

Limavady
Grammar School

Aoife
Rebecca
Darcy
Ella

Introduction

In this project we developed skills in coding with Python. Using these skills we were able to write the code necessary to solve projectile motion questions. This provided us with an insight into the advantages and disadvantages of using this method over completing these calculations manually.

Aims

Using Python Programming Language to write a programme to find the distance travelled by a 10 kg projectile fired at 15 m/s at 45° above the horizontal from a point 2 m above the ground.

Background Information

Python is a very popular general-purpose programming language that can be used for a wide variety of applications. It is used to build websites, software, and conduct analysis⁽¹⁾. Python also offers the ability to easily automate processes through scripting, making it key for software testing, troubleshooting and bug tracking. It plays a key role in data science tasks and is used to perform complex statistical calculations, visualize data and create machine learning algorithms⁽²⁾.

Python can also be used to solve many of the worlds complex physics problems. However, it may also be used to solve rather basic physics equations as well, for example it can be used for physics calculations involving constant velocity. This can be done by simulating the motion of an object. If we know the object's *position* (*x*-coordinate) at a particular *time* *t* and it's *velocity* *v* at that time, this will allow us to find the object's position a small time later Δt . These values can be subbed into the equation:

$$x_t + \Delta t = x_t + v\Delta t.$$

Another way it can be used to solve basic physics is for constant acceleration, the only difference here is that, since the object is now accelerating, its velocity is changed at every step and must be updated accordingly. The updated formula would be:

$$v_t + \Delta t = v_t + a\Delta t$$

Projectile motion is the motion of an object thrown or projected into the air, subject to only the acceleration of gravity. The object is called a projectile, and its path is called its trajectory⁽³⁾. The resultant path is in effect a combination of 2 motions - horizontal and vertical. This allows us to apply the equations of motion separately. Particles in a projectile follow a curved path known as a parabola.

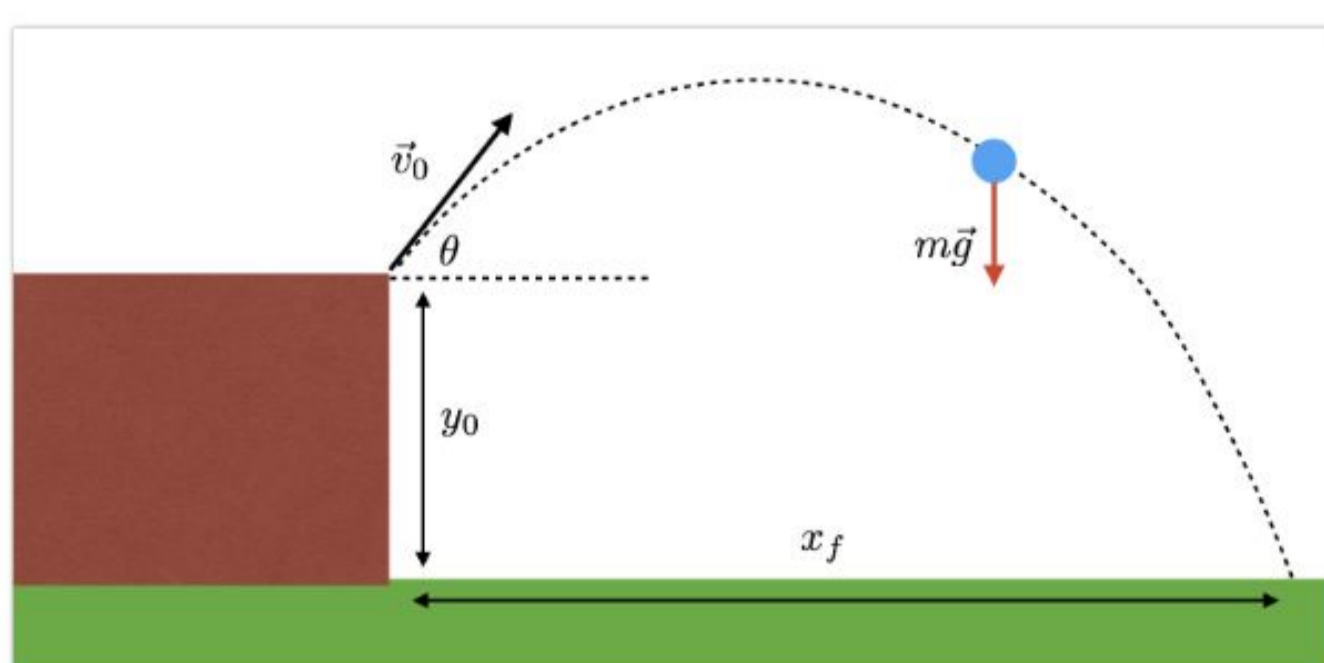


Figure 1: Diagram of a projectile.

Experimental method

To code the simulation for a projectile motion we used the following method;

- Set the initial conditions of the projectile
- Make a loop over time steps
- In the loop, update the velocity of the projectile (only its y-coordinate changes)
- Update the position coordinates of the ball
- Update the time

```
from numpy import sin, cos
v_y = 15*sin(45)
d_y = 2
a = -9.81
t = 0
dt = 0.05
v_x = 15*cos(45)
d_x = 0

while d_y >= -2:
    v_y += a * dt
    d_y += v_y*dt + 1/2*a*dt**2
    d_x += v_x * dt
    t += dt
```

Figure 2: Python code written to simulate projectile motion

Results

The ball hits the ground, -2m below its starting position, at time $t = 2.80s$. At this time it is at a horizontal distance of 22.1m from its starting point, moving with a vertical velocity of -14.7m/s and a horizontal velocity of 7.9m/s.

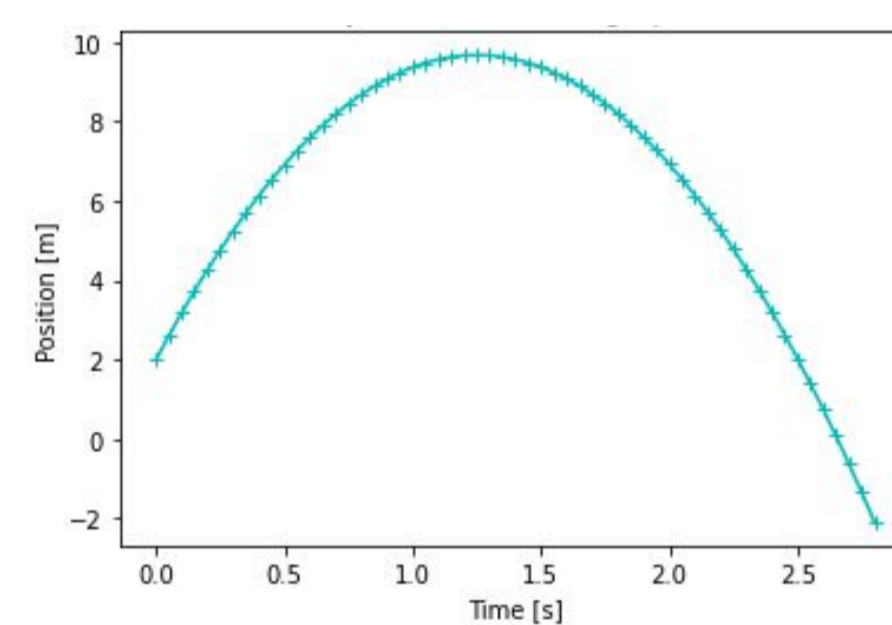


Figure 3: Graph showing the "Y position against time"

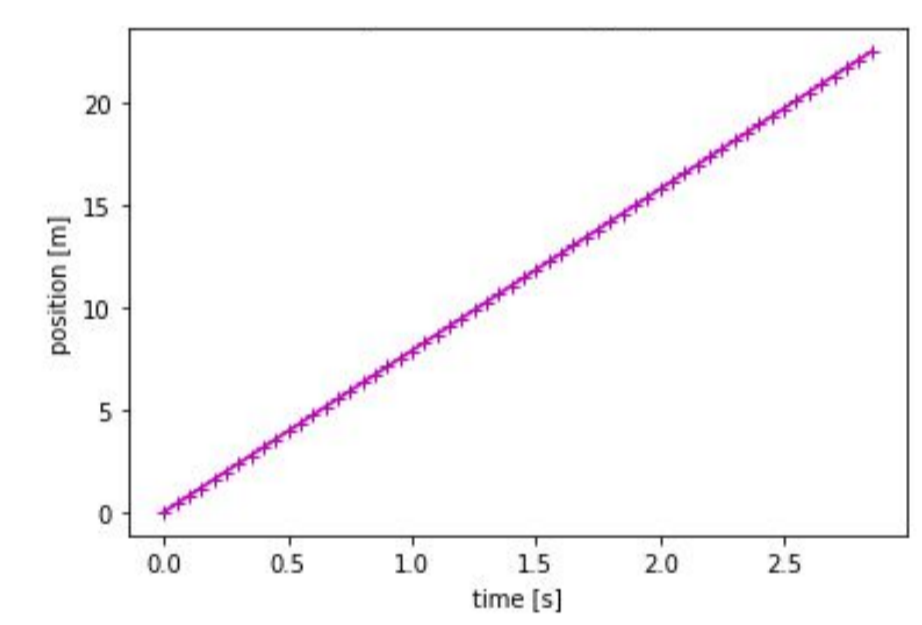


Figure 4: Graph showing the "X position against time"

Analysis & Conclusions

Advantages of simulating projectile motion using Python:

- Good method of graphically displaying projectile motion
- Efficient technique for analysing large amounts of data
- Once the code is correct there is no room for human error
- Provides a clear representation of calculations

Disadvantages of simulating projectile motion using Python:

- Requires a comprehensive understanding of Python code
- May be difficult to apply to more complex physics problems

References

- 1) <https://opensource.com/resources/python>
- 2) <https://builtin.com/software-engineering-perspectives/python>
- 3) <https://courses.lumenlearning.com/suny-physics/chapter/3-4-projectile-motion/>

Limavady Grammar
School

Annabelle
Emily
Norman
Olivia

Introduction

Through this project we made use of Python to create a histogram via coding. The histograms produced show that it is most likely that only one lepton will be produced in each collision event inside the Large Hadron Collider.

Aims

To use Python Programming Language to display data in the form of a histogram from high energy collisions in the Large Hadron Collider.

Background Information

The Large Hadron Collider:

The Large Hadron Collider is the world's largest and most powerful particle accelerator. It consists of a 27-kilometre ring of superconducting magnets with a number of accelerating structures to boost the energy of the particles along the way⁽¹⁾.

ATLAS:

ATLAS is a general-purpose particle physics experiment at the Large Hadron Collider (LHC) at CERN. ATLAS is the largest detector of its kind and is designed to record the high-energy particle collisions of the LHC, which take place at a rate of over a billion interactions per second in the centre of the detector. ATLAS is a large global collaboration with a common aim to further understand the fundamental constituents of matter and their interactions⁽²⁾.

Fundamental Particles - Leptons:

Leptons are a subgroup of elementary particles that make up the matter all around us. There are six different types of lepton that are arranged in pairs (generations). The first generation is made up of the lightest and most stable particles, which are the electron and electron neutrino, and the heavier less-stable particles belong to the second and third generations; the muon and muon neutrino in the second generation and the tau and the tau neutrino in the third generation. The electron, muon and tau all have an electric charge and a sizeable mass, whereas the neutrinos are electrically neutral and have very little mass⁽³⁾.

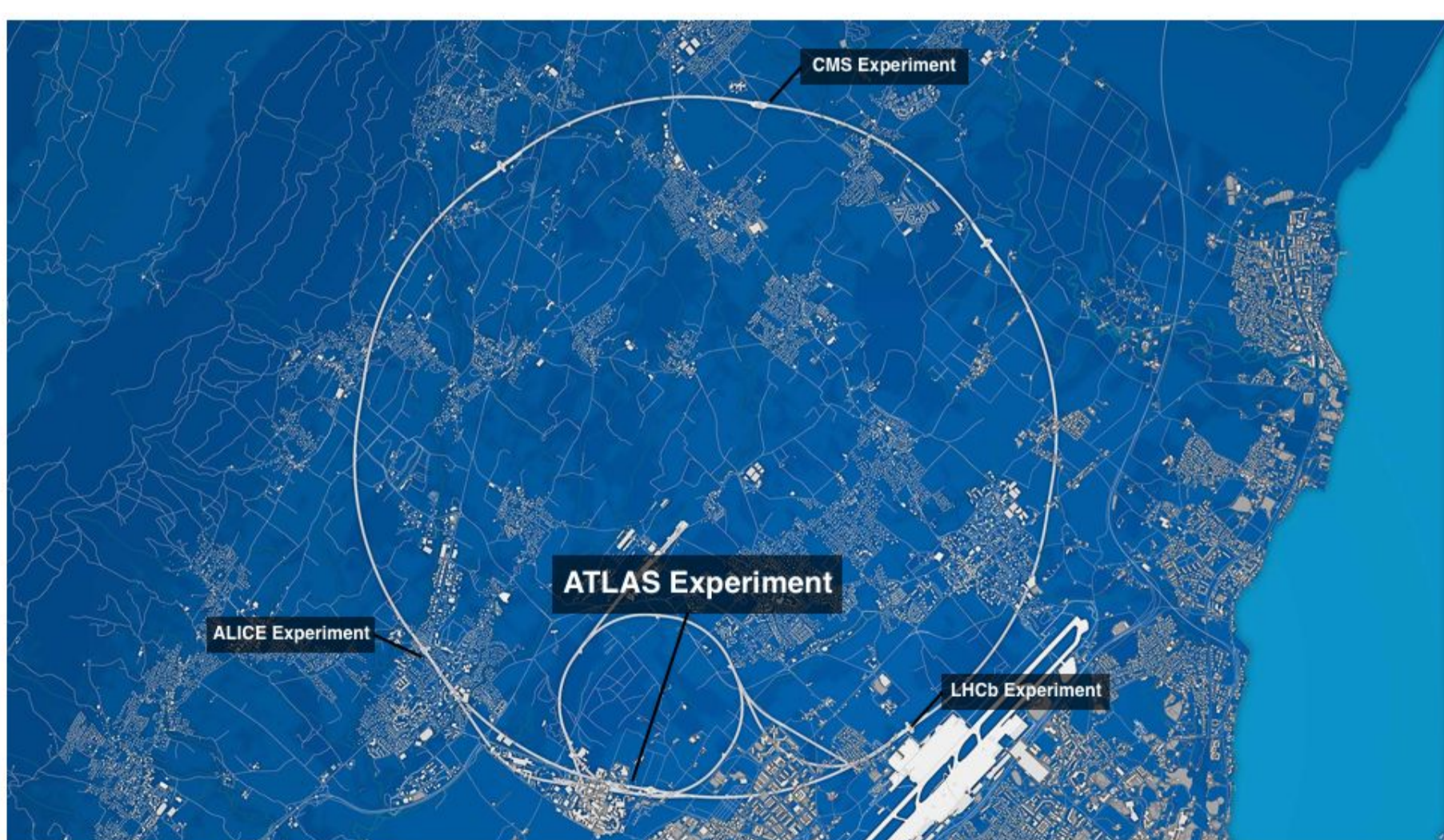


Figure 1: A diagram of the Large Hadron Collider at CERN, showing the ATLAS collision point⁽²⁾.

Experimental Method

1. Load the data from the ATLAS database by using “uproot”.
2. To fill histogram extract the number of leptons from our TTree using “uproot”. A TTree is a container that keeps track of the information from a collision event. Fill the histogram using the “.fill()” function from “hist”.
3. Plot the histogram using the “.plot()” and “plt.show()” functions.
4. Title the histogram and create a labelled x-axis.
5. Normalise the histogram so that it shows the proportion of each number of leptons produced, up to a maximum value of 1, instead of the number of collision events that produced the different numbers of leptons.

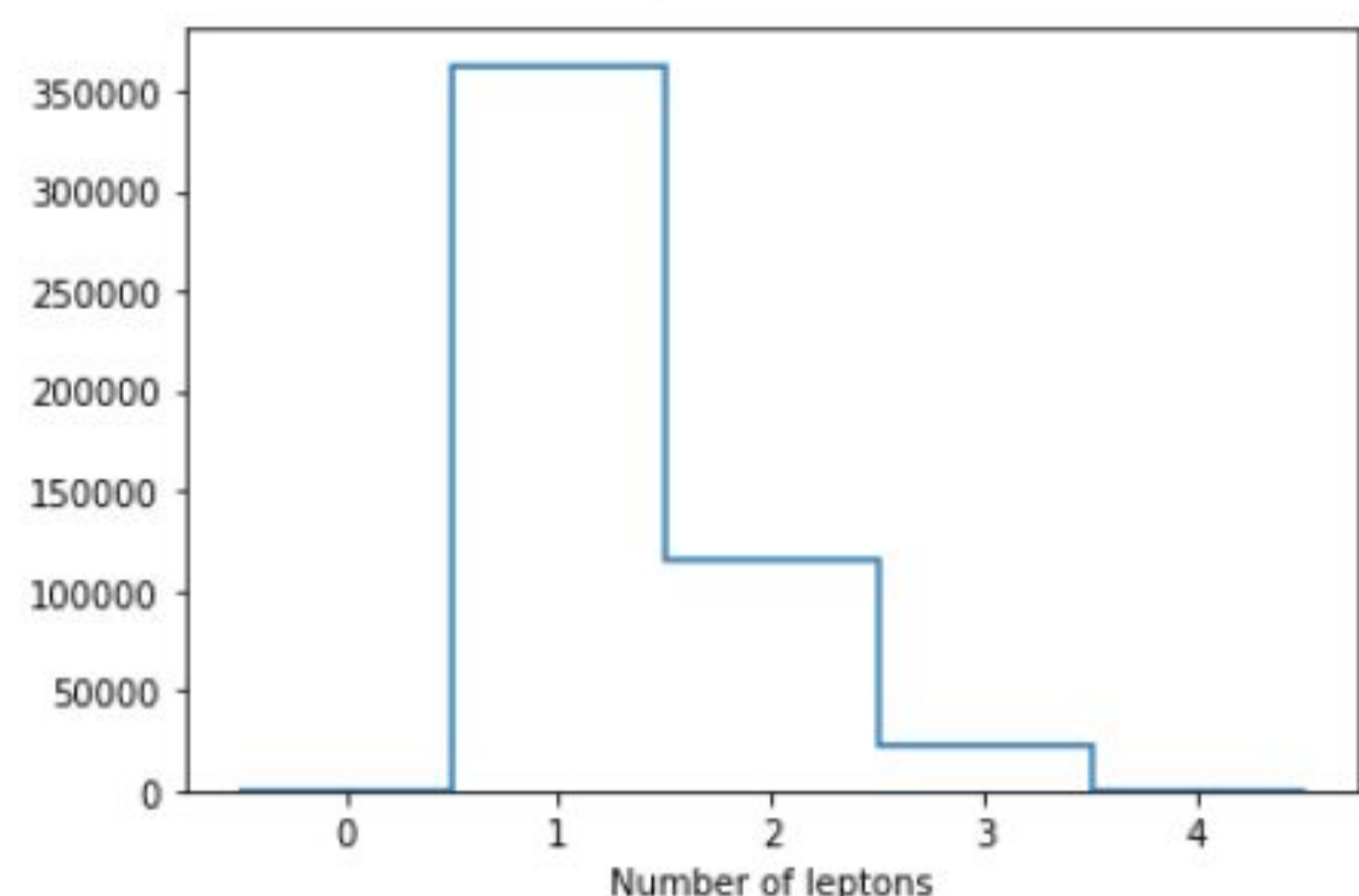


Figure 2: A histogram showing the number of leptons produced for the collision event.

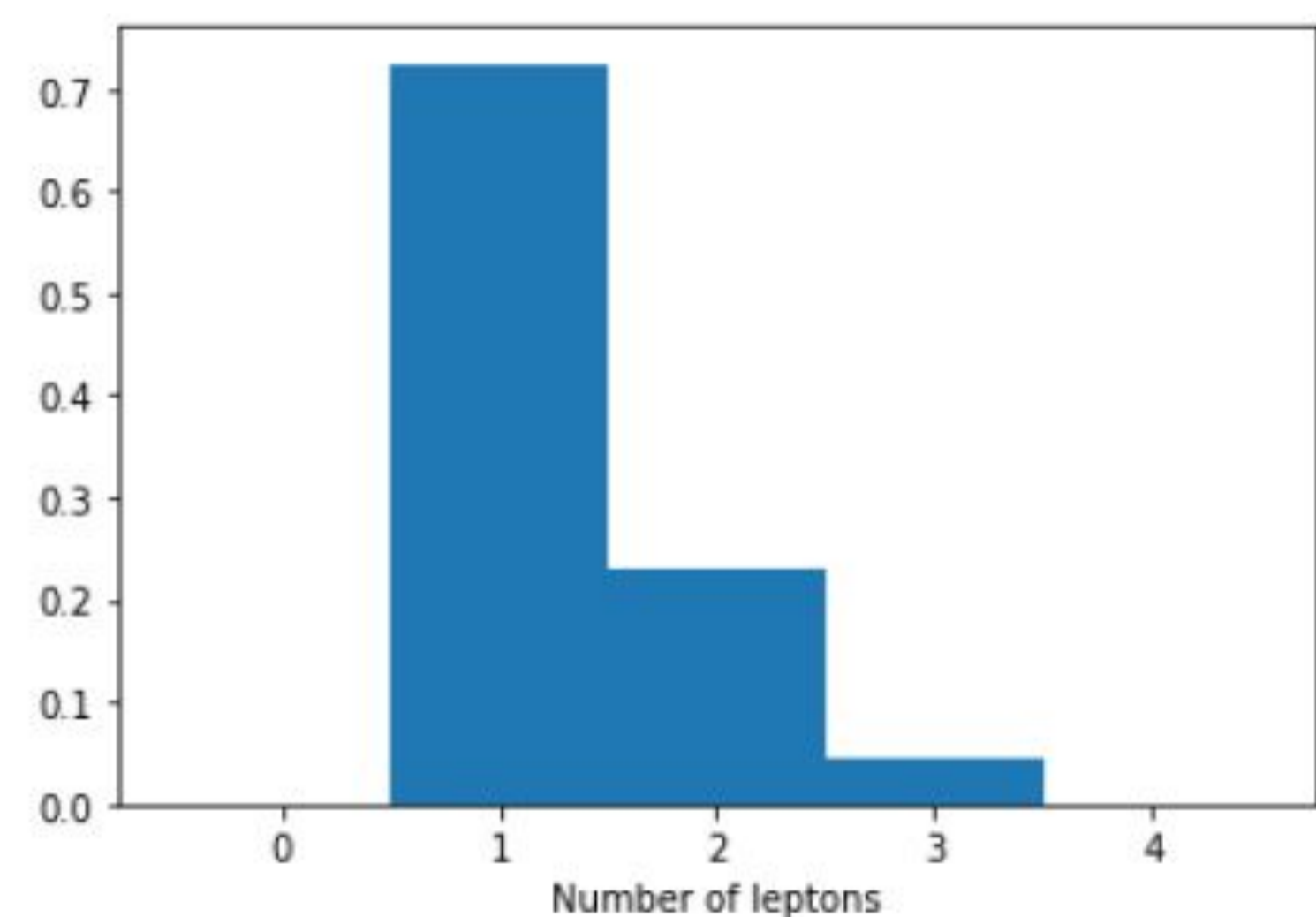


Figure 3: A normalised histogram, with y axis totalling to 1, showing the proportion of the number of leptons produced in collision events.

Analysis & Conclusions

By displaying large sets of data in histograms, the information can be viewed in a clear, concise way, allowing major features of distribution of the data to be seen. From the LHC data analysed, we were able to see that how many leptons were produced in each collision event. The histograms produced show that, most commonly, only 1 lepton is produced per collision event.

References:

1. <https://home.cern/science/accelerators/large-hadron-collider>
2. <https://atlas.cern/about>
3. <https://home.cern/science/physics/standard-model>

Big Data: ATLAS - Finding the Mass of the Z^0 Boson

Limavady Grammar
School

Callum
Ryan
Leo

Introduction

In this project we used data from CERN to reconstruct the Z^0 boson in order to determine its mass. Via the use of python code, we determined the mass of the Z^0 boson to be approximately equal to 90GeV.

Aims

To use Python to analyse data from high energy collisions in the Large Hadron Collider and determine the mass of the Z^0 boson.

Background Information

The Large Hadron Collider (LHC) is the world's largest and most powerful particle accelerator. It consists of a 27 km kilometer ring of superconducting magnets and accelerating structures to propel particles⁽¹⁾. The ATLAS project at CERN is a particle physics experiment which is designed to use the LHC's capabilities to further the world's understanding of particle physics.⁽²⁾ The ATLAS project investigates a wide variety of particles from the Higgs Boson to what makes up dark matter⁽³⁾.

There are 2 types of fundamental particles, quarks and leptons. Each of these groups have 6 particles, related in pairs. The 6 quarks are paired into 3 generations; the up and down quark, the charm and the strange quark; and the top and bottom quark. The 6 leptons are arranged into 3 generations; electrons and electron neutrinos, muons and muon neutrinos, and the tau and tau neutrino⁽⁴⁾. There are 4 fundamental forces, 3 of which result from the exchange of particles called bosons. The W and Z bosons are responsible for the weak force⁽⁴⁾.

The Z^0 Boson is very unstable and does not live long enough to be detected, so to find the Z^0 Boson we reconstruct it from it's decay products. We will reconstruct Z^0 Bosons which have decayed into 2 leptons. To conserve charge and electron number these leptons need to have an opposite charge and the same flavour, meaning we will be looking for a muon and an antimuon or an electron and an antielectron.

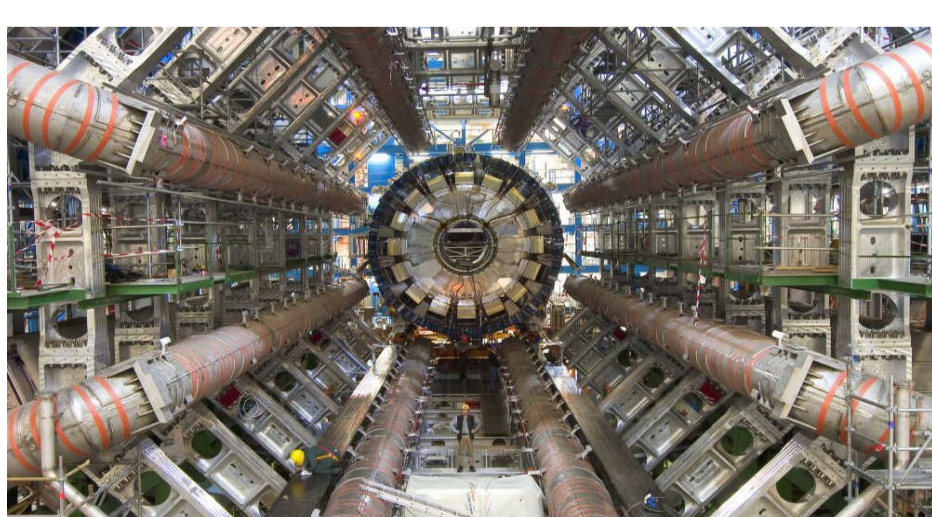


Figure 1: The Large Hadron Collider at CERN⁽⁵⁾

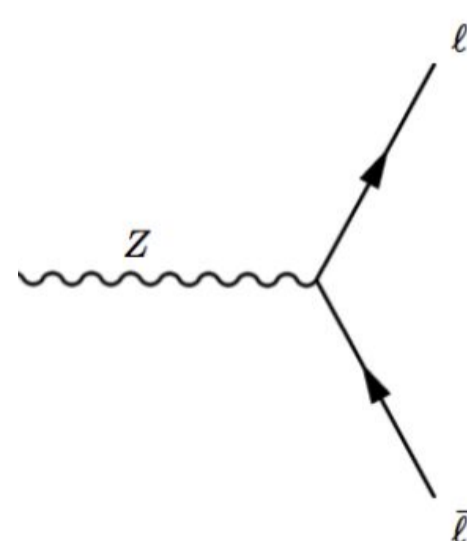


Figure 2: Feynman diagram showing the decay of a Z^0 boson into a lepton and anti-lepton.

Experimental Method

- First, open the data using “uproot”, and inspect the contents of the file.
- Use the “.arrays method” to access events from specific columns, and then define a histogram, with x axis named mass/GeV.
- Make cuts in the data, on the fact that there are two leptons of the same flavour, and then cut the data again on the fact that those two leptons are oppositely charged, to reconstruct the Z^0 boson.
- Import “matplotlib” and plot the histogram.

Results

This graph shows the mass of the particles fitting the above criteria. The strong peak at around 90GeV shows that this is the mass of the Z^0 Boson.

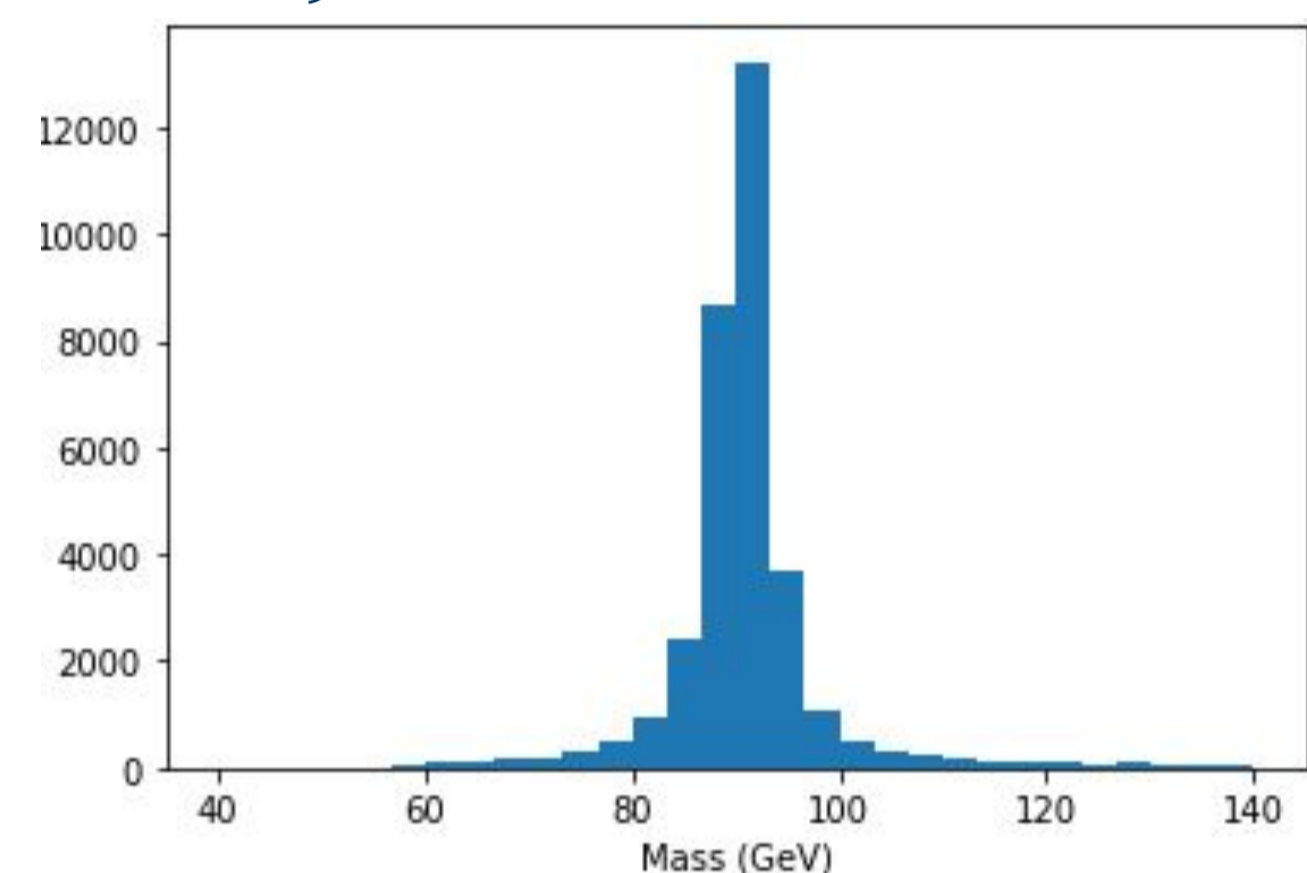


Figure 3: Graph showing the mass of a Z^0 Boson

Analysis & Conclusions

Physicists theorise that interactions between quarks and antiquarks are more complex and that the quarks and antiquarks quickly come toward each other and share a particle as seen in the Feynman diagram below. They both turn into two Z^0 bosons which then decay in the same way as mentioned above. Exploring this interaction could be a further method to determine the mass of the Z^0 boson.

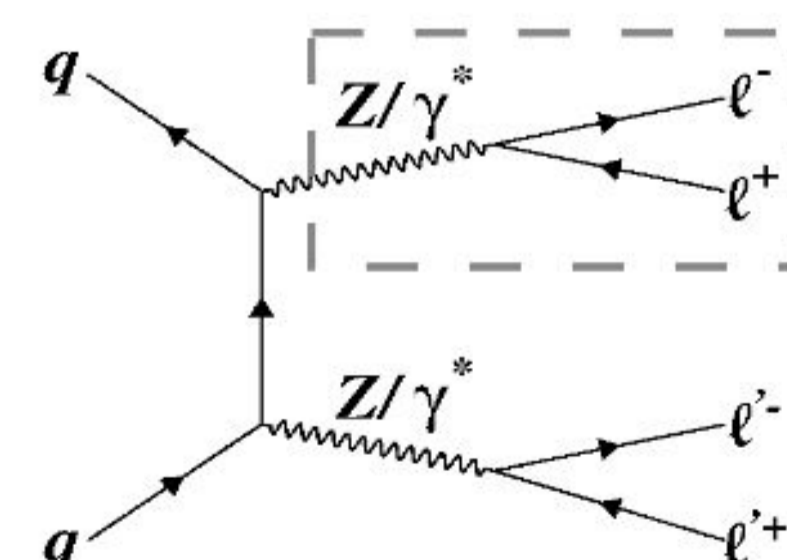


Figure 4: Feynman diagram showing the interaction between a quark and an antiquark.

References

1. <https://home.cern/science/accelerators/large-hadron-collider>
2. <https://atlas.cern/about#:~:text=ATLAS%20is%20a%20general%20purpose,the%20frontiers%20of%20scientific%20knowledge.>
3. <https://home.cern/science/experiments/atlas#:~:text=ATLAS%20is%20one%20of%20two,could%20make%20up%20dark%20matter.>
4. <https://home.cern/science/physics/standard-model>
5. <https://www.npr.org/2010/12/28/132406486/the-hunt-is-on-massive-collider-churning-out-data>