ATLAS Detector Monitoring with Jets

Master's Degree Thesis in Experimental Particle Physics

by

Kent Olav Skjei

Department of Physics and Technology
University of Bergen
Norway

October 2009

Acknowledgements

First I would like to thank my supervisor professor Anna Lipniacka for all help during work with my thesis. I am most grateful for her having offered me the opportunity to work on this project. It has been extremely interesting and educating.

Thomas Burgess deserve my deepest gratitude. He has been a key person as co-author of the program DiJet, he has provided me with all necessary ROOT-script and data sets used in the analyses, has helped me setting up the different software frameworks required, as well as helping me get the necessary computer accounts, and probably many, many other things which I have forgotten now. The project would not have been possible without Thomas' help. Although it may not always have been apparent, due to all stress and excitement, etc., I am truely grateful for your help, Thomas.

Konstantinos Alex Kastanas is another person who have been of invaluable help when it comes to setting up Athena and providing me with, and reconstructing data.

Also Arshak Tonoyan must be mentioned here for explaining me the W mass reconstruction.

I would also like to thank my parents, who have always been there for me. I would never have been where I am today without their support. My father's work finding mistakes in the text is also worth mentioning.

And last, but not least; Annelin, for being there for me, caring for me and making sure that I get enough food and sleep.

Without the help from these people this thesis would never have happened.

-Kent Olav Skjei

Abstract

The purpose of this thesis was to provide a high level software framework for detecting large problems with the ATLAS detector at the LHC and estimate the jet resolution from the first data. The result was the program DiJet, written in C++ by Kent Olav Skjei and Thomas Burgess. DiJet can be run both as a program within Athena (ATLAS software framework) or as a script in AthenaROOTAccess.

We have looked at and compared several variables for back-to-back jets using data sets with Monte Carlo dijets, Monte Carlo top and the simulated FDR2 data. Dijets were chosen due to the high cross section expected at the LHC. It became clear that for real data, one will need to review the dijet selection criteria in order to supress multijet background.

DiJet also reconstruct the W mass as a method for estimating hadron resolution. This method requires more data than methods for back-to-back jets do, and was not useful for FDR2 data. However, based on Monte Carlo dijet and top events we were able to compare jet resolution estimates for the methods for back-to-back jets and the method based on the W mass. As expected, these variables seem to be related.

In connection with the back-to-back jets we saw a certain η dependence of transverse energy.

The last variables we looked at were missing transverse energy and the vectorial sum over jet momenta. From the latter results, it became apparent that missing transverse energy contains more than just the sum over jet momenta. And it also became clear that the jet reconstruction algorithms themselves create some false missing transverse energy.

The position of bad eta and phi regions were found by looking at profile plots of these variables. Based on this we decided to only look at the region $-3 \leq |\eta| \leq 3$ due to low statistics in high $|\eta|$.

We have seen that the methods for estimating jet resolution in this thesis gives half the value of the mass resolution of particles decaying into jets.

Based on the results, we seem to have found an asymmetry in ϕ in the results from the jet reconstruction. We did not have the opportunity to investigate the cause of this asymmetry, due to lack of time.

The original objective of this thesis was to look at the Express Stream. This turned out not to be possible due to the decision of not making the Express Stream generally available. In addition, this thesis has been affected by the delay of the LHC schedule, changes in the policy concerning streaming and evolution of the ATLAS software frameworks. Upgrades on the local

computer cluster used for data analysis have also caused considerable loss of time.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Year 2010, the LHC at CERN (European Organization for Nuclear Research) will start running, and hopefully collide protons at $7 - 10$TeV. This will probably mark the beginning of a new era in particle physics. Physisists will be able to dig deeper into the mysteries surrounding matter and our early universe than ever before. Hopefully our theories on origion of mass will be confirmed, and the matter - antimatter asymmetry and thereby the existence of our universe might get a plausible explanation. Dark matter, which is believed to be essential for the existence and behaviour of galaxies, could possibly be produced in the laboratories. A new symmerty, called supersymmetry, could be experimentally observed.

CERN has constucted the Large Hadron Collider (LHC), with four large facilities connected to it. One of these is ATLAS. ATLAS will study events from proton - proton collisions, and hopefully contribute to the discovery of the Higgs boson, supersymmetric particles and extra dimensions. Such tasks will require the detector to handle, i.e. be collecting and processing, enormous amounts of data. This thesis will study a small part of that process, namely jet reconstruction performance and resolution.

The thesis will first give a short introduction to the relevant parts of the standard model of particle physics. There will be a chapter on CERN, the LHC and ATLAS and its hardware, and further an introduction to jet reconstruction. The concluding sections will present the computer program DiJet that was specially written for this thesis, and its suggestion for estimating the jet resolution of ATLAS, as well as its ability to spot and make coarse removal of the noisy tower in the original FDR2 (Full Dress Rehersal no.2) data. Further will any connection between resolution the W mass and the jet resolution be discussed.

Chapter 2

Physics Processes at the LHC

2.1 Introduction to the Standard Model and Beyond

The standard model (SM) is the currently accepted theory of particle physics. It has been tested to great accuracy in laboratories all around the world. In the energy range tested until this day, no indications of the model being wrong has been observed. However, a vital particle for the SM has yet to be observed, namely the Higgs boson.

Mathematically, the SM is a field theory of quanta. It is the combination of the groups $U(1)$ (Quantum Electrodynamics or QED), $SU(2)$ (Weak theory) and $SU(3)$ (Quantum Chromodynamics or QCD) into the the group $SU(3) \times SU(2) \times U(1)$.

The SM seperates particles into two groups. One group consists of particles with half-numbered spins, fermions, and the other of particles with whole-numbered spins, bosons. The fermions are the leptons (electrons, muons, taus and neutrinos) and quarks, and consists of three generations (see fig 2.1). The first generation consists of the up- and down-quarks, the electron and the electron neutrino and their respective antiparticles. The second generation consists of the charm- and strange-quarks as well as the muon and the muon neutrino, and their antiparticles. The third generation consists of the top- and bottom-quarks, the tau and tau neutrino, and the respective antiparticles of these particles.

The fermions interacts through forces mediated by the bosons, namely the photons (γ), $W^{\pm}$, Z^0 and gluons (g).

QED, belonging to the gauge group $U(1)$, has only one gauge boson, namely the photon. The photon is thereby responsible for all electromagnetic interactions, and it interacts with all charged particles. The electromagnetic

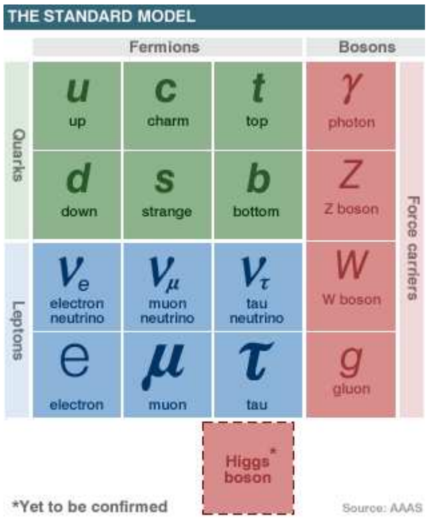

Figure 2.1: The particles and generations of the Standard Model[53]

force binds electrons and nuclei into atoms and molecules, and thereby controls the chemistry and physics of materials.

The $SU(2)$ weak theory has 3 gauge bosons, i.e. the $W^{\pm}$ and Z^0. These particles interacts with all particles carrying weak charge (all fermions), and they are responsible for processes like β decay.

The weak bosons are currently the only observed massive gauge bosons. In the current model their masses originate from the spontaneous symmetry breaking of the $SU(2) \times U(1)$ symmetry (electroweak theory), also known as the Higgs mechanism. The idea of the Higgs mechanism is that there exists a Higgs field and its corresponding quantum, the Higgs boson. Particles interacting with the Higgs boson will we percieve as massive.

Finally, there are eight gluons from the $SU(3)$ symmetry group, or QCD. Gluons mediates the strong force and interacts with quarks. The strong force binds quarks together in hadrons, and binds nucleons in nuclei. Hadrons are further seperated into baryons, containing partons (three valence quarks, sea quarks and gluons), and mesons, containing two valence quarks.

However successful the model has been until now, it does have limitations. The SM is, for example, not able to describe gravitation, it doesn't include dark matter, it assumes massless neutrinos, it gives rise to the hierarchy problem, it doesn't solve the questions surrounding the matter-antimatter asymmetry and it requires 19 numerical constants that are found ad hoc. In addition, the Higgs boson, which is needed to make the SM mathematically consistent, is per the writing of this thesis unobserved.

At the energies reached in LHC the collision between protons is really

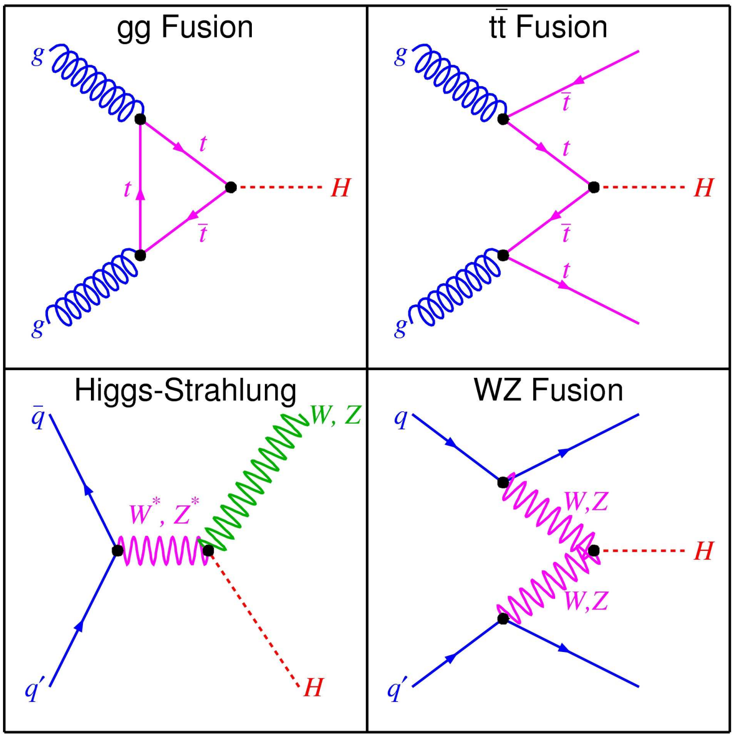

Figure 2.2: Some Feynman diagrams for Higgs production[54]

a collision between partons, creating new resonances. Some examples of Feynman diagrams for the processes relavant at the LHC are shown in fig. 2.2, 2.3 and 2.4.

2.2 QCD Processes and Hadronization

Two important features of QCD are asymptotic freedom and confinement. Contrary to electroweak theory, the QCD coupling constant decreases at high energies and increases at low energies. This is referred to as asymptotic freedom, which is the ability of interactions to become arbitrarily weak at short distances. At longer distances, however, one observes color confinement, meaning until now no free quarks or gluons have been found. This is because the QCD coupling constant increases with the distance. When the coupling constant gets large, that leads to nonperturbative processes. Perturbative processes are processes that can be described using perturbation theory. The essence of perturbation theory is to start with a mathematical

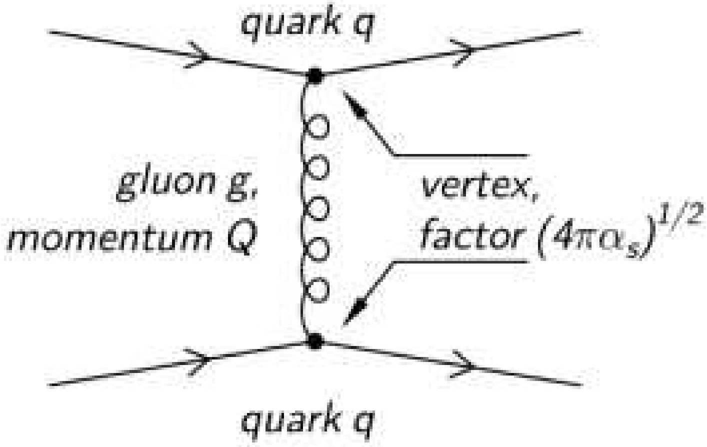

Figure 2.3: Feynman diagram for quark scattering through gluon exchange[55]

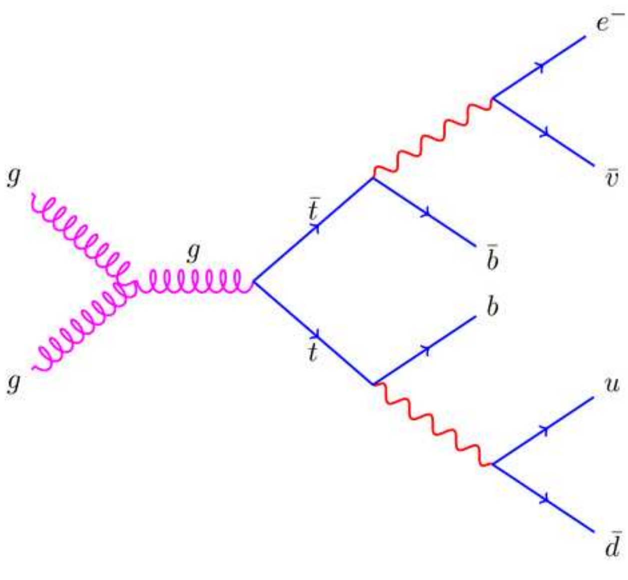

Figure 2.4: Feynman diagram showing a possible production of $t\bar{t}$ decaying to two W's and two b jets[56]. This decay mode for the tops was used in this thesis

description that has a known solution, and then introduce a small perturbation. In QCD, such an approximation is valid when the coupling constant is small, i.e. for short distances. Color is the "charge" of the strong force, and can be red, blue or green. One specific gluon carries both one color and one anticolor. This is the reason why gluons can couple directly to other gluons. Quarks only carry either one color or one anticolor. Both theory and experiment suggests that hadrons are neutral in terms of color. Due to vacuum polarization, the energy needed to seperate two quarks increases linearily with the distance. This has the consequence that it becomes energetically more advantageous to create a new quark pair instead of seperating the two original quarks. The new quark pairs then form hadrons through hadronization. Hadronic interactions therefore involve jet formation.

2.2.1 Dijets

The LHC will collide two beams of protons at high energies ($7-10$TeV). At such high energies it will be the partons that are interacting, and one gets parton-parton scattering between two quarks, two antiquarks, one quark and one antiquark, one quark and one gluon or two gluons. Approximately 80% of the collisions will be between gluons. This will lead to a new final state which most often consists of hadrons. The final state is dependent on the physical conservation laws. Parton-parton scattering can then give four or more jets; jets from partons radiating off the scattered partons, one jet from each of the scattered partons heading into the detector and two for the spectator partons. The latter move along the beam axis. This occurs when the center of mass energy is approximately 5GeV per parton[26]. To avoid confusion with the beam jets, one usually demand that the other two jets have a direction with a certain angle with respect to the beam axis. In other words that they have a transverse energy larger than approximately 5GeV. Such jets are referred to as dijets. In addition one might have other jets, e.g. from gluon radiation.

2.3 Top and W

LHC will be the first collider $t\bar{t}$ factory. Top quark physics will therefore be another important aspect of the standard model to be studied at the LHC. There are several reasons for this. One is that the top quark signal will be a clear one, and easily seperated from the background, even in case of an imperfectly calibrated detector. This means that the top mass will provide insight on the detector performance at an early stage. Consequently, top quark events allows detector calibration and will provide a certain under-

Table 2.1: Main Decay Modes of the W^+ Particle[27]

The W^- decay modes are charge conjugates of the modes above.
l indicates each type of lepton (e, μ and tau), not sum over them.
Γ_{10} represents the width for the decay of the W boson into a charged particle with momentum below detectability, $p <$ 200MeV.

Γ_i	Mode	Fraction (Γ_i/Γ	p Confidence level (MeV/c)
Γ_1	$e^+\nu$	$(10.75 \pm 0.13) \times 10^{-2}$	40199
Γ_2	$\mu^+\nu$	$(10.57 \pm 0.15) \times 10^{-2}$	40199
Γ_3	$\tau^+\nu$	$(11.25 \pm 0.20) \times 10^{-2}$	40179
Γ_4	Hadrons	$(67.60 \pm 0.27) \times 10^{-2}$	-

standing of the detectors jet energy scale of light jets, as well as of b-tagging, since b quarks is an ordinary decay product of the t quark. Top quark physics will also be important when searching for new physics, both as a background and as a decay mode for currently unobserved particles.

The particles W^+ and W^- are the charged bosons of the weak theory and a decay product of the top quark (see fig. 2.4 for the decay mode used in this thesis). Toghether with the Z particle, they are the only known fundamental massive bosons. The W mass is presently measured[27] to be 80.398 ± 0.025GeV and has a full width of $\Gamma = 2.141 \pm 0.041$GeV. It therefore decays very quickly and can only be seen indirectly in the detectory from it's decay products. $W^\pm$ will be a part of many important decay modes in ATLAS, and one hopes to determine it's mass with a much higher accuracy than what has presently been achieved.

The W^+ can decay into[27] $l^+\nu$, i.e. $e^+\nu_e$, $\mu^+\nu_e$ or $\tau^+\nu_e$, or hadronically into $u\bar{d}$ or $c\bar{s}$. The main branching fractions are given in Table 2.1. The decay modes of W^- are just charge conjugates of the mentioned modes.

2.4 Missing Transverse Energy

In the LHC it is the partons that are colliding. We can not know the exact momentum in the beam direction for partons. However, in the transverse plane, which is orthogonal to the beam axis, the sum of particle momenta should be 0. However, due to the presence of neutrinos, we expect a certain amount of missing transverse energy. Also the jet reconstruction algorithms will lead to a certain amount of missing E_{T}, due to outflow from the jet cones. It is important to have an idea of how much these effects will contribute, since missing transverse energy can be an indication of supersymmetric particles being created in the event. Supersymmetric particles are among the discoveries one hopes to make at the LHC.

2.5 B Physics

B Physics is the study of the physics related to the processes containing bottom quarks. The ATLAS B physics program will conduct detailed tests of the standard model which will hopefully provide indications of the existence of new physics, and put constraints on non-SM physics.

The B physics program in ATLAS will provide

- High precision tests of QCD predictions for cross sections of beauty and charmed hadrons[9]
- Alignment and calibration of the trigger tracking and muon systems[9]
- Tests of both perturbative and non-perturbative predictions of QCD[9]
- Study of quarkonia states that are part of the decay modes of heavier resonances[9]
- Study of the $b\bar{b}$ state, the largest background for many of the events expected at the LHC[10]
- Detector performance checks[11]
- Give an understanding of some flavour tagging methods[11]
- Study of flavour changing neutral currents, which is forbidden in the standard model[12]
- Precise determination of the weak mixing angle induced by CP violation[13]

2.6 Higgs

Detection of the Higgs[14] boson is the primary goal of LHC and the particle physics experiments connected to it. SM predicts the Higgs boson to be a neutral scalar. There are also other possibilities concerning the Higgs particle. One extension of SM, called the Minimal Supersymmetric Standard Model (MSSM)(see next chapter), predicts five Higgs bosons; three neutral and two charged $H^{\pm}$. The neutral ones are two CP-even (h and H) and one CP-odd (A). For the neutral SM Higgs, the channels relevant at LHC are

- $pp, gg \to H \to \gamma\gamma$
- $pp, gg \to H \to ZZ^{(*)} \to 4l(l = e, \mu)$

- $pp, gg \to qqH \to qq\tau^+\tau^-$
- $pp, gg \to H \to W^+W^- \to l\nu l\nu, l\nu qq$
- $pp, gg \to t\bar{t}H \to t\bar{t}b\bar{b}$
- $pp, gg \to t\bar{t}H \to t\bar{t}W^+W^-$
- $pp, gg \to ZH \to l^+l^-W^+W^-$

Looking at the range 100GeV−1000GeV, $H \to b\bar{b}$, $H \to \tau^+\tau^-$, $H \to \gamma\gamma$ is mostly relevant in the Higgs mass range 100GeV $\leq M_H \leq$ 200GeV. $H \to t\bar{t}$ is relevant in the range 250GeV $\leq M_H \leq$ 1000GeV. $H \to WW(*)$ and $H \to ZZ^{(*)}$ is relevant in the entire mass range. These are the most relevant decay modes.

The neutral MSSM Higgs can be produced either from direct production or associated production. In direct production, the neutral Higgs is produced from a fermion loop originating from the interaction of two gluons. In associated production Higgs is produced either from the process $gg \to \phi b\bar{b}$, $b\bar{b} \to \phi$, $gb \to b\phi$ or $q\bar{q} \to g \to b\bar{b} \to b\bar{b}\phi$.

For charged MSSM Higgs production below the top quark mass is $t \to H^+b$ the dominant process. $H^+ \to \tau^+\nu$ dominates among the decay modes. In case of a charged Higgs mass above the top quark, the process $g\bar{b} \to \bar{t}H^+$ becomes the most important one for production. $H^+ \to t\bar{b}$ will then be the most important decay mode, although $H^+ \to \tau^+\nu$ still contributes. Around the top mass, $gg \to \bar{t}bH^+$ is important. LHC being a $t\bar{t}$ factory, light Higgs might be produced through the process $q\bar{q}, gg \to t\bar{t} \to \bar{t}bH^+$.

2.7 Supersymmetry

Supersymmetry (SUSY) is a possible extension to the standard model. The theory introduces a new symmetry, namely a symmetry between fermions and bosons. This gives rise to new, until now unobserved particles. If supersymmetric particles exists, it is assumed that supersymmetric events will be characterized[15] by several high-momentum jets as well as missing transverse energy. Electrons, muons and taus will also be present in a large amount of the events. The most relevant background to SUSY events are $t\bar{t}$, $W+$ jets, $Z+$ jets, jets from QCD processes, and diboson production, i.e. WW, ZZ and WZ[16].

2.8 Exotic Processes

Many extensions of SM predicts the existence of a new heavy state decaying into two leptons. Among these extensions we find grand unified theories (GUTs), Technicolor, little Higgs models and models including extra dimensions[17]. The advantage of states decaying into two leptons is the simplicity of the final state.

The ability to investigate states consisting of one lepton and missing transverse energy, i.e. a neutrino, is important for the ability to reconstruct gauge bosons not predicted by SM.

Processes involving final states with two leptons, two jets and no missing transverse energy is another area of interest. Such states are predicted in models like leptoquarks, i.e. hypothetical bosons carrying both quark and lepton quantum numbers, and Left-Right Symmetry, addressing both the non-zero masses of the three known left-handed neutrinos and baryogenesis.

If there doesn't exist a light Higgs particle, electroweak symmetry breaking can most easily be studied looking at vector boson scattering at high mass, for example WW scalar and vector resonances, WZ vector resonances and ZZ scalar resonances.

Black holes is another possibility of what can be observed in ATLAS. Such events are characterized by a large number of final state particles with high transverse momentum.

2.9 Interaction of Particles with Matter

The interaction of particles with matter is the foundation for particle detectors.

When discussing the interaction of particles with matter[28], one seperates the interactions into two groups:

1. Interactions of charged particles
2. Interactions of neutral particles

Charged particles can experience several types of interactions when traversing matter. They are:

- Deflection through elastic scattering with nuclei
- Excitation and ionization of a medium
- Cherenkov radiation

- Transition radiation
- Bremsstrahlung
- Nuclear interactions

Cherenkov radiation is the emission of a real photon if the velocity of the particle exceeds the phase velocitiy of light in the medium.

If the phase velocity of light in a certain medium is larger than the velocity of light in vacuum, emission of Cherenkov radiation is still possible if the medium contains discontinuities. This latter effect is referred to as transition radiation.

Bremsstrahlung is the emission of photons when charged particles are accelerated or decellerated. This effect is only interesting for electrons.

The cross section of nuclear interactions is generally small.

Particles traversing matter will due to the above mentioned interactions only move a certain length. The length they move is related to the radiation length X_0. If we consider electrons with initial energy E_0, its mean energy $\langle E \rangle$ after having traversed a certain mass thickness X is

$$\langle E \rangle = E_0 e^{-}\frac{X}{X_0} \tag{2.1}$$

That means that the mean energy of an electron beam that traverses a radiation length is reduced by a factor e from Bremsstrahlung. Bremsstrahlung is the emission of photons from electrons.

In this connection, it's also interesting to define the interaction length[29], or mean free path of a process:

$$\lambda(E) = \left(\sum_i \frac{N_0 \rho \omega}{A_i} \cdot \sigma \right)^{-1} \tag{2.2}$$

σ is the cross-section, N_0 is Avogadro's number, A_i is the mass of a mole of the ith element of the material, ω_i is the proportion of mass of the ith element and ρ is the density of the material.

Among the neutral particles are the photon, neutrons and neutrinos. The photons can be detected through three effects:

1. Photoelectric effect
2. Compton effect
3. Creation of an electron-positron pair

The photoelectric effect is the ionization of an atom with the emission of an electron. The Compton effect is the electron-photon scattering. Pair creation is the creation of an electron-positron pair in the presence of a nucleus. This is the dominant effect for energies above 5GeV, and leads to the creation of electromagnetic cascades. Electrons and positrons from pair creation are affected by bremsstrahlung, which means that they emit photons. These secondary photons can again create an electron-positron pair, leading to a cascade of particles.

If we consider photons with an intensity I_0 that traverses a material, the intensity emerging from the layer is

$$I(X) = I_0 e^{-\mu x} = I_0 e^{-\frac{\mu}{\rho}} \tag{2.3}$$

Here x is the thickness of the layer. $X = \rho x$ is the mass thickness. μ is the linear absorption coefficient. The mass absorption coefficient is defined as $\frac{\mu}{\rho}$. For high photon energies, the mass absorption coefficient for pair creation is $\frac{\mu}{\rho} = \frac{\sigma_p N_0}{A}$.

For high photon energies the mass absorption coefficient reaches an asymptotic value

$$\frac{\mu_0}{\rho} = \frac{7}{9}\frac{1}{X_0} \tag{2.4}$$

where X_0 is the rediation length

$$X_0 = \frac{7}{9}\frac{\rho}{\mu_0} \tag{2.5}$$

The radiation length is related to the typical length a photon will traverse matter before it transforms into an electron-positron pair.

Neutrons are detected through their strong interaction with nuclei. Such interactions cause the creation of charged secondary particles.

Neutrinos can be only be detected indirectly through their weak interaction with nuclei or electrons.

Chapter 3

The Experiment

3.1 CERN

The European Organization for Nuclear Research, CERN, is the largest particle physics laboratory in the world. It was established in 1954 on the border between France and Switzerland, just outside Geneva. CERN currently has 20 European membership countries, provides work for approximately 2600 full-time employees, as well as 8000 physicists and engineers from 580 Universities and research facilities, and representing 80 different nationalities.

3.1.1 LHC

The Large Hadron Collider (LHC) is a hadron storage ring at CERN, that will provide proton-proton collisions, as well as heavy ion collisions using lead nuclei. It was constructed in the old Large Electron Positron collider(LEP) tunnel and thereby has a circumference of 27 km, currently making it the largest particle accelerator in the world, and is designed to provide the highest energy particle collisions currently available.

The main purpose for the construction of the LHC was to search for the Higgs boson. Due to the high energies that should be reached, finding supersymmetric particles and signatures of extra dimensions may also be a possibility. More precise measurements of standard model parameters is another important feature of the LHC. In addition, heavy ion collisions will allow the study of strongly interacting matter at an extremely high energy density and perhaps also the hypothetical quark-gluon plasma.

The design luminosity of LHC in the context of particle physics is $10^{34}\text{cm}^{-2}\text{s}^{-1}$[18], but will start out a luminosity of $10^{31}\text{cm}^{-2}\text{s}^{-1}$[18]. It was originally planned to collide bunches of up to 10^{11} protons at an energy of 14TeV 40 million times per second. It has however recently become clear

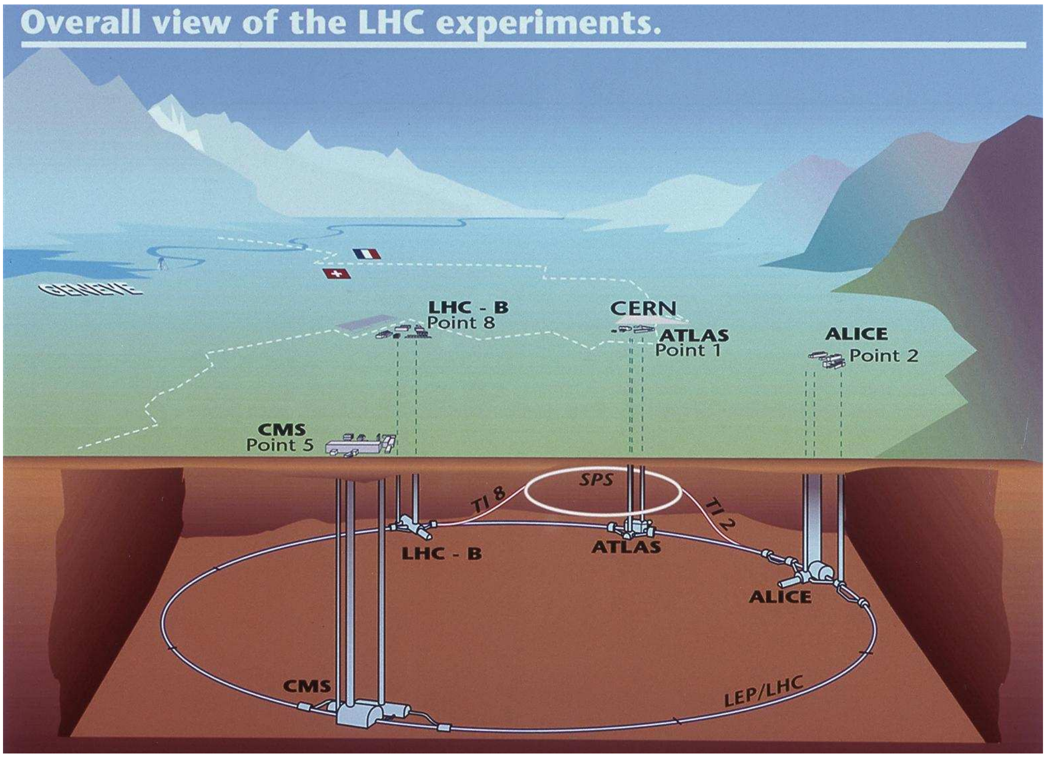

Figure 3.1: Large Hadron Collider and it's attached experiments[40]

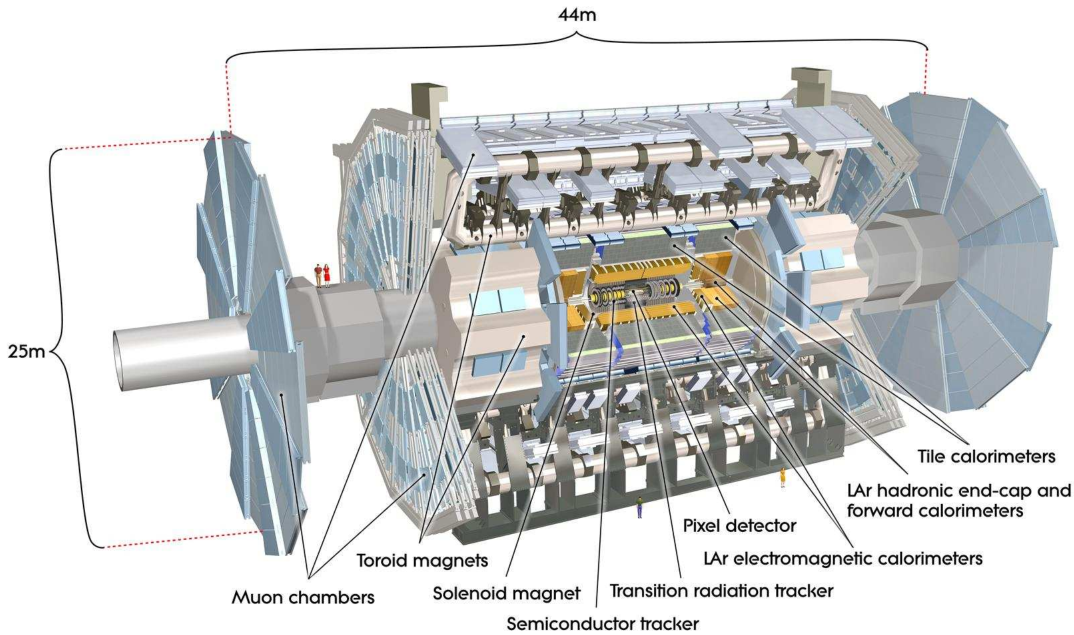

Figure 3.2: ATLAS Detector[31]

that such a high energy will be hard to achieve. The current plan is to reach 7TeV in the center-of-mass system by end of 2009, and 10TeV in 2010[42].

The design luminosity for heavy ion collisions is $10^{27}\text{cm}^{-2}\text{s}^{-1}$. These will reach an energy of 5.5TeV in the center of mass frame.

There are four detectors connected to the LHC; two general purpose detectors, one detector focusing on B physics, and one focusing on nuclear physics. The two general purpose detectors are ATLAS and CMS. LHCb is focusing on B physics, and Alice is focusing on heavy ion collisions.

3.2 The ATLAS Detector

ATLAS (A Toroidal LHC ApparatuS) (see fig. 3.2), being one of the general purpose detectors connected with LHC is designed to search for the Higgs boson. It will also be investigating CP violations, supersymmtry, extra dimensions and put more stringent tests and constraints on the standard model.

ATLAS consists of four major components, namely the inner tracker for momentum measurments, the calorimeter for measuring energy, the muon spectrometer for muon identification and measurment and the magnet system. At design luminosity the detector system will need to have the abillity to handle 1 collision every 25ns[27] within $|\eta| < 2.5$.

3.2.1 General Outline of the Detector

All modern general purpose detectors, including ATLAS, are constructed from a basic idea (see fig. 3.3). I.e., they consists of four main parts:

1. Magnet system
2. Tracking detectors
3. Calorimeter
4. Muon spectrometer

The purpose of the magnet system is to bend the trajectory of charged particles. Their trajectory in a magnetic field $\vec{B}$, where $\vec{B}$ is measured in tesla, then becomes a helix with a radius of curvature R defined by the relation[27]

$$p \cos \lambda = 0.3zBR \tag{3.1}$$

where p is the particle's momentum in (GeV/c), z is the particle's proton number and λ the pitch angle. Knowledge about a particle's trajectory can thereby give information about a particle's mass and momentum.

The magnet system and its ability to bend the trajectory of charged particles is used by the tracking detectors. This is the innermost part of a detector. These detector parts are used to measure the tracks of the charged particles passing through. Ideally, the trackers should not affect a particle's trajectories, as the calorimeters does. Trackers cannot detect neutral particles.

Outside the trackers are the calorimeters. The first ones being the electromagnetic calorimeters, and outside of these, the hadronic calorimeters. These measure the energy of traversing particles by completely stopping them.

And finally we have the muon spectrometer. The purpose of the muon spectrometer is to identify muons. Most of the muons pass straight through the detector.

3.2.2 Coordinates

ATLAS uses a right handed coordinate system with an x-, a y- and a z-axis. The x-axis points towards the center of the LHC ring, the y-axis upwards in the vertical direction and the z-axis along the beam direction[19]. The origin

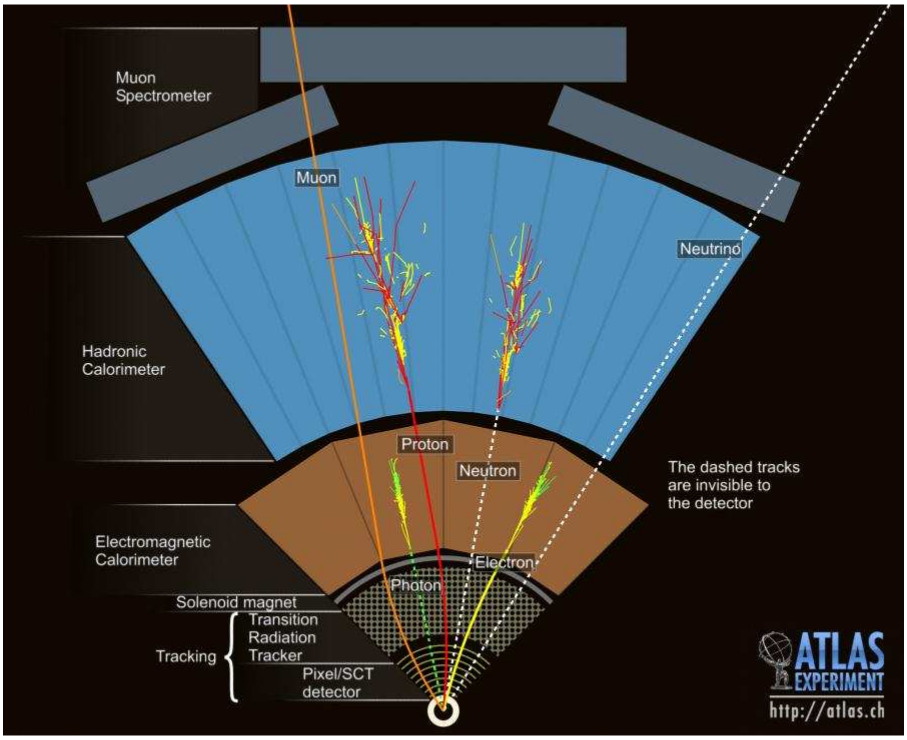

Figure 3.3: Interaction of particles in ATLAS[32]

lies at the nominal interaction point. Transverse momentum, transverse energy and missing transverse energy are usually defined in the $x-y$ plane[2]. The detector has two sides: Side-A with positive z and Side-B with negative z[2]

There are also other coordinates used in ATLAS. These are defined by equations 3.2, 3.3, 3.4 and 3.5. η is referred to as pseudorapidity and is related to the polar angle θ from the beam axis. The rapidity 3.2 is used in the case of massive objects, such as jets[2]. ϕ is the azimuthal angle measured around the beam axis[20],[21]. ΔR is distance in the pseudorapidity-azimuthal angle space[2].

$$y = -\frac{1}{2}\ln\left(\frac{E+p_z}{E-p_z}\right) \tag{3.2}$$

$$\eta = -\frac{1}{2}\ln\left(\frac{p+p_z}{p-p_z}\right) = -\ln\left[\tan\left(\frac{\theta}{2}\right)\right] \tag{3.3}$$

$$\phi = \arctan\left(\frac{p_y}{p_x}\right) \tag{3.4}$$

$$\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2} \tag{3.5}$$

Using these coordinates, one can define the transverse energy as[26]

$$E_{\mathrm{T}} = E\sin(\theta) \tag{3.6}$$

3.2.3 Inner Detector

The inner detector (ID)[19][3] consists of

- pixel detectors
- semiconducting trackers (SCT)
- transition radiation trackers (TRT)

The semiconductor trackers are made from silicon microstrips.

The Pixel detector and the SCT will provide high momentum and vertex resolution for $|\eta| < 2.5$, which will be needed at the design luminosity. In the barrel region pixel and SCT have been arranged as concentric cylinders around the beam. In the end-regions they are placed on circular disks perpendicular to the beam. Silicon pixel sensors have been used around the collision point(see fig. 3.5). The straw tubes of the TRT will make possible the necessary tracking over a larger area(see fig. 3.4 and 3.6). Since the ID

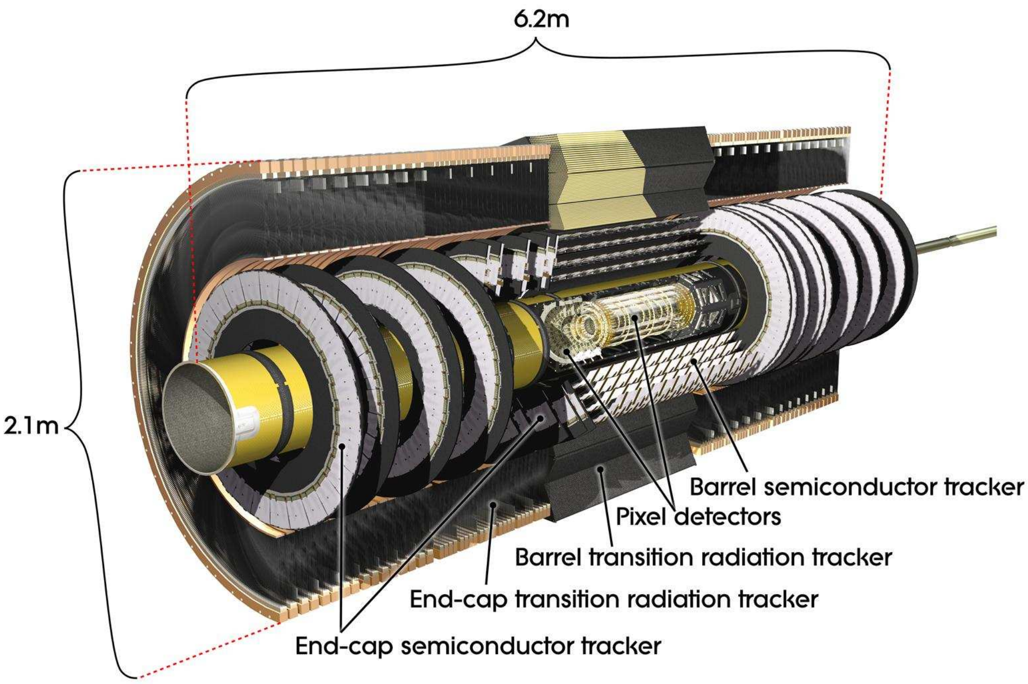

Figure 3.4: Inner Detector[33]

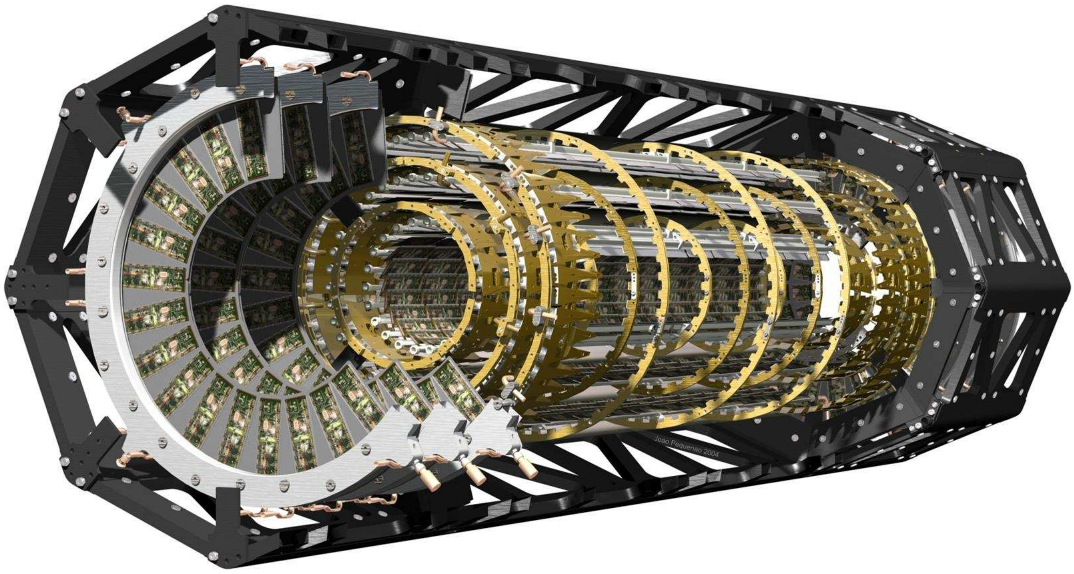

Figure 3.5: Pixel Detectors[34]

measures particle momenta from the curvature of the tracks, it is immersed in a 2T magnetic field originating from the central solenoid.

The pixel detector has approximately 80.4 million readout channels.

The SCT is designed such that each track crosses eight strip layers. The SCT is able to measure both coordinates $R{-}\phi$. This is made possible by using small-angle stereo strips with one set of strips parallell to the beam direction in each layer in the barrel region, and a combination of strips running radially and stereo strips at an angle of 40mrad in the end-caps. The SCT has approximately 6.3 million readout channels.

The TRT provides approximately 36 hits per track. This enables track-following up to $|\eta| = 2.0$. The large number of hits and the longer measured track length made possible by the TRT has a significant contribution to momentum measurment precision. The TRT provides $R - \phi$ information. Each straw has an accuracy of 130μm. The straws are 144cm long in the barrel region, where they are placed parallell to the beam axis. Their wires are divided into two halves at approximately $\eta = 0$. In the end-cap regions, the straws are 37cm long. Here they are arranged radially. The TRT has approximately 351000 readout channels.

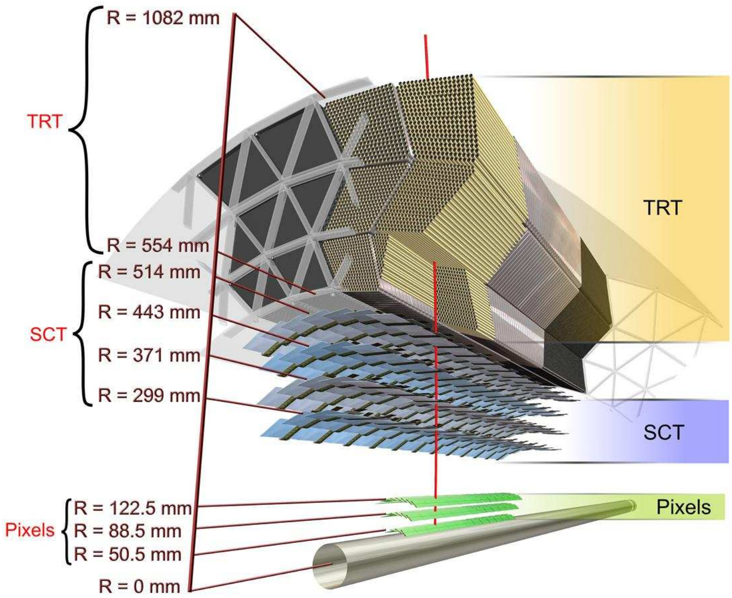

Figure 3.6: Inner Detector[35]

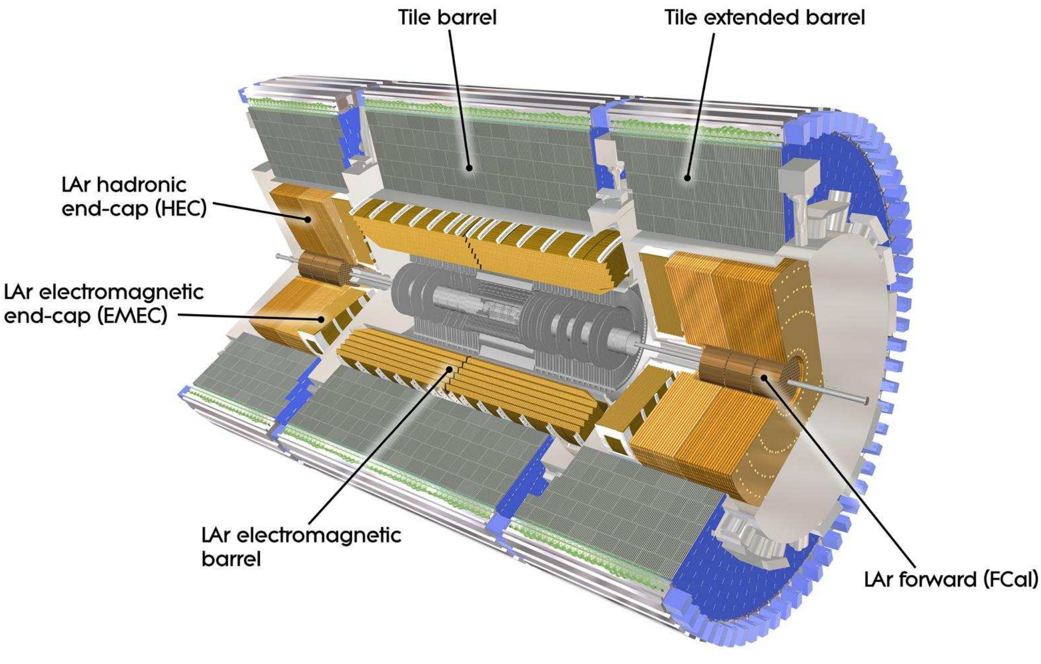

Figure 3.7: Electromagnetic Calorimeters[36]

3.2.4 Electromagnetic Calorimeter

Due to the large energy spectra available at the LHC, the EM calorimeter[20][4] must be able to identify electrons and photons with energy ranging from 5GeV to 5TeV. The EM calorimeter should be able to measure the electron and photon energies with a linearity better than 0.5%.

The EM calorimeter covers the region $|\eta| < 3.2$, but precission measurments are limited to $|\eta| < 2.5$.

The EM calorimeter is a lead-LAr detector and consists of three parts; the barrel region ($|\eta| < 1.475$) and the two end-caps ($1.375 < |\eta| < 3.2$). Each part is installed in a cryostat. It is supposed to provide complete ϕ symmetry without azimuthal cracks. This is due to its accordion structure.

As seen from fig. 3.7), the EM calorimeter consists of three longitudinal layers around the beam axis.

The EM calorimeter is more than 22 radiation lengths thick in the barrel and more than 24 radiation lengths thick in the end-caps. The end-caps have 10 interaction lenghts of active calorimeters. At $\eta = 0$ the total thickness is 11 interaction lenghts, including 1.3 interaction lengths from the outer support.

3.2.5 Hadronic Calorimeter

The calorimeters are the most important detectors for jet reconstruction.

The hadronic calorimeter[21][4] consists of

- A tile calorimeter in the barrel.
- A Liquid Argon (LAr) hadronic end-cap calorimeter (HEC)
- A LAr forward calorimeter (FCal)

The tile calorimeter is a sampling calorimeter consisting of steel and scintillating tiles as absorber and active material, respectively. The barrel part of the tile calorimeter covers the region $|\eta| < 1$. In addition, the tile calorimeter also has two extended barrels covering $0.8 < |\eta| < 1.7$. At $\eta = 0$, the thickness at the outer edge of the tile-instrumented region is 9.7 interaction lengths.

The hadronic end-cap calorimeter is located directly behind the end-cap parts of the EM calorimeter, and consists of two independent wheels. The wheels are made from copper surrounding LAr gaps. The hadronic end-cap calorimeter and the end-cap EM calorimeter shares LAr cryostats.

The forward calorimeter consists of three parts. The part closest to the interaction point is made from copper for EM measurements. The other two

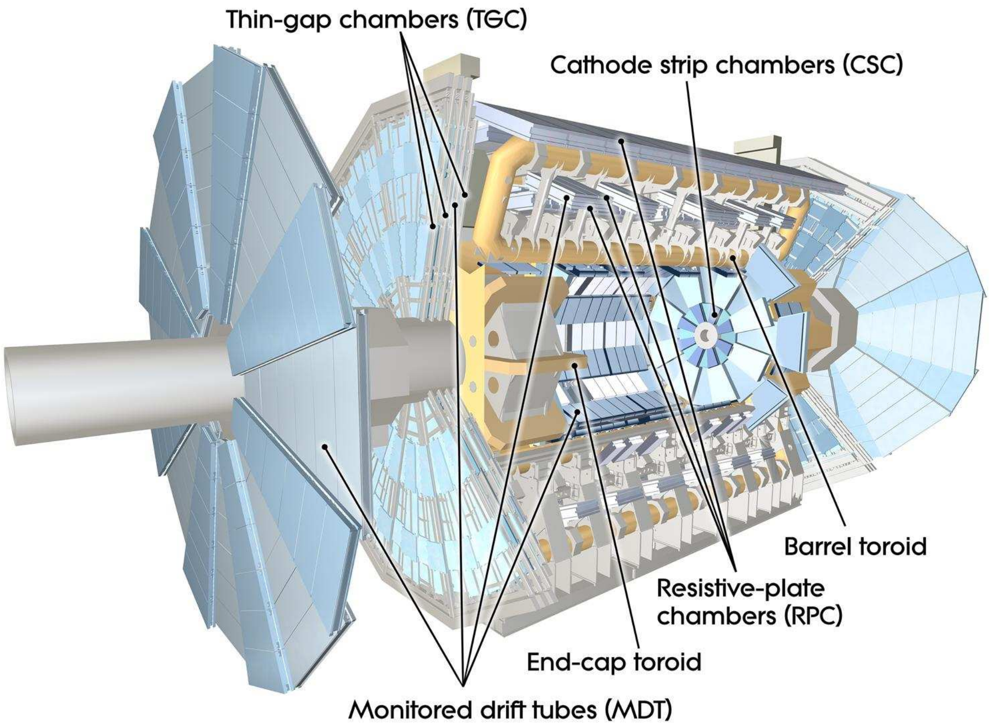

Figure 3.8: Muon Spectrometer[37]

are made from tungsten for hadronic measurements. The depth of the FCal is 10 interaction lengths. It is integrated into the end-cap cryostats.

The calorimeter system in ATLAS consists of approximately 200000 individual cells. The cells has different readout technologies and electrode geometries, as well as different sizes. The calorimeter system has an acceptance region and pseudorapidity $|\eta| < 5$ and $-\pi < \phi < \pi$

3.2.6 Muon Spectrometer

The muon spectrometer[22][5] is able to identify muons with momenta above 3GeV, and provides precise measurment of the muon p_{T} up to approximately 1TeV. It covers the region $|\eta| < 2.7$.

The muon spectrometer consists of superconducting coils, precision detectors and resistive plate and thin gap chambers (see fig. 3.8). The purpose of the superconducting coils is to provide a toroidal magnetic field that deflects muons. The muon spectrometer is instrumented with separate trigger and high-precision tracking chambers.

The detectors mainly consists of monitored drift tubes, but in the high-

η region ($|\eta| > 2.0$) of the innermost station they are replaced by cathode strip chambers, which are multiwire proportional chambers with cathodes segmented into strips. There the cathode strip chambers provide a rough 1cm measurment of ϕ.

The detectors measuring the magnetic field have a high precision ($< 100\mu$m) and are seperated into three stations.

In certain regions resistive plate and thin gap chambers provide rough measurments of both η and ϕ. Each station measure the magnetic field as a function of the η coordinate. That's the direction where most of the field deflection occurs. The stations are placed far appart.

Except from regions with support structures or passages for services, muons with high transverse momentum traverse all three stations.

3.2.7 Forward Detectors

There are also three smaller detector systems in ATLAS. They are all in the forward regions. These forward detectors[6] are

- LUCID (LUminosity measurement using Cherenkov Integrating Detector)
- ALFA (Absolute Luminosity For ATLAS)
- ZDC (Zero-Degree Calorimeter)

LUCID and ALFA are supposed to determine the luminosity delivered to ATLAS.

LUCID is placed at $\pm$17m from the interaction point and detects inelastic proton-proton scattering in the forward direction.

ALFA is placed at $\pm$240m from the interaction point. It consists of scintillating fibre trackers located inside Roman pots.

ZDC is important in heavy-ion collisions and is located at $\pm$140m from the interaction point. It consists of layers of alternating quartz rods and tungsten plates. These will measure neutral particles at $|\eta| \geq 8.2$

3.2.8 Magnet System

The ATLAS magnet system[7] consists of one solenoid (figure 3.9) and three toroids. The three toroids have been placed with one toroid in the barrel (figure 3.10) and one in each end-cap (figure 3.11). The entire magnetic system is 22m in diameter and 26m in lenght. It has a stored energy of 1.6GJ.

Figure 3.9: The ATLAS Solenoid[57]

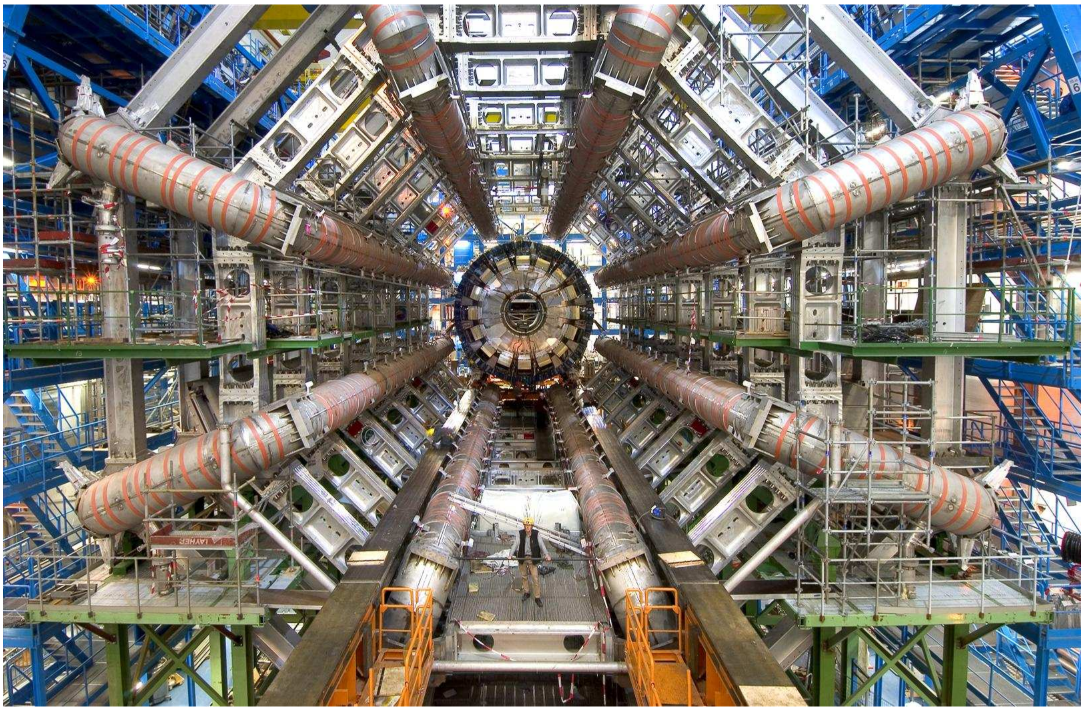

Figure 3.10: The ATLAS barrel toroid and hadronic calorimeter[1]

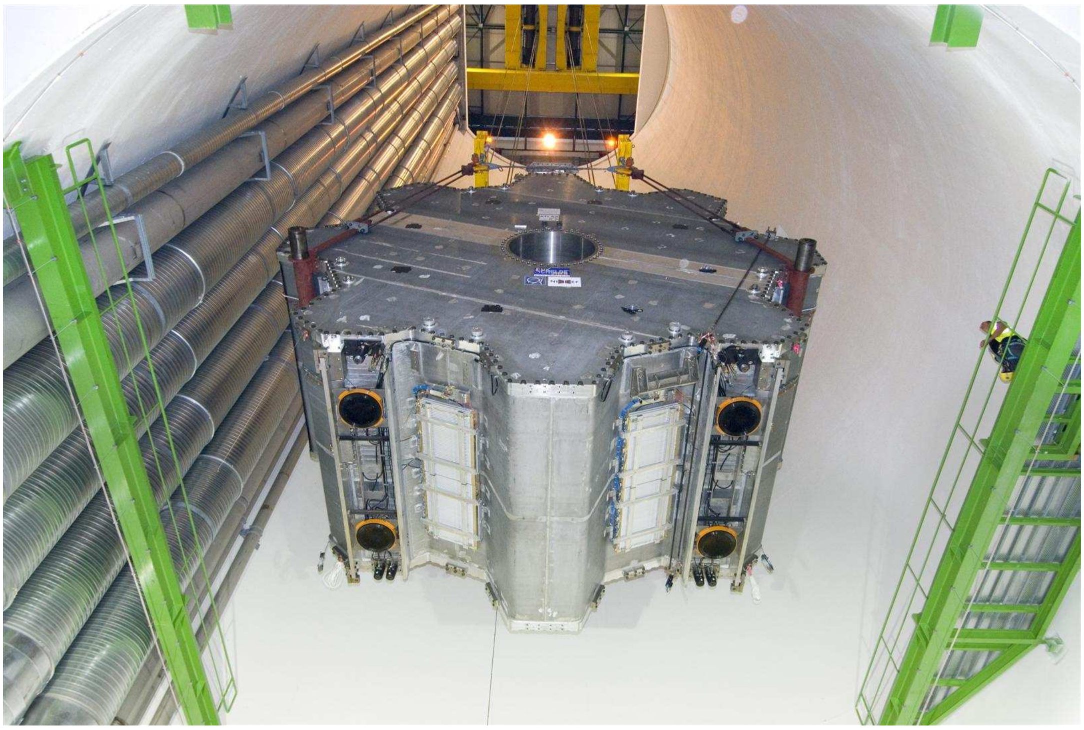

Figure 3.11: The ATLAS end-cap toroid being lowered into the cavern[30]

The central solenoid which is aligned with the beam axis extends over a length of $5.3m$ and has a diameter of $2.5m$. It generates a $2T$ magnetic field for the ID. The flux of the solenoid is returned by the steel of the ATLAS hadronic calorimeter.

The torroids produce a magnetic field of about 0.5T (1T for the muon detectors) in the central and end-cap regions, respectively.

3.2.9 Triggers and Data Streams

The crossing rate for bunches in the LHC will be 40MHz at design luminosity. This is way above the rate which is feasible to store and analyse. Therefore ATLAS has incorporated a trigger and data aquisition system[18][8] that will reduce the rate of events that will be stored to 200Hz, while keeping the physically interesting events. This is done by introducing three levels of triggers. The three levels are

1. Level 1 (L1)
2. Level 2 (L2)
3. Event Filter (EF)

Level 1 is hardware based. Level 2 and Event Filter, collectively known as High Level Trigger (HLT) are based on software algorithms.

The Level 1 trigger system must reduce the 40MHz crossing rate to an output rate of 75kHz. The purpose of this level is to select events with signatures of leptons and jets with high transverse momentum, as well as events with high missing transverse energy and events with high transverse energy.

Level 2 triggers should reduce the rate further to approximately 2kHz. The regions where the Level 1 triggers has found events that can be physically interesting are seeded into the Level 2 trigger system. The purpose of Level 2 triggers are then to reduce the necessary amount of data to be collected from the detector output, to avoid loosing physically interesting information.

And finally, the Event filter should reduce the rate of events to 200Hz, which is to be recorded for offline analysis.

Based on the selection of triggers, data can be sorted into data streams according to area of interest. The suggestion for the initial data streams is to have four data streams; egamma, jetTauEtmiss, muons and minibias. It has been decided that the streams should have approximately the same size, and that the fraction of events overlapping in in any two streams should be less than 10% of the total stream size.

Chapter 4

Real and Simulated data in ATLAS

4.1 Data in ATLAS

Data production in ATLAS is done in six steps:

1. Basic physics processes which occurs when two high energy partons collide.

2. New particles created through decays and fragmentation.

3. Particles and/or their decay products interact with the detector.

4. The detector response to the interaction with particles is digitalized and the output is collected in a bit stream.

5. Reconstruction of basic information, like particle tracks and energy deposits.

6. Reconstruction of high level information, jets, muon identification, tau identification and photon identification.

Since we do not per date have any real data, we need other methods of getting results from the first three steps. This is done using Monte Carlo simulation techniques. The output from the simulations can be collected into a bitstream. Such simulations are also done if we want complete control over what goes into the data stream, e.g. if we want to know what kind of signal to expect from a certain process. The last four steps are the same for both real and simulated data.

4.2 Simulations

To be able to test current theories, one needs to know what kind of signals are predicted by these theories and in what amount. Therefore, data sets with a known composition are needed for comparison with real data and for testing of software and hardware. Such comparison will also be the basis for being able to see if what will be observed is in accordance with what one expects given the current theories. These data sets are produced by Monte Carlo (MC) generators[23]. The most important general purpose MC generators are PYTHIA, HERWIG, Sherpa, AcerMC, ALPGEN, MadGraph/MadEvent and MC@NLO. There are also generators like Charybdis, CompHEP, TopReX and WINHAC available for more specific tasks.

The interaction of the Monte Carlo simulated particles with the detector is handled using GEANT4, which is a "toolkit for the simulation of the passage of particles through matter."[50] In order to simulate a realistic detector in the MC data sets, misalignments were introduced for the inner detector. And additional material for the inner detector and the front of the calorimeters and distorted magnetic fields were also introduced. In addition two different geomtetries have been used in most simulations, namely an *as-built geometry* and a *distorted geometry*. The as-built geometry uses realistic alignment shifts and distortions of the magnetic field. The distorted geometry is based on the as-built geometry, but adds extra material.

4.3 PYTHIA

PYTHIA is a Monte Carlo event generator which is important for hadronic processes. The dijet events examined in this thesis were generated using PYTHIA. PYTHIA was written by Torbjörn Sjöstrand, Stefan Ask, Richard Corke, Stephen Mrenna and Peter Skands. "PYTHIA is a program for the generation of high-energy physics events, i.e. for the description of collisions at high energies between elementary particles such as e+, e-, p and pbar in various combinations. It contains theory and models for a number of physics aspects, including hard and soft interactions, parton distributions, initial- and final-state parton showers, multiple interactions, fragmentation and decay. It is largely based on original research, but also borrows many formulae and other knowledge from the literature."[39]

4.4 JIMMY

The top events used in this thesis were created using JIMMY. JIMMY is used for generating multiple parton scattering events in hadron-hadron, photon-photon or photon-hadron events[48]. It was developed by Jon Butterworth, Jeff Forshaw, Mike Seymour and Rod Walker, and is a library of routines supposed to be linked with the HERWIG Monte Carlo event generator.

4.5 ATHENA

Athena is the ATLAS framework[45] that includes the software for event simulation, event trigger, event reconstruction as well as physics analysis tools[47]. It is a derivative of the Gaudi Common Framework Project, which was originally developed for LHCb. The project is now a shared project between ATLAS and LHCb. Today Guadi makes up the kernel of Athena, and Athena is the sum of that kernel and ATLAS-specific enhancements[44]. The framework is component-based, allowing great flexibility.[43] As a framework, Athena is an application skeleton where developers can plug in their code. It provides most of the communications between components, as well as providing the common functionality.[44]

4.6 ROOT

The plots in this thesis were made using the ROOT framework. According to the ROOT users guide[41], "ROOT is an object-oriented framework aimed at solving the data analysis challenges of high-energy physics". It was developed in the mid 1990's by RenÂ´e Brun and Fons Rademakers and is based on the programming language C++. ROOT is able to handle scripts through CINT, it's C++ interpreter. CINT was developed by Masa Goto.

4.7 Data Formats

To ease analyses, there is a hierarchy of dataformats[47] used by the ATLAS collaboration.

The most basic format is called "RAW". RAW is the "ByteStream" format with approximately 1.6MB/event. Raw data can be processed into ESDs, AODs and in the end DPDs.

ESD (Event Summary Data) gives a full output of reconstruction in object format. These include tracks and their hits, Calo Clusters, Calo Cells, combined reconstruction objects, etc. In the ESD format, each event takes up approximately 1MB.

AOD (Analysis Object Data) gives a summary of event reconstruction with "physics" objects, like electrons, muons, jets, etc. Size per event for the AOD format is 100kB, although at the start of 2008 it was approximately the double of that.

ESDs and AODs uses the POOL/ROOT format.

DPDs (Derived Physics Data) are "skimmed/slimmed/thinned events in addition to other useful "user" data derived from AODs and conditions data"[47]. On average these take up 10kB/event.

There is also a database used to quickly select events in AODs or ESDs, namely TAG.

Data can also be converted into ROOT ntuples.

4.8 Cross Section and Luminosity at LHC

The differential cross section per scattering center is given by[29]

$$\frac{d\sigma}{d\Omega} = \frac{\frac{\text{Scattered flux}}{\text{Incident flux}} \times \text{Unit of surface}}{\text{Unit of solid angle}} \tag{4.1}$$

meaning that the integrated cross section can be thought of as

$$\sigma = \int d\Omega \frac{d\sigma}{d\Omega} = \frac{\text{Probability of interaction}}{\text{Number of particles per unit of surface}} \tag{4.2}$$

The unit for cross section is the barn, $1\text{b} = 10^{28}\text{m}^2$

Luminosity in accelerator physics is defined by the relations[27]

$$\frac{dN}{dt} = L\sigma \tag{4.3}$$

and

$$\frac{d\sigma}{d\Omega} = \frac{1}{L}\frac{d^2N}{d\Omega dt} \tag{4.4}$$

Here, L is the luminosity, N is number of interactions, t is time, σ is the total cross section and Ω is the solid angle. By comparing equations (4.1) and (4.4) we see that the luminosity is related to the incoming flux.

For a storage ring collider like the LHC

$$L = fn\frac{N_1 N_2}{A} \tag{4.5}$$

where f is frequency of revolution, n is number of bunches headed in one direction, N_i is number of particles in bunch i and A is the beam cross section.

The common unit for luminosity is $\text{cm}^{-2}\text{s}^{-1}$.

The design luminosity of LHC is $10^{34}\text{cm}^{-2}\text{s}^{-1}$.

4.9 Full Dress Rehearsal

To give an idea of the current stage of development and preparedness before getting real data, Full Dress Rehearsals (FDR)[46] have been held. These are simulations of what one would expect to observe when LHC starts providing physically interesting collisions. FDR have had three phases; Phase-0, FDR1 and FDR2.

Phase-0 was held to give an evaluation of different data streaming techniques on physics analysis, testing of the bytestream production and decoding software. Phase-0 was called the streaming test, and was held in summer 2007.

The focus of FDR1 was to simulate the full data processing chain from the SFO output disk at point-1 through to Tier-2 AOD distribution and analysis. The simulated luminosity was according to what was expected for the 2008 runs. This took place at the start of february 2008.

The final rehearsal, FDR2, took place in the beginning of june 2008. It had similar objectives as FDR1, but with higher luminosity and more realistic samples. For example, simulated noisy towers were added to the data.

Chapter 5

Jet Reconstruction Algorithms

5.1 Introduction

Precise jet reconstruction[24] is an important tool for almost all physics analysis to be performed with the ATLAS detector. The following chapters will define some of the key terms, ond give an outline of how the jets are being reconstructed.

5.2 Definition of a Jet

A jet is a number of mainly hadronic particles passing through the detector in a tight cone originating from the interaction region. These particles are a result of the fragmentation of a parton (quark or gluon). For this thesis we decided to study perturbative processes, i.e. high energy phenomena. It was therefore necessary to introduce a threshold on the transverse energy, and we required the jet to have a transverse energy above 20GeV. This is due to the jet reconstruction algorithms not beeing good below that energy threshold[29]. In this thesis, dijets are of interest. These are events where the final state consists of two jets. The definition of a jet in ATLAS is closely related to the jet reconstruction algorithms outlined below.

5.3 Jet Reconstruction Algorithms

Currently, there is no general way of reconstructing the final hadronic state for all cases of interest. That means that several different jet reconstruction algorithms[24] are needed. In ATLAS, one has tried to include all relevant ones. They are the

- Iterative seeded fixed-cone jet finder
- Sequential recombination algorithms
  - Seeded fixed cone
  - k_{T}
- Alternative jet finders
  - Mid-point
  - "Optimal jet finder"

All these algorithms provide full four-momentum recombination whenever the constituents of a jet change.

5.3.1 Guidelines for Jet Reconstruction

The guidelines below have been partly quoted from the reference [24] and are assumed to be robust and carefully formulated.

Theoretical Guidelines for Jet Reconstruction

The main theoretical guidelines for jet reconstruction are

- Infrared safety: Soft particles not coming from the fragmentation of a hard scattered parton should not affect the number of jets produced. In particular, the presence or absence of soft particles between two particles belonging to the same jet should not disturb the correct reconstruction of the jet.
- Collinear safety: Even though a certain amount of the transverse momentum is carried by one particle, the jet should be reconstructed. The same goes for when a particle is split into two collinear particles.
- Order independence: Level of reconstruction (parton-, particle- or detector level) should be irrelevant for reconstruction of hard scattering in specific cases.

Experimental Guidelines for Jet Reconstruction

In addition to the theoretical guidelines, experimental guidelines are also provided according to the design of the detector. They are

- Detector technology independence: Detector specific signal characteristic and detector inefficiencies must be calibrated out.
  - Detector resolution: Effects originating from the finite detector resolution must be at a minimum.
  - Detector environment: Electronics noice, signal losses, cracks between detectors and other contributions from the detector environment must be kept at a minimum.
  - Stable signals: The jet reconstruction needs a stable input signal provided by the detector signal reconstruction and calibration.
- Environment independence: The jet reconstruction needs to be independent of e.g. multiple interactions and pile-up, source of the jet and underlying event activity, i.e. the low energy processes between the scattered particles heading into the detector and the particles that collided.
  - Stability: Jet finding and reconstruction should not be disturbed by changing underlying event activity and changing instantaneous luminosity, even though the number of multiple interactions will change.
  - Efficiency: All physically interesting jets originating from partons with energy above a certain threshold must be reconstructed with high efficiency.
- Implementation: The jet algorithm implementation must be fully specified in that the jet definition must be complete. The jet definition consists of the jet finder and its configuration together with the choice of kinematic recombination. One must also include all selections and the signal choices whenever they are relevant for the jet in question. The implementation of the jet reconstruction must be fast and avoid excessive memory consumption.

5.3.2 Description of the Algorithms

The algorithms that has formed the basis for most predictions related to the performance of the hadronic final state reconstruction are the Iterative Seeded Fixed-Cone Jet Finder and the Sequential Recombination Algorithms. They have been used for almost all pre-collision studies. Alternative jet finders for precision analysis of specific final states are also available.

Iterative Seeded Fixed-Cone Jet Finder

The idea behind the Iterative Seeded Fixed-Cone Jet Finder is to construct cones with a certain radius around energetical objects. Objects within the cone are assumed to be part of what will be defined as a jet when the direction of the cone is stabilized. In detail what happens is that all input is ordered in decreasing order of transverse momentum. If the object with the highest transverse momentum is above the seed threshold (> 1GeV) a cone is constructed around it. The cone has a fixed cone radius R_{cone}. All objects within the cone, i.e. that has $\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2} < R_{\text{cone}}$ are combined with the seed. A new direction is calculated from the four-momenta inside the initial cone and a new cone is centered around it. Objects are then ordered accordingly, before the direction is updated again. This process continues until the direction is stabilized, and the cone is considered a jet. The seed is removed from the input list, and the process then continues until no more seeds are available. As for the cone redius, a narrow and a wide cone jet option are available. The narrow cone jet option, $R_{\text{cone}} = 0.4$, is mainly used for $W \to jj$ in $t\bar{t}$ and supersymmetric events. And the wide cone jet option, $R_{\text{cone}} = 0.7$, is most often used for inclusive jet cross-section, $Z' \to jj$.

This algorithm is only meaningful to leading order for inclusive jet cross-section measurments and final states like W/Z + 1 jet. The algorithm is not meaningful at any order for 3-jets final states, W/Z + 2 jets, and for the measurment of the dijet invariant mass in 2 jets + X final states.

The algorithm is not infrared safe. This can, however, be partly fixed. In the Iterative Seeded Fixed-Cone Jet Finder it is possible for jets to share constituents. If these shared constituents carry more than a certain amount of the transverse momentum of the less energetic jet, the two jets are merged. If the shared transverse momentum is less than the mentioned threshold, the jets are split. In ATLAS this threshold is set to a factor 0.5, i.e., if the shared constituents share more than half of the transverse momentum of the less energetic jet, the jets are merged.

Signal objects contributing to the cone at some iteration may be lost due to recalculations at later iterations.

Sequential Recombination Algorithms

The k_{T} algorithm is the default implementation of a sequential recombination algorithm in ATLAS. Contrary to the fixed-cone jet finder, no objects are shared between jets, and the procedure is infrared and collinear safe. This algorithm sees all jets as either a jet or a part of a jet. All jets are removed from the selection, while the rest is combined into jets, or into a new part of a jet. The basic idea is that objects that are very close to each other probably belongs to the same jet, and therefore will be combined. To figure out which objects are jets, and wich are parts of a jet, the minimum, d_{min} of all d_{ij} and d_i is found for objects i and j. d_i is defined by equation 5.1 and d_{ij} is defined by equation 5.2.

$$d_i = p^2_{\mathrm{T},i} \tag{5.1}$$

$$d_{ij} = min(p^2_{\mathrm{T},i}, p^2_{\mathrm{T},j})\frac{\Delta R^2_{ij}}{R^2} = min(p^2_{\mathrm{T},i}, p^2_{\mathrm{T},j})\frac{\Delta\eta^2_{ij} + \Delta\phi^2_{ij}}{R^2}, \tag{5.2}$$

If d_{min} is a d_{ij}, the corresponding i and j are combined into a new object k using four-momentum recombination. i and j are then removed from the list, while k is added to it. If d_{min} is a d_i the object is considered to be a jet and thereby removed from the list. The above algorithm is repeated until all objects are removed from the list.

The distance parameter R allows some control on the size of jets. Default configurations for the distance parameter R in ATLAS are $R = 0.4$ and $R = 0.6$. $R = 0.4$ is mostly used for $W \rightarrow jj$ in $t\bar{t}$, SUSY, while $R = 0.6$ is mostly used for inclusive jet cross-section, $Z' \rightarrow jj$.

Alternative Jet Finders

The alternative jet finders available in ATLAS are the mid-point variant of the fixed-cone algorithm, and the "optimal jet finder". Their performance are very similar to the default jet finder implementations.

The mid-point algorithm places the seeds between two particles of significant transverse momentum, rather than just using a single particle p_{T} as seed.

The "optimal jet finder" introduces a test function. By minimizing that function, one is able to calculate a particle's contribution to a jet. In cases where one gets busy final states, e.g. full hadronic top decays in $t\bar{t}$ production, this might be the preferred algorithm.

5.3.3 Jet reconstruction and the calorimeters

The calorimeters are the most important detectors for jet reconstruction. The calorimeters consists of a large number of cells. Without combining these into larger groups one will not obtain a meaningful four-momenta when looking for jets. There are two ways of doing this. *Signal towers* and *topological cell* *clusters* are the two available concepts. We will here only discuss topological cell clusters.

The concept of topological cell clusters is an attempt to reconstruct 3-dimensional structures representing the showers originating from the individual particles entering the calorimeter. The clustering starts with a seed cell with a signal significance, Γ, above a certain threshold, S. I.e. $|\Gamma| = \left|\frac{E_{\text{cell}}}{\sigma_{\text{noise,cell}}}\right| > S = 4$. All neighbouring cells to the cells fullfilling this criteria are included in the cluster. If any of these neighbouring cells have $|\Gamma| > N = 2$ their neighbours are also added. Finally a ring of guard cells with $|\Gamma| > P = 0$ is included in the cluster. The initial clusters are then analyzed for local maximums, and split between those maximums if any are found. Due to larger cell sizes and shower overlap for $|\eta| \leq 1.5$ and the increase in shower overlap and larger cell sizes for $|\eta| \geq 2.5$, this algorithm works best for $1.5 \leq |\eta| \leq 2.5$. The increase in shower overlap in the forward region is caused by a decreasing linear distance between jet particles.

Chapter 6

Results

6.1 Objective

The purpose of this thesis is to provide a software framework for a high-level detector check with early data. In particular the program should provide an estimate of the jet resolution of the ATLAS detector and compare several of such methods. The program should also be able to spot and locate large detector problems, like noisy or non-working detector modules. At start-up, LHC will provide a relatively low luminosity. It was therefore decided that dijets were most useful in the early running period, since other methods gives smaller statistics.

6.2 Introduction

It should be noted that, due to delay of the experiments, change of concepts for data streams, and difficulties in achieving stability of the software framework necessary to run the thesis specific computer program, the work scope for the thesis has kept changing with respect to the original scope. The initial concept was to run on the Express Stream. However, it was decided that the Express Stream would not be made generally available, with the consequence that the program developed for this thesis needed to run on any stream of both ESD and AOD format. This required much additional work. The final concept then was a program that compares several different variables for measuring jet resolution. The program was formulated in a way that can both be compiled and run within Athena, or run as a script within AthenaROOTAccess.

For the analyses we have looked at AODs from FDR2, AODs from Monte Carlo dijet simulations and Monte Carlo top simulations.

6.3 The Program

The results in this thesis are based on the output from the program DiJet. DiJet was written by Kent Olav Skjei and Thomas Burgess. DiJet can be run both in AthenaROOTAccess (ARA) as a script, and in Athena as a compiled program. The program can be run on both AODs and ESDs. The program looks at different variables for estimating jet resolution using dijets and a reconstruction of the W mass.

The dijet selection method we have used is to demand at least two jets to have transverse energy above a certain transverse energy cut, and that the cosine of the angle between the leading and next to leading jet is less than -0.92.

We selected truth jets assumed to belong to the reconstructed jet by minimizing[25]

$$\Delta R = \sqrt{(\Delta\phi)^2 + (\Delta\eta)^2} \tag{6.1}$$

A truth jet is a collection of truth particles moving inside a jet cone. Truth particles are particles with their true energy available from the simulation, not the reconstructed energy.

To reconstruct the W mass we use $t\bar{t}$ events with at least one lepton (electron or muon) with transverse momentum larger than 20GeV in the η region $-2.5 \leq \eta \leq 2.5$, two b jets in the same η region as the leptons, with transverse momentum above 40GeV, and two non-b-tagged jets with the same cuts as the b tagged ones (see fig. 2.4). We also demand at least 20GeV missing transverse energy. The invariant mass of the W is then calculated from the formula

$$m_W = \sqrt{E^2 - \vec{p}^2} \tag{6.2}$$

not using b-tagged jets, i.e. jets not originating from a b quark.

6.4 Data Sets Used for the Results

For the results in this thesis, we have looked at Monte Carlo dijets, Monte Carlo top events and FDR2 data.

The Monte Carlo dijets used was 39900 events gathered from the set[49] “mc08.105013.J4_pythia_jetjet.recon.AOD.e344_s479_d150_r642_tid046394”. These are events from 10TeV collisions, giving J4 QCD dijets with the transverse momentum range 140 − 280GeV. The cross section for this process is $3.08 \cdot 10^5$pb.[52] This corresponds to an equivalent luminosity of 0.13pb^{-1}.

The Monte Carlo top events are 26922 events from the set[49] "mc08.105200.T1_McAtNlo_Jimmy.recon.AOD.e357_s462_r579_tid028663". These are events from 10TeV collisions, giving top quarks with a mass of 172.5GeV. This process has a cross section of $1.07 \cdot 401.60$pb.[51] This corresponds to an equivalent luminosity of 62.65pb^{-1}.

The FDR2 data consists of 20000 events collected from the set[49] "fdr08_run2.0052280.physics_Jet.merge.AOD.o3_f8_m10". The entire set consists of 48471 events, corresponding to 1 hour of running with a luminosity of 10^{32}cm^{-2}s^{-1}. The equivalent luminosity in our analysis was then $0.15 \cdot$ pb^{-1}.

We treat FDR2 as semi-real data in order to see how DiJet will work on real data. This means that we for each subject studied, we start out by looking at the Monte Carlo dijet or top sets. This gives us an idea of what to expect according to present knowledge about the detector and the physics processes. This is then compared with FDR2 data, to see if what we observe is in accordance with expectations.

6.5 Variables and Plots

We have looked at several variables used for estimating jet resolution. One of them was the energy fraction of reconstructed and truth jet [25]

$$\frac{E_{\mathrm{T},0}}{E_{\mathrm{T},0,Truth}} \tag{6.3}$$

where $E_{\mathrm{T},0}$ is the transverse energy of the highest energy reconstructed jet. $E_{\mathrm{T},0,Truth}$ is the transverse energy of the highest energy truth jet in Monte Carlo data. This was the variable used for optimizing the jet resolution in ATLAS. This variable does not take into account the outflow from the jets, and it is not affected by gluon radiation. As such it can be more or less considered as the resolution of the particles rather than actual jets.

The second variable used in the analysis was the energy balance

$$\frac{E_{\mathrm{T}-pos.-\phi} - E_{\mathrm{T}-neg.-\phi}}{E_{\mathrm{T},0}} \tag{6.4}$$

where $E_{\mathrm{T}-pos.-\phi}$ is the transverse energy of the jet in positive ϕ for dijets. $E_{\mathrm{T}-neg.-\phi}$ is the transverse energy of the jet in negative ϕ, and $E_{\mathrm{T},0}$ is the transverse energy of the highest energy jet. A similar variable was used by CMS[38].

While working with the energy balance, something indicating a detector asymmetry in phi was discovered. As (6.4) is explicitly dependent on phi, the energy balance was not suitable for studying the jet resolution as a function in ϕ. Instead we also started looking at a combined plot of the variable[29]

$$\frac{E_{\mathrm{T},0}}{E_{\mathrm{T},1}} \tag{6.5}$$

and

$$\frac{E_{\mathrm{T},1}}{E_{\mathrm{T},0}} \tag{6.6}$$

where $E_{\mathrm{T},1}$ is the transverse energy of the next to leading jet in dijet events. This will be referred to as the energy fraction between leading jets.

In an ideal situation one would have $E_{\mathrm{T},0} = E_{\mathrm{T},1}$. In the real world this is not the case. There are several things affecting this:

- Statistical nature of fragmentation
- Gluon emission from the final state
- Jet definitions
- Hadron energy resolution
- Problems with the detector
- Underlying event

Monte Carlo simulations which takes into account the predictions of QCD, should be able to give the exptected results given fragmentation, gluon emission and jet definition. So deviations from these expected results should be able to provide information on the jet resolution and major problems with the detector.

The last variable we looked at for estimating the jet resolution was the W mass resolution. We here expected to find

$$\frac{\Delta E_{\mathrm{T},jet}}{E_{\mathrm{T},jet}} \approx \frac{\Delta m_W}{m_W} \tag{6.7}$$

This distribution was normed by dividing the reconstructed W mass by 80.4, which is the W mass given by the Particle Data Group [27].

In addition we looked at resolution of missing transverse energy and the definition of missing transverse energy with the variables

$$\frac{E_{\mathrm{T},missing}}{\sum_{i=0}^{n} E_{\mathrm{T},i}} \tag{6.8}$$

where $E_{\mathrm{T},missing}$ is the missing transverse energy of the event, and $\sum_{i=0}^{n} E_{\mathrm{T},i}$ is the sum of all visible transverse energy in the event, and

$$\frac{\sum_{i=0}^{n} p_{\vec{\mathrm{T}},i}}{\sum_{i=0}^{n} E_{\mathrm{T},i}} \tag{6.9}$$

where $\sum_{i=0}^{n} p_{\vec{\mathrm{T}},i}$ is the sum over transverse momentum of jets in the event. The latter will be referred to as resolution of vectorial sum over momenta. Naively, one might expect $E_{\mathrm{T},missing} \approx \sum_{i=0}^{n} p_{\vec{\mathrm{T}},i}$. However, as we shall see, this is not correct, as the vectorial sum over momenta does not take into account the outflow of particles from the jet cones, and the presence of muons.

In addition to ploting all the above mentioned plots, we also looked at the profiles of both the variable and the root mean square (RMS) of the variable in η and ϕ.

Since all of the above mentioned distributions are continous and should be clustered around a mean, the Gaussian distribution has been used for the fits. Of course, this is not necessarily theoretically correct, since (6.4) is the only distribution one would expect to be symmetric. This is due to the fact that for the other variables we plot $\frac{n}{m}$, where $n \leq m$ is in the interval $[0, 1]$, meaning that the distribution has a lower bound at 0. For the interval $[1, \infty]$ that is not the case. In that interval there is no clear upper bound, and the distributions should be broader in that interval. However, we still feel that the Gaussian is justified for the central part of the distribution, around the mean. All mean values and width of distributions quoted in this text comes from the Gaussian fits. They can be read off the two bottom values on the plots.

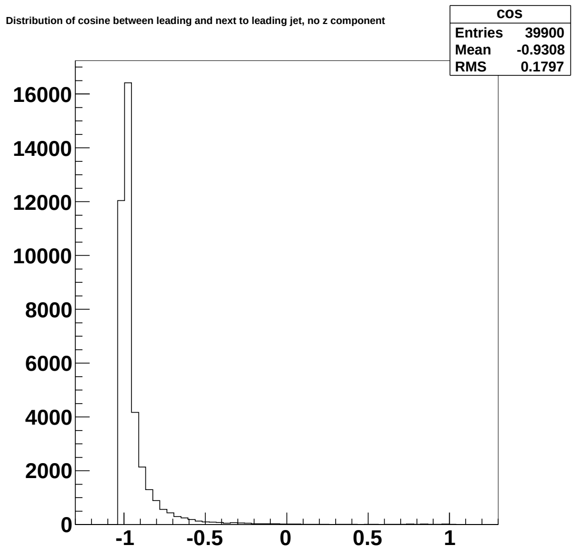

Figure 6.1: Cosine to the angle between leading and next to leading jet in the transverse plane in Monte Carlo dijet events with $\cos(\alpha) < -0.92$

6.6 Cut for Back-To-Back Jets in Dijet Selection

In figures 6.1 and 6.2 the cosine of the angle between the two most energetic jets in Monte Carlo dijet events have been plotted. We see that a very strict angle cut will give low statistics. As high statistics is a necessity for the use of the dijet-based methods studied in this thesis for use on early data, we could not afford a cut too strict. We therefore decided to call jets which angle fulfills $\cos(\alpha) < -0.92$ in the transverse plane back-to-back dijets.

We made a cut in the transverse plane since the cosine distributions have a long tale when we take the z-direction into account (fig. 6.3). This is because the collisions happen between partons, and we therefore don't expect the interactions to happen in the center of mass system.

6.7 Good η and ϕ regions

First we looked at was the distribution of (6.4) in Monte Carlo dijet data. Starting out we had introduced a 20GeV cut on transverse energy to let a jet be part of a dijet event. This because jet reconstruction algorithms are not reliable for transverse energies less than 20GeV[29]. As can be seen in fig.

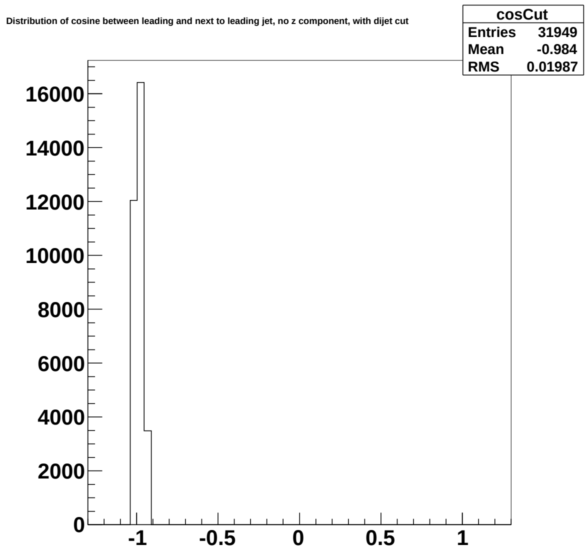

Figure 6.2: Cosine to the angle between leading and next to leading jet in the transverse plane in Monte Carlo dijet events with $\cos(\alpha) < -0.92$

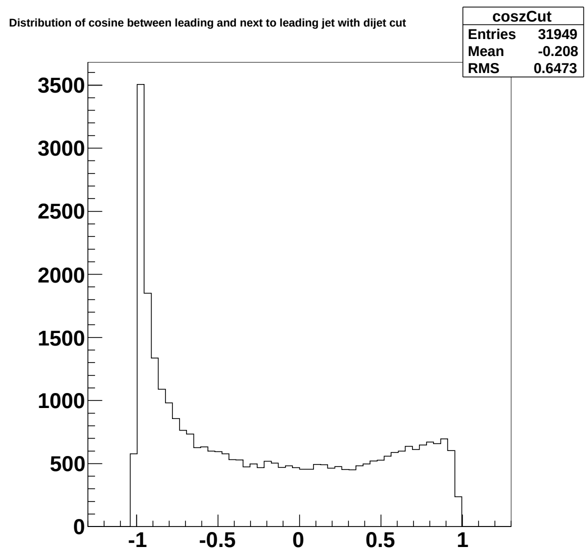

Figure 6.3: Cosine to the angle between leading and next to leading jet in Monte Carlo dijet events with $\cos(\alpha) < -0.92$

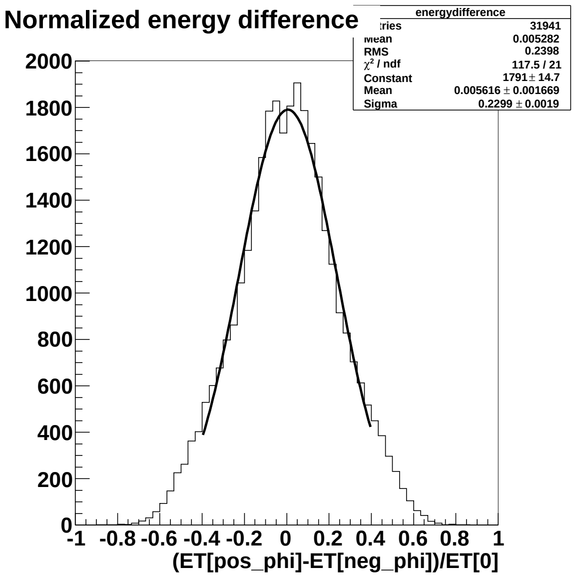

Figure 6.4: Energy balance from Monte Carlo dijets (6.4) with a 20GeV E_{T} cut

6.4 this gave a rather broad distribution. Therefore we introduced a 100GeV cut on transverse energy for the next to leading jet instead. This gave a more narrow distribution (fig. 6.5). Then we looked at the RMS distribution of (6.4) in η and ϕ to look for any large deviations due to low statistics or bad detector parts. As susptected, there is a decrease in events for large $|\eta|$ (fig. 6.6). This leads to an apparent decrease in RMS, but we suspect that the error of RMS increases (fig. 6.7). Based on this information we made the decision to look at the region $-3 \leq \eta \leq 3$. As suspected nothing similar was observed in ϕ (fig. 6.8). The peak in fig. 6.8 is due to the bin being filled twice. The value of the RMS comes from looking at half of the distribution. This is due to the construction of the variable.

6.8 Asymmetry in ϕ

By looking at the energy balance with a 100GeV cut on E_{T} for Monte Carlo dijets (fig. 6.9), one observes that there is an approximate $(0.7 \pm 0.2) \cdot 10^{-2}$ deviation from the expected mean of 0.

This led to the production of several other plots to investigate the matter. A simple counting of leading jets for positive and negative ϕ (fig. 6.10), respectively, shows explicitly that there seems to be more leading jets in positive ϕ. Counting, one finds 16120 leading jets in positive ϕ and 15236

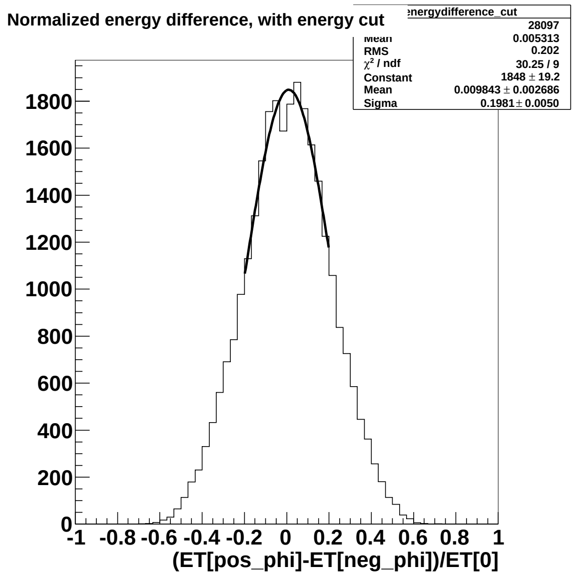

Figure 6.5: Energy balance from Monte Carlo dijets (6.4) with a 100GeV E_{T} cut

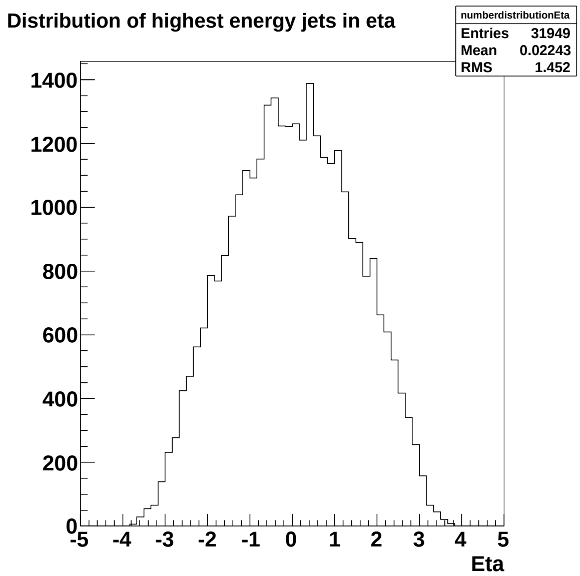

Figure 6.6: Distribution of leading jets in eta from Monte Carlo dijets

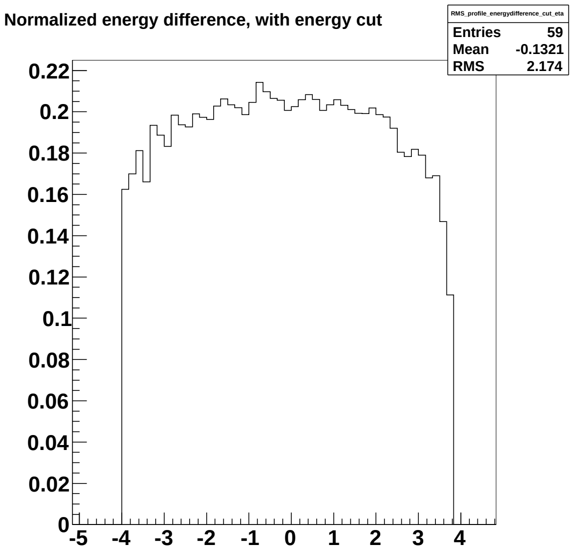

Figure 6.7: RMS profile of distribution energy balance in fig. 6.5 in η from Monte Carlo dijets

leading jets in negative ϕ. This gives the relative difference

$$\frac{N_{pos.\phi - N_{neg.\phi}}}{N_{tot}} \pm \frac{\sqrt{N_{pos.\phi} + N_{neg.\phi}}}{N_{tot}} = 0.028 \pm 0.006 \tag{6.10}$$

It is interesting to note that there is a related structure in the profile plot of the missing energy resolution for Monte Carlo dijets (fig. 6.11). This distribution shows that there is more missing transverse energy in negative ϕ, which indicates a higher number of leading jets in positive ϕ.

Comparing fig. 6.11 with fig. 6.12 we see that the structure is smaller for the resolution of vectorial sum over momenta. So it seems that particles which are left out of the jet definition are increasing the phi asymmetry.

Looking at the plot of distribution of leading jets in ϕ (fig. 6.13), we observe something similar to a hole in the detector just below $\phi = -1$. But even if this “hole” wasn’t there it visually looks like there are more leading jets in positive ϕ.

What was observed in fig. 6.13 can also be seen in the profile plot of the energy fraction of reconstructed and truth jet (fig. 6.14) So again it seems as if the energy scale is too high for positive ϕ.

Now we can compare with the results for Monte Carlo top events and FDR2 data. In Monte Carlo top events (fig. 6.15) we actually see the opposite of what we see for Monte Carlo dijets. We find 3737 of leading jets in positive

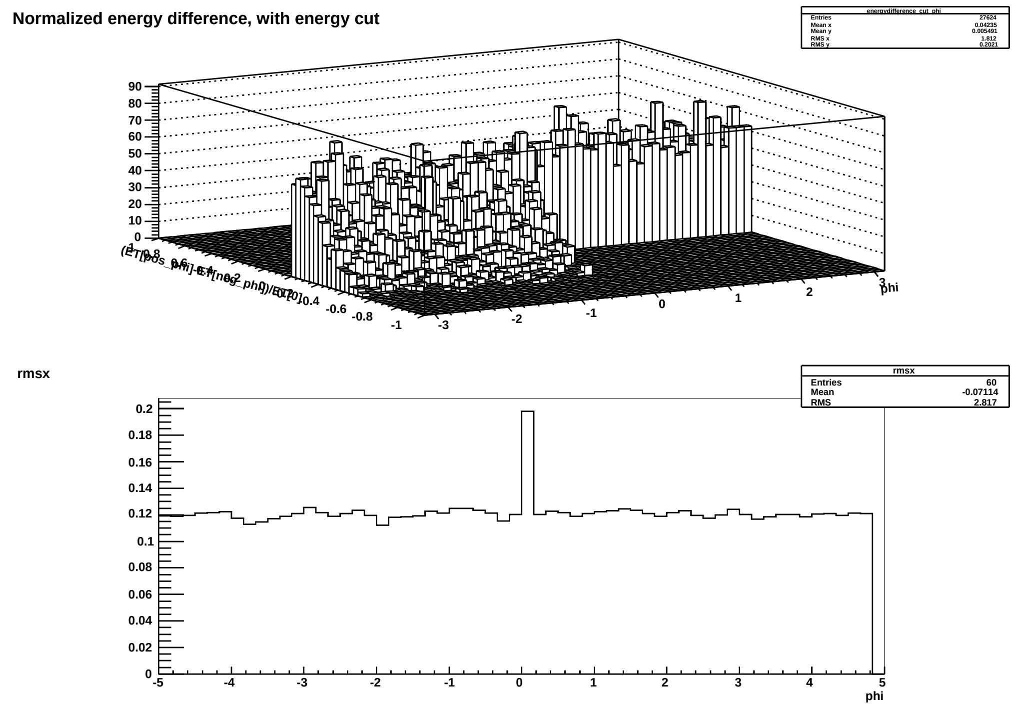

Figure 6.8: RMS profile of distribution of energy balance in fig. 6.5 in ϕ from Monte Carlo dijets. The peak in this figure is due to the bin being filled twice. The value of the RMS comes from looking at half of the distribution. This is due to the construction of the variable.

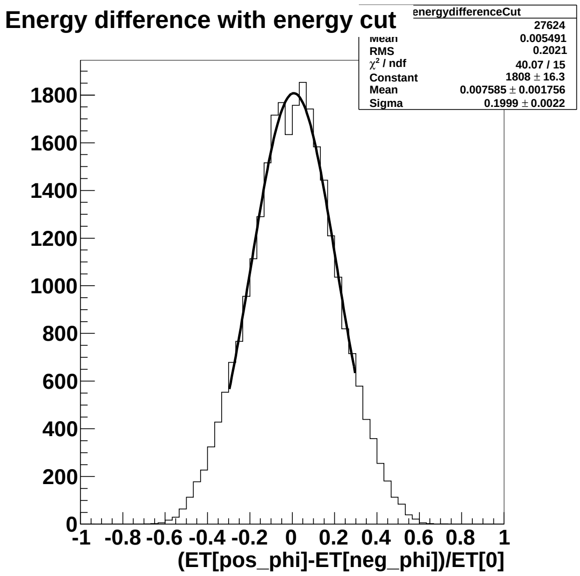

Figure 6.9: Energy balance from Monte Carlo dijets (6.4) with a 100GeV E_{T} cut

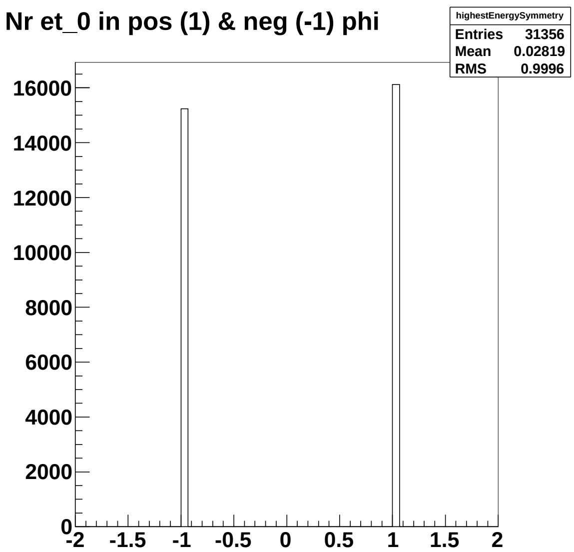

Figure 6.10: Counting number of leading jets in positive (1) and negative (-1) ϕ for Monte Carlo dijets

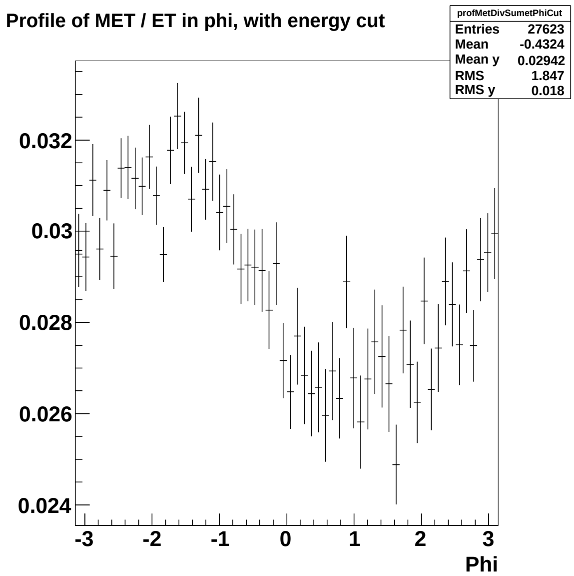

Figure 6.11: Profile of the missing energy resolution in ϕ, with a 100GeV cut on E_{T} for Monte Carlo dijets

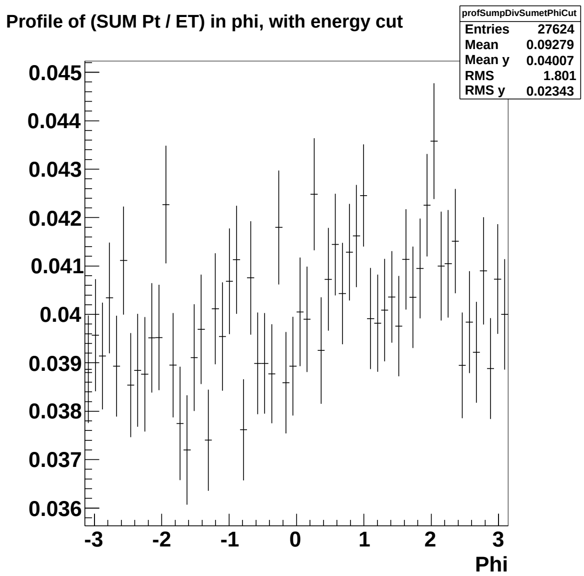

Figure 6.12: Profile of the resolution of vectorial sum over momenta in ϕ, with a 100GeV cut on E_{T} for Monte Carlo dijets

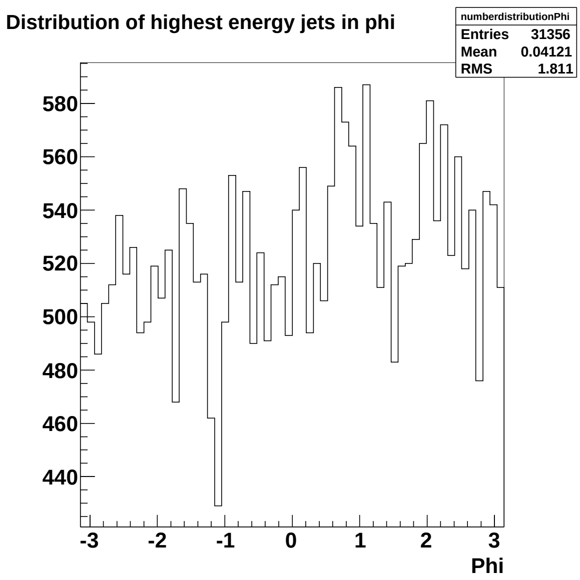

Figure 6.13: Distribution of leading jets in ϕ for Monte Carlo dijets

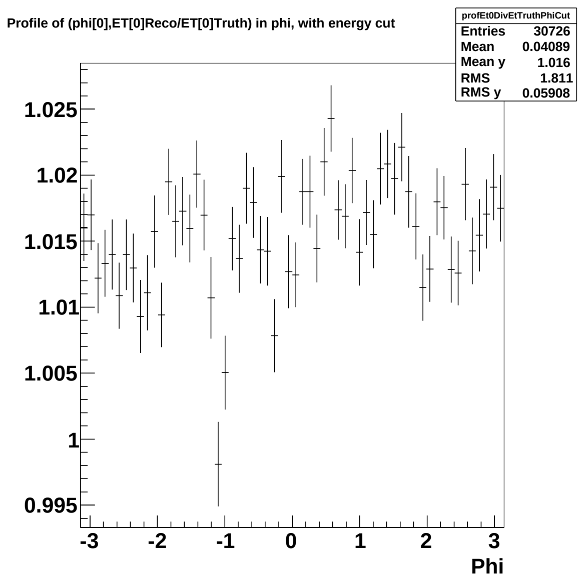

Figure 6.14: Profile of the energy fraction of reconstructed and truth jet with a 100GeV cut on transverse energy for Monte Carlo dijets

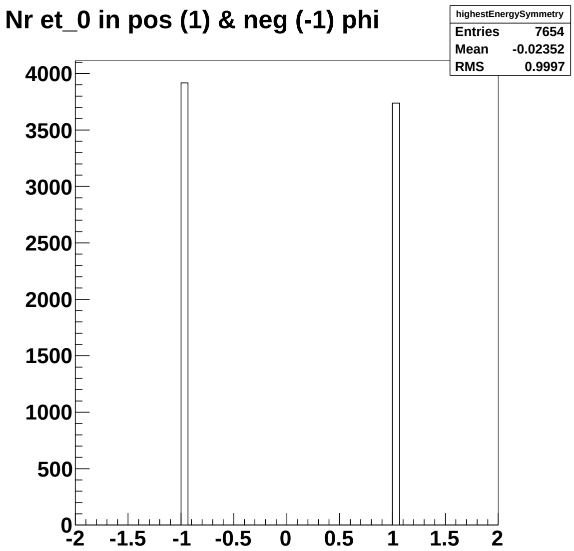

Figure 6.15: Counting number of leading jets in positive (1) and negative (-1) ϕ for Monte Carlo top events

ϕ and 3917 leading jets in negative ϕ. This gives a relative difference -0.02 ± 0.01. There it seems as if there are most leading jets in negative ϕ. However, here the statistics are lower, and the difference is smaller, increasing the influence of statistical errors.

Finally, in FDR2 data, we observe the same as for Monte Carlo dijets (fig. 6.16).

There are 79769 leading jets in positive ϕ and 77147 leading jets in negative ϕ. This gives the relative difference 0.017 ± 0.003. So again it seems there is a asymmetry in phi. So the difference is smaller than for Monte Carlo dijets. FDR2 data contain, as we shall see later, mostly events that would be considered dijets by the program developed in connection with this thesis. One might speculate that the energy scale is higher in positive ϕ for dijets, and lower for other Monte Carlo simulations, as seen for top events. That could explain the smearing out of the relative difference observed in FDR2 data(0.017 ± 0.003 for FDR2 compared with 0.028 ± 0.006 for Monte Carlo dijets) when several Monte Carlo simulations are put together.

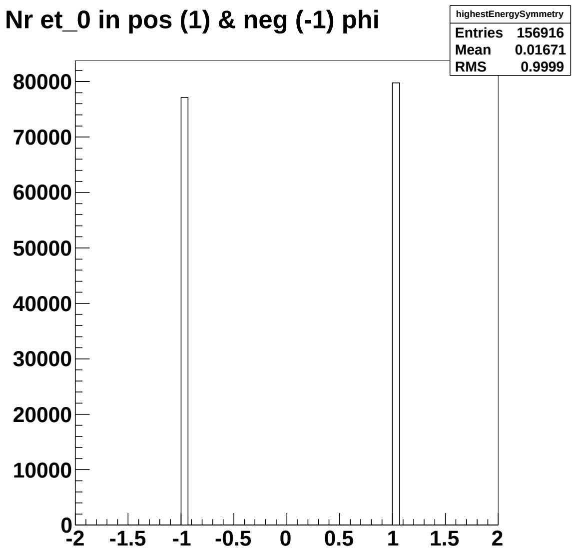

Figure 6.16: Counting number of leading jets in positive (1) and negative (-1) ϕ for FDR2 data

6.9 Jet Resolution Estimates

6.9.1 η Dependence of Transverse Energy

By looking at the profile plot of the energy fraction of reconstructed and truth jet for Monte Carlo dijets (fig. 6.17), we see that the energy scale appears to have a certain dependence on η. As this plot does not take into account the outflow of particles from the jet cone, or gluon radiation, the η dependence might be stronger than what is seen in fig. 6.17.

The same effect could be the reason for the double peak in fig. 6.4. This could also be the reason why there is an unexpected flat peak in fig. 6.18 and 6.19. Although the flat peak might also be a consequence of final state gluon radiation not rejected by our back-to-back cut[38]. The last three examples though, could be consequences of the possible ϕ asymmetry discussed in the previous section.

Now, if we look at the RMS distribution in η for the energy fraction of reconstructed and truth jet (fig. 6.20), we see that the resolution gets worse in the region where the precission measurments of the electromagnetic calorimeter are less accurate (ref. Chapter 3.2.4).

The structure observed for the RMS distribution in η for the energy fraction of reconstructed and truth jet (fig. 6.20) can also be seen for the RMS distribution in η for the reconstruction of the W mass (fig. 6.21).

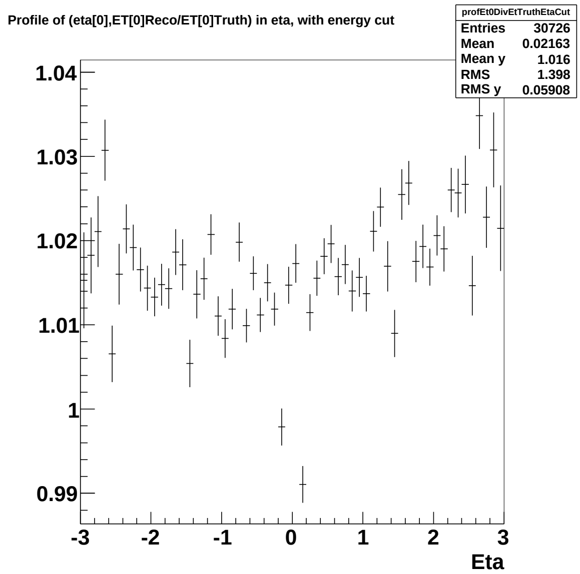

Figure 6.17: Profile plot of the energy fraction of reconstructed and truth jet in eta, with a 100GeV cut on transverse energy, from Monte Carlo dijets

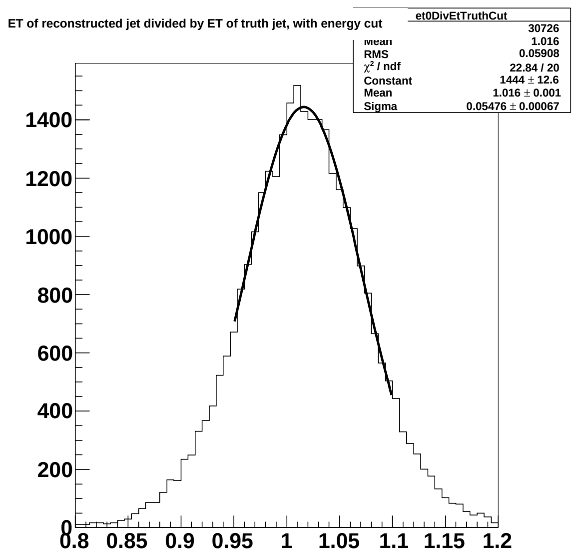

Figure 6.18: Plot of the energy fraction of reconstructed and truth jet with a 100GeV cut on transverse energy, from Monte Carlo dijets

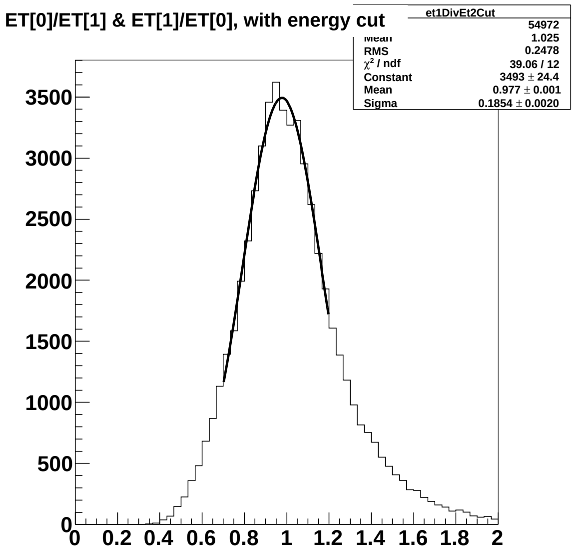

Figure 6.19: Plot of energy fraction of leading jets with a 100GeV cut on transverse energy, from Monte Carlo dijets

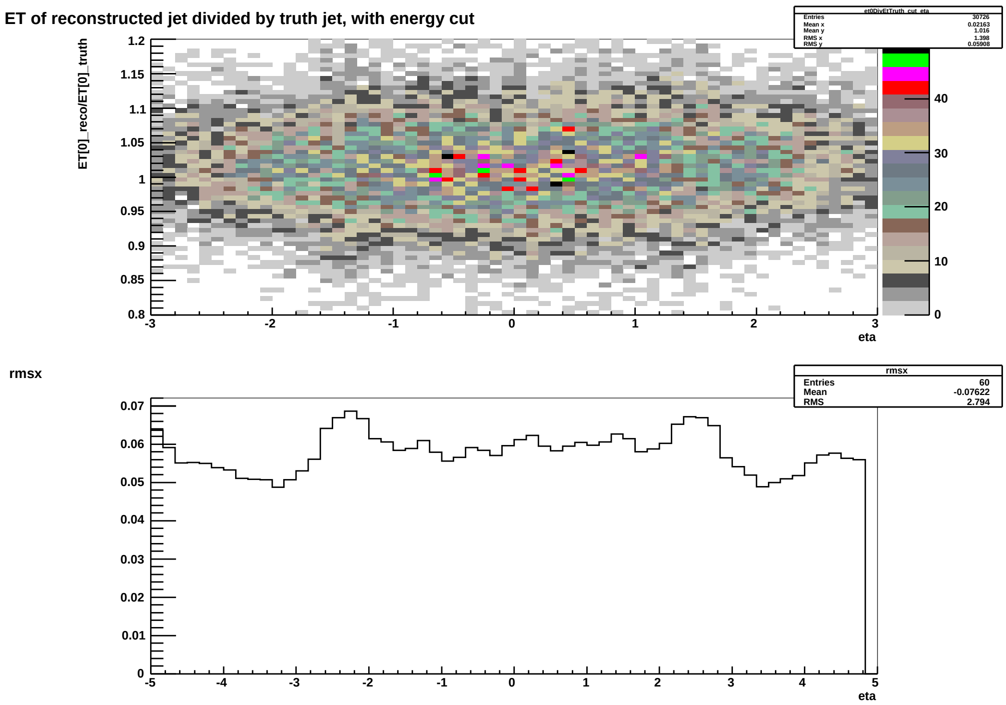

Figure 6.20: RMS profile in η for the energy fraction of reconstructed and truth jet with a 100GeV cut on transverse energy, from Monte Carlo dijets

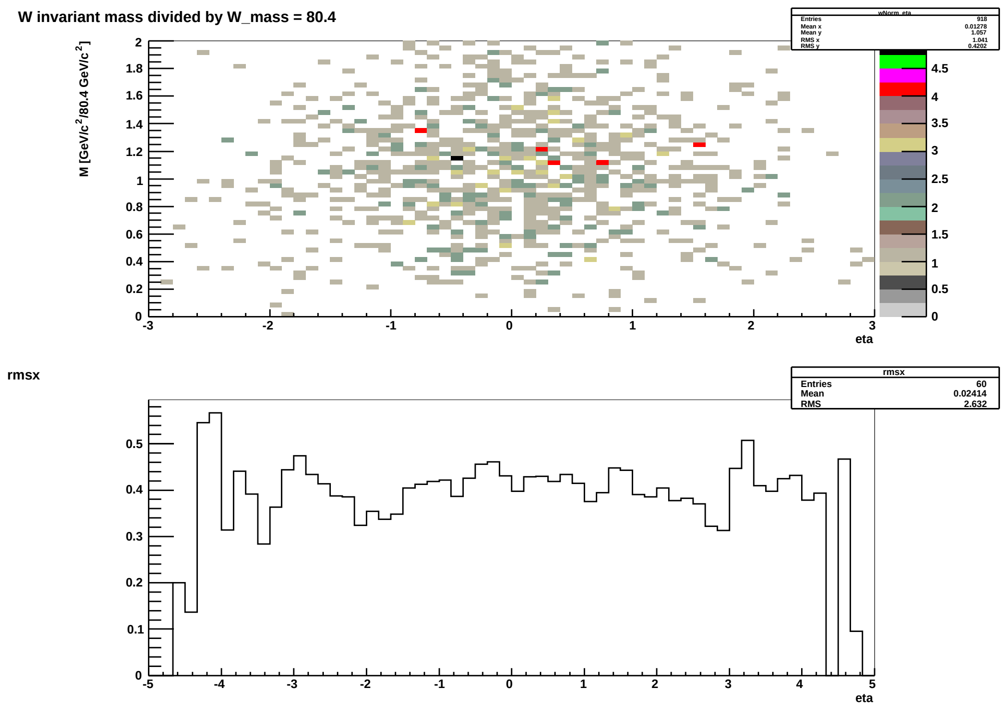

Figure 6.21: RMS profile in η for the reconstructed W mass divided by 80.4, from Monte Carlo top events

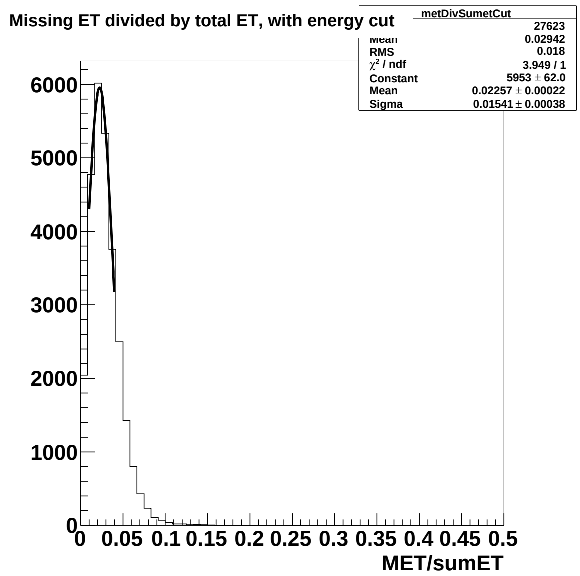

Figure 6.22: Plot of the missing energy resolution, with a 100GeV cut on transverse energy, from Monte Carlo dijets

6.9.2 Jet Resolution Estimates From Dijets

First we look at the missing transverse energy resolution plotted according to variable (6.8) (fig. 6.22). These results are from dijets produced from Monte Carlo simulation and contains very little real missing transverse energy.This because one needs neutrinos to get real missing transverse energy, unless one is dealing with SUSY particles, which is not the case here. And neutrinos are only present in events dealing with b and c jets. Yet the mean is at approximately 0.02, and not 0. So we seem to loose approximately 2% of visible transverse energy, probably due to reconstruction. In other words, we create 2% missing transverse energy from visible transverse energy.

Looking at the profile plot in fig. 6.22, we see that the amount of missing transverse energy increases at high η, as expected.

Comparing this with the resolution of vectorial sum over momenta (fig. 6.24), we observe that the vectorial sum over jet momenta gives approximately 1% more missing transverse energy. This is because the missing energy resolution takes into account particles that are left out from the jet definition. Those particles are not included in the resolution of vectorial sum over momenta.

Now, if we look at the energy balance for Monte Carlo dijets, with a 100GeV transverse energy cut on both leading and next to leading jet, (fig. 6.9) with a Gaussian fit, we obtain a width of approximately 0.200 ± 0.002.

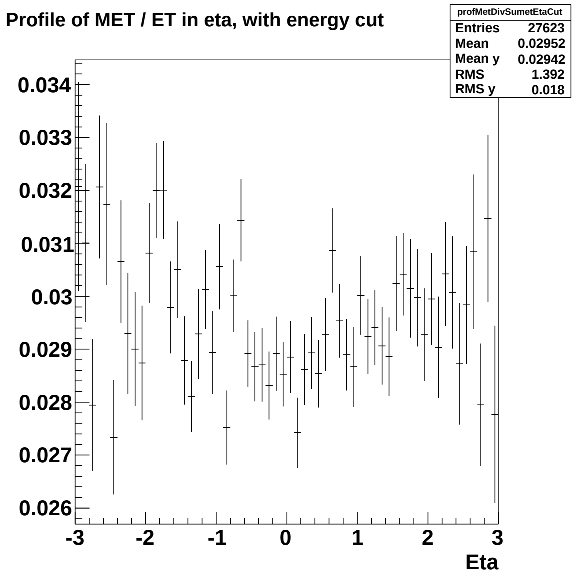

Figure 6.23: Profile plot of the missing energy resolution, with a 100GeV cut on transverse energy, from Monte Carlo dijets

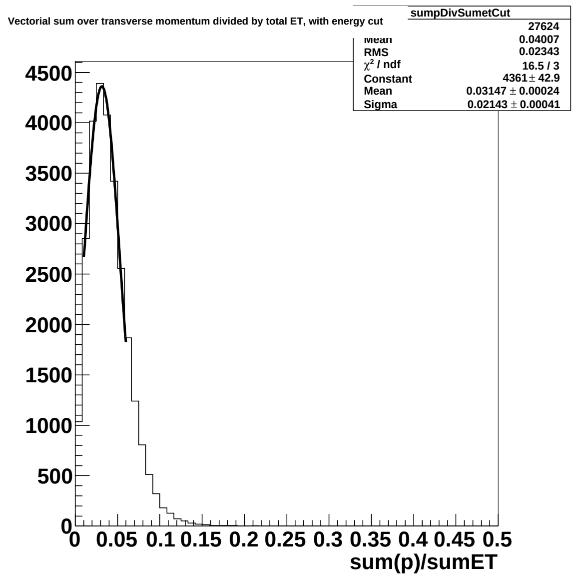

Figure 6.24: Plot of the resolution of vectorial sum over momenta with a 100GeV cut on transverse energy, from Monte Carlo dijets

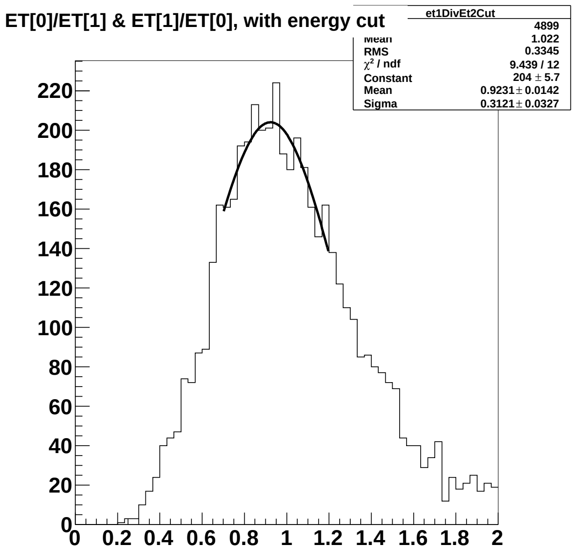

Figure 6.25: Plot of the energy fraction of leading jets with a 100GeV cut on transverse energy, from Monte Carlo top events

The width of the energy fraction of leading jets for the same dataset, and a 100GeV E_{T} cut on the leading jet is approximately 0.185 ± 0.002. So the energy balance and the energy fraction between leading and next to leading jet gives essentially the same result and can be considered as the same variable.

Comparing these results with the corresponding results for the energy fraction of reconstructed and truth jet (fig. 6.18) ($\sigma = 0.0548 \pm 0.0007$), we see that there is a difference by a factor of 4. This is assumed to be due to the energy fraction of reconstructed and truth jet not beeing affected by the outflow of particles from the jet cone, or by gluon radiation, both of which affects the energy fraction of leading jets. I.e., the jet definition together with radiation of partons leads to a difference between the variables. That means that variable energy fraction of reconstructed and truth jet, that was used for optimiziation of jet reconstruction, perhaps should rather be considered a resolution for transverse energy of hadronic particles in the jet, and less a jet resolution.

Looking at the energy fraction of leading jets for Monte Carlo top events (fig. 6.25), we see that the width is much larger ($\sigma = 0.31 \pm 0.03$ for top, compared to $\sigma = 0.185 \pm 0.002$ for dijets).

This is as expected, since the jets we are looking at is not necessarily dijet events, and therefore can not be expected to have opposite momenta in the

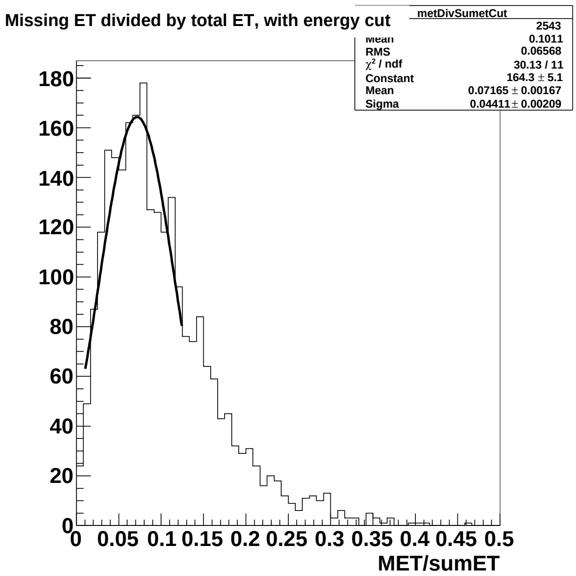

Figure 6.26: Plot of the missing energy resolution, with a 100GeV cut on transverse energy, from Monte Carlo top events

transverse plane. One way of fixing this is to take a closer look at the dijet event selection used in this thesis.

As expected, there is a width difference when comparing dijet variables for Monte Carlo dijets and Monte Carlo top events. This is because there are no dijets in top events. Top events can decay into W's and b quarks, both of which can decay into neutrinos. The presence of neutrinos means that jets can be back-to-back, without actually being dijets, i.e. they are not expected to have the same transverse energy. The neutrinos show up in the plots of the missing energy resolution (fig. 6.26) and the resolution for vectorial sum over momenta (fig. 6.27).

Here we see that there is more missing transverse energy in top events than in dijet events (mean is shifted by approximately 0.05 (compare with fig. 6.22)). This is due to the neutrinos from W decays

Looking at the energy fraction of leading jets for FDR2 data, which contains mostly dijets, we still get a broader distribution than for dijets ($\sigma = 0.214 \pm 0.003$ for FDR2 and $\sigma = 0.185 \pm 0.002$ for dijets) (see fig. 6.28 and 6.19).

This result supports the review of the dijet definition used in this thesis before looking at real data.

Missing transverse energy in FDR2 is similar to the result for Monte Carlo dijets (fig. 6.26 and 6.22). This is as expected since there should not be much

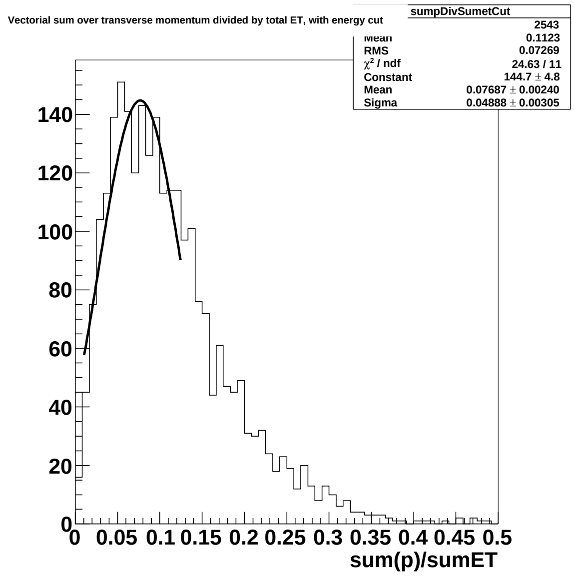

Figure 6.27: Plot of the resolution of vectorial sum over momenta with a 100GeV cut on transverse energy, from Monte Carlo top events

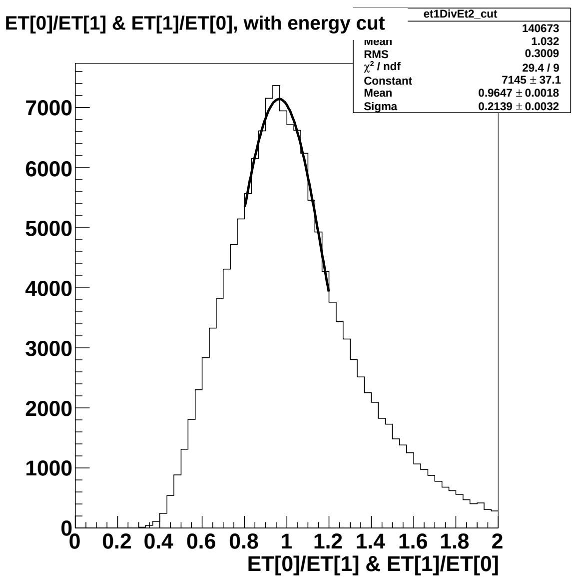

Figure 6.28: Plot of energy fraction of two leading jets with a 100GeV cut on transverse energy, from FDR2 data

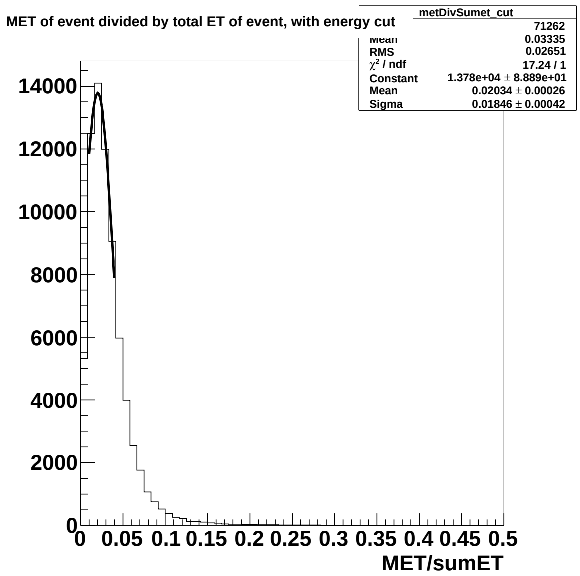

Figure 6.29: Plot of the missing energy resolution, with a 100GeV cut on transverse energy, from FDR2 data

real missing transverse energy in the events put into FDR2 data.

6.9.3 W Mass Resolution and Jet Resolution

In the following, the W mass is calculated from the two non b-tagged jets that gives a mass closest to the W mass.

In the approximation that jets are massless particles, we can find a relation between the W mass resolution and the jet transverse energy resolution[29]. For decays to massless particles $1, 2$, we have

$$M_W = \sqrt{(E_1 + E_2)^2 - (\vec{p_1} + \vec{p_2})^2} = \sqrt{2E_1E_2(1 - \cos\theta_{12})} \tag{6.11}$$

where θ_{12} is the angle between the particles.

Thus, neglecting $\cos\theta_{12}$ and assume

$$\frac{\Delta E_1}{E_1} = \frac{\Delta E_2}{E_2} = \frac{\Delta E}{E} \tag{6.12}$$

we have

$$\frac{\Delta M_W}{M_W} = \frac{\Delta E}{E} = \frac{\Delta E_{\mathrm{T}}}{E_{\mathrm{T}}} \tag{6.13}$$

which should be valid for small $|\eta|$.

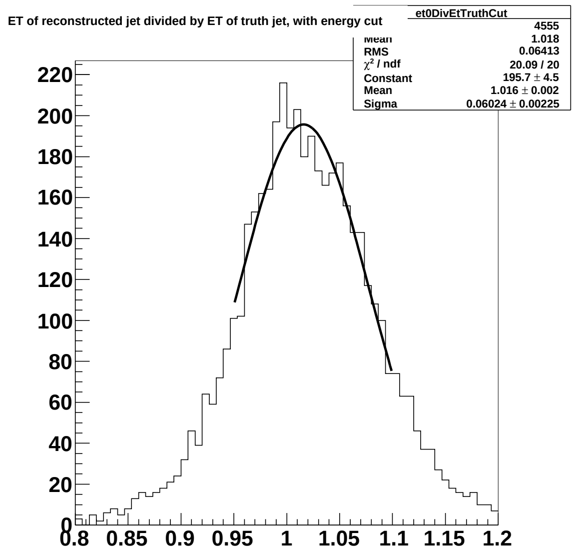

Figure 6.30: Plot of the energy fraction of reconstructed and truth jet with a 100GeV cut on transverse energy, from Monte Carlo top events

However, the angle for jets (the angle $\cos\theta_{12}$ resolution) cannot be neglected. And besides, the jets will have mass due to fragmentation effects.

Thus, the formula for jets is

$$M_W = \sqrt{m_1^2 + m_2^2 + 2E_1E_2 - 2|\vec{p_1}||\vec{p_2}|\cos\theta_{12}} \tag{6.14}$$

So then the jet mass resolution enters into the formula as well.

If we look at the energy fraction of reconstructed and truth jet for top events, we see that the mean is 1.016 ± 0.002 (fig. 6.30). That means that the jet energy scale is calibrated too high, and we thereby get too high a W mass. Since we reconstruct the W mass from two jets, we expect that the reconstructed W mass divided by 80.4GeV should have a mean at 1.03.

If we look at the plot of the reconstructed W mass divided by 80.4GeV, we see that the mean is at 1.06 ± 0.03.

What we observe is that the width is $\sigma = 0.40 \pm 0.06$ for W mass distribution from Monte Carlo top events (fig. 6.31) and $\sigma = 0.185 \pm 0.002$ for the energy fraction of leading jets in Monte Carlo dijet events (fig. 6.19). I.e., the width for the W mass distribution from top events is approximately twice of the width for the energy fraction of leading jets in dijet events.

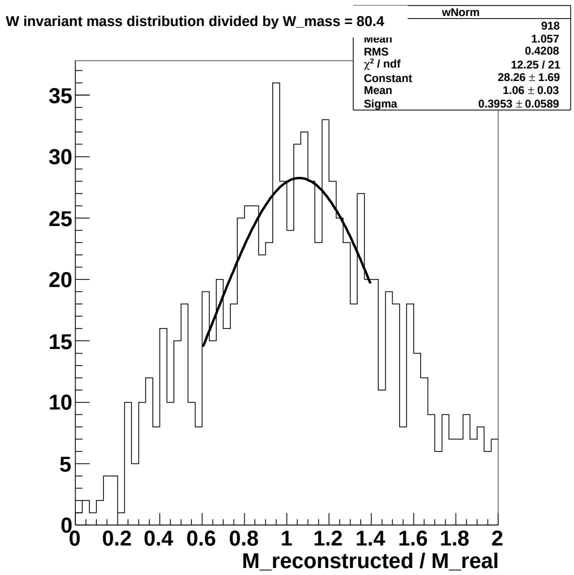

Figure 6.31: Plot of reconstructed W mass divided by 80.4GeV, from Monte Carlo top events

6.10 Conclusion

Based on the plot of the energy balance for Monte Carlo dijets, it was decided to introduce a 100GeV cut on the transverse energy for relevant jets in connection with the dijet-based variables. The resulting profile of the energy balance suggested that we should only look at the region $-3 \leq \eta \leq 3$. This study seemed to suggest some kind of ϕ asymmetry, which was supported by several other plots. Currently the reason for this is unknown, but one can speculate if it might be due to statistics, although it seems unlikely, some feature of the Monte Carlo generators or an actual asymmetry in the detector.

We reconstructed the W mass and found that it's width is approximately twice the width of the energy fraction between the two leading jets. The jet estimation methods we have studied gives jet resolution estimates of the same order as the mass resolution for W.

However, these methods are not the same as the method used for jet reconstruction optimization (variable (6.3)). There we found a difference of a factor 4. That is because the fraction between reconstructed and truth jet is rather an estimate of the resolution of jet particles, not jets.

The jet reconstruction algorithms is creating some false missing transverse energy.

It is also apparent that missing transverse energy in ATLAS is not just the vector sum over jet momenta.

Chapter 7

Summary and Conclusions

A short overview of the standard model of particle physics was given, then an introduction to CERN, LHC and one of the experiments associated with it, ATLAS. The thesis then discusses data formats and simulations, and further gives an introduction to jet reconstruction algorithms.

To obtain the results presented in the final chapter, the program DiJet was written by Kent Olav Skjei and Thomas Burgess. DiJet studies several variables for dijets (energy balance, energy fraction, fraction of reconstructed jet energy divided by the energy of the truth jet), it reconstruct the W mass, it looks at missing transverse energy and vector sum over jet momenta and distribution of jets.

Due to low statistics, we decided to cut out the regions $|\eta| \geq 3$.

We found an asymmetry in ϕ that showed up in both the variables used for jet resolution estimates, in the jet distributions and when studying missing transverse energy. For Monte Carlo dijets and FDR2, there were most leading jets in positive ϕ. For Monte Carlo top events the situation was the opposite. There we found most leading jets in negative ϕ.

We have seen that the energy scale has a dependency on η.

We used Gaussian fits to estimate the widths and means of the distributions in this thesis

We have seen that the jet reconstruction algorithms creates a certain amount of false missing transverse energy (approximately 2%). When comparing this with the vector sum over jet momenta, it became apparent that the concept of missing transverse energy in the data sets contain more information than just the sum of jet momenta.

The energy fraction of reconstructed and truth jet gave a width of $\sigma = 0.0548 \pm 0.0007$ for Monte Carlo dijets. This is a factor of about 4 less than the results for the energy fraction between the two leading jets in Monte Carlo dijets. The latter having a width of $\sigma = 0.185 \pm 0.002$. So the energy fraction of reconstructed and truth jet is an estimation of the resolution of jet particles, while the energy fraction between the two leading jets seem to be more suitable for estimating the jet resolution.

When comparing the results for Monte Carlo dijets with the W mass distribution from Monte Carlo top events with a width of $\sigma = 0.40 \pm 0.06$, we see that the width of the latter is about twice the width of the first.

Comparing the width of the energy fraction of the two leading jets for Monte Carlo dijets ($\sigma = 0.185 \pm 0.002$) and FDR2 ($\sigma = 0.236 \pm 0.006$), we see that the latter is too wide, meaning that our dijet selection algorithm should be reviewed before applied to real data.

Chapter 8

Appendix 1: DiJet Documentation

DiJet Reference Manual

DiJet-00-00-01

Generated by Doxygen 1.3.9.1

Sat Sep 26 19:46:14 2009

8.1 DiJet

Author:

Kent Olav Skjei <kent.skjei-at-gmail-com>
Thomas Burgess <tburgess-at-cern-ch>

8.1.1 Introduction

DiJet is a dual use ARA and Athena tool - to use in ARA try share/DiJet_-root.C and to use in Athena try share/DiJet_topOptions.py

DiJetAraTool(p. 80) contains the main code and documentation for this project.

```
package DiJet
author Thomas Burgess <tburgess-at-cern-ch>

use AtlasPolicy    AtlasPolicy-*
use GaudiInterface GaudiInterface-*    External
use AraTool        AraTool-*           PhysicsAnalysis
use JetEvent       JetEvent-*          Reconstruction/Jet
use egammaEvent    egammaEvent-*       Reconstruction/egamma
use MissingETEvent MissingETEvent-*    Reconstruction
use muonEvent      muonEvent-*         Reconstruction/MuonIdentification

library DiJet *.cxx -s=components *.cxx
apply_pattern component_library
apply_pattern dual_use_library files=*.cxx
apply_pattern declare_python_modules files="*.py"
private
use AtlasReflex   AtlasReflex-*        External -no_auto_imports
apply_pattern lcgdict dict=DiJet selectionfile=selection.xml headerfiles=
                                                     "..\/DiJet/DiJetDict.h"
end_private
```

8.2 DiJet Directory Documentation

8.2.1 /afs/cern.ch/user/t/tburgess/scratch0/testarea_-14.5.2/DiJet/src/components/ Directory Reference

Files

- file **DiJet_entries.cxx**

  DiJet packade entries declaration.

- file **DiJet_load.cxx**

  DiJet packade load file.

8.2.2 /afs/cern.ch/user/t/tburgess/scratch0/testarea_-14.5.2/DiJet/DiJet/ Directory Reference

Files

- file **DiJetAraTool.h**

  Definition of DiJet Athena Root Access Tool.

- file **DiJetAraToolAlg.h**

  Implementation of ATHENA Algorithm for DiJet Athena Root Access Tool.

- file **DiJetAraToolWrapper.h**

  Definition of wrapper for **DiJetAraTool**(p. 80).

- file **DiJetDict.h**

  DiJet packade dictionary file.

8.2.3 /afs/cern.ch/user/t/tburgess/scratch0/testarea_-14.5.2/DiJet/share/ Directory Reference

Files

- file **DiJet_topOptions.py**
- file **init_root.py**

8.2.4 /afs/cern.ch/user/t/tburgess/scratch0/testarea_-14.5.2/DiJet/src/ Directory Reference

Directories

- directory**components**

Files

- file **DiJetAraTool.cxx**

  Implementation of DiJet Athena Root Access Tool.

- file **DiJetAraToolAlg.cxx**

  Definition of ATHENA Algorithm for DiJet Athena Root Access Tool.

- file **DiJetAraToolWrapper.cxx**

  Implementation of wrapper for **DiJetAraTool**(p. 80).

8.3 DiJet Class Documentation

8.3.1 DiJetAraTool Class Reference

DiJet Athena Root Access tool.

`#include <DiJetAraTool.h>`

Public Member Functions

- **DiJetAraTool** (PropertyMgr ∗pmgr=0)

  Constructor.

- virtual ∼**DiJetAraTool** ()

  Virtual destructor.

- virtual StatusCode **initialize** ()

  Initialize.

- virtual StatusCode **finalize** ()

  Finalize.

- void **bookHistograms** ()

  Book histograms, call before event loop.

- void **updateCollections** (const JetCollection ∗jets, const JetCollection ∗truthJets, const ElectronContainer ∗electrons, const Analysis::MuonContainer ∗muons, const MissingET ∗missingET)

Set the collection used in mainLoop.

- StatusCode **eachEvent** ()

  Main function, call inside event loop.

- StatusCode **araEventLoop** (TTree *tree)

  Event loop if running in ARA mode.

SetFunctions @{

Set functions for options (used for ARA)

- void **setHistoFile** (std::string histoFile)

  Set output histogram file name.

- void **setDoDiJetCut** (bool doDiJetCut)

  Set flag for doing diJetCuts.

- void **setCosAlfaCut** (bool cosAlfaCut)

  Set cut to define back to back jets in cos alfa.

- void **setEtaJetCutLow** (double etaJetCutUpLow)

  Set eta leading jet cut low value.

- void **setEtaJetCutUp** (double etaJetCutUp)

  Set eta leading jet cut upper value.

- void **setPhiJetCutLow** (double phiJetCutLow)

  Set phi leading jet cut lower value.

- void **setPhiJetCutUp** (double phiJetCutUp)

  Set phi leading jet cut upper value.

- void **setScaleEtaPhiRange** (bool flag)

  Set if histogram scale should change with eta phi range.

- void **setEtaRangeLow** (double etaRangeLow)

  Set leading jet eta lower range value.

- void **setEtaRangeUp** (double etaRangeUp)

Set leading jet eta upper range value.

- void **setPhiRangeLow** (double phiRangeLow)
  Set leading jet phi lower range value.

- void **setPhiRangeUp** (double phiRangeUp)
  Set leading jet phi upper range value.

- void **setCut_et0DivTruthLo** (double cut_et0DivTruthLo)
  Set leading jet et by truth et cut value.

- void **setCut_et0DivTruthHi** (double cut_et0DivTruthHi)
  Set leading jet et by truth et cut value.

- void **setCut_et1DivEt2** (double cut_et1DivEt2)
  Set leading jet by next to leading jet ratio cut value.

- void **setCut_sumPtDivEt** (double cut_sumPtDivEt)
  Set P_t / E_t cut value.

- void **setCut_metDivSumEt** (double cut_metDivSumEt)
  Set E_t^miss / E_t cut value.

- void **setCut_wNorm** (double cut_wNorm)
  Set fraction of W Mass cut value.

Private Member Functions

- void **book2DHistos** (TH2 ∗&eta, TH2 ∗&phi, TH2 ∗&etacut, TH2 ∗&phicut, std::string name, std::string title, std::string xaxis, double ylow, double yup, double etalow, double etaup, double philow, double phiup)

  Book a set of 2d histograms (helper for bookhistograms).

- TH1 ∗ **makeRmsProfile** (TH2 ∗h, TProfile ∗&profx) const

  Helper function to make RMS profile from 2d histogram.

- void **write2DHistogram** (TFile ∗f, TH2 ∗h_eta, TH2 ∗h_phi) const

Helper function to write 2D histograms.

- void **writeHistograms** () const

  Write histogram file.

- void **fillCosAlfaPlots** ()

  Fill cos alfa plots.

- template<class T> const T * **getEventObject** (std::string name, TTree *tree, Long64_t ievent) const

  Get a object from branch in tree for an event.

- bool **checkIfDiJetEvent** () const

  Check if Jet collection is valid DiJet.

- double **calcDeltaR** (double phi1, double phi2, double eta1, double eta2) const

  Calculate delta R.

- void **fillLeadinJetsEtEtaPhiPxyz** ()

  Fill et eta,phi and Px,y,z of leading and next to leading jet.

- int **countGoodElectrons** () const

  Count number of good electrons in container.

- int **countGoodMuons** () const

  Count number of good muons in container.

- void **countGoodJets** (int &pjet_good_N, int &bjet_good_N) const

  Count number of good jets in container.

- void **calcWMass** ()

  Calculate WMass and fill W histograms.

- void **fillHighestEnergySymmetry** ()

  Fill highest energysymmetry.

- void **fillEnergyDifferenceBasis** (TH2 *ediffEta, TH2 *ediffPhi, TH2 *et1DivEt2Eta, TH2 *et1DivEt2Phi) const

  Fill energy difference (used for both _cut and normal histos).

- void **fillEnergyDifference** () const

  Fill energy difference.

- void **fillEtDivEtTruth** ()

  Fill et0 of reconstructed jet divided by et0 of truth jet.

- void **fillNumberDistribution** () const

  Fill number distribution plots.

Private Attributes

- bool **m_booked**

  True if histograms are booked (to avoid seg faults for 0 histograms).

- TObjString * **m_fileInfo**

  String with event id:s.

- unsigned int **m_nEvt**

  Number of total event.

- unsigned int **m_nDiJetEvt**

  Number of events that passes di jet cut.

- unsigned int **m_nEtaPhiEvt**

  Number of events that passes eta phi range cut.

AthenaCollections

Containers with all necessary per event data

- const JetCollection * **m_jets**

  Jets.

- const JetCollection ∗ **m_truthJets**
  Truth Jets.

- const ElectronContainer ∗ **m_electrons**
  Electrons.

- const Analysis::MuonContainer ∗ **m_muons**
  Muons.

- const MissingET ∗ **m_missingET**
  Missing ET.

JetInfo

Information on jets in the events

- double **m_et0**
  et of leading jet

- double **m_eta0**
  eta of leading jet

- double **m_phi0**
  phi of leading jet

- double **m_px0**
  Px of leading jet.

- double **m_py0**
  Py of leading jet.

- double **m_pz0**
  Pz of leading jet.

- double **m_et1**
  et of next to leading jet

- double **m_eta1**
  eta of next to leading jet

- double **m_phi1**
  phi of next to leading jet
- double **m_px1**
  Px of next to leading jet.
- double **m_py1**
  Py of next to leading jet.
- double **m_pz1**
  Pz of next to leading jet.
- double **m_nrJets**
  Number of jets above m_jetCut in the event.

Options

- std::string **m_histoFile**
  File name of histogram output file (option).
- bool **m_doDiJetCut**
  True if we should do diJet cut (option).
- double **m_cut_cosAlfa**
  Cut on the angle between jets, default (-0.92).
- double **m_etaJetCutLow**
  Lower cut on eta.
- double **m_etaJetCutUp**
  Upper cut on eta.
- double **m_phiJetCutLow**
  Lower cut on phi.
- double **m_phiJetCutUp**
  Upper cut on phi.
- bool **m_scaleEtaPhiRange**
  True if scaling histo range with eta phi range.

- double **m_etaRangeLow**
  Lower cut on eta range.
- double **m_etaRangeUp**
  Upper cut on eta range.
- double **m_phiRangeLow**
  Lower cut on phi range.
- double **m_phiRangeUp**
  Upper cut on phi range.
- double **m_jetCut**
  For definition of jet.
- double **m_energyCut**
  To look at high energy jets.
- unsigned int **m_nbins**
  Number of bins.
- double **m_cut_et0DivTruthLo**
  Et0 over Et Truth cut (0).
- double **m_cut_et0DivTruthHi**
  Et0 over Et Truth cut (0).
- double **m_cut_et1DivEt2**
  Et1 / Et2 cut (0).
- double **m_cut_sumPtDivEt**
  Momentum sum / E_t sum cut (0).
- double **m_cut_metDivSumEt**
  Missing et / sum et cut (0).
- double **m_cut_wNorm**
  W normalized mass cut (0).

energydifference

Normalized energy difference

- TH2 ∗ **m_energydifference_eta**
- TH2 ∗ **m_energydifference_phi**
- TH2 ∗ **m_energydifference_cut_eta**
- TH2 ∗ **m_energydifference_cut_phi**

et1DivEt2

Fraction of energy of highest energy jet to energy of next to highest energy jet & fraction of energy of next to highest energy jet to energy of highest energy jet

- TH2 ∗ **m_et1DivEt2_eta**
- TH2 ∗ **m_et1DivEt2_phi**
- TH2 ∗ **m_et1DivEt2_cut_eta**
- TH2 ∗ **m_et1DivEt2_cut_phi**

et0DivEtTruth

ET of reconstructed jet divided by truth jet

- TH2 ∗ **m_et0DivEtTruth_eta**
- TH2 ∗ **m_et0DivEtTruth_phi**
- TH2 ∗ **m_et0DivEtTruth_cut_eta**
- TH2 ∗ **m_et0DivEtTruth_cut_phi**

metDivSumet

MET of event divided by total ET of event, with cuts

- TH2 ∗ **m_metDivSumet_eta**
- TH2 ∗ **m_metDivSumet_phi**
- TH2 ∗ **m_metDivSumet_cut_eta**
- TH2 ∗ **m_metDivSumet_cut_phi**

sumpDivSumet

Vectorial sum over momentum divided by total ET

- TH2 ∗ **m_sumpDivSumet_eta**
- TH2 ∗ **m_sumpDivSumet_phi**
- TH2 ∗ **m_sumpDivSumet_cut_eta**
- TH2 ∗ **m_sumpDivSumet_cut_phi**

wNorm

W invariant mass distribution

- TH2 * **m_wNorm_eta**
- TH2 * **m_wNorm_phi**
- TH2 * **m_wNorm_cut_eta**
- TH2 * **m_wNorm_cut_phi**

otherHistograms

Histograms filled by DiJet

- TH1F * **m_highestEnergySymmetry**

  Counts number of highest energy jets in positive and negative phi.

- TH1F * **m_numberdistributionEta**

  Distribution of highest energy jets in eta.

- TH1F * **m_numberdistributionEtaW**

  Distribution of highest energy jets in eta weighted.

- TH1F * **m_numberdistributionPhi**

  Distribution of highest energy jets in phi.

- TH1F * **m_numberdistributionPhiW**

  Distribution of highest energy jets in phi weighted.

- TH1F * **m_cos**

  Distribution of cosine between leading and next to leading jet, no z component.

- TH1F * **m_cosCut**

  Distribution of cosine between leading and next to leading jet with dijet cut, no z component.

- TH1F * **m_cosz**

  Distribution of cosine between leading and next to leading jet.

- TH1F * **m_coszCut**

  Distribution of cosine between leading and next to leading jet.

Detailed Description

DiJet Athena Root Access tool.

Note: you may need to compile muonEvent and JetEvent before DiJet-AraTool

This is a dual use ARA & Athena tool, it can can be run in athena using a python script like share/DiJet_topOptions.py or it can be run in ROOT using ARA using the CINT script share/DiJet_root.C

The tool produces the following sets of 2D histograms (histogram name, title/help text, histogram X-title, range) energydifference

- Normalized energy difference (ET[pos_phi]-ET[neg_phi])/ET[0] -1, 1 et1DivEt2
- Fraction of energy of highest energy jet to energy of next to highest energy jet & fraction of energy of next to highest energy jet to energy of highest energy jet
- ET[0]/ET[1] & ET[1]/ET[0] 0,m_cut_et1DivEt2 et0DivEtTruth
- ET of reconstructed jet divided by truth jet
- ET[0]_reco/ET[0]_truth
- m_cut_et0DivTruthLo,m_cut_et0DivTruthHi metDivSumet
- MET of event divided by total ET of event
- MET/sumET
- 0,m_cut_metDivSumEt sumpDivSumet
- Vectorial sum over momentum divided by total ET
- sum(p)/sumET
- 0,m_cut_sumPtDivEt wNorm
- W invariant mass divided by W_mass = 80.4
- M [GeV/c^{2}/80.4 GeV/c^{2}]
- 0, m_cut_wNorm

Each set contains one histogram of X versus eta and one versus phi In addition the projection in Y (i.e. the 1D histogram of X), the profile in X (the means in Y of each X bin) and the RMS profile (the RMS of Y in each X bin). For each histogram an additional histogram with a cut on leading jet energy is produced (except in the case of wNorm)

Has the following options (default value in parenthesis)

- histoFile : Histogram File (DiJetHistograms.root)
- doDiJetCut : True if we should do a DiJetCut (Require two back to back jets) (true)
- cut_cosAlfa : Cut on the angle between jets, default (-0.92) Cuts to remove events with leading jet in an eta/phi region (to kill noisy towers)
- EtaJetCutLow : Lower Eta jet Cut (-5)
- EtaJetCutUp : Upper Eta jet Cut (5)
- PhiJetCutLow : Lower Phi jet Cut (-4)
- PhiJetCutUp : Upper Phi jet Cut (4) Cuts to remove events with leading jet outside an eta region (to kill tails)
- scaleEtaPhiRange: scale histogram ranges with eta phi region (true)
- EtaRangeLow : Lower Eta range (0)
- EtaRangeUp : Upper Eta range (0)
- PhiRangeLow : Lower Phi range (0)
- PhiRangeUp : Upper Phi range (0) Cuts that skips filling of certain histograms
- cut_et0DivTruthLo : Et0 over Et Truth cut (0.8)
- cut_et0DivTruthHi : Et0 over Et Truth cut (1.2)
- cut_et1DivEt2 : Et1 / Et2 cut (2.0)
- cut_sumPtDivEt : Momentum sum / E_t sum cut (0.5)
- cut_metDivSumEt : Missing et / sum et cut (0.5)
- cut_wMass : W mass cut (160.8)

- cut_wNorm : W normalized mass cut (2.0)

Follows the AraToolExample closely
Definition at line 103 of file DiJetAraTool.h.

Constructor & Destructor Documentation

DiJetAraTool::DiJetAraTool (PropertyMgr ∗ *pmgr* = 0)

Constructor.

Parameters:

pmgr Property manager pointer

Definition at line 32 of file DiJetAraTool.cxx.

```
32                                                     :
33      AraToolBase(pmgr),
34      m_booked(false),
35      //Athena containers
36      m_jets(0),
37      m_truthJets(0),
38      m_electrons(0),
39      m_muons(0),
40      m_missingET(0),
41      //Jet information
42      m_et0(0),
43      m_eta0(0),
44      m_phi0(0),
45      m_px0(0),
46      m_py0(0),
47      m_pz0(0),
48      m_et1(0),
49      m_eta1(0),
50      m_phi1(0),
51      m_px1(0),
52      m_py1(0),
53      m_pz1(0),
54      m_nrJets(0),
55      //Cuts/options
56      m_histoFile("DiJetHistograms.root"),
57      m_doDiJetCut(true),
58      m_cut_cosAlfa(-0.92),
59      m_etaJetCutLow(0),
60      m_etaJetCutUp(0),
61      m_phiJetCutLow(0),
62      m_phiJetCutUp(0),
63      m_scaleEtaPhiRange(true),
64      m_etaRangeLow(-5),
```

```
m_etaRangeUp(5),
m_phiRangeLow(-4),
m_phiRangeUp(4),
m_jetCut(20*GeV),
m_energyCut(100*GeV),
m_nbins(60),
m_cut_et0DivTruthLo(0.8),
m_cut_et0DivTruthHi(1.2),
m_cut_et1DivEt2(2.0),
m_cut_sumPtDivEt(0.5),
m_cut_metDivSumEt(0.5),
m_cut_wNorm(2.0),
m_fileInfo(new TObjString),
//Counters
m_nEvt(0),
m_nDiJetEvt(0),
m_nEtaPhiEvt(0),
//2d histograms
m_energydifference_eta(0),
m_energydifference_phi(0),
m_energydifference_cut_eta(0),
m_energydifference_cut_phi(0),
m_et1DivEt2_eta(0),
m_et1DivEt2_phi(0),
m_et1DivEt2_cut_eta(0),
m_et1DivEt2_cut_phi(0),
m_et0DivEtTruth_eta(0),
m_et0DivEtTruth_phi(0),
m_et0DivEtTruth_cut_eta(0),
m_et0DivEtTruth_cut_phi(0),
m_metDivSumet_eta(0),
m_metDivSumet_phi(0),
m_metDivSumet_cut_eta(0),
m_metDivSumet_cut_phi(0),
m_sumpDivSumet_eta(0),
m_sumpDivSumet_phi(0),
m_sumpDivSumet_cut_eta(0),
m_sumpDivSumet_cut_phi(0),
m_wNorm_eta(0),
m_wNorm_phi(0),
//Other histograms
m_highestEnergySymmetry(0),
m_numberdistributionEta(0),
m_numberdistributionEtaW(0),
m_numberdistributionPhi(0),
m_numberdistributionPhiW(0),
m_cos(0),
m_cosCut(0),
m_cosz(0),
```

```
    m_coszCut(0)
{
    declareProperty( "EtaJetCutLow", m_etaJetCutLow, "Lower Eta jet Cut" );
    declareProperty( "EtaJetCutUp", m_etaJetCutUp, "Upper Eta jet Cut " );
    declareProperty( "PhiJetCutLow", m_phiJetCutLow, "Lower Phi jet Cut" );
    declareProperty( "PhiJetCutUp", m_phiJetCutUp, "Upper Phi jet Cut " );
    declareProperty( "ScaleEtaPhiRange", m_scaleEtaPhiRange,
                     "enable scaling of histogram bounds with eta phi range" );
    declareProperty( "EtaRangeLow", m_etaRangeLow, "Lower Eta range" );
    declareProperty( "EtaRangeUp", m_etaRangeUp, "Upper Eta range" );
    declareProperty( "PhiRangeLow", m_phiRangeLow, "Lower Phi range" );
    declareProperty( "PhiRangeUp", m_phiRangeUp, "Upper Phi range" );
    declareProperty( "histoFile", m_histoFile, "Histogram File" );
    declareProperty( "doDiJetCut", m_doDiJetCut,
                     "True if we should do a DiJetCut" );
    declareProperty( "cut_et0DivTruthLo",
                     m_cut_et0DivTruthLo, "Et0 over Et Truth cut" );
    declareProperty( "cut_et0DivTruthHi",
                     m_cut_et0DivTruthHi, "Et0 over Et Truth cut" );
    declareProperty( "cut_et1DivEt2", m_cut_et1DivEt2, "Et1 / Et2 cut" );
    declareProperty( "cut_sumPtDivEt",
                     m_cut_sumPtDivEt, "Momentum sum / E_t sum cut" );
    declareProperty( "cut_metDivSumEt",
                     m_cut_metDivSumEt, "Missing et / sum et cut" );
    declareProperty( "cut_wNorm", m_cut_wNorm, "w normalized mass cut" );
    declareProperty( "cut_cosAlfa", m_cut_cosAlfa, "Cut on Dijet Angle" );
}
```

DiJetAraTool::∼DiJetAraTool () `[virtual]`

Virtual destructor.

Definition at line 142 of file DiJetAraTool.cxx.

```
{
    //2d histograms
    delete m_energydifference_eta;
    delete m_energydifference_phi;
    delete m_energydifference_cut_eta;
    delete m_energydifference_cut_phi;
    delete m_et1DivEt2_eta;
    delete m_et1DivEt2_phi;
    delete m_et1DivEt2_cut_eta;
    delete m_et1DivEt2_cut_phi;
    delete m_et0DivEtTruth_eta;
    delete m_et0DivEtTruth_phi;
    delete m_et0DivEtTruth_cut_eta;
    delete m_et0DivEtTruth_cut_phi;
    delete m_metDivSumet_eta;
```

```
    delete m_metDivSumet_phi;
    delete m_metDivSumet_cut_eta;
    delete m_metDivSumet_cut_phi;
    delete m_sumpDivSumet_eta;
    delete m_sumpDivSumet_phi;
    delete m_sumpDivSumet_cut_eta;
    delete m_sumpDivSumet_cut_phi;
    delete m_wNorm_eta;
    delete m_wNorm_phi;
    delete m_highestEnergySymmetry;
    delete m_numberdistributionEta;
    delete m_numberdistributionEtaW;
    delete m_numberdistributionPhi;
    delete m_numberdistributionPhiW;
    delete m_cos;
    delete m_cosCut;
    delete m_cosz;
    delete m_coszCut;
}
```

Member Function Documentation

StatusCode DiJetAraTool::initialize () [virtual]

Initialize.

Definition at line 178 of file DiJetAraTool.cxx.

```
{
    return StatusCode::SUCCESS;
}
```

StatusCode DiJetAraTool::finalize () [virtual]

Finalize.

Definition at line 183 of file DiJetAraTool.cxx.

```
{
    writeHistograms();
    return StatusCode::SUCCESS;
}
```

void DiJetAraTool::bookHistograms ()

Book histograms, call before event loop.

Definition at line 334 of file DiJetAraTool.cxx.

```
{
    //Set flag that histos have been booked
    m_booked = true;
```

```

    //Set histograms ranges
    double etarng=5;
    double phirng=TMath::Pi();
    double etarangelow=-etarng;
    double etarangeup=etarng;
    double phirangelow=-phirng;
    double phirangeup=phirng;
    if (m_scaleEtaPhiRange) {
        etarangelow = (m_etaRangeLow>-etarng)?m_etaRangeLow:-etarng;
        etarangeup = (m_etaRangeUp<etarng)?m_etaRangeUp:etarng;
        phirangelow = (m_phiRangeLow>-phirng)?m_phiRangeLow:-phirng;
        phirangeup  = (m_phiRangeUp<phirng)?m_phiRangeUp:phirng;
    }

    //Book 2d histograms

    // === energydifference ===========================================
    book2DHistos(
        m_energydifference_eta,m_energydifference_phi,
        m_energydifference_cut_eta,m_energydifference_cut_phi,
        "energydifference","Normalized energy difference",
        "(ET[pos_phi]-ET[neg_phi])/ET[0]",
        -1.0,1.0,etarangelow,etarangeup,phirangelow,phirangeup);

    // === et1DivEt2==================================================
    book2DHistos(
        m_et1DivEt2_eta,m_et1DivEt2_phi,
        m_et1DivEt2_cut_eta,m_et1DivEt2_cut_phi,
        "et1DivEt2","ET[0]/ET[1] & ET[1]/ET[0]","ET[0]/ET[1] & ET[1]/ET[0]",
        0.0,m_cut_et1DivEt2,etarangelow,etarangeup,phirangelow,phirangeup);

    // === et0DivEtTruth ==============================================
    book2DHistos(
        m_et0DivEtTruth_eta,m_et0DivEtTruth_phi,
        m_et0DivEtTruth_cut_eta,m_et0DivEtTruth_cut_phi,
        "et0DivEtTruth",
        "ET of reconstructed jet divided by truth jet",
        "ET[0]_reco/ET[0]_truth",
        m_cut_et0DivTruthLo,m_cut_et0DivTruthHi,
        etarangelow,etarangeup,phirangelow,phirangeup);

    // === metDivSumet ================================================
    book2DHistos(
        m_metDivSumet_eta,m_metDivSumet_phi,
        m_metDivSumet_cut_eta,m_metDivSumet_cut_phi,
        "metDivSumet","MET of event divided by total ET of event","MET/sumET",
        0.0,m_cut_metDivSumEt,etarangelow,etarangeup,phirangelow,phirangeup);

```

```
    // === sumpDivSumet ===============================================
    book2DHistos(
        m_sumpDivSumet_eta,m_sumpDivSumet_phi,
        m_sumpDivSumet_cut_eta,m_sumpDivSumet_cut_phi,
        "sumpDivSumet",
        "Vectorial sum over momentum divided by total ET","sum(p)/sumET",
        0.0,m_cut_sumPtDivEt,etarangelow,etarangeup,phirangelow,phirangeup);

    // === w ==========================================================
    TH2* wNormDummy_eta=0; //We do not produce cut histograms for wNorm
    TH2* wNormDummy_phi=0;
    book2DHistos(
        m_wNorm_eta,m_wNorm_phi,
        wNormDummy_eta,wNormDummy_phi,
        "wNorm","W invariant mass divided by W_mass = 80.4",
        "M [GeV/c^{2}/80.4 GeV/c^{2}]",
        0.0,m_cut_wNorm,etarangelow,etarangeup,phirangelow,phirangeup);
    delete wNormDummy_eta;
    delete wNormDummy_phi;

    //Book other histograms
    m_highestEnergySymmetry = new TH1F(
        "highestEnergySymmetry",
        "Nr et_0 in pos (1) & neg (-1) phi",
        m_nbins,-2,2);
    m_numberdistributionEta = new TH1F(
        "numberdistributionEta",
        "Distribution of highest energy jets in eta",
        m_nbins,etarangelow,etarangeup);
    m_numberdistributionEta->SetXTitle("Eta");
    m_numberdistributionEtaW = new TH1F(
        "numberdistributionEtaW",
        "Distribution of highest energy jets in eta weighted",
        m_nbins,etarangelow,etarangeup);
    m_numberdistributionEtaW->SetXTitle("Eta");
    m_numberdistributionPhi = new TH1F(
        "numberdistributionPhi","Distribution of highest energy jets in phi",
        m_nbins,phirangelow,phirangeup);
    m_numberdistributionPhi->SetXTitle("Phi");
    m_numberdistributionPhiW = new TH1F(
        "numberdistributionPhiW",
        "Distribution of highest energy jets in phi weighted",
        m_nbins,phirangelow,phirangeup);
    m_numberdistributionPhiW->SetXTitle("Phi");
    m_cos = new TH1F(
        "cos",
        "Distribution of cosine between leading "
        "and next to leading jet, no z component",
        m_nbins,-1.3,1.3);
```

```
    m_cosCut = new TH1F(
        "cosCut",
        "Distribution of cosine between leading "
        "and next to leading jet, no z component, with dijet cut",
        m_nbins,-1.3,1.3);
    m_cosz = new TH1F(
        "cosz",
        "Distribution of cosine between leading and next to leading jet",
        m_nbins,-1.3,1.3);
    m_coszCut = new TH1F(
        "coszCut",
        "Distribution of cosine between leading "
        "and next to leading jet with dijet cut",
        m_nbins,-1.3,1.3);
}
```

void DiJetAraTool::updateCollections (const JetCollection ∗ *jets*, const JetCollection ∗ *truthJets*, const ElectronContainer ∗ *electrons*, const Analysis::MuonContainer ∗ *muons*, const MissingET ∗ *missingET*)

Set the collection used in mainLoop.
Must be done once per event

Parameters:

jets Jet container "Cone4H1TowerJets"

truthJets MC Truth jet container "Cone4TruthJets"

electrons Electron container "ElectronAODCollection"

muons Muon container "StacoMuonCollection"

missingET missing ET container "MET_Final"

Definition at line 189 of file DiJetAraTool.cxx.

```
{
    m_jets = jets;
    m_truthJets = truthJets;
    m_electrons = electrons;
    m_muons = muons;
    m_missingET = missingET;
}
```

StatusCode DiJetAraTool::eachEvent ()

Main function, call inside event loop.
Note that collections needs to be updated before each call!

Returns:

SUCCESS if nothing bad happens, FAILURE if there are no histograms booked or the collections are not filled

Definition at line 203 of file DiJetAraTool.cxx.

```
{
    if (! m_booked ) {
        std::cerr<<"ERROR DiJetAraTool::eachEvent failed: "
            " histograms not booked!" << std::endl;
        return StatusCode::FAILURE;
    }
    //Check collections
    if ((m_jets==0) || (m_electrons==0) || (m_muons==0) || (m_missingET==0)) {
        std::cerr<<"ERROR DiJetAraTool::eachEvent failed: "
            " one or more collections missing!"<< std::endl;
        return StatusCode::FAILURE;
    }
    //Increase event counter
    m_nEvt++;

    //Fill et eta,phi and Px,y,z of leading and next to leading jet
    fillLeadinJetsEtEtaPhiPxyz();

    //Perform the w mass calculation _before_ DiJetCut
    calcWMass();

    fillCosAlfaPlots();

    //Skip non DiJet events
    if (m_doDiJetCut)
        if (checkIfDiJetEvent() == 0) return StatusCode::SUCCESS;
    //Increase counter for number of events that survive DiJetCut
    m_nDiJetEvt++;

    //Skip events by eta phi cut on leading jet
    if ( (m_eta0>m_etaJetCutLow)&&(m_eta0<m_etaJetCutUp) &&
         (m_phi0>m_phiJetCutLow)&&(m_phi0<m_phiJetCutUp))
    {
        return StatusCode::SUCCESS;
    }
    //Skip events by eta range on leading jet
    if ( (m_eta0<m_etaRangeLow)||(m_eta0>m_etaRangeUp) )
    {
        return StatusCode::SUCCESS;
    }
    //Skip events by phi range on leading jet
    if ( (m_phi0<m_phiRangeLow)||(m_phi0>m_phiRangeUp) )
    {
```

```
        return StatusCode::SUCCESS;
    }
    //Increase counter for number of events that survive eta phi range cut
    m_nEtaPhiEvt++;

    fillHighestEnergySymmetry();
    fillEnergyDifference();
    //Only fill the truth info if there is truth
    if (m_truthJets!=0) fillEtDivEtTruth();
    fillNumberDistribution();

    return StatusCode::SUCCESS;
}
```

StatusCode DiJetAraTool::araEventLoop (TTree ∗ *tree*)

Event loop if running in ARA mode.

This replaces the old di_jet function, it will do the following

- bookHistograms
- get branches and set addresses
- loop over all entries
  - update collections
  - call eachEvent
- saveHistograms

Parameters:

tree Tree to loop over

Definition at line 453 of file DiJetAraTool.cxx.

```
{
    bookHistograms();
    if (tree==0) {
        std::cerr<<"ERROR DiJetAraTool::AraEventLoop failed: "
            " tree = 0 "<< std::endl;
        return StatusCode::FAILURE;
    }
    if (! m_booked ) {
        std::cerr<<"ERROR DiJetAraTool::AraEventLoop failed: "
            " histograms not booked!" << std::endl;
        return StatusCode::FAILURE;
    }
    //Event loop
```

```
    Long64_t nentries = tree->GetEntriesFast();
    for(Long64_t ientry=0; ientry < nentries; ientry++) {
        //Get info from TTree
        m_jets=getEventObject<JetCollection>(
            "Cone4H1TowerJets", tree, ientry);
        m_truthJets=getEventObject<JetCollection>(
            "Cone4TruthJets",tree, ientry);
        m_electrons=getEventObject<ElectronContainer>(
            "ElectronAODCollection", tree, ientry);
        m_muons=getEventObject<Analysis::MuonContainer>(
            "StacoMuonCollection",tree ,ientry);
        m_missingET=getEventObject<MissingET>(
            "MET_Final",tree,ientry);
        eachEvent();
    }
    return StatusCode::SUCCESS;
}
```

void DiJetAraTool::setHistoFile (std::string *histoFile*)

Set output histogram file name.

Parameters:

histoFile filename to set

Definition at line 924 of file DiJetAraTool.cxx.

```
{
    m_histoFile = histoFile;
}
```

void DiJetAraTool::setDoDiJetCut (bool *doDiJetCut*)

Set flag for doing diJetCuts.

Parameters:

doDiJetCut flag value to set

Definition at line 929 of file DiJetAraTool.cxx.

```
{
    m_doDiJetCut = doDiJetCut;
}
```

void DiJetAraTool::setCosAlfaCut (bool *cosAlfaCut*)
Set cut to define back to back jets in cos alfa.

Parameters:
cosAlfaCut cosine value

Definition at line 934 of file DiJetAraTool.cxx.

```
{
    m_cut_cosAlfa = cosAlfaCut;
}
```

void DiJetAraTool::setEtaJetCutLow (double *etaJetCutUpLow*)
Set eta leading jet cut low value.

Parameters:
etaJetCutUpLow eta value to set

Definition at line 939 of file DiJetAraTool.cxx.

```
{
    m_etaJetCutLow = etaJetCutLow;
}
```

void DiJetAraTool::setEtaJetCutUp (double *etaJetCutUp*)
Set eta leading jet cut upper value.

Parameters:
etaJetCutUp eta value to set

Definition at line 944 of file DiJetAraTool.cxx.

```
{
    m_etaJetCutUp = etaJetCutUp;
}
```

void DiJetAraTool::setPhiJetCutLow (double *phiJetCutLow*)
Set phi leading jet cut lower value.

Parameters:
phiJetCutLow phi value to set

Definition at line 949 of file DiJetAraTool.cxx.

```
{
    m_phiJetCutLow = phiJetCutLow;
}
```

void DiJetAraTool::setPhiJetCutUp (double *phiJetCutUp*)

Set phi leading jet cut upper value.

Parameters:

phiJetCutUp phi value to set

Definition at line 954 of file DiJetAraTool.cxx.

```
{
    m_phiJetCutUp = phiJetCutUp;
}
```

void DiJetAraTool::setScaleEtaPhiRange (bool *flag*)

Set if histogram scale should change with eta phi range.

Parameters:

flag true or false

Definition at line 959 of file DiJetAraTool.cxx.

```
{
    m_scaleEtaPhiRange = flag;
}
```

void DiJetAraTool::setEtaRangeLow (double *etaRangeLow*)

Set leading jet eta lower range value.

Parameters:

etaRangeLow eta value to set

Definition at line 964 of file DiJetAraTool.cxx.

```
{
    m_etaRangeLow = etaRangeLow;
}
```

void DiJetAraTool::setEtaRangeUp (double *etaRangeUp*)

Set leading jet eta upper range value.

Parameters:

etaRangeUp eta value to set

Definition at line 969 of file DiJetAraTool.cxx.

```
{
    m_etaRangeUp = etaRangeUp;
}
```

void DiJetAraTool::setPhiRangeLow (double *phiRangeLow*)

Set leading jet phi lower range value.

Parameters:

phiRangeLow phi value to set

Definition at line 974 of file DiJetAraTool.cxx.

```
975 {
976     m_phiRangeLow = phiRangeLow;
977 }
```

void DiJetAraTool::setPhiRangeUp (double *phiRangeUp*)

Set leading jet phi upper range value.

Parameters:

phiRangeUp phi value to set

Definition at line 979 of file DiJetAraTool.cxx.

```
980 {
981     m_phiRangeUp = phiRangeUp;
982 }
```

void DiJetAraTool::setCut_et0DivTruthLo (double *cut_et0DivTruthLo*)

Set leading jet et by truth et cut value.

Parameters:

cut_et0DivTruthLo ratio to set

Definition at line 984 of file DiJetAraTool.cxx.

```
985 {
986     m_cut_et0DivTruthLo = cut_et0DivTruthLo;
987 }
```

void DiJetAraTool::setCut_et0DivTruthHi (double *cut_et0DivTruthHi*)

Set leading jet et by truth et cut value.

Parameters:

cut_et0DivTruthHi ratio to set

Definition at line 989 of file DiJetAraTool.cxx.

```
990 {
991     m_cut_et0DivTruthHi = cut_et0DivTruthHi;
992 }
```

void DiJetAraTool::setCut_et1DivEt2 (double *cut_et1DivEt2*)

Set leading jet by next to leading jet ratio cut value.

Parameters:

cut_et1DivEt2 ratio to set

Definition at line 994 of file DiJetAraTool.cxx.

```
995 {
996     m_cut_et1DivEt2=cut_et1DivEt2;
997 }
```

void DiJetAraTool::setCut_sumPtDivEt (double *cut_sumPtDivEt*)

Set P_t / E_t cut value.

Parameters:

cut_sumPtDivEt value to set

Definition at line 999 of file DiJetAraTool.cxx.

```
1000 {
1001     m_cut_sumPtDivEt = cut_sumPtDivEt;
1002 }
```

void DiJetAraTool::setCut_metDivSumEt (double *cut_metDivSumEt*)

Set E_t^miss / E_t cut value.

Parameters:

cut_metDivSumEt ratio to set

Definition at line 1004 of file DiJetAraTool.cxx.

```
1005 {
1006     m_cut_metDivSumEt = cut_metDivSumEt;
1007 }
```

void DiJetAraTool::setCut_wNorm (double *cut_wNorm*)

Set fraction of W Mass cut value.

Parameters:

cut_wNorm fraction to set

Definition at line 1009 of file DiJetAraTool.cxx.

```
1010 {
1011     m_cut_wNorm = cut_wNorm;
1012 }
```

void DiJetAraTool::book2DHistos (TH2 *& *eta*, TH2 *& *phi*, TH2 *& *etacut*, TH2 *& *phicut*, std::string *name*, std::string *title*, std::string *xaxis*, double *ylow*, double *yup*, double *etalow*, double *etaup*, double *philow*, double *phiup*) `[private]`

Book a set of 2d histograms (helper for bookhistograms).

Parameters:

eta eta histogram

phi phi histogram

etacut eta histogram with energy cut

phicut phi histogram with energy cut

name base of name for histograms (_eta, _cut will be added)

title base of title for histograms (cut will be added)

ylow Lower Y range

yup Upper Y range

etalow Lower Eta range

etaup Upper Eta range

philow Lower Phi range

phiup Upper Phi range

Definition at line 293 of file DiJetAraTool.cxx.

```
{
    eta = new TH2D(
        (name+"_eta").c_str(),title.c_str(),
        m_nbins,etalow,etaup,m_nbins,ylow,yup);
    eta->SetXTitle("Eta");
    eta->SetYTitle(xaxis.c_str());

    phi = new TH2D(
        (name+"_phi").c_str(),title.c_str(),
        m_nbins,philow,phiup,m_nbins,ylow,yup);
    title+=", with energy cut";
    phi->SetXTitle("Phi");
    phi->SetYTitle(xaxis.c_str());

    etacut = new TH2D(
        (name+"_cut_eta").c_str(),title.c_str(),
        m_nbins,etalow,etaup,m_nbins,ylow,yup);
    etacut->SetXTitle("Eta");
    etacut->SetYTitle(xaxis.c_str());
```

```
    phicut = new TH2D(
        (name+"_cut_phi").c_str(),title.c_str(),
        m_nbins,philow,phiup,m_nbins,ylow,yup);
    phicut->SetXTitle("Phi");
    phicut->SetYTitle(xaxis.c_str());
}
```

TH1 ∗ DiJetAraTool::makeRmsProfile (TH2 ∗ *h*, TProfile ∗& *profx*) const `[private]`

Helper function to make RMS profile from 2d histogram.

Parameters:

h 2d histogram to process

profx Created x-profile RMS histogram for x-profile

Definition at line 612 of file DiJetAraTool.cxx.

```
{
    if ((h==0)||(profx==0)) return 0;
    int nbins=profx->GetNbinsX();
    double lo=profx->GetBinLowEdge(0);
    double hi=profx->GetBinLowEdge(nbins);
    std::string name="RMS_profile_";
    name+=h->GetName();
    TH1* RMSprof = new TH1D(
        name.c_str(),profx->GetTitle(),nbins,lo,hi);
    for (int bin=1; bin<nbins; ++bin)
    {
        double rms=h->ProjectionY("",bin-1,bin)->GetRMS();
        RMSprof->SetBinContent(bin,rms);
    }
    return RMSprof;
}
```

void DiJetAraTool::write2DHistogram (TFile ∗ *f*, TH2 ∗ *h_eta*, TH2 ∗ *h_phi*) const `[private]`

Helper function to write 2D histograms.

Generates a projection and call make RMS profile

Parameters:

f File to write to

eta eta histogram to write

phi phi histogram to write

Definition at line 632 of file DiJetAraTool.cxx.

```
636 {
637     if ((f==0)||(h_eta==0)||(h_phi==0)) return; //Avoid seg faults
638
639     //Make eta and phi X profiles
640     std::string name="Profile_{X}";
641     name+=h_eta->GetName();
642     TProfile* h_eta_profx = h_eta->ProfileX(name.c_str());
643     name="Profile_{X}";
644     name+=h_phi->GetName();
645     TProfile* h_phi_profx = h_phi->ProfileX(name.c_str());
646     TH1* h_eta_rms = makeRmsProfile(h_eta,h_eta_profx);
647     TH1* h_phi_rms = makeRmsProfile(h_phi,h_phi_profx);
648
649     //Make projection in Y, only for eta as it is the same in phi
650     std::string title = h_eta->GetTitle();
651     name = h_eta->GetName();
652     name=name.substr(0,name.size()-4); //trim off _eta
653     TH1* h_proj = h_eta->ProjectionY();
654     h_proj->SetName(name.c_str());
655     h_proj->SetTitle(title.c_str());
656
657     //proj, profx
658     h_eta->Write();
659     h_phi->Write();
660 }
```

void DiJetAraTool::writeHistograms () const `[private]`

Write histogram file.

Definition at line 662 of file DiJetAraTool.cxx.

```
663 {
664     if (! m_booked ) return;
665     TFile* f = new TFile(m_histoFile.c_str(),"recreate");
666     f->cd();
667
668     //2d histograms
669     write2DHistogram(
670         f,m_energydifference_eta,m_energydifference_phi);
671     write2DHistogram(
672         f,m_energydifference_cut_eta,m_energydifference_cut_phi);
673     write2DHistogram(
674         f,m_et1DivEt2_eta,m_et1DivEt2_phi);
675     write2DHistogram(
676         f,m_et1DivEt2_cut_eta,m_et1DivEt2_cut_phi);
677     write2DHistogram(
678         f,m_et0DivEtTruth_eta,m_et0DivEtTruth_phi);
```

```
679     write2DHistogram(
680         f,m_et0DivEtTruth_cut_eta,m_et0DivEtTruth_cut_phi);
681     write2DHistogram(
682         f,m_metDivSumet_eta,m_metDivSumet_phi);
683     write2DHistogram(
684         f,m_metDivSumet_cut_eta,m_metDivSumet_cut_phi);
685     write2DHistogram(
686         f,m_sumpDivSumet_eta,m_sumpDivSumet_phi);
687     write2DHistogram(
688         f,m_sumpDivSumet_cut_eta,m_sumpDivSumet_cut_phi);
689     write2DHistogram(
690         f,m_wNorm_eta,m_wNorm_phi);
691     f->cd();
692     //The rest of the histograms
693     if(m_highestEnergySymmetry!=0) m_highestEnergySymmetry->Write();
694     if(m_numberdistributionEta!=0) m_numberdistributionEta->Write();
695     if(m_numberdistributionEtaW!=0) m_numberdistributionEtaW->Write();
696     if(m_numberdistributionPhi!=0) m_numberdistributionPhi->Write();
697     if(m_numberdistributionPhiW!=0) m_numberdistributionPhiW->Write();
698     if(m_cos!=0) m_cos->Write();
699     if(m_cosz!=0) m_cosz->Write();
700     if(m_cosCut!=0) m_cosCut->Write();
701     if(m_coszCut!=0) m_coszCut->Write();
702     if (m_fileInfo!=0) {
703         std::stringstream s;
704         s << "nEvt=" << m_nEvt
705           << " nDiJetEvt=" << m_nDiJetEvt
706           << " nEtaPhiEvt="<< m_nEtaPhiEvt;
707         m_fileInfo->SetString(s.str().c_str());
708         m_fileInfo->Write();
709     }
710     f->Write();
711     f->Close();
712     delete f;
713 }
```

void DiJetAraTool::fillCosAlfaPlots () [private]

Fill cos alfa plots.

Definition at line 261 of file DiJetAraTool.cxx.

```
262 {
263     if ( (m_cos==0) || (m_cosz==0) ||(m_cosCut==0) || (m_coszCut==0)) return;
264     //Fill distribution of cosine between leading and next to leading jet
265     double P1z = TMath::Sqrt(m_px0*m_px0+m_py0*m_py0+m_pz0*m_pz0);
266     double P2z = TMath::Sqrt(m_px1*m_px1+m_py1*m_py1+m_pz1*m_pz1);
267     double P1noz = TMath::Sqrt(m_px0*m_px0+m_py0*m_py0);
268     double P2noz = TMath::Sqrt(m_px1*m_px1+m_py1*m_py1);
269     //Angle between jets
```

```
    const double cos_alphaz=(m_px0*m_px1+m_py0*m_py1+m_pz0*m_pz1)/(P1z*P2z);
    const double cos_alphanoz=(m_px0*m_px1+m_py0*m_py1)/(P1noz*P2noz);
    m_cos->Fill(cos_alphanoz);
    m_cosz->Fill(cos_alphaz);

    //Run the DiJet cut and exit function if there is no DiJet
    if (checkIfDiJetEvent() == 0) return;

    //Fill distribution of cosine between leading
    //and next to leading jet after dijetcut
    double P1zCut = TMath::Sqrt(m_px0*m_px0+m_py0*m_py0+m_pz0*m_pz0);
    double P2zCut = TMath::Sqrt(m_px1*m_px1+m_py1*m_py1+m_pz1*m_pz1);
    double P1nozCut = TMath::Sqrt(m_px0*m_px0+m_py0*m_py0);
    double P2nozCut = TMath::Sqrt(m_px1*m_px1+m_py1*m_py1);
    //Angle between jets
    const double cos_alphazCut=
        (m_px0*m_px1+m_py0*m_py1+m_pz0*m_pz1)/(P1zCut*P2zCut);
    const double cos_alphanozCut=
        (m_px0*m_px1+m_py0*m_py1)/(P1nozCut*P2nozCut);
    m_cosCut->Fill(cos_alphanozCut);
    m_coszCut->Fill(cos_alphazCut);
}
```

template<class T> const T∗ DiJetAraTool::getEventObject (std::string *name*, TTree ∗ *tree*, Long64_t *ievent*) const `[inline, private]`

Get a object from branch in tree for an event.

Parameters:

name Name of branch

tree tree to get branch from

ievent

Returns:

0 faliure, class otherwise

Definition at line 295 of file DiJetAraTool.h.

```
                            {
        if (tree==0) {
            std::cerr << "ERROR DiJetAraTool::getBranchAddress failed to get "
                      << name << " beacause tree is 0" << std::endl;
            return 0;
        }
        TBranch* br = tree->GetBranch(name.c_str());
        if (br==0) {
            std::cerr << "ERROR DiJetAraTool::getBranchAddress failed to get "
```

```
                        << name << " because branch is 0 " << std::endl;
            return 0;
        }
        T* ret = *((T**)br->GetAddress());
        if (ret==0) {
            std::cerr << "ERROR DiJetAraTool::getBranchAddress failed to get"
                      << name << " because address is 0" << std::endl;
            return 0;
        }
        if ( br->GetEntry(ievent) < 0 ) {
            std::cerr << "ERROR DiJetAraTool::getBranchAddress failed to get"
                      << name << " because TBranch::GetEntry I/O error"
                      << std::endl;
            return 0;
        }
        return ret;
    }
```

bool DiJetAraTool::checkIfDiJetEvent () const `[private]`

Check if Jet collection is valid DiJet.

Definition at line 485 of file DiJetAraTool.cxx.

```
{
    const double P1 = TMath::Sqrt(m_px0*m_px0+m_py0*m_py0);
    const double P2 = TMath::Sqrt(m_px1*m_px1+m_py1*m_py1);
    //Angle between jets
    const double cos_alpha=(m_px0*m_px1+m_py0*m_py1)/(P1*P2);
    return ((m_nrJets>=2) && (cos_alpha<m_cut_cosAlfa));
}
```

double DiJetAraTool::calcDeltaR (double *phi1*, double *phi2*, double *eta1*, double *eta2*) const `[private]`

Calculate delta R.

Definition at line 823 of file DiJetAraTool.cxx.

```
{
    double dphi=TMath::Abs(phi1-phi2);
    if (dphi>TMath::Pi()) dphi-=2*TMath::Pi();
    const double deta=eta1-eta2;
    return TMath::Sqrt(dphi*dphi+deta*deta);
}
```

void DiJetAraTool::fillLeadinJetsEtEtaPhiPxyz () `[private]`

Fill et eta,phi and Px,y,z of leading and next to leading jet.

Definition at line 879 of file DiJetAraTool.cxx.

```
{
    //Leading jet
    m_et0  = 0;
    m_eta0 = 0;
    m_phi0 = 0;
    m_px0  = 0;
    m_py0  = 0;
    m_pz0  = 0;
    //Next to leading jet
    m_et1  = 0;
    m_eta1 = 0;
    m_phi1 = 0;
    m_px1  = 0;
    m_py1  = 0;
    m_pz1  = 0;
    m_nrJets=0;
    if (m_jets==0) return;
    for( JetCollection::const_iterator jetItr = m_jets->begin();
         jetItr != m_jets->end();
         ++jetItr) {
        Jet* jet = (*jetItr);
        if (jet==0) continue;

        if( jet->et() > m_et0 ) {
            //Leading jet
            m_et0 = jet->et();
            m_eta0= jet->eta();
            m_phi0= jet->phi();
            m_px0 = jet->px();
            m_py0 = jet->py();
            m_pz0 = jet->pz();
        } else if(jet->et() > m_et1) {
            //Next to leading jet
            m_et1 = jet->et();
            m_eta1= jet->eta();
            m_phi1= jet->phi();
            m_px1 = jet->px();
            m_py1 = jet->py();
            m_pz1 = jet->pz();
        }
        if( jet->et() > m_jetCut) m_nrJets++;
    }
}
```

int DiJetAraTool::countGoodElectrons () const `[private]`

Count number of good electrons in container.

Definition at line 494 of file DiJetAraTool.cxx.

```
{
    if (m_electrons==0) return 0;
    int el_good_N = 0;
    for( ElectronContainer::const_iterator elItr = m_electrons->begin();
         elItr != m_electrons->end(); ++elItr) {
        Analysis::Electron* el = (*elItr);
        if (el==0) continue;
        const double isEM = el->isem(egammaPID::ElectronLoose);
        const double elEta = el->eta();
        const double elPT = el->pt();
        if ( (isEM == 0) &&
             (TMath::Abs(elEta) < 2.5 && elPT > 20e3))
            el_good_N++;
    }
    return el_good_N;
}
```

int DiJetAraTool::countGoodMuons () const `[private]`

Count number of good muons in container.

Definition at line 512 of file DiJetAraTool.cxx.

```
{
    if (m_muons==0) return 0;
    int mu_good_N = 0;
    for ( Analysis::MuonContainer::const_iterator muItr = m_muons->begin();
          muItr != m_muons->end(); ++muItr) {
        Analysis::Muon* mu = (*muItr);
        if (mu==0) continue;
        const double muEta = mu->eta();
        const double muPT = mu->pt();
        if ((TMath::Abs(muEta) < 2.5) && (muPT > 20e3))
            mu_good_N++;
    }
    return mu_good_N;
}
```

void DiJetAraTool::countGoodJets (int & *pjet_good_N*, int & *bjet_good_N*) const `[private]`

Count number of good jets in container.

Parameters:

- ***pjet_good_N*** Number of good non bjets to be filled
- ***bjet_good_N*** Number of good bjets to be filled

Definition at line 528 of file DiJetAraTool.cxx.

```
{
    pjet_good_N = 0;
    bjet_good_N = 0;
    if (m_jets==0) return;
    for (JetCollection::const_iterator jetItr = m_jets->begin();
         jetItr != m_jets->end(); ++jetItr) {
        Jet* jet = (*jetItr);
        if (jet==0) continue;
        const double jetEta = jet->eta();
        const double jetPT = jet->pt();
        if (TMath::Abs(jetEta) < 2.5 && jetPT > 40e3)
            if ( jet->getFlavourTagWeight() > 0.87 )
                bjet_good_N++;
            else
                pjet_good_N++;
    }
}
```

void DiJetAraTool::calcWMass () `[private]`

Calculate WMass and fill W histograms.

Definition at line 549 of file DiJetAraTool.cxx.

```
{
    if ((m_jets==0)||(m_wNorm_eta==0)||(m_wNorm_phi==0)) return;
    const int el_good_N = countGoodElectrons();
    const int mu_good_N = countGoodMuons();
    int pjet_good_N=0, bjet_good_N=0;
    countGoodJets(pjet_good_N, bjet_good_N);
    const double missET = m_missingET->et();
    if ( (el_good_N+mu_good_N<1) ||     //At least one good lepton
         (bjet_good_N<2) ||             //At least 2 good b-tagged jets
         (pjet_good_N<2) ||             //At least 2 good non b-tagged jets
         (missET<=20e3) )               //At least 20 GeV missing E_T
        return;
    double mass_W = 1e6; //Initial value
    double px_W = 0;
    double py_W = 0;
    double pz_W = 0;
    double E_W = 0;
    //Should really be class variables with set-funcions
    const double eta_range = 2.5;    //cut is |eta|<eta_range
    const double pt_min = 40e3;      //cut below this, in GeV
    const double bweight_max = 0.87; //cut above this
    for (JetCollection::const_iterator jetItr1 = m_jets->begin();
         jetItr1 != m_jets->end();
         ++jetItr1) {
        Jet* jet1 = (*jetItr1);
        if (jet1==0) continue;
```

```
576         //Skip high eta, low pt or non b tagged jets
577         if ((TMath::Abs(jet1->eta())>eta_range)||
578             (jet1->pt()<pt_min)||
579             (jet1->getFlavourTagWeight()>bweight_max)) continue;
580         for (JetCollection::const_iterator jetItr2 = jetItr1+1;
581              jetItr2 != m_jets->end();
582              ++jetItr2) {
583             Jet* jet2 = (*jetItr2);
584             if (jet2==0) continue;
585             //Skip high eta, low pt or non b tagged jets
586             if ((TMath::Abs(jet2->eta())>eta_range)||
587                 (jet2->pt()<pt_min)||
588                 (jet2->getFlavourTagWeight()>bweight_max)) continue;
589             const double W_px = jet1->px() + jet2->px();
590             const double W_py = jet1->py() + jet2->py();
591             const double W_pz = jet1->pz() + jet2->pz();
592             const double W_E = jet1->et() + jet2->et();
593             const double W_m=
594                 TMath::Sqrt(W_E*W_E-W_px*W_px-W_py*W_py-W_pz*W_pz);
595             if (TMath::Abs(W_m-80e3)<TMath::Abs(mass_W-80e3)) {
596                 mass_W = W_m;
597                 px_W = W_px;
598                 py_W = W_py;
599                 pz_W = W_pz;
600                 E_W = W_E;
601             }
602         }
603     }
604     //Fill histogram
605     const double wMassN = mass_W/1.0e3/80.4; //In GeV
606     if ((wMassN>0) && (wMassN<m_cut_wNorm)) {
607         m_wNorm_eta->Fill(m_eta0,wMassN);
608         m_wNorm_phi->Fill(m_phi0,wMassN);
609     }
610 }
```

void DiJetAraTool::fillHighestEnergySymmetry () [private]

Fill highest energysymmetry.

Definition at line 715 of file DiJetAraTool.cxx.

```
716 {
717     if((m_highestEnergySymmetry==0)||(m_highestEnergySymmetry==0))return;
718     if(m_et0 > m_jetCut) {
719         if(m_phi0 > 0) {
720             m_highestEnergySymmetry->Fill(1);
721         } else  {
722             m_highestEnergySymmetry->Fill(-1);
723         }
```

```
    }
}
```

void DiJetAraTool::fillEnergyDifferenceBasis (TH2 ∗ *ediffEta*, TH2 ∗ *ediffPhi*, TH2 ∗ *et1DivEt2Eta*, TH2 ∗ *et1DivEt2Phi*) const `[private]`

Fill energy difference (used for both _cut and normal histos).

Parameters:

ediffEta Energy difference vs eta

ediffPhi Energy difference vs phi

et1DivEt2Eta E_t ratio of leading and next to leading jet vs eta

et1DivEt2Phi E_t ratio of leading and next to leading jet vs phi

Definition at line 727 of file DiJetAraTool.cxx.

```
{
    if((ediffEta==0)||(ediffPhi==0)||
       (et1DivEt2Eta==0)||(et1DivEt2Phi==0)||
       (m_et0==0)||(m_et1==0)) return;
    const double et01 = m_et0/m_et1;
    if (et01<m_cut_et1DivEt2) {
        et1DivEt2Eta->Fill(m_eta0,et01);
        et1DivEt2Phi->Fill(m_phi0,et01);
    }
    const double et10 = m_et1/m_et0;
    if (et10<m_cut_et1DivEt2){
        et1DivEt2Eta->Fill(m_eta0,et10);
        et1DivEt2Phi->Fill(m_phi0,et10);
    }
    if (m_phi0>=0) {
        const double et01diff =(m_et0-m_et1)/m_et0;
        ediffEta->Fill(m_eta0,et01diff);
        ediffPhi->Fill(m_phi0,et01diff);
    } else {
        const double et10diff =(m_et1-m_et0)/m_et0;
        ediffEta->Fill(m_eta0,et10diff);
        ediffPhi->Fill(m_phi0,et10diff);
    }
}
```

void DiJetAraTool::fillEnergyDifference () const `[private]`

Fill energy difference.

Definition at line 758 of file DiJetAraTool.cxx.

```
{
    if ((m_jets==0) ||
        (m_metDivSumet_eta==0)||
        (m_metDivSumet_phi==0)||
        (m_metDivSumet_cut_eta==0)||
        (m_metDivSumet_cut_phi==0)||
        (m_sumpDivSumet_eta==0)||
        (m_sumpDivSumet_phi==0)||
        (m_sumpDivSumet_cut_eta==0)||
        (m_sumpDivSumet_cut_phi==0))
        return;

    double sumPx = 0;
    double sumPy = 0;
    double sumPz = 0;
    for( JetCollection::const_iterator jetItr = m_jets->begin();
         jetItr < m_jets->end(); ++jetItr)
    {
        const Jet* jet = (*jetItr);
        if (jet==0) continue;
        sumPx += jet->px();
        sumPy += jet->py();
        sumPz += jet->pz();
    }
    double sumEt = m_missingET->sumet();
    double metPhi = m_missingET->phi();
    double met = m_missingET->et();
    double sumPt = TMath::Sqrt( (sumPx*sumPx) + (sumPy*sumPy) );
    double sumPtPhi = TMath::ATan2(sumPy,sumPx);

    const double metDivSumEt=met/sumEt;
    const double sumPtDivSumEt=sumPt/sumEt;
    if(m_et1>m_jetCut) {
        fillEnergyDifferenceBasis(
            m_energydifference_eta,
            m_energydifference_phi,
            m_et1DivEt2_eta,
            m_et1DivEt2_phi);
        if ( metDivSumEt < m_cut_metDivSumEt ) {
            m_metDivSumet_eta->Fill(m_eta0,metDivSumEt);
            m_metDivSumet_phi->Fill(metPhi,metDivSumEt);
        }
        if ( sumPtDivSumEt < m_cut_sumPtDivEt ) {
            m_sumpDivSumet_eta->Fill(m_eta0,sumPtDivSumEt);
            m_sumpDivSumet_phi->Fill(sumPtPhi,sumPtDivSumEt);
        }
    }
    if(m_et1>m_energyCut) {
        fillEnergyDifferenceBasis(
```

```
            m_energydifference_cut_eta,
            m_energydifference_cut_phi,
            m_et1DivEt2_cut_eta,
            m_et1DivEt2_cut_phi);
        if ( metDivSumEt < m_cut_metDivSumEt ) {
            m_metDivSumet_cut_eta->Fill(m_eta0,metDivSumEt);
            m_metDivSumet_cut_phi->Fill(metPhi,metDivSumEt);
        }
        if ( sumPtDivSumEt < m_cut_sumPtDivEt ) {
            m_sumpDivSumet_cut_eta->Fill(m_eta0,sumPtDivSumEt);
            m_sumpDivSumet_cut_phi->Fill(sumPtPhi,sumPtDivSumEt);
        }
    }
}
```

void DiJetAraTool::fillEtDivEtTruth () [private]

Fill et0 of reconstructed jet divided by et0 of truth jet.

Definition at line 832 of file DiJetAraTool.cxx.

```
{
    if ((m_truthJets==0)||(m_et0DivEtTruth_eta==0)||(m_et0DivEtTruth_phi==0))
        return;
    double etTruth = 1;
    double deltaRmin= TMath::Sqrt((2*TMath::Pi())*(2*TMath::Pi())+10*10);
    for( JetCollection::const_iterator jetItr = m_truthJets->begin();
         jetItr < m_truthJets->end(); ++jetItr)
    {
        const Jet* tJet = (*jetItr);
        if (tJet==0) continue;
        double deltaR = calcDeltaR(m_phi0,tJet->phi(),m_eta0,tJet->eta());
        if( deltaR < deltaRmin )
        {
            deltaRmin = deltaR;
            etTruth = tJet->et();
        }
    }
    double etratio=0;
    if (etTruth>0) etratio = m_et0/etTruth;

    if ((etratio<m_cut_et0DivTruthLo)||
        (etratio>m_cut_et0DivTruthHi)) return;

    //Cut on et ratio
    m_et0DivEtTruth_eta->Fill(m_eta0,etratio);
    m_et0DivEtTruth_phi->Fill(m_phi0,etratio);
    if(m_et0 > m_energyCut)
    {
        m_et0DivEtTruth_cut_eta->Fill(m_eta0,etratio);
```

```
        m_et0DivEtTruth_cut_phi->Fill(m_phi0,etratio);
    }
}
```

void DiJetAraTool::fillNumberDistribution () const `[private]`

Fill number distribution plots.

Definition at line 866 of file DiJetAraTool.cxx.

```
{
    if((m_et0<=m_jetCut)||
       (m_numberdistributionEta==0)||
       (m_numberdistributionEtaW==0)||
       (m_numberdistributionPhi==0)||
       (m_numberdistributionPhiW==0)) return;
    m_numberdistributionEta->Fill(m_eta0);
    m_numberdistributionEtaW->Fill(m_eta0,m_et0);
    m_numberdistributionPhi->Fill(m_phi0);
    m_numberdistributionPhiW->Fill(m_phi0,m_et0);
}
```

Member Data Documentation

bool DiJetAraTool::m_booked `[private]`

True if histograms are booked (to avoid seg faults for 0 histograms).

Definition at line 374 of file DiJetAraTool.h.

const JetCollection∗ DiJetAraTool::m_jets `[private]`

Jets.

Use in eventloop: should be updated before calling eachEvent

Definition at line 382 of file DiJetAraTool.h.

const JetCollection∗ DiJetAraTool::m_truthJets `[private]`

Truth Jets.

Use in fillEtDivEtTruth

Definition at line 387 of file DiJetAraTool.h.

const ElectronContainer∗ DiJetAraTool::m_electrons `[private]`

Electrons.

Use in eventloop: should be updated before calling eachEvent

Definition at line 392 of file DiJetAraTool.h.

const Analysis::MuonContainer* DiJetAraTool::m_muons `[private]`

Muons.
Use in eventloop: should be updated before calling eachEvent
Definition at line 397 of file DiJetAraTool.h.

const MissingET* DiJetAraTool::m_missingET `[private]`
Missing ET.
Use in eventloop: should be updated before calling eachEvent
Definition at line 402 of file DiJetAraTool.h.

double DiJetAraTool::m_et0 `[private]`
et of leading jet
Definition at line 409 of file DiJetAraTool.h.

double DiJetAraTool::m_eta0 `[private]`
eta of leading jet
Definition at line 412 of file DiJetAraTool.h.

double DiJetAraTool::m_phi0 `[private]`
phi of leading jet
Definition at line 415 of file DiJetAraTool.h.

double DiJetAraTool::m_px0 `[private]`
Px of leading jet.
Definition at line 418 of file DiJetAraTool.h.

double DiJetAraTool::m_py0 `[private]`
Py of leading jet.
Definition at line 421 of file DiJetAraTool.h.

double DiJetAraTool::m_pz0 `[private]`
Pz of leading jet.
Definition at line 424 of file DiJetAraTool.h.

double DiJetAraTool::m_et1 `[private]`
et of next to leading jet
Definition at line 427 of file DiJetAraTool.h.

double DiJetAraTool::m_eta1 [private]

eta of next to leading jet

Definition at line 430 of file DiJetAraTool.h.

double DiJetAraTool::m_phi1 [private]

phi of next to leading jet

Definition at line 433 of file DiJetAraTool.h.

double DiJetAraTool::m_px1 [private]

Px of next to leading jet.

Definition at line 436 of file DiJetAraTool.h.

double DiJetAraTool::m_py1 [private]

Py of next to leading jet.

Definition at line 439 of file DiJetAraTool.h.

double DiJetAraTool::m_pz1 [private]

Pz of next to leading jet.

Definition at line 442 of file DiJetAraTool.h.

double DiJetAraTool::m_nrJets [private]

Number of jets above m_jetCut in the event.

Definition at line 445 of file DiJetAraTool.h.

std::string DiJetAraTool::m_histoFile [private]

File name of histogram output file (option).

Definition at line 452 of file DiJetAraTool.h.

bool DiJetAraTool::m_doDiJetCut [private]

True if we should do diJet cut (option).

Definition at line 455 of file DiJetAraTool.h.

double DiJetAraTool::m_cut_cosAlfa [private]

Cut on the angle between jets, default (-0.92).

Definition at line 458 of file DiJetAraTool.h.

double DiJetAraTool::m_etaJetCutLow [private]

Lower cut on eta.

Definition at line 461 of file DiJetAraTool.h.

double DiJetAraTool::m_etaJetCutUp [private]
Upper cut on eta.
Definition at line 464 of file DiJetAraTool.h.

double DiJetAraTool::m_phiJetCutLow [private]
Lower cut on phi.
Definition at line 467 of file DiJetAraTool.h.

double DiJetAraTool::m_phiJetCutUp [private]
Upper cut on phi.
Definition at line 470 of file DiJetAraTool.h.

bool DiJetAraTool::m_scaleEtaPhiRange [private]
True if scaling histo range with eta phi range.
Definition at line 473 of file DiJetAraTool.h.

double DiJetAraTool::m_etaRangeLow [private]
Lower cut on eta range.
Definition at line 476 of file DiJetAraTool.h.

double DiJetAraTool::m_etaRangeUp [private]
Upper cut on eta range.
Definition at line 479 of file DiJetAraTool.h.

double DiJetAraTool::m_phiRangeLow [private]
Lower cut on phi range.
Definition at line 482 of file DiJetAraTool.h.

double DiJetAraTool::m_phiRangeUp [private]
Upper cut on phi range.
Definition at line 485 of file DiJetAraTool.h.

double DiJetAraTool::m_jetCut [private]
For definition of jet.
Definition at line 488 of file DiJetAraTool.h.

double DiJetAraTool::m_energyCut [private]
To look at high energy jets.
Definition at line 491 of file DiJetAraTool.h.

unsigned int DiJetAraTool::m_nbins [private]

Number of bins.

Definition at line 494 of file DiJetAraTool.h.

double DiJetAraTool::m_cut_et0DivTruthLo [private]

Et0 over Et Truth cut (0).

Definition at line 497 of file DiJetAraTool.h.

double DiJetAraTool::m_cut_et0DivTruthHi [private]

Et0 over Et Truth cut (0).

Definition at line 500 of file DiJetAraTool.h.

double DiJetAraTool::m_cut_et1DivEt2 [private]

Et1 / Et2 cut (0).

Definition at line 503 of file DiJetAraTool.h.

double DiJetAraTool::m_cut_sumPtDivEt [private]

Momentum sum / E_t sum cut (0).

Definition at line 506 of file DiJetAraTool.h.

double DiJetAraTool::m_cut_metDivSumEt [private]

Missing et / sum et cut (0).

Definition at line 509 of file DiJetAraTool.h.

double DiJetAraTool::m_cut_wNorm [private]

W normalized mass cut (0).

Definition at line 512 of file DiJetAraTool.h.

TObjString∗ DiJetAraTool::m_fileInfo [private]

String with event id:s.

Definition at line 517 of file DiJetAraTool.h.

unsigned int DiJetAraTool::m_nEvt [private]

Number of total event.

Definition at line 520 of file DiJetAraTool.h.

unsigned int DiJetAraTool::m_nDiJetEvt [private]

Number of events that passes di jet cut.

Definition at line 523 of file DiJetAraTool.h.

unsigned int DiJetAraTool::m_nEtaPhiEvt [private]

Number of events that passes eta phi range cut.

Definition at line 526 of file DiJetAraTool.h.

TH2* DiJetAraTool::m_energydifference_eta [private]

Definition at line 530 of file DiJetAraTool.h.

TH2* DiJetAraTool::m_energydifference_phi [private]

Definition at line 531 of file DiJetAraTool.h.

TH2* DiJetAraTool::m_energydifference_cut_eta [private]

Definition at line 532 of file DiJetAraTool.h.

TH2* DiJetAraTool::m_energydifference_cut_phi [private]

Definition at line 533 of file DiJetAraTool.h.

TH2* DiJetAraTool::m_et1DivEt2_eta [private]

Definition at line 541 of file DiJetAraTool.h.

TH2* DiJetAraTool::m_et1DivEt2_phi [private]

Definition at line 542 of file DiJetAraTool.h.

TH2* DiJetAraTool::m_et1DivEt2_cut_eta [private]

Definition at line 543 of file DiJetAraTool.h.

TH2* DiJetAraTool::m_et1DivEt2_cut_phi [private]

Definition at line 544 of file DiJetAraTool.h.

TH2* DiJetAraTool::m_et0DivEtTruth_eta [private]

Definition at line 549 of file DiJetAraTool.h.

TH2* DiJetAraTool::m_et0DivEtTruth_phi [private]

Definition at line 550 of file DiJetAraTool.h.

TH2* DiJetAraTool::m_et0DivEtTruth_cut_eta [private]

Definition at line 551 of file DiJetAraTool.h.

TH2* DiJetAraTool::m_et0DivEtTruth_cut_phi [private]

Definition at line 552 of file DiJetAraTool.h.

TH2∗ DiJetAraTool::m_metDivSumet_eta [private]
Definition at line 557 of file DiJetAraTool.h.

TH2∗ DiJetAraTool::m_metDivSumet_phi [private]
Definition at line 558 of file DiJetAraTool.h.

TH2∗ DiJetAraTool::m_metDivSumet_cut_eta [private]
Definition at line 559 of file DiJetAraTool.h.

TH2∗ DiJetAraTool::m_metDivSumet_cut_phi [private]
Definition at line 560 of file DiJetAraTool.h.

TH2∗ DiJetAraTool::m_sumpDivSumet_eta [private]
Definition at line 565 of file DiJetAraTool.h.

TH2∗ DiJetAraTool::m_sumpDivSumet_phi [private]
Definition at line 566 of file DiJetAraTool.h.

TH2∗ DiJetAraTool::m_sumpDivSumet_cut_eta [private]
Definition at line 567 of file DiJetAraTool.h.

TH2∗ DiJetAraTool::m_sumpDivSumet_cut_phi [private]
Definition at line 568 of file DiJetAraTool.h.

TH2∗ DiJetAraTool::m_wNorm_eta [private]
Definition at line 573 of file DiJetAraTool.h.

TH2∗ DiJetAraTool::m_wNorm_phi [private]
Definition at line 574 of file DiJetAraTool.h.

TH2∗ DiJetAraTool::m_wNorm_cut_eta [private]
Definition at line 575 of file DiJetAraTool.h.

TH2∗ DiJetAraTool::m_wNorm_cut_phi [private]
Definition at line 576 of file DiJetAraTool.h.

TH1F∗ DiJetAraTool::m_highestEnergySymmetry [private]
Counts number of highest energy jets in positive and negative phi.
Definition at line 583 of file DiJetAraTool.h.

TH1F∗ DiJetAraTool::m_numberdistributionEta [private]

Distribution of highest energy jets in eta.

Definition at line 586 of file DiJetAraTool.h.

TH1F∗ DiJetAraTool::m_numberdistributionEtaW [private]

Distribution of highest energy jets in eta weighted.

Definition at line 589 of file DiJetAraTool.h.

TH1F∗ DiJetAraTool::m_numberdistributionPhi [private]

Distribution of highest energy jets in phi.

Definition at line 592 of file DiJetAraTool.h.

TH1F∗ DiJetAraTool::m_numberdistributionPhiW [private]

Distribution of highest energy jets in phi weighted.

Definition at line 595 of file DiJetAraTool.h.

TH1F∗ DiJetAraTool::m_cos [private]

Distribution of cosine between leading and next to leading jet, no z component.

Definition at line 599 of file DiJetAraTool.h.

TH1F∗ DiJetAraTool::m_cosCut [private]

Distribution of cosine between leading and next to leading jet with dijet cut, no z component.

Definition at line 603 of file DiJetAraTool.h.

TH1F∗ DiJetAraTool::m_cosz [private]

Distribution of cosine between leading and next to leading jet.

Definition at line 606 of file DiJetAraTool.h.

TH1F∗ DiJetAraTool::m_coszCut [private]

Distribution of cosine between leading and next to leading jet.

Definition at line 609 of file DiJetAraTool.h.

The documentation for this class was generated from the following files:

- /afs/cern.ch/user/t/tburgess/scratch0/testarea_14.5.2/DiJet/DiJet/**DiJetAraTool.h**
- /afs/cern.ch/user/t/tburgess/scratch0/testarea_14.5.2/DiJet/src/**DiJetAraTool.cxx**

8.3.2 DiJetAraToolAlg Class Reference

ATHENA Algorithm for DiJet Athena Root Access Tool.

`#include <DiJetAraToolAlg.h>`

Public Member Functions

- **DiJetAraToolAlg** (const std::string &name, ISvcLocator *pSvcLocator)

  Constructor.

- virtual StatusCode **initialize** ()

  Initialize.

- virtual StatusCode **execute** ()

  Execute.

- virtual StatusCode **finalize** ()

  Finalize.

Protected Attributes

- ToolHandle< **DiJetAraToolWrapper** > **m_tool**

  Tool.

- MsgStream **m_log**

  Message stream.

- StoreGateSvc * **m_storeGate**

  A handle on the Store Gate service for access to the Event Store.

Detailed Description

ATHENA Algorithm for DiJet Athena Root Access Tool.

See also:

DiJetAraTool(p. 80)

Definition at line 22 of file DiJetAraToolAlg.h.

Constructor & Destructor Documentation

DiJetAraToolAlg::DiJetAraToolAlg (const std::string & *name*, ISvcLocator ∗ *pSvcLocator*)

Constructor.

Parameters:

name name

pSvcLocator Service locator

Definition at line 18 of file DiJetAraToolAlg.cxx.

```
19                                                         :
20     Algorithm(name, pSvcLocator),
21     m_tool( "DiJetAraToolWrapper", this ),
22     m_log(msgSvc(),name),
23     m_storeGate(0)
24 {
25     declareProperty( "DiJetTool", m_tool, "DiJet tool" );
26 }
```

Member Function Documentation

StatusCode DiJetAraToolAlg::initialize () `[virtual]`

Initialize.

Returns:

status code

Definition at line 28 of file DiJetAraToolAlg.cxx.

```
28                                         {
29     // verify that our tool handle is pointing to an accessible tool
30     if ( m_tool.retrieve().isFailure() ) {
31         m_log << MSG::FATAL << "Failed to retrieve " << m_tool << endreq;
32         return StatusCode::FAILURE;
33     }
34 
35     //Get pointer to storegate service
36     if (service("StoreGateSvc", m_storeGate).isFailure()) {
37         m_log << MSG::ERROR
38               << "Unable to retrieve pointer to StoreGateSvc"
39              << endreq;
40         return StatusCode::FAILURE;
41     }
42 
43     m_tool->getTool()->bookHistograms();
44     return StatusCode::SUCCESS;
45 }
```

StatusCode DiJetAraToolAlg::execute () `[virtual]`

Execute.

Returns:

status code

Definition at line 47 of file DiJetAraToolAlg.cxx.

```
47                                     {
48      //Get collections from store gate
49      const JetCollection*  jets = 0;
50      if ((m_storeGate->retrieve(jets,"Cone4H1TowerJets")).isFailure()
51          || (jets==0)) {
52          m_log << MSG::WARNING
53                << "No Cone4H1TowerJets found in infile" << endreq;
54          return StatusCode::FAILURE;
55      }
56      //Get collections from store gate
57      const JetCollection*  truthJets = 0;
58      if ((m_storeGate->retrieve(truthJets,"Cone4TruthJets")).isFailure()
59          || (truthJets==0)) {
60          m_log << MSG::WARNING
61                << "No Cone4TruthJets found in infile
                     (acceptable when there is no MC info)" << endreq;
62      }
63      const ElectronContainer* electrons = 0;
64      if ((m_storeGate->retrieve(electrons,"ElectronAODCollection")).isFailure()
65          || (electrons==0)) {
66          m_log << MSG::WARNING
67                << "No ElectronAODCollection found in infile" << endreq;
68          return StatusCode::FAILURE;
69      }
70      const Analysis::MuonContainer*  muons = 0;
71      if ((m_storeGate->retrieve(muons,"StacoMuonCollection")).isFailure()
72          || (muons==0)) {
73          m_log << MSG::WARNING
74                << "No StacoMuonCollection found in infile" << endreq;
75          return StatusCode::FAILURE;
76      }
77      const MissingET* missingET = 0;
78      if ((m_storeGate->retrieve(missingET,"MET_Final")).isFailure()
79          || (missingET==0)) {
80          m_log << MSG::WARNING
81                << "No MET_Final found in infile" << endreq;
82          return StatusCode::FAILURE;
83      }
84      //Update collection
85      m_tool->getTool()->updateCollections(
            jets,truthJets,electrons,muons,missingET);
```

```
86     //Perform this event loop
87     m_tool->getTool()->eachEvent();
88     return StatusCode::SUCCESS;
89 }
```

StatusCode DiJetAraToolAlg::finalize () `[virtual]`

Finalize.

Returns:

status code

Definition at line 91 of file DiJetAraToolAlg.cxx.

```
91                                  {
92     m_tool->getTool()->finalize();
93     m_tool.release();
94     return StatusCode::SUCCESS;
95 }
```

Member Data Documentation

ToolHandle< DiJetAraToolWrapper > DiJetAraToolAlg::m_tool `[protected]`

Tool.

Definition at line 44 of file DiJetAraToolAlg.h.

MsgStream DiJetAraToolAlg::m_log `[protected]`

Message stream.

Definition at line 47 of file DiJetAraToolAlg.h.

StoreGateSvc* DiJetAraToolAlg::m_storeGate `[protected]`

A handle on the Store Gate service for access to the Event Store.

Definition at line 50 of file DiJetAraToolAlg.h.

The documentation for this class was generated from the following files:

- /afs/cern.ch/user/t/tburgess/scratch0/testarea_14.5.2/DiJet/DiJet/**DiJetAraToolAlg.h**
- /afs/cern.ch/user/t/tburgess/scratch0/testarea_14.5.2/DiJet/src/**DiJetAraToolAlg.cxx**

8.4 DiJet File Documentation

8.4.1 /afs/cern.ch/user/t/tburgess/scratch0/testarea_-14.5.2/DiJet/DiJet/DiJetAraTool.h File Reference

Definition of DiJet Athena Root Access Tool.

```
#include "AraTool/AraToolBase.h"
#include <iostream>
#include "TTree.h"
#include "TBranch.h"
#include "TH1.h"
#include "TH2.h"
#include "TProfile.h"
#include "TObjString.h"
#include "JetEvent/Jet.h"
#include "JetEvent/JetCollection.h"
#include "egammaEvent/egammaPIDdefs.h"
#include "egammaEvent/ElectronContainer.h"
#include "egammaEvent/Electron.h"
#include "muonEvent/Muon.h"
#include "muonEvent/MuonContainer.h"
#include "MissingETEvent/MissingET.h"
```

Classes

- class **DiJetAraTool**

  DiJet Athena Root Access tool.

Detailed Description

Definition of DiJet Athena Root Access Tool.

Author:

Kent Skjei <kent.skjei-at-gmail.com>
Thomas Burgess <tburgess-at-cern.ch>

Created 2009-02-16.

Id

Definition in file **DiJetAraTool.h**.

8.4.2 /afs/cern.ch/user/t/tburgess/scratch0/testarea_-14.5.2/DiJet/DiJet/DiJetAraToolAlg.h File Reference

Implementation of ATHENA Algorithm for DiJet Athena Root Access Tool.

```
#include "StoreGate/StoreGateSvc.h"
#include "GaudiKernel/Algorithm.h"
#include "GaudiKernel/ToolHandle.h"
#include "DiJet/DiJetAraToolWrapper.h"
#include <string>
```

Classes

- class **DiJetAraToolAlg**
  ATHENA Algorithm for DiJet Athena Root Access Tool.

Detailed Description

Implementation of ATHENA Algorithm for DiJet Athena Root Access Tool.

Author:

Thomas Burgess <tburgess-at-cern.ch>

Created 2009-02-16.

Id

Definition in file **DiJetAraToolAlg.h**.

8.4.3 /afs/cern.ch/user/t/tburgess/scratch0/testarea_-14.5.2/DiJet/DiJet/DiJetAraToolWrapper.h File Reference

Definition of wrapper for **DiJetAraTool**(p. 80).

```
#include "DiJet/DiJetAraTool.h"
#include "AraTool/AraAlgToolWrapper.h"
```

Typedefs

- typedef AraAlgToolWrapper< **DiJetAraTool** > **DiJetAraToolWrapper**

    Wrapper for **DiJetAraTool**(p. 80).

Detailed Description

Definition of wrapper for **DiJetAraTool**(p. 80).

Author:
Thomas Burgess <tburgess-at-cern.ch>

Created 2009-02-16.

Id

Definition in file **DiJetAraToolWrapper.h**.

Typedef Documentation

typedef AraAlgToolWrapper< DiJetAraTool > DiJetAraToolWrapper

Wrapper for **DiJetAraTool**(p. 80).
Definition at line 18 of file DiJetAraToolWrapper.h.

8.4.4 /afs/cern.ch/user/t/tburgess/scratch0/testarea_-14.5.2/DiJet/DiJet/DiJetDict.h File Reference

DiJet packade dictionary file.

```
#include "DiJet/DiJetAraTool.h"
```

Detailed Description

DiJet packade dictionary file.

Author:
Thomas Burgess <tburgess-at-cern.ch>

Created 2009-02-16.
Definition in file **DiJetDict.h**.

8.4.5 MainPage.h File Reference

8.4.6 /afs/cern.ch/user/t/tburgess/scratch0/testarea_-14.5.2/DiJet/share/DiJet_topOptions.py File Reference

8.4.7 /afs/cern.ch/user/t/tburgess/scratch0/testarea_-14.5.2/DiJet/share/init_root.py File Reference

8.4.8 /afs/cern.ch/user/t/tburgess/scratch0/testarea_-14.5.2/DiJet/src/components/DiJet_entries.cxx File Reference

DiJet packade entries declaration.

```
#include "DiJet/DiJetAraTool.h"
#include "DiJet/DiJetAraToolAlg.h"
#include "DiJet/DiJetAraToolWrapper.h"
#include "GaudiKernel/DeclareFactoryEntries.h"
```

Functions

- **DECLARE_FACTORY_ENTRIES** (DiJet)

Detailed Description

DiJet packade entries declaration.

Author:

Thomas Burgess <tburgess-at-cern.ch>

Created 2009-02-16.

Id

Definition in file **DiJet_entries.cxx**.

Function Documentation

DECLARE_FACTORY_ENTRIES (DiJet)

Definition at line 18 of file DiJet_entries.cxx.

```
18                                {
19     DECLARE_ALGTOOL( DiJetAraToolAlg )
20     DECLARE_ALGTOOL( DiJetAraToolWrapper )
21 }
```

8.4.9 /afs/cern.ch/user/t/tburgess/scratch0/testarea_-14.5.2/DiJet/src/components/DiJet_load.cxx File Reference

DiJet packade load file.

```
#include "GaudiKernel/LoadFactoryEntries.h"
```

Detailed Description

DiJet packade load file.

Author:

Thomas Burgess <tburgess-at-cern.ch>

Created 2009-02-16.

Id

Definition in file **DiJet_load.cxx**.

8.4.10 /afs/cern.ch/user/t/tburgess/scratch0/testarea_-14.5.2/DiJet/src/DiJetAraTool.cxx File Reference

Implementation of DiJet Athena Root Access Tool.

```
#include "GaudiKernel/StatusCode.h"
#include "DiJet/DiJetAraTool.h"
#include "GaudiKernel/SystemOfUnits.h"
#include "TMath.h"
#include "TH1.h"
#include "TH2.h"
#include "TFile.h"
#include "TTree.h"
#include "TBranch.h"
#include "TProfile.h"
#include "JetEvent/Jet.h"
#include "JetEvent/JetCollection.h"
#include "egammaEvent/egammaPIDdefs.h"
#include "egammaEvent/ElectronContainer.h"
#include "egammaEvent/Electron.h"
#include "muonEvent/Muon.h"
#include "muonEvent/MuonContainer.h"
#include "MissingETEvent/MissingET.h"
#include <sstream>
```

Detailed Description

Implementation of DiJet Athena Root Access Tool.

Author:

Kent Skjei <kent.skjei-at-gmail.com>
Thomas Burgess <tburgess-at-cern.ch>

Created 2009-02-16.

Id

Definition in file **DiJetAraTool.cxx**.

8.4.11 /afs/cern.ch/user/t/tburgess/scratch0/testarea_-14.5.2/DiJet/src/DiJetAraToolAlg.cxx File Reference

Definition of ATHENA Algorithm for DiJet Athena Root Access Tool.

```
#include "StoreGate/StoreGateSvc.h"
#include "DiJet/DiJetAraToolAlg.h"
#include "GaudiKernel/MsgStream.h"
#include "GaudiKernel/Message.h"
#include "GaudiKernel/GaudiException.h"
#include <string>
```

Detailed Description

Definition of ATHENA Algorithm for DiJet Athena Root Access Tool.

Author:

Thomas Burgess <tburgess-at-cern.ch>

Created 2009-02-16.

Id

Definition in file **DiJetAraToolAlg.cxx**.

8.4.12 /afs/cern.ch/user/t/tburgess/scratch0/testarea_-14.5.2/DiJet/src/DiJetAraToolWrapper.cxx File Reference

Implementation of wrapper for **DiJetAraTool**(p. 80).

```
#include "DiJet/DiJetAraToolWrapper.h"
```

Detailed Description

Implementation of wrapper for **DiJetAraTool**(p. 80).

Author:

Thomas Burgess <tburgess-at-cern.ch>

Created 2009-02-16.

Id

Definition in file **DiJetAraToolWrapper.cxx**.

Bibliography

[1] Maximilien Brice. $[x]/2005/0511013/0511013\_01/0511013\_01 - A4 - at - 144 - dpi.jpg$, 2005-11-04 [Downloaded 2009-07-20]. $x = http://mediaarchive.cern.ch/MediaArchive/Photo/Public$.

[2] The ATLAS Collaboration. *The ATLAS Experiment at the CERN Large Hadron Collider*, chapter Chapter 1 Overview of the ATLAS detector - 1.1 Physics requirements and detector overview, pages 1–5. JINST,3,S08003, 2008.

[3] The ATLAS Collaboration. *The ATLAS Experiment at the CERN Large Hadron Collider*, chapter Chapter 1 Overview of the ATLAS detector - 1.2 Tracking, pages 5–7. JINST,3,S08003, 2008.

[4] The ATLAS Collaboration. *The ATLAS Experiment at the CERN Large Hadron Collider*, chapter Chapter 1 Overview of the ATLAS detector - 1.3 Calorimetry, pages 7–11. JINST,3,S08003, 2008.

[5] The ATLAS Collaboration. *The ATLAS Experiment at the CERN Large Hadron Collider*, chapter Chapter 1 Overview of the ATLAS detector - 1.4 Muon system, pages 11–13. JINST,3,S08003, 2008.

[6] The ATLAS Collaboration. *The ATLAS Experiment at the CERN Large Hadron Collider*, chapter Chapter 1 Overview of the ATLAS detector - 1.5 Forward detectors, page 14. JINST,3,S08003, 2008.

[7] The ATLAS Collaboration. *The ATLAS Experiment at the CERN Large Hadron Collider*, chapter Chapter 2 Magnet system and magnetic field, pages 19–37. JINST,3,S08003, 2008.

[8] The ATLAS Collaboration. *The ATLAS Experiment at the CERN Large Hadron Collider*, chapter Trigger, data acquisition, and controls - 8.1 Introduction to event selection and data acquisition, page 218. JINST,3,S08003, 2008.

[9] The ATLAS Collaboration. *Expected Performance of the ATLAS Experiment - Detector, Trigger and Physics*, chapter B Physics - Heavy Quarkonium Physics with Early Data, page 1083. arXiv:0901.0512, 2008.

[10] The ATLAS Collaboration. *Expected Performance of the ATLAS Experiment - Detector, Trigger and Physics*, chapter B Physics - Production Cross-Section Measurments and the Study of the Properties of the Exclusive $B^+ \rightarrow J/\psi K^+$ Channel - Introduction, page 1083. arXiv:0901.0512, 2008.

[11] The ATLAS Collaboration. *Expected Performance of the ATLAS Experiment - Detector, Trigger and Physics*, chapter B Physics - Physics and Detector Performance Measurements with the Decays $B_s^0 \rightarrow J/\psi\phi$ and $B_d^0 \rightarrow J/\psi K^{0*}$ with Early Data - Introduction, page 1121. arXiv:0901.0512, 2008.

[12] The ATLAS Collaboration. *Expected Performance of the ATLAS Experiment - Detector, Trigger and Physics*, chapter B Physics - Study of the Rare Decay $B_0^s \rightarrow \mu^+\mu^-$ - 1 Introduction, page 1154. arXiv:0901.0512, 2008.

[13] The ATLAS Collaboration. *Expected Performance of the ATLAS Experiment - Detector, Trigger and Physics*, chapter B Physics - Trigger and Analysis Strategies for B_s^0 Oscillation Measurments in Hadronic Decay Channels - 1 Introduction, pages 1167–1168. arXiv:0901.0512, 2008.

[14] The ATLAS Collaboration. *Expected Performance of the ATLAS Experiment - Detector, Trigger and Physics*, chapter Higgs Boson - Introduction on Higgs Boson Searches, pages 1198–1212. arXiv:0901.0512, 2008.

[15] The ATLAS Collaboration. *Expected Performance of the ATLAS Experiment - Detector, Trigger and Physics*, chapter Supersymmetry - Supersymmetry Searches - 3 Object Identification for SUSY Analysis, page 1518. arXiv:0901.0512, 2008.

[16] The ATLAS Collaboration. *Expected Performance of the ATLAS Experiment - Detector, Trigger and Physics*, chapter Supersymmetry - Supersymmetry Searches - 2 Signal and Background generation - 2.2 Backgrounds, pages 1517–1518. arXiv:0901.0512, 2008.

[17] The ATLAS Collaboration. *Expected Performance of the ATLAS Experiment - Detector, Trigger and Physics*, chapter Exotic Processes, pages 1695–1828. arXiv:0901.0512, 2008.

[18] The ATLAS Collaboration. *Expected Performance of the ATLAS Experiment - Detector, Trigger and Physics*, chapter Trigger - The Trigger for Early Running, pages 550–553. arXiv:0901.0512, 2008.

[19] The ATLAS Collaboration. *Expected Performance of the ATLAS Experiment - Detector, Trigger and Physics*, chapter PERFORMANCE - Tracking, pages 15–43. arXiv:0901.0512, 2008.

[20] The ATLAS Collaboration. *Expected Performance of the ATLAS Experiment - Detector, Trigger and Physics*, chapter PERFORMANCE - Electrons and Photons - Calibration and Performance of the Electromagnetic Calorimeter, pages 44–72. arXiv:0901.0512, 2008.

[21] The ATLAS Collaboration. *Expected Performance of the ATLAS Experiment - Detector, Trigger and Physics*, chapter PERFORMANCE - Jets and Missing Transverse Energy, pages 261–397. arXiv:0901.0512, 2008.

[22] The ATLAS Collaboration. *Expected Performance of the ATLAS Experiment - Detector, Trigger and Physics*, chapter PERFORMANCE - Muons - Muon Reconstruction and Identification: Studies with Simulated Monte Carlo Samples, pages 162–185. arXiv:0901.0512, 2008.

[23] The ATLAS Collaboration. *Expected Performance of the ATLAS Experiment - Detector, Trigger and Physics*, chapter Cross-Sections, Monte Carlo Simulations and Systematic Uncertainties, pages 3–15. arXiv:0901.0512, 2008.

[24] The ATLAS Collaboration. *Expected Performance of the ATLAS Experiment - Detector, Trigger and Physics*, chapter PERFORMANCE - Jets and Missing Transverse Energy - Jet Reconstruction Performance, pages 262–298. arXiv:0901.0512, 2008.

[25] The ATLAS Collaboration. *Expected Performance of the ATLAS Experiment - Detector, Trigger and Physics*, chapter Trigger - Overview and Performance Studies of Jet Identification in the Trigger System, page 698. arXiv:0901.0512, 2008.

[26] R. K. Ellis, W. J. Stirling, and B. R. Webber. *QCD and Collider Physics*, chapter 7 Hadroproduction of Jets and Photons, page 246. Cambridge University Press, 2003.

[27] C. Amsler et al. (Particle Data Group). $http$: $//pdglive.lbl.gov/Rsummary.brl?nodein = S043$, (2008) and 2009 partial update for the 2010 edition. [Read 2009-07-15].

[28] Konrad Kleinknecht. *Detectors for Particle Radiation - second edition*, chapter 1 Physics foundations, pages 9–39. Cambridge University Press, 1998.

[29] A. Lipniacka. private communication (serious) (accepted).

[30] Claudia Marcelloni and Maximilien Brice. $[x]/2007/0706013/0706013\_03/0706013\_03 - A4 - at - 144 - dpi.jpg$, 2007-07-13 [Downloaded 2009-07-20]. $x = http : //mediaarchive.cern.ch/MediaArchive/Photo/Public.$

[31] Joao Pequenao. $[x]/2008/0803012/0803012\_01/0803012\_01 - A4 - at - 144 - dpi.jpg$, 2008-03-27 [Downloaded 2009-07-09]. $x = http : //mediaarchive.cern.ch/MediaArchive/Photo/Public.$

[32] Joao Pequenao. $[x]/2008/0803022/0803022\_01/0803022\_01 - A4 - at - 144 - dpi.jpg$, 2008-03-27 [Downloaded 2009-07-09]. $x = http : //mediaarchive.cern.ch/MediaArchive/Photo/Public.$

[33] Joao Pequenao. $[x]/2008/0803014/0803014\_01/0803014\_01 - A4 - at - 144 - dpi.jpg$, 2008-03-27 [Downloaded 2009-07-09]. $x = http : //mediaarchive.cern.ch/MediaArchive/Photo/Public.$

[34] Joao Pequenao. $[x]/2008/0803013/0803013\_02/0803013\_02 - A4 - at - 144 - dpi.jpg$, 2008-03-27 [Downloaded 2009-07-09]. $x = http : //mediaarchive.cern.ch/MediaArchive/Photo/Public.$

[35] Joao Pequenao. $[x]/2008/0803014/0803014\_03/0803014\_03 - A4 - at - 144 - dpi.jpg$, 2008-03-27 [Downloaded 2009-07-09]. $x = http : //mediaarchive.cern.ch/MediaArchive/Photo/Public.$

[36] Joao Pequenao. $[x]/2008/0803015/0803015\_01/0803015\_01 - A4 - at - 144 - dpi.jpg$, 2008-03-27 [Downloaded 2009-07-09]. $x = http : //mediaarchive.cern.ch/MediaArchive/Photo/Public.$

[37] Joao Pequenao. $[x]/2008/0803017/0803017\_01/0803017\_01 - A4 - at - 144 - dpi.jpg$, 2008-03-27 [Downloaded 2009-07-09]. $x = http : //mediaarchive.cern.ch/MediaArchive/Photo/Public.$

[38] Philipp Schieferdecker. $[t]/twiki/bin/view/CMS/JetResolution$, 2009-07-02 [Read 2009-09-30]. $t = https : //twiki.cern.ch.$

[39] Torbjörn Sjöstrand. *http : //home.thep.lu.se/ torbjorn/Pythia.html*, [Read 2009-09-29].

[40] AC Team. $[x]/1999/9906026/9906026/9906026 - A4 - at - 144 - dpi.jpg$, 1999-06 [Downloaded 2009-07-09]. $x = http : //mediaarchive.cern.ch/MediaArchive/Photo/Public.$

[41] The ROOT team. The root users guide, v.5.24. *http : //root.cern.ch/drupal/content/users − guide*, 2009-09-07[Read 2009-09-29].

[42] Richard Teuscher. $[s]5\&resId = 0\&materialId = slides\&confId = 48779$, 2009-09-01 [Read 2009-09-30]. $s = http : //indico.cern.ch/getFile.py/access?contribId =.$

[43] Uknown. *http : //atlas − proj − computing − tdr.web.cern.ch/atlas − proj − computing − tdr/Html/Computing − TDR − 21.htm#pgfId − 1019542*, 2005-06-04 [Read 2009-07-20].

[44] Uknown. $[y]/Atlas/WorkBookAthenaFramework$, 2006-01-23 [Read 2009-07-20]. $y = https : //twiki.cern.ch/twiki/bin/view.$

[45] Uknown. $[y]/Atlas/AthenaFramework$, 2007-09-14 [Read 2009-07-20]. $y = https : //twiki.cern.ch/twiki/bin/view.$

[46] Uknown. $[y]/Atlas/FullDressRehearsal$, 2007-11-21 [Read 2009-07-20]. $y = https : //twiki.cern.ch/twiki/bin/view.$

[47] Uknown. $[y]/Atlas/WorkBookComputingModel$, 2008-01-16 [Read 2009-07-20]. $y = https : //twiki.cern.ch/twiki/bin/view.$

[48] Uknown. *http : //projects.hepforge.org/jimmy/*, 2009-08-18 [Read 2009-09-29].

[49] Uknown. *http : //ami.in2p3.fr/opencms/opencms/AMI/www/*, 2009-09-14 [Read 2009-09-28].

[50] Uknown. *http : //geant4.web.cern.ch/geant4/*, 2009-09-18[Read 2009-09-29].

[51] Uknown. $[y]/AtlasProtected/TopReferences10TeV$, 2009-09-24[Read 2009-09-28]. $y = https : //twiki.cern.ch/twiki/bin/view.$

[52] Uknown. *http : //jarguin.home.cern.ch/jarguin/dc3requests_sm.html*, [Read 2009-09-28].

[53] Unknown. $[z]/2007/07/standard\_model\_316.gif$, 2007 [Downloaded 2009-07-09]. $z = http://gabrielbcn.files.wordpress.com.$

[54] Unknown. $[v]/ brock/feynman/vtp_w s0506/chapter09/higgs_f eyn_p p.jpg$, [Downloaded 2009-09-30]. $v = http : //www - zeus.physik.uni - bonn.de.$

[55] Unknown. $[v]/AdventsKalenderPlots/QCDRunningAlpha/qq_g lue_s.jpg$, [Downloaded 2009-09-30]. $v = http : //user.uni - frankfurt.de/ scherers/blogging.$

[56] Unknown. $http : //farm4.static.flickr.com/3075/3160860000_2 b147b89d8.jpg$, [Downloaded 2009-09-30].

[57] Unknown. $[k]/selected - photos/solenoid/solenoid\_magnet\_2.jpg$, Unknown [Downloaded 2009-07-20]. $k = http : //atlas.ch/photos/atlas\_photos.$