Modeling the Underlying Event using Herwig++

PHASM201 Project Report

1st Supervisor: Prof. Jon Butterworth
2nd Supervisor: Dr Ben Waugh

Anjum Sayed
March 2009

Table of Contents

List of Figures

List of Tables

Acknowledgements

I'd like to thank my supervisors Prof. Jon Butterworth and Dr Ben Waugh for their guidance in this project. I would also like to thank Alex Richards, James Monk and Andy Buckley for their technical assistance with the software and physics involved in this project. Special thanks to Dr Glen Cowan from Royal Holloway for his help with the statistics involved with this project.

Website

Uploads of all the generated plots discussed in this report and be found at:

http://www.ucl.ac.uk/~zcapbb4/

Abstract

With the Large Hadron Collider (LHC) expected to have first collisions soon, at energies never before seen in a particle collider, the fundamental forces of nature will be probed at distances smaller than ever before. Therefore it is very important that we fully understand what we have already seen with current experiments, so that it will be possible to make optimal use of new data from the LHC. The focus of this project is examining the "underlying event", which contains all activity in a hadronic collision that is not related to the signal particles from the hard process [1]. Presently, the underlying event is poorly understood, and with the higher energies and luminosities at the LHC, the underlying event will become even more apparent, making it important to properly understand it.

This project uses underlying event data from proton–antiproton collisions at the CDF detector at Tevatron (the most powerful particle collider currently in operation) and compares it to results generated from Monte Carlo models; Fortran Herwig (with the Jimmy plug-in) and the newer Herwig++. The goal is to confirm current observations regarding the underlying event and to investigate the difference in physics between the two models. As a result, an improved tune for Herwig++ was found using recent Drell-Yan data, and confirmed the results the authors of the generator found using older QCD 2-to-2 data in a different experiment at CDF. This allowed Monte Carlo data to match CDF data more accurately, which will help allow the main events of interest to be distinguished easier from the background created by the underlying event at future experiments.

1 Background

1.1 Overview of the Standard Model

The Standard Model is the most popular theory of elementary particles. It aims to explain all of the phenomena of particle physics, except those due to gravity, in terms of a small number of elementary particles. Elementary particles are defined as being point-like, without internal structure or excited states. Elementary particles can be characterised by, amongst other things, its mass, electric charge and its spin [2]. Spin is a permanent angular momentum possessed by all particles in quantum theory, even when they are at rest. Spin is measured in units of $s\hbar$ (Planck's constant), where s is the spin quantum number or spin for short.

Particles with half integer spin are called fermions and those with integer spin are called bosons. There are three families of elementary particles in the standard model: two spin-½ families of fermions called leptons and quarks; and one family of spin-1 bosons, and at least one other spin-0 particle called the Higgs boson, which is postulated to explain the origin of mass within the Standard Model.

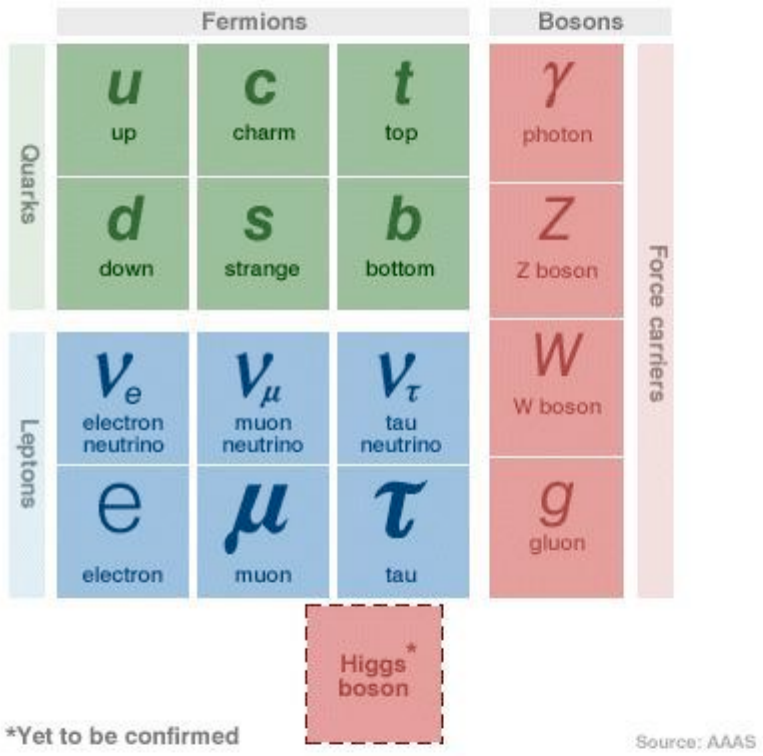


Figure 1.1: Table of Elementary Particles (Source: AAAS)

One of the most familiar particles is the electron, which is a part of the family of leptons, which we know is bound in atoms by electromagnetic interaction, one of the four forces observed in nature, and is described well by Maxwell's equations. Another lepton is the neutrino, which is a product of nuclear β-decay. Each lepton such as the electron has an associated neutrino, such as the electron neutrino. The force responsible for β-decay is the weak force (another of the four fundamental forces) and is the mediator for all interactions involving neutrinos. The third of the fundamental forces is the strong force which describes interactions between coloured (colour is another quantum number, analogous to electric charge) components such as quarks, which do not exist freely but are the building blocks of all hadrons such as protons, neutrons and pions and is mediated by gluons. The final force is gravity, which the Standard Model fails to describe.

As in the example of β-Decay, weak interactions and electromagnetic interactions are related, and can be unified to electroweak interactions, in the same way Maxwell unified electric and magnetic fields. The Standard Model is initially formulated with four massless particles which carry these forces. A process of symmetry breaking gives mass to three of these four particles – the W^+, the W^- and the Z^0, which particles are the carriers of the weak force. The particle that remains massless is the photon, which is the carrier of the electromagnetic force. This theory is termed as Quantum Electrodynamics (QED), as this a quantum version of the classical electrodynamics. This theory is extended to describe the strong colour charge and that is termed Quantum Chromodynamics (QCD). The carriers of the colour force are eight massless coloured gluons, and just like the quarks, they cannot be observed in isolation [3]. The electroweak theory together with QCD forms the standard model we know today.

We have discussed the force carrying gauge bosons, but the particles that matter is made up of are the fermions; the leptons and quarks. The leptons consist of electrons, muons and the tau and their associated neutrinos. There are six different flavours of quarks: up (u), down (d), charm (c), strange (s), top (t) and bottom (b). The u, c and t quarks have electric charge +2/3 and the d, s and b quarks have -1/3. These six quarks come in three different colours (red, green and blue), and there is an associated antiquark for all of them, resulting in 36 different quarks which in combinations make hadrons. The quarks and gluons constituting the hadrons are termed as partons.

The Standard Model is the simplest summary of current knowledge and is very consistent with experimental data. However, it does have some shortcomings, where the theory needs to be severely "tweaked" to match some observations. It does not provide a unified theory of all the forces since the Standard model cannot describe gravity. It also cannot explain why particles have mass. One explanation is the Higgs Mechanism, which is another boson which provides the symmetry breaking required to give particles mass but, to this date the Higgs boson has not been observed in a particle collider [4]. Also, the Standard Model requires that the neutrino has zero mass, however there has been experimental evidence that it has mass [5]. Cosmologists have found that the Standard Model only accounts for 4% of the visible matter.

An alternative theory which is highly popular is Supersymmetry, which is an extension to the Standard Model where every fermion has a superpartner which is a boson and vice versa. Since all particles are fermions, and all force carriers and bosons, this symmetry unifies matter and force [6]. This also attempts to fix the visible matter problem by postulating the existence of dark matter (23%) and dark energy (73%) [7]. However, good evidence for Supersymmetry can only be seen at energies higher than those at current particle accelerators. Another alternative is String Theory, which tries to unify gravity with the Standard model; however, the distances that current generation particle accelerators probe are too large to observe any evidence of String Theory, and require energies many times greater than those of the LHC.

1.2 The Particle Accelerator

To observe microscopic phenomena such as a biological cell, the projectile probing it needs a wavelength at least as small as the cell itself. In an optical microscope, the projectiles are photons. To achieve higher resolutions, a projectile with a smaller wavelength is needed, for example an electron in an electron microscope. Therefore, to probe a proton, a projectile with a wavelength smaller than the effective radius of the proton is needed. The wavelength of a particle is given by the de Broglie relation.

$$p = \frac{\hbar}{\lambda}$$

Therefore by increasing the momentum of a projectile (and hence its energy) the wavelength decreases allowing smaller distances to be probed, increasing the resolution of your “microscope”. This is the main aim of particle accelerators.

In early particle physics, fixed target experiments were used. This is where the projectile is fired at a stationary target. However, these are now outdated as it is difficult to achieve a high centre of mass energy, which is required to produce new and heavy particles. The centre of mass energy is the energy available to create new particles. In a fixed target experiment, most of the projectile energy reappears as kinetic energy of the final state particles, and is therefore unavailable for creating new particles, and the centre of mass energy varies as $E_{CM} = (E_L)^{1/2}$ (in the massless limit) where E_L is the laboratory energy.

In modern particle physics, colliding beam accelerators are used where two beams of particles travelling in almost opposite directions, and made to collide at a small or zero crossing angle. For the proton – antiproton case, the masses of particles are equal, so the centre of mass energy is $E_{CM} = 2E_L$. This means that the centre of mass energy increases linearly with the laboratory energy, where in the fixed target experiment, it gets increasingly difficult to attain high centre of mass energies.

Most particle accelerators such as Tevatron and the LHC are built as large circular rings where the colliding particles travel around at speeds close to the speed of light. There are powerful magnets placed all the way around the beam tubes which accelerate the particles, until the two beam tubes overlap in a place where the detector is located. The particles are usually accelerated in bunches to achieve high luminosities and to make as many final state particles as possible. The luminosity in proton-antiproton colliders such as Tevatron is usually limited by the production of antimatter (antiprotons in this case) which usually takes a long time to produce a small amount and is kept in storage rings.

Detecting the exotic particles made in these high energy collisions is very difficult due to their short lifetimes and because there are so few of them. All detectors try to manipulate the particles to see how they interact with matter and strong electromagnetic fields. This means that the detectors need to be very large to be able to capture as many particles as possible, and to gather as much information as possible. Detectors consist of many components, each specialised to detecting certain types of particles, which can be divided into two groups; tracking devices and calorimeters.

Tracking devices reveal the tracks of electrically charged particles through the tracks they leave by ionising matter. By measuring how the trajectory of a particle changes within the detector, its speed and lifetime, a lot of useful information can be deduced enabling physicists to determine what the particles are and what the initial events were which created them. In modern tracking devices, the particles interact with matter to create electrical pulses which are recorded, and used to recreate the trajectory by a computer program. Two specialised tracking devices are vertex chambers, which are located close to the interaction point, and muon chambers which are located on the outer parts of the detector as muons are the only charged particles able to travel through metres of material.

The two main techniques to build tracking devices are gaseous chambers and semiconductor devices. In gaseous chambers, the gas is ionised by the charged particles, and the resulting ions or electrons are collected by electrodes in the form of wires in electric fields. In drift chambers, the position of the track is found by timing how long the electrons take to reach an anode wire, measured from the moment that the charged particle passed through. This results in a spatial resolution of 50-100 μm. In semiconductor devices, the charged particle creates free electrons and holes as it passes through the semiconductor. The devices are usually made of silicon divided into strips or pixels and the typical resolution is 10 μm [7].

Calorimeters are the parts of the detector which measure the energy of particles by stopping them and measuring the energy released. They are different from other detectors in that the nature of the particle is changed by the detector and they can detect neutral particles such as photons and neutrons which are invisible to tracking devices, but can be seen by their energy deposits in the calorimeters. There are two types of calorimeter: Hadronic calorimeters (HCAL) and Electromagnetic calorimeters (ECAL). They both use different stopping materials depending on what they are stopping. Most calorimeters consist of layers of an absorber (e.g. lead) and detector (e.g. lead glass or liquid argon). These are known as sampling calorimeters. During the absorption process, the particle will interact with the absorber, generating secondary particles which in turn generate more particles creating a shower or cascade which are then detected. The number of particles created is a measure of how much energy the initial particle had.

Other detectors include instruments to measure Cherenkov and Transition radiation to measure particle velocities.

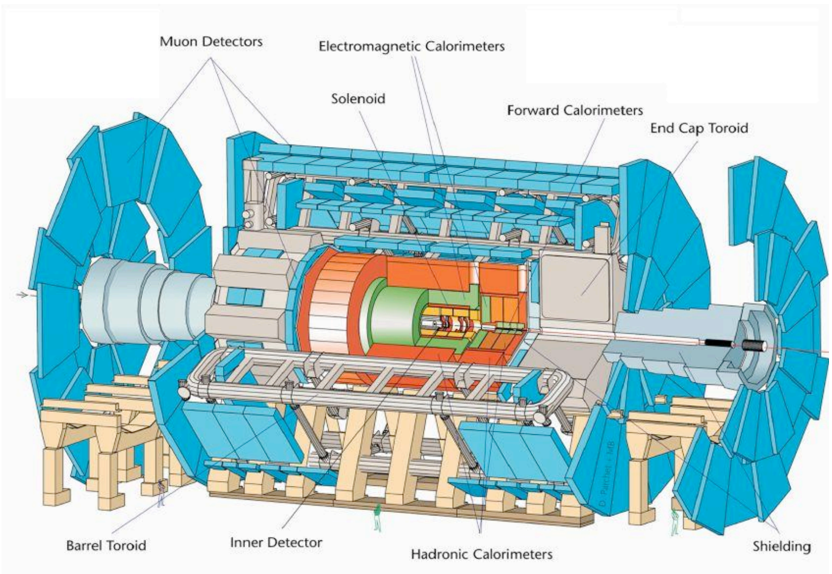


Figure 1.2: A diagram of the ATLAS detector at the LHC (Source: CERN)

1.2.1 Large Hadron Collider (LHC)

The Large Hadron Collider is the most recent particle accelerator built with state of the art technology designed to collide protons and also lead ions. It was built in the 27km circumference tunnel at CERN which was previously home to LEP (Large Electron Positron collider) which remained operational until the year 2000. The LHC is made up of a total of 9593 magnets and 1232 main dipoles which focus and steer the proton bunches around the ring [7]. In a proton accelerator like the LHC, the maximum energy that can be achieved is directly proportional to the strength of the dipole field, given a specific acceleration circumference. At the LHC, the dipole magnets are superconducting electromagnets able to provide at magnetic field of 8.3 T over their length. These superconducting magnets require to be cooled to a temperature of 1.9 K (-271.3 °C) to reach this strength. As a result these magnets require a complex cryogenic system pumping superfluid helium to keep the magnets cool enough to remain at 8.3 T. No room temperature magnets are able to reach 8.3 T efficiently, hence the complex superconductor approach. These highly powerful magnets accelerate protons up to 7 TeV, resulting in a total centre of mass energy of 14 TeV.

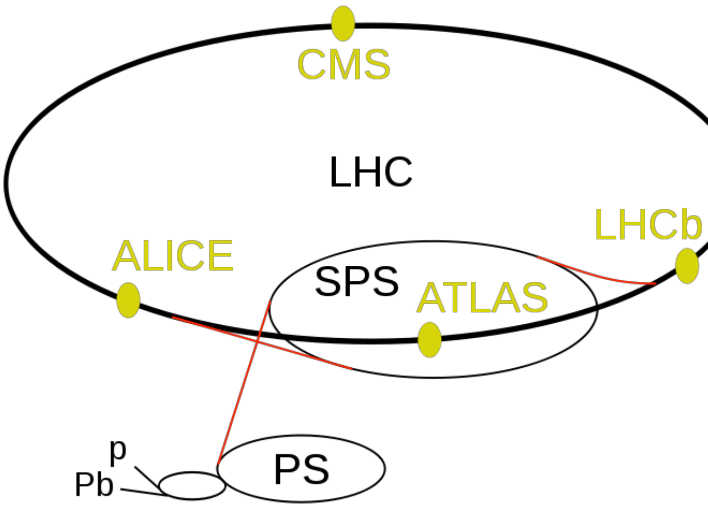


Figure 1.3: A schematic diagram of the LHC

The main ring cannot create the energies of 14 TeV from stationary protons. First protons are generated from LINAC 2 with and energy of 50 GeV. These are then fed into the Proton Synchrotron Booster (PSB) accelerating them to 1.4 GeV, and then the Proton Synchrotron (PS) boosting them to 26 GeV. Next is the Super Proton Synchrotron accelerating them to 450 GeV before entering the main ring which finally accelerates them to 14 TeV. All of the injector components were older experiments at CERN.

The LHC also has higher luminosities than any other previous collider, meaning that it is more likely that “interesting” events will be created. Under nominal operating conditions, each proton beam has 2808 bunches with each bunch containing about 10^{11} protons. As the bunches travel around the ring, they are squeezed as much as possible at the interaction points to maximise the probability of a head on collision.

The LHC has six experiments, each with their own detectors; ATLAS, CMS, ALICE, LHCb, LHCf, and TOTEM, each with their own goals in mind. At these detectors need a strong magnetic field to manipulate particles, they also use superconducting technology and require complex cryogenic systems. The ATLAS detector has one of the largest superconducting magnets ever made. The huge increase in energy compared to previous particle accelerators will give the LHC an insight into whether or not the Higgs boson exists, evidence for Supersymmetry and extra dimensions.

1.2.2 Tevatron

Tevatron is the predecessor of the LHC. It was where some of the technologies such as superconducting magnets and semiconductor tracking devices found in the LHC were first developed. It is a proton – antiproton collider located at Fermilab in Illinois. Its main ring is 6.3 km in circumference, and like the LHC uses superconducting magnets which are cryogenically cooled by liquid helium, enabling Tevatron to accelerate protons to 980 GeV, with a centre of mass energy of 1.96 TeV [8]. One of the initial goals of the Tevatron was to look for the Higgs boson. Due to recent delays at the LHC, operations at the Tevatron have been ramped up to beat the LHC to answer some of the questions surrounding the Higgs boson [9].

There are two detectors on the Tevatron ring, CDF and DØ. Both collaborations have observed the top quark and found its mass to within 1%, and also found many new particles with different hadronic combinations. Both the collider and the detectors have all undergone substantial upgrades, such as an increase in luminosity, more powerful injectors and energy since first being built [10] [11].

Again, like the LHC, there are a number of stages before the protons (and antiprotons) reach their maximum energy of 980 GeV. The first stage is the 750 keV Cockcroft-Walton pre-accelerator, which is effectively a giant capacitor which ionizes hydrogen gas and accelerates the positive ions. Next the hydrogen ions pass though a linear accelerator (Linac) where they are accelerated to 400 MeV using oscillating electric fields, which groups them into bunches. The ions are then passed through carbon foil to filter out the electrons and then passed onwards to the Booster. The Booster then accelerates protons to 8 GeV in a rapid cycling synchrotron powered by conventional magnets.

The antiprotons are created by accelerating protons from the Booster to 120 GeV in the Main Injector (the latest upgrade to Tevatron) and collided with a nickel target creating a wide spectrum of particles. About 20 antiprotons are made for every 1 million protons, which is the main restriction on the luminosity of Tevatron. It takes between 10 and 20 hours to produce a sufficient stack of antiprotons which can be used in Tevatron [6]. Once enough antiprotons are collected the protons and antiprotons are then accelerated in the Main Injector to 150 GeV, ready for the final acceleration to 980 GeV in the main ring.

The bulk of the data used in this project will be using data from Tevatron and CDF with Run I (1800 GeV) and Run II (1960 GeV).

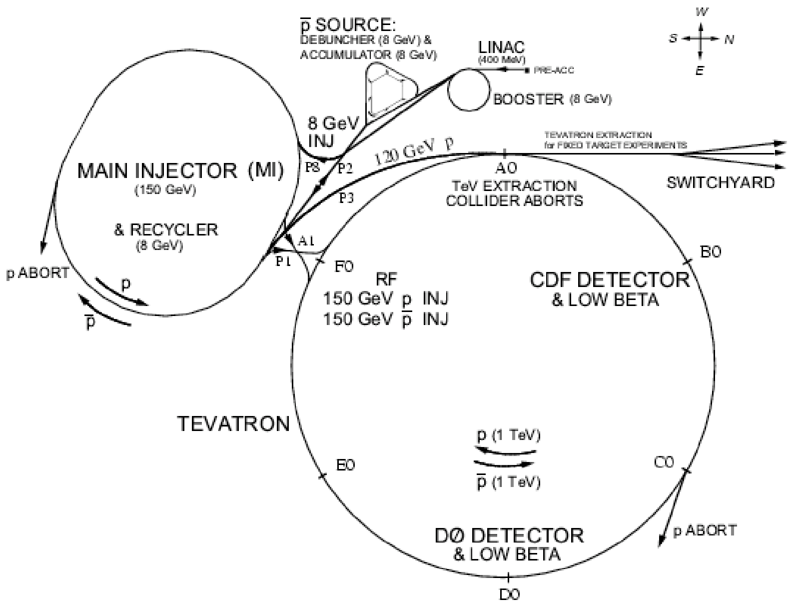


Figure 1.4: A schematic diagram of Tevatron (Source: Fermilab)

1.3 Collider Phenomenology

High energy collisions between protons and antiprotons can cause the quarks within hadrons, or newly created quark-antiquark pairs to fly apart from each other at very high energies. Before they can be observed, these quarks are converted into “jets” of hadrons (a process known as fragmentation) whose production rates and angular distributions reflect those of the quarks from which they originated [2].

The total cross section of a proton is given by:

$$\sigma_{Total} = \sigma_{Elastic} + \sigma_{Inelastic}$$

In most collisions, the particles just pass through each other, without breaking apart, with a small amount of momentum (low p_T) being transferred between them. Here, the proton and antiproton scatter elastically through a small angle, and there are no new particles or energy loss, which isn’t of much interest in this context.

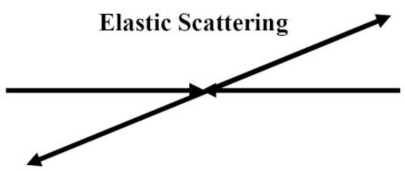


Figure 1.5: Elastic scattering with little momentum transferred between the particles

The inelastic cross section can be split into three terms; single diffraction (SD), double-diffraction (DD), and non-diffractive (ND, i.e. everything else).

$$\sigma_{Inelastic} = \sigma_{ND} + \sigma_{SD} + \sigma_{DD}$$

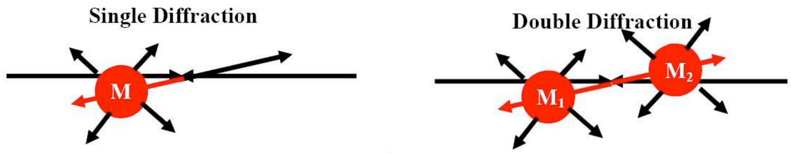


Figure 1.6: Single and Double Diffraction

In double or single diffraction, one or both of the beam particles are broken apart, as shown in Figure 1.6.

1.3.1 Minimum Bias

It is not unusual to find different definitions of minimum bias events in the literature. However, most groups define minimum bias events as non-diffractive inelastic collisions [12]. It is this non-diffractive part that is interesting in the context of particle physics. Most of the time the colour exchange between partons in the beam hadrons occurs through a soft interaction (i.e. no high transverse momentum) and the two beam hadrons “ooze” through each other producing lots of soft particles with a uniform distribution in rapidity and many particles flying down the beam pipe. Occasionally there is a hard scattering among the constituent partons producing outgoing particles and “jets” with high transverse momentum [13].

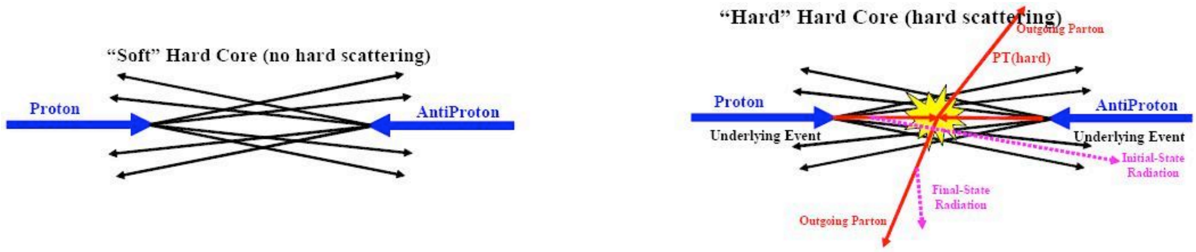


Figure 1.7: Proton-antiproton collision with 2-to-2 parton scattering.

The left image in Figure 1.7 shows soft scattering, and the right image shows a hard collision resulting in the production of high p_T jets.

A soft collision is when there is a low transverse momentum transfer from initial to final state, and few or no particles are produced with a significant p_T. On the other hand, a hard scatter is an interaction involving the creation of at least one particle with appreciable p_T. Hard interactions can be calculated reliably using Perturbative QCD while soft interactions are not easily calculable within QCD and rely on ad-hoc models which are taken from data, with some theory [14].

1.3.2 The Underlying Event

The focus of this project is the underlying event, which contains all activity in a hadronic collision that is not related to the signal particles from the hard process, e.g. leptons or missing transverse energy. Analyses developed by the CDF Collaboration indicate that the underlying event contains soft and hard components. The soft component is mainly associated with beam–beam remnant interactions. Particles composing the hard component come from the initial and final state radiation (caused by bremsstrahlung and gluon emission), from colour strings stretching between the underlying event and the highest-p_T jet and from secondary parton interactions [12].

A hard scattering event like in Figure 1.7, consists of large transverse momentum outgoing hadrons that come from the large transverse momentum partons (the outgoing hard scattering jets) and also hadrons that originate from the break-up of the proton and the antiproton (the beam-beam remnants). In addition to beam-beam remnants, the underlying event may contain hadrons resulting from initial state radiation. It is also possible that multiple parton scattering as shown in Figure 1.8 can contribute to the underlying event [15].

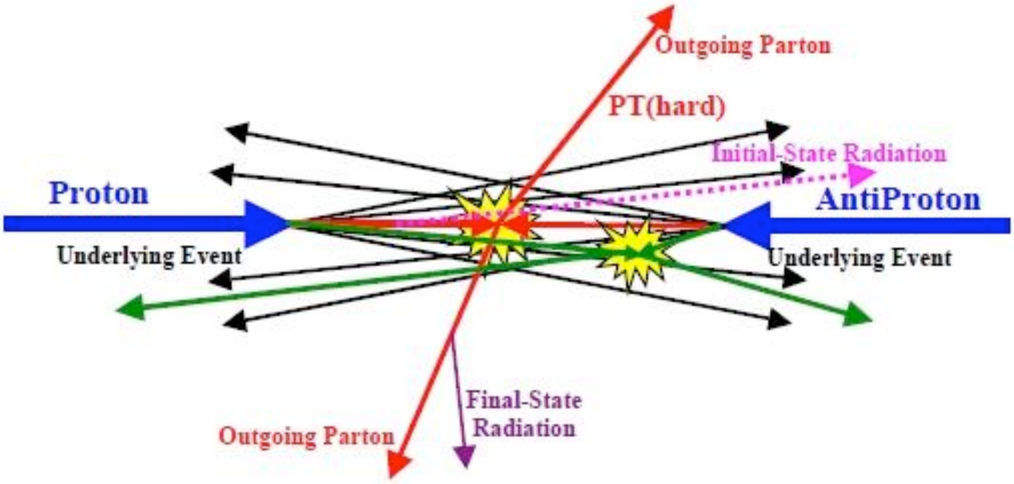


Figure 1.8: A proton-antiproton collision where a multiple parton interaction has occurred.

Figure 1.8 shows a proton-antiproton collision. As well as the hard 2–to-2 scattering, there is an additional soft or semi-hard parton-parton scattering that contributes particles to the underlying event.

The underlying event is an unavoidable background to most collider observables and it is not possible on an event-by-event basis to be certain what particles came from the underlying event and, which particles originated from the hard scattering. Understanding the underlying event well leads to more precise collider measurements. This becomes more important at the energies seen at the LHC, as the underlying event becomes more apparent, and large amounts of missing energy due to the underlying event could severely compromise the precision of LHC measurements [16].

1.3.3 Dividing into regions

To investigate the underlying event, after the charged particles were selected using a set of cuts, jets were defined as circular regions found in cones in η-ϕ space, where the radius of the cone is given by:

$$R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$$

Where η is the pseudorapidity and φ is the azimuthal scattering angle. The cone radius used in all of the analyses examined in this project is R = 0.7.

The transverse momentum of a charged particle jet is defined as the scalar sum of the transverse momenta of the charged particles making up the jet. The jet with highest transverse momentum is taken to be the “leading charged particle jet”, referred to as the leading jet as shown in Figure 1.9 below [12].

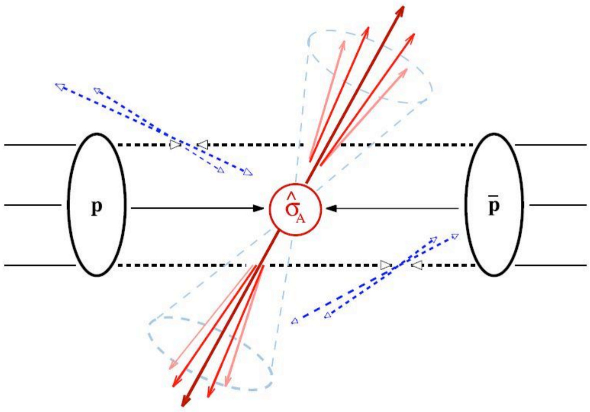


Figure 1.9: Illustration of a jet produced by a hard proton–antiproton scattering

The direction of the leading jet is used to isolate three regions of η-ϕ space that are sensitive to the underlying event. The angle $\Delta\phi = \phi_{particles} - \phi_{leading\ jet}$ is the relative azimuthal angle between charged particles coming from the underlying event and the direction of the hard scattered leading jet.

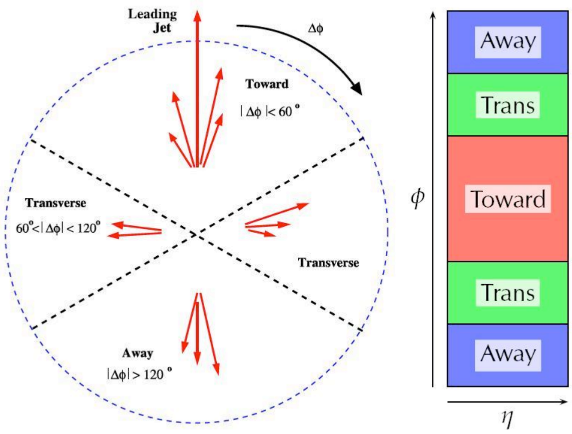


Figure 1.10: Event regions defined in terms of the azimuthal angle

The regions are divided as:

- $|\Delta\phi| < 60°$ as the toward region
- $60° < |\Delta\phi| < 120°$ as the transverse region
- $|\Delta\phi| > 120°$ as the away region

1.4 Event Processes

There and two main event processes that were studied in this project; QCD 2-to-2, and Drell-Yan lepton pair production.

1.4.1 QCD 2-to-2

A QCD 2-to-2 process is when two partons collide to produce a quark – antiquark pair which produce into two hard jets of colourless hadrons, photons and leptons to form reasonably long lived observable hadronic particles such as π^+, π^-, π^0, K^+, K^-, η, p, n etc.

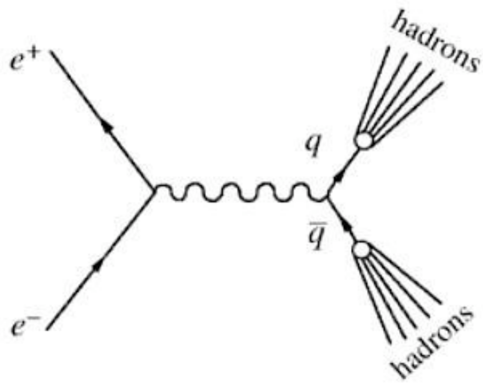


Figure 1.11: Basic Mechanism of two-jet production

Figure 1.11 shows two a two jet event is created in electron – positron annihilation, to create two cones of hadrons. The underlying event in QCD 2-to-2 events has been well studied. The two studies that this project analyses are:

"Charged jet evolution and the underlying event in proton - antiproton collisions at 1.8-TeV" [15] (*referred to as the CDF 2001 analysis in this report*)

"The Underlying event in hard interactions at the Tevatron antiproton - proton collider", [17] (*referred to as the CDF 2004 analysis*)

1.4.2 Drell-Yan Process

The Drell-Yan process is where quarks and antiquarks from the incoming hadron beams annihilate to produce a virtual photon or Z^0, which decays to a lepton pair (such as electrons or muons), as shown in Figure 1.12 below.

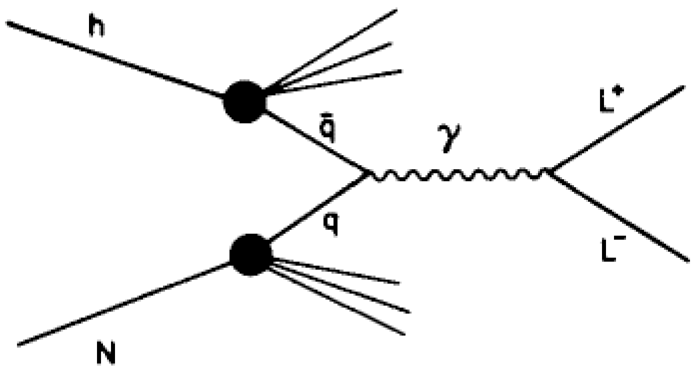


Figure 1.12: Schematic representation of the Drell-Yan lepton pair production

The underlying event in the Drell-Yan process has not been studied as much as QCD 2-to-2 and is the main focus of this project. A previous study of this process is a recent PhD thesis by Deepak Kar:

"Using Drell-Yan to probe the underlying event in Run II at Collider Detector at Fermilab (CDF)" [6] (*referred to as the CDF 2008 analysis*).

1.5 Monte Carlo Event Generators

The Monte Carlo method (MC) is a numerical technique for calculating probabilities and related quantities by using sequences of random numbers [18]. As opposed to deterministic simulation methods, they use random numbers and hence are stochastic. It is useful in many interesting calculations, such as determining the cross section for a scattering process that has too many degrees of freedom for direct numerical integration.

In practice, Monte Carlo models are run on computers. Since computers cannot generate randomness, Monte Carlo generators use sequences of pseudo-random numbers. The disadvantage of this is that computers use an algorithm to generate these numbers, and depending on the power of algorithm, these "random" numbers may become periodic when large volumes of numbers are generated. This leaves the accuracy of Monte Carlo models open for debate, as their accuracy depends on their degree randomness. However, modern number generators have very large periods, and can be considered accurate for this implementation. For example, the TRandom3 number generator found in ROOT has a period of approximately 10^{6000} [19].

The advantage of using pseudorandom numbers is that the can be recreated if the starting conditions are the same. Therefore, in a Monte Carlo generator, if there is an interesting event, or a bug, that exact event can be recreated perfectly unlike in a real life experiment.

As well as event generation, the ways the particles interact with the detectors also have to be simulated. The behaviour of the detectors, how particles produced by the event generator traverse the detector, spiral in magnetic fields, shower in calorimeters, or sneak out through cracks, etc., all affect the distributions produced.

The two generators discussed in detail in this project are Jimmy and Herwig++. The software used to manipulate these generators and analyse the data is Rivet, and ROOT was used to plot the histograms produced from the Monte Carlo data.

1.5.1 Jimmy

Jimmy is a multiple parton interaction model which can be added to Fortran Herwig to improve agreement with the underlying event observables. It is designed purely with the underlying event in mind, so it cannot be used for minimum bias events. To correctly simulate minimum bias collisions one must have the correct mixture of hard and soft processes together with a good model of the multiple-parton interactions, which in practice is difficult to do. Jimmy avoids this problem by only modelling the hard part of the collision, which is why it is unsuitable for minimum bias events.

Jimmy is based on an eikonal model which is derived from the observation that that for partonic scatters above some minimum transverse momentum, p_T^{min}, the values of the hadronic momentum fraction x which are probed decrease as the centre-of-mass energy, s, increases, and since the proton structure function rises rapidly at small x, high parton densities are probed. Thus the perturbatively-calculated cross section grows rapidly with s. However, at such high densities, the probability of more than one partonic scattering in a single hadron-hadron event may become significant. Allowing such multiple scatters reduces the total cross section, and increases the activity in the final state of the collisions [20] [21]. This allows the underlying event to be modelled more accurately.

1.5.2 Herwig++

Herwig++ is the replacement for the original Fortran Herwig which ceased development in 2005 [22], which is translated from FORTRAN to C++ with a modern object orientated design, built on a platform called ThePEG. The version used in this project is 2.3.0 with ThePEG version 1.4.0 [23].

This most recent version has a different model for the underlying event compared to the previous versions. Previously, the default model for the underlying event was the UA5 model. Here, additional (soft) hadronic activity is generated as a number of additional clusters are generated flat in rapidity with an exponential transverse momentum distribution. These clusters eventually give the required additional activity of soft hadrons. However, since Herwig++ 2.1, a model more similar to Jimmy was adopted implementing physics much closer to the eikonal approach [24]. The changes between 2.1 and 2.2 were minor and didn't change the physics much [25]. However, version 2.3 extends the Jimmy-like model to use a Pythia-like regularised cross-section and the inclusion of soft scatters below p_T^{min} to allow for min-bias simulation [26] (Pythia is another event generator which attempts to model both minimum bias and underlying events). The differences between the models are discussed in detail in section 2.3.

1.5.3 Rivet

Rivet is a toolkit for validation of Monte Carlo event generators [27]. It is an environment where MC generators can be run and steered, and their outputs analysed efficiently in a set of prewritten analyses. These analyses recreate a particular study, and generate the plots found in the paper of the study, so that they can be compared easily with the results of the paper. The corresponding Rivet analyses to the studies mentioned earlier in sections 1.4.1 and 1.4.2 are:

- CDF 2001: CDF_2001_S4751469
- CDF 2004: CDF_2004_S5839831
- CDF 2008: CDF_2008_NOTE_9351

The numbers at the end of the analysis name is the SPIRES id for the corresponding paper in the SPIRES HEP database (The CDF 2008 study is an exception and has a Fermilab database id in the mentioned in the bibliography). Rivet also provides the experimental data for each analysis for comparison.

1.5.4 ROOT

ROOT is an object oriented framework for large scale data analysis [28]. It is a highly powerful data analysis package, which can be used for a variety of applications. The main use in this project is to generate plots to compare different MC generator runs with the experimental data from CDF.

2 Analysis

2.1 Introduction: Goal

The goal of this project is to improve how Herwig++ models the underlying event. The initial step is to confirm current observations regarding the underlying event and to investigate the difference in physics between Jimmy and Herwig++. Then the models will be tuned to match data from CDF so that the underlying event in the Drell-Yan process is modelled more accurately. The two parameters used to tune the generators are the inverse proton radius (μ^2), and the minimum p_T cut-off (p_T^{min}). The defaults for both generators are shown in Table 2.1 below [21] [26].

Generator	**Inverse Proton Radius, μ^2 (GeV^2)**	**Minimum p_T cut-off, p_T^{min} (GeV)**
Jimmy	1.4	3.0
Herwig++ (2.3.0)	1.5	4.0

Table 2.1: Default parameters for Jimmy and Herwig++

All the plots and tunes in this section are done with 1 million events unless stated.

2.2 Confirming current observations

To investigate whether or not the generators and Rivet were working properly, each analysis was tested to see if it matched observations from the papers. This part of the project was the most time consuming part since lots of small technical details needed to be fixed, and since Rivet is a new project, there are always going to be bugs at this stage.

2.2.1 CDF 2001 Analysis

The CDF 2001 analysis (page 16) is one of the most heavily studied papers regarding MC generators and the underlying event and was conducted at the CDF detector at Tevatron (Run 1). It is often used as the benchmark test when testing the underlying event model in most generators. The most sensitive region to the underlying event is the transverse region. All of these plots were done using Jimmy and Herwig++ with their default parameters. In this analysis, the data is split into min-bias events (those where the energy of the leading jet is below 20 GeV) and JET20 (where the energy of the leading jet is above 20 GeV). In all of the plots there is a slight overlap where min-bias and JET20 data agree.

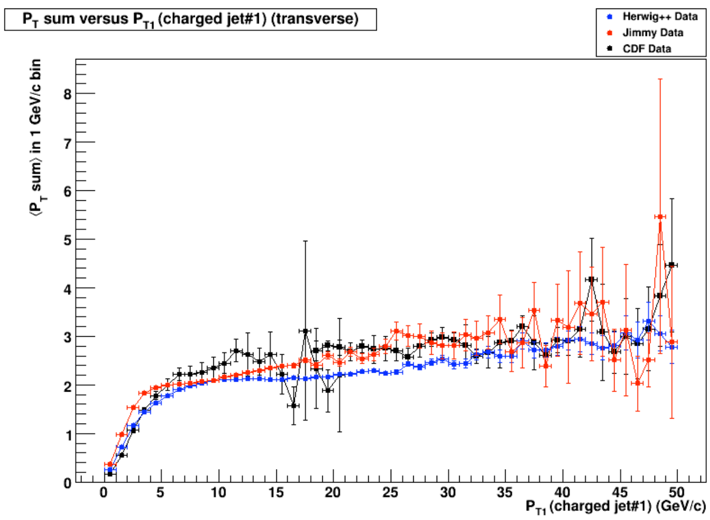


Figure 2.1: Transverse p_T sum with Jimmy and Herwig++

The paper reports that Fortran Herwig does not produce enough transverse p_T sum, which agree with Figure 2.1 above which was generated in Rivet, more so with Herwig++.

The paper also mentions that the beam-beam remnant components have the wrong p_T dependence, and Fortran Herwig predicts a p_T distribution that is too steep. This disagrees with plots generated with Jimmy and Herwig++ as the opposite is observed below. This could be due to the fact that the underlying event models work poorly at low p_T's and cannot accurately model minimum bias events and why agreement with the data improves with higher energies.

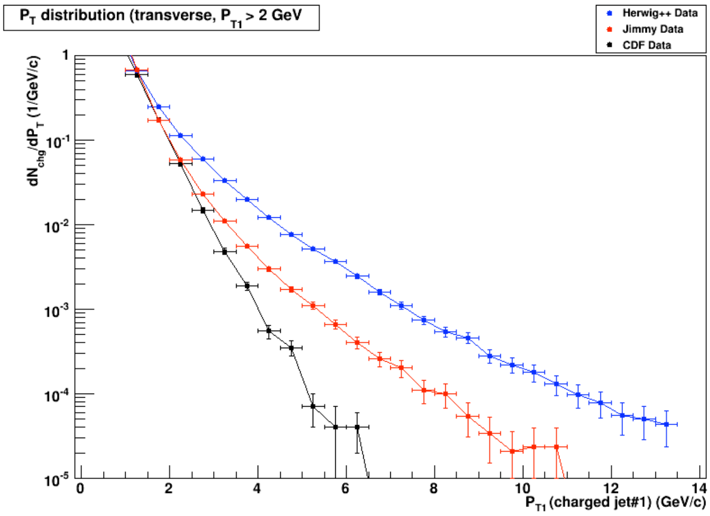


Figure 2.2: p_T distribution (transverse) for $p_T > 2$ GeV for Jimmy (red) and Herwig++ (blue)

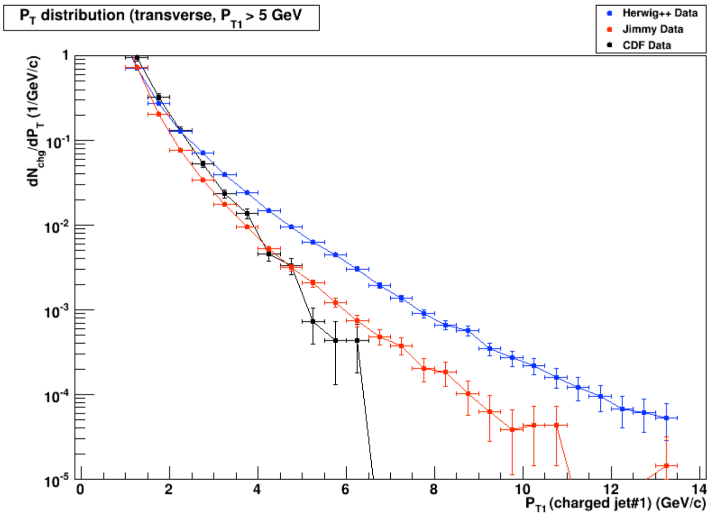


Figure 2.3: p_T distribution (transverse) for $p_T > 5$ GeV for Jimmy (red) and Herwig++ (blue)

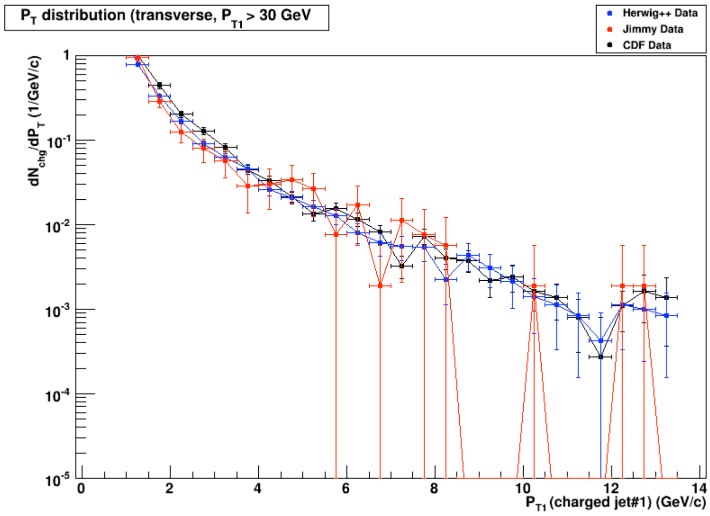


Figure 2.4: p_T distribution (transverse) for p_T > 30 GeV for Jimmy (left) and Herwig++ (right)

Another study of this analysis finds that Herwig++ has a slight trend to produce too much charged particle multiplicity in all the regions, most noticeably in the toward region, and too little p_T sum in all the regions, most noticeably the away region, which corresponds to a too soft spectrum of individual particles, and is also observed in Jimmy [1]. This is confirmed by the plot made in Rivet shown below and the p_T sum plot in Figure 2.1.

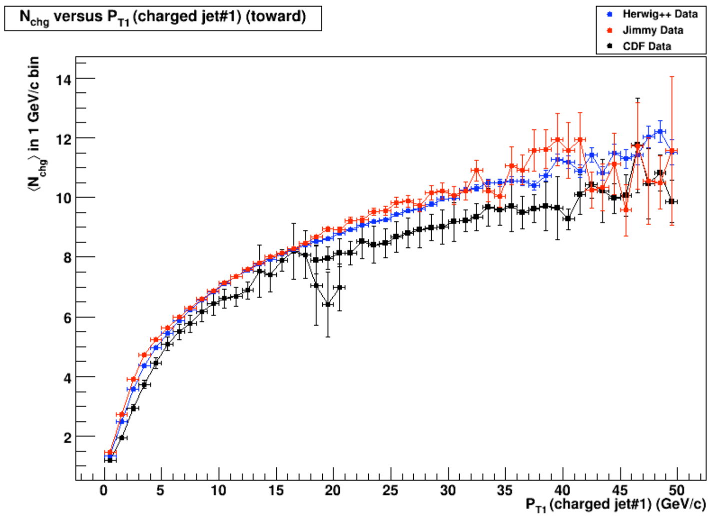


Figure 2.5: Charged multiplicity in the toward region for Jimmy and Herwig++

Both the lack of p_T sum and the excess charged multiplicity can be slightly improved by using tune optimised for Jimmy by setting μ^2 = 2.13 GeV^2 and p_T^{min} = 3.0 GeV (referred to as the Jimmy tune), and the improvement is also visible in Herwig++. On the whole, the plots generated using Rivet agreed with the findings in the paper which shows that the generators and Rivet is working as it should.

2.2.2 CDF 2004 Analysis

The CDF 2004 (page 16) analysis is an extension to the 2001 analysis. It investigates the energy dependence of events at √s = 630 GeV and √s = 1800 GeV. At these energies, jet production rate was measured for jets of 15-150 GeV and 15-450 GeV. Obviously in the latter case, these very high energy jets become increasingly rare, which means that the accuracy is reduced and the uncertainty is increased. In fact, in some of these plots the error bars tended to infinity and manually had to be set to 0. The next set of plots show the number of tracks in the max and min cone as a function of the leading jet (Figure 4 in [17]).

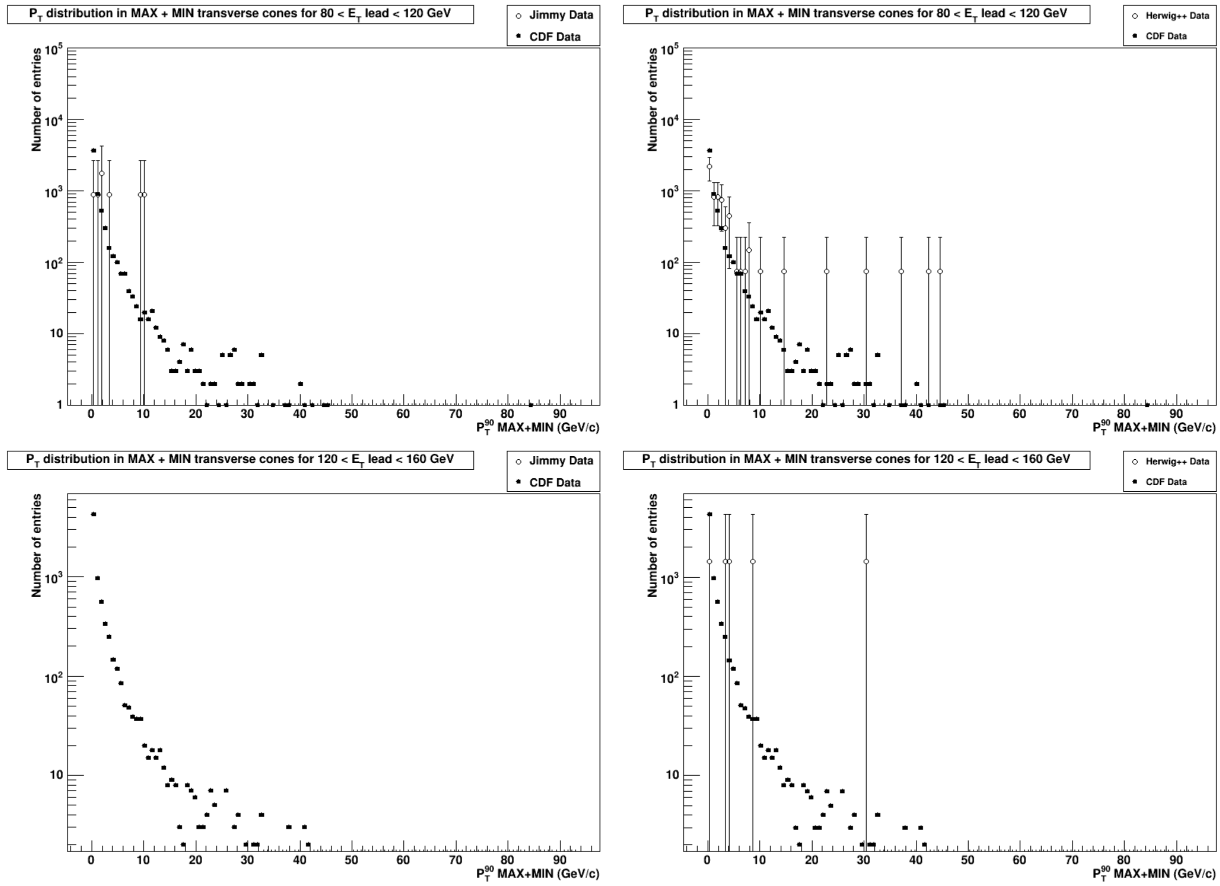
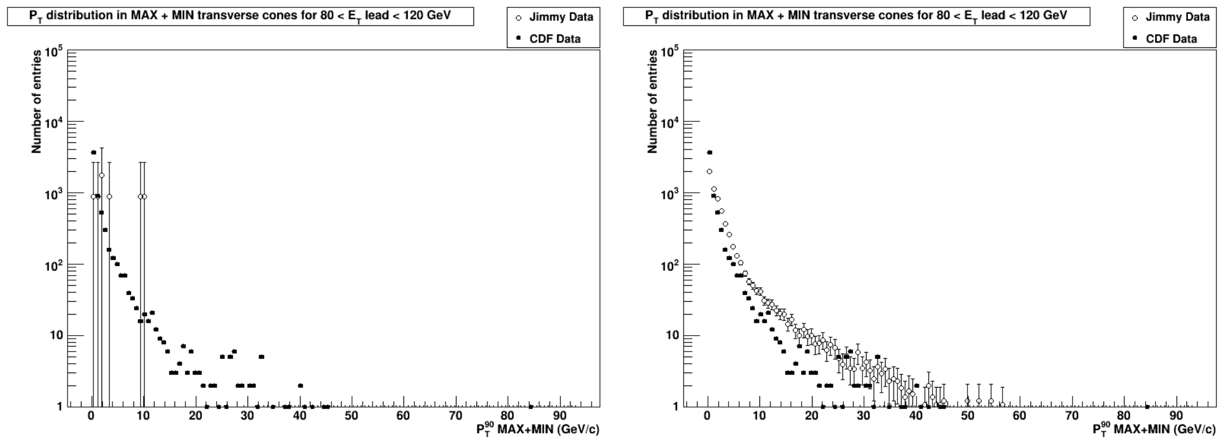


Figure 2.6: p_T distributions in *max* and *min* cone for Jimmy and Herwig++ using default tunes

Both generators fail here since not enough high p_T jets are made. One way of working around this is using more events. These plots were generated using 1 million events. For comparison, Jimmy was run with 10 million events.



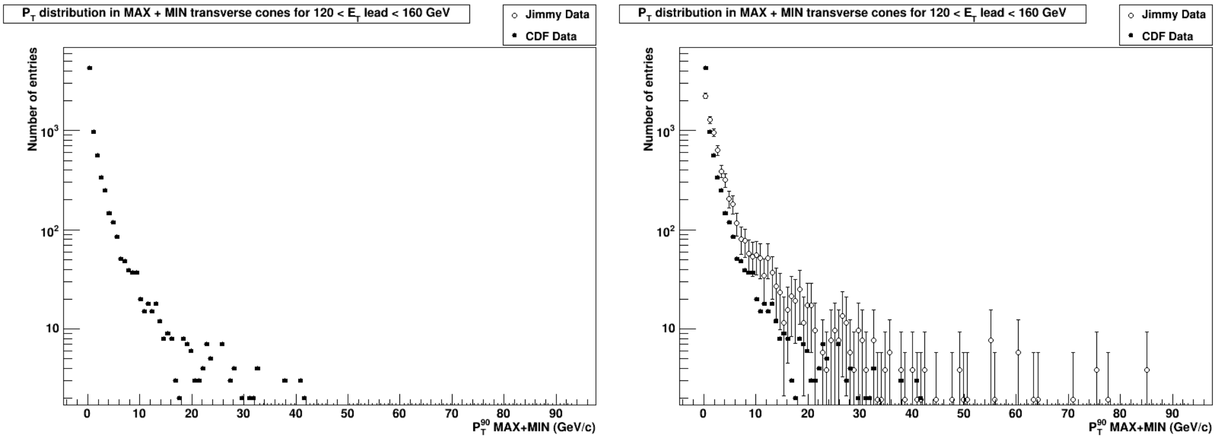


Figure 2.7: p_T distributions in *max* and *min* cone for Jimmy with 1M (left) and 10M (right) events

As shown in Figure 2.7, there is a clear improvement, and the results match that of the data and the paper. However, increasing the number of events is an “expensive” workaround as the simulations would take a long time to run. An easier solution is manually setting the minimum p_T of the leading jets to encourage higher p_T jets. This is done using the PTMIN (not to be confused with p_T^{min}) parameter in Jimmy and was set to 45 GeV. The disadvantage of this is that although it may fix the plots at higher p_T's, anything below 45 GeV will not be valid data. However for the purposes of validating Rivet and the generators, this is a reasonable measure to take.

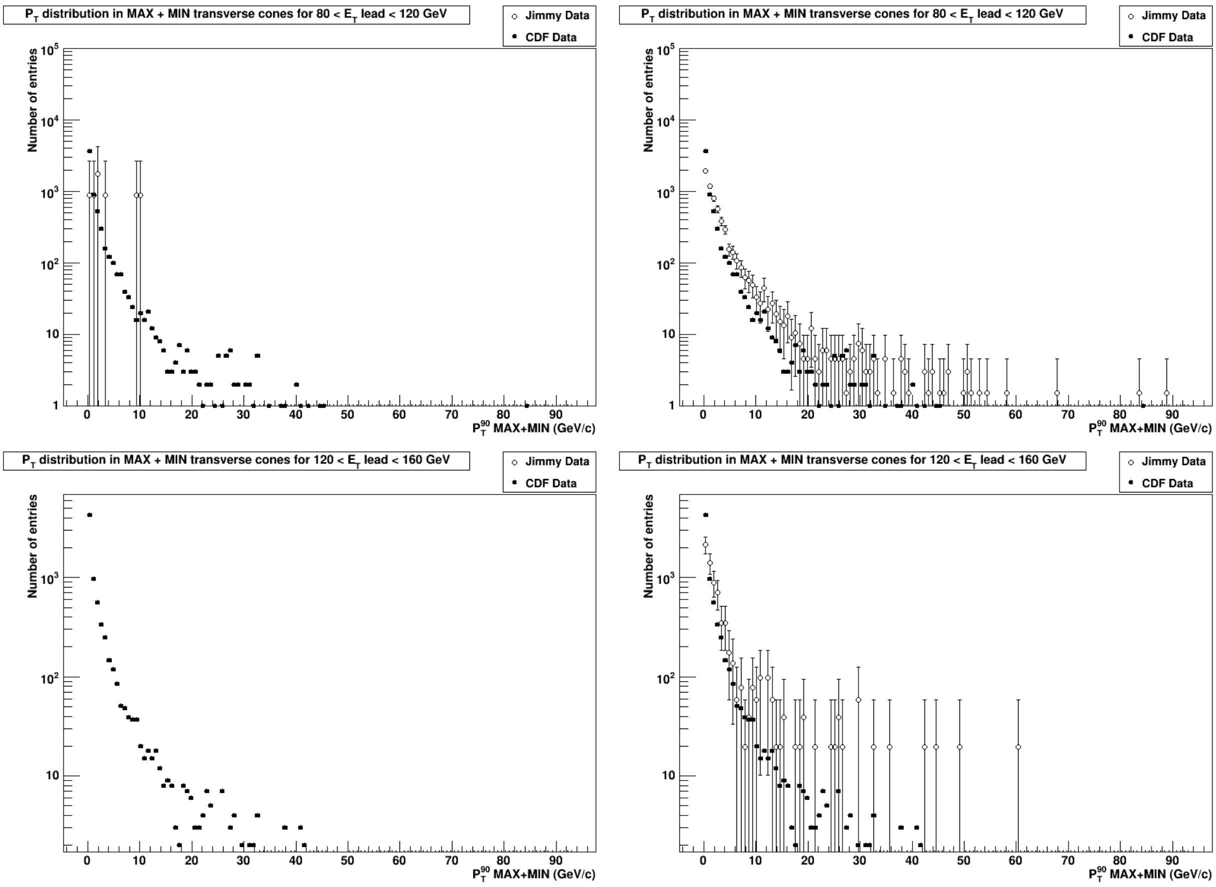


Figure 2.8: p_T distributions in *max* and *min* cones for Jimmy with default and PTMIN = 45 (right)

Although these plots do not agree with the paper well, we know that the problem lies with the production of high p_T jets and is not a problem with the analysis or Rivet.

Plots in the paper show that the comparisons of p_T 90, *max* and *min* all agree well with MC data at √s = 1800 GeV (Figure 2 in [17]) and at √s = 630 GeV (Figure 8 in [17]). The plots generated from Rivet agree with this at low energies. The disagreement at high energies is because not enough high energy events were generated.

The paper reports that a slightly higher track multiplicity is observed when compared to data (Figure 5 in [17]). This is confirmed in the plots generated from Rivet. (Note: at the time of writing, Herwig++ was not steerable through Rivet, so the PTMIN parameter was not set).

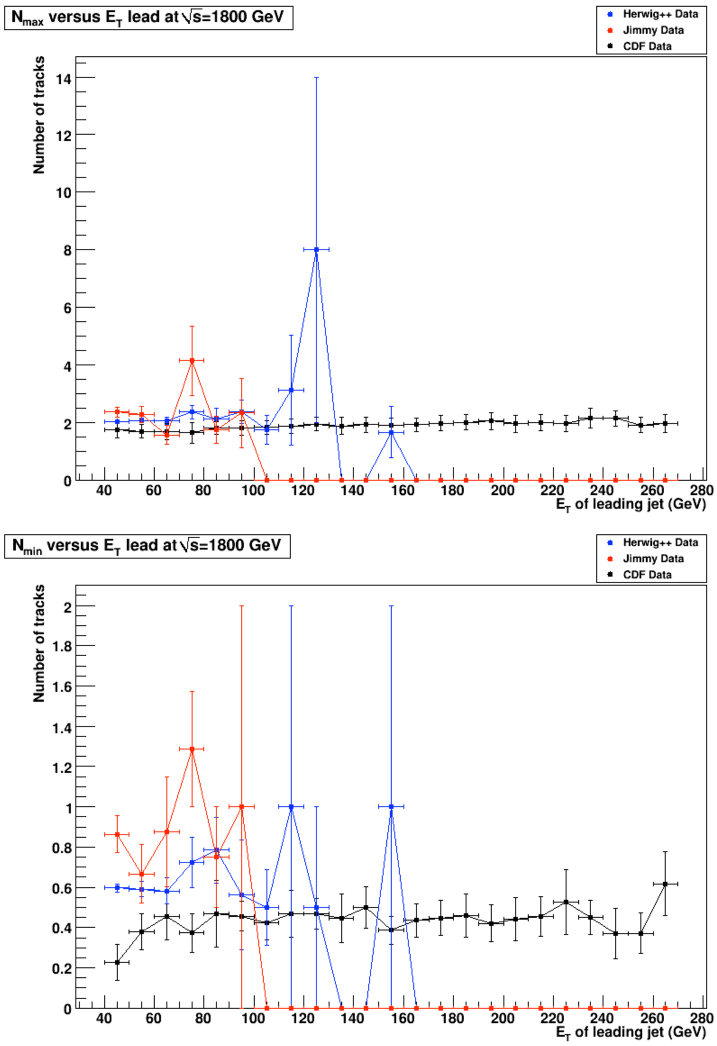
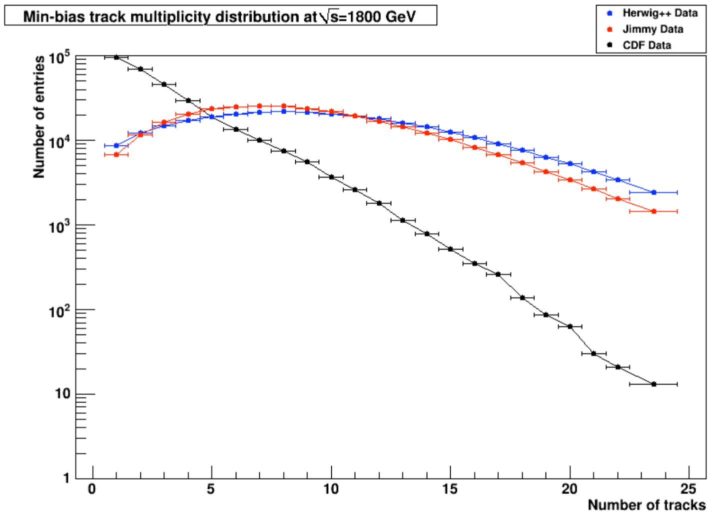


Figure 2.9: Number of tracks in *max* and *min* cones for Jimmy and Herwig++ at √s = 1800

The track multiplicity and track momentum distributions (Figure 6 in [17]) are not well produced by Jimmy and Herwig++ and the same is true at √s = 630 GeV (Figure 10 in [17]). The paper states that Fortran Herwig does not produce enough high p_T tracks. In the plots below the opposite is the case, where too many high p_T tracks are being made. This may again be due to the underlying event models not being able to model minimum bias events accurately, causing this strange behaviour. This could also be caused by a bug in the Rivet analysis.



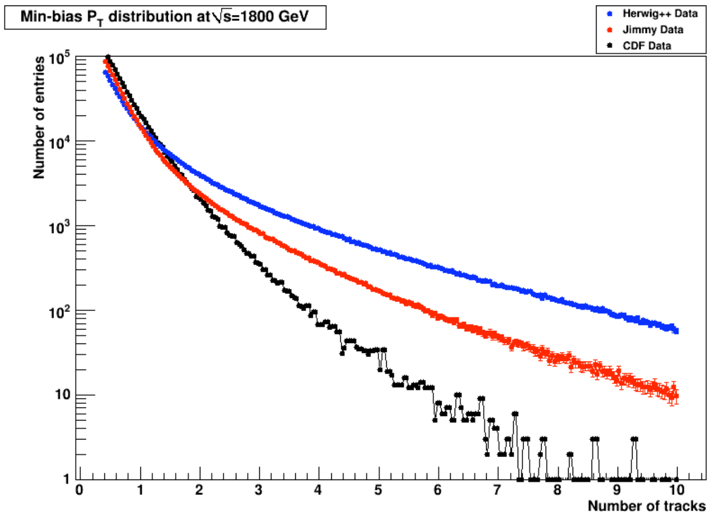


Figure 2.10: Distribution of track multiplicity (top) and transverse momentum (bottom)

The next set of plots show the Swiss cheese distributions (Figure 7 in [17] at √s = 1800 GeV and Figure 9 at √s = 630 GeV). For the removal to two jets, the paper reports that the MC data lies above the data. This was not observed in the plots generated by Rivet, as the MC data agrees with the experimental data and also for the √s = 630 GeV case. Note how the Jimmy data fits the CDF data poorly below 45 GeV due to the addition of the PTMIN parameter which was set to 45 GeV.

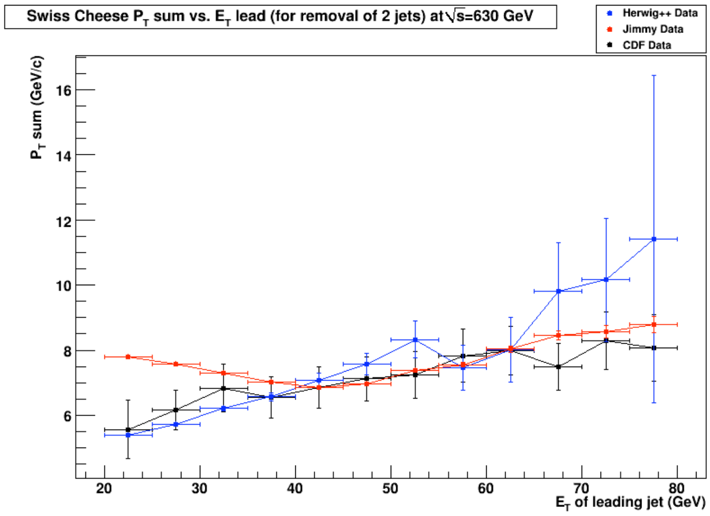


Figure 2.11: Swiss cheese removal of two jets at √s = 1800 GeV

The plots below show Swiss cheese removal of three jets. The plots in the paper show that the MC data agrees well with the experimental data. This is not the case with the plots generated from Rivet.

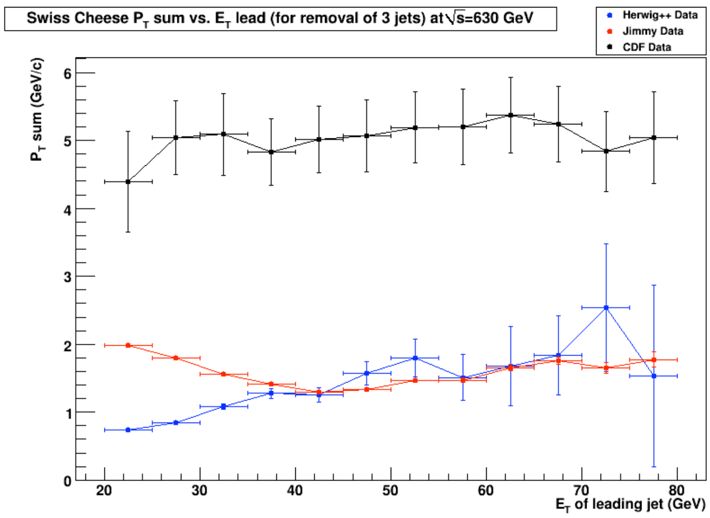


Figure 2.12: Swiss cheese removal of three jets at √s = 1800 GeV

As shown, there is a poor fit with the data. After a discussion with Andy Buckley (a developer of Rivet) it is clear that this is a problem with the analysis, as the Rivet team has been working for some time to reproduce this plot. Therefore it is safe to say that this is not a problem with the generators.

2.2.3 CDF 2008 Analysis

The CDF 2008 (page 17) analysis studies the underlying event in the Drell-Yan process. It is much less studied than the previous two analyses, and current Monte Carlo models often struggle with Drell-Yan data. As before the transverse region is sensitive to the underlying event, as the scattered jets are perpendicular to the plane of the 2-to-2 hard scattering. This makes it difficult to separate the outgoing jets from the background. By splitting the two transverse regions into min and max transverse regions, it is possible to identify the hard component (max) and the initial state radiation (min) from the beam-beam remnants. Therefore, it is the transMIN region that is most sensitive to the underlying event as well as the toward region as there are no QCD processes contaminating this region.

In the paper, the charged particle densities for all the regions produced by Fortran Herwig somewhat agree with the data, although being slightly low. This is confirmed in the plots produced by Jimmy and Herwig++ using the default parameters. However, Jimmy does slightly better than Herwig++ here, hence the need to find a better tune for Herwig++ with respect to the underlying event. This has already been seen before as shown in Figure 2.15.

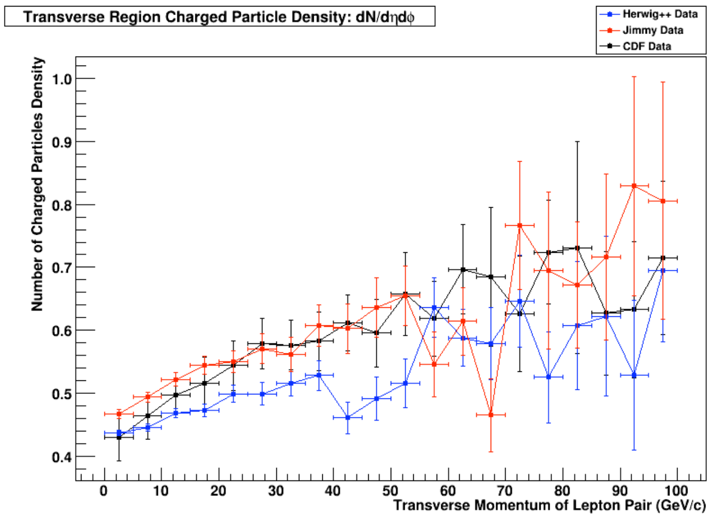


Figure 2.13: Charged Particle Density in Transverse Region (CDF 2008)

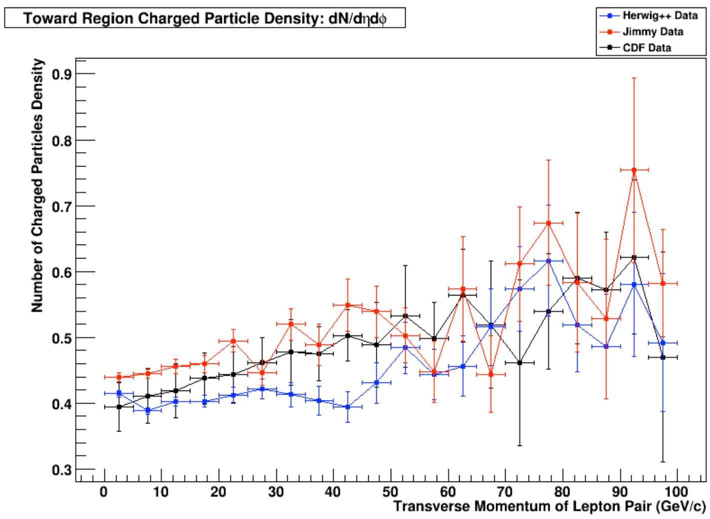


Figure 2.14: Charged Particle Density in Toward Region (CDF 2008)

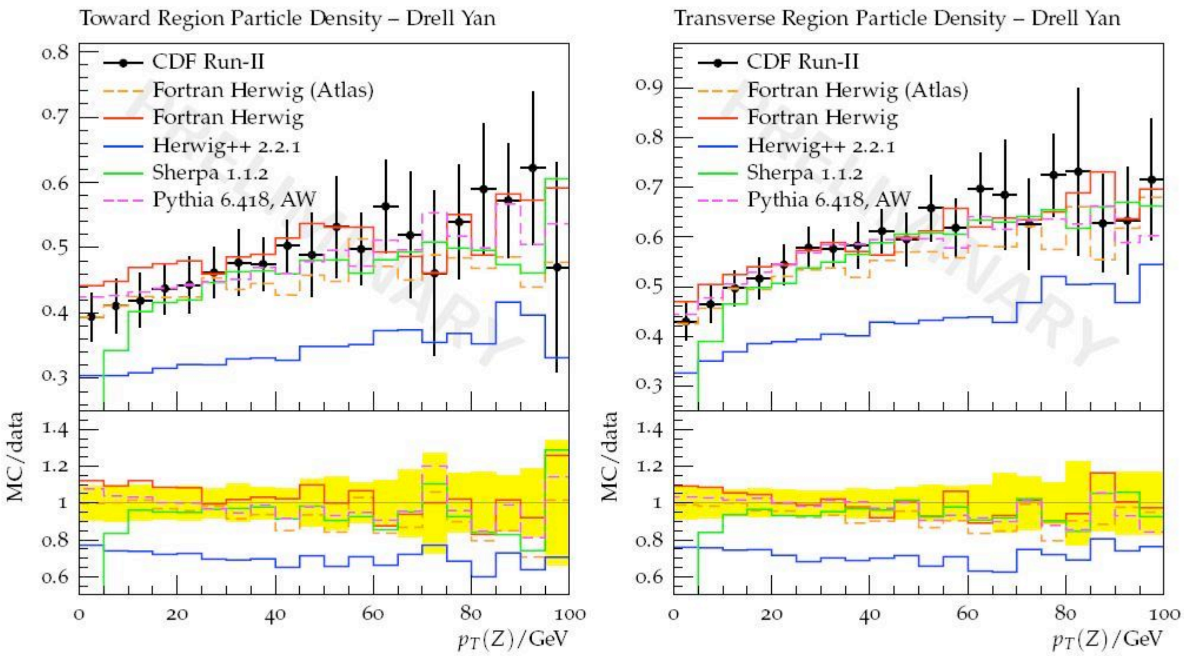


Figure 2.15: Lack of charged particle density in Herwig++ when compared to other generators [29]

The paper also notes that Fortran Herwig produces a p_T distribution of charged particles that is too soft. This is also confirmed in the plots generated from Rivet, as lines generated by Jimmy and Herwig++ lie below the experimental data.

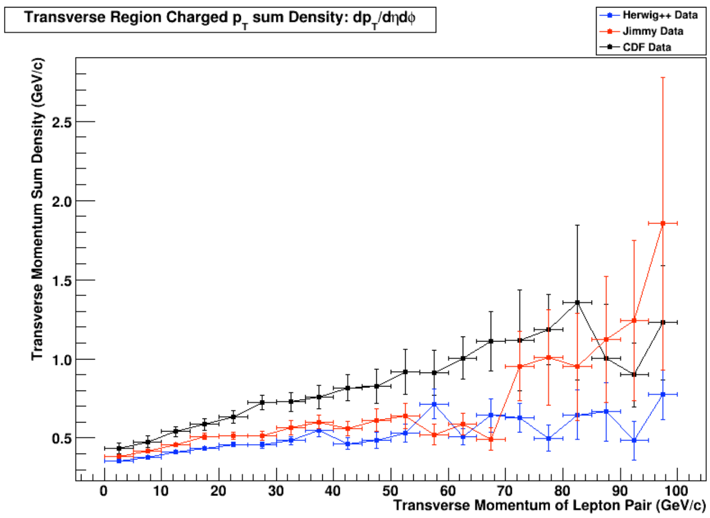


Figure 2.16: p_T sum density in Transverse region (CDF 2008)

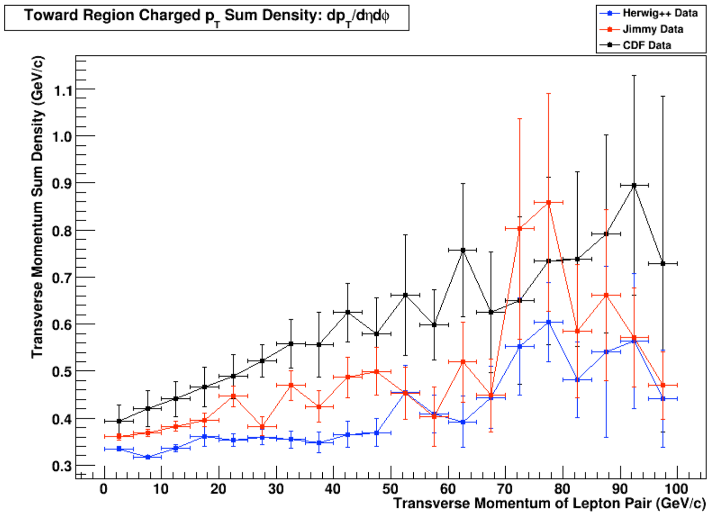


Figure 2.17: p_T sum density in Toward region (CDF 2008)

On the whole, the results generated using Jimmy and Herwig++ look consistent with the findings in the paper, and it is safe to say the Rivet and the analysis is working well and any discrepancies between the generated data and experimental data is due to the physics in the models.

2.3 Differences in physics between Herwig++ and Jimmy

So far, in the plots shown, results from Jimmy and Herwig++ agree well with each other. This shows that the physics in their eikonal models are very similar. This is clear in Figure 2.18 where Jimmy and Herwig++ have the same parameters.

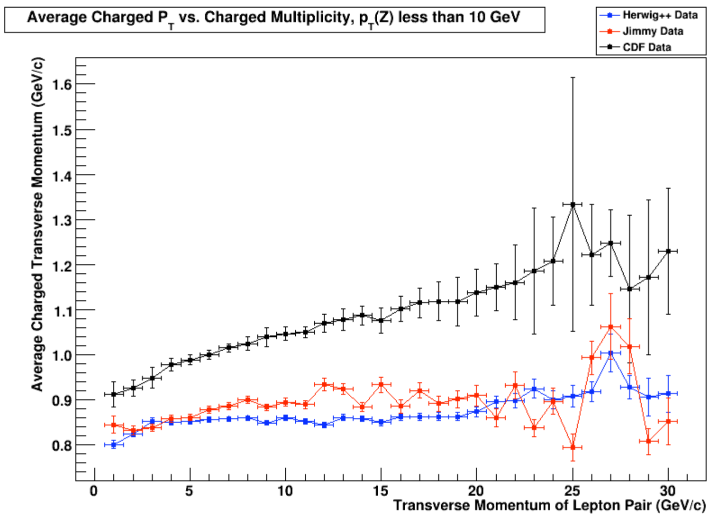


Figure 2.18: Addition of soft scatters below p_T^{min} in Herwig++

However, below p_T^{min}, the differences between the two models can be seen. In Figure 2.18 p_T^{min} was set to 3.4 GeV. Below this, Herwig++ starts to curve downwards as the underlying event model starts to generate soft scatters. Jimmy does not do this, as it cannot model soft scatters very well. Figure 2.19 shows how at low p_T's, the eikonal model makes the cross section tend to infinity, hence the need for a minimum p_T cut off, and is why Jimmy doesn't model soft scatters.

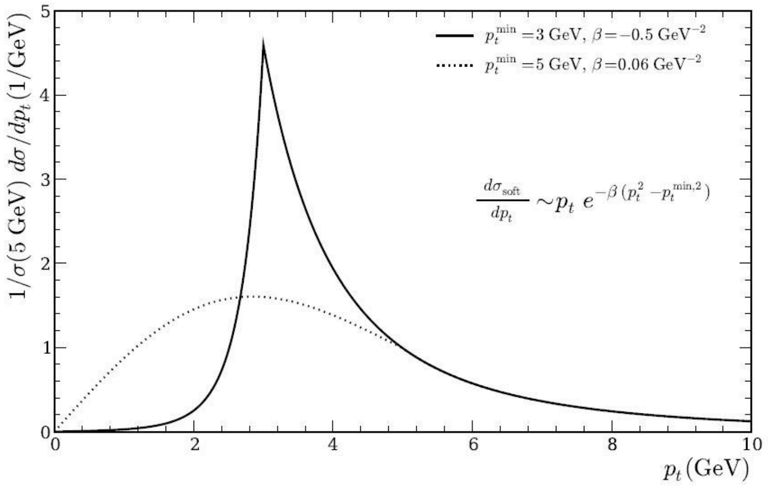


Figure 2.19: Illustration of the regularised cross section in Herwig++

Herwig++ attempts to add soft scatters below p_T^{min}, by regularising the cross section, and gently brings it back down to zero using a Gaussian functional form. This is what causes the slight difference in Figure 2.18, and allows Herwig++ to model minimum bias events [30].

2.4 Investigating Existing Tunes with Herwig++

After the generators and analyses were checked for discrepancies with previous work and any possible pitfalls identified, the next stage was to investigate the effects of trying different tunes.

The CDF 2008 analysis used Jimmy in its analysis, and used a tune where μ^2 =1.8 GeV^2 and p_T^{min} = 3.6 GeV. These parameters were then put into Herwig++ and set using the input file (shown in Appendix A) and plotted for comparison.

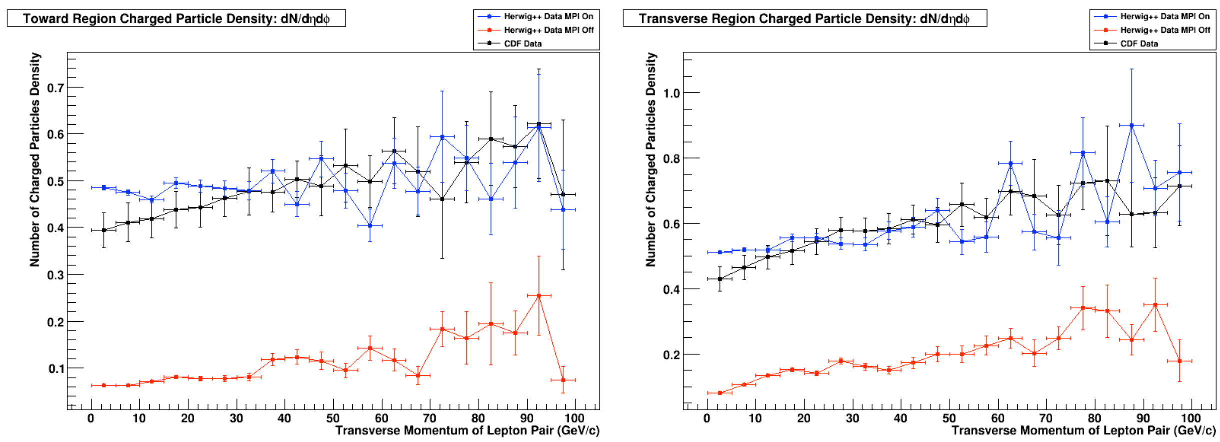


Figure 2.20: Charged particle densities using the tune given in the CDF 2008 analysis

As shown above, there is an obvious improvement to the default tune when compared to Figure 2.13 - Figure 2.15. For illustrative purposes, the red lines in Figure 2.20 show the results when the multiple parton interation (MPI) model is switched off. It is quite clear that the underlying event plays a large role here. The p_T sum densities are also improved, but still match the CDF data poorly.

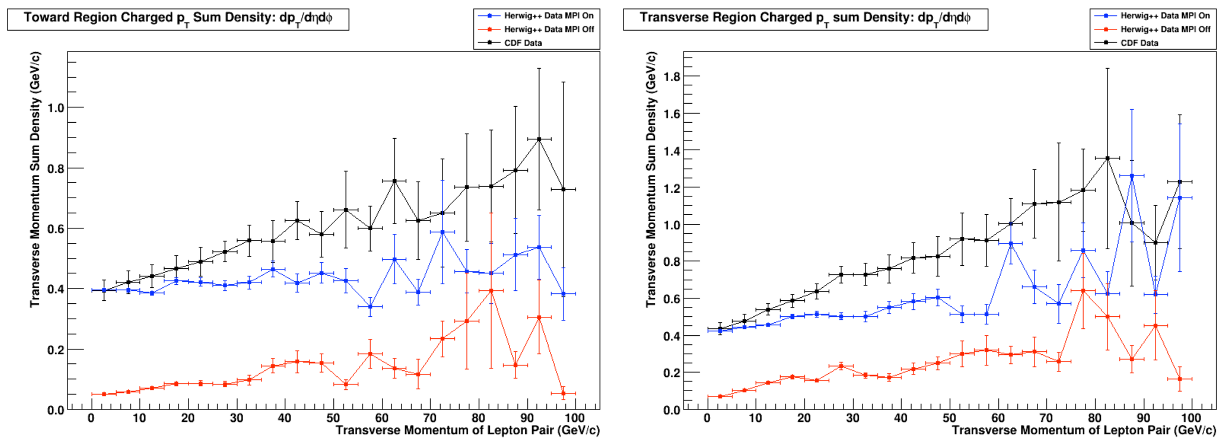


Figure 2.21: p_T sum densities using the tune given in the CDF 2008 analysis

The authors of Herwig++ published a paper investigating the effects of changing μ^2 and p_T^{min} in Herwig++ called "Underlying Events in Herwig++"by Bahr, Gieseke and Seymour [16]. In this paper, a brute force method was used to try a whole range of tunes which were compared to the data in the CDF 2001 analysis. A χ^2 test was then used and minimised to find the best tune. Figure 2.22 shows the results for the overall χ^2 and the results just for the transverse region.

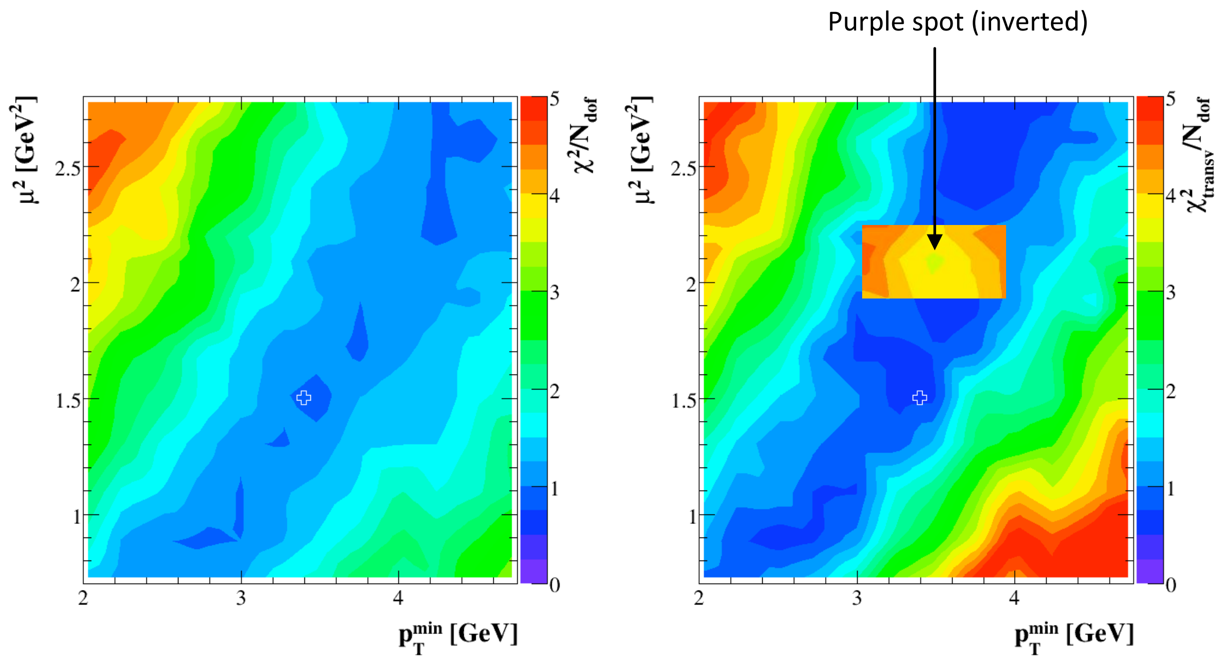


Figure 2.22: χ^2 results from the Herwig++ underlying event tuning with respect to CDF 2001 data

As shown in Figure 2.22, a valley is formed (shown in blue) in which Herwig++ would work well. The minimum of the valley is shown by the cross, and corresponds to μ^2 =1.5 GeV^2 and p_T^{min} = 3.4 GeV. The next stage of tuning Herwig++ was to investigate the effects of roaming in this valley. Several different tunes were investigated shown below.

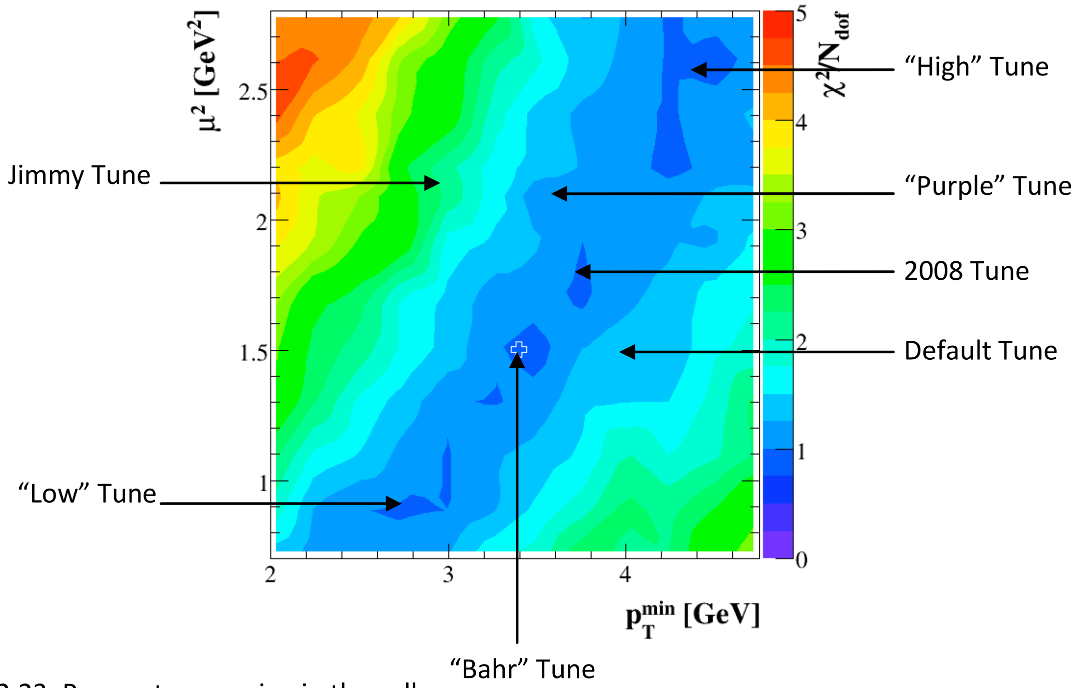


Figure 2.23: Parameter roaming in the valley

The different tunes are as follows:

- Default – The default tune in Herwig++
- “Bahr” Tune – The tune given in the “Underlying Events in Herwig++” paper (Manuel Bahr is the author)
- “Low” Tune – a point chosen to investigate the lower extreme of the valley
- Jimmy Tune – The best current tune used for Jimmy shown on page 22
- “High” Tune – a point chosen to investigate the upper extreme of the valley
- “Purple” Tune – A purple spot in the transverse valley shown in Figure 2.22 (colours inverted for illustrative purposes)
- 2008 Tune – the tune used with Jimmy in the CDF 2008 analysis

Tune	**Inverse Proton Radius, μ^2 (GeV^2)**	**Minimum p_T cut off, p_T^{min} (GeV)**
Default	1.5	4.0
Jimmy Tune	2.13	3.0
Bahr Tune	1.5	3.4
2008 Paper Tune	1.8	3.6
Low Tune	0.9	2.65
High Tune	2.6	4.4
Purple Tune	2.1	3.5

Table 2.2: Values for the different tunes used to explore the valley

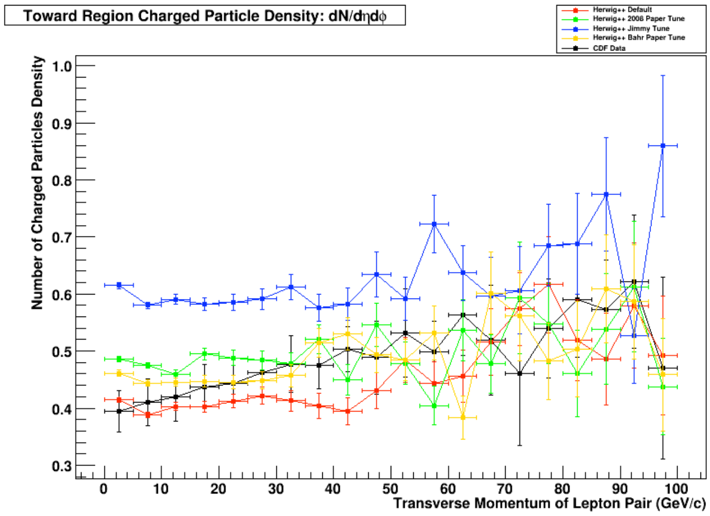


Figure 2.24: Toward region particle density for Jimmy, Bahr, 2008 and Default tunes

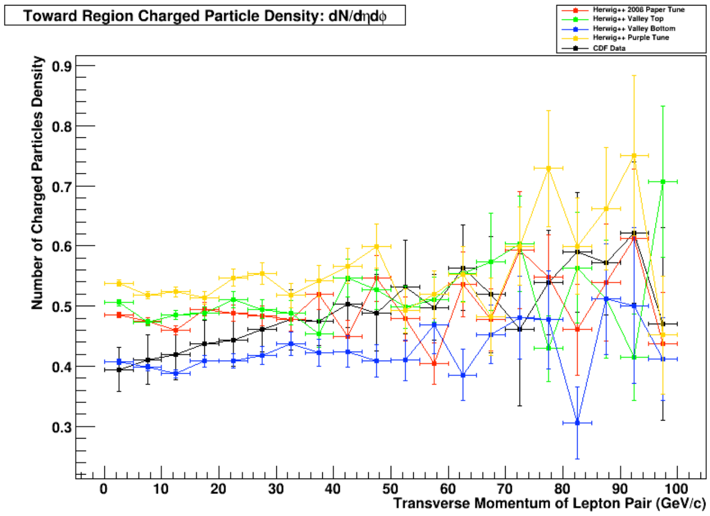


Figure 2.25: Toward region particle density for Low, High and Purple tunes

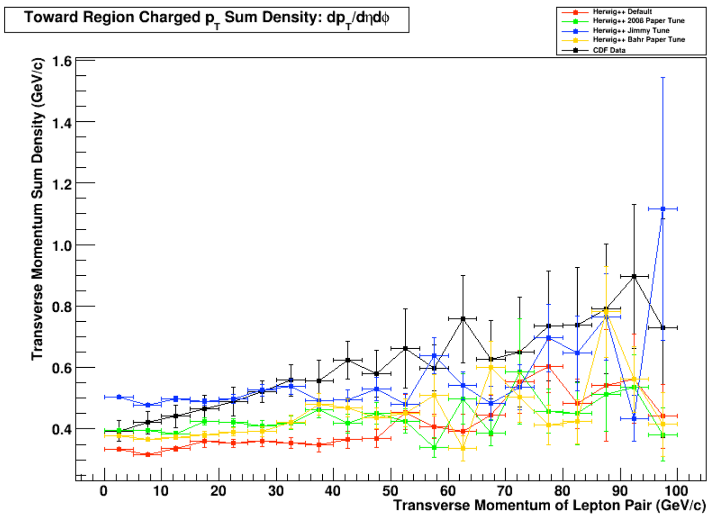


Figure 2.26: Toward p_T sum density for Jimmy, Bahr, 2008 and Default tunes

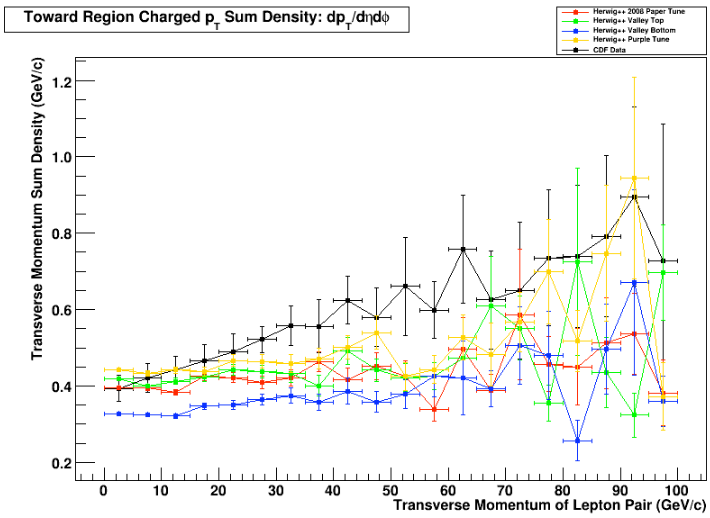


Figure 2.27: Toward p_T sum density for Low, High and Purple tunes

As shown in Figure 2.24 and Figure 2.25, most of the tunes model the charged particle densities well. However all of the tunes fail to model the p_T sum densities. One interesting case is the Jimmy tune. The Jimmy tune does the worst in terms of describing the charged particle density, but does the best out of all the tunes when describing p_T sum densities. This means that if a tune is chosen to improve the accuracy of p_T sum densities, the accuracy of the charged particle densities is compromised.

From this observation, it can be concluded that the missing p_T sum cannot easily be improved by tuning the model and is most likely to be a flaw in the physics of Herwig++. Figure 2.14 and Figure 2.17 show that this can also be observed with Jimmy.

2.5 χ^2 Testing

Since most of the tunes look very similar by eye, a χ^2 test was used to grade the quality of the tunes. A χ^2 test is a measure of how likely it is for a given set of data to result from a given hypothesis, and the measured value is tested against an expected value given by the hypothesis [18]. In this case, since both the CDF data and the Monte Carlo data have uncertainties associated with them, we need to decide on an expected value.

Let us consider a single bin with measured CDF value $y \pm \sigma_y$, and MC data $v \pm \sigma_v$. The hypothesis we want to test is that both are Gaussian distributed with a common mean value λ. This is a reasonable assumption if the errors are not large (i.e. lots of events are generated).

The χ^2 value for that single bin becomes:

$$\chi^2 = \frac{(y-\lambda)^2}{\sigma_y^2} + \frac{(v-\lambda)^2}{\sigma_v^2}$$

Minimising with respect to λ gives the least squares estimator for λ:

$$\hat{\lambda} = \frac{\left({}^{y}/_{\sigma_y^2}\right) + \left({}^{v}/_{\sigma_v^2}\right)}{\left(\sigma_y^{-2}\right) + \left(\sigma_v^{-2}\right)}$$

This is simply the weighted mean. Putting this into the original χ^2 equation and summing over all bins gives:

$$\chi^2 = \sum_i^N \frac{(y_i - v_i)^2}{\sigma_{y_i}^2 + \sigma_{v_i}^2}$$

This was implemented into a short program (given in Appendix B) to manipulate the ROOT files from Rivet. The results are shown in Table 2.3 below shown as the χ^2 value divided by the number of degrees of freedom (number of observations – number of free parameters i.e. 2).

	Default Tune	**2008 Tune**	**Jimmy Tune**	**Bahr Tune**	**Purple Tune**	**Low Tune**	**High Tune**
χ^2 Total:	17.803	17.874	20.701	**17.133**	17.948	18.083	17.165
χ^2 Toward:	22.492	20.855	19.720	19.754	19.085	23.415	**18.998**
χ^2 Transverse:	25.074	22.616	21.917	23.032	**21.180**	24.355	21.714
χ^2 TransMAX:	9.056	4.832	7.393	5.921	4.645	8.793	**4.197**
χ^2 TransMIN:	0.848	0.576	1.710	0.534	0.779	0.861	**0.499**
χ^2 TransDIF:	3.178	2.055	**1.591**	2.497	1.770	2.930	1.899
χ^2 Away:	**18.326**	20.137	28.620	19.058	21.436	18.759	21.090
χ^2 Particle Density:	1.222	0.970	6.578	**0.816**	2.513	1.250	1.227
χ^2 P_T^{sum}:	6.744	3.637	**1.943**	4.417	2.269	6.585	2.992

Table 2.3: χ^2 results for various tunes

Overall, the Bahr tune works best in all areas of the analysis and in the charged particle density. As discussed earlier, the Jimmy tune does well with the p_T sum densities. The "Purple" tune worked the best in the transverse region, which is as expected, as that spot was found in the transverse χ^2 valley. Interestingly the "high" tune did well in a number of areas.

2.6 Searching for an Improved Tune

Using the tools written for the χ^2 testing in the previous section, 64 simulations were run to find a set of parameters with overall χ^2 values lower than those already seen. The range investigated was μ^2 = 1.1 – 2.5 GeV^2, and p_T^{min} = 3.0 – 4.4 GeV, both in increments of 0.2. This range seems appropriate for the eikonal model as given in the Jimmy manual [21]. The results are shown in the contour plot below.

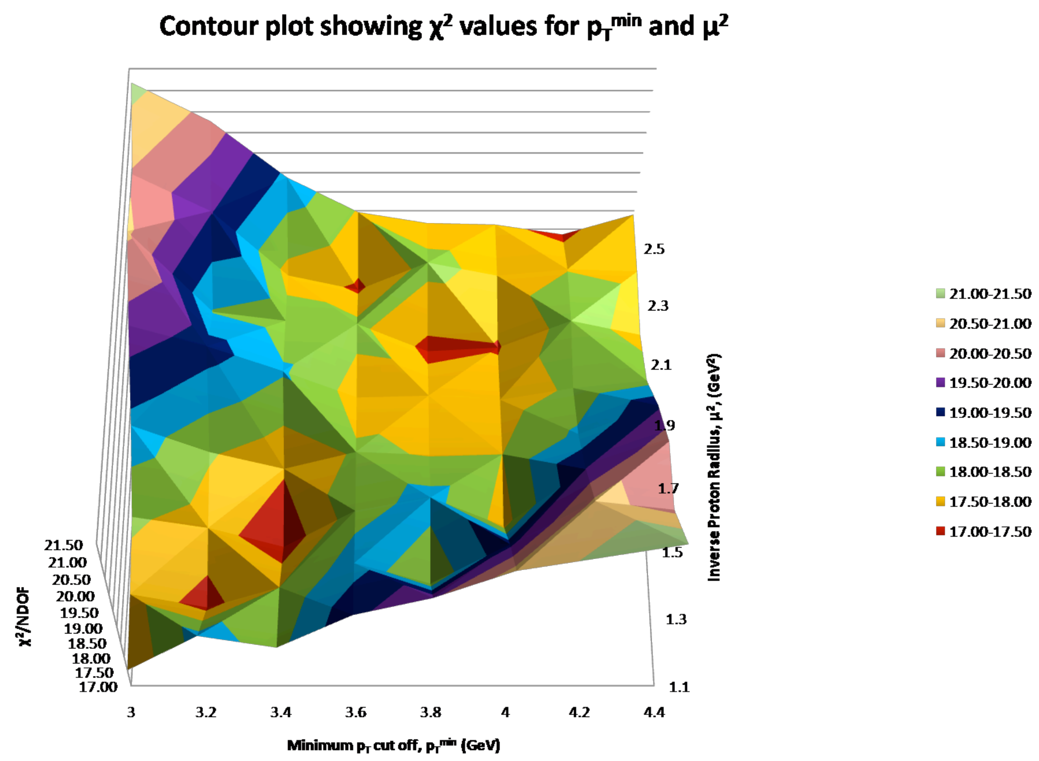


Figure 2.28: Contour plot showing overall χ^2 values for p_T^{min} and μ^2

As shown above, the χ^2 results above form a valley similar to that seen in Figure 2.22. The lowest parts of the valley are shown in red, and the lowest χ^2 value is 17.1329 at μ^2 = 1.5 GeV^2 and p_T^{min} = 3.4 GeV. This agrees exactly with the findings in the "Underlying Events in Herwig++" paper [16] and shows that the tune discussed in that paper is the best possible in that given range of parameters. This is also an important result, as that paper used the CDF 2001 data (Tevatron Run 1) to find that tune, which is a QCD 2-to-2 process. Here, Drell-Yan data was used from CDF 2008 data (Tevatron Run 2) and the same tune was found.

2.6.1 Adding Additional Soft Scatters

As shown earlier in Figure 2.16 and Figure 2.17, Herwig++ does not produce enough activity in the p_T sum distributions. This means that Herwig++ needs to be somehow tuned to produce more soft scatters to help increase the p_T sum densities. However, the soft part of the MPI model is not tuneable, and is currently hardcoded to prevent numerical instabilities [30].

One way of adding soft scatters is to simply increase p_T^{min}. This forces the soft part of the MPI model to play a larger role and subsequently lowers the ratio of hard to soft scatters. As shown earlier in the valley plots, an increase in p_T^{min} naturally requires an increase in μ^2. Although this is an unconventional approach to tuning, and would be deemed unwise ramping up p_T^{min} when using Jimmy, this method uses the new soft scatter model in Herwig++ to its potential. Also, the success of the "High" tune in Table 2.3, suggests that there may be interesting activity further up in the valley.

As an extension to Figure 2.28, the range of parameters was increased to μ^2 = 1.1 – 3.1 GeV^2, and p_T^{min} = 3.0 – 4.8 GeV, and then χ^2 tested to look for any improvements. The results are shown in Figure 2.29 below.

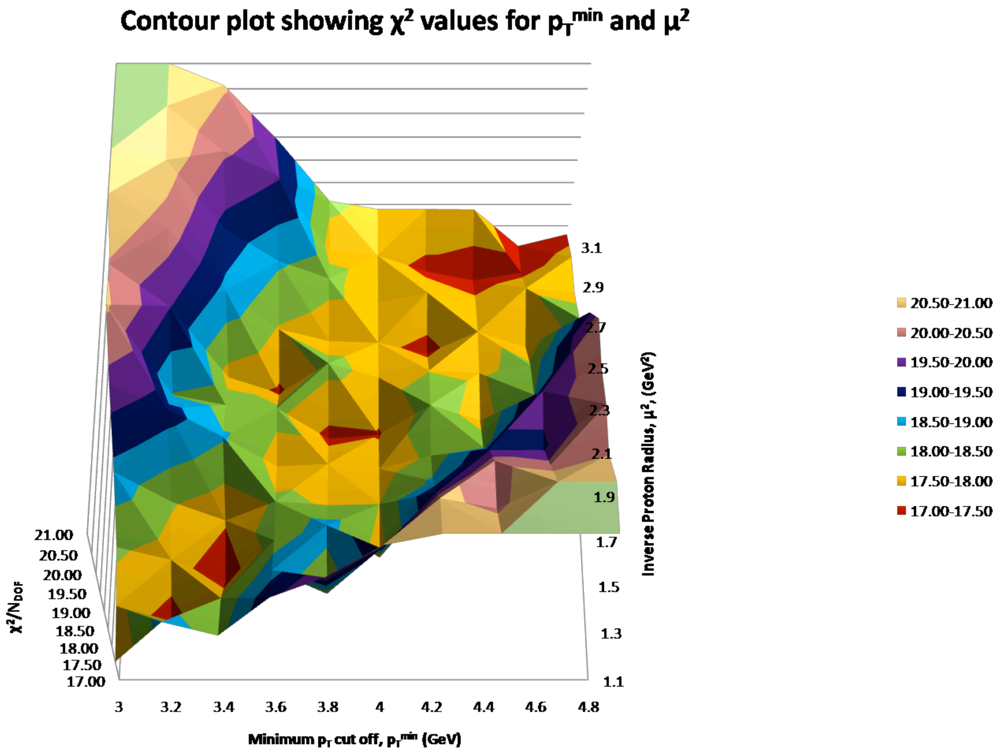


Figure 2.29: Contour plot showing overall χ^2 values for p_T^{min} and μ^2 extended to high values of p_T^{min}

As shown in Figure 2.29 above, there is a significant red area at the top part of the valley. The lowest χ^2 value in this region is 16.581 corresponding to μ^2 = 2.9 GeV^2 and p_T^{min} = 4.5, which we will call the *"Soft"* tune.

	Default Tune	Bahr Tune	Soft Tune
χ^2 Total:	17.803	17.133	16.581
χ^2 Toward:	22.492	19.754	17.769
χ^2 Transverse:	25.074	23.032	19.543
χ^2 TransMAX:	9.056	5.921	3.980
χ^2 TransMIN:	0.848	0.534	0.530
χ^2 TransDIF:	3.178	2.497	1.678
χ^2 Away:	18.326	19.058	19.798
χ^2 Particle Density:	1.222	0.816	1.488
χ^2 P_T^{sum}:	6.744	4.417	2.590

Table 2.4: χ^2 values for the Default, "Bahr" and "Soft" tunes

It can be seen from the χ^2 values that this new "Soft" tune outperforms the Default and the Bahr tune in all areas except in the away region and the charged particle density. This is because this tune produces more p_T sum density, lowering the χ^2 values in all the regions as shown in Figure 2.31.

However, as a result it produces slightly too much charged particle density, as shown the green line in Figure 2.30, hence the higher χ^2 value for the particle density. This was already observed in the case of the Jimmy tune in Section 2.4.

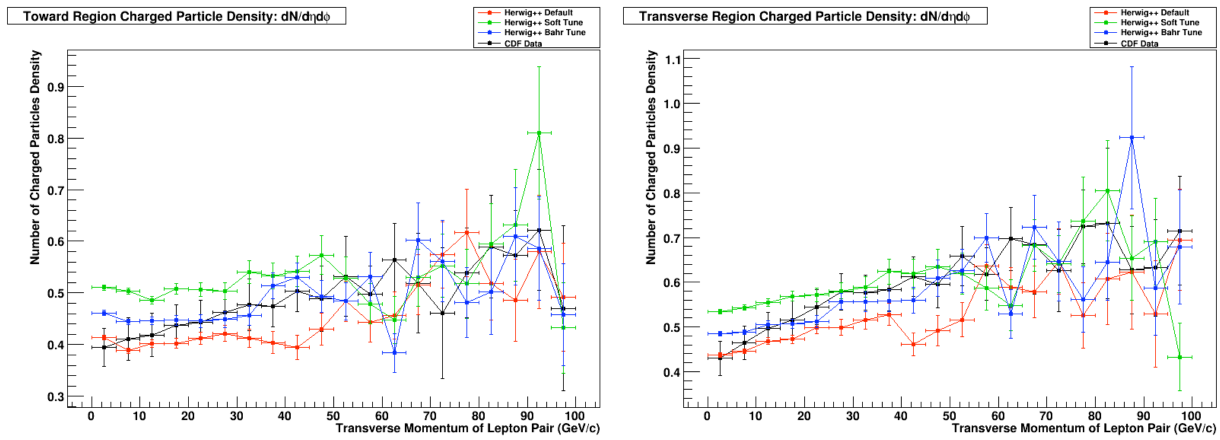


Figure 2.30: Charged Particle Densities for the Default (red), Soft (green) and Bahr (blue) tunes

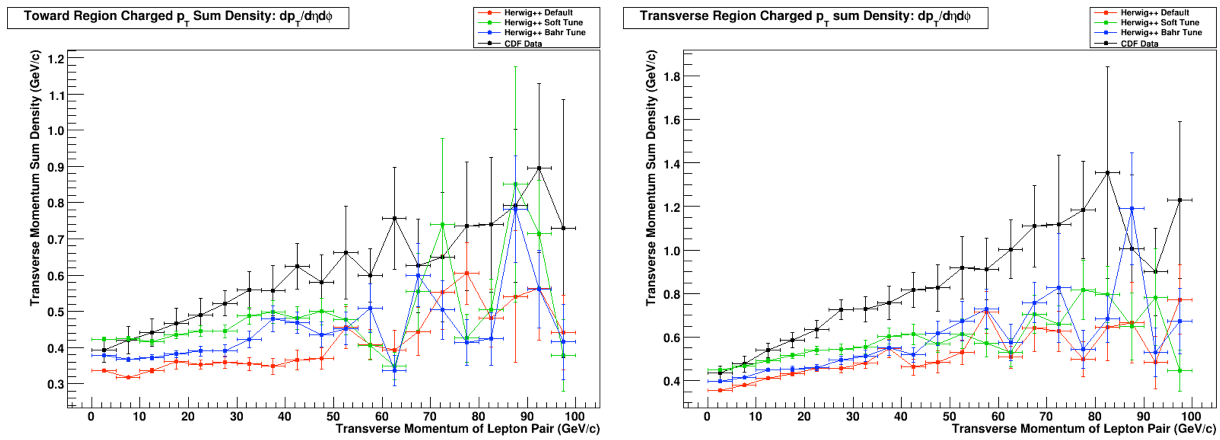


Figure 2.31: p_T sum densities for the Default (red), Soft (green) and Bahr (blue) tunes

2.7 Validation using the CDF 2001 analysis

The soft tune has a much lower χ^2 value due to the fact that it produces much more p_T sum, without overcompensating and affecting the particle densities too much. To check whether if this tune is only optimised for the CDF 2008 analysis or can be used with other data, it was checked against another analysis.

As shown in Section 2.2.1, the CDF 2001 analysis also lacks activity in the p_T sum densities, which means that the "Soft" tune should be a good candidate. A χ^2 was also done to compare the results of both tunes and the results are shown below. Only the leading jet data > 20 GeV is shown as this is the region where the underlying event is most apparent.

	Default	Bahr	Soft
χ^2 Toward Particle Density	65.8635	127.902	144.156
χ^2 Transverse Particle Density	10.0742	37.2214	47.1426
χ^2 Toward p_T sum	6.15794	1.42627	2.36793
χ^2 Transverse p_T sum	4.15282	0.731308	2.59454
χ^2 Total	15.9016	32.2026	38.1859

Table 2.5: χ^2 values for the Default, “Bahr” and “Soft” tunes using the CDF 2001 analysis

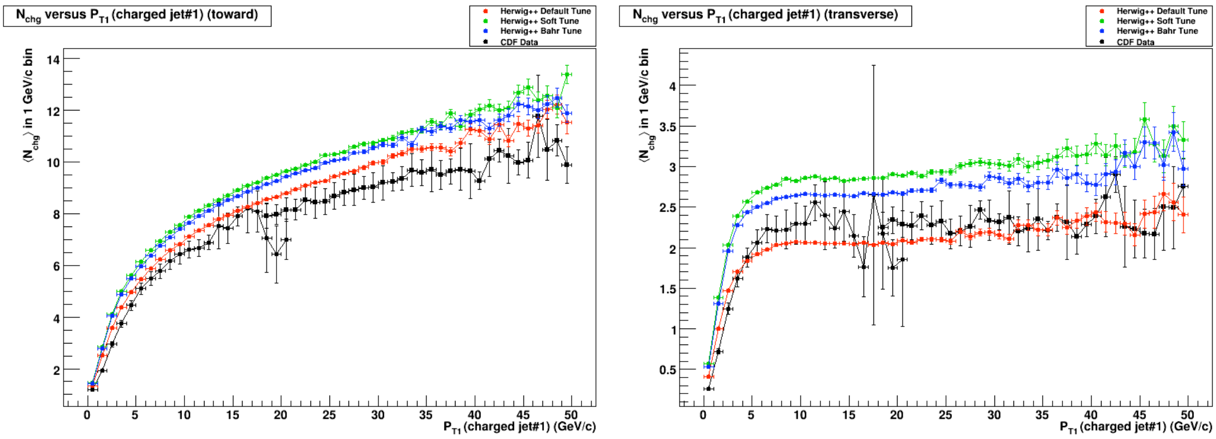


Figure 2.32: Charged Particle Densities for the Default (red), Soft (green) and Bahr (blue) tunes

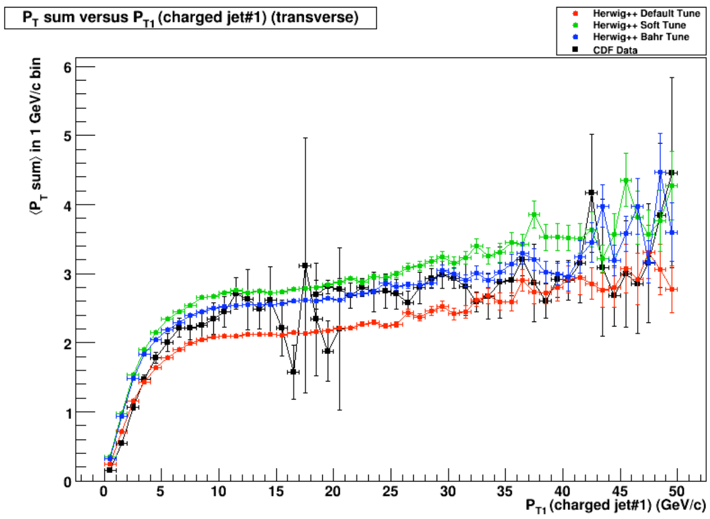


Figure 2.33: Transverse p_T sum densities for the Default (red), Soft (green) and Bahr (blue) tunes

As shown in Table 2.5, both the Bahr and Soft tunes do poorly with respect to charged particle densities. This is because these tunes produce too much overall charged multiplicity as shown in Figure 2.32. However, as expected, these tunes do well where the p_T sum densities are involved as shown by the low χ^2 values and in Figure 2.33. However, unlike the CDF 2008 analysis, the overall χ^2 values for these tunes are higher than the default, with the Soft tune being the worst. In conclusion, for the CDF 2001 analysis, the default tune is the best (as found in the Herwig++ 2.3 release note [26]), followed by the Bahr tune, which outperformed the default tune on p_T sum densities, and then the Soft tune.

3 Summary and Conclusion

3.1 Summary

It was shown that Herwig++ behaves very similar to its forerunners Fortran Herwig and Jimmy. In the CDF 2001 analysis, Herwig++ had a slight trend to produce too much multiplicity in all the regions, most notably the toward region, and too little p_T sum in all the regions, which agreed with results from that particular paper. Herwig++ also agreed with observations in the CDF 2004 analysis, and reasons for why it may not have agreed with certain data were found. In the default tune of Herwig++ it was shown to agree with observations from the analysis, and from other sources, by showing a lack of charged multiplicity and p_T sum density, which is what this project was aiming to improve by finding a better tune. The reason that these areas were all checked was so that any differences between the CDF data and the Monte Carlo data would be due to the generator itself and not the Rivet analysis.

Various different tunes were investigated to try and improve the Monte Carlo agreement with the data, all found from various sources. One particularly interesting source was the "Underlying Events in Herwig++" paper [16], which showed a χ^2 valley in μ^2 and p_T^{min} space, where the minimum of that valley was the best tune. Various tunes were investigated in this valley.

To try and distinguish which tune was better than the others, a χ^2 test was used as a goodness of fit test against the CDF data. It showed that the tune found in the paper mentioned above (referred to as the Bahr tune) worked very well by improving the charged multiplicity. Another tune developed especially for Jimmy, seemed to increase the p_T sum density but at the expense of producing too much charged multiplicity. A tune chosen high up in the valley did well in many areas, although the Bahr tune was slightly better.

In an attempt to find a better tune than the Bahr tune, a range of parameters were tried in an acceptable range (μ^2 = 1.1 – 2.5 GeV^2, and p_T^{min} = 3.0 – 4.4 GeV, both in increments of 0.2), and then each tune was checked using the χ^2 test. The conclusion was that in this range, the best tune was μ^2 = 1.5 GeV^2, and p_T^{min} = 3.4 GeV, which corresponds exactly to the Bahr tune. This is an important result, as the same tune was found using a different process (Drell-Yan as opposed to QCD 2-to-2) and from a different experiment (Tevatron Run 2 instead of Run 1).

To improve Monte Carlo agreement with CDF data, more soft scatters needed to be included. Herwig++ 2.3.0 automatically adds soft interaction below p_T^{min}. However, this feature is not tuneable and is currently hardcoded to prevent numerical instabilities. A method to increase the soft scatters was to vastly increase p_T^{min}. This means there is a larger range where the soft MPI model can work, and reduces the ratio of hard to soft scatters. Naturally, a higher p_T^{min} means that a higher μ^2 is needed. Also, the success of the earlier venture high up in the valley indicated that there may be something interesting at high values of p_T^{min}.

As before, a range of parameters were investigated but extended to μ^2 = 1.1 – 3.1 GeV^2, and p_T^{min} = 3.0 – 4.8 GeV. Using these parameters in Jimmy would have been deemed unwise, but the purpose of this task was to use the new soft scatter model in Herwig++ to its potential. A distinct area was found at the top of the valley which had very low χ^2 values. The lowest part corresponded to μ^2 = 2.9 GeV^2 and p_T^{min} = 4.5 (which is referred to as the soft tune). This tune also outperformed the Bahr tune in all areas (toward, transverse, etc.).

The reason for these low χ^2 values was that the p_T sum densities had been greatly improved, while minimising the trade-off by generating too much charged multiplicity.

To investigate if this tune was only specific to this data set, or could be used with other analyses investigating the underlying event, it was used with the CDF 2001 analysis. The soft tune did very poorly with regard to charged multiplicity, as it generated too much activity. It did however do very well in the p_T sum densities, but on the whole it did worse than the default tune and the Bahr tune.

3.2 Conclusion

In conclusion, two good tunes were found which modelled Drell-Yan data well. However the Soft tune was at the extreme of the spectrum, and performed poorly when used with QCD 2-to-2 data. The Bahr tune worked almost just as well with Drell-Yan data, and has a better ability to work well with other data, which is why the findings of this report favour this tune.

No tune was able to effectively model both the charged multiplicity and the p_T sum density. A tune strong with regard to p_T sum density would hinder charged multiplicity and vice versa. This is a problem with the physics in Herwig++ and cannot be fully resolved by changing μ^2 and p_T^{min}. The reason for the strength of the Soft tune is that it finds the compromise between the both, but this compromise is no longer applicable when used with other analyses.

One possible route to improve this is modifying the MPIHandler class in Herwig++ to allow the soft part of the MPI model to be tuned. Then more soft scatters could be added without having to go to impractical values of p_T^{min}.

3.3 Final Words

The underlying event is an important area in High Energy Physics which is important to our understanding of particle physics. Good progress has been made throughout the years to improve this area. Two tunes were found in this project which can fit the CDF data well, however both the tunes fail to fully describe all of the features of the underlying event. The underlying event will become more apparent than ever at the LHC, and it will be possible to learn a lot about the energy dependence of the underlying event by comparing Tevatron and LHC data. LHC data will also help show what areas of the physics in Herwig++ will need to be improved to allow it to model the underlying event more accurately.

Appendix A

A typical input file for Herwig++

```
##################################################
# Tune parameters
##################################################
cd /Herwig/UnderlyingEvent

# Inverse Radius (PRRAD)
set MPIHandler:InvRadius 1.5

# Cutoff for secondary scatters (PTJIM)
set KtCut:MinKT 3.4
#
# This should always be 2*MinKT!!
set UECuts:MHatMin 6.8


# Underlying Event Option (JMUEO)
# set /Herwig/Shower/ShowerHandler:MPIHandler NULL

##################################################
# Technical parameters for this run
##################################################
cd /Herwig/Generators
set LHCGenerator:NumberOfEvents 10000000
set LHCGenerator:RandomNumberGenerator:Seed 31122001
set LHCGenerator:DebugLevel 1
set LHCGenerator:PrintEvent 10
set LHCGenerator:MaxErrors 1000000

##################################################
# Tevatron physics parameters (override defaults)
##################################################
set LHCGenerator:EventHandler:LuminosityFunction:Energy 1960.0
set LHCGenerator:EventHandler:BeamB /Herwig/Particles/pbar-

##################################################
# Matrix Elements for hadron-hadron collisions
# (by default only gamma/Z switched on)
##################################################
cd /Herwig/MatrixElements

# Drell-Yan Z/gamma
insert SimpleQCD:MatrixElements[0] MEqq2gZ2ff
# Drell-Yan W
# insert SimpleQCD:MatrixElements[0] MEqq2W2ff
# gamma-gamma
# insert SimpleQCD:MatrixElements[0] MEGammaGamma
# gamma+jet
# insert SimpleQCD:MatrixElements[0] MEGammaJet
# gg/qqbar -> Higgs
# insert SimpleQCD:MatrixElements[0] MEHiggs
# higgs+jet
# insert SimpleQCD:MatrixElements[0] MEHiggsJet
# QCD 2-2 scattering
# insert SimpleQCD:MatrixElements[0] MEQCD2to2
# top-antitop production
# insert SimpleQCD:MatrixElements[0] MEHeavyQuark


#################################################
#  Useful analysis handlers for HepMC related output
##################################################
# Schematic overview of an event (requires --with-hepmc to be set at configure time
# and the graphviz program 'dot' to produce a plot)
# insert LHCGenerator:AnalysisHandlers 0 /Herwig/Analysis/Plot
# A HepMC dump file (requires --with-hepmc to be set at configure time)
insert LHCGenerator:AnalysisHandlers 0 /Herwig/Analysis/HepMCFile
set /Herwig/Analysis/HepMCFile:PrintEvent 1000000
set /Herwig/Analysis/HepMCFile:Format GenEvent
set /Herwig/Analysis/HepMCFile:Units GeV_mm

##################################################
# Save run for later usage with 'Herwig++ run'
##################################################
saverun TVT LHCGenerator
```

Appendix B

Source code for the program written for χ^2 testing

```
// Program for calculating Chi squared values for the
// CDF_2008_NOTE_9351 analysis

#include <iostream>
#include <cmath>
#include <TGraphAsymmErrors.h>
#include <TFile.h>

using namespace std;

double graph1();
double graph2();
double graph3();
double graph4();
double graph5();
double graph6();
double graph7();
double graph8();
double graph9();
double graph10();
double graph11();
double graph12();
double graph13();
double graph14();
double graph15();
double graph16();
double graph17();
double graph18();
double graph19();
double graph20();
double graph21();


const int params = 2;

//Open files
TFile* realData = new TFile("/usr/share/root/macros/CDF_2008_NOTE_9351.root");
TFile* mcData = new TFile("Herwig++.root");

int main(){

  //Do chi-squared analysis
  double chiTotal = 0;
  double chiToward = 0;
  double chiTrans = 0;
  double chiAway = 0;
  double chiTransMin = 0;
  double chiTransMax = 0;
  double chiTransDif = 0;
  double chiChg = 0;
  double chiPtsum = 0;
  double chi1 = graph1(); chiTotal += chi1; chiChg += chi1; chiToward += chi1;
  double chi2 = graph2(); chiTotal += chi2; chiChg += chi2; chiTrans += chi2;
  double chi3 = graph3(); chiTotal += chi3; chiChg += chi3; chiTransMax += chi3;
  double chi4 = graph4(); chiTotal += chi4; chiChg += chi4; chiTransMin += chi4;
  double chi5 = graph5(); chiTotal += chi5; chiChg += chi5; chiTransDif += chi5;
  double chi6 = graph6(); chiTotal += chi6; chiChg += chi6; chiAway += chi6;
  double chi7 = graph7(); chiTotal += chi7; chiPtsum += chi7; chiToward += chi7;
  double chi8 = graph8(); chiTotal += chi8; chiPtsum += chi8; chiTrans += chi8;
  double chi9 = graph9(); chiTotal += chi9; chiPtsum += chi9; chiTransMax += chi9;
  double chi10 = graph10(); chiTotal += chi10; chiPtsum += chi10; chiTransMin += chi10;
  double chi11 = graph11(); chiTotal += chi11; chiPtsum += chi11; chiTransDif += chi11;
  double chi12 = graph12(); chiTotal += chi12; chiPtsum += chi12; chiAway += chi12;
  double chi13 = graph13(); chiTotal += chi13; chiToward += chi13;
  double chi14 = graph14(); chiTotal += chi14; chiTrans += chi14;
  double chi15 = graph15(); chiTotal += chi15; chiAway += chi15;
  double chi16 = graph16(); chiTotal += chi16; chiToward += chi16;
  double chi17 = graph17(); chiTotal += chi17; chiTrans += chi17;
  double chi18 = graph18(); chiTotal += chi18;
  double chi19 = graph19(); chiTotal += chi19;
  double chi20 = graph20(); chiTotal += chi20;
  double chi21 = graph21(); chiTotal += chi21;
  cout << "\nChi-squared/NDOF Values:" << endl;
  cout << "\nd01-x01-y01: " << chi1/(20-params) << endl;
  cout << "d02-x01-y01: " << chi2/(20-params) << endl;
  cout << "d03-x01-y01: " << chi3/(20-params) << endl;
  cout << "d04-x01-y01: " << chi4/(20-params) << endl;
  cout << "d05-x01-y01: " << chi5/(20-params) << endl;
  cout << "d06-x01-y01: " << chi6/(20-params) << endl;
  cout << "d07-x01-y01: " << chi7/(20-params) << endl;
  cout << "d08-x01-y01: " << chi8/(20-params) << endl;
```

```
  cout << "d09-x01-y01: " << chi9/(20-params) << endl;
  cout << "d10-x01-y01: " << chi10/(20-params) << endl;
  cout << "d11-x01-y01: " << chi11/(20-params) << endl;
  cout << "d12-x01-y01: " << chi12/(20-params) << endl;
  cout << "d13-x01-y01: " << chi13/(20-params) << endl;
  cout << "d14-x01-y01: " << chi14/(20-params) << endl;
  cout << "d15-x01-y01: " << chi15/(20-params) << endl;
  cout << "d16-x01-y01: " << chi16/(20-params) << endl;
  cout << "d17-x01-y01: " << chi17/(20-params) << endl;
  cout << "d18-x01-y01: " << chi18/(20-params) << endl;
  cout << "d19-x01-y01: " << chi19/(31-params) << endl;
  cout << "d20-x01-y01: " << chi20/(30-params) << endl;
  cout << "d21-x01-y01: " << chi21/(30-params) << endl;
  cout << "\nChi-squared Total:       " << (chiTotal/(451-params)) << endl;
  cout << "\nChi-squared Toward:      " << (chiToward/(80-params)) << endl;
  cout << "Chi-squared Transverse:  " << (chiTrans/(80-params)) << endl;
  cout << "Chi-squared TranMAX:     " << (chiTransMax/(40-params)) << endl;
  cout << "Chi-squared TranMIN:     " << (chiTransMin/(40-params)) << endl;
  cout << "Chi-squared TranDIF:     " << (chiTransDif/(40-params)) << endl;
  cout << "Chi-squared Away:        " << (chiAway/(40-params)) << endl;
  cout << "\nChi-squared Chg Density: " << (chiChg/(120-params)) << endl;
  cout << "Chi-squared PTsum:       " << (chiPtsum/(120-params)) << endl;

  //Close Files
  realData->Close();
  mcData->Close();

  return 0;
}

double graph1(){

  //Get Plots
  TGraphAsymmErrors *graph1 = (TGraphAsymmErrors*)realData->Get("Graph;1");
  TGraphAsymmErrors *d01_x01_y01 = (TGraphAsymmErrors*)mcData->Get("CDF_2008_NOTE_9351/d01-
x01-01");

  //Build Array's with y-values
  Double_t *cdf = graph1->GetY();
  Double_t *mc = d01_x01_y01->GetY();

  //Do chi-squared
  double chi = 0;
  int points = graph1->GetN();
  for (int i=0; i < points; i++){
    chi += pow(mc[i]-cdf[i],2)/(pow(graph1->GetErrorY(i),2)+pow(d01_x01_y01-
>GetErrorY(i),2));
  }
  return chi;
}

double graph2(){

  TGraphAsymmErrors *graph2 = (TGraphAsymmErrors*)realData->Get("Graph;2");
  TGraphAsymmErrors *d02_x01_y01 = (TGraphAsymmErrors*)mcData->Get("CDF_2008_NOTE_9351/d02-
x01-01");

  Double_t *cdf = graph2->GetY();
  Double_t *mc = d02_x01_y01->GetY();

  double chi = 0;
  int points = graph2->GetN();
  for (int i=0; i < points; i++){
    chi += pow(mc[i]-cdf[i],2)/(pow(graph2->GetErrorY(i),2)+pow(d02_x01_y01-
>GetErrorY(i),2));
  }
  return chi;
}

double graph3(){

        ..............
        ..............

       //large repeated sections – not very elegant but hey, it works! ☺

        ..............
        ..............

double graph21(){

  TGraphAsymmErrors *graph21 = (TGraphAsymmErrors*)realData->Get("Graph;21");
  TGraphAsymmErrors *d21_x01_y01 = (TGraphAsymmErrors*)mcData->Get("CDF_2008_NOTE_9351/d21-
x01-1");

  Double_t *cdf = graph21->GetY();
  Double_t *mc = d21_x01_y01->GetY();
```

```
  double chi = 0;
  int points = graph21->GetN();
  for (int i=0; i < points; i++){
    chi += pow(mc[i]-cdf[i],2)/(pow(graph21->GetErrorY(i),2)+pow(d21_x01_y01-
>GetErrorY(i),2));
  }
  return chi;
}
```

Appendix C

Typical ROOT macro for plotting a graph

```
// ROOT macro for plotting experimental data
// against Monte Carlo data
// d03-x01-y01

{
  // Set some defaults
  gROOT->Reset();
  gROOT->SetStyle("Plain");
  gStyle->SetCanvasBorderMode(0); // turn off canvas borders
  gStyle->SetTitleFontSize(0.03);
  gStyle->SetTitleSize(0.03,axis="xy");
  gStyle->SetLabelSize(0.03,axis="xy");
  gStyle->SetNdivisions(512);
  gStyle->SetOptLogy(0);

  //Open files
  TFile* realData = new TFile("/usr/share/root/macros/CDF_2001_S4751469.root");
  TFile* herwigData = new TFile("Herwig++.root");
  TFile* jimmyData = new TFile("Jimmy.root");

  // Create a canvas
  TCanvas *c1 = new TCanvas("c1", "Canvas 1", 400, 10, 1000, 700);

  //Make multigraph
  TMultiGraph *mg = new TMultiGraph("multiGraph","N_{chg} versus P_{T1} (charged jet#1)
(toward) min-bias");

 //Real Data Plot
  TGraphAsymmErrors *rd1 = (TGraphAsymmErrors*)realData->Get("Graph;7");
  rd1->SetMarkerStyle(8); //black circles
  rd1->SetMarkerSize(0.8);
  mg->Add(rd1);

  //Herwig++ Plot
  TGraphAsymmErrors *hd1 = (TGraphAsymmErrors*)herwigData->Get("CDF_2001_S4751469/d03-x01-
y01");
  hd1->SetMarkerColor(4);
  hd1->SetMarkerStyle(8);
  hd1->SetMarkerSize(0.8);
  hd1->SetLineColor(4);
  mg->Add(hd1);

  //Jimmy Plot
  TGraphAsymmErrors *jd1 = (TGraphAsymmErrors*)jimmyData->Get("CDF_2001_S4751469/d03-x01-
y01");
  jd1->SetMarkerColor(2);
  jd1->SetMarkerStyle(8);
  jd1->SetMarkerSize(0.8);
  jd1->SetLineColor(2);
  mg->Add(jd1);

  //Draw MultiGraph
  mg->Draw("alp");
  mg->GetXaxis()->SetTitle("P_{T1} (charged jet#1) (GeV/c)");
  mg->GetYaxis()->SetTitle("#LTN_{chg}#GT in 1 GeV/c bin");

  //Legend
  TLegend *myLegend = new TLegend(0.76,0.905,0.90,0.995);
  myLegend->AddEntry(hd1,"Herwig++ Data","P");
  myLegend->AddEntry(jd1,"Jimmy Data","P");
  myLegend->AddEntry(rd1,"CDF Data","P");
  myLegend->SetFillColor(10);
  myLegend->Draw();

  // Print in .eps and .gif
  c1->Print("d03-x01-y01.eps");
  c1->Print("d03-x01-y01.gif");

  //Close Files
  realData->Close();
  herwigData->Close();
  jimmyData->Close();

}
```

Bibliography

1. *Simulation of multiple partonic interactions in Herwig++.* **Manuel Bahr, Stefan Gieseke (Karlsruhe U., ITP) , Michael H. Seymour (Manchester U. & CERN).** March 2008, JHEP 0807:076. arXiv:0803.3633.

2. **Martin, B. R.** *Nuclear and Particle Physics An Introduction.* s.l. : Wiley, 2007.

3. *Structure of the Standard Model.* **Langacker, P.** arXiv:hep-ph/0304186v1.

4. *PHASM442 Lecture Notes.* **Lancaster, Mark.** 2008-2009.

5. *Leptonic CP violation studies at MiniBooNE in the (3+2) sterile neutrino oscillation hypothesis.* **G. Karagiorgi, A. Aguilar-Arevalo, J.M. Conrad, M.H. Shaevitz (Columbia U.) , K. Whisnant (Iowa State U.) , M. Sorel (Valencia U., IFIC) , V. Barger (Wisconsin U., Madison).** September 2006, Phys. Rev. D75 013011. hep-ph/0609177.

6. *Using Drell-Yan to probe the underlying event in Run II at Collider Detector at Fermilab (CDF).* **Kar, Deepak.** December 2008, FERMILAB-THESIS-2008-54.

7. CERN FAQ - LHC: The Guide. [Online] 2008. http://cdsmedia.cern.ch/img/CERN-Brochure-2008-001-Eng.pdf.

8. *Tevatron.* **Wilson, R. R.** 1978, Fermilab. FERMILAB-TM-0763.

9. **Gilliland, Ben.** Metro. [Newspaper]. 20 February 2009.

10. **McGinnis, Dave.** Fermilab Run II Handbook. [Online] http://www-bd.fnal.gov/runII/index.html.

11. *The CDF-II detector: Technical design report.* **CDF-II Collaboration, R. Blair et. al.** November 1996. FERMILAB-PUB-96-390-E.

12. *Prediction for minimum bias and the underlying event at LHC energies.* **A. Moraes (Brookhaven), C. Buttar (Glasgow U.) , I. Dawson (Sheffield U.).** 2007, Eur.Phys.J.C50, pp. 435-466.

13. *Min-Bias at the Tevatron.* **CDF Collaboration, Rick Field.** October 2007. FERMILAB-CONF-07-465-E.

14. **Moraes, Arthur M.** Minimum bias interactions and the underlying event. [Online] January 2003. http://www.hep.ucl.ac.uk/~jmb/Durham2003/arthur.pdf.

15. *Charged Jet Evolution and the underlying event in proton - antiproton collisions at 1.8 TeV.* **CDF Collaboration, A. A. Affolder et. al.** 2001, Phys. Rev. D65 092002.

16. *Underlying events in Herwig++.* **Manuel Bahr, Stefan Gieseke (Karlsruhe U., ITP) , Michael H. Seymour (CERN & Manchester U.).** September 2008. arXiv:0809.2669.

17. *The Underlying event in hard interactions at the Tevatron anti-p p collider.* **CDF Collaboration, Darin E. Acosta et al.** 2004, Phys.Rev. D70 072002. arXiv:hep-ex/0404004.

18. **Cowan, Glen.** *Statistical Data Analysis.* s.l. : Oxford, 1998.

19. **Malzacher, Peter.** TRandom3. [Online] 1999. http://root.cern.ch/root/html/TRandom3.html.

20. *Multiparton interactions in photoproduction at HERA.* **J.M. Butterworth (University Coll. London), Jeffrey R. Forshaw (Manchester U.) , M.H. Seymour (CERN).** 1996, Z.Phys.C72, pp. 637-646. hep-ph/9601371.

21. **Butterworth, J.M.** Jimmy4: Multiparton Interactions in Herwig for the LHC (Draft). [Online] http://projects.hepforge.org/jimmy/draft20051116.ps.

22. *Herwig 6.5.* **G. Corcella, I.G. Knowles, G. Marchesini, S. Moretti, K. Odagiri, P. Richardson, M.H. Seymour and B.R. Webber.** JHEP 0101 (2001) 010. hep-ph/0011363; hep-ph/0210213.

23. ThePEG / Herwig++ Wiki. [Online] http://projects.hepforge.org/herwig/trac/wiki.

24. *Herwig++ 2.1 Release Note.* **M. Bahr, et. al.** November 2007. arXiv:0711.3137.

25. *Herwig++ 2.2 Release Note.* **M. Bahr, et. al.** April 2008. arXiv:0804.3053.

26. *Herwig++ 2.3 Release Note.* **M. Bahr, et. al.** December 2008. arXiv:0812.0529.

27. Rivet Documentation. [Online] http://projects.hepforge.org/rivet/trac/wiki/.

28. Root Users Guide 5.21. [Online] December 2008. ftp://root.cern.ch/root/doc/Users_Guide_5_21.pdf.

29. **Buckley, Andy.** Generator comparisons and tuning. [Presentation Slides]. 3 December 2008. ATLAS MC Meeting.

30. *Herwig++ Physics and Manual.* **M. Bahr, et. al.** 2008, Eur.Phys.J.C58, pp. 639-707. arXiv:0803.0883.

31. *The Underlying Event and the Total Cross Section from Tevatron to the LHC.* **Manuel Bahr (Karlsruhe U., ITP & University Coll. London) , Jonathan M. Butterworth (University Coll. London) , Michael H. Seymour (Manchester U. & CERN).** 2009, JHEP 0901:065. arXiv:0806.2949.

32. Root 5.20 Reference Guide. [Online] June 2008. http://root.cern.ch/root/html520/ClassIndex.html.