Jets in Switzerland

By

Khalid Omer Elamin Hassan (khalid.hassan@aims.ac.rw)

June 2017

AN ESSAY PRESENTED TO AIMS RWANDA IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF MASTER OF SCIENCE IN MATHEMATICAL SCIENCES

DECLARATION

This work was carried out at AIMS Rwanda in partial fulfilment of the requirements for a Master of Science Degree.

I hereby declare that except where due acknowledgement is made, this work has never been presented wholly or in part for the award of a degree at AIMS Rwanda or any other University.

Scan your signature

Student: Khalid Omer Elamin Hassan

Scan your signature

Supervisor: Firstname Middlename Surname

Abstract

At the Large Hadron Collider highly energetic protons are being collided in order to study the subatomic physics. As a result of this collision the proton constituents (quarks and gluons) are being produced. However, in practical we can not detect them. The quarks and gluons undergo a showering process and then hadronize. In order to understand them a Monte Carlo simulations were were used, including a simple toy simulation and Pythia simulation. In both cases we studied these simulations using jet algorithms.

Contents

1. General Introduction

High energy particle physics is the field of physics that studies the fundamental structure of matter. This can be done in two ways, one is to look for elementary particles, the building blocks of matter at their smallest scale. The other is to define what interactions are acting among them *(forces)* to build the matter as we see it. In order to study the fundamental structure of the matter, we require tools. For this purpose scientists have built the particle accelerators such as the Large Hadron Collider.

1.1 Large Hadron Collider

Hadron colliders are devices made to explore the world of particle physics, they work as theories testers and also as discovery machines, an example of these hadron colliders is the Large Hadron Collider (LHC) at CERN (Evans and Bryant, 2008).

Among the main missions of the LHC is the investigation of the properties of the Higgs boson. The importance of the Higgs boson stems from investigating the validity of the standard model suggests that the masses of the elementary particles are generated through the Higgs mechanism (Nagashima and Nambu, 2010). In July 2012, the ATLAS and CMS experiments at CERN's Large Hadron Collider announced that they have had each observed new particle in the mass region around 126 GeV (Aad et al., 2012). This particle is consistent with the Higgs boson predicted by the standard model.

Inside the Large Hadron Collider, 10^{11} protons are being accelerated to a high kinetic energy and then collided up 40 million times per second to provide 14 TeV proton-proton collision. The LHC also collides heavy ions. Two general purpose detectors, ATLAS (A Toroidal LHC ApparatuS) Aad et al. (2008) and CMS (Compact Muon Solenoid) Collaboration (2008) have been built for probing proton-proton and ion-ion collisions.

Most of the interesting physics at LHC involves studying the results of these interactions (collisions). It has been observed that as a result of these collisions, stable partons are formed, partons consists of quarks and gluons. From experimental point of view, these collisions lead to the production of these partons alongside the production of another particles like the *(Higgs)* boson. However, new particles are not detected. The study of quarks and gluons in LHC is challenging because of the production of a single quark or a gluon will actually appear in the detectors as many particles collimated in the same general direction, all arriving at once. The detection of a collimated flow of particles of this nature is called a **jet** clustering (Ellis et al., 2008).

1.2 Quarks and Gluons

In our current view, there are three types of elementary particles, *leptons*, *quarks* and *mediators*. The first two are matter particles and the third allow the interaction between the first two

(Griffiths, 2008).

There are six "flavours" of quarks, these are up, down, charm, strange, top and bottom. Quarks can successfully account for all known mesons and baryons which are known as hadrons, hadrons are colour charge neutral particles. Quarks carry colour charge: red, green or blue, the colour name here is an analogy that is famously used among physicists to describe a three kind of generating force. The colour charges are called by their number or any other index. We can also we think of it as a three primary additive colours.

The other partons (gluons) mediate the interaction between quarks.

Gluon is a massless spin 1 particle, carrying charge called colour charges similar to the quarks, because of the colour charge gluons can interact among themselves.

The theory that describes the colour charge of the quarks and gluons is called *(Quantum Chromodynamics)*. The colour charge has strange property that it exerts a constant force that binds colour carrying particles together, this can be visualized using the analogy of a rubber band, the stronger you pull on the rubber band the tighter it feels. If we do not pull on it at all, it hangs loose. The same thing happens for the particles, that means at a very short distance, the force is relaxed and the particles behave as free particles. As the distance between them increases, the force acts like a rubber band and pulls them back stronger. When the rubber band is stretched beyond its limits, it cuts into many pieces producing more particles. This phenomenon is known as the colour confinement. In other words, these particles tend not to be separated by a macroscopic distance (Nagashima and Nambu, 2010).

Due to the complex nature of the event, event in this context refers to the result of the particle interaction (collision), at the Large Hadrons Collider, the description of the final state involves a multi-particle calculations. Monte Carlo event generators provide a good approximation of collisions.

2. Monte Carlo Computational Techniques

The name Monte Carlo method is a set of mathematical tools that was first used by scientists working on nuclear projects in Los Alamos. The essence of this is to generate numbers with probability that can be used to study physical phenomena. In our context, the definition of Monte Carlo method is the use of randomly generated numbers to imitate a physical behaviour that is considered to be random (quantum mechanical) (Kalos and Whitlock, 1986).

One of the most important components of the Monte Carlo method is generating samples of different probability distribution functions (pdf). This is essential since we are simulating various variables that have different numerical behaviour. For example if our pseudo-random numbers are uniformly[1] distributed in the interval $[0, 1]$ and instead we need numbers that have normal distribution restricted to the same interval. This part of the essay discusses two methods which are widely used for that purpose.

2.1 The Inverse Transform Method

The inverse transform method is used for generating random numbers that are distributed according to a specific distribution. Common case is that we assume that we are able to generate uniformly distributed numbers and we want to sample them according a specific distribution $p(x)$.

Let x be a random variable distributed with probability density function $p(x)$. Let u be a random variable that is uniformly distributed in $[0, 1]$ and $P(x)$ is the cumulative density function (CDF), then we set

$$x = P^{-1}(u) \tag{2.1.1}$$

(Weinzierl, 2000).

The following pseudo-code shows the inverse-transform method algorithm

Generate a uniform random u in $[0, 1]$
return $x = P^{-1}(u)$

The inversion method is exact when an explicit form of the CDF is known. The CDF is a continuous and a strictly increasing function. It can be obtained either analytically through the integration of the PDF or numerically. Sometimes finding the CDF if computationally difficult, this causes the inverse transform method to become less efficient (Devroye, 1986).

[1] Uniform means that every value in the range of the distribution is equally likely to occur. This distribution is widely used for generating random numbers for other distributions, it is denoted by $U(0, 1)$.

2.2 The Accept Reject Method

As the inverse transform method, the accept reject method also is used for generating numbers according to a specific distribution.

Let x be a uniformly distributed variable in the interval $[0, 1]$, which is the variable of interest. A different distribution of x is required, let $p(x)$ denotes the pdf of the required distribution. Another random uniformly distributed variable y is generated, this will serve as the accept reject tool. First we calculate the maximum of $p(x)$, hence the y is generated in the interval $[0, p(x)_{max}]$. Now we check if $y \leq p(x)$. If this is the case, accept x, otherwise reject and start again.

$x \leftarrow$ uniform in $[0, 1]$
$y \leftarrow$ uniform in $[0, p_{max}(x)]$
if $y \leq p(x)$ **then**
    accept x
else
    Reject
end if
Begin again

For example, if we have a number x that falls under uniform distribution and we want to reshape this so that we get a number that has the distribution $\frac{1}{x}$, then we find the maximum value of probability density function $(p(x))$ for x. Here, we add small number to x so that we avoid the singular point when $x = 0$. After that we generate another sample y that is also uniformly distributed in the interval $[0, p_{max}]$. Now we check if y is less than $p(x)$. If so, we add x to our distribution, if not we start again (Weinzierl, 2000).

The histogram in figure 2.1 demonstrate the example above.

Beside the fact that the numbers which are generated from Monte Carlo method are not truly random, whereby the numbers are generated through a deterministic algorithms (Kalos and Whitlock, 1986). Monte Carlo method also faces another challenge which it requires generating large samples to give sensible results. The histograms in figure 2.2 demonstrate that, in which pi is calculated using numbers generated using Monte Carlo method. One sees that the error in the calculation when is decreasing as the number of the samples is increasing .

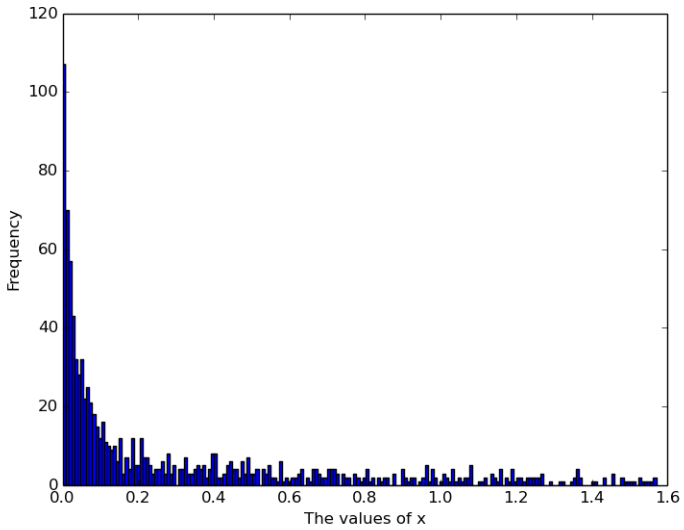
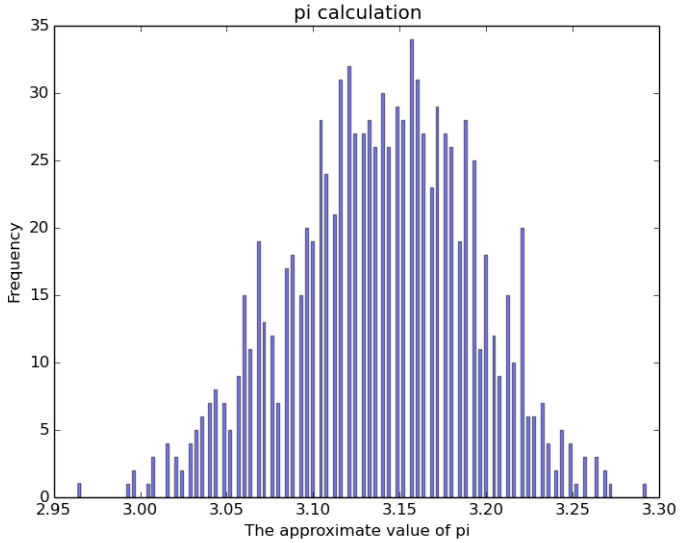
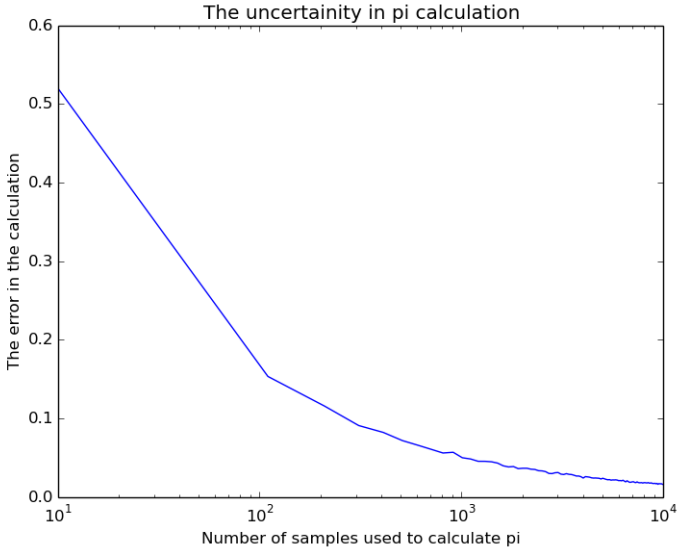


Figure 2.1: Sampling 1000 numbers with distribution $\frac{1}{x+\epsilon}$ using accept-reject method, where ϵ is a small number added to avoid the singularity, here it is taken to be 0.01.



(a) pi value



(b) The uncertainty

Figure 2.2: These histograms show the values of π (a) and the error that are generated from sampling 1000 numbers (b).

3. The Parton Shower

Parton showers are approximations of what is seen in the detectors (stable particles) to the hard scattering. The word "hard", means the process involves a transfer of large momentum, in which the parons are splitting and branching. They locally conserve flavour and four momentum, and also conserve the probability, in a sense that, the particle either splits into two or not.

Since the parton showers are simulations of the branching and splitting processes, the quality of their predictions depend on how precise is the implementation For example, the angular ordering, which accounts for knowing the prehistory of the splitting particles, i.e., knowing exactly from which partons are they coming from. (Höche, 2014).

3.1 The Parton Shower Model

The parton shower model is built around the idea of successive one-to-two splittings, which are combined together forming a tree-like sequence.

The Monte Carlo model of the parton shower can be described as a sequence of stochastic and deterministic processes.

The deterministic stage is where the particle is produced with some four momentum and travels for a finite time and distance. After the time distance travel, the particle will spilt into two, with one the particles which we call soft (radiated) having angle θ with respect to the original particle and taking Z fraction of the initial particle energy, θ and Z represents the stochastic part of the model.

There is one QCD approximation that will be useful regarding θ and Z distributions. This is the *soft* and *collinear* approximation. *Soft* implies that an emitted particle has a very little energy compare to the particle that emitted it. *Collinear* means that it is emitted with angle very small relative to another particle in the event (Salam, 2010b). With each splitting the particle loses energy, at some point the particles will have low energy and then begin to stabilize (hadronize) leading to a colour neutral particles (hadrons).

To reflect the colour neutrality of the particles in our model the partons will be transformed into a stable hadrons which are colour neutral. This process is called hadronization. The process in which this happens is not quantitatively well understood. For this model, a very simple approach is taken, where a direct translation between each parton and hadron is made. To achieve this, a threshold energy is defined. Below it, the partons will begin to hadronize or stop splitting (Salam, 2010b).

3.2 Parton Shower and Hadronization Simulation

The following is a simple simulation for a splitting of a single particle into two. The parton shower simulation describes the parton shower and hadronization. The general idea of the two processes, parton shower and the hadonization can be simplified by Algorithm 1.

Algorithm 1 Parton shower

Make list of particles L
Add initial particle P_i to L
Define stability limit (hadronization limit) to be S
While L is not empty
if energy of $P_i \geq S$ **then**
    Split into two and add to list and check again
else
    Begin hadronization
end if

This process can be simulated in two or three space dimensions, for the case of three dimensions, another angle phi is needed to account for the third dimension.

3.3 Two Dimensional Parton Shower

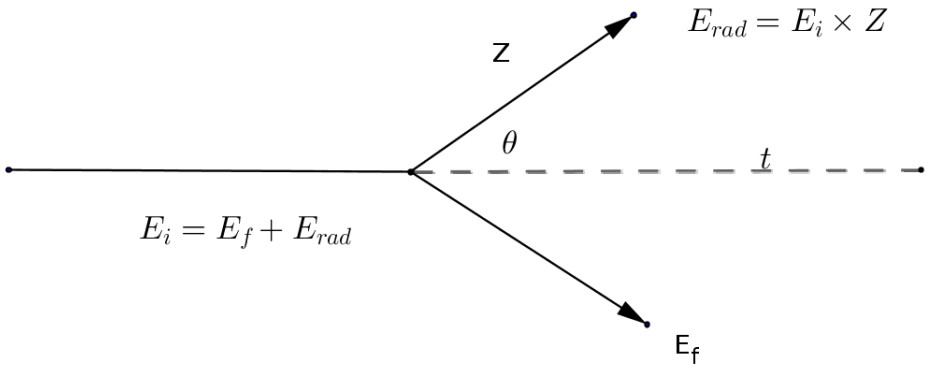


Figure 3.1: Illusration describing the splitting of a prticle into in two dimensions.

The two dimensional model accounts for one rotation with angle θ (see figure 3.1). It involves generation of two random numbers, one of them represent the angle and the other represents the energy of the radiated particle. Here, both the energy fraction Z and the angle θ are following the distribution $1/Z$, this comes from the QCD approximation that was mentioned in 3.1. Where Z lies in the interval $[0, 1]$. To avoid the singularity at $Z = 0$, a cutoff value of $\epsilon = 0.01$ is added to the denominator by setting $p(Z) = \frac{1}{\epsilon+z}$. Considering the later in the interval [0,$\pi/2$], θ is

modified in a similar way as Z. The variables θ and z are generated by applying the accept-reject method on a set of numbers that are uniformly distributed.

As for the kinematics description, the initial particle has four momentum (E, p_x, p_y, p_z)[1], for simplicity we assume that the particle is travelling in x direction with energy E. Therefore, the four momentum which describes this particle is the following

$$P_i^\mu = \begin{pmatrix} E \\ p_x \\ 0 \\ 0 \end{pmatrix} \tag{3.3.1}$$

Where p_y and p_z are equal zero following our direction assumption. In a given splitting, with generated pair (θ, z), the kinematics of the radiated particle are determined.

To continue the showering process and to allow the model to proceed, it is necessary to determine the kinematics of the final state particle by applying the conservation of energy and momentum

$$P_i^\mu = P_f^\mu \tag{3.3.2}$$

And Since

$$P^\mu P_\mu = E^2 - (p_x^2 + p_y^2 + p_z^2) = E^2 - \|p\|^2 = m_0^2 \tag{3.3.3}$$

Which is lorentz invariant quantity, i.e., it does not depend on the frame. Here, we can make a simplifying assumption, in which we assume that the mass of the radiated particle is zero. This assumption is based on the fact that, in LHC the quark energy is $\sim$ 1MeV, while the hadron energy is $\sim$ 1 TeV. Now equation 3.3.2 will become

$$E^2 = \|p\| \tag{3.3.4}$$

(Salam, 2010b).

In describing the direction of the radiated particle after it is being produced, we use the rotation matrix in two dimensions which rotates the radiated particle with the angle θ

$$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \tag{3.3.5}$$

From this we can determine the direction of the other particle. Here we will denote its four momentum by P_{part}^μ . The final momentum P_f^μ after the splitting is

$$P_{rad}^\mu + P_{part}^\mu \tag{3.3.6}$$

and from the conservation equation 3.3.2 then

$$P_i^\mu = P_{rad}^\mu + P_{part}^\mu \tag{3.3.7}$$

[1]The four momentum vector is usually written as $(E/c, p_x, p_y, p_z)$. Henceforth, we are considering the natural units in which $c = 1$.

Therefore,

$$P^{\mu}_{part} = P^{\mu}_{i} - P^{\mu}_{rad} \tag{3.3.8}$$

As in the algorithm 1, a tunable parameter (energy threshold) is made to account for the hadronization. The figure 3.2 shows a graphical representation of the two dimensions parton shower as explained above.

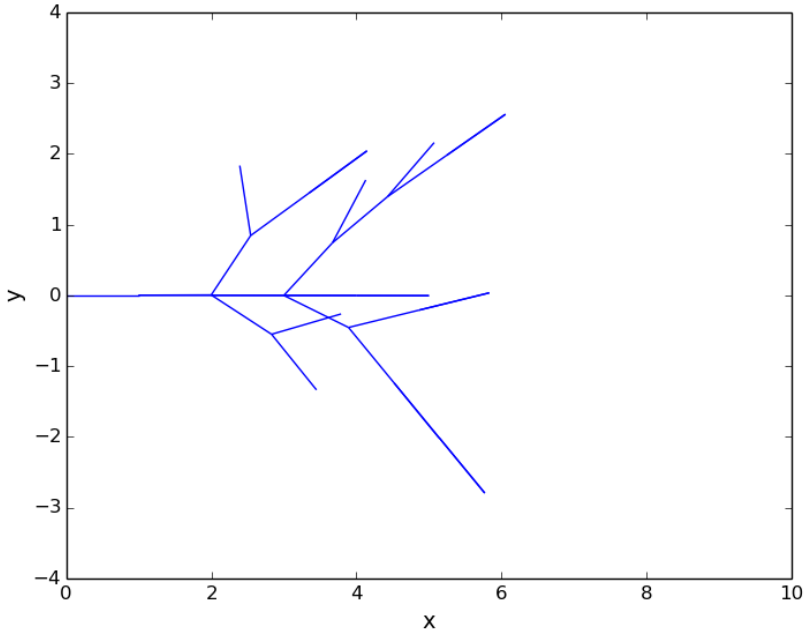


Figure 3.2: A two dimesions simulation of the spliting of a single parton. In this illusrtaion the the threshold energy (hadronizaion limit) is chosen to be 0.09

3.4 Three Dimensional Parton Shower

Since the real physics we are modelling is in three dimension, we need to modify our model to account for this. An additional splitting angle ϕ will be included, which is the azimuthal angle of the decay around the direction of travel of the initial particle, as shown in figure 3.3.

Unlike the angle θ, there is no preference for the value of this angle and it should therefore be chosen from a uniform distribution within the interval $[0, 2\pi]$ (Salam, 2010b).

There are no major differences between the three and two dimensional models, except that in the three dimensional model the particle will be rotated twice with the angles θ and ϕ. Given a unit vector **u**, where **u** is in the direction of the axis of rotation, the rotation around the axis by angle

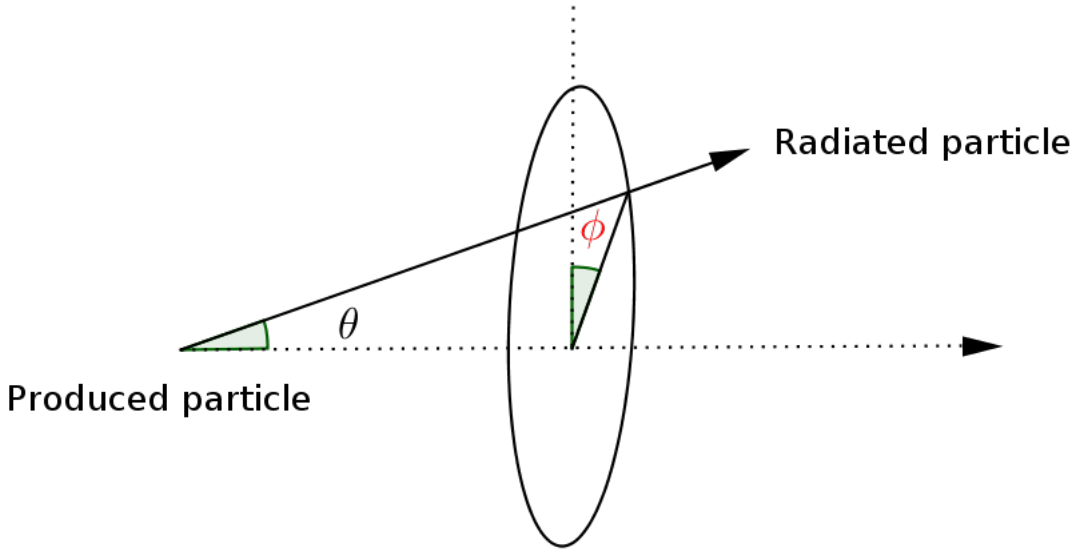


Figure 3.3: An illustration of the azimuthal angle ϕ of a one-to-two splitting occurring in three dimensions.

240 θ can by found by matrix

$$\begin{pmatrix} \cos\theta + u_x^2(1-\cos\theta) & u_x u_y(1-\cos\theta) - u_z\sin\theta & u_x u_z(1-\cos\theta) + u_y\sin\theta \\ u_y u_x(1-\cos\theta) + u_z\sin\theta & \cos\theta + u_y^2(1-\cos\theta) & u_y u_z(1-\cos\theta) - u_x\sin\theta \\ u_z u_x(1-\cos\theta) - u_y\sin\theta & u_z u_y(1-cos\theta) + u_x\sin\theta & \cos\theta + u_z^2(1-\cos\theta) \end{pmatrix} \tag{3.4.1}$$

241 In order to compute the momentum of radiated particle in three dimensions, first the momentum
242 vector is rotated angle θ around an arbitrary axis that is perpendicular to it, after that the resulting
243 vector is rotated around the original vector with the angle ϕ. The figure 3.4 shows a graphical
244 representation of the three dimensional parton shower.

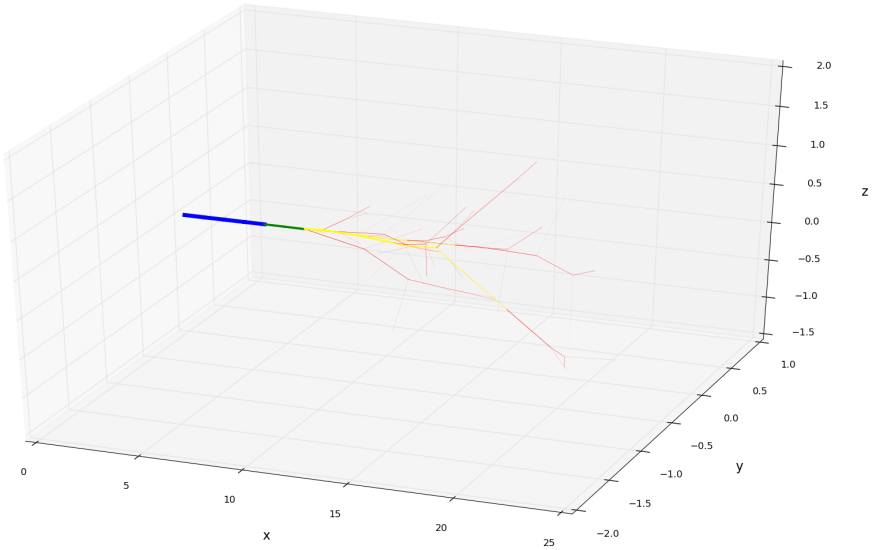


Figure 3.4: A three dimensions simulation of the spliting of a single parton. In this ilustraion the blue colour represents the paricles with energy ≥ 0.8, the green colour represnts the particles with energy ≥ 0.5, the yellow colour represents the particles with energy ≤ 0.5 and the red colour represents the particles with energy ≤ 0.1.

245 3.5 An Improved Model

246 In order to build more realistic model, it is necessary to include more details, as we did with the
247 three dimensions model. Here we start by looking at the particle in the moment of production,
248 right after the collision of the two protons. For simplicity One can assume that the collision of
249 the protons conserves the momentum. Hence the produced particle also conserve momentum.
250 Following this assumption, it can be said the produced particles come in pairs that have opposite

directions. The produced particles have energy that is randomly distributed as well as the polar angle θ and the azimuthal ϕ angle. As for the energy, it has the exponential distribution $e^{-\alpha E_{parton}}$, where α is a physically meaningful parameter that characterizes the collisions at the LHC and E_{parton} is randomly chosen energy of the parton. As for the polar and the azimuthal angles, the collision has no preferred angle in terms of the both angles θ and ϕ, hence, the former is uniformly distributed in the range $[0, \pi]$ and the later is uniformly distributed in the range $[0, 2\pi]$. (Salam, 2010b).

There are three main programs in use at present for the generation of simulated collider events. They incorporate different combinations of the approaches of the model described above. These are HERWIG, PYTHIA and SHERPA (Buckley et al., 2011).

4. Jet Reconstruction

After the process of the splitting and branching, the quarks and gluons start to hadronize leading to a collimated spray of stable colourless hadrons. Hence in practice the only outcome of this process is hadrons 1.1. In order to study the quarks and the gluons, it is plausible to think of reversing this process. This process is called jet reconstruction, and the final outcome is called jets (quarks or gluons).

The jet reconstruction is essential in understanding the link between the observed physics or the long distance physics and the underlying physics (short distance physics) in the parton level.

The definition of the jet is central in comparing the data and the theoretical predictions. The definition is provided in the form of the jet algorithm, this means the jet algorithm and its corresponding parameters and recombination scheme.

4.1 Jet Algorithms

There are two broad classes of jet algorithms, the Cone algorithms and the Sequential algorithms. Both algorithms work on defining the jets by the idea of nearness.

It is important to recognize that jet algorithms involve two distinct steps. The first step is to identify the members of the jet, i.e., the partons that make-up the final stable jets. The second step is to construct the kinematic properties that will characterize the jet (Berger et al., 2001). For merging and combining the objects, we use the 4 -vector recombination scheme, whereby to combine two particles we add their four momenta (Blazey et al., 2000).

Here, we will focus on the second class: the sequential algorithms.

4.2 Sequential Algorithms

These algorithms work by defining a distance between pairs of objects, performing a successive recombination of the pair of closest objects, and stopping when all objects are far apart.

One starts by first defining these distances, d_{ij} (4.2.1) and d_{iB} (4.2.2), where d_{ij} is the distance between objects (pseudo-jets)[1] i and j, and d_{iB} is the distance between the pseudo-jet i and the beam B. The clustering proceeds by identifying the smallest of the distances and if it is d_{ij} combine the pseudo-jets i and j. Otherwise, if it is d_{iB} calling i a jet and removing it from the list of objects. The distances are recalculated and the procedure repeated until no objects are left (Cacciari et al., 2008).

[1] Pseudo-jet since it is neither a full jet, nor yet a full particle. It can be a single particle or a composition of particles.

The difference between the sequential algorithms lies in the definition of the distances measures:

$$d_{ij} = \min(k_{ti}^{2p}, k_{tj}^{2p}) \frac{\Delta_{ij}}{R^2}, \tag{4.2.1}$$

$$d_{iB} = k_{ti}^{2p}, \tag{4.2.2}$$

Where $\Delta_{ij}^2 = (y_i - y_j)^2 + (\phi_i - \phi_j)^2$ and k_{ti}, η_i and ϕ_i are the transverse momentum, rapidity and the azimuth of the particle i (Cacciari et al., 2008). The exact formula for the transverse momentum depends on the beam axis, which is z hence, $k_t = \sqrt{p_x^2 + p_y^2}$. The rapidity describes the angle relative to the beam axis. The rapidity is defined as $\frac{1}{2} \ln \frac{E_i + p_{zi}}{E_i - p_{zi}}$, where E_i and p_{zi} are the energy and component of the momentum along the beam axis of the particle i (Cacciari et al., 2012). The parameter R describes the jet radius.

The inclusive k_t algorithm is defined with choice of $p = 1$, the case where $p = 0$ corresponds to the inclusive Cambridge/Aachen algorithm, where $p = -1$ refers to anti-k_t jet clustering algorithm (Cacciari et al., 2008).

The implementation of anti-k_t clustering algorithm can be illustrated by the following pseudocode

Calculate all the d_{ij} and d_{iB}
Find the minimum of d_{ij} and d_{iB}
if minimum distance is d_{ij} **then**
    Recombine i and j in a single object
    Go to step one
else
    the minimum is d_{iB}
    Declare i as a jet and remove from the list
    Start from the beginning
end if
Repeat until no objects left

5. Jet Observables

The next step after performing the jet clustering on the generated parton shower, is to define final state observables, that can be used to extract specific properties of the underlying physics. Here, there are two kinds of observables, event observables and jet observables.

The most important feature of the observables is that they can be defined both for the Monte Carlo simulation and the real collision data, allowing for tuning the underlying parameters of the model used to generate the parton shower.

5.1 Single Jet Observables

An example of a jet observable is the number of the constituents in the jet with highest energy. This gives an intuition about the formation of the jet, and allow as to tune the parameters of the parton shower.

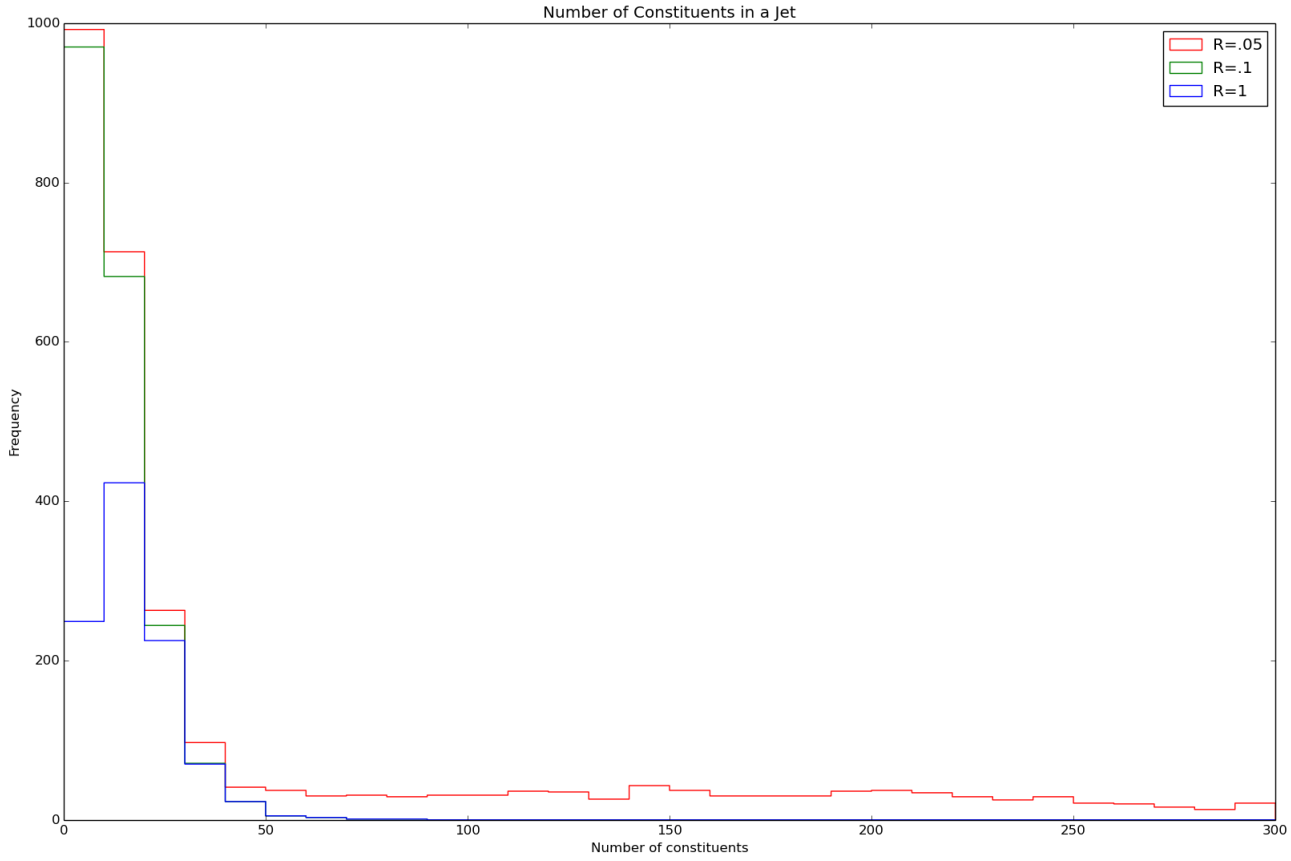


Figure 5.1: The number of constituents of a jet observable for different values of R.

Another single jet observable that we have worked on is *pseudo-mass* observable. This observable is described as follows

$$pseudo - mass(J) = E(j_1) \times E(j_2) \times \Delta_{ij}, \tag{5.1.1}$$

Where j_1 and j_2 are the two highest momenta of of the constituents of the jet J and Δ_{ij} as given above. Conditions whenever the jet with the highest energy has one constituent, this observable was not calculated.

The histograms in figures 5.1 and 5.2 illustrate the results of this calculation for 1000 events clustered with anti-k_t algorithm with different values of R. Where the first shows the number of constituents in a jet and the second shows the pseudo-mass observable.

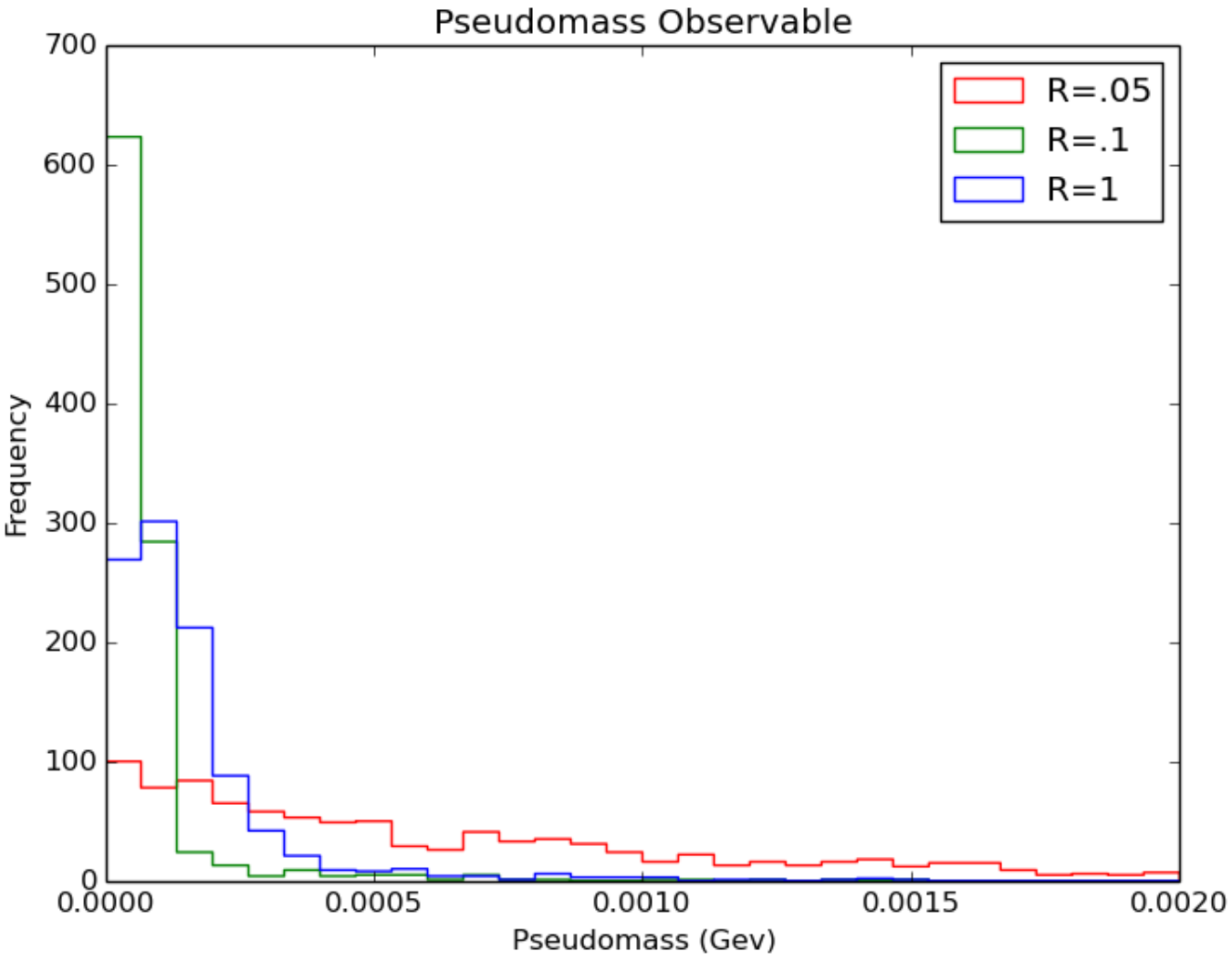


Figure 5.2: The pseudo-mass of jet observable for different values of R.

5.2 Event Observables

These observables associate the elements of initial event and final state particles (jets). An example of this is the number of jets in the event. This observable shows the number of parton that the event started with. The histogram in figure 5.3 shows the results of clustering 1000 events using anti-k_t with different values of R. One sees the number of jets if it is affected by the choice of R. It worth noting that as R gets smaller, the number of jets in the event increases.

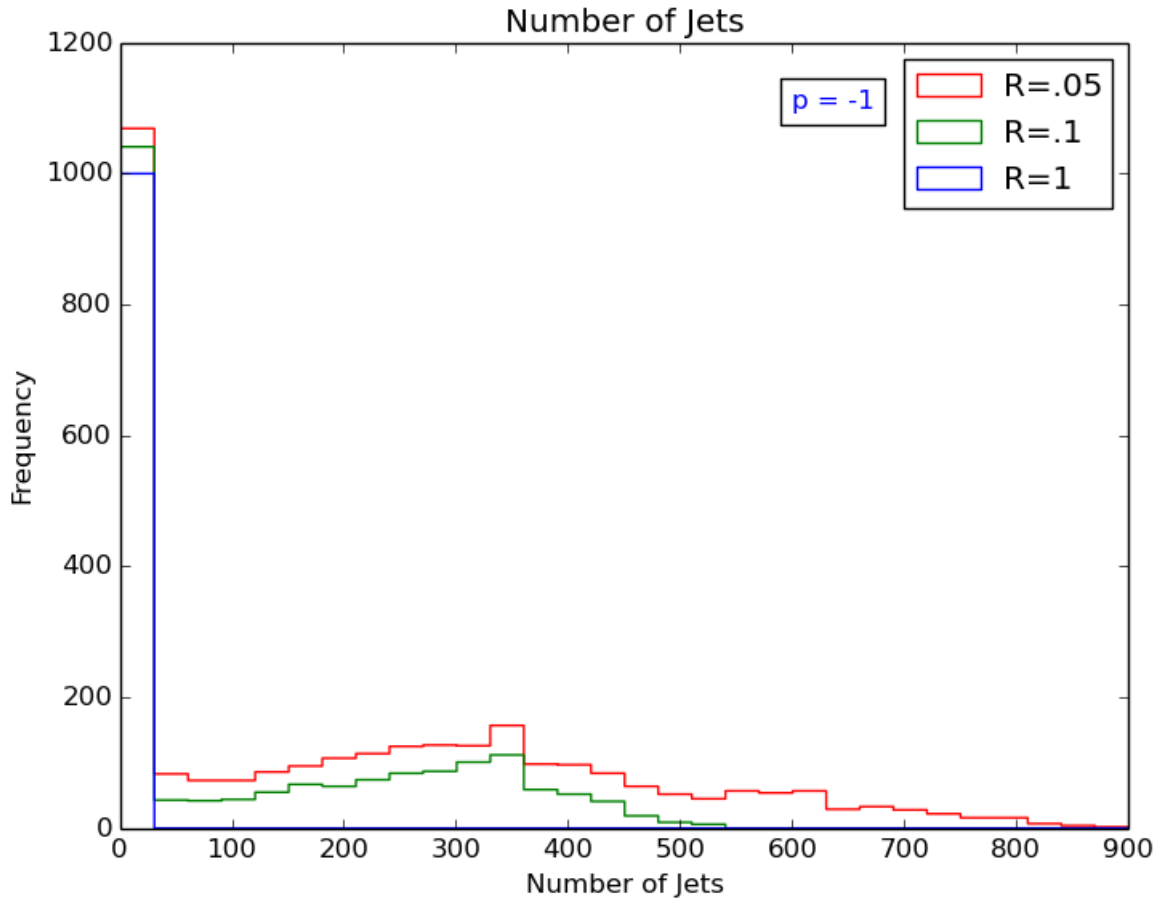


Figure 5.3: Number of jets for different values of R. Note that here we are looking at the difference on the y axis.

6. C++ Packages

6.1 Pythia, FastJet and ROOT

Pythia is a general-purpose event generator. It is extensively used for studying physics at LHC. At the beginning the code was written in `Fortran 77`, recently it has been moved to `C++`. The latest version of Pythia (8.1) was released in 2007, which is written in `C++` (Buckley et al., 2011).

In general Pythia performs a similar simulations as our model of the parton shower. However, it accounts for more of the underlying physics,that we have not covered in our model. Such as the quark flavour or in general the type of the particle which is splitting.

`FastJet` is an analysis tools, it written in `C++`. It includes efficient implementation of all widely used sequential recombination jet algorithms. In the case of jet clustering, the working principle is the same as the algorithm that we implemented in chapter 4, but it is much faster (Buckley et al., 2011).

`ROOT` is a scientific software library that provides a wide range of tools that can be used for big data processing. `ROOT` provides a histogrammming packages (which we used), as well as graphing. This alongside with other features concerning data analysis.

6.2 Our Simulation

In our work we generated 1000 events using Pythia 8.1 and analysed them using `FastJet` using the configuration $dijet$, which is the a simulation of a collision that produces two particles. First two protons will collide and we focus on three types of interactions, quark gluon produces quark gluon, two quarks produces two quarks and quark anti quark produces two gluons.

We extracted the same observables as in 5.1, 5.3 and 5.2 the pseudo-mass, the number of constituents in the jet with highest energy and the number of jet observables, and using `ROOT` we produced histograms of these observables. In this analysis the value of the parameter R is set to 1.

The `C++` code that we used in this work is based on the code shared with us form Samuel Meheen and it is available on https://github.com/smeehan12/PythiaSimulatorAnalysis

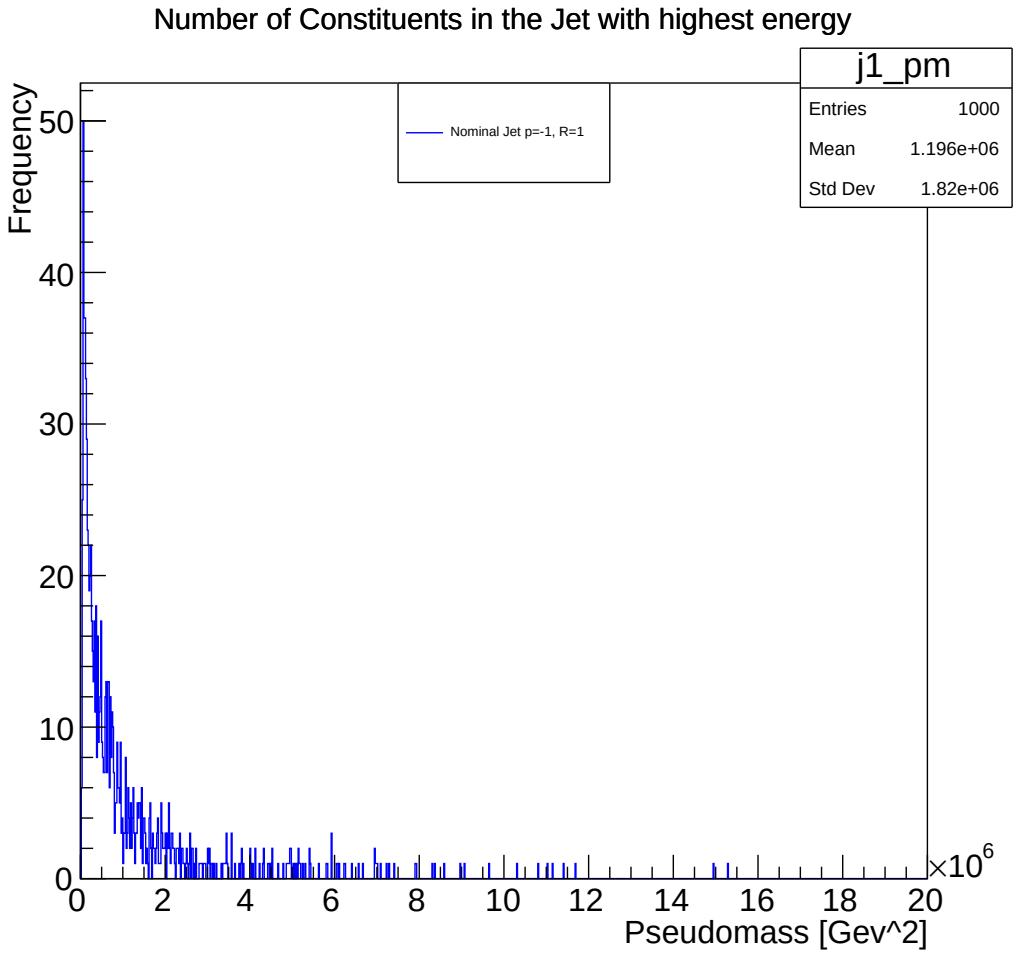


Figure 6.1: A histogram of the pseudo-mass of a jet observable obtained from analysing 1000 events. R is chosen to be 1.

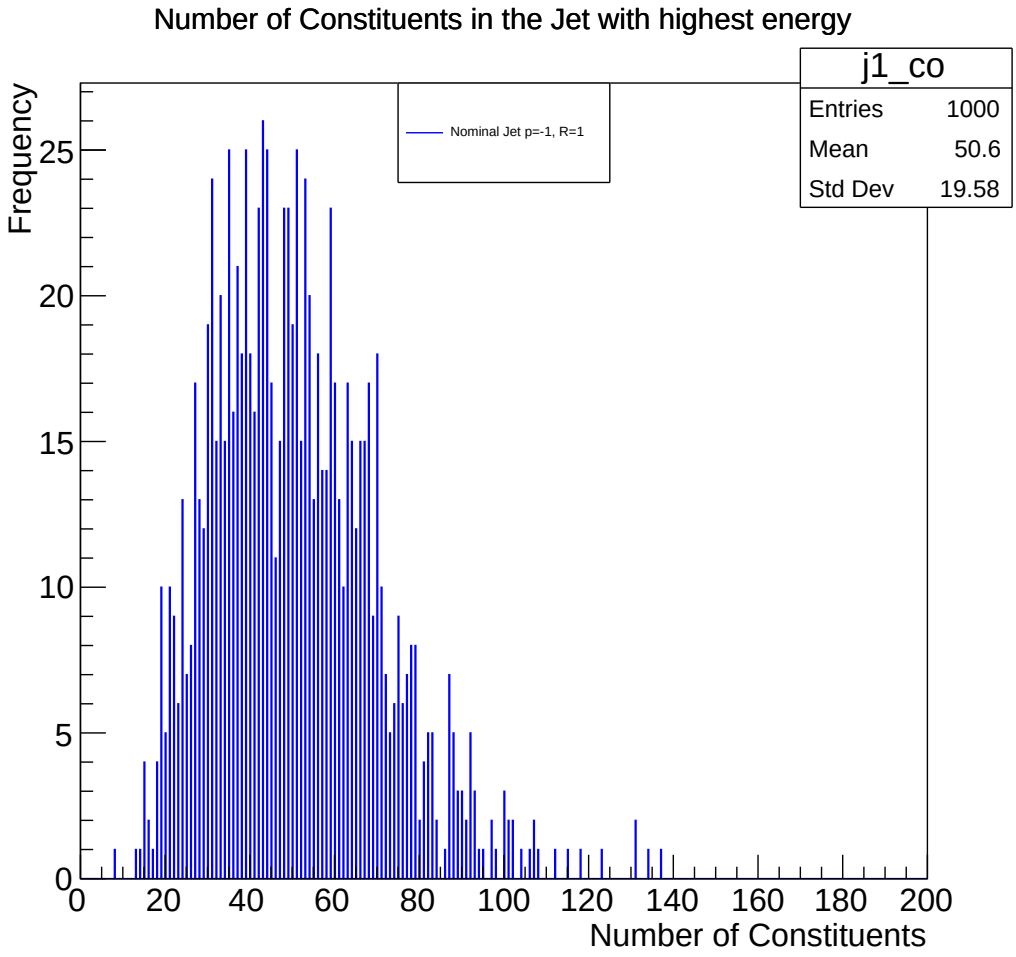


Figure 6.2: A histogram of the number of constituents in the jet with highest energy obtained from analysing 1000 events.

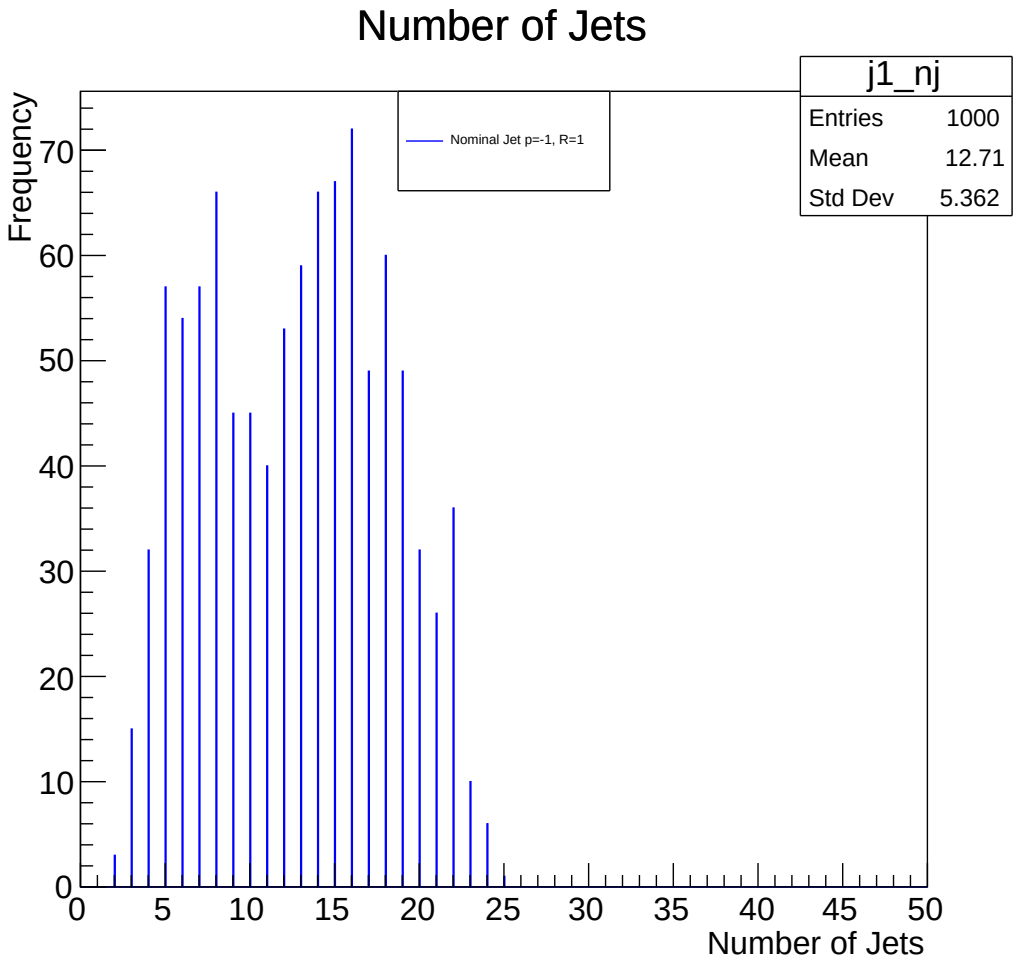


Figure 6.3: A histogram of the number of jet obtained analysing from 1000 events.

7. Conclusion

We have implemented a toy Monte Carlo simulation for the process that happens after the collision of two protons that produces two partons (parton shower) in `Python`. The products were grouped using an implementation of anti-k_t algorithm and jet observables were extracted from the data. Then the we did the same simulation using `C++` packages `Pythia`, `FastJet` and `Root` where the `Pythia` is used for generating the parton shower and `FastJet` for analysing the data and `ROOT` for drawing the histograms. It is noted that even though unlike in `Pythia` the simple toy simulation of the parton shower in `Python` does not account for some physical properties like the particle type, the results from extracting the observables were qualitatively close in both cases.

References

G. Aad et al. The ATLAS Experiment at the CERN Large Hadron Collider. *JINST*, 3:S08003, 2008. doi: 10.1088/1748-0221/3/08/S08003.

Georges Aad et al. Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC. *Phys. Lett.*, B716:1–29, 2012. doi: 10.1016/j.physletb.2012.08.020.

C. F. Berger et al. Snowmass 2001: Jet energy flow project. *eConf*, C010630:P512, 2001.

Gerald C. Blazey et al. Run II jet physics. In *QCD and weak boson physics in Run II. Proceedings, Batavia, USA, March 4-6, June 3-4, November 4-6, 1999*, pages 47–77, 2000. URL http://lss.fnal.gov/cgi-bin/find_paper.pl?conf-00-092.

Andy Buckley et al. General-purpose event generators for LHC physics. *Phys. Rept.*, 504:145–233, 2011. doi: 10.1016/j.physrep.2011.03.005.

Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. The Anti-k(t) jet clustering algorithm. *JHEP*, 04:063, 2008. doi: 10.1088/1126-6708/2008/04/063.

Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. FastJet User Manual. *Eur. Phys. J.*, C72: 1896, 2012. doi: 10.1140/epjc/s10052-012-1896-2.

The CMS Collaboration. The cms experiment at the cern lhc. *Journal of Instrumentation*, 3(08): S08004, 2008. URL http://stacks.iop.org/1748-0221/3/i=08/a=S08004.

Luc Devroye. Sample-based non-uniform random variate generation. In *Proceedings of the 18th Conference on Winter Simulation*, WSC '86, pages 260–265, New York, NY, USA, 1986. ACM. ISBN 0-911801-11-1. doi: 10.1145/318242.318443. URL http://doi.acm.org/10.1145/318242.318443.

S. D. Ellis, J. Huston, K. Hatakeyama, P. Loch, and M. Tonnesmann. Jets in hadron-hadron collisions. *Prog. Part. Nucl. Phys.*, 60:484–551, 2008. doi: 10.1016/j.ppnp.2007.12.002.

Lyndon Evans and Philip Bryant. Lhc machine. *Journal of Instrumentation*, 3(08):S08001, 2008. URL http://stacks.iop.org/1748-0221/3/i=08/a=S08001.

D. Griffiths. *Introduction to Elementary Particles*. Physics textbook. Wiley, 2008. ISBN 9783527406012. URL https://books.google.rw/books?id=w9Dz56myXm8C.

Stefan Höche. Introduction to parton-shower event generators. 2014. URL https://inspirehep.net/record/1328513/files/arXiv:1411.4085.pdf.

M.H. Kalos and P.A. Whitlock. *Monte Carlo Methods*. Monte Carlo Methods. Wiley, 1986. ISBN 9780471898399. URL https://books.google.rw/books?id=GyfvAAAAMAAJ.

Y. Nagashima and Y. Nambu. *Elementary Particle Physics: Quantum Field Theory and Particles*. Number v. 1. Wiley, 2010. ISBN 9783527630103. URL https://books.google.rw/books?id=J0l8s3pdOksC.

www.python.org Python foundation. Python reference manual. Technical report, The Us, 2007.

Gavin P. Salam. Towards Jetography. *Eur. Phys. J.*, C67:637–686, 2010a. doi: 10.1140/epjc/s10052-010-1314-6.

Gavin P. Salam. Elements of QCD for hadron colliders. In *High-energy physics. Proceedings, 17th European School, ESHEP 2009, Bautzen, Germany, June 14-27, 2009*, 2010b. URL https://inspirehep.net/record/880643/files/arXiv:1011.5131.pdf.

Stefan Weinzierl. Introduction to Monte Carlo methods. 2000.

Appendix

The codes that are used in this essay are available on https://github.com/KhalidOmer/MonteCarloTutorial.

The file `README.md` contains a brief description of the codes and what they do as well as how to run them.

Note that all codes are written in `python`, except for chapter six the codes are in `C++`.

The codes are:

Chapter two:

- Obtaining a sample with $p(x) = \frac{1}{x+\epsilon}$, the file name is `accept-reject.py`.
- For the calculation of π and the error calculation see `Calculation_of_pi.y` and `uncertainty_in_pi_calculation.py`.

Chapter three:

- The two dimensions parton shower: `two_D_partonshower.py`.
- The three dimensions parton shower: `partonshower3d.py`

Chapter four:

- Anti-k_t algorithm: `antikT_algorithm.py`.

Chapter five:

In this chapter the codes are divided into two files, one file for running the code and the other is for producing the histogram.

- The Pseudo-mass observable:
  `Pseudomass_observable.py`
  and `hist_pseudomass.py`.
- The number of constituents in the jet with highest energy observable:
  `number_of_constituents_observable.py`
  and `hist_of_n_of_constituents.py`.
- The number of jets observable:
  `number_of_Jets_observable.py`
  `hist_n_of_Jets.py`.

Chapter six:

The codes are available in the folder named PythiaSimulatorAnalysis. A file called `README.md` inside this folder contains the instruction on how to run these codes.