines(fmt_str%('SAMPLE NAME -> Min Mass :',300,400,500,600,700,800,900,1000,

1100,1200,1300,1400,1500,1600,1700,1800,1900,2000,2100,2200,2300,2400,2500))
file_event.writelines('\n' )
file_error.writelines('\n' )
for stuff in WHOLE_THING:
    newlength=len('SAMPLE NAME -> Min Mass :')-len(stuff[1]+':')
    mero_afnaistring=' '*newlength
    file_event.writelines(stuff[1]+mero_afnaistring+':',)
```

```python
file_error.writelines(stuff[1]+mero_afnaistring+':',)
RECO_TEMP,RECO_ERR_TEMP=0.0,0.0
for m in range(0,23):
    RECO_TEMP,RECO_ERR_TEMP=0.0,0.0
    if m<3:
        for i in range(m,3):
            RECO_TEMP+=stuff[0][i][0].RECO_EVENT
            RECO_ERR_TEMP+=pow(stuff[0][i][0].re_r,2)

        for i in range(3,7):
            RECO_TEMP+=stuff[0][i][1].RECO_EVENT
            RECO_ERR_TEMP+=pow(stuff[0][i][1].re_r,2)

        for i in range(7,23):
            RECO_TEMP+=stuff[0][i][2].RECO_EVENT
            RECO_ERR_TEMP+=pow(stuff[0][i][2].re_r,2)

    if m>2 and m<7:

        for i in range(m,7):
            RECO_TEMP+=stuff[0][i][1].RECO_EVENT
            RECO_ERR_TEMP+=pow(stuff[0][i][1].re_r,2)

        for i in range(7,23):
            RECO_TEMP+=stuff[0][i][2].RECO_EVENT
            RECO_ERR_TEMP+=pow(stuff[0][i][2].re_r,2)
    if m>6:
```

```
        for i in range(m,23):

            RECO_TEMP+=stuff[0][i][2].RECO_EVENT

            RECO_ERR_TEMP+=pow(stuff[0][i][2].re_r,2)

        file_event.writelines(fmt_str1%(RECO_TEMP,))

        file_error.writelines(fmt_str1%(math.sqrt(RECO_ERR_TEMP),))

    file_event.writelines('\n' )

    file_error.writelines('\n' )

file_event.close()

file_error.close()
```

D.5 Python script to produce generator and FULL-simulation config files for Contact Interaction process

```
####################
# Module Name : CONFIG_CREATOR.py
# Author : Pramod Lamichhane
# Date : Aug 2013
# Last modified : Feb 2013
# Purpose : This program creates all the configuration files (gen fragments, GEN config and
FULL-SIM config) required for CI analysis
####################
import os
import string
lambda_=[9000,11000,13000,15000,17000,19000]
ckin1=[300,500,800]
```

ckin2=500

CI_CONFIG2,CI_CONFIG3='',''

CI_CONFIG1='cmsDriver.py Configuration/GenProduction/python/EightTeV/'

FULL_SIM_KI_GEN=input('1 FOR FULL SIM AND 2 FOR VALIDATION STUFF:\t \n')

if FULL_SIM_KI_GEN==1:

    CI_CONFIG2=' -s GEN,SIM,DIGI,L1,DIGI2RAW,HLT:GRun,RAW2DIGI,RECO –pileup

    2012_Startup_50ns_PoissonOOTPU –conditions auto:startup –datatier GEN-SIM-RECO –eventcontent

    AODSIM -n 10 –no_exec'

else:

    CI_CONFIG2=' -s GEN –conditions START53_V6::All –beamspot

    Realistic8TeVCollision –datatier GEN-SIM –eventcontent RAWSIM -n 50000 –no_exec'

CI_CONFIG3=' -s GEN,SIM –conditions START53_V6::All –beamspot

    Realistic8TeVCollision –datatier GEN-SIM –eventcontent RAWSIM -n 20 –no_exec'

FRAGMENT_ONE='''

import FWCore.ParameterSet.Config as cms

source = cms.Source("EmptySource")

from Configuration.Generator.PythiaUEZ2starSettings_cfi import *

generator = cms.EDFilter("Pythia6GeneratorFilter",

pythiaHepMCVerbosity = cms.untracked.bool(False),

maxEventsToPrint = cms.untracked.int32(0),

pythiaPylistVerbosity = cms.untracked.int32(0),

filterEfficiency = cms.untracked.double(1.0),

comEnergy = cms.double(8000.0),

PythiaParameters = cms.PSet(

pythiaUESettingsBlock,

processParameters = cms.vstring(

"',

FRAGMENT_TWO="'

'MSEL = 0 !User defined process',

'MSUB(165) = 1 !CI+g*/Z–>ee',

'MSTP(32) = 4 !forcing a 2–>2 process to 2–>1 process',

"',

CONSTRUCTIVE="'

'RTCM(42) =-1 !Constructive Interference',

"',

DESTRUCTIVE="'

'RTCM(42) =1 !Destructive Interference',

"',

FRAGMENT_MSUB="'

'MSEL = 0 !User defined process',

'MSUB(1) = 1 !Inclusive Z/Gamma* production',

'MSTP(43) = 3 !Both Z0 and gamma*',

'MDME(174,1) = 0 !Z decay into d dbar',

'MDME(175,1) = 0 !Z decay into u ubar',

'MDME(176,1) = 0 !Z decay into s sbar',

'MDME(177,1) = 0 !Z decay into c cbar',

'MDME(178,1) = 0 !Z decay into b bbar',

'MDME(179,1) = 0 !Z decay into t tbar',

'MDME(182,1) = 1 !Z decay into e- e+',

'MDME(183,1) = 0 !Z decay into nu_e nu_ebar',

'MDME(184,1) = 0 !Z decay into mu- mu+',

'MDME(185,1) = 0 !Z decay into nu_mu nu_mubar',

'MDME(186,1) = 0 !Z decay into tau- tau+',

'MDME(187,1) = 0 !Z decay into nu_tau nu_taubar',

",'

ITCM5_0="'

'ITCM(5) = 0 !LL, all upper quarks composite',

'KFPR(165,1) = 11 !e+ e- final state',

",'

LAMBDA_PACHHIKO="'

'ITCM(5) = 2 !LL, all upper quarks composite',

'KFPR(165,1) = 11 !e+ e- final state',

",'

FRAGMENT_THREE="'

parameterSets = cms.vstring('pythiaUESettings',

'processParameters')

)

)

eegenfilter = cms.EDFilter("MCParticlePairFilter",

Status = cms.untracked.vint32(1, 1),

MinPt = cms.untracked.vdouble(30, 30),

MaxEta = cms.untracked.vdouble(3.0, 3.0),

MinEta = cms.untracked.vdouble(-3.0, -3.0),

ParticleCharge = cms.untracked.int32(-1),

ParticleID1 = cms.untracked.vint32(11),

ParticleID2 = cms.untracked.vint32(11)

)

```python
ProductionFilterSequence = cms.Sequence(generator*eegenfilter)
"'

kun_process=input('Which process you want?
MSUB=1,ITCM(5)=2,CI(CONSTRUCTIVE)=3,CI(DESTRUCTIVE=4)')



for b in range(0,len(ckin1)):
    CKIN1=" 'CKIN(1) = "+str(ckin1[b]) +" !Minimum sqrt (s_hat) value'),\n"
    CKIN2=" # 'CKIN(2) = "+str(ckin2)+" !Maximum sqrt (s_hat) value'),"


    if kun_process==1:
        fil=open('DYToEE_MSUB1_M_'+str(ckin1[b])+'_TuneZ2star_8TeV_pythia6_cfi.py','w')
        str_pramod='DYToEE_MSUB1_M_'+str(ckin1[b])+'_TuneZ2star_8TeV_pythia6_cfi.py'
        fil.writelines(FRAGMENT_ONE)
        fil.writelines(FRAGMENT_MSUB)
        fil.writelines(CKIN1)
        fil.writelines(CKIN2)
        fil.writelines(FRAGMENT_THREE)
        fil.close()
        print str_pramod,'is generated and successfully written.'
        os.system(CI_CONFIG1+str_pramod+CI_CONFIG2)
        if FULL_SIM_KI_GEN ==2:
            os.system(CI_CONFIG1+str_pramod+CI_CONFIG3)


    if kun_process==2:
        fil=open('CIToEE_ITCM5_M_'+str(ckin1[b])+'_TuneZ2star_8TeV_pythia6_cfi.py','w')
        str_pramod='CIToEE_ITCM5_M_'+str(ckin1[b])+'_TuneZ2star_8TeV_pythia6_cfi.py'
        fil.writelines(FRAGMENT_ONE)
```

```python
        fil.writelines(FRAGMENT_TWO)

        fil.writelines(ITCM5_0)

        fil.writelines(CKIN1)

        fil.writelines(CKIN2)

        fil.writelines(FRAGMENT_THREE)

        fil.close()

        print str_pramod,'is generated and successfully written.'

        os.system(CI_CONFIG1+str_pramod+CI_CONFIG2)


        if FULL_SIM_KI_GEN ==2:

            os.system(CI_CONFIG1+str_pramod+CI_CONFIG3)


if kun_process==3:

    for a in range(0,len(lambda_)):

        ENERGY=" 'RTCM(41) = "+str(lambda_[a])+" !Lambda = "+str((lambda_[a])/1000)

        +" TeV',"

        fil=open('CIToEE_Con_Lambda_'+str((lambda_[a])/1000)+'_M_'

        +str(ckin1[b])+'_TuneZ2star_8TeV_pythia6_cfi.py','w')

        str_pramod='CIToEE_Con_Lambda_'+str((lambda_[a])/1000)+'_M_'

        +str(ckin1[b])+'_TuneZ2star_8TeV_pythia6_cfi.py'

        fil.writelines(FRAGMENT_ONE)

        fil.writelines(FRAGMENT_TWO)

        fil.writelines(CONSTRUCTIVE)

        fil.writelines(ENERGY)

        fil.writelines(LAMBDA_PACHHIKO)

        fil.writelines(CKIN1)

        fil.writelines(CKIN2)

        fil.writelines(FRAGMENT_THREE)
```

```
            fil.close()

            print str_pramod,'\t is generated and successfully written.'

            os.system(CI_CONFIG1+str_pramod+CI_CONFIG2)



            if FULL_SIM_KI_GEN ==2:

                os.system(CI_CONFIG1+str_pramod+CI_CONFIG3)



if kun_process==4:

    for a in range(0,len(lambda_)):

        ENERGY=" 'RTCM(41) = "+str(lambda_[a])+" !Lambda = "+str((lambda_[a])/1000)

        +" TeV',"



        fil=open('CIToEE_Des_Lambda_'+str((lambda_[a])/1000)+'_M_'

        +str(ckin1[b])+'_TuneZ2star_8TeV_pythia6_cfi.py','w')



        str_pramod='CIToEE_Des_Lambda_'+str((lambda_[a])/1000)+'_M_'

        +str(ckin1[b])+'_TuneZ2star_8TeV_pythia6_cfi.py'



        fil.writelines(FRAGMENT_ONE)

        fil.writelines(FRAGMENT_TWO)

        fil.writelines(DESTRUCTIVE)

        fil.writelines(ENERGY)

        fil.writelines(LAMBDA_PACHHIKO)

        fil.writelines(CKIN1)

        fil.writelines(CKIN2)
```

```
        fil.writelines(FRAGMENT_THREE)

        fil.close()

        print str_pramod,'\t is generated and successfully written.'

        os.system(CI_CONFIG1+str_pramod+CI_CONFIG2)


        if FULL_SIM_KI_GEN ==2:

            os.system(CI_CONFIG1+str_pramod+CI_CONFIG3)


    globals().clear()
```

BIBLIOGRAPHY

[1] S. Chatrchyan *et al.*, "Observation of a new boson at a mass of 125 gev with the cms experiment at the lhc," *Phys.Lett.*, vol. B716, pp. 30–61, 2012.

[2] G. Aad *et al.*, "Observation of a new particle in the search for the standard model higgs boson with the atlas detector at the lhc," *Phys.Lett.*, vol. B716, pp. 1–29, 2012.

[3] A. Aguilar-Arevalo *et al.*, "Evidence for neutrino oscillations from the observation of anti-neutrino(electron) appearance in a anti-neutrino(muon) beam," *Phys.Rev.*, vol. D64, p. 112007, 2001.

[4] R. Peccei, "QCD, strong CP and axions," *J.Korean Phys.Soc.*, vol. 29, pp. S199–S208, 1996.

[5] E. Fermi, "Versuch einer theorie der β-strahlen. i," *Zeitschrift für Physik*, vol. 88, no. 3-4, pp. 161–177, 1934.

[6] S. Weinberg, "What is quantum field theory, and what did we think it is?," 1996.

[7] http://commons.wikimedia.org/wiki/.

[8] P. W. Higgs, "Spontaneous Symmetry Breakdown without Massless Bosons," *Phys.Rev.*, vol. 145, pp. 1156–1163, 1966.

[9] R. Brout and F. Englert, "Spontaneous symmetry breaking in gauge theories: A Historical survey," 1998.

[10] K. G. Wilson, "Confinement of quarks," *Phys. Rev. D*, vol. 10, pp. 2445–2459, Oct 1974.

[11] T. Mohaupt, "Introduction to string theory," *Lect.Notes Phys.*, vol. 631, pp. 173–251, 2003.

[12] G. Belanger, U. Ellwanger, J. Gunion, Y. Jiang, and S. Kraml, "Two Higgs Bosons at the Tevatron and the LHC?," 2012.

[13] J. Pati, A. Salam, and J. Strathdee, "Are quarks composite?," *Phys. Lett. B*, vol. 59, p. 265, 1975.

[14] J. Pati, A. Salam, and J. Strathdee, "Are quarks composite?," *Int. Centre Theor. Phys.*, vol. IC/75/139, addendum, 1975.

[15] S. Drell and T. Yan, "Massive lepton-pair production in hadron-hadron collisions at high energy," *Phys. Rev. Lett.*, vol. 25, p. 316, 1970.

[16] E. Eichten, K. Lane, and M. Peskin, "New tests for quark and lepton substructure," *Phys. Rev. Lett.*, vol. 50, p. 811, 1983.

[17] E. Eichten *et al.*, "Supercollider physics," *Rev. Modern Phys.*, vol. 56, p. 579, 1984.

[18] S. Chatrchyan *et al.*, "Search for contact interactions using the inclusive jet p_T spectrum in pp collisions at $\sqrt{s} = 7$ TeV," *Phys.Rev.*, vol. D87, p. 052017, 2013.

[19] S. Chatrchyan *et al.*, "Search for new physics in final states with a lepton and missing transverse energy in pp collisions at the LHC," *Phys.Rev.*, vol. D87, p. 072005, 2013.

[20] S. Schael *et al.*, "Fermion pair production in e^+e^- collisons at 189–209 GeV and constraints on physics beyond the standard model," *Eur. Phys. J. C*, vol. 49, p. 411, 2007.

[21] J. Abdallah *et al.*, "Measurement and interpretation of fermion-pair production at lep energies above the z resonance," *Eur. Phys. J. C*, vol. 45, p. 589, 2006.

[22] M. Acciarri *et al.*, "Search for manifestations of new physics in fermion-pair production at lep," *Phys. Lett. B*, vol. 489, p. 81, 2000.

[23] K. Ackerstaff *et al.*, "Production of fermion-pair events in e$^+$e$^-$ collisions at 161 gev centre-of-mass energy," *Phys. Lett. B*, vol. 391, p. 221, 1997.

[24] G. Abbiendi *et al.*, "Tests of the standard model and constraints on new physics from measurements of fermion-pair production at 189–209 GeV at lep," *Eur. Phys. J. C*, vol. 33, p. 173, 2004.

[25] F. Aaron *et al.*, "Search for contact interactions in e$^\pm$p collisions at hera," *Phys. Lett. B*, vol. 705, p. 52, 2011.

[26] S. Chekanov *et al.*, "Search for contact interactions, large extra dimensions and finite quark radius in ep collisions at hera," *Phys. Lett. B*, vol. 591, p. 23, 2004.

[27] F. Abe *et al.*, "Search for new gauge bosons in p̄p collisions at $\sqrt{s} = 1.8$ TeV," *Phys. Rev. Lett.*, vol. 68, p. 1463, 1992.

[28] F. Abe *et al.*, "Limits on quark-lepton compositeness scales from dileptons produced in 1.8 tev pp̄ collisions," *Phys. Rev. Lett.*, vol. 79, p. 2198, 1997.

[29] T. Affolder *et al.*, "Search for quark-lepton compositeness and a heavy w boson using the ev channel in pp̄ collisions at $\sqrt{s} = 1.8$ TeV," *Phys. Rev. Lett.*, vol. 87, p. 231803, 2001.

[30] A. Abulencia *et al.*, "Search for z$'$ → e$^+$e$^-$ using dielectron mass angular distribution," *Phys. Rev. Lett.*, vol. 96, p. 211801, 2006.

[31] B. Abbott *et al.*, "Measurement of the high-mass drell-yan cross section and limits on quark-electron compositeness scales," *Phys. Rev. Lett.*, vol. 82, p. 4769, 1999.

[32] V. Abazov *et al.*, "Measurement of dijet angular distribution at $\sqrt{s} = 1.96$ TeV and searches for quark compositeness and extra spatial dimensions," *Phys. Rev. Lett.*, vol. 103, p. 191803, 2009.

[33] G. Aad *et al.*, "Search for quark contact interactions in dijet angular distributions in pp collisions at $\sqrt{s} = 7$ TeV measured with the atlas detector," *Phys. Lett. B*, vol. 694, p. 327, 2011.

[34] G. Aad *et al.*, "Search for contact interactions in dimuon events from pp collision at $\sqrt{s} = 7$ TeV with the atlas detector," *Phys. Rev. D*, vol. 84, p. 011101, 2011.

[35] G. Aad *et al.*, "Search for contact interactions in dimuon events from pp collisions at $\sqrt{s} = 7$ TeV with the atlas detector," *Phys. Lett. B*, vol. 712, p. 40, 2012.

[36] G. Aad *et al.*, "Search for contact interactions and large extra dimensions in dilepton events from pp collisions at $\sqrt{s} = 7$ TeV with the ATLAS detector," *Phys. Rev. D*, vol. 87, p. 015010, 2013.

[37] G. Aad *et al.*, "The ATLAS Experiment at the CERN Large Hadron Collider," *JINST*, vol. 3, p. S08003, 2008.

[38] R. P. Feynman, "Very high-energy collisions of hadrons," *Phys. Rev. Lett.*, vol. 23, pp. 1415–1417, Dec 1969.

[39] C. G. Callan and D. J. Gross, "Bjorken scaling in quantum field theory," *Phys. Rev. D*, vol. 8, pp. 4383–4394, Dec 1973.

[40] http://cernmasterclass.uio.no/2013/webpages/testpage/version2013/zpath_protoncollisions.htm.

[41] R. D. Ball, S. Carrazza, L. Del Debbio, S. Forte, J. Gao, *et al.*, "Parton Distribution Benchmarking with LHC Data," *JHEP*, vol. 1304, p. 125, 2013.

[42] E. D. Bloom, D. H. Coward, H. Destaebler, J. Drees, G. Miller, L. W. Mo, R. E. Taylor, M. Breidenbach, J. I. Friedman, G. C. Hartmann, and H. W. Kendall, "High-Energy Inelastic e-p Scattering at $6^{\deg}$ and $10^{\deg}$," *Physical Review Letters*, vol. 23, pp. 930–934, Oct. 1969.

[43] A. Martin, W. Stirling, R. Thorne, and G. Watt, "Parton distributions for the LHC," *Eur.Phys.J.*, vol. C63, pp. 189–285, 2009.

[44] P. Nadolsky, H. Lai, Q. Cao, J. Huston, J. Pumplin, *et al.*, "Implications of CTEQ global analysis for collider observables," *Phys. Rev. D*, vol. 78, p. 013004, 2008.

[45] R. D. Ball, L. Del Debbio, S. Forte, A. Guffanti, J. I. Latorre, *et al.*, "A first unbiased global NLO determination of parton distributions and their uncertainties," *Nucl.Phys.*, vol. B838, pp. 136–206, 2010.

[46] J. Kretzschmar, "Proton Structure Measurements and the HERAPDF fit," 2009.

[47] S. Alekhin, J. Blumlein, S. Klein, and S. Moch, "The 3, 4, and 5-flavor NNLO Parton from Deep-Inelastic-Scattering Data and at Hadron Colliders," *Phys.Rev.*, vol. D81, p. 014032, 2010.

[48] P. Jimenez-Delgado and E. Reya, "Variable Flavor Number Parton Distributions and Weak Gauge and Higgs Boson Production at Hadron Colliders at NNLO of QCD," *Phys.Rev.*, vol. D80, p. 114011, 2009.

[49] H. Lai, M. Guzzi, J. Huston, Z. Li, P. Nadolsky, and othePDFrs, "New parton distributions for collider physics," *Phys. Rev. D*, vol. 82, p. 074024, 2010.

[50] T. Sjöstrand, S. Mrenna, and P. Skand, "Pythia 6.4 Physics and Manual," *JHEP*, vol. 05, p. 026, 2006.

[51] S. Frixione and B. Webber, "Matching nlo qcd computations and parton shower simulations," *JHEP*, vol. 06, p. 029, 2002.

[52] P. Nason, "A new method for combining nlo qcd with shower Monte Carlo algorithms," *JHEP*, vol. 11, p. 040, 2004.

[53] S. Frixione, P. Nason, and C. Oleari, "Matching nlo qcd computations with Parton Shower simulations: the powheg method," *JHEP*, vol. 11, p. 070, 2007.

[54] S. Alioli *et al.*, "Nlo vector-boson production matched with shower in powheg," *JHEP*, vol. 07, p. 060, 2008.

[55] S. Frixione, P. Nason, and C. Oleari, "Matching NLO QCD computations with Parton Shower simulations: the POWHEG method," *JHEP*, vol. 0711, p. 070, 2007.

[56] G. Corcella, I. Knowles, G. Marchesini, S. Moretti, K. Odagiri, *et al.*, "HERWIG 6: An Event generator for hadron emission reactions with interfering gluons (including supersymmetric processes)," *JHEP*, vol. 0101, p. 010, 2001.

[57] C. Carloni Calame, G. Montagna, O. Nicrosini, and A. Vicini, "Precision electroweak calculation of the production of a high transverse-momentum leptonPDF pair at hadron colliders," *JHEP*, vol. 0710, p. 109, 2007.

[58] C. Gerber *et al.*, "Tevatron-for-LHC Report: Top and Electroweak Physics," 2007.

[59] C. Carloni Calame, G. Montagna, O. Nicrosini, and A. Vicini, "Precision electroweak calculation of the charged current Drell-Yan process," *JHEP*, vol. 0612, p. 016, 2006.

[60] C. Buttar, S. Dittmaier, V. Drollinger, S. Frixione, A. Nikitenko, *et al.*, "Les houches physics at TeV colliders 2005, standard model and Higgs working group: Summary report," 2006.

[61] C. Carloni Calame, G. Montagna, O. Nicrosini, and M. Treccani, "Multiple photon corrections to the neutral-current Drell-Yan process," *JHEP*, vol. 0505, p. 019, 2005.

[62] C. Carloni Calame, G. Montagna, O. Nicrosini, and M. Treccani, "Higher order QED corrections to W boson mass determination at hadron colliders," *Phys. Rev. D*, vol. 69, p. 037301, 2004.

[63] http://hepoutreach.syr.edu/Index/accelerator_science/accel_overview.html.

[64] http://lhc-machine-outreach.web.cern.ch/lhc-machine outreach/images/complex/.

[65] http://en.wikipedia.org/wiki/File:CMS_Slice.gif.

[66] V. Karimäki, M. Mannelli, P. Siegrist, H. Breuker, A. Caner, R. Castaldi, K. Freuden-
reich, G. Hall, R. Horisberger, M. Huhtinen, and A. Cattai, *The CMS tracker system
project: Technical Design Report.* Technical Design Report CMS, Geneva: CERN,
1997.

[67] "Addendum to the CMS tracker TDR," 2000.

[68] http://www.physics.purdue.edu/particle/cms/plaquette_assembly.

[69] *The CMS electromagnetic calorimeter project: Technical Design Report.* No. CMS-
TDR-004 in Technical Design Report CMS, Geneva: CERN, 1997.

[70] http://www.lepp.cornell.edu/Research/EPP/CMS/EventReconstruction.html.

[71] *The CMS hadron calorimeter project: Technical Design Report.* No. CMS-TDR-002 in
Technical Design Report CMS, Geneva: CERN, 1997.

[72] *The CMS muon project: Technical Design Report.* No. CMS-TDR-003 in Technical
Design Report CMS, Geneva: CERN, 1997.

[73] "Proceedings of 11th Workshop on Resistive Plate Chambers and Related Detectors
(RPC2012)," *PoS*, vol. RPC2012, 2012.

[74] T. C. Collaboration, *CMS TriDAS project: Technical Design Report, Volume 1: The
Trigger Systems.* No. CMS-TDR-006-1 in Technical Design Report CMS, 2000.

[75] T. C. Collaboration, *CMS TriDAS project: Technical Design Report, Volume 2: Data
Acquisition and High Level Trigger.* No. CMS-TDR-006-2 in Technical Design Report
CMS, 2002.

[76] D. Hufnagel, "The architecture and operation of the CMS Tier-0," *J.Phys.Conf.Ser.*,
vol. 331, p. 032017, 2011.

[77] A. Read, "Presentation of search results: the cls technique," *J. Phys. G*, vol. 28, p. 2693, 2002.

[78] "Electron performance with 19.6 fb^{-1} of data collected at $\sqrt{s} = 8$ TeV with the CMS detector.," 2013.

[79] "Electron Reconstruction and Identification at s = 7 TeV," 2010.

[80] M. Anderson *et al.*, "A review of clustering algorithms and energy corrections in the electromagnetic calorimeter," no. CMS IN 2010/008, 2008.

[81] W. Adam, R. Frühwirth, A. Strandlie, and T. Todorov, "RESEARCH NOTE FROM COLLABORATION: Reconstruction of electrons with the Gaussian-sum filter in the CMS tracker at the LHC," *Journal of Physics G Nuclear Physics*, vol. 31, p. 9, Sept. 2005.

[82] "Search for resonances in the dilepton mass distribution in pp collisions at sqrt(s) = 8 tev," Tech. Rep. CMS-PAS-EXO-12-061, CERN, Geneva, 2013.

[83] S. Agostinelli *et al.*, "GEANT4: A Simulation toolkit," *Nucl.Instrum.Meth.*, vol. A506, pp. 250–303, 2003.

[84] S. Frixione and B. R. Webber, "The MC and NLO 3.4 Event Generator," 2008.

[85] "Search for large extra spatial dimensions in dielectron production with the cms detector," Tech. Rep. CMS-PAS-EXO-12-031, CERN, Geneva, 2013.

[86] M. Whalley, D. Bourilkov, and R. Group, "The Les Houches accord PDFs (LHAPDF) and LHAGLUE," 2005.

[87] J. Pumplin, D. Stump, R. Brock, D. Casey, J. Huston, *et al.*, "Uncertainties of predictions from parton distribution functions. 2. The Hessian method," *Phys.Rev.*, vol. D65, p. 014013, 2001.

[88] CMS Collaboration, "Absolute calibration of the luminosity measurement at cms: Winter 2012 update," *CMS Physics Analysis Summary* CMS-PAS-SMP-12-008, 2012.

[89] V. Khachatryan *et al.*, "Measurements of Inclusive W and Z Cross Sections in pp Collisions at $\sqrt{s} = 7$ TeV," *JHEP*, vol. 1101, p. 080, 2011.

[90] S. Chatrchyan *et al.*, "Energy calibration and resolution of the CMS electromagnetic calorimeter in pp collisions at $\sqrt{s} = 7$ TeV," 2013.

[91] V. Chalotra, A. Bhasin, A. Gupta, and S. S. Sambyal, "Hep cluster first step towards grid computing," *CoRR*, vol. abs/1210.7626, 2012.

[92] L. Sala, "CRAB: Distributed analysis tool for CMS," *AIP Conf.Proc.*, vol. 1504, pp. 1009–1012, 2009.

[93] G. Cowan, K. Cranmer, E. Gross, and O. Vitells, "Asymptotic formulae for likelihood-based tests of new physics," *European Physical Journal C*, vol. 71, p. 1554, Feb. 2011.

[94] B. Mistlberger and F. Dulat, "Limit setting procedures and theoretical uncertainties in Higgs boson searches," 2012.

[95] G. Ranucci, "The profile likelihood ratio and the look elsewhere effect in high energy physics," *Nuclear Instruments and Methods in Physics Research A*, vol. 661, pp. 77–85, Jan. 2012.

[96] R. Barate *et al.*, "Search for the standard model Higgs boson at LEP," *Phys.Lett.*, vol. B565, pp. 61–75, 2003.

ABSTRACT

SEARCH FOR CONTACT INTERACTIONS IN THE DIELECTRON CHANNEL IN P-P COLLISIONS AT $\sqrt{s} = 8$ TeV AT CMS

by

PRAMOD LAMICHHANE

Advisor: Dr. Paul E. Karchin

Major : Experimental Particle Physics

Degree : Doctor of Philosophy

A possible explanation of mass hierarchy, which is not explained by the Standard Model, is that quarks and leptons are composite objects made of more fundamental particles known as preons. The existence of preons will be manifest as a four fermion contact interaction in the annihilation of a quark and anti-quark, in a p-p collision, producing positron-electron pairs. At high mass, such pairs are also produced from off-shell Z and γ bosons. This thesis provides a detailed discussion of the analysis strategy to study these processes using the Compact Muon Solenoid Experiment at the Large Hadron Collider. The study utilizes data recorded in 2012 at $\sqrt{s} = 8$ TeV, corresponding to an integrated luminosity of 19.6 fb^{-1}. The dielectron mass spectrum above 300 GeV shows no significant deviation from the prediction of the Standard Model. In the framework of the left-left iso-scalar model of eeqq contact interactions, 95% CL lower limits on the energy scale parameter are found for destructive (13.1 TeV) and constructive (18.3 TeV) interference between the contact and standard model amplitudes. These limits are the most stringent to date.

AUTOBIOGRAPHICAL STATEMENT

PRAMOD LAMICHHANE

I am interested in the frontier of the fundamental physics since my college days, and as, the deepest exploration of the fundamental physics can be achieved by the experiments offered in the high energy particle physics, I decided to start my career in this fascinating field by joining Wayne State University in the fall of 2008 as a graduate student. My association with the Compact Muon Solenoid (CMS) experiment at the European Center for Nuclear Research (CERN), Switzerland began in the summer of 2009 as a part of the research project which was a search for substructure of quarks and leptons. After a successful completion of my course works as well as acquired a two years teaching experience in the undergraduate Physics Lab in Physics department of Wayne State University, In the summer of 2010, I moved to Fermilab to start an additional research project at siDet, Fermilab, as a part of the Phase II Tracker Upgrade project that involved the characterization of various types of the silicon sensors followed by the beam test. As a part of my Ph.D program, I received my M.S. degree in Physics from Wayne State University in 2011. I have been stationed in Fermilab since the summer of 2010. The work done using CMS data from 2011 "Search for Contact Interactions in $\mu^+\mu^-$ Events in pp Collisions at $\sqrt{s} = 7$ TeV" is published in Phys. Rev. D 87, 032001 (2013). I had an opportunity to present this work at April meeting of the American Physics Society (APS) in Atlanta, Georgia in April 2012 as a part of my thesis work. In 2012, I had a leading role to establish the analysis in the dielectron channel while taking the full responsibility in the dimuon channel. The work "Search for contact interactions in the di-lepton spectra in pp collisions at $\sqrt{s} = 8$ TeV" is currently in the approval process. I have presented this work in "Meeting of the American Physical Society", Division of Particles and Fields, University of California at Santa Cruz in Aug 2013.

During my Ph.D career, I had opportunities to acquire the computing technique in grid computing system as well as software skills in C++, PYTHON, ROOT in the Linux based operating systems along with the good understanding of the general Physics courses. I have been awarded Universities Research Association, Inc (URA) visiting scholar award at Fermilab in Fall of 2012. In summer of 2012 and 2013, I received LPC Short Term Visitor Award at Fermilab for the recognition of my contribution in research towards the CMS experiment. In addition to that, I received the Summer 2013 dissertation fellowship by the graduate school of Wayne State University.

I'm interested in exploring the physics beyond the standard model with collider experiments. With the unprecedented center of mass energy in the upcoming collision, this is the most exciting time with the golden opportunity for the CMS experiment, to uncover the secret of the new physics lurking around. I appreciate the potential of discovery or ruling out of the various models using the upcoming LHC data in the near future. I want to pursue my career in this fascinating field of experimental particle physics searching for new phenomena.