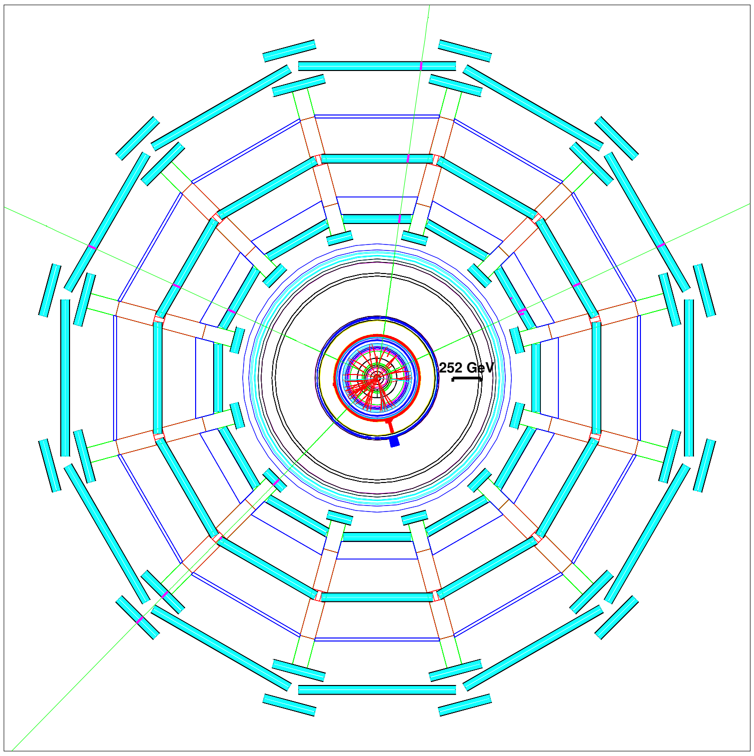
In Situ Jet Energy Calibration In Atlas Based On Z+Jet Events

By

Hermen Jan Hupkes



SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE AT THE

Leiden Institute of Physics

OF THE

University of Leiden

P. O. Box 9504

2300 RA Leiden

The Netherlands

IN SITU JET ENERGY CALIBRATION IN ATLAS BASED ON Z+JET EVENTS

Hermen Jan Hupkes
Mathematisch Instituut
Universiteit Leiden
P.O. Box 9512
2300 RA Leiden, The Netherlands
hhupkes@math.leidenuniv.nl

Supervised by

Irene Vichou
Aristotle University of Thessaloniki
Thessaloniki, Greece
irene.vichou@cern.ch

MARCH 2004

Contents

Abstract

We study the performance of the Athena jet reconstruction algorithms on a sample of simulated Z+jet events, in view of using such a sample to calibrate these algorithms during the initial low luminosity run of the LHC. A new set of H1 weights is calculated based on meusarable quantities extracted from these events and the performance of this new set is found to be similar to that of the existing weights. In addition, we propose a new implementation of the $K_\perp$ algorithm which reduces the expected running time from $O(N^3)$ to $O(N^2)$ for a certain class of recombination schemes, where N is the number of registered energy deposits.

Acknowledgements

This research was conducted during the 2003 Summer Studentship Programme at CERN in the TileCal group of the ATLAS collaboration. The author would like to thank CERN and in particular the TileCal group for the enjoyed hospitality. In addition, special thanks go to my mother Rineke Hupkes for her moral support and grammatical advice.

Chapter 1

Introduction

The Large Hadron Collider (LHC) at CERN is designed to be a proton-proton collider operating with 14 TeV center of mass energy at a luminosity of 10^{34} cm^{-2}s^{-1} [30, 31]. There will be beam-crossings at 25 ns intervals with an average number of 23 events per crossing. The high energy and luminosity of the LHC offers a large scala of physics opportunities, ranging from the precise measurement of the properties of known objects to the exploration of new phenomena at the high energy frontier. The first collisions are planned for the summer of the year 2006, after which the LHC is scheduled to remain operational for at least twenty years. During approximately the first two years of operation, the collider will operate at one tenth of the design luminosity. This phase will allow for the testing and calibration of the various experiments present at the LHC.

One of these experiments which will operate at the LHC is based upon the ATLAS detector [5]. The primary goal of this experiment is to operate at the high luminosity present in the LHC environment with a detector that is capable of exploiting the full physics potential offered by the LHC. To achieve this aim Atlas must be sensitive to a large spectrum of physics signatures. This variety of signatures allows one to perform internal cross-checks and is necessary to perform robust and redundant physics measurements in the harsh operating environment of the LHC. The overall dimensions of the detector will be 22m $\times$ 22m $\times$ 42m, the total weight will surpass 7000 tons and there will be over 1 million readout channels.

A wide range of physics studies will be conducted at the Atlas experiment [9, Volume II]. One of the primary objectives of these studies is to understand the mechanism that leads to electro-weak symmetry breaking. There is thus a major focus on the discovery of the Higgs boson and Atlas is required to be sensitive to it over the full range of allowed masses. Other important components of the Atlas physics programme are the searches for other phenomena possibly associated to the electro-weak symmetry breaking. This includes particles predicted by supersymmetry or

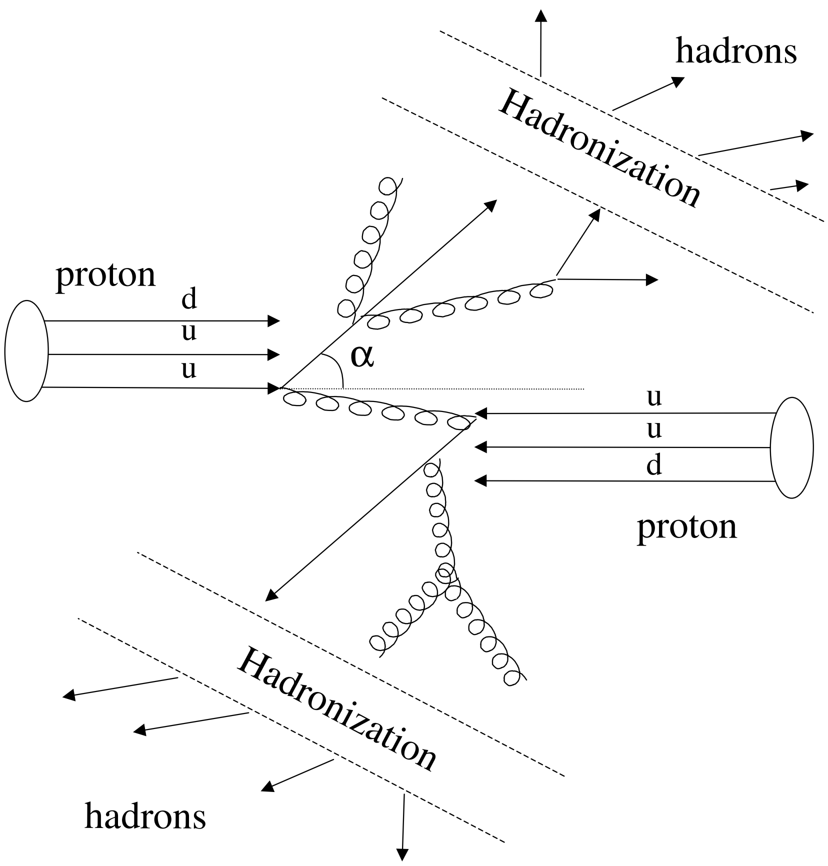


Figure 1.1: *Schematic illustration of a proton-proton collision in which jets are produced.*

technicolour theories, as well as new gauge bosons and evidence for composite quarks and leptons. The investigation of CP violation in B-meson decays and precision measurements of the W and top quark masses and triple gauge boson couplings will also play a large role in the physics studies.

In order to meet these physics goals an excellent detector performance is required. For example, high resolution measurements of electrons, photons and muons, excellent secondary vertex detection for τ leptons and b-quarks and high-resolution calorimetry for jets and missing transverse energy are essential to explore the full range of possible Higgs boson masses [9].

In this thesis we will focus our attention on the measurement of jets in the Atlas experiment. In general, it is very difficult to give a precise definition of a jet. Roughly speaking, a jet is a spatially localized collection of colourless hadrons that have approximately collinear momenta and are all fragments of the same parton. It is however far more illustrative to describe how jets arise. In hard qq scattering processes like $uu \to uu$, which arise in pp collisions, the two u quarks may scatter by exchange of a gluon. If the scattering angle α is large enough, the constituents of the two original protons separate and the protons are destroyed. The dynamical behaviour of systems of quarks is described by QCD, which is a $SU(3)$ non-abelian gauge theory. The massless gauge bosons of the perfect $SU(3)$ symmetry are called gluons and the strength of the quark-gluon interactions, also called the *colour force,* is given by a single coupling constant α_s. It has been shown both experimentally and theoretically that the coupling constant α_s depends on the momentum transfer q^2. In particular, for large momenta transfer α_s tends to zero, while for small momenta transfer the coupling constant becomes infinite. Since small distances correspond with large momenta transfer, the quarks that remain after the collision will behave as free particles for a short time. However, since the colour force increases with separation, these free quark states soon undergo the process of hadronization. Quark anti-quark pairs pop up from the vacuum and combine with the free quarks to form colourless hadrons, which do not experience the colour force. Extra hadrons also arise due to soft (that is, low relative transverse momentum) quark and gluon radiation. See Figure 1.1 for a further illustration. This process is not well understood theoretically, since it occurs in the non-perturbative region of QCD where $\alpha_s \sim 1$. There are however a number of phenomenological models which are used to describe the hadronization process, such as the Lund string model [4].

In practice, a jet is defined in terms of a *jet-finding algorithm,* which groups particles together based on criteria that correspond to our intuitive notion of a jet. The precise form of these criteria may differ from analysis to analysis and depend closely on the physics channels under investigation and the specific requirements of the study at hand. In Chapter 3 we give a detailed discussion concerning these issues.

In simple terms, the goal of the jet measurement process is to extract kinematical information about the partons produced in the hard scattering event from the energy deposits that are observed in the detector after the hadronization phase. This procedure is part of the total event reconstruction process, which we shall discuss in the sequel.

As mentioned above, the ability to perform precise jet measurements is crucial to the success of the Atlas physics programme. To give an example, we consider the role jets play in the search for the Higgs boson. In Figure 1.2 the significances of the various decay channels for the Higgs boson have been plotted as a function of the unknown Higgs mass. Jets are formed in the decay channels $H \to \gamma\gamma + WH, ttH(H \to \gamma\gamma)$, $H \to \gamma\gamma + WH, ttH(H \to b\bar{b})$ and $H \to WW \to l\nu jj$. Here l stands for an arbitrary lepton and ν for the corresponding neutrino. The first two channels are dominant for low values of the Higgs boson mass m_H, while the last channel is dominant at high m_H. A thorough understanding of the behaviour of jets in the detector will thus be needed if the actual Higgs mass lies in one of these regions.

Other important channels involving jets are $W \to jj$, $Z \to b\bar{b}$ and $t \to bW$, which will be used in the study of resonances. In addition, jet multiplicity and total jet energy measurements

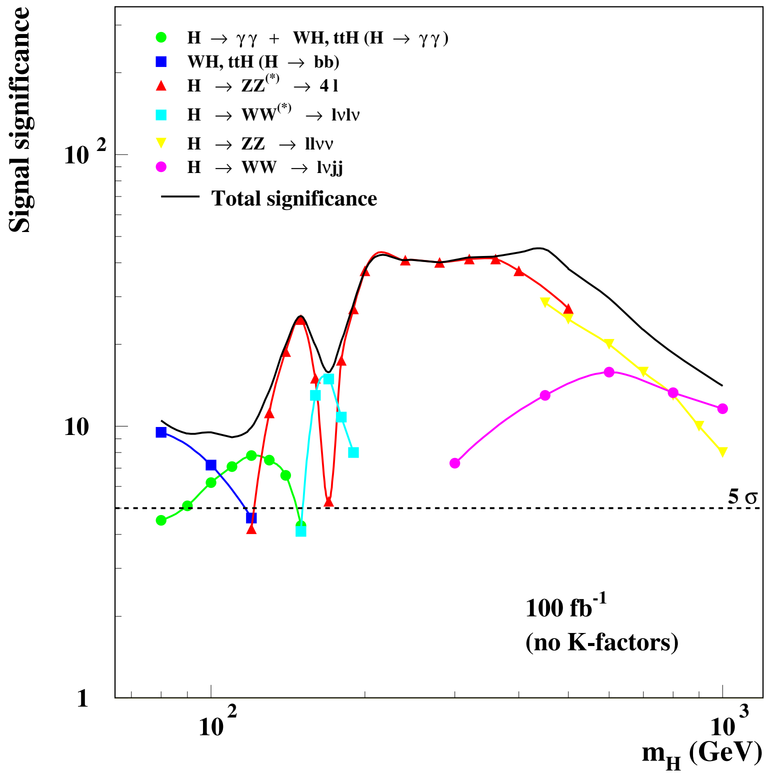


Figure 1.2: *Decay channels for the Higgs boson together with their significances for different values of the Higgs mass m_H. The 5σ line represents the minimum signal to background ratio for which a clear signal can be claimed. This figure was taken from [9].*

will play an important role in the search for supersymmetry [9, Chapter 20]. If the Atlas physics targets are to be met, jets will have to be reconstructed with an accuracy of 1% and a resolution of $50\%/\sqrt{E/[\mathrm{GeV}]} \oplus 3\%$ in the central barrel region, in a sense which will be made precise in the sequel.

At the moment, a large effort is underway to provide the software tools that will be required for the operation of the Atlas detector. These tools can be divided into two categories: detector simulation and event reconstruction [6]. The task of the event reconstruction software is to extract the structure and kinematical properties of the physics events from the raw data produced by all the detector readout channels. Since the amount of data that will be produced by Atlas each year is phenomenal, it is essential to be able to perform this task efficiently and easily. The detector simulation software allows one to study in detail the response of the Atlas detector to a wide range of physics events. It can be split into two distinct parts: simulation of physics events and simulation of the detector response to these events. Currently, a stand alone Monte Carlo simulation program is used to generate proton-proton collision events. A hadronization model then simulates the transition from final state partons to hadrons, after which the response of the Atlas detector to the final particles can be calculated. There are in fact two different programs available for this last phase, which differ in the quality of the prediction of the detector response. The fast simulation software gives reasonable results in a short time, while the full reconstruction algorithms take considerably more CPU time but increase the quality of the predictions. These full reconstruction algorithms are currently bundled in the Athena [13] package. The algorithms involved in the event reconstruction can also be run on simulated data, which means the performance of these algorithms can be tested and tuned by comparing the reconstructed event with the simulated event. In particular, the algorithms involved in the jet reconstruction phase have been tuned by comparing the reconstructed jet energies with the energies of the simulated partons. To verify the validity of the Monte Carlo simulations, their predictions should be compared with real physics data. Small subcomponents of the Atlas detector have therefore been tested in testbeams and these results are begin used to increase the accuracy of the detector simulation software.

The theoretical precision of the existing Monte Carlo generators is however far from adequate for the challenging requirements of the LHC experiments. Despite the huge efforts that have been put into the development of physics generators for hadron colliders over the last years, the precision with which present data can be reproduced is not better than 10-30% and in some cases not better than a factor of two. Furthermore, the testbeam data is not good enough to tune the jet measurements to the desired degree of accuracy [9]. Therefore it is extremely important that the jet reconstruction algorithms can be calibrated using real physics events, which will be available after the startup of the LHC. This presents an interesting challenge, since in this case the simulation data is of course no longer available.

In this thesis we study the performance of the existing jet reconstruction algorithms on a sample of simulated Z+jet events. We also study the feasibility of using such a sample of physics events to calibrate the algorithms during the initial low luminosity run of the LHC. In addition, we discuss a method to improve the efficiency of the jet algorithms currently in use, which will lead to a considerable reduction of the computer time needed to reconstruct physics events. This work can partly be seen as a continuation of the research started in [29], where a similar analysis on a sample of Z+jet events was conducted. However, the sample used in [29] was simulated using the fast detector simulation software, while here the full detector simulation program was used. In addition, the jet reconstruction algorithms have been restructured and improved considerably in the mean time.

This work is organized as follows. In Chapter 2, we briefly introduce the Atlas detector, define the coordinate system we shall use and discuss the different detector components on which the jet measurement process chiefly depends. Chapter 3 deals with a number of issues connected to jet reconstruction algorithms. We study the theoretical requirements these algorithms should satisfy and outline the implementation of the $k_\perp$ algorithm. In addition, we propose a number of modifications that will considerably increase the efficiency of this algorithm. In the first part of Chapter 4, we present the simulated dataset, discuss the cuts that were applied and investigate the performance of the current software. In the second part of this chapter the simulated sample is used to calibrate the jet algorithms using measurable quantities only. The performance of the resulting reconstruction scheme is then compared to that of the scheme currently in use. Finally, in Chapter 5 we give a summary of the obtained results and outline several directions for future research.

Chapter 2

The Atlas Detector

In this chapter we give a brief overview of the structure of the Atlas detector and introduce the notation we shall use throughout the rest of this work. Since many of the issues involved in jet reconstruction are very detector-specific, we will focus specifically on those components of the detector that are relevant for this process.

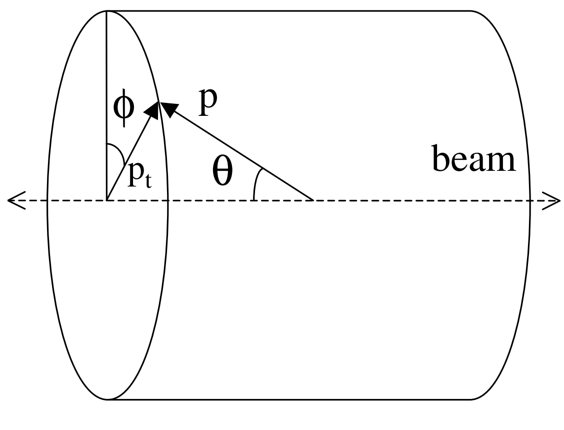


Figure 2.1: *Schematic overview of the coordinate system of the Atlas detector, which provides almost complete* 4π *angle coverage.*

The overall layout of Atlas is shown in Figure 2.2. The large number of subsystems, which each have their own specific sensitivities, can roughly be divided into three main components: the inner detector, the calorimeters and the muon system. The inner detector is maintained in a solenoidal magnetic field of $2T$ and provides high quality pattern recognition, momentum and vertex measurements and electron identification. It is surrounded by the various calorimeters, which provide an excellent spatial coverage for energy measurements on electromagnetic and hadronic particles. The outermost component is the muon system, which surrounds the calorimeters and provides precise energy and momentum measurements for muons. The magnetic fields required in the detector are provided by two systems. The inner detector cavity is surrounded by a thin superconducting solenoid magnet, while large superconducting air-core toroids mounted outside the calorimeters provide the magnetic field for the hadronic calorimeters and the muon system.

In Figure 2.1 the coordinate system we shall use is defined. The beam direction defines the z-axis, the azimuthal angle ϕ is measured in the plane transverse to the z-axis and the polar angle θ is the angle from the beam axis. The pseudorapidity is defined as $\eta = -\ln\tan\frac{\theta}{2}$ and the projection of a momentum vector $\vec{p}$ onto the plane transverse to the z-axis will be denoted by $\vec{p}_t$. Note that the pseudorapidities of massless particles behave extremely well under boosts in the z-direction; they are simply shifted by a constant. Since the true center-of-mass system in hadronic collisions often has a nonvanishing momentum along the z-axis with respect to the laboratory system, pseudorapidity is a very convenient quantity to use in hadron-hadron colliders.

The sections of the Atlas detector that are the most important for jet reconstruction are the calorimeters. Figure 2.3 gives a graphical view of the different calorimeter regions in the detector. There are four main components, namely the electromagnetic (EM), hadronic barrel, hadronic end-cap and forward calorimeters. The different components use different technologies to perform the energy measurements, based upon the performance required in the respective regions of the detector and the various operating conditions. The EM calorimeter is a lead/liquid-argon (LAr) based detector with accordion geometry [7]. It is divided into three parts: a barrel part enclosed by two endcaps, which together cover the pseudorapidity range $|\eta| \leq 3.2$. The barrel part spans the $|\eta| \leq 1.475$ range. It consists of two identical half barrels separated by a small gap at $\eta = 0$. These half barrels can be further divided into three regions, each with a distinct $\Delta\eta \times \Delta\phi$ granularity, ranging from 0.003×0.1 to 0.05×0.025. Each EM end-cap calorimeter consists of two coaxial wheels: an outer wheel covering the region $1.375 \leq |\eta| \leq 2.5$ and an inner wheel covering the $2.5 \leq \eta \leq 3.2$ region. The granularity in the end-caps differs between seven possible values, ranging from 0.003×0.1 to 0.05×0.025.

The hadronic barrel calorimeter, which spans the $|\eta| \leq 1.7$ range, is a cylinder divided into three sections: the central barrel and two identical extended barrels. It is based on a sampling technique with plastic scintillator plates (tiles) embedded in an iron absorber and has a granularity of 0.1×0.1 [8]. The remaining calorimeters are all based upon the intrinsically radiation-hard liquid-argon technology, since at higher pseudorapidities a higher radiation resistance is needed. The hadronic end-cap calorimeter is a copper LAr detector with parallel-plate geometry, which spans the pseudorapidity range $1.5 \leq |\eta| \leq 3.2$ and has two granularities: 0.1×0.1 and 0.2×0.2. Finally, the forward calorimeter spans the remaining range $3.1 \leq \eta \leq 4.9$ and has a granularity of

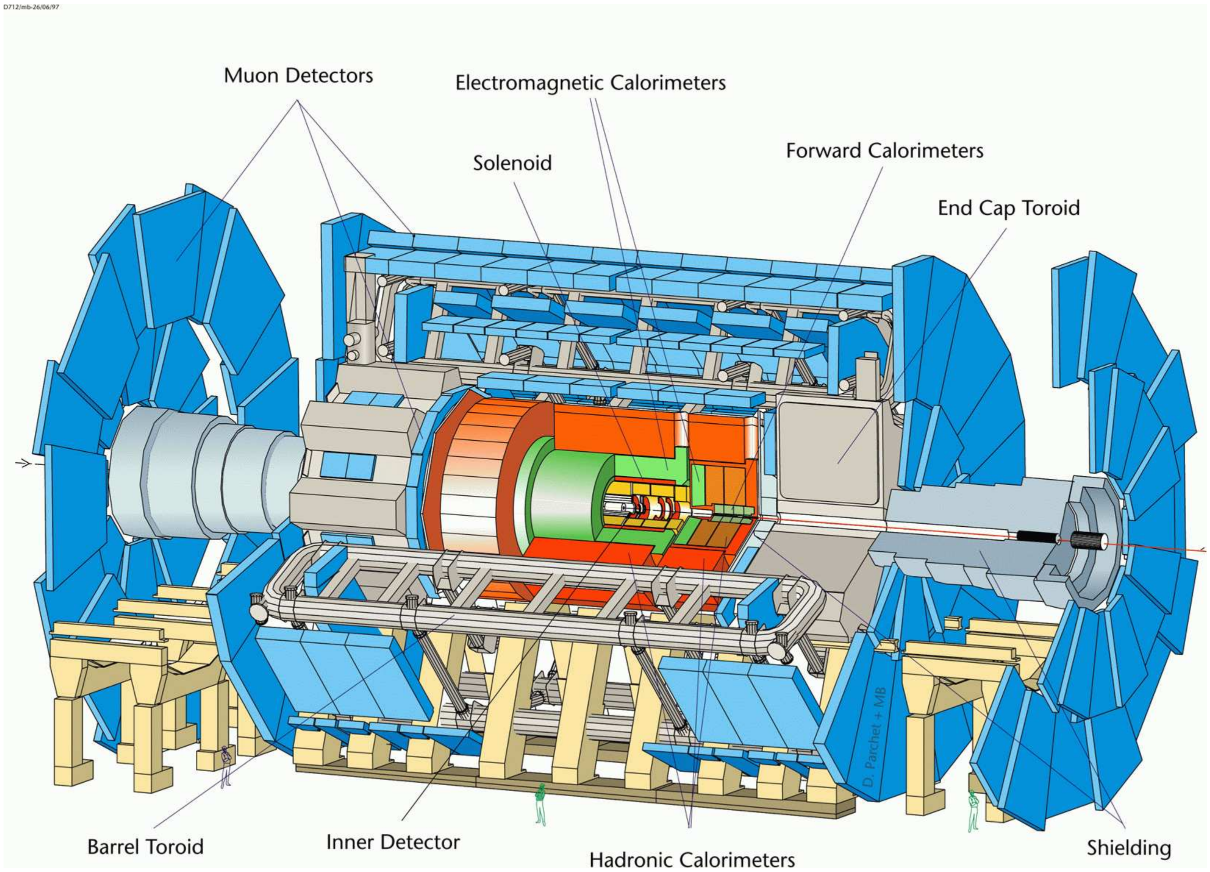


Figure 2.2: *Global view of the Atlas detector.*

approximately 0.2×0.2. All the liquid-argon calorimeters are housed within cryostats. The EM barrel calorimeter is located within the barrel cryostat, which also contains the inner detectors. The EM end-cap calorimeters are situated within two end cap cryostats, together with the hadronic end-cap and forward calorimeters.

There are a number of gaps in the spatial coverage of the various calorimeters to provide room for cables from the inner detector and service pipes for the cryostats and the electromagnetic calorimeter. Scintillating tiles are placed at various location in these gaps to sample the energy losses caused by the presence of these dead materials.

An extensive programme to calibrate and study the performance of the different calorimeters has been underway since 1996. The calorimeter responses to electrons, charged pions and muons have been studied by placing calorimeter modules in a testbeam of known composition. Comparisons between the responses to charged pions and electrons based on the various testbeam results have shown that the Atlas calorimeters are non-compensating. This means that the ratio e/h of the

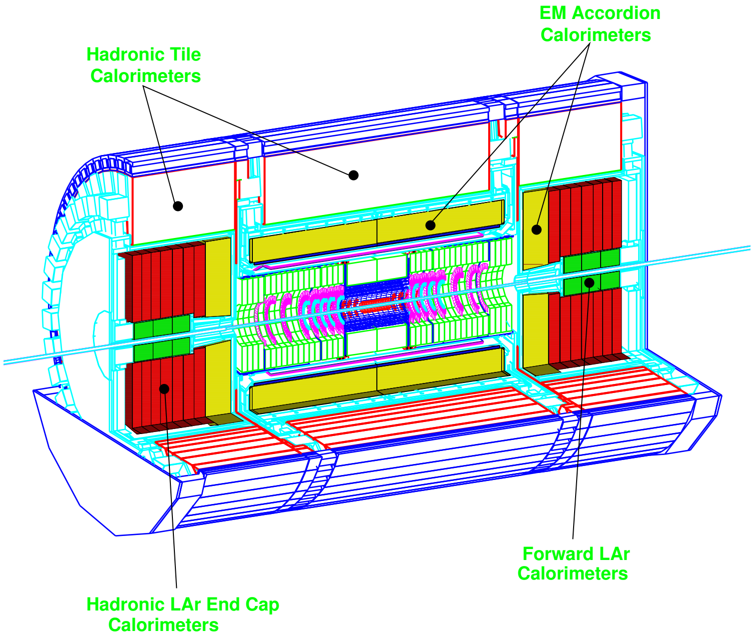


Figure 2.3: *Subview of the various calorimeters in the Atlas detector. These are the regions that are crucial for jet measurements.*

calorimeter response to the electromagnetic and purely hadronic component of the hadron showers differs from one. Typical values of the e/h ratio range from 1.2 to 1.4 [9, Chapter 5], indicating that electromagnetic particles induce a higher response than hadronic particles. The detectors will be calibrated at the electromagnetic scale, so corrections will have to be applied if one is interested in hadronic energy deposits.

Chapter 3

Jet Measurements in Atlas

The primary goal of jet reconstruction is to help extract the structure of the underlying hard scattering event as precisely as possible from all the energy deposits in the detector caused by hadronic showers. At first order in the strong coupling constant α_s, each jet is caused by the showering of one final state parton and thus in this regime jet reconstruction can be seen as the process of extracting the kinematical properties of these final state partons from the observed energy deposits in the detector. This problem can be split into two distinct phases. The first phase consists of identifying the regions of hadronic activity in the detector and deciding how to group these regions into distinct jets. After this so-called jet-finding phase, the kinematical properties of each jet, such as transverse momentum p_t, pseudo-rapidity η and azimuthal angle ϕ, have to be defined in such a way to best reflect the corresponding properties of the originating parton(s). We will refer to this second step as the energy calibration process.

In the study of any physical theory it is of course very important to be able to relate experimental data to theoretical predictions. In the case of jet physics, this amounts to the ability to calculate cross-sections for the production of jets in the detector. Since the precise definition of a jet is determined by the jet-finding algorithm used in the jet-finding phase, it is crucial to define the algorithm in such a way, that the resulting jet definition can be easily handled both experimentally and theoretically. In Section 3.1 we discuss the theoretical requirements that a jet definition must fulfill in order for it to be useful.

A number of different jet-finding algorithms have been proposed in the literature and many are still in use. As examples we mention the `JADE` [27], `ConeJet` [36], `SeededConeJet` and `MGS` [33] algorithms. In this paper we focus our attention on the $K_\perp$ jet-finding algorithm, also known as the `Durham` algorithm. It was first proposed and used in the framework of e^+e^- collisions [14, 17] and was found to have many phenomenological and theoretical advantages over the other jet-finding algorithms that were in use at that time. However, in [18] a generalization of the algorithm to deep inelastic lepton-hadron scattering (DIS) and hadron-hadron collisions was proposed. Detailed comparisons have been performed [19, 23] between the traditional `Jade` and `Cone` algorithms and the new `Durham` algorithm for hadron-hadron collisions, which yielded promising results for the latter. In Section 3.2 we introduce this algorithm and discuss its properties based on the requirements listed in Section 3.1. We also propose a new implementation, which is expected to be considerably faster than the form in which it is currently used. Finally, the jet energy calibration problem is treated in Section 3.3.

3.1 Theoretical Aspects of Jet Definitions

As we have seen above, it is imperative to employ a jet definition for which theoretical predictions of the production cross-sections can be made. In this section we review some of the issues involved in calculating production cross-sections and the implications these issues have on the formulation of a useful jet definition.

Although the process of hadronization cannot be described in terms of perturbative QCD, a number of powerful tools are available which allow relatively accurate predictions of cross-sections. One such tool is the hypothesis of Local Parton Hadron Duality (LPHD), which constructs a bridge between the world of partons and the world of final state hadrons. The hypothesis states that single particle inclusive spectra are simply required to be proportional to single parton inclusive spectra obtained from perturbative QCD, provided the parton cascade is evolved down to a cutoff scale of the order of the hadronic masses $Q_0 \sim m_h$ [3]. The LPHD hypothesis has been particularly successful in describing both the shape and the position of the maximum of single inclusive charged particle momentum spectra in hadronic jets resulting from e^+e^- annihilations ranging from PETRA/PEP [38] to LEP energies [1, 21, 28, 32]. Similar results have been obtained in DIS for inclusive momentum spectra in the Breit frame at HERA [26, 39].

It is essential that jet cross-sections are finite order by order. When performing calculations of such cross-sections, one commonly utilizes the LPHD hypothesis and studies the corresponding cross-sections at the parton level. However, some Feynman graphs encountered in this process give rise to infrared divergencies when they are evaluated. These divergences come from configurations in which a parton emits a soft $q \to 0$ gluon, an outgoing parton divides into two collinear partons or an incoming parton emits another parton that carries away a fraction of its longitudinal momentum but no transverse momentum. The probability for one of these configurations to occur is infrared sensitive and infinite in fixed order perturbation theory. It is possible to get rid of these divergencies by performing a resummation procedure. Intuitively, this can be explained by arguing that unitarity requires the sum of the probabilities for each of the above configurations to happen or not to happen to be equal to one. One can therefore reasonably assume the relation between parton cross-sections and jet cross-sections to be stable under the resummation mentioned above if the measured jet variables do not change when a parton with small relative transverse momentum is emitted or when a parton divides into collinear partons. We will refer to this property as infrared and collinear safety.

In the case of collisions involving initial state hadrons there are new features that are not present in leptonic collisions. In particular, uncertainties in the longitudinal momenta of the partons contained in the hadrons lead to initial state collinear singularities in QCD calculations at the parton level. It this therefore crucial to separate the hard scattering subprocess from the underlying soft event. This can be done by employing another powerful tool, the factorizability theorem of QCD. This theorem states that these singular contributions can be factorized out in a process independent way by associating them with hadronic structure functions [2]. In particular, the physical cross-section for the formation of jets above a p_t cutoff d_{cut} for center-of-mass energy S can be written as

$$\sigma_{ab}^{\mathrm{jets}}(S, d_{\mathrm{cut}}) = \sum_{ij} \int_0^1 dz_1 \int_0^1 dz_2 F_i^a(z_1, \mu_F) F_j^b(z_2, \mu_F) \hat{\sigma}_{ij}(z_1\sqrt{S}, z_2\sqrt{S}, d_{\mathrm{cut}}, \mu_F, \mu_R, \alpha_s(\mu_R)). \tag{3.1}$$

Here $\hat{\sigma}_{ij}$ denotes the partonic cross-section for the formation of hard jets when colliding partons i and j, μ_F and μ_R are arbitrary renormalization scales and F_a^i and F_b^j define the parton distributions of the incoming hadrons a and b. These distributions functions are universal and can be measured in inclusive processes such as DIS or in Drell-Yan pair production [9, Section 15.7]. The partonic cross-sections can be calculated by invoking the LPHD principle, as discussed above. If one employs a jet definition that references only the kinematical properties of the hard-scattering subprocess, one can split the cross-section into its multi-jet constituents and write [19]:

$$\begin{aligned}\sigma_{ab}^n(S, d_{\mathrm{cut}}, y_{\mathrm{cut}}, \{p_j\}) &= \\ &\textstyle\sum_{ij} \int_0^1 dz_1 \int_0^1 dz_2 F_i^a(z_1, \mu_F) F_j^b(z_2, \mu_F) \hat{\sigma}_{ij}^n(z_1\sqrt{S}, z_2\sqrt{S}, d_{\mathrm{cut}}, \mu_F, \mu_R, \alpha_s(\mu_R), y_{\mathrm{cut}}, \{p_j\}).\end{aligned} \tag{3.2}$$

The parameter y_{cut} is part of the jet definition and defines the fragmentation scale, as we shall see later. In the case that jets are defined using only longitudinal-boost invariant quantities, the partonic cross-section no longer depends on the momenta of the incoming partons but only on the subprocess center-of-mass energy $z_1 z_2 S$, hence (3.2) simplifies even further to

$$\begin{aligned}\sigma_{ab}^n(S, d_{\mathrm{cut}}, y_{\mathrm{cut}}, \{p_j\}) &= \\ &\textstyle\sum_{ij} \int_0^1 dz_1 \int_0^1 dz_2 F_i^a(z_1, \mu_F) F_j^b(z_2, \mu_F) \hat{\sigma}_{ij}^n(z_1 z_2 S, d_{\mathrm{cut}}, \mu_F, \mu_R, \alpha_s(\mu_R), y_{\mathrm{cut}}, \{p_j\}).\end{aligned} \tag{3.3}$$

There are several advantages of using a longitudinal-boost-invariant algorithm of the type leading to equation (3.3). For instance, in this regime the Born cross-section for two hard final-state jets simply corresponds to the usual leading-order two-jet prediction with $d_{\mathrm{cut}} = p_{t\,\mathrm{min}}^2$. Furthermore, the tools exist for a full next-to-leading-order calculation [22, 25], after minor modifications to convert from a cone to a cluster algorithm. In addition, the resummation of large logarithms in the region of small d_{cut} and/or y_{cut} along the lines of [17, 18] remains possible.

As a final and rather obvious requirement which is also specific to hadronic collision experiments, we insist that the reconstructed jets are not strongly affected by contamination from initial state hadron remnants and the underlying soft event.

3.2 The $K_\perp$ Jet-Finding Algorithm

In this section we introduce the $K_\perp$ or `Durham` jet-finding algorithm. However, before focusing on this specific algorithm, we will discuss some notions that are common to all jet-finding algorithms.

For convenience, we introduce the concept of a *pseudo-particle* as an abstract entity having the kinematical properties p_t, η and ϕ and keeping a list of associated pseudo-particles. A jet-finding algorithm takes as input a set S of pseudo-particles and outputs an integer n_{jets}, together with a mapping $\mu : S \rightarrow \{1 \ldots n_{\text{jets}}\}$, which associates each pseudo-particle to a jet. Notice that the concept of a pseudo-article allows us to state a jet-finding algorithm in such a way that it can be applied to different types of inputs. For example, one can regard a set of final state hadrons produced by a MC simulation as a set of pseudo-particles and the same can be done for a set of energy-deposits in a calorimeter, which results from the reconstruction of a physics event.

An essential operation in jet algorithms is the combination of two pseudo-particles into a new one. A number of schemes exists to define the kinematical properties of the merged pseudo-particle. The most obvious scheme is to simply add the 4-momenta of the two particles. The disadvantage of this method is that the combination of two massless pseudo-particles will in general no longer be massless, hence the pseudo-rapidity η will no longer be a boost-invariant quantity. If one wants to preserve the advantages of using a boost-invariant algorithm, considerable complications arise. Currently, the Snowmass recombination scheme is used in the Atlas software. A p_t weighted average is performed on the coordinates ϕ and η, as illustrated in Algorithm `Combine`.

Algorithm `Combine`**(x, y)**
Input: Two pseudo-particles x and y.
Output: The pseudo-particle z that arises after the merging of x and y.
1. z`.associated` $\leftarrow x$`.associated` $\cup$ y`.associated`

(∗ Calculate new kinematical properties, based on Snowmass recombination scheme. ∗)

2. $p_t(z) \leftarrow p_t(x) + p_t(y)$
3. $\eta_z \leftarrow \frac{p_t(x)\eta_x + p_t(y)\eta_y}{p_t(x)+p_t(y)}$
4. $\phi_z \leftarrow \frac{p_t(x)\phi_x + p_t(y)\phi_y}{p_t(x)+p_t(y)}$

5. **return** z

The monotonic p_t^2-weighted scheme is defined somewhat differently. Instead of specifying new η and ϕ values for the merged pseudo-particle, we prescribe how to calculate distances R_{ij} between pseudo-particles recursively. Since all the algorithms given in the sequel only reference the inter-particle distances R_{ij} and not the individual η and ϕ coordinates, jet-finding using this monotonic scheme is well defined. For the Snowmass scheme one simply uses $R_{ij}^2 = (\phi_i - \phi_j)^2 + (\eta_i - \eta_j)^2$. The recursive relations that define the monotonic scheme are given in (3.4).

$$\begin{aligned} p_t\big((ij)\big) &= p_t(i) + p_t(j), \\ R_{(ij)k}^2 &= \frac{p_t^2(i)R_{ik}^2 + p_t^2(j)R_{jk}^2}{p_t^2(i)+p_t^2(j)}. \end{aligned} \tag{3.4}$$

This monotonic recombination scheme leads to some interesting properties, which we shall investigate later on. For the moment we only remark that the inequality $R_{(ij)k}^2 \geq \min\left(R_{ik}^2, R_{jk}^2\right)$ holds, which can be easily verified.

The $k_\perp$ or Durham jet-finding algorithm as proposed in [18] can be split into two phases: a selection phase and a clustering phase. In addition to regular jets, the $k_\perp$ algorithm defines a so-called beam jet, to which all pseudo-particles that did not participate in the hard scattering process should be assigned. The goal of the selection phase is to filter out all such pseudo-particles, after which the clustering phase groups the energy deposits that are left into a number of spatially non-overlapping jets.

The selection phase takes as input a set of pseudo-particles S and produces a subset $U \subset S$ of pseudo-particles which participated in the hard scattering process. The procedure attempts to identify low p_t scattering fragments and removes them from the full set S when found. Please note that pseudo-particles with small p_t are not necessarily removed; they may be considered to be part of a scattering fragment with a sufficiently large average p_t to be kept as a hard fragment. This property allows for the correct identification of hard final state jets that suffer soft radiation near the beam axis. The parameter d_{cut} is used in a stopping condition and defines the hard scale of the method. The general structure of this phase is outlined in Algorithm `HardPTSelection`.

Algorithm `HardPTSelection`**(**S**,** d_{cut}**)**
Input: A set S of pseudo-particles and a stopping parameter d_{cut}
Output: A set U of pseudo-particles
(∗ Initialize output set U and lists of associated particles. ∗)
1. $U \leftarrow S$
2. **for** all $i \in S$
3. **do** i`.associated` $\leftarrow \{i\}$
4. **repeat**

(∗ Calculate distances d_{iB} and d_{ij} and find the minimum. ∗)
$d_{\min} \leftarrow \infty$
for all $i \in S$
  do $d_{iB} \leftarrow p_t^2(i)$
    if $d_{iB} < d_{\min}$
      then `minBeam` ←**true**
        `minParticle` ←i
        $d_{\min} \leftarrow d_{iB}$
for all pairs $(i, j) \in S \times S$ with $i \neq j$
  do $d_{ij} \leftarrow \min\left(p_t^2(i), p_t^2(j)\right) R_{ij}^2$
    if $d_{ij} < d_{\min}$
      then `minBeam` ←**false**
        `minPair` ←(i, j)
        $d_{\min} \leftarrow d_{ij}$
(∗ If minimum is d_{iB}, i should be assigned to beam jet. Otherwise merge particles. ∗)
if $d_{\min} < d_{\mathrm{cut}}$ **then**
    if minBeam
      then $S \leftarrow S - \{$`minParticle`$\}$
        $U \leftarrow U -$ `minParticle.associated`
      else $(x, y) \leftarrow$`minPair`
        $z \leftarrow$`Combine`$(x,\ y)$
        $S \leftarrow S \cup \{z\} - \{x, y\}$
until $S = \emptyset$ **or** $d_{\min} \geq d_{\mathrm{cut}}$
return U

Notice that two separate notions of distance were introduced in `HardPTSelection`. For every pseudo-particle i, the quantity $d_{iB} = p_t(i)^2$ measures the distance of i to the beam axis. If it is too close, i will be associated with the beam jet. For every pair of pseudo-particles d_{ij} measures the relative separation. Nearby particles are merged to form combined particles. The role of these two distances will be further discussed at the end of this section.

The clustering phase divides the pseudo-particles attributed to hard fragments into final state jets. It also uses a cut off parameter d_{cut}, which this time defines the fragmentation scale of the algorithm. The pseudo-code for this phase is given in Algorithm `FindHardJets`.

Algorithm `FindHardJets`**(**S**,** d_{cut}**)**
Input: A set S of pseudo-particles and a stopping parameter d_{cut}.
Output: A set J of pseudo-particles, the final hard jets. For each jet j in J, the parameter

j.`associated` lists the pseudo-particles from S that the jet contains.

(∗ Initialize the particles in S and initialize J ∗)

for all $i \in S$
    do i.`associated` $\leftarrow \{i\}$
$J \leftarrow S$
repeat
    $d_{\min} \leftarrow \infty$
    for all pairs $(i, j) \in J \times J$ with $i \neq j$
        do $d_{ij} \leftarrow \min\big(p_t^2(i), p_t^2(j)\big) R_{ij}^2$
            if $d_{ij} < d_{\min}$
              then `minPair` $\leftarrow (i, j)$
                  $d_{\min} \leftarrow d_{ij}$
    if $d_{\min} < d_{\text{cut}}$
        then $(x, y) \leftarrow$ `minPair`
            $k \leftarrow$ `Combine`(x, y)
            $J \leftarrow J \cup \{k\} - \{x, y\}$
until $d_{\min} \geq d_{\text{cut}}$
return J

We are now ready to state the complete algorithm. It is defined using a hard scale d_{cut} and a fragmentation parameter y_{cut}, which satisfies $0 \leq y_{\text{cut}} \leq 1$. The fragmentation scale is defined to be $y_{\text{cut}} d_{\text{cut}}$.

Algorithm `DurhamFindJets`$(S, d_{\text{cut}}, y_{\text{cut}})$
Input: A set S of pseudo-particles, a stopping parameter d_{cut} and a fragmentation parameter y_{cut} satisfying $0 \leq y_{\text{cut}} \leq 1$.
Output: A set of jets J and a beam jet B.

1. `hardParticles` $\leftarrow$ `hardPTSelection`(S, d_{cut})
2. B.`associated` $\leftarrow S \setminus$ `hardParticles`
3. $J \leftarrow$ `FindHardJets`(`hardParticles`, $y_{\text{cut}} d_{\text{cut}}$)
4. `return`(J, B)

Let us consider the parameters d_{ij} and d_{iB} introduced in `HardPTSelection` in more detail. For small angles θ_i and $\theta_{ij} = \angle(i, j)$, we have

$$\begin{gathered} p_t^2(i) = E_i^2 \sin^2 \theta_i \simeq E_i^2 \theta_{iB}^2 \\ p_t^2(i)\big((\eta_i - \eta_j)^2 + (\phi_i - \phi_j)^2\big) \simeq E_i^2\big((\theta_i - \theta_j)^2 + (\phi_i - \phi_j)^2 \sin^2 \theta_i\big) \simeq E_i^2 \theta_{ij}^2. \end{gathered} \tag{3.5}$$

Thus for $\theta_{iB} \to 0$ and $\theta_{ij} \to 0$, the distances d_{ij} and d_{iB} for the Snowmass recombination scheme reduce to

$$d_{iB} \simeq k_{\perp iB}^2, \tag{3.6}$$

$$d_{ij} \simeq \min(E_i^2, E_j^2)\theta_{ij}^2 \simeq k_{\perp ij}^2, \tag{3.7}$$

where $k_{\perp ij}$ is the minimum relative transverse momentum of pseudo-particles i and j and $k_{\perp iB}$ is the transverse momentum of i relative to the beam. Equation (3.7) implies the infrared and collinear safety of the $k_\perp$ algorithm, since pairs of pseudo-particles with vanishing energies or relative angles will be combined. Equation (3.6), together with the fact that pseudo-particles allocated to the beam jet do not take part in the clustering scheme, ensures that the algorithm factorizes initial state collinear singularities correctly. The fact that the inter-particle distances depend only on the longitudinal boost invariant quantities p_t, η and ϕ ensures that the resulting jet definition is also invariant under longitudinal boosts. Similar observations can be made for the monotonic recombination scheme, ensuring that the `Durham` jet-finding algorithm as defined here satisfies the theoretical requirements discussed in Section 3.1.

Recall the inequality $R_{(ij)k}^2 \geq \min\left(R_{ik}^2, R_{jk}^2\right)$, which holds whenever the monotonic recombination scheme is used. Suppose that at some point during the execution of `HardPTSelection` line 25 is reached and that (x, y) is the pair for which d_{xy} is minimal amongst all calculated distances at that stage. It is then easy to see that $d_{(xy)B} = \left(p_t(x) + p_t(y)\right)^2 \geq \min\left(d_{xB}, d_{yB}\right) \geq d_{xy}$. Furthermore, for every other pseudo-particle k the inequality

$$d_{(xy)k} = \min\left(p_t(k)^2, (p_t(x) + p_t(y))^2\right) R_{(xy)k}^2 \geq \min\left(p_t^2(k), p_t^2(x)\right) R_{xk}^2 = d_{xk} \geq d_{xy} \tag{3.8}$$

holds. Applying these inequalities to `HardPTSelection`, we see that $d_{\min}$ does not decrease between subsequent iterations of the outer loop in line 4. This feature explains why this particular recombination scheme is called a monotonic recombination scheme. An interesting consequence of this fact is that the jet-finding algorithm is stable against small variations of the cut-off parameters d_{cut} and y_{cut}. While the number of jets returned by non-monotonic recombination schemes may vary wildly in both directions when the hard scale d_{cut} is increased, this number will only decrease in steps of one when a monotonic scheme is used. Here we ignore possible degeneracies in the selection of the minimum distance. Several other advantages of using a monotonic recombination scheme are discussed in [19], where numerical evidence is also given to support these claims.

As a final remark, we note that the `Durham` jet-finding algorithm is almost always preceded by a preclustering phase, in which nearby detector regions with energy deposits are merged together. This is done to reduce the number of input pseudo-particles to a level for which the algorithm has an acceptable running time. This is necessary because the `Durham` algorithm is quite slow at the moment when compared to other jet-finding algorithms.

Operation	Description
`add(key, value)`	Add the (`key`, `value`) pair to the queue.
`remove(key)`	Remove the pair with the specified key from the queue if present.
`extractMin()`	Return a (`key`, `value`) pair whose `value` field is minimal among all the pairs in the queue.

Table 3.1: Operations supported by a priority queue.

3.2.1 Implementation

At first glance, `HardPTSelection` and `FindHardJets` both seem to require $O(N^3)$ expected running time, where N is the size of the input set of pseudo-particles S. Since both algorithms have the same generic structure, we will focus our attention on `FindHardJets`. The loop defined in line 4 is executed at most N times, since in each iteration the number of elements in the set J is decreased by one. The selection of the minimum in the for loop at line 6 requires $O(N^2)$ iterations, hence the algorithm requires $O(N^3)$ running time. In the literature there appears to be consensus that this is indeed the fastest possible implementation of the $k_\perp$ algorithm. However, using the concept of priority queues, it is easy to provide an $O(N^2 \log N)$ implementation of the $k_\perp$ algorithm. If a monotonic recombination scheme is used, it is even possible to provide an implementation that has $O(N^2)$ expected running time.

A priority queue is an abstract list of (`key`, `value`) pairs that supports the operations listed in Table 3.1. Using a heap, it is possible [20] to provide an implementation of a priority queue for which all the operations listed in Table 3.1 require $O(\log N)$ running time, where N is the number of elements in the queue. A priority queue thus enables us to keep track of the minimum of a number of values efficiently. Noticing that one can restrict the calculation of the distances d_{ij} in the inner loop at line 6 of `FindHardJets` to pairs that have been affected during an execution of the outer loop defined in line 4, one can use a priority queue to improve the performance of this algorithm considerably. The modified procedure uses a priority queue Q and is described in `FindHardJetsFast`. Please note that the keys that will be used in the queue are pairs of pseudo-particles (i, j) with $i \neq j$. Throughout this section these keys are to be treated symmetrically, that is $(i, j) = (j, i)$.

Algorithm `FindHardJetsFast`**(**S**,** d_{cut}**)**

Input: A set S of pseudo-particles and a stopping parameter d_{cut}.

Output: A set J of pseudo-particles, the final hard jets. For each jet j in J, the parameter j.associated lists the pseudo-particles from S that the jet contains.

(∗ Initialize J and the particles in S. ∗)

1. **for** $i \in S$
2. **do** i`.associated` $\leftarrow \{i\}$
3. $J \leftarrow S$

4. (∗ Initialize the priority queue Q and calculate initial distances. ∗)
5. $Q \leftarrow \emptyset$
6. (∗ Loop over pairs in $S \times S$. We consider $(i, j) = (j, i)$ and treat each pair only once. ∗)
7. **for** all pairs $(i, j) \in S \times S$ with $i \neq j$
8. **do** $d_{ij} \leftarrow \min\left(p_t^2(i), p_t^2(j)\right) R_{ij}^2$
9. **if** $d_{ij} < d_{\text{cut}}$
10. **then** Q.`add`$\big((i, j), d_{ij}\big)$
11. **while** $Q \neq \emptyset$
12. $\big((x, y), d_{\min}\big) \leftarrow Q$.`extractMin`()
13. $k \leftarrow$`Combine`(x, y)
14. **for** all $i \in J \setminus \{x, y\}$
15. **do** Q.`remove`$((x, i))$
16. Q.`remove`$((y, i))$
17. $d_{ik} \leftarrow \min\left(p_t^2(i), p_t^2(k)\right) R_{ik}^2$
18. **if** $d_{ik} < d_{\text{cut}}$
19. **then** Q.`add`$\big((i, k), d_{ik}\big)$
20. $J \leftarrow J - \{x, y\} \cup \{k\}$
21. **return** J

It is easy to see that using the heap priority queue implementation, the running time has been reduced to $O(N^2 \log N)$, where N is the number of pseudo-particles in the input set S. Indeed, at all times the queue Q contains less than $\frac{1}{2}N^2$ entries, which means each priority queue operation costs at most $O(\log N^2) = O(\log N)$ time. Since the for loop in 14 now requires $O(N)$ priority queue operations and is executed at most N times, while the initial for loop in line 7 requires $O(N^2)$ priority queue insertions, the total running time is $O(N^2 \log N)$.

The running time can be further decreased by using a priority queue that has even faster performance than the heap based queue. The special structure of the algorithm when a monotonic recombination scheme is used can be exploited to use a relatively new data structure called a calendar queue, which was introduced in [15]. In the next sections, we introduce this data structure and discuss how it can be used to speed up `FindHardJetsFast`, for both monotonic and non-monotonic recombination schemes.

Calendar Queue

A calendar queue is an efficient implementation of a priority queue designed to give $O(1)$ expected performance for a class of enqueue / dequeue patterns. The data structure maintains M buckets which we shall number as 1 through M. Each bucket can store an arbitrary number of entries, for example in a doubly linked list. A parameter δ, called the *bucket-width,* controls the distribution of the entries over the buckets. In particular, bucket i contains only those entries whose priorities p satisfy $(i-1)\delta \leq p - p_{\min} \mod M\delta < i\delta$, where $p_{\min}$ is a constant associated to the calendar queue that represents the minimum possible priority. The queue keeps track of a current priority p_{cur}, defined to be the last priority returned during an `extractMin()` call. The calendar queue differs from ordinary priority queues in the fact that it only allows addition of entries that have priority p satisfying $p \geq p_{\text{cur}}$. The elements in a bucket are stored in an arbitrary order. This means that the operation `insert` runs in constant time, since the bucket into which an element should be inserted can be calculated directly and insertion into the front of a linked list also requires $O(1)$ time. The operation `remove` also requires constant time if the linked list element containing a key is stored along with the key, otherwise time proportional to the number of elements in a bucket is required. The operation `extractMin` proceeds by examining the bucket that would contain entries with priority p_{cur}. If this bucket is empty, the next bucket is repeatedly examined until a non-empty bucket is found. A simple linear search through the elements of the first non-empty bucket is then performed to find the entry with lowest priority, which is then returned and removed from the queue. The variable p_{cur} is then set to the priority of this last returned entry.

The $O(1)$ performance of the calendar queue relies crucially on the distribution of the entries over the bins. It is intuitively clear that the best performance occurs when each bin contains roughly the same small number of entries and the number of empty bins is small. This distribution depends on the bucket-width δ and on the pattern of dequeue / enqueue operations applied. If δ is very small, there will be a lot of empty bins, while if δ is too large, some buckets will have a lot of elements. Both conditions slow down the operation of the `extractmin` procedure and will destroy the $O(1)$ running time. The behaviour of the calendar queue under numerous operation conditions has been studied, both experimentally and theoretically. In these studies one executes a large number of iterations consisting of one `extractMin` call followed by a number n of insertion operations and counts the number of elementary steps needed by the queue. The priorities of the newly inserted entries p_{new} are randomly generated by drawing the quantities $p_{\text{new}} - p_{\text{cur}}$ from some nonnegative probability distribution f.

The case in which each `extractMin` call is followed by exactly one insertion operation has been analyzed theoretically in [24]. Note that in this case the number of elements N in the queue remains a constant between iterations and one can thus study the long term average behaviour of the queue. Under the assumption that the probability distribution function f satisfies the conditions

(A1) $f(x) > 0$ for all $x \in [0, \beta)$ where $\beta = \sup\{x \geq 0 \mid f(x) > 0\}$,

(A2) The mean μ is finite,

(A3) $\int_0^x f(s)ds \leq c_0 x$ for all $x \leq \epsilon_0$ for some $c_0, \epsilon_0 > 0$, the following theorem gives bounds for the performance of the static calendar queue.

Theorem 3.2.1. *Fix an integer $N_{\min}$ and let f be a positive probability distribution satisfying the conditions $(A1)$ through $(A3)$ above. Then there exists a constant $\gamma = \min\left(\frac{1}{2c_0}, \epsilon_0\right) N_{\min}$ such that for all $N \geq N_{\min}$ and all δ that satisfy $0 < \delta < \gamma/N$, the expected time $K_N(\delta)$ per dequeue / enqueue operation pair with probability distribution f in the infinite bucket ($M = \infty$) calendar queue with bucket width δ and N entries satisfies*

$$K_N(\delta) = C_1 + C_2 N \frac{\delta}{\mu} + C_3 \frac{\mu}{N\delta} + O(N^{-1}). \tag{3.9}$$

The constants C_1, C_2 and C_3 are implementation specific and do not depend on N,δ and μ. The $O(N^{-1})$ term can be bounded by $(\tau_2\gamma^2 + \tau_1\gamma)/N$ for some constants τ_1 and τ_2.

In the case of a non-static calendar queue where the number of entries N may vary, one can use the result in Theorem 3.2.1 to define optimal values for the parameter δ. In particular, if one enforces the bounds

$$\lambda\mu/N \leq \delta \leq \min(\mu, \gamma)/N, \tag{3.10}$$

for some $0 < \lambda < 1$ throughout the operation of the calendar queue, it is reasonable to assume that the `extractMin` operation will take $O(1)$ expected time. This assumption has been validated experimentally [34] for a wide range of probability distributions f, including uniform, exponential and gamma distributions. It is however a nontrivial issue to ensure that the bounds (3.10) remain valid. In particular, if the number of entries in the queue varies wildly, one will frequently have to vary the bucket width δ. This is a relatively expensive operation, as it requires time proportional to the number of buckets. The performance of the calendar queue thus degrades if one has to resize the calendar queue too often.

Application to monotonic $K_\perp$ clustering

Recall that in the monotonic recombination scheme the inequality $\min\left(d_{(ij)B}, d_{(ij)k}\right) \geq d_{ij}$ holds for all $k \neq i, j$, whenever i and j are about to be combined into a new pseudo-particle in `FindHardJets` and `HardPTSelection`. Applying this observation to `FindHardJetsFast`, it is easy to see that at each call to Q.`add`() in line 19, the priority of the inserted entry is higher than the priority of the last entry that was extracted from the queue in line 12. This fact allows us to use a calendar queue to manage the distances involved in the algorithm.

Since we are only interested in distances that satisfy $d_{ij} < d_{\text{cut}}$, the priority queue only needs to contain entries in the bounded priority range $[0, d_{\text{cut}}]$. Furthermore, the inequality $d_{(ij)k} \geq \min\left(d_{ik}, d_{jk}\right)$ implies that the number of entries stored in the priority queue does not increase between each `extractMin` call. Indeed, between lines 15 and 16 two elements are possibly deleted from the queue whilst at most one is added in line 19. The only way in which the number of stored entries can increase is if both remove operations have no effect and line 19 is executed. However, this can only be the case if $\min\left(d_{xi}, d_{yi}\right) > d_{\mathrm{cut}}$ and $d_{(xy)i} \leq d_{\mathrm{cut}}$, which is a contradiction. These observations simplify the operation of the calendar queue considerably, since it is now trivial to calculate an appropriate number of buckets M once the bucket width δ is known and the number of resize operations can be tightly bounded.

A priori, the function f describing the probability distribution for $p_{\mathrm{new}} - p_{\mathrm{cur}}$ is unknown. The mean μ of this distribution can be determined by experiment, but for the moment we shall assume a uniform distribution over the interval $[p_{\mathrm{cur}}, d_{\mathrm{cut}}]$, which leads to $\mu = (d_{\mathrm{cut}} - p_{\mathrm{cur}})/2$. This expression can be used in conjunction with (3.10) to determine the bucket-width δ. When using (3.10), it is very convenient to have $\gamma > d_{\mathrm{cut}}$ and hence $\gamma > \mu$. This can always be achieved by choosing N_{min} to be large enough, but for large values of N_{min} Theorem 3.2.1 loses its practical value. In case of a uniform distribution it suffices to have $N_{\mathrm{min}} = 2$, since in this case $\epsilon_0 \sim \infty$ and $c_0 = d_{\mathrm{cut}}^{-1}$. For other distributions the situation will not be too different, so it seems reasonable to assume that we can practically achieve $\gamma > d_{\mathrm{cut}}$.

Since the number of elements stored in the queue is non-increasing, one only needs to resize the queue if the number of stored entries drops below a minimum threshold. The new threshold is then defined to be half the previous threshold. This procedure is described in `Resize`, which is executed at the start of the `extractMin` procedure. For analysis purposes, we define the running time required by the `extractMin`, `insert` and `remove` operations excluding the time spent in `Resize` as the *bare* running time of the priority queue operations.

Algorithm `Resize`

Input: A calendar queue Q

(∗ Q.`size` denotes the number of stored entries in the queue, while Q.`minsize` is the threshold at which a resize operation must take place. ∗)
if Q.`size` $<$ Q.`minsize`
  then
    (∗ Calculate new bucket size Q.`delta`. The μ below is valid for a uniform distribution. ∗)
    (∗ In general, one should include boundary effects when calculating μ ∗)
    $\mu \leftarrow \left(Q.\mathtt{right} - Q.p_{\mathrm{cur}}\right)/2$
    Q.`delta` $\leftarrow \mu / Q.size$
    Q.`rebucket`()
    Q.`minsize` $\leftarrow Q$.`size` div 2

Lemma 3.2.2. *Consider Algorithm* `FindHardJetsFast` *and let f be the probability distribution for the quantities $d_{ik} - d_{\mathrm{min}}$ calculated in line 17. Suppose that f does not vary during the execution of* *the algorithm and that it satisfies the conditions* ($A1$) *through* ($A3$) *defined above. Suppose further that the expected time taken by the bare calendar queue operations is independent of the size* N *of the input set* S *provided the bounds (3.10) are enforced. Then the expected running time of* `FindHardJetsFast` *with a calendar priority queue* Q *is* $O(N^2)$.

Proof. The initial bucket width of the queue can be set to d_{cut}/N^2, together with $Q.\texttt{minsize} = \frac{1}{2}N^2$, which ensures the correct operation of the `Resize` function. By choosing $N_{\min}$ large enough, the inequality $\gamma > d_{\text{cut}}$ can be achieved. Since the time required to insert an element into the queue is constant, $O(N^2)$ elements can be inserted in $O(N^2)$ expected time. This means that the initialization of the queue in line 7 takes $O(N^2)$ time, as $O(N^2)$ distances have to be calculated and inserted. After each execution of the while loop in lines 11 to 20, the number of elements in Q has decreased. The updating of the distances in the queue costs $O(N)$ expected time for each iteration, since at most $2N$ distances are recalculated and each insert and removal operation is executed in $O(1)$ time. Each `extractMin` call also takes $O(1)$ time, which can be seen by noting that at the execution of `Resize` in the `extractMin()` call has guaranteed the bounds $\frac{1}{2}\mu/N \leq \delta \leq \mu/N$. Here μ is the expected mean value of the jumps $p_{\text{new}} - p_{\text{cur}}$ the last time the queue was rebucketed. It may be the case that between queue rebucketings the mean value of the jump will change due to boundary effects. However, this will not affect the performance of the calendar queue. This can be seen by comparing our calendar queue to a calendar queue that does not restrict its entries to the range $[0, d_{\text{cut}}]$, but has the same population as our calendar queue after the last rebucketing operation. The performance of our queue will not be slower than this new queue, but for this new queue the mean jump is not affected by the boundary and hence the bounds (3.10) still hold. The total time spent in resizing the queue can be bounded by noting that the rebucketing operations cost time proportional to the number of elements in the queue. Since the maximum number of elements in the queue is N^2 and the number of elements is halved between two resize operations, the total time spent is proportional to $T \sim N^2 + \frac{1}{2}N^2 + \frac{1}{4}N^2 + \ldots \leq 2N^2$. The time spent in recursively calculating the inter-particle distances R_{ij} can also be limited to $O(N^2)$ by keeping track of these distances at all times during the execution. When two pseudo particles i and j are combined, each new distance $R_{(ij)k}$ can be calculated in constant time using $R^2_{(ij)k} = \frac{p^2_{ti}R^2_{ik} + p^2_{tj}R^2_{jk}}{p^2_{ti} + p^2_{tj}}$. This completes the proof. $\square$

Lemma 3.2.2 shows that the expected running time of `FindHardJetsFast` with a calendar queue is $O(N^2)$, under some assumptions which we have seen to be quite reasonable. In particular, the conditions on the probability distribution f hold when f is a uniform distribution and the assumption that the bare calendar queue operations take $O(1)$ time can be justified by looking at a large number of calendar queue simulations [34, 37]. A lot of research has been done involving calendar queues and implementations have been proposed [37] that are stable against probability distributions that vary during the operation of the queue. These implementations attempt to identify optimal operating conditions for the queue on the fly. In particular, the `Resize` operation can be used to adjust the expected mean μ based on the previous series of entries. The factor of two between successive resize operations is completely arbitrary and can thus be modified without destroying the $O(N^2)$ expected running time. This allows one to optimize the number of recalculations of the expected mean μ.

Non-monotonic recombination

In the non-monotonic case some interesting and useful properties that allowed for the use of calendar queues are no longer valid. However, for "near-monotonic" recombination schemes, a slight modification of the calendar queue used in the previous section may be expected to give similar performance.

In order to accommodate for insertion of entries that have priorities $p < p_{\text{cur}}$, an additional bucket called the *back-bucket* will be kept in the calendar queue. This will store all entries for which the priorities p satisfy $p < p_{\text{cur}}$. In the `extractMin` call, the back-bucket will be examined first before the regular buckets. If this bucket is non-empty, the entry with minimal priority in the back-bucket will be returned. The $O(1)$ performance of the bare calendar queue operations will not be affected if the number of elements stored in the back-bucket is not significantly larger than the number of elements stored in the regular buckets. Thus if a recombination scheme is "near-monotonic", that is, if $P(p_{\text{new}} - p_{\text{cur}} < 0) \simeq \frac{1}{N}$, where N is the number of entries in the calendar queue, the algorithm using a calendar queue will still have a performance comparable to that of a monotonic algorithm. The degree of near-monotonicity will have to be examined experimentally to study the feasibility of this approach. An initial study in [19] already suggests that the standard Snowmass recombination scheme does not deviate too much from a monotonic scheme, which is a promising result.

3.3 Jet Energy Calibration

Once a jet has been found, it is still a non trivial task to assign the correct kinematical properties. One has to deal with a number of different effects that play a role in the chain from partons to reconstructed jets. These effects can be divided into two categories: physics effects and detector effects. We have already discussed various physics effects such as fragmentation and the presence of soft remnants from the hard scattering effect. In addition, since there will be approximately 23 events per bunchcrossing at the LHC, one will have to deal with the presence of additional soft events which take place simultaneously. Typical examples of detector effects include sudden jumps in granularity between detector components, electronics noise and pile-up, that is, the influence of the tails of the responses from previous bunchcrossings on the current detector response. Another such effect is caused by the magnetic fields present in the detector, which will curve the trajectory of charged particles and hence influence the showering. A very important aspect of jet energy calibration was already encountered in Chapter 2, where the ratio e/h was discussed. Since $e/h > 1$, the Atlas calorimeters respond differently to hadrons than to electromagnetic particles. The response even differs between charged and neutral hadrons [9, Chapter 9]. Since hadronic jets produce both electromagnetic and hadronic particles when they shower in the calorimeters, this fact has a great impact on the energy determination. Finally, due to the spatial width of jets, jet measurements are affected more strongly by the presence of dead materials in the detector and cracks between calorimeter components than single particle measurements.

The aim of the jet energy calibration phase is to combine the kinematical properties of the pseudo-particles assigned to a jet in a collinear and infrared safe way, whilst minimizing the impact of the detector effects mentioned above. Currently, this phase is split into two parts [10]. The first part concerns overall corrections such as the magnetic field correction, which compensates for the curved trajectories of certain shower components. The second part consists of a detailed cell by cell weighting procedure to reduce the effects caused by the dead regions in the detector, the cracks between calorimeters and the e/h ratio. This procedure is called H1-weighting and is discussed in detail in Section 3.4.

3.4 H1 Weighting

The H1-weighting procedure is applied after the magnet corrections have been performed and the jet-finding algorithm has divided the energy deposits in the calorimeter into jets. We start this section by describing the general procedure, after which the application to the Atlas detector is discussed.

In general, a detector is divided into R different regions V_r, $r = 1 \dots R$. For each detector region, n_r different energy bins are defined, which we shall denote as B_i^r for $1 \leq i \leq n_r$. For every jet we introduce the quantities E_i^r for $1 \leq r \leq R$ and $1 \leq i \leq n_r$, which denote the sum of energies in cells in region r, where only cells that have energy in bin B_i are counted. Formally, one has

$$E_i^r = \sum_{k \in V_r} \pi_i(E_k), \tag{3.11}$$

where E_k is the energy deposited in cell k, π_i is the projection function $\pi_i(E) = E$ if $E \in B_i$ and $\pi_i(E) = 0$ otherwise. A H1-weighting scheme consists of a series of weights $\{w_i^r\}_{i=1\dots n_r}^{r=1\dots R}$, which allows one to define the reconstructed jet energy

$$E^{\text{rec}} = \sum_{r}^{R} \sum_{i=1}^{n_r} w_i^r E_i^r. \tag{3.12}$$

To determine an optimal set of H1-weights one has to perform a calibration procedure. This is of course very detector dependent, since the energy deposits in the different cells will vary from detector to detector. Given a set S of jets for which both the true energy E^{true} and the energy deposits in the detector are known, one can determine a corresponding optimal set of H1-weights by minimizing the least squares residual

$$\mathcal{F}(\{w_i^r\}, S) = \sum_{j \in S} \Big(\frac{E^{\text{rec}}(j) - E^{\text{true}}(j)}{\sigma_{\text{jet}}(j)} \Big)^2, \tag{3.13}$$

in which σ_{jet} is the standard deviation of the jet measurement error.

In the case of the Atlas detector, we use eight different detector regions, that is, $R = 8$. The hadronic barrel calorimeter is treated as one region (Tile), as is the hadronic end-cap (HEC). The barrel electromagnetic calorimeter is divided into two regions (EMB2, EMB3) and the same holds for the end-cap electromagnetic calorimeter (EME2, EME3). This splitting is based upon the distance to the beam axis. For each of these regions ten different energy bins were used, that is $n_r = 10$. The remaining two regions are the cryostat and the gap. The energy lost in the cryostat is estimated based upon the geometric mean of the energy deposits in the last layer of the EM calorimeter and the first layer of the hadronic calorimeter, while the gap scintillators are used to estimate the energy lost in the gap. Only one energy bin is assigned to both these final regions. In this analysis, the forward calorimeter was not used in the H1-weighting procedure.

When minimizing the system (3.13), there is of course no way of knowing the standard deviation σ_{jet} a priori. In practice however, it has been observed that the model $\sigma_{\text{jet}}(E) = \frac{a\%}{\sqrt{E/[\text{GeV}]}} \oplus b\%$ provides a good fit for the jet measurement error, hence we shall use the Atlas jet reconstruction requirement $\sigma_{\text{jet}}(E) = \frac{50\%}{\sqrt{E/[\text{GeV}]}} \oplus 3\%$ in our calculations to arrive at

$$\mathcal{F}(\{w_i^r\}, S) = \sum_{j \in S} \Big(\frac{E^{\text{rec}}(j) - E^{\text{true}}(j)}{E^{\text{true}}(j)\sqrt{(0.03)^2 + \big(\frac{0.5}{\sqrt{E^{\text{true}}(j)/[\text{GeV}]}}\big)^2}} \Big)^2. \tag{3.14}$$

The hard part is of course to determine an appropriate set S of jets to perform such a calibration. The H1-weights that are now used in the Atlas software were obtained by performing this procedure on a dataset generated by a Monte-Carlo simulation. A large number of physics events in which jet production occurs were simulated, together with the detector response to these events. The $k_\perp$ jet-finding algorithm was applied to these simulated events to generate the sample of jets, together with their energy deposits. To define the true energies of these jets, the $k_\perp$ algorithm was applied on the particles from the MC truth information to produce so-called truthjets. The thus reconstructed energy of the nearest truthjet was defined to be the true energy of a jet.

The problem with this procedure is that in general the true energy of a jet is unknown. When calibrating ATLAS in the LHC with real physics events, the Monte Carlo truth information will no longer be available and another way will have to be found to supply the true jet energy. In this thesis one other possible jet calibration method is investigated, which is based upon the study of certain Z+jet events, in which the Z-particle decays into an electron-positron pair or a $\mu^+\mu^-$-pair, which is registered in the detector. Since leptons can be reconstructed with a relatively large degree of accuracy as compared to jets, the kinematical properties of the Z-particle can be accurately reproduced and used to calibrate the jet reconstruction algorithms. If the quark that causes the jet is produced back to back with the Z-particle, one can match the transverse momentum of the Z-particle with the transverse momentum of the jet. One can thus use the kinematical properties of the Z-particle to define the true jet energy required for the H1 calibration procedure.

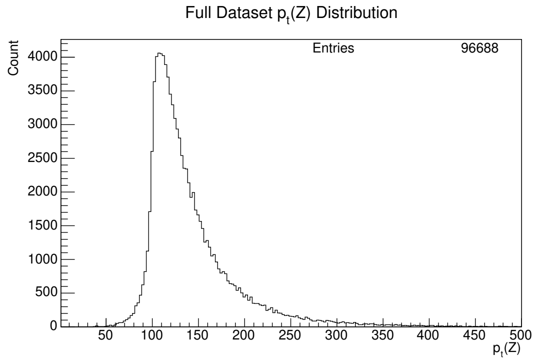
Chapter 4

Jet Reconstruction Performance

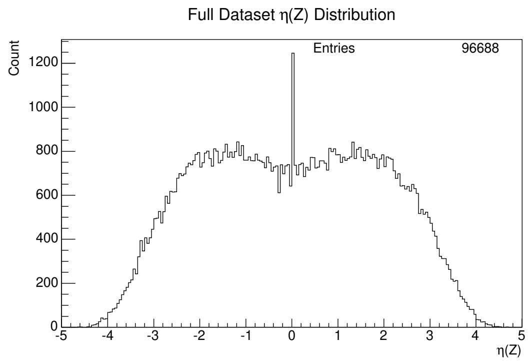
In this chapter we discuss our test of the existing jet reconstruction software based on a sample of simulated Z+jet events. The dataset under consideration was generated as part of the Atlas Data Challenge I project and contains 10^5 Z+jet events. A calculation of the cross-section at leading order for such events can be found in [12]. The transverse momenta of the produced Z particles range from approximately 50 GeV/c to 450 GeV/c, while the pseudorapidities of these particles cover the range $-5 \leq \eta \leq 5$, as illustrated in Figure 4.1. The integrated cross-section corresponding to the events in this sample is given by $\sigma = 3\,\mathrm{fb}^{-1}$. This means that after approximately 4 months of data taking at low luminosity a comparable sample of physics events will be available.

The events were generated with a Monte-Carlo simulation, using the `Phytia` and `Jetset` [35] code. These programs simulate hard scattering events together with initial and final state radiation and implement a hadronization scheme for the partons produced in the collision. In particular, this means that the kinematical properties of all the particles that participate in the event are known and available for use in our analysis. We will refer to these quantities as the MC truth information. To simulate the response of the Atlas detector to the particles produced in the event, the Geant package [16] was used. The Geant package is a toolkit for the simulation of the passage of particles through matter. The geometry of the Atlas detector has been modelled in great detail and this model is used by Geant to track each individual simulated particle through the detector. The energy losses in each individual part of Atlas are calculated together with the effect these losses will have on the raw readout channels. After this step, the dataset has the same format real physics events will have, with the exception that for simulated datasets the MC truth information is available. The presence of this information enables one to study the performance of the event reconstruction software by comparing the reconstructed event topology with the Monte-Carlo data.

We introduce the physics quantities used in our analysis in Section 4.1, after which a number of selection cuts that were applied to the sample are discussed in Section 4.2. In Section 4.3 we define the criteria that are traditionally used to measure the performance of the jet reconstruction algorithms and apply these criteria to the existing software. These criteria reference the Monte-Carlo data, which will not be available during real physics runs. To resolve this problem, we introduce the bisector method in Section 4.3.1, which allows the estimation of the performance of the jet reconstruction algorithms using only measured quantities.



(a)



(b)

Figure 4.1: *Distribution plots for the full reconstructed dataset. In (a) the distribution of the transverse momentum of the Z particle* p_t^Z *is plotted, while in (b) the distribution of the pseudorapidity of the Z particle* η^Z *is given.*

4.1 Definitions

In each simulated event in our sample, an interaction takes place in which a Z particle is produced together with a quark or gluon. The Z-particle eventually decays into a e^+e^- pair or a $\mu^+\mu^-$ pair, while the opposing quark or gluon showers and produces a jet. From the MC truth information one can extract the kinematical properties of all the particles that participate in the event. We define the quantities p_t^Z, η^Z and ϕ^Z to be the (p_t, η, ϕ) values of the Z-particle just before it decays into the e^+e^- or $\mu^+\mu^-$ pair. Please note that the kinematical properties of the Z-particle just before this decay need not be the same as the kinematical properties at the time of creation, as the Z-particle may have undergone one or more scattering processes. We also define the quantities p_t^{Opp}, η^{Opp} and ϕ^{Opp} as the (p_t, η, ϕ) values of the quark or gluon that was created along with the Z-particle. After processing an event, the jet reconstruction algorithm reports the number of jets n_{jets} that it has reconstructed. For every reconstructed jet, the kinematical properties (p_t, η, ϕ) are made available. We shall refer to these quantities as $p_t^{\mathrm{jet}}(i)$, $\eta^{\mathrm{jet}}(i)$ and $\phi^{\mathrm{jet}}(i)$ for $1 \leq i \leq n_{jets}$. In the sequel, we shall always be interested in the kinematical properties of the jet with maximal p_t. We thus define

$p_t^{\text{jet}} = p_t^{\text{jet}}(m)$, $\eta^{\text{jet}} = \eta^{\text{jet}}(m)$ and $\phi^{\text{jet}} = \phi^{\text{jet}}(m)$, where $1 \leq m \leq n_{\text{jets}}$ is the index of the jet with maximal p_t, i.e. $p_t^{\text{jet}}(m) = \max\left\{p_t^{\text{jet}}(i) \mid 1 \leq i \leq n_{\text{jets}}\right\}$.

The jet reconstruction algorithm can also be applied to the final state partons present in the MC truth information. We shall refer to the jets produced in this manner as MC truth jets. It is useful to compare the reconstructed jets with the MC truth jets, but this immediately presents a matching problem, since the former will not correspond one-to-one with the latter. Since we are only interested in the jet with the highest transverse momentum, we will only consider the MC truth jet that has minimal $\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2}$ with respect to this highest p_t jet. We define the quantities $(p_t^{\text{MCJet}}, \eta^{\text{MCJet}}, \phi^{\text{MCJet}})$ to be the corresponding kinematical properties of this nearest MC truth jet.

In order to compare different reconstructed quantities, we introduce the so-called fractional imbalance

$$f^{\text{imb}}(x, y) = (x - y)/y. \tag{4.1}$$

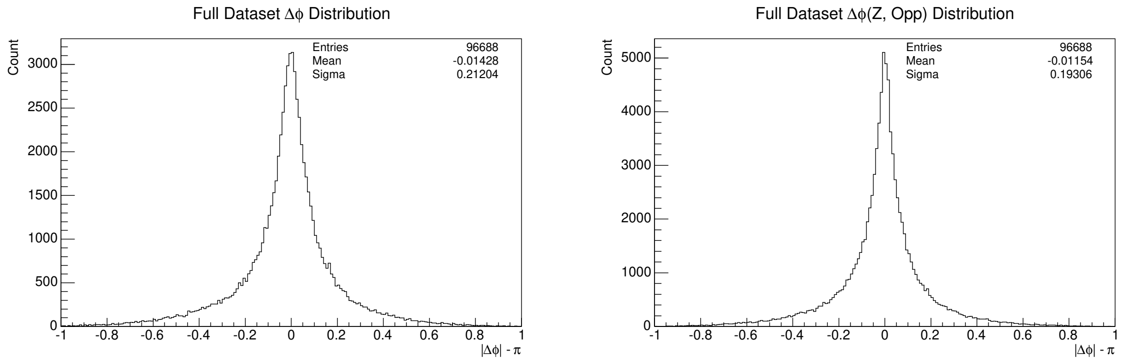
Using this imbalance we can define the following average quantities for a sample S of reconstructed events.

$$\begin{array}{lcl} F_S^{\text{imb}}(\text{Jet}, \text{Z}) & = & \frac{1}{\#S}\sum_{i\in S} f^{\text{imb}}(p_t^{\text{jet}}(i), p_t^{\text{Z}}(i)), \\ F_S^{\text{imb}}(\text{Jet}, \text{Opp}) & = & \frac{1}{\#S}\sum_{i\in S} f^{\text{imb}}(p_t^{\text{jet}}(i), p_t^{\text{Opp}}(i)), \\ F_S^{\text{imb}}(\text{Jet}, \text{MC}) & = & \frac{1}{\#S}\sum_{i\in S} f^{\text{imb}}(p_t^{\text{jet}}(i), p_t^{\text{MCJet}}(i)). \end{array} \tag{4.2}$$

We will often drop the subscript S if it is clear which sample is meant.

4.2 Selection cuts

Not every Z+jet event in the sample is equally suited for calibration purposes. In the ideal case, there is exactly one reconstructed jet, whose transverse momentum perfectly balances that of the Z-particle. In practice however, there are a number of effects that disturb this ideal situation. In Figure 4.2(a) for example it is clearly illustrated that $\vec{p}_t^{\,jet} + \vec{p}_t^{\,Z} \neq 0$. There are a number of effects that are responsible for this situation such as initial state radiation (ISR), final state radiation (FSR), detector resolution and jet reconstruction resolution. If one wants to use the recoiling Z to balance the p_t of the jet, it is clear that some events will be more useful than others. To clean up the sample under consideration, one can define a number of selection criteria called cuts. Only those events in the sample that pass these criteria will then be used for calibration purposes. The aim of this selection procedure is to minimize the fractional imbalance $\left|F^{\text{imb}}(\text{Jet}, \text{Z})\right|$, whilst keeping a reasonable fraction of the original events. It should be obvious that this is a somewhat subjective goal and that trade-offs will have to be made. The cuts we introduce should also be applicable to the situation during Atlas startup, which means that we should restrict the quantities we use to



(a) (b)

Figure 4.2: *In (a) the $|\Delta\phi| - \pi$ distribution of the full dataset is shown. As can be seen, not all jets are back to back in the transverse plane with the recoiling* Z*-particle. From (b) the effect of initial state radiation can be seen: the opposing parton is not back to back with the Z particle.*

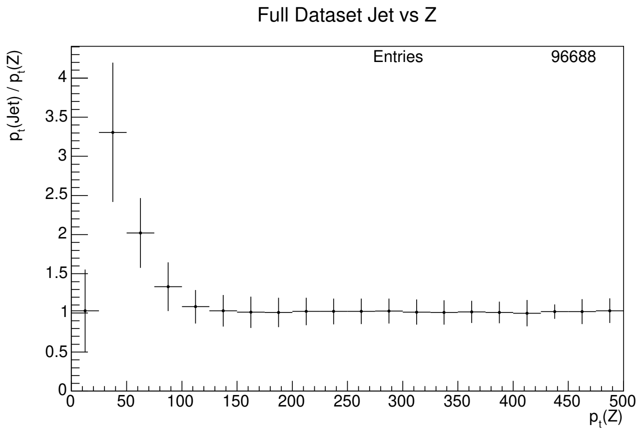


Figure 4.3: *As can be seen from this plot, at low values of p_t^Z, the match between the quantities p_t^{jet} and p_t^Z is lost. In this regime our assumption that the reconstructed jet with highest p_t was caused by the parton created along with the Z-particle is no longer valid.*

Cut	Parameters α						
$\Delta\Phi$-cut	0.14	0.34	0.64	1.14	∞		
n_{jets}-cut	3	4	5	6	∞		
$n_{central}$-cut	1	2	∞				
ΔR - cut	0	1	1.5	2	2.5	3	3.5
p_t^Z-cut	0	90	100				
η^Z-cut	2.3	∞					

Table 4.1: *The values of the cut parameter α that were used for each different type of cut.*

those that will be known during the calibration. For example, it would be pointless to use p_t^{jet} in our cuts, since this is the quantity we are calibrating.

An obvious selection cut to ensure the back-to-backness of the Z particle and the corresponding jet is to restrict the quantity $\Delta\phi = \phi^Z - \phi^{\text{jet}}$ by imposing a constraint $\big||\Delta\phi| - \pi\big| < \alpha$ for some appropriate value of α. Suitable values for the parameter α can be determined by examining Figure 4.2(a).

It should also be clear that it is undesirable to keep events in which a large number of jets were reconstructed. An increasing number of reconstructed jets increases the chance that the jet originating from the $Z + q/g$ vertex was incorrectly split into multiple jets. It is therefore logical to use a restriction of the form $n_{\text{jets}} \leq \alpha$ for some integer α. It should be noted that simply requiring $n_{\text{jets}} = 1$ does not give acceptable results, since the $k_\perp$ algorithm has a tendency to reconstruct spurious jets. Also remnants from the underlying events are reconstructed, possibly together with jets resulting from initial and final state radiation. As a weaker alternative for this cut, the number n_{central} of jets in the central barrel region $-3.2 \leq \eta \leq 3.2$ can be restricted, leading to a constraint of the form $n_{\text{central}} \leq \alpha$. In order to eliminate events in which the jet originating from the $Z+q/g$ vertex was incorrectly split, one can consider the minimal spatial separation ΔR between the highest p_t jet and the other reconstructed jets. In case of an incorrect split, two nearby jets will be reconstructed and thus this quantity will be small. This observation suggests the cut $\Delta R\big(\text{jet}(m), \text{jet}(i)\big) > \alpha$ for all $1 \leq i \leq n_{\text{jets}}$ with $i \neq m$, where m is the index of the jet with maximal transverse momentum p_t.

During the generation of the dataset a hard momentum cutoff of 100 GeV/c was employed. As can be seen from Figure 4.3, this leads to a matching problem in the low momentum range. A constraint of the form $p_t^Z > \alpha$ can be used to get rid of this effect. The last cut we employed, was a restriction on the pseudo rapidity η of the Z-particle, i.e. a constraint of the form $\eta^Z < \alpha$. Such a cut is related to the geometry of the detector and allows one to discard regions of the Atlas detector where jet reconstruction is still problematic, such as the forward calorimeters.

In order to evaluate the performance of these cuts, all possible combinations of the criteria discussed above were considered. In Table 4.1 the different values of the parameter α that were used are listed. Each combination was applied to the full dataset to give a reduced dataset S_{red}. For each

	Nr events	$\Delta\phi$-cut	n_{jets}-cut	p_t^Z-cut	η^Z-cut	$F^{\text{imb}}(\text{Jet}, \text{Z})$	$F^{\text{imb}}(\text{Opp}, \text{Z})$
	96000	-	-	-	-	6.3%	1.9%
A	32000	$\lvert\lvert\Delta\phi\rvert - \pi\rvert < 0.14$	$n_{\text{jets}} < 6$	$p_t^Z > 110$	-	0.60%	0.33%
B	28000	$\lvert\lvert\Delta\phi\rvert - \pi\rvert < 0.14$	$n_{\text{jets}} < 6$	$p_t^Z > 110$	$\eta^Z < 2.3$	1.9%	1.7%

Table 4.2: *Performance of the cut combinations that will be used in the sequel. The cuts left unspecified in this table were not used.*

reduced dataset, the average fractional imbalances $F^{\text{imb}}_{S_{\text{red}}}(\text{Jet}, \text{Z})$ and $F^{\text{imb}}_{S_{\text{red}}}(\text{Z}, \text{Opp})$ were calculated and stored, together with the number of events in the reduced dataset. It is intuitively clear, that the accuracy with which the reconstructed transverse jet momentum $p_t(\text{jet})$ matches the transverse momentum of the Z-particle p_t^Z, can never exceed the accuracy with which the transverse momenta of the parton that caused the jet and the Z-particle match. An effective fractional imbalance was therefore associated to each combination of cuts, defined by

$$F^{\text{imb}}_{\text{eff}}(\{\text{cuts}\}) = \max\left(\left|F^{\text{imb}}_{S_{\text{red}}}(\text{Jet}, \text{Z})\right|, \left|F^{\text{imb}}_{S_{\text{red}}}(\text{Z}, \text{Opp})\right|\right), \tag{4.3}$$

where S_{red} is the reduced dataset resulting after the application of the cuts.

In Figure 4.4 a graphical overview of the effectiveness of the cut combinations is given, while Table 4.2 reports some statistics connected to two specific reduced datasets. Keeping in mind the requirements we impose on our cuts, it is clear that one cannot unambiguously decide which combination is best. The point in Figure 4.4 that corresponds to the selection criteria we shall use in the sequel to investigate the performance of the jet reconstruction algorithms can be found within the circle on that chart. Its properties can be found in Table 4.2, where it is listed as cut A. As can be seen, it combines a relatively small effective imbalance of less than 1% with a relatively high statistics ratio of more than 30%. The average imbalance between the Z and jet transverse momentum has been considerably reduced. Figure 4.5(a) also shows how the matching problem has been solved. However, for low momenta of the opposing parton there is still a residual imbalance, as shown in Figure 4.5(b). The cut B, which leads to $28k$ events, will be used to perform the calibration procedure.

4.3 Resolution

In this section, we study the resolution and jet energy scale that result from an application of the existing software on the sample of events under consideration. There are a number of natural ways in which these parameters can be defined. In the Athena framework, the custom way to measure these quantities is to compare the reconstructed jet energies with the MC truth jet energies. We study the performance of the jet reconstruction in eight different transverse momentum regions. The jet energy scale (linearity) and resolution are defined to be the mean and standard deviation

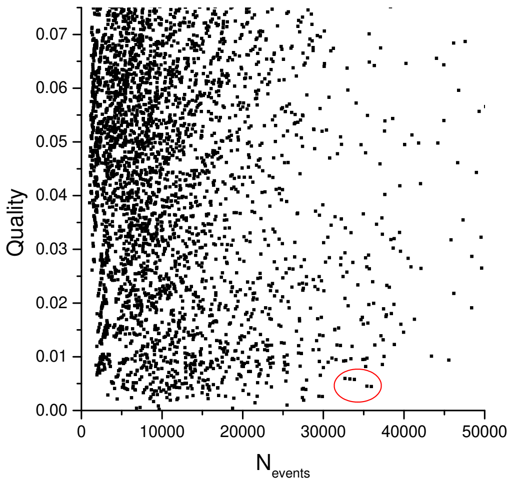
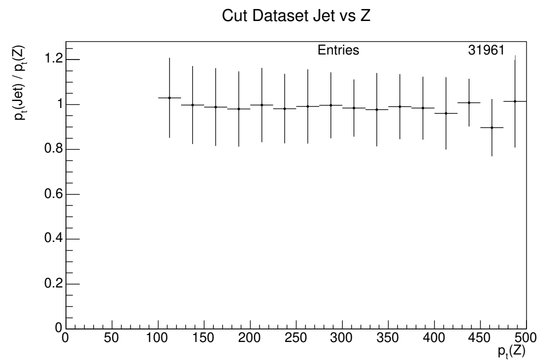


Figure 4.4: *Graphical view of the quality and effectiveness of the applied cuts. Each point corresponds to a single combination of cuts.*

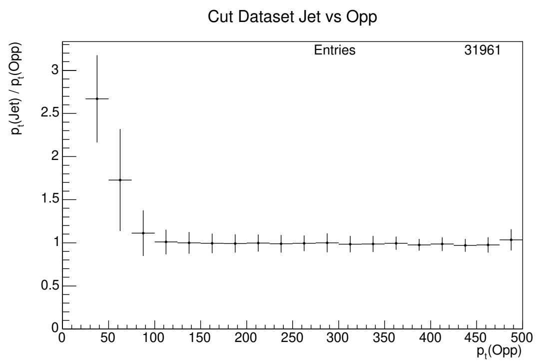
respectively of the best guassian fit through the fractional imbalances $f^{\text{imb}}(p_t^{\text{jet}}, p_t^{\text{MCJet}})$. The results can be found in Figure 4.6 and the corresponding resolutions and linearities have been reported in Table 4.3. For comparison purposes the mean μ and standard deviations σ of the corresponding fits through $f^{\text{imb}}(p_t^{\text{jet}}, p_t^{\text{opp}})$ have also been reported. It is interesting to observe that the mean values of these distributions are much better than the corresponding MCJet linearities. This could indicate that there is a problem with the procedure that calculates the MC truth jet energies. In any case, this effect has to be studied in greater detail. One other effect that needs to be corrected is illustrated in Figure 4.7(b). The average imbalance between the MCJet and the reconstructed jet is found to have an η dependence. The kinks in the distribution correspond precisely to the transition regions between different detectors. This will be corrected in future versions of the software. Comparing the resolutions with the design requirement $\sigma = \frac{50\%}{\sqrt{p_t/[\text{GeV/c}]}} \oplus 3\%$, it is clear that further improvements are still necessary. The MCJet linearities also do no yet meet the design goal of 1%, but the parton linearities do, which indicates that the 1% target is within reach.

4.3.1 Bisector Method

The calculation of the jet energy scale and resolution as defined above require knowledge of the MC truth information. In this section, we investigate a method to estimate these scales based only upon measured quantities. This method is called the bisector method and was introduced by the UA2 collaboration [11].



(a)



(b)

Figure 4.5: *In (a) one can see that the cut has improved the matching between the Z particle and the corresponding jet. However, from (b) we see that there is still an imbalance between the opposing parton and the jet.*

p_t^{MCJet}-bin [GeV/c]	100-140	140-180	180-220	220-260	260-300	300-340	340-380	380-500
Linearity μ	2.9%	2.6%	2.5%	1.7%	1.9%	1.7%	1.9%	2.2%
Resolution σ	7.4%	6.6%	5.8%	5.9%	5.3%	5.1%	4.8%	4.8%
$\mu(f^{\text{imb}}(p_t^{\text{jet}}, p_t^{\text{Opp}}))$	-0.75%	0.20%	0.28%	0.22%	0.32%	0.76%	0.23%	1.6%
$\sigma(f^{\text{imb}}(p_t^{\text{jet}}, p_t^{\text{Opp}}))$	10.5%	8.8%	8.0%	7.3%	6.9%	6.9%	5.7%	6.0%
Req. Res. σ_{req}	5.5%	5.0%	4.6%	4.4%	4.2%	4.1%	4.0%	3.8%

Table 4.3: *Linearities and resolutions in different p_t bins for the existing jet reconstruction software. The required resolutions σ_{req} follow from the Atlas design target $\sigma = \frac{50\%}{\sqrt{p_t/[\text{GeV/c}]}} \oplus 3\%$.*

As can be seen from Table 4.3, the resolution in the $F^{\text{imb}}(\text{Jet}, \text{Z})$ measurement is much larger than the corresponding $F^{\text{imb}}(\text{Jet}, \text{Opp})$ and $F^{\text{imb}}(\text{Jet}, \text{MCJet})$ resolutions. There are two different effects that contribute to the spread of the difference between the transverse momentum of the Z-particle and the jet. The first contribution is due to initial state radiation (ISR), which causes an imbalance between the transverse momentum of the Z-particle p_t^Z and that of the opposing parton $p_t(\text{Opp})$. This was illustrated in Figure 4.2(b). The second effect concerns the difference between $p_t(\text{jet})$ and $p_t(\text{Opp})$ and is characteristic for the jet energy reconstruction method. It takes into account the deterioration introduced by the calorimetric measurement and the physical hadronization process. A priori there is no correlation between these two effects, so we can write $\sigma_Z = \sqrt{\sigma_{\text{opp}}^2 + \sigma_{\text{ISR}}^2}$, where σ_{ISR} is the resolution caused by the ISR and σ_{opp} is the jet reconstruction resolution we are interested in. The bisector method gives us a way to estimate σ_{ISR} and hence σ_{opp}, since σ_Z can be measured.

Consider a Z+jet event and let $\vec{K}_T$ be the difference in transverse momentum between the Z-particle and the jet, i.e. $\vec{K}_T = \vec{p}_T^{\,jet} + \vec{p}_T^{\,Z}$. Define an orthogonal axis system $(\vec{e}_\eta, \vec{e}_\xi)$, where $\vec{e}_\eta$

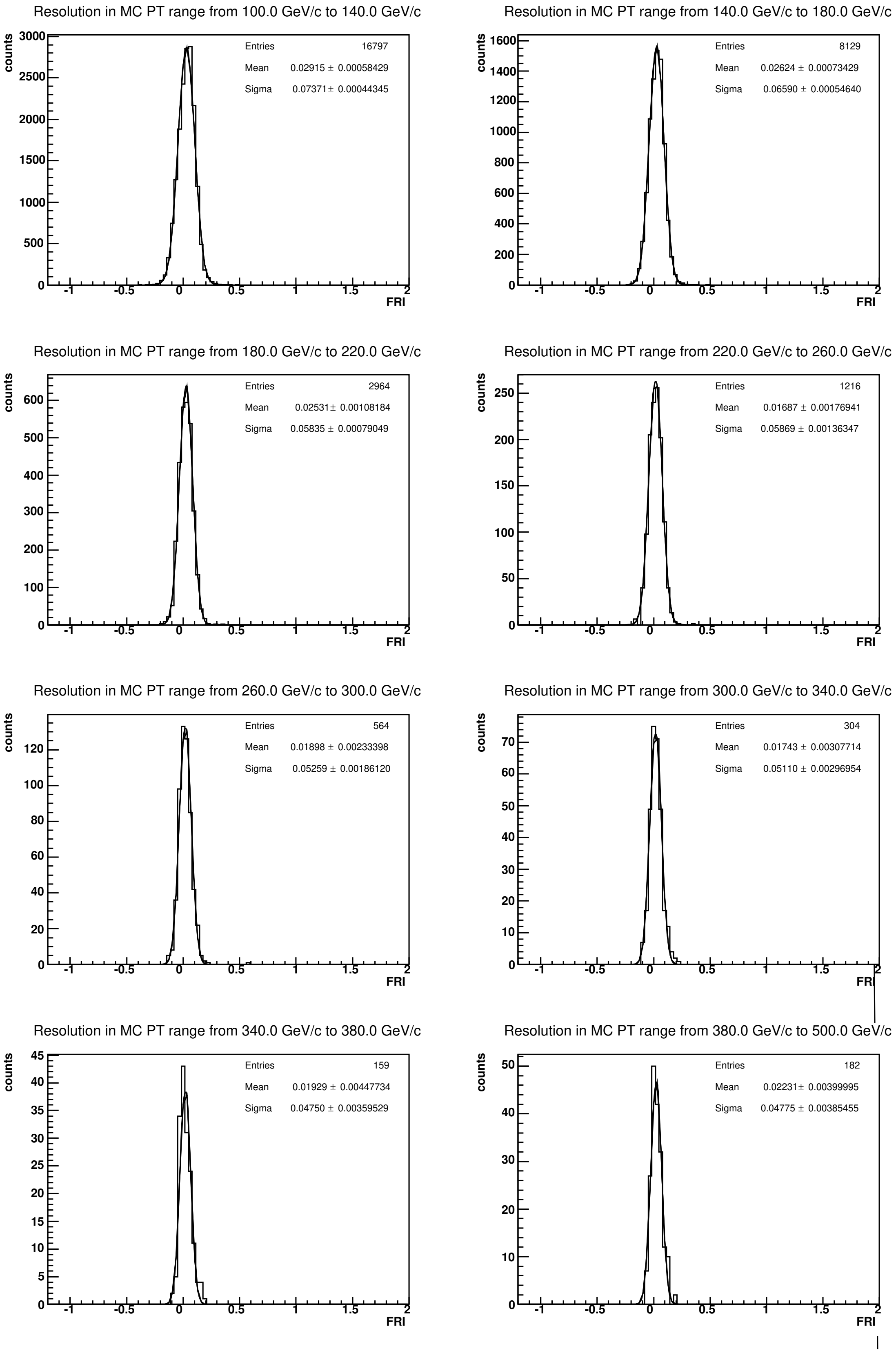
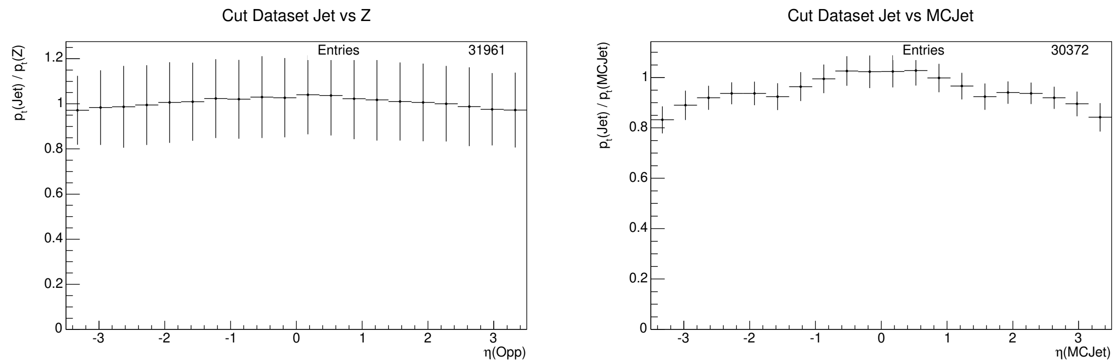


Figure 4.6: Resolution and linearity in different energy bins.



(a) (b)

Figure 4.7: *In (b) an effect in η is visible, which is not present in (a). The kinks in the graph correspond to the transition regions between different detector components. This issue has been addressed in later versions of Athena.*

is a unit vector that bisects the angle defined by the directions of the Z and the jet, i.e. $\vec{e}_\eta = \Big(\cos(\frac{\phi_{jet}+\phi_Z}{2}), \sin(\frac{\phi_{jet}+\phi_Z}{2})\Big)$. Let (K^T_η, K^T_ξ) be the components of $\vec{K}_T$ with respect to this system, i.e. $\vec{K}_T = K^T_\eta \vec{e}_\eta + K^T_\xi \vec{e}_\xi$. In [11] it was argued that the component K^T_η is more or less unaffected by the jet reconstruction resolution σ_D, while both components K^T_η and K^T_ξ are affected by ISR. This can be seen by noting that the $|\Delta\phi|$ distribution is relatively tightly centered around π, which means that K^T_ξ is the difference between two large opposite vectors, while K^T_η is the sum of two small contributions. If one does not apply a cut that destroys the isotropy in ϕ, the ISR will have the same effect on both components, since the coordinate system $(\vec{e}_\eta, \vec{e}_\xi)$ is spatially isotropic. One can thus make the estimation $\sigma_D = \sqrt{\sigma^2_\xi - \sigma^2_\eta}$. Based upon the ratio $\frac{\sigma_\eta}{\sigma_\xi}$, one calculates $\sigma_{\text{opp}} = \sigma_Z \frac{\sigma_D}{\sigma_\xi} = \sigma_Z \sqrt{1 - \frac{\sigma^2_\eta}{\sigma^2_\xi}}$. Since this method involves the ratio of two widths, it is less sensitive to the exact definition of σ_η and σ_ξ. This is particularly interesting due to the fact that K^T_η and K^T_ξ both have non-gaussian distributions. A Lorentzian fit works well, but has the problem that the standard deviation is not defined.

This procedure was applied to our dataset. Some typical K^T_η and K^T_ξ distributions can be found in Figures 4.8 and 4.9. The distributions were fitted with Lorentzian curves and the half widths at half maximum were taken as the standard deviation. The results of this procedure, together with the predicted values for σ_{opp}, can be found in Table 4.4. As can be seen in this table, the resolutions σ_{calc} calculated this way systematically overestimate the real parton resolutions σ_{opp}. This has to be investigated in further detail, particularly since a similar method, which was used for the fast-detector simulation analysis in [29], yielded very accurate results.

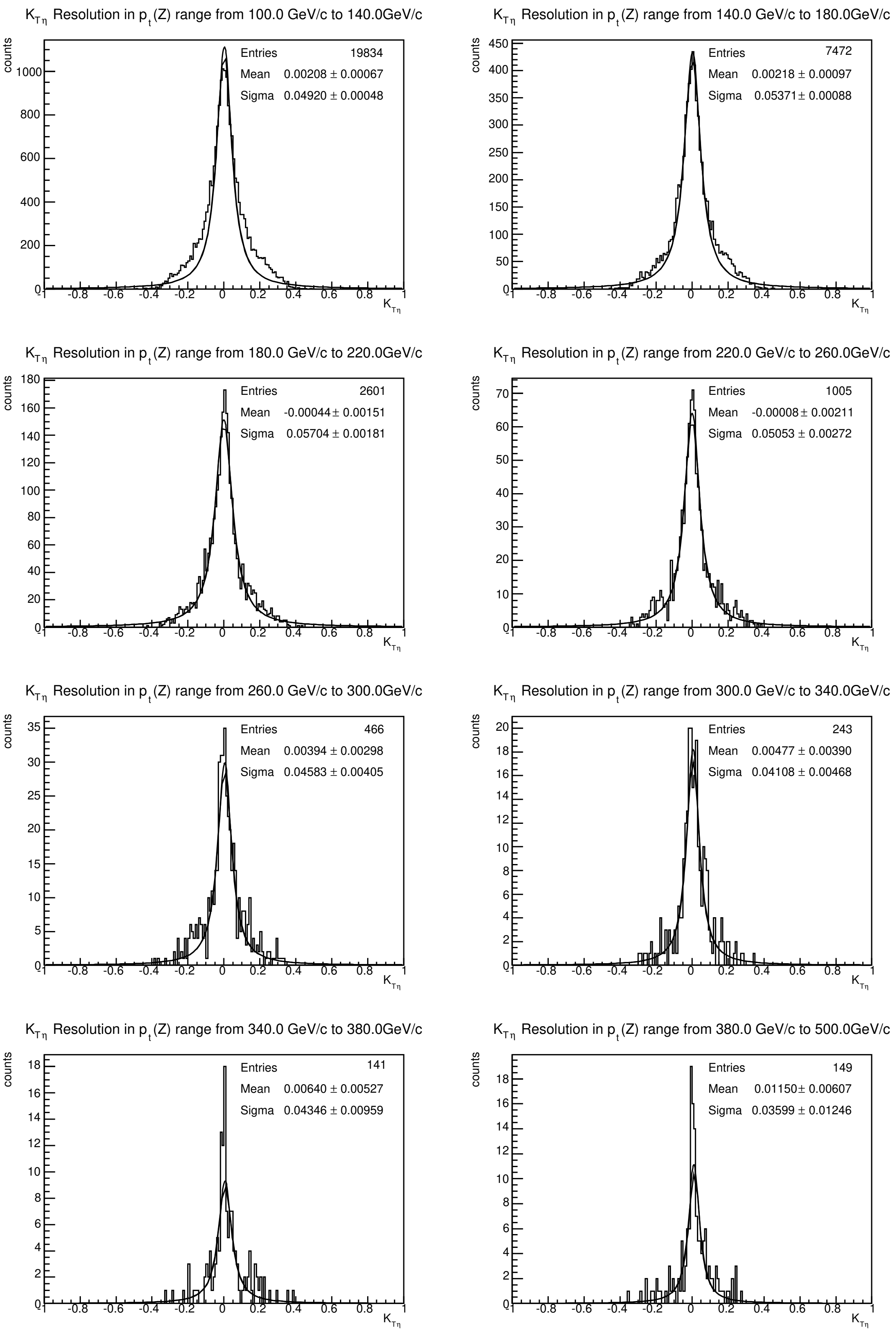


Figure 4.8: K_{η}^{T} *distributions in different* p_t^Z *bins.*

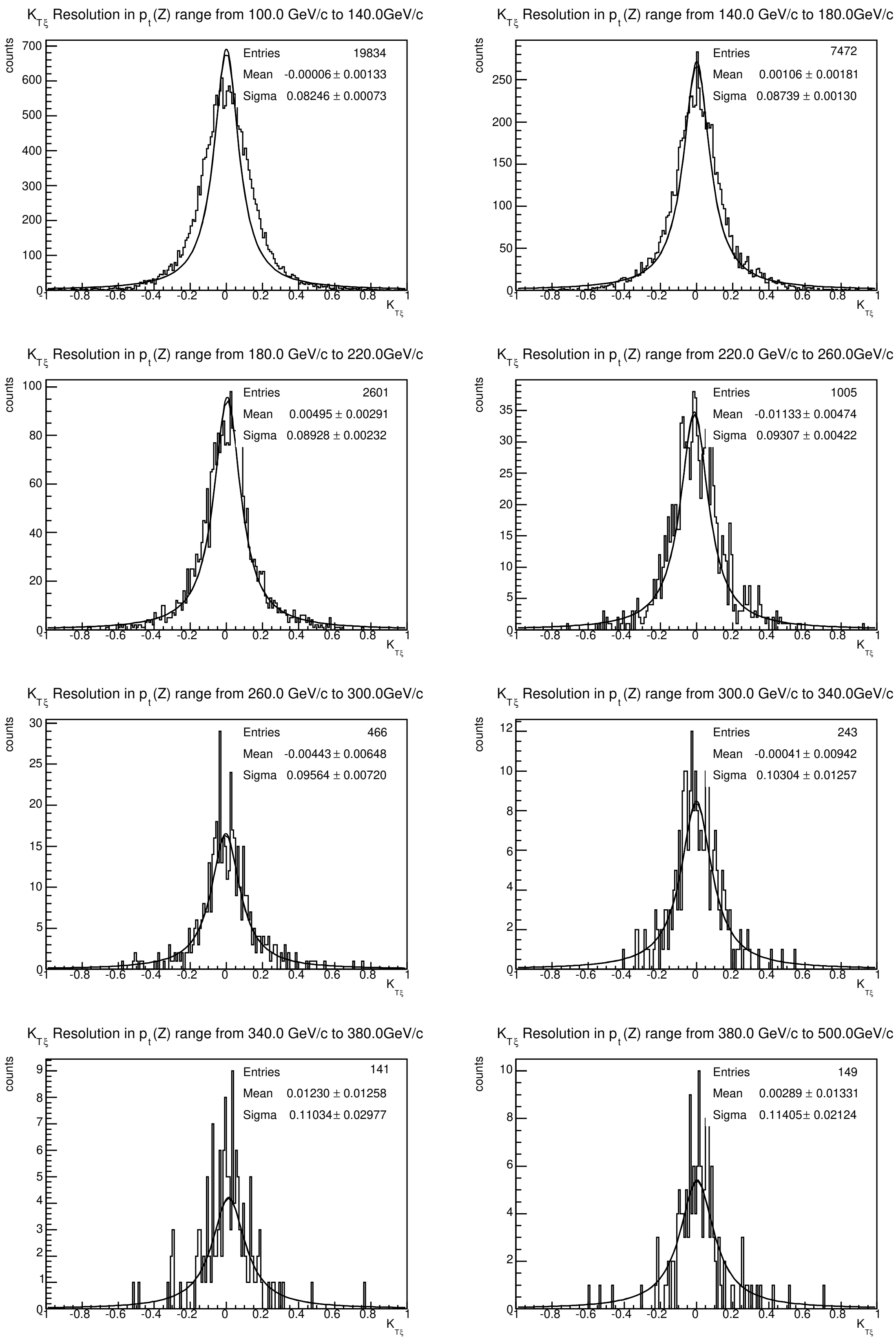


Figure 4.9: K_ξ^T *distributions in different* p_t^Z *bins.*

p_t^Z-bin [GeV/c]	100-140	140-180	180-220	220-260	260-300	300-340	340-380	380-500
σ_η	4.9%	5.4%	5.7%	5.1%	4.6%	4.1%	4.3%	3.6%
σ_ξ	8.2%	8.7%	8.9%	9.3%	9.6%	10.3%	11.0%	11.4%
σ_Z	14.8%	14.0%	13.0%	13.3%	11.5%	10.8%	10.3%	8.8%
σ_{calc}	11.9%	11.0%	10.0%	11.1%	10.1%	9.9%	9.5%	8.3%
σ_{opp}	10.5%	8.8%	8.0%	7.3%	6.9%	6.9%	5.7%	6.0%

Table 4.4: *Results yielded by the bisector method. The resolutions σ_{calc} are the predicted values for the parton resolutions, while the σ_{opp} values are the true resolutions.*

4.4 H1 Weighting

In the previous section, all the results were obtained by means of a H1-weighting procedure, where the weights were extracted using MC truth information. The goal of this section is to derive a new set of H1-weights exploiting the balance between p_t^Z and p_t^{jet} present in the dataset under consideration and to compare the performance of these new weights with that of the old weights. A similar procedure will have to be performed during the LHC startup period to calibrate the jet reconstruction software.

The dataset we used to perform the weighting was obtained by applying the cut B in Table 4.2 to the full sample of reconstructed events. Due to technical considerations, we had to restrict the dataset to the range $\eta < 2.3$. As can be seen from this table, this leads to a slight deterioration of the balance between p_t^Z and p_t^{jet} when compared with cut A. This performance decrease is limited mostly to the lowest energy bin and hence does not affect our results too much.

An optimal set of weights was determined by minimizing the linear least squares residual

$$\mathcal{F}(\{w_i^r\}, S) = \sum_{j \in S} \frac{1}{n(p_t^{\text{MCJet}}(j))} \Big(\big(\frac{p_t^{\text{rec}}(\{w_i^r\})(j) - p_t^{\text{MCJet}}(j)}{\sigma_{\text{jet}}(j)} \big)^2 + \Lambda \big(\frac{p_t^{\text{rec}}(\{w_i^r\})(j) - p_t^{\text{MCJet}}(j)}{\sigma_{\text{jet}}(j)} \big) \Big), \tag{4.4}$$

where S is the cut dataset under consideration. The function n is the density of events in the specified p_t bin. Normalizing the contribution of each jet in the residual $\mathcal{F}$ according to the number of jets in the relevant p_t bin allows each energy range to have an equal contribution, which prevents a deterioration of the resolution at high energies for a slight performance increase at low energies. The Lagrange multiplier Λ can be used to control the absolute energy scale. In our case, it was made energy dependent to minimize the fractional imbalance $F^{\text{imb}}(\text{Jet}, \text{Z})$. As discussed in Section 3.4, the jet energy resolution σ_{jet} was taken to be

$$\sigma_{\text{jet}}(j) = p_t^{\text{MCJet}}(j) \sqrt{(0.03)^2 + \Big(\frac{0.5}{\sqrt{p_t^{\text{MCJet}}(j)/[\text{GeV}/c]}} \Big)^2}. \tag{4.5}$$

A singular value decomposition was used to minimize the residual $\mathcal{F}$, which led to a set of weights which we shall refer to as the *unparametrized* weights. In order to reduce the instabilities that arise due to the presence of so many free variables and to compare our results with the set of weights now in use in the Atlas software, a parametrization $w_i^r = a^r + b^r \exp(-c^r i)$ was introduced. The shape is based upon the fact that lower cells need to be corrected upward further than higher energy cells, since they contain mostly hadronic energy. We shall refer to the set of weights corresponding to the minimum of $\mathcal{F}$ with this parametrization as the *parametrized* weights.

The weights yielded by the above procedure can be found in Tables 4.7 and 4.6. For comparison purposes, the weights that are currently in use in the reconstruction software are displayed in Table 4.5. In Figure 4.10 the various sets of weights have been plotted together in order to further ease comparison. As can be expected, the parametrized weights can be seen as an exponential curve more or less fitting the unparametrized weights. The graphical view also illustrates that the form of the parametrization is relatively well justified. Notice also that the parametrized set of weights does not deviate too much from the original weights, which is quite promising.

The performance of the two new set of H1 weights was studied, using the same p_t bins as in Section 4.3. The imbalances are illustrated graphically in Figures 4.11 to 4.13, while the parameters corresponding to the guassian fits are tabulated in Tables 4.8 to 4.13. In order to test the robustness of the various fits, the unparametrized fitting procedure was repeated using only half of the events in the sample and the resulting resolutions and linearities for the full sample are presented in Figures 4.14 to 4.15 and the tables listed above.

Both the parametrized and unparametrized minimization procedures could be tuned using the Lagrange multiplier Λ to yield Z+jet imbalances of less than 1.2% in all momentum bins. As can be expected, the corresponding resolutions are better than the Z-resolutions yielded by the original weights. The unparametrized set of weights leads to MCJet linearities and resolutions comparable to those of the old weights. It performs slightly better than the old set of weights with respect to the opposing partons. It is interesting to observe how the parametrized weights outperform both the unparametrized weights and the old weights when measured with respect to the opposing parton, while the MCJet linearities are considerably worse. As in Section 4.3, we see that the linearities with respect to the opposing parton are significantly better than their MCJet counterparts. In general however, we find that the performance of the new set of H1-weights is similar to that of the existing set of weights, which is a promising result. Furthermore, the performance decrease obtained when only half the dataset was used to generate the H1 weights is relatively small. This fact allows one to conclude that the H1 weighting procedure using $Z + jet$ events is quite robust in the momentum region examined here. However, we have to remark that this region is relatively insensitive to inbalances caused by ISR when compared to the low momentum region.

Bin [GeV]	<0.044	<0.088	<0.18	<0.35	<0.71	<1.41	<2.83	<5.66	<11.3	>11.3
EMB2	1.75	1.42	1.23	1.13	1.07	1.04	1.02	1.01	1.01	1.00
EMB3	0.93	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87
EME2	1.11	1.06	1.04	1.03	1.02	1.02	1.02	1.02	1.02	1.02
EME3	1.24	1.22	1.20	1.19	1.18	1.18	1.17	1.17	1.17	1.16
Tile	6.62	4.05	2.65	1.88	1.46	1.23	1.10	1.03	1.00	0.98
Hec	1.11	1.11	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10
Cryo	0.73									
Gap	0.81									

Table 4.5: The H1 weights that are currently used in the Atlas software.

Bin [GeV]	<0.044	<0.088	<0.18	<0.35	<0.71	<1.41	<2.83	<5.66	<11.3	>11.3
EMB2	0.71	-0.33	2.27	1.48	1.40	0.79	1.22	1.27	1.18	0.87
EME2	1.58	2.71	2.28	0.69	0.92	1.38	1.24	0.87	0.94	1.00
EMB3	4.00	0.77	2.59	2.52	0.54	0.97	1.70	1.57	1.31	0.61
EME3	1.34	3.75	3.17	-0.92	0.01	1.42	1.36	0.11	1.47	0.66
Tile	5.32	2.75	0.46	0.78	1.10	1.18	1.32	1.14	1.20	1.08
Hec	-4.30	3.28	0.82	-2.42	1.09	1.88	1.40	1.07	1.65	1.49
Cryo	0.25									
Gap	0.98									

Table 4.6: The H1 weights obtained using the unparametrized fitting procedure.

Bin [GeV]	<0.044	<0.088	<0.18	<0.35	<0.71	<1.41	<2.83	<5.66	<11.3	>11.3
EMB2	1.34	1.22	1.14	1.10	1.07	1.05	1.04	1.04	1.03	1.03
EME2	3.08	1.63	1.16	1.01	0.96	0.94	0.94	0.93	0.93	0.93
EMB3	4.70	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93
EME3	0.23	0.62	0.86	1.00	1.09	1.14	1.18	1.19	1.21	1.21
Tile	4.20	1.74	1.21	1.09	1.07	1.06	1.06	1.06	1.06	1.06
Hec	0.74	0.83	0.90	0.97	1.02	1.07	1.10	1.14	1.17	1.19
Cryo	0.64									
Gap	1.33									

Table 4.7: The H1 weights obtained using the parametrized fitting procedure.

Bin [GeV/c]	100-140	140-180	180-220	220-260	260-300	300-340	340-380	380-500
Old	2.9%	2.6%	2.5%	1.7%	1.9%	1.7%	1.9%	2.2%
Unparametrized	3.5%	2.9%	2.2%	1.8%	1.7%	1.4%	1.4%	1.3%
Parametrized	3.0%	3.2%	3.1%	2.7%	2.5%	2.4%	2.5%	2.0%
Unpar. (Half Set)	3.9%	3.5%	2.8%	2.5%	2.7%	2.4%	2.2%	2.6%

Table 4.8: Linearities resulting from guassian fits through $f^{\mathrm{imb}}(\mathrm{MCJet}, \mathrm{Jet})$ for different p_t bins.

Bin [GeV/c]	100-140	140-180	180-220	220-260	260-300	300-340	340-380	380-500
Old	7.4%	6.6%	5.8%	5.9%	5.3%	5.1%	4.8%	4.8%
Unparametrized	7.4%	6.7%	5.9%	5.3%	5.0%	4.9%	4.7%	5.3%
Parametrized	5.6%	5.1%	4.6%	4.2%	3.9%	3.7%	4.0%	3.6%
Unpar. (Half Set)	8.6%	7.6%	6.6%	5.9%	5.5%	5.8%	5.5%	5.9%

Table 4.9: Resolutions resulting from guassian fits through $f^{\rm imb}(\rm MCJet, Jet)$ for different p_t bins.

Bin [GeV/c]	100-140	140-180	180-220	220-260	260-300	300-340	340-380	380-500
Old	2.3%	0.74%	0.54%	0.33%	0.06%	0.26%	0.27%	1.3%
Unparametrized	-0.48%	-0.28%	-0.60%	-0.46%	-0.54%	-0.26%	-0.54%	0.73%
Parametrized	-0.56%	0.34%	0.39%	0.45%	0.61%	0.84%	0.66%	0.80%
Unpar. (Half Set)	-0.16%	-0.32%	0.10%	0.08%	0.44%	0.64%	0.61%	1.7%

Table 4.10: Linearities resulting from guassian fits through $f^{\rm imb}(\rm Opp, Jet)$ for different p_t bins.

Bin [GeV/c]	100-140	140-180	180-220	220-260	260-300	300-340	340-380	380-500
Old	10.1%	8.6%	7.8%	7.2%	6.8%	6.8%	5.5%	5.9%
Unparametrized	9.6%	8.1%	7.0%	6.7%	6.2%	5.7%	4.8%	5.4%
Parametrized	7.4%	6.2%	5.8%	5.5%	4.7%	4.9%	4.6%	4.2%
Unpar. (Half Set)	10.9%	9.2%	8.0%	7.3%	6.9%	6.0%	6.0%	6.0%

Table 4.11: Resolutions resulting from guassian fits through $f^{\rm imb}(\rm Opp, Jet)$ for different p_t bins.

Bin [GeV/c]	100-140	140-180	180-220	220-260	260-300	300-340	340-380	380-500
Old	2.9%	0.09%	0.5%	0.09%	0.27%	0.19%	0.68%	0.93%
Unparametrized	-0.08%	0.69%	-0.34%	0.24%	0.39%	-1.21%	-0.75%	0.41%
Parametrized	-1.2%	0.85%	0.55%	1.22%	0.93%	1.00%	0.56%	0.18%
Unpar. (Half Set)	-0.53%	1.30%	0.38%	0.74%	1.60%	0.35%	0.77%	1.30%

Table 4.12: Linearities resulting from guassian fits through $f^{\rm imb}(\rm Z, Jet)$ for different p_t bins.

Bin [GeV/c]	100-140	140-180	180-220	220-260	260-300	300-340	340-380	380-500
Old	14.6%	13.8%	12.6%	13.1%	11.5%	11.2%	10.3%	8.8%
Unparametrized	14.1%	13.5%	12.2%	12.9%	10.6%	10.0%	10.1%	7.0%
Parametrized	12.5%	11.7%	11.1%	11.7%	9.7%	10.1%	9.6%	6.8%
Unpar. (Half Set)	15.1%	14.2%	12.7%	12.9%	11.2%	10.7%	8.5%	9.4%

Table 4.13: Resolutions resulting from guassian fits through $f^{\rm imb}(\rm Z, Jet)$ for different p_t bins.

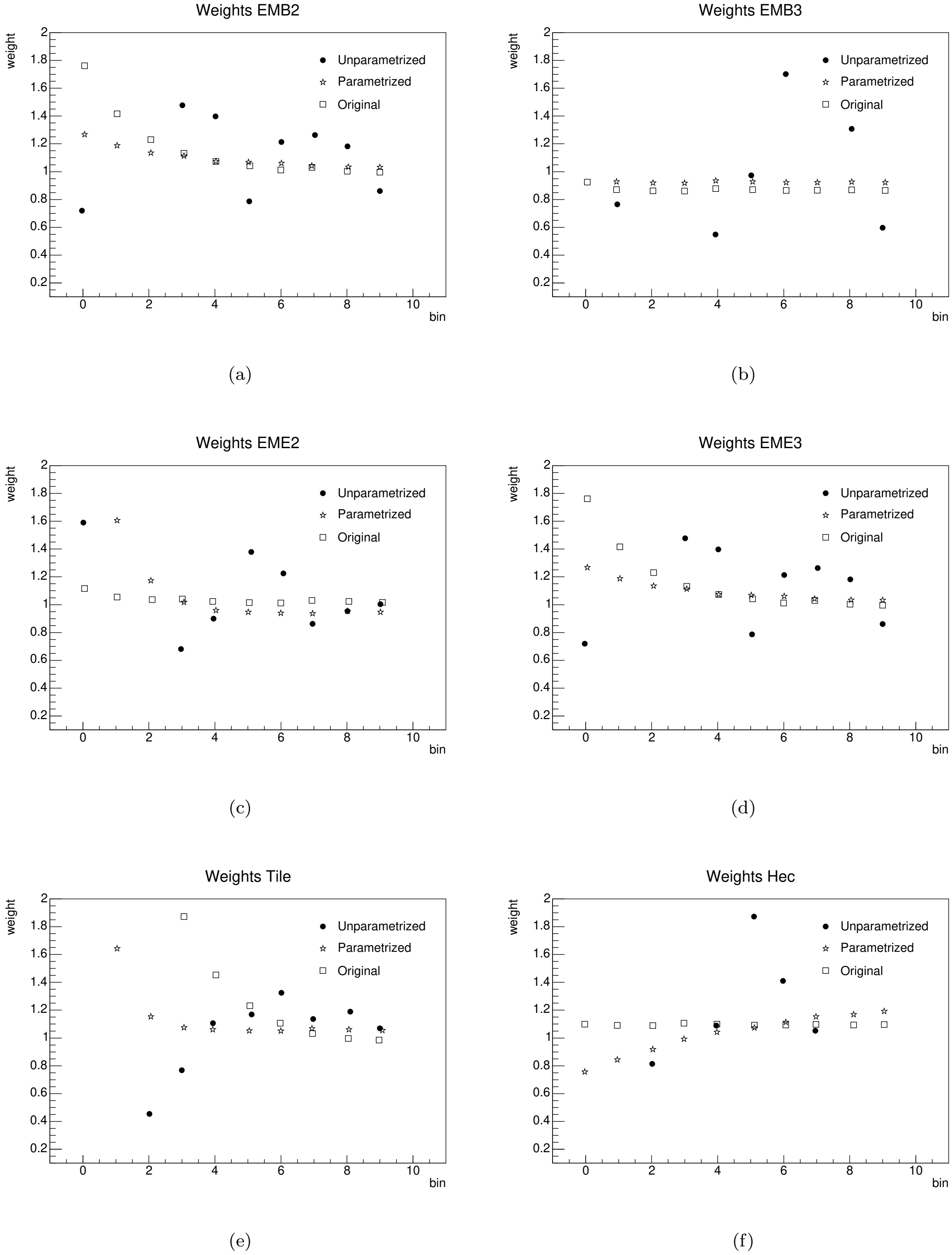


Figure 4.10: *Original, unparametrized and parametrized H1-weights for different regions in the detector.*

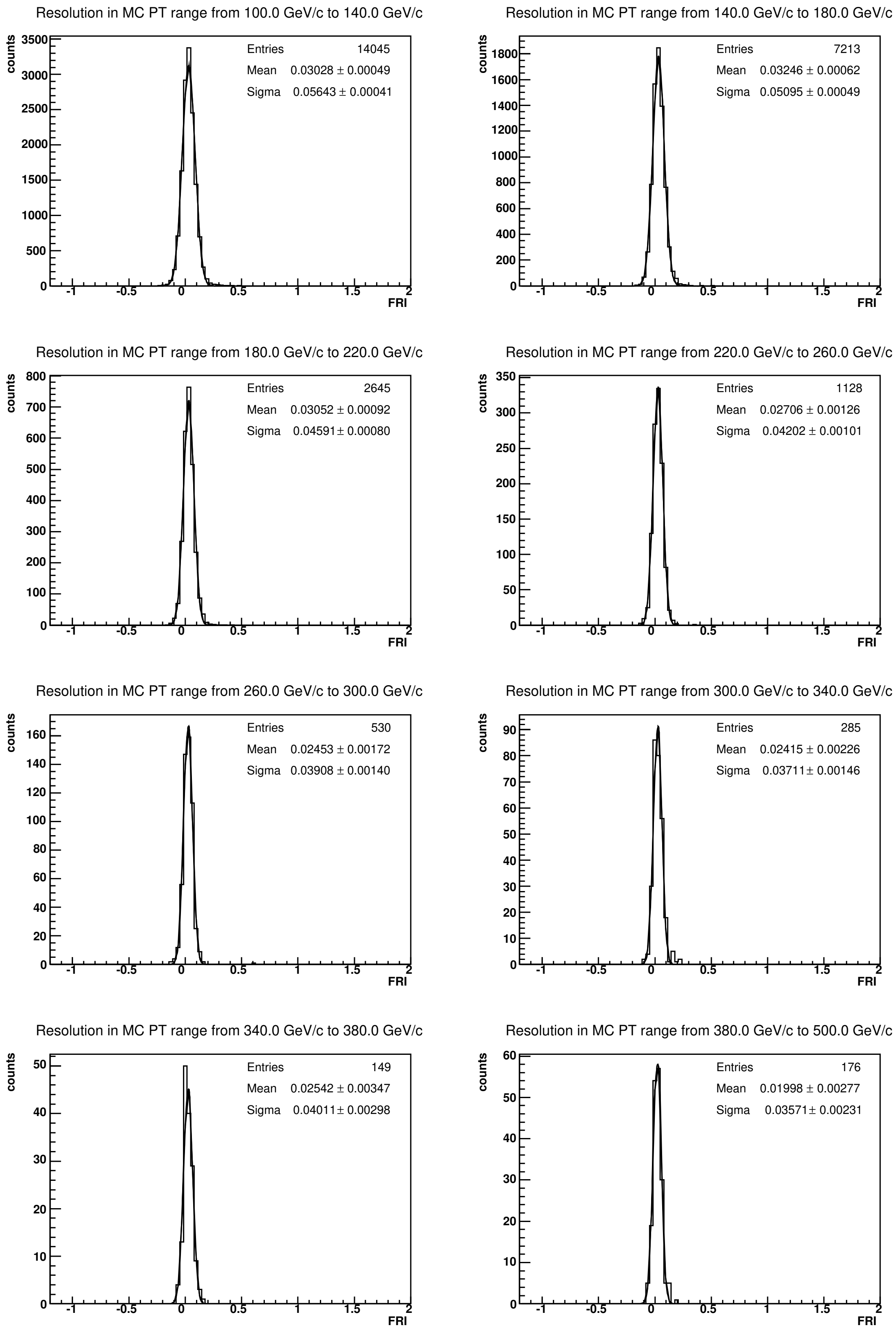


Figure 4.11: *Resolution plots with new parametrized H1 Weights.*

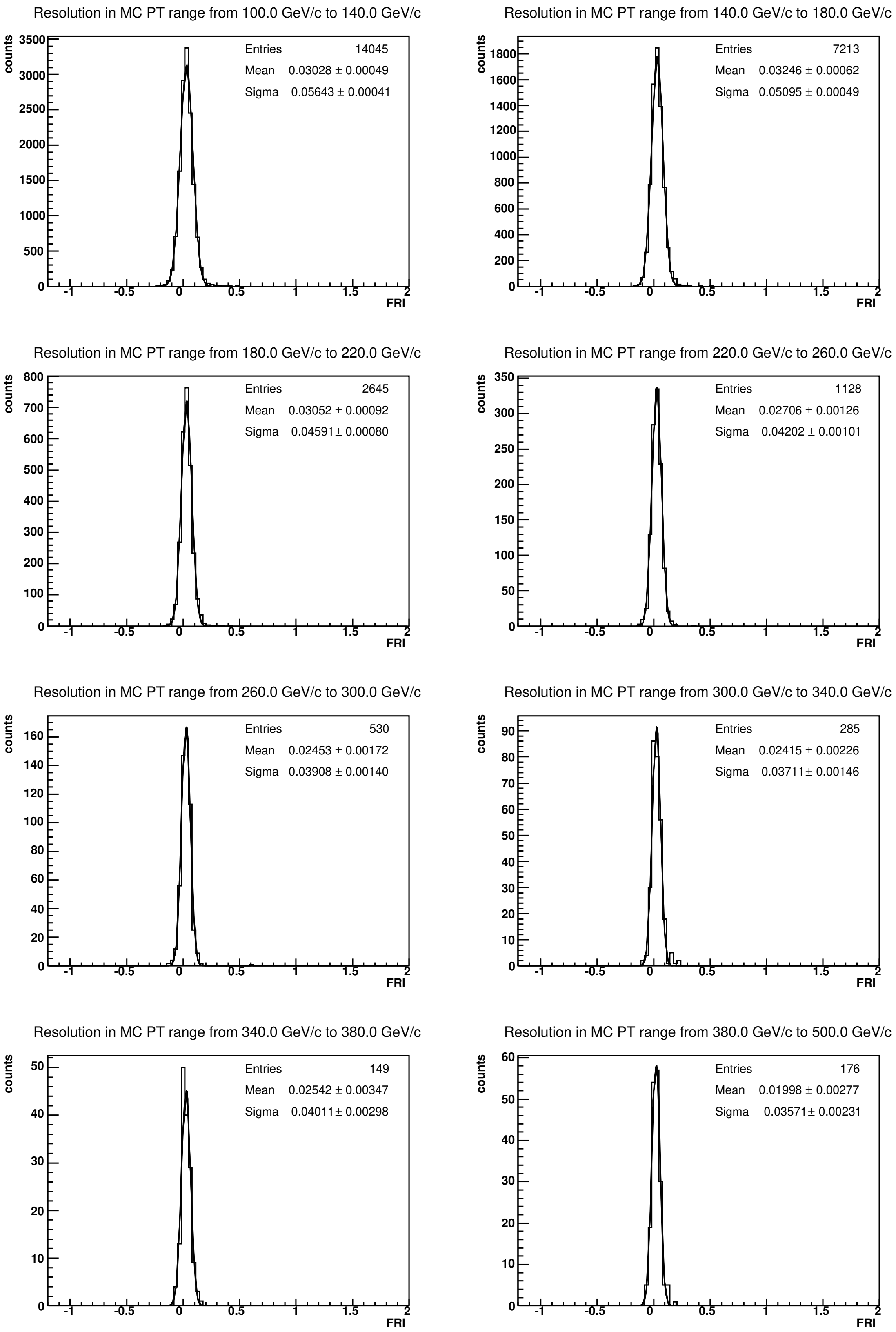


Figure 4.12: *Resolution plots with new unparametrized H1 Weights.*

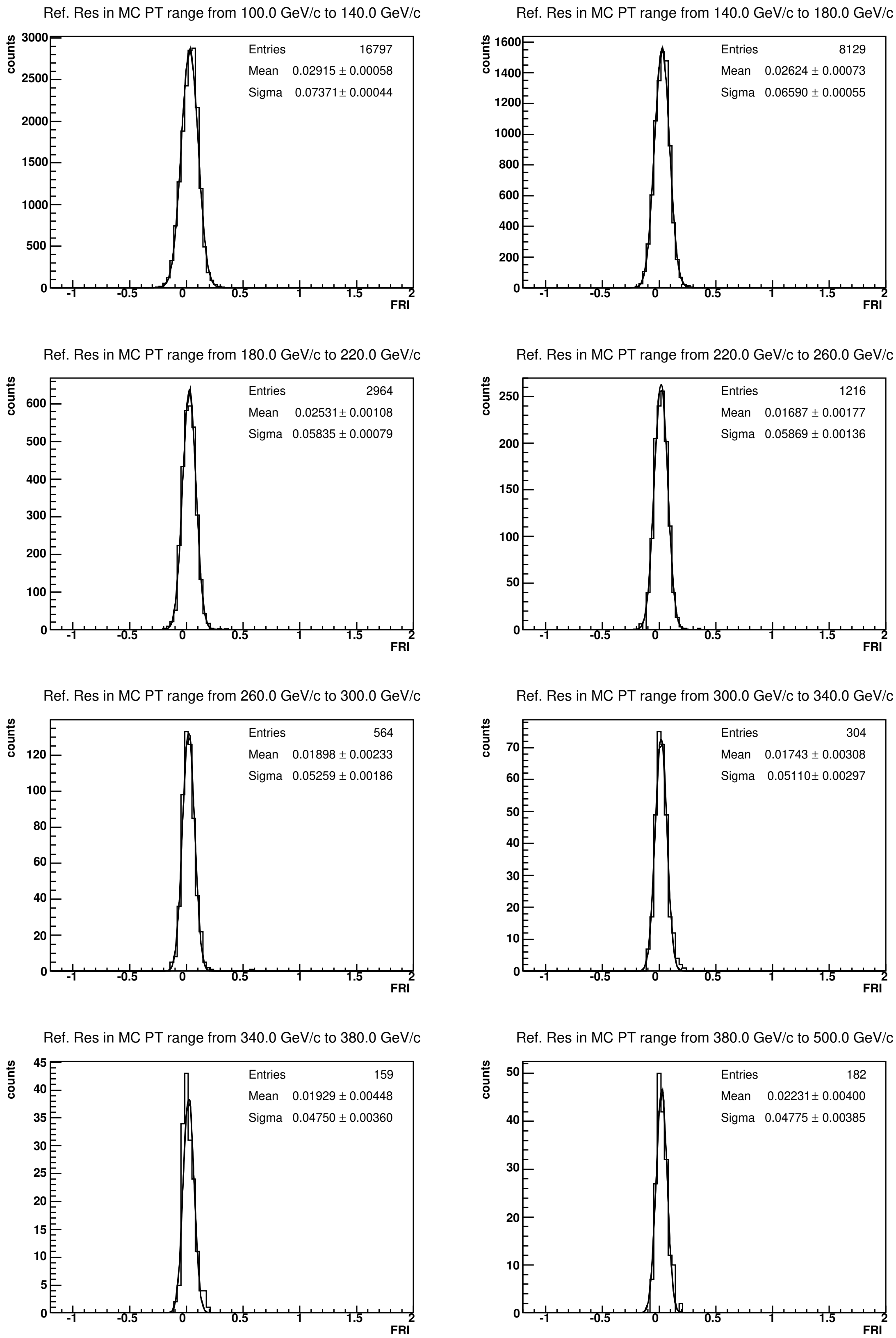


Figure 4.13: *Resolution plots with old H1 Weights.*

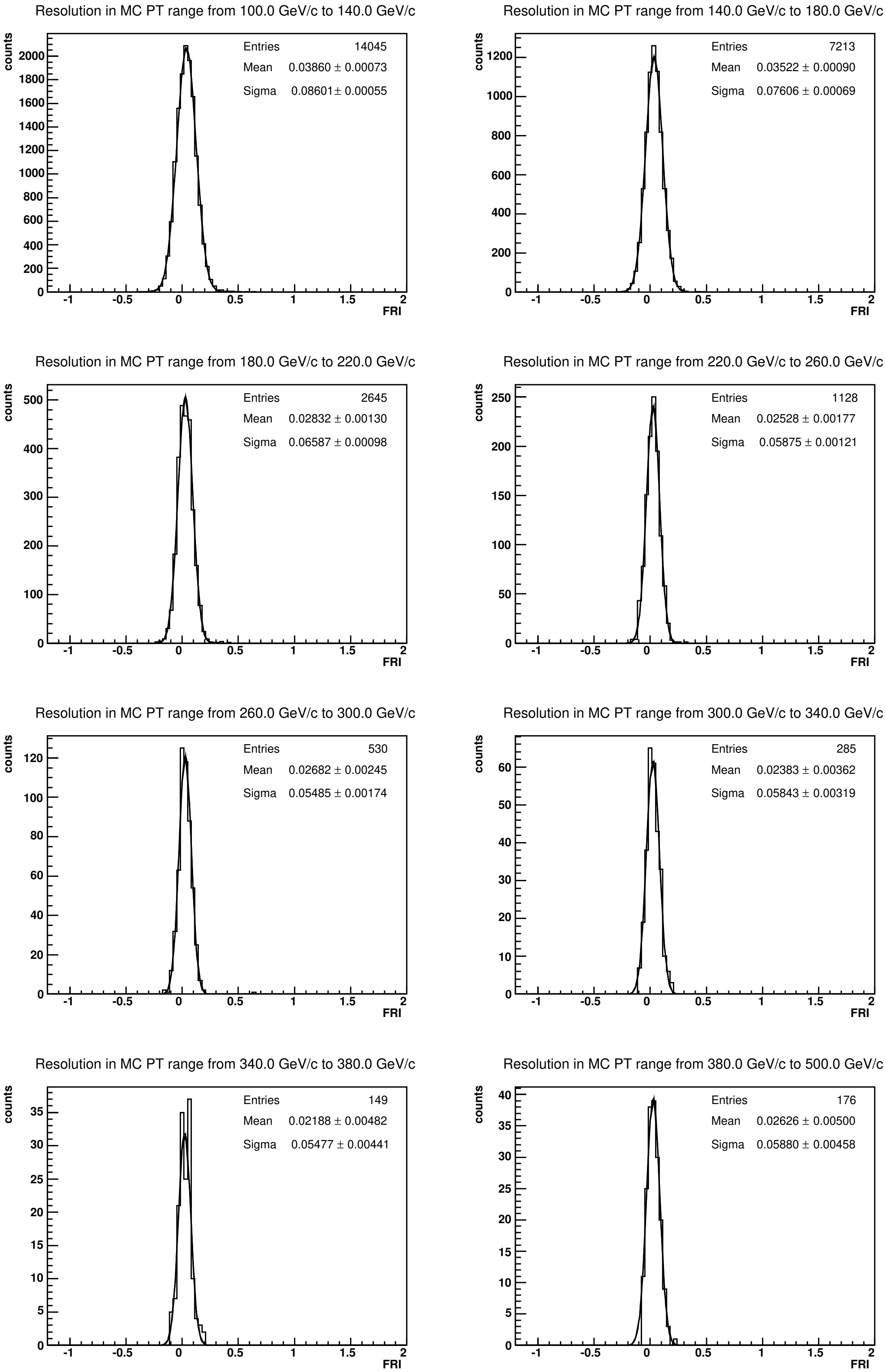


Figure 4.14: *Resolution plots with unparametrized H1 Weights obtained using only half of the dataset.*

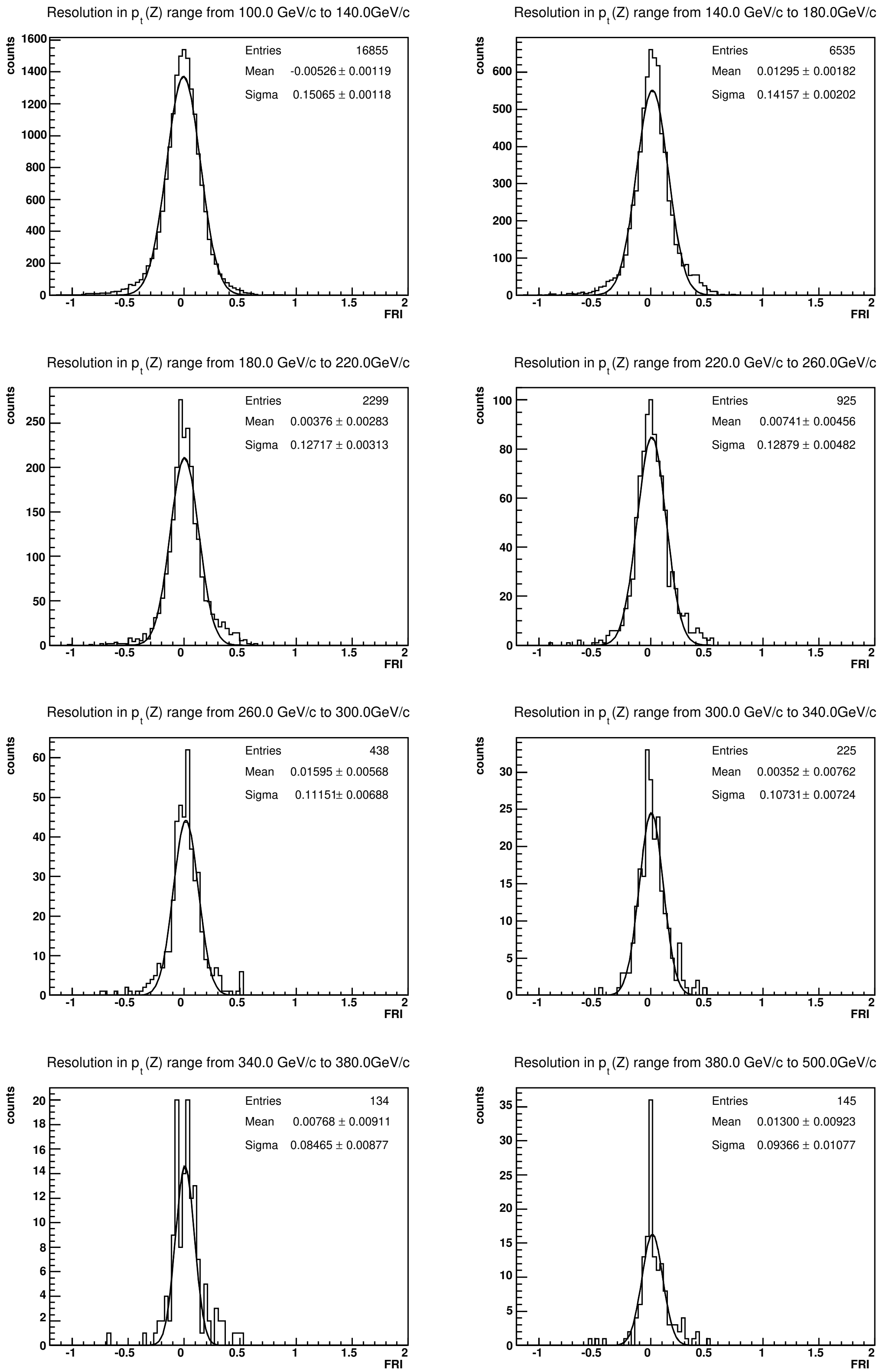


Figure 4.15: *Resolution plots with unparametrized H1 Weights obtained using only half of the dataset.*

Chapter 5

Summary & Outlook

We have analyzed a sample of simulated Z+jet events in view of using such a sample to calibrate the jet reconstruction software during the initial LHC calibration run at low luminosity. The effect of a large number of cuts was examined in order to determine a way to filter out Z+jet events suitable for calibration purposes, whilst keeping a reasonably large fraction of the events to provide sufficient statistics for the procedure. An efficient cut has been proposed which keeps 33% of the initial events and reduces the imbalance between the transverse momentum of the Z-particle and the highest p_t jet to less than one percent. The existing jet reconstruction software performs reasonably well on the events selected by this cut, however the Atlas design linearity and resolution for jet reconstruction have not been achieved yet. It was observed that the linearity and resolution become much better when they are measured with respect to the opposing parton. This fact is very interesting and might indicate a problem with the determination of the MCTruth jets. This effect has to be looked into in further detail.

An H1 calibration procedure was performed based upon the assumed balance between p_t^Z and p_t^{jet} in order to evaluate the resolution and linearity after calibration with the Z+jet sample. After imposing the same parametrization as is in use now on the H1-weights, the performance of the jet algorithms with the new set of H1-weights is found to be comparable to the performance with the existing weights. The raw values of the new weights also do not deviate too much from the current weights. This is a promising result, since the current weights were determined using a dijet sample with much higher statistics than the Z+jet sample under investigation. After one year of data taking at the LHC the available statistics will be approximately three times higher than the statistics used in this analysis, which will further improve the obtained resolutions and linearities.

Due to a bug in the reconstruction software this work only considered events in the $\eta \leq 2.3$ region. Further research in this direction should include the full η region, so that the forward and end-cap calorimeter calibration can be fully examined. The $p_t^Z < 100$ GeV/c momentum range should also be examined in greater detail. In particular, an improved method of matching the right jet with the parton opposite to Z-particle will need to be found for low momenta. One could for example consider the jet for which $\Delta\phi(\mathrm{Jet}, \mathrm{Z})$ is minimal. The bisector method discussed in Section 4.3.1 should also be examined in greater detail to improve the resolution estimation. Finally, one should split the Z-jet sample into b-jets and non b-quark jets and perform separate analyses on each of these parts, since b-jets play a very important role in the Atlas physics program.

At the moment efforts are underway to further improve the jet reconstruction resolution. Currently, the H1-weighting scheme is based only upon the total energy in different detector cells. Since different cells in the same H1-region may have different volumes, one may expect that the H1-weighting will yield better results if one corrects for this volume effect and considers the energy density in the cells. This feature is scheduled to be implemented in the near future. The modifications to the $K_\perp$ jet-finding algorithm presented in this thesis may reduce the running time of the jet reconstruction phase considerably. This can be exploited by increasing the number of energy regions that are used as an input to the jet-finding phase. The preclustering phase can thus be slackened and it is reasonable to assume that an input with a higher resolution will in general yield better results.

Bibliography

[1] ALEPH Collaboration, D. Buskulic et al., *Z. Phys.* C55 (1992) p209.

[2] D. Amati, R. Petronzio and G. Veneziano, *Nuclear Physics B* 140 (1978) p54, /em Nuclear Physics B /em 146 (1978) p29; R.K. Ellis, H. Georgi, M. Machacek and G.G. Ross, *Physical Letters* 78B (1978) p281, *Nuclear Physics B* 152 (1979) p285; C.T. Sachrajda, *Physical Letters* 73B (1978) p185, *Physical Letters* 76B (1978) p100.

[3] D. Amati and G. Veneziano, *Physical Letters* 83B (1979) p87; A. Bassetto, M. Ciafaloni and G. Marchesini, *Physical Letters* 83B (1979) p207; G. Marchesini, L. Trentadue and G. Veneziano, *Nuclear Physics B* 181 (1980) p335; Ya.I. Azimov, Yu.L. Dokshitzer, V.A. Khoze and S.I. Troyan, *Physical Letters* 165B (1985) p147, *Zeit. Phys.* C27 (1985) p65.

[4] B. Andersson et al., *Physics Reports 97, numbers 2 and 3* (1983) p159.

[5] ATLAS Collaboration, "Technical Proposal for a General Purpose pp Experiment at the Large Hadron Collider at CERN",*CERN/LHCC/94-43, LHCC/P2* (1994).

[6] ATLAS Collaboration, "ATLAS Computing Technical Proposal", *CERN/LHCC/96-43* (1996).

[7] ATLAS Collaboration, "Liquid Argon Calorimeter Technical Design Report", *CERN/LHCC/96-41* (1996).

[8] ATLAS Collaboration, "Tile Calorimeter Technical Design Report", *CERN/LHCC/96-42* (1996).

[9] ATLAS Collaboration, "Detector and Physics Performance Technical Design Report", *CERN/LHCC/99-14* (1999).

[10] ATLAS Collaboration, Jet Reconstruction Design document, *to appear.*

[11] P. Bagnaia et al. (UA2 Collaboration), *Physics Letters* 144B (1999) p283.

[12] F.A. Berends et al., *Nuclear Physics B* 357 (1991) p32.

[13] V. Boisvert et al., “Final Report of the ATLAS Reconstruction Task Force”, *CERN ATL-SOFT-2003-0101* (2003).

[14] N. Brown and W.J. Stirling, *Physical Letters* 252B (1990) p657; *Zeit. Phys.* C53 (1992) p629, S. Bethke, Z. Kunszt, D.E. Soper and W.J. Stirling, *Nuclear Physics B* 370 (1992) p310.

[15] R. Brown, “Calendar queue, a fast O(1) priority queue implementation for the simulation event set problem”, *Communications of the ACM,* 31(10) (1998) p1220-1227.

[16] R. Brun et al., “GEANT 3”, *CERN/DD/EE/81-1* (1996).

[17] S. Catani, Yu.L. Dokshitzer, M. Olsson, G. Turnock and B.R. Webber, *Physical Letters* B269 (1991) p432.

[18] S. Catani, Yu.L. Dokshitzer and B.R. Webber, *Physical Letters* 285B (1992) p291.

[19] S. Catani, Yu.L. Dokshitzer, M.H. Seymour and B.R. Webber, “Longitudinally-Invariant $k_\perp$-Clustering Algorithms for Hadron-Hadron Collisions”, *Nuclear Physics B* 406 (1993) p187.

[20] T.H. Cormen, C.E. Leiserson and R.L. Rivest, “Introduction to Algorithms”, *McGraw-Hill* (1990).

[21] DELPHI Collaboration, P. Abreu et al., *Physical Letters* B347 (1995) p447.

[22] S.D. Ellis, Z. Kunszt and D.E. Soper, *Physical Review Letters* 62 (1989) p726, *Physical Review D* 40 (1989) p2188, *Physical Review Letters* 64 (1990) p2121, *Physical Review Letters* 69 (1992) p1496; Z. Kunszt and D.E. Soper, *Physical Review D* 46 (1992) p192.

[23] S. Ellis and D. Soper, “Successive Combination Jet Algorithm For Hadron Collisions”, *CERN-TH-6860-93* (1993).

[24] K.B. Erickson, R.E. Ladner and A. LaMarca, “Optimizing Static Calendar Queues”, *IEEE Symposium on Foundations of Computer Science* (1998).

[25] W.T. Giele, E.W.N. Glover and D.A. Kosower, *preprint CERN-TH-6750-92* (1992).

[26] H1 Collaboration, S. Aid et al., *Nuclear Physics B* 445 (1995) p3.

[27] JADE Collaboration, W. Bartel et al., *Zeit. Phys.* C33 (1986) p23; S. Bethke et al., *Physical Letters* 213B (1988) 235.

[28] L3 Collaboration, B. Adeva et al., *Physical Letters* B259 (1991) p199.

[29] R. Lefévre and Claudio Santoni, “In Situ Determination of the scale and resolution of the Jet Energy Measurements using Z^0+jet events”, *CERN ATL-COM-PHYS-2002-026* (2002).

[30] LHC White Book, *CERN/AC/93-03* (1993).

[31] LHC Conceptual Design Report, *CERN/AC/95-05* (1995).

[32] OPAL Collaboration, M. Z. Akrawy et al., *Physical Letters* B247 (1990) p617.

[33] I.C. Park, ”A new Clustering Algorithm: Mulguisin”, *CERN ATL-COM-PHYS-99-055* (1999).

[34] R. Rönngren and R. Ayani, “A Comparative Study of Parallel and Sequential Priority Queue Algorithms”, *ACM Transactions on Modelling and Computer Simulation,* Vol. 7, No. 2 (1997) p157-209.

[35] T. Sjöstrand, ”PYTHIA 5.7 and JETSET 7.4 physics and manual”, *hep-ph/9508391, CERN-TH-7112-93, LU-TP-95-20* (1994).

[36] G. Sterman and S. Weinberg, *Physical Review Letters* 39 (1977) p1436.

[37] K.L. Tan and L.J. Thng, “Snoopy Calendar Queue”, *Proceedings of the 2000 Winter Simulation Conference* (2000).

[38] TASSO Collaboration, W. Braunschweig et al., *Z. Phys.* C47 (1990) p187.

[39] ZEUS Collaboration, M. Derrick et al., *Z. Phys.* C67 (1995) p93.