The ALICE Forward Hadronic Calorimeter: Construction, Testing, and Understanding the Second Prototype.

Master's in particle physics

This project has been focused on the development and understanding of the second prototype of the ALICE Foward Hadronic Calorimeter FoCal-H. Specifically the construction, testing and data analysis of it, with a focus on simulating results to compare the experimental data with predicted results.

Report written by:

Christian Lykner
KUid: grb189

Supervision by:
Ian Bearden

UNIVERSITY OF
COPENHAGEN

Seal of the university of Copenhagen
Founded in 1479

Written as part of *Msc.Scien. Physics*
University of Copenhagen, 2020-2023
Denmark

Date of hand in: 22/5/2023

Abstract

This report is written as the master's thesis for the master's project in physics at the university of Copenhagen. In this report I will cover the various physics motivations behind the ALICE forward calorimeter upgrade intended to be installed in time for run 4 (2027 - 2029), as well as the general physics of calorimetry to ensure a proper background is established to explain the physical processes taking place. Further more this report will then also cover my participation in the construction of the second prototype of the forward hadronic calorimeter, as well as the data taking processes of the September and November Beamtests in 2022. Further on the report will include a description of two main simulation packages as well as their use for simulating the second prototype and how it responds to particle beams at varying energies. Finally an analysis of the distribution of energy deposits throughout the detector will act as the finale of the project, including various simulation methods, configurations and even different beam energies ranging from 200GeV to 1 TeV.

Contents

1 Introduction 1

1.1 The goal of FoCal 1

1.2 The FoCal-H prototypes 3

1.2.1 The project goals 4

2 The physics motivation 5

2.1 The basics of calorimetry 5

2.1.1 The electromagnetic interactions 6

2.1.2 The photon EM interactions 7

2.1.3 The strong interaction 8

2.1.4 The Bjorken x 11

2.1.5 Shower developments 13

2.2 The goals of FoCal 15

2.2.1 Nuclear Shadowing 15

2.2.2 Jet-quenching 20

2.2.3 Long range correlations 22

3 The 2nd prototype, construction and the beamtests 24

3.1 The prototype 24

3.2 The beamtests 29

4 The simulations 36

4.1 The testbeamsimulation simulation - geant4 version tests 36

4.1.1 The contents of the main folder 37

4.1.2 Running the Testbeam simulation package 42

4.1.3 Visualizing the setup 43

4.1.4 The longitudinal problem 46

4.2 The Focal_sim simulation 48

4.2.1 The VMC folder 50

4.2.2 Running the VMC implementation 53

4.2.3 The Geant4 implementation 58

4.2.4 Running the Geant4 implementation 62
4.2.5 Final notes on the Focal_sim simulation package 65

5 Results 66

5.1 Expected results and their physical interpretations 66
5.1.1 Electromagnetic showers within FoCal-E 66
5.1.2 Muons passing through the calorimeters 67
5.1.3 Hadronic showers starting in FoCal-H 68
5.1.4 Hadronic showers starting in FoCal-E 68
5.2 Early comparisons between beamtest results and simulations 69
5.3 The 350GeV simulations 73
5.4 Energy distribution within FoCal-H 80
5.4.1 350GeV - VMC implementation 80
5.4.2 Other energies - VMC 86
5.4.3 Energy distribution of various parts of the calorimeter - VMC . . 87
5.4.4 Runs with FoCal-E - g4 92
5.4.5 Comparison g4 vs. vmc 95
5.5 Final beamtest data 99

6 Conclusion 102

A Appendix 105

A.1 Further results 105
A.1.1 Further comparisons between testbeam package and recorded data 105
A.1.2 Additional heatmaps from testbeam simulation with geant4 v11.1.1 109
A.1.3 Additional heatmaps from testbeam simulation with geant4 v10.7.3 110
A.1.4 Extra runs with updated testbeam simulation package 111
A.1.5 Extra digitized outputs 112
A.1.6 Full profile 350 GeV without FoCal-E - g4 implementation 117
A.1.7 Full profile 1 TeV without FoCal-E - vmc implementation 120
A.2 Visualized setups 123

1 Introduction

1.1 The goal of FoCal

The letter of intent for the ALICE Foward Calorimeter (henceforth referred to as FoCal) [6] states the intention to insert a foward calorimeter in the ALICE detector as part of upgrades in preparation for data taking in 2027-2029. The FoCal Calorimeter is intended to consist of two parts, an electromagnetic calorimeter (FoCal-E) and a hadronic calorimeter (FoCal-H) in this project I have been a part of the group working on the second prototype of FoCal-H.

The foward calorimeter is a proposed upgrade to the ALICE detector at the LHC, intended to be installed for use by run 4 in the period of 2027-2029. The purpose of the ALICE detector is to study the physics of strongly interacting highly dense partonic matter [1]. The FoCal detectors are meant to act as an extension of this purpose by studying the partonic structure down to the low bjorken-x and low momentum transfer regime wherein the structures are predicted to develop in a nonlinear fashion. This nonliearity occurs due to QCD processes that become increasingly likely in low-x scenarios. Thus the broader objectives of these ALICE upgrades is to study this transition from linear to nonlinear development. Because of this the various foward parts of alice seeks to cover a wide range of x and Q allowing this transition to be studied.

The FoCal upgrade is intended to be located in the front of the ALICE detector at a small angle as to cover low pseudorapidities in the range of $3.4 < \eta < 5.8$ within ALICE. This will allow the calorimeters to measure isolated photons and hadrons, which are particularly sensitive to Gluon density and saturation. This is because these isolated photons will most commonly occur due to quark-gluon Compton scattering.[7]

Specifically, the FoCal upgrade has several physical phenomena that it seeks to study in detail and as such the FoCal calorimeters are constructed with the accomplishment of these goals in mind, these goals are outlined as the following:

Measure the gluon density in protons and lead nuclei and quantify its nuclear modification at small x and Q_2[7]: Measurements and fits have demonstrated that for low momentum fractions, specifically in the realm of $x \approx 10^{-2}$, the gluon density will be notably lower in heavier nuclei compared to free neutrons and protons. Thus measurements of this suppression is limited and can only be determined indirectly and with limited x-dependence constraints. However direct measurements of photons at forward rapidity gives clear x-dependence and gluon density constraints, which allows the possibility to test whether this suppression is dependant on the type of parton and on the smattering process. This suppression is called nuclear shadowing and will be discussed further in the report.

Explore the physical origin of shadowing effects[8]: As mentioned in the above the suppression of the gluon density is called nuclear shadowing. However more broadly shadowing effects occur for nuclear parton distributions and affect observable properties. Specifically for small values of the bjorken-x, the hadronic structure of the parton distribution will develop non-linearly. Due to the structure being described by momentum distributions which have a scale dependence which is affected by and can be calculated with QCD interactions. Thus the high gluon density at low bjorken-x will result in changes to the distribution.

Investigate the origin of long range flow-like correlations in pp and p–Pb collisions[8]: There have been consistently observed azimuthal correlations in pp and p-Pb collisions for long range in η. These correlations between particles observed within the central ALICE detector or its forward muon detection system have a particularly large pseudorapidity separation, in the area of $\Delta\eta \sim 5 - 9$. These particular measurements will serve to better understand whether these "ridge" effects are caused by initial-state or final-state effects.

Explore jet quenching at forward rapidity in Pb–Pb collisions[8]: One of the most notable phenomena observed in heavy ion collisions is the alteration of the hadron and jet production due to interactions of high energy partons in Quark Gluon Plasma. Here FoCal will directly provide measurements of jet production and neutral high transversive momentum mesons, at a notably larger rapidity than previously observed, due to the small angle off of the collision beam. This will allow the measurement of the Quark Gluon Plasma density as a function of the rapidity, due to the observable effects that the QGP will have on the jet production, as well as the difference in energy loss between gluon and quark jets.

1.2 The FoCal-H prototypes

The above is a broad description of the FoCal project goals in general, however this project is focused on the development of the hadronic calorimeter FoCal-H, specifically the second protype. In particular this project has focused on finishing up the second prototype, the testing of the second prototype during Beamtests taking place during September and November of 2022, as well as the data-analysis of the results from these tests. Within this data-analysis the focus has been on simulations of the detector, to compare with the results to tune the simulations, to allow for predictions on possible changes for the third prototype.

As such this report will contain the collection of different parts to this project, from assisting with the assembly of the individual modules of the calorimeter, to the data-taking during the beamtests and the setup of the detector, to the data-analysis and simulation of the energy-loss throughout the detector at various energies, which allows us to view where in the detector showers will develop. As well as comparing both simulation and data from runs with and without FoCal-E in front of our calorimeter, to see the effects that the electromagnetic calorimeter can have upon our results.

In the process of this, the project also contains a significant section on how the simulation software works, and how it compares to the physical setup of the detector.

1.2.1 The project goals

1. Aid in finishing the creation of the second prototype in time for beam tests during September and November
2. Assist in the setup and data collecting during the September and November beam tests.
3. Make fits comparing data taken during beam tests to early simulated results.
4. Begin own simulations at various energies, in the process of this, testing various configurations of the simulation software for differences between versions.
5. Get a longitudinal distribution of energy deposits within the simulated detector to grant a better understanding of the shower developments within the detector.
6. In the process of the above: Get an understanding of the software and process used for simulation and data analysis.
7. Finally, compare the energy deposit distributions between different beam energies, as well as between different configurations, specifically with or without the FoCal-E calorimeter in front of the FoCal-H calorimeter.

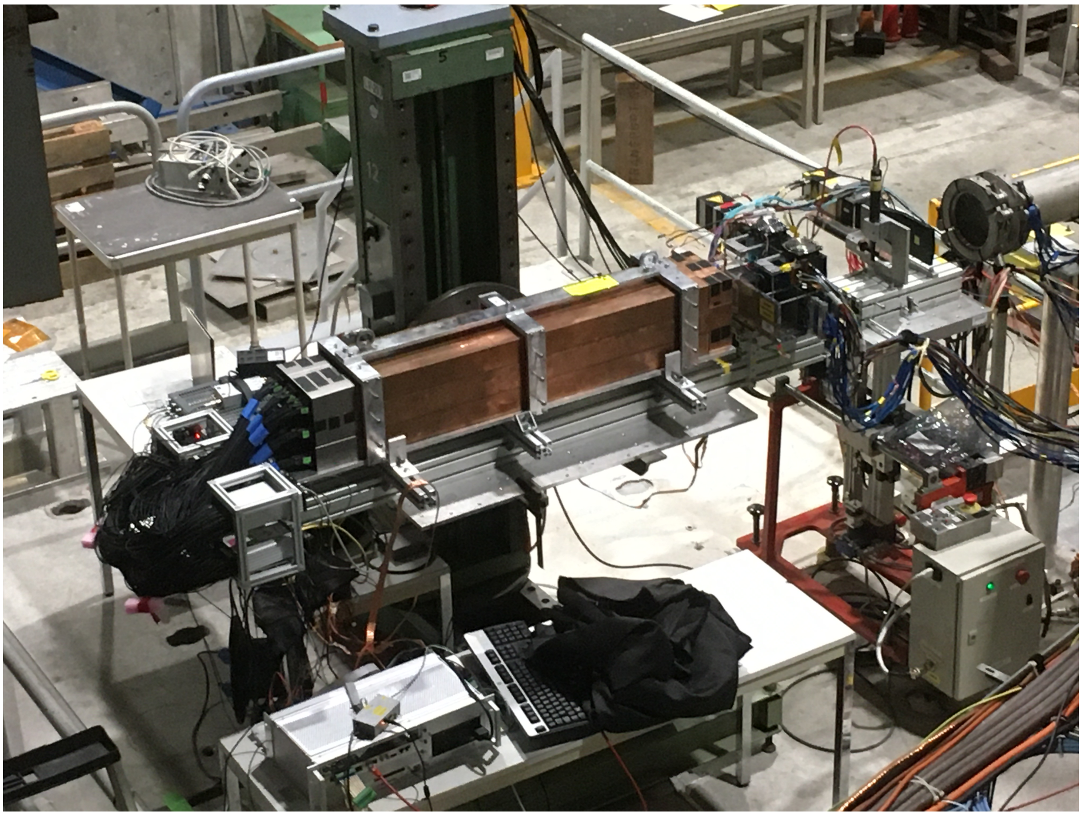


Figure 1: The full setup of the second prototype during the November 2022 SPS beamtest at CERN.

2 The physics motivation

2.1 The basics of calorimetry

In order to adequatly describe the goals of the FoCal upgrade, as well as the methods used to achieve them, we need to build the necessary understanding of the underlying physical processes taking place, and indeed how and why the calorimeters work to achieve these goals.

Fundamentally the key process taking place in calorimetry and collider physics generally is energy transfer of high energy particles into particles within a lower energy medium. These high energy particles will then share that energy with surrounding particles, who will in turn share this with other particles and so on, until the energy of these particles is so low that it only leads to a simple excitation, which will then de-excite to a lower energy state, emitting light in the process. Indeed this light is what we will use to detect hits within the calorimeters.

The emitted light doesn't always come due to de-excitations and similar, sometimes processes will create free high energy photons, or the light might just come from cherenkov radiation. Cherenkov radiation occurs due to the relative speed of light within various media, it is therefore possible for particles to move faster than light within this particular medium (though not faster than c, the speed of light in a vacuum) this will cause the emission of light, which is called cherenkov radiation. Cherenkov radiation is indeed used to detect certain particles in certain detectors, it can for example be used to detect atmospheric muons. Fortunately the setup for our detector makes the occurrences of cherenkov radiation, and indeed hits of atmospheric muons, rather negligible compared to the many hits by particle beams, and eventually the output of collisions within the center of the ALICE detector.

2.1.1 The electromagnetic interactions

The most well known energy-loss is that of the electromagnetic interaction, which is done by charged particles as the traverse the given medium. The energy of these travelling charged particles can excite the electrons in surrounding atoms, to the point that it can ionize the atoms, by exciting the electrons to the point that they can pass out of the coulomb field generated by the nuclei of the atoms in the surrounding medium. While not the method we use for FoCal-H, this can be used for detectors, as electric fields can be used to collect these free electrons, thus giving a method to detect particle interactions by collecting and counting these freed electrons.

These are not the only types of electromagnetic interactions, as mentioned above cherenkov radiation can occur, which is an electromagnetic interaction. For even higher energies Bremsstrahlung can also occur, in this case the charged particle travelling at high energy will be affected by the coulomb fields of the surrounding nuclei, and in the process of passing by them it will then begin to emit photons, reducing its energy in the process, thus the name Bremsstrahlung which roughly translates to braking radiation.

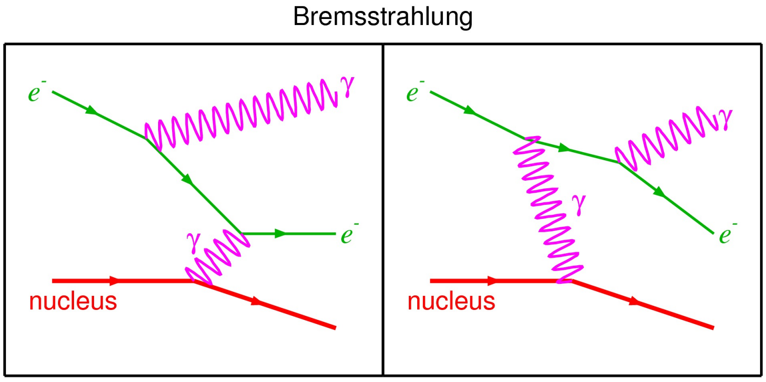


Figure 2: Feynmann diagram demonstrating the process of bremsstrahlung between an electron and a surrounding neutron

2.1.2 The photon EM interactions

The preceding electromagnetic interactions have mainly been focused on charged particles like electrons and positrons, and to a limited extent muons. However photons also interact electromagnetically and as the particle that we utilize for detection within the FoCal-H calorimeter it is prudent to briefly touch upon those as well.

The photoelectric effect

This interaction occurs primarily at low energies and is defined by the absorption of the photon into an electron which will then emit out of the atom, ionizing it in the process. This process primarily occurs at lower energies, as the cross section of this interaction depends on the energy of the photon to the order of: E^{-3}. As a consequence this process will not be particularly relevant for the FoCal calorimeters as we're working with energies in the orders of GeV.

Rayleigh scattering

This process is not particularly relevant either, essentially it occurs for low-energy photons, where they will scatter off of electrons, but not deposit any energy within these electrons, as a consequence this will alter the energy distribution, but not the actual energy deposited.

Compton scattering

Now, compton scattering is more interesting for our case, since it occurs at higher energies, within in the range of a few KeV to several MeV for the given photon, as a consequence it has a major contribution within several GeV events in detectors.

Compton scattering is the process in which the photon will scatter off of a nearby electron and in the process of this excite the electron into an unbound state. Because the photon isn't wholly absorbed by the surrounding atomic electron, the photon moves on, and will in many cases then interact in sequence with several electrons, each interaction lowering the energy of the photon, until the energy is low enough that photoelectric effect occurs instead, thus while photoelectric effect does occur, it is dominated by compton scattering in any higher energy process.

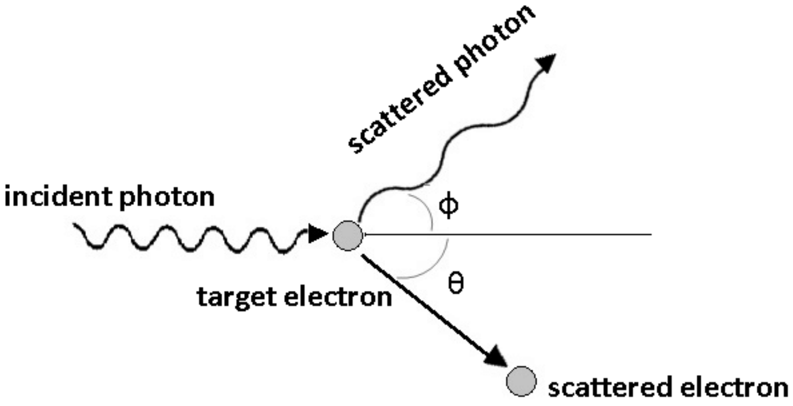


Figure 3: The geometry of a compton scattering process, demonstrating the process and the scattering angles of the photon and electron during the interaction.

2.1.3 The strong interaction

The specific part of FoCal that I have been assisting with is of course the Hadronic calorimeter (FoCal-H), therefore the interactions of hadrons within a calorimeter is of course of greater interest to this project.

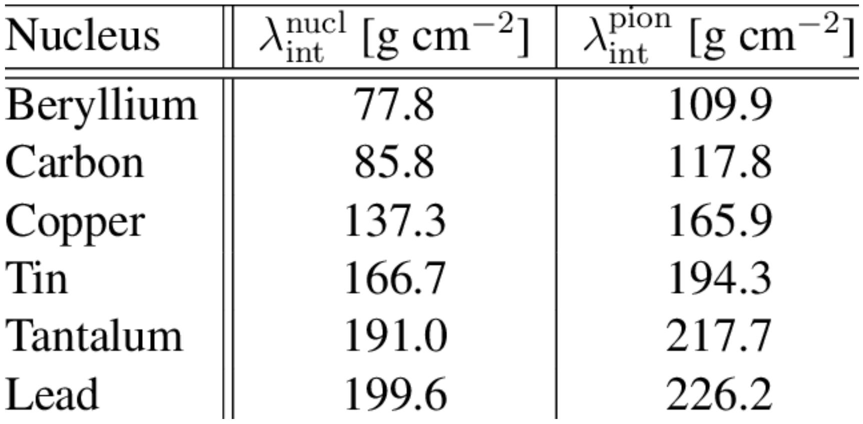
For hadronic particles the electromagnetic interaction is no longer the only interaction playing a significant role, as the strong interaction begins to influence the shower developments, due to the nature of the strong interaction they are necessarily more complicated than the electromagnetic, since a hadronic particle may both interact electromagnetically if it is charged, giving it some of the properties shown above, but at some point it will also most likely interact strongly with a particle within the medium. In this case a large variety of things can occur, the hadron may completely change its own properties as it could turn into multiple hadrons of lower energy, or into a different hadron, since the strong interaction is what holds the nuclei together it will naturally alter the nuclei if the constituent particles were to change during a strong interaction. It should be noted that the surrounding particles in the medium would of course also have the possibility change in such an interaction.

It should be noted, that for charge-less particles like neutrons this is the only type of interaction they can make, as they wont interact electromagnetically. As a consequence neutrons for instance will deposit their energy in a noticably different way in comparison to for instance protons.

When strong interactions occur it will often lead to the creation of new particles that will then constitute the parts of the hadronic shower. However many particles created as such will have the ability to interact electromagnetically, and as such may lose energy through ionization and other electromagnetic processes before they can reach another particle to have nuclear (strong) interaction with. The length that such a particle needs to travel on average in order to interact, is called the interaction length, and the amount of energy lost during this travel is dependant on the Z value of the given medium, for instance a hadron travelling through uranium will lose more energy to electromagnetic interaction, than the same hadron travelling through copper. Since the FoCal-H prototype is using copper in its construction, here are the relevant values for copper:

$$Z = 29 \parallel \lambda_{int} = 15.32cm \parallel \Delta E/\lambda_{int} = 193MeV$$

The $\Delta E/\lambda_{int}$ refers to the average energy lost due to electromagnetic interaction before the hadron will reach the nuclear interaction length. [11]



Nucleus	$\lambda_{\text{int}}^{\text{nucl}}$ [g cm^{-2}]	$\lambda_{\text{int}}^{\text{pion}}$ [g cm^{-2}]
Beryllium	77.8	109.9
Carbon	85.8	117.8
Copper	137.3	165.9
Tin	166.7	194.3
Tantalum	191.0	217.7
Lead	199.6	226.2

Figure 4: Table showing the nuclear and pion interaction lengths for various materials, including copper

Nuclear spallation

The most likely hadronic reaction to occur is that of nuclear spallation, in this process the high energy hadron begins to make quasi-free collisions within the nucleus itself of the collision particle. In so doing the hadron interacts directly with the partons, rather than with the nucleus as a whole. In this case the hadron can then transfer energy to the partons, leading to them transferring energy to other partons within the nucleus, and so on. Some partons may even get energy enough to cross the nuclear border and exit the nucleus wholly, others will instead settle back in the nucleus, leading to an overall excitement of the nucleus. The nucleus will then de-excite by evaporating particles away from it, mostly free nucleons, but it can at times even be alpha particles or heavier nucleon combinations. This will continue until the excitation energy of the nucleus will fall below the binding energy of its constituent nucleons. This can of course then lead to further collisions with other surrounding nuclei, leading to a shower development.

During a spallation interaction the nucleus will of course emit various nucleons during both the initial collision and the cascade of energy transfers occuring there, but also during the final state of evaporation. In this case, protons are mainly emitted during the initial cascading energy transfer, whereas the main particles being emitted during the evaporation stage is neutrons. Since neutrons of course carry no charge, the interact purely due to direct collisions and strong force interactions, and as a consequence their shower behavior is predictably different compared to protons which carry a charge, and as such will contribute electromagnetic interactions to the shower development.

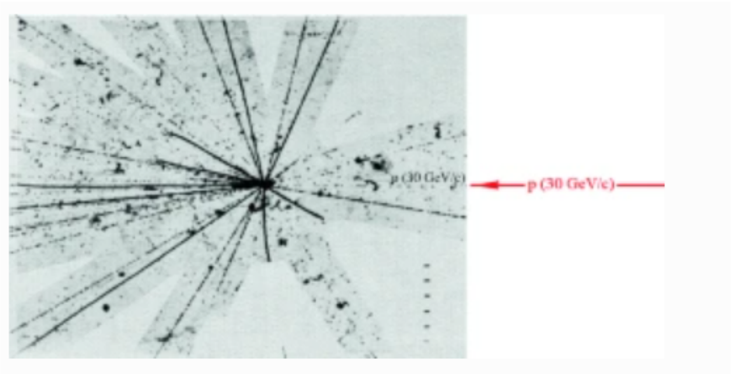


Figure 5: A proton nucleus interaction in a nuclear emmision stack.[12]

2.1.4 The Bjorken x

An important quantity for calorimetry, and in particular for the FoCal upgrades is the Bjorken x, which is referenced in the core motivations of the FoCal upgrades in the letter of intent [8].

The bjorken x can be defined in an electron-proton inelastic scattering process by and written out as[15]:

$$x = \frac{Q^2}{2Pq}$$

With Q^2 being the negative four momentum of the transfer virtual photon, P being the 4 momentum of the initial state proton, and q of course beong the 4 momentum of the transferred virtual photon.

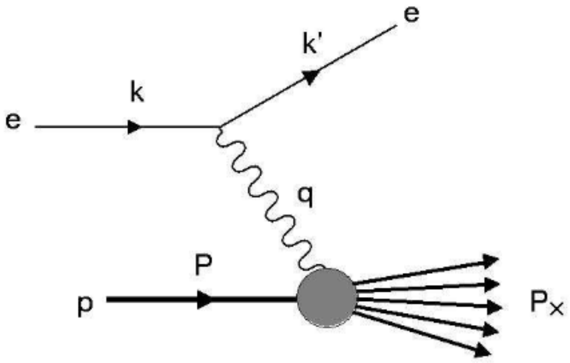


Figure 6: Illustration of an electron proton inelastic scattering process, demonstrating the various quantities used to calculate the bjorken x

Essentially the bjorken x corresponds to the momentum fragment carried by the individual partons within a hadron. As such the bjorken x is always within the rang of 0 to 1. This is all tied to the bjorken scaling phenomena, which is the scaling of the structure functions that are used to calculate cross sections, in this scaling the structure functions, normally determined to be dependant on Q^2 and the bjorken x, have been found to become almost completely independant of Q^2. This means that at a certain point the inelastic scattering process stops considering the total energy of this process, rather caring about this dimensionless quantity, the fraction of momentum within the hadron.

This became a significant step in determining that inelastic scatterings are scatterings with point like constituent particles[16]. Further, for high enough values of Q^2 the two structure function are not independant of one another, but are instead related through the *Callan-Gross relation*:

$$F_2(x) = 2xF_1(x)$$

Which is explained by the underlying process of an inelastic collision, is elastic collisions with the point-like particles within the hadron, namely the quarks[16].

Now, since the bjorken x represents the momentum fraction carried by the partons of a hadron, one might then consider how this momentum can be less than for instance, 0.3333.. inside of a proton or neutron as they have 3 main partons. However this can be explained by the sea quarks. While a proton has only 3 valence quarks, they do transfer gluons between them, and these gluons can split into quark-antiquark pairs, thus creating the sea quarks for the hadron. And in the case of low x, these sea quarks and gluons carry a very significant portion of the momentum within the hadron, this is due to the higher gluon density which occurs in these higher energy situations. This then becomes quite relevant when one discusses phenomena like nuclear shadowing, which will be covered in a later section.

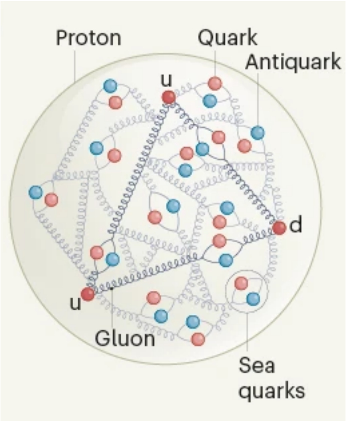


Figure 7: Illustration of quark structure of a proton, demonstrating the 3 valence quarks surrounded by the "sea" of gluons and quarks.

2.1.5 Shower developments

Showers were quickly described earlier in the descriptions of various interactions, but they are of vital importance to calorimetry overall and more specifically to the FoCal prototypes. As particles interact and deposit energy into the calorimeter particles, those hit particles may become excited or even unbound, allowing them to then deposit energy into other surrounding particles, and those then repeat this process until all the energy has been expended, or they leave the calorimeter. Alternatively collisions might result in the creation of whole new particles, which will then continue the shower development, again ending either when all energy has been spent, or they leave the calorimeter.'

Now a calorimeter seeks to completely capture the shower, this is necessary in order to accurately measure an event. This is because for us to be able to identify what the properties of the particles that entered the calorimeter, we want to know their momentum, however shower containment is necessary in order to get a complete view of the momentum, since a particle that leaves the calorimeter wont deposit its energy into the calorimeter, and as such will the calorimeter then under-report on the energy in this particular event.

Electromagnetic showers

Electromagnetic showers develop as a consequence of the electromagnetic interactions described earlier, when the incoming particles energize and ionize the surrounding particles, resulting in electrons and postrons being released which will then proceed to either excite and/or ionize even more particles or radiate photons which can then make more electron-postitron pairs until the energy of the initial event has been spent.

Broadly speaking an electromagnetic shower can occur through two main means: Photon induced or induced by charged particles. The main difference between these two cases are that since photons can travel further through matter they will, on average, deposit their energy further inside the calorimeter, and will the amount of energy deposited in a given part of the detector will fluctuate, which occurs due to the variation in *where* the shower begins. Meanwhile a charged particle induced shower occurs earlier and is broadly speaking more consistent, since charged particles will begin to deposit energy immediately, due to interacting with the coulomb field which also ensures the consistency of these showers[13].

Hadronic showers

Hadron showers are notably more varied and complex than purely electromagnetic showers, notably a hadron entering a calorimeter will also ionize the surroundings as it's in most cases also a charged particle, while it will lose energy in this process, if it has a high enough energy it will eventually encounter a particle with which it will interact strongly. This will then in most cases result in the creation of several new hadrons, in addition to leaving the struck nucleus in an excited state, resulting in it emitting out more hadrons and photons in order to reduce its energy further.

Hadronic particles will need to travel some distance in order to collide with a nucleus that it can interact strongly with, this is where the interaction length comes from, as it is defined as the average distance that a hadron will have to travel in order to interact strongly, notably pions and other similar small hadrons will on average have to travel further than larger hadrons, since their small size makes it less likely to collide with a nucleus, thus their interaction length is often listed seperately from the nuclear interaction length.

As the shower occurs it results in several new hadrons, these hadrons will then proceed to travel through the medium, in the process of this they will then lose energy by ionizing the surrounding atoms, if their energy is low enough this will just cause a shower similar to an electromagnetic shower, however if their energy is high enough these hadrons may interact strongly resulting in new hadrons, this continues until the energy becomes too low, after which the number of particles in the shower starts to decrease as the shower particles get absorbed, this continues until the whole shower has been absorbed or the remaining particles leave the calorimeter.

In difference from electromagnetic showers, a hadronic shower will always have its peak energy deposits in a similar region, showing very little difference there. The main energy deposits within a hadronic shower will almost always be at one interaction length.

2.2 The goals of FoCal

As outlined in the introduction, there are the main goals for the FoCal upgrade, a group of phenomena that it seeks to study in greater detail, and to properly appreciate the reasoning and motivation for the work on the FoCal-H prototypes as well as the work on the entire FoCal upgrade, an explanation of these goals is of particular importance. The gluon density of particles plays a particular role for the structure of the particle, however the overall structure of a particle can change dependant on its energy, how much of its momentum is carried by the individual partons, and the gluon density of a particle is one of these things that it can affect, all the energy of the particle has to be *somewhere* in it after all.

2.2.1 Nuclear Shadowing

The structure of nucleis and particles in general can be described through the structure functions, these functions in particular can be used to explain inelastic collisions between particles as rather being elastic collisions between the component partons, with the structure functions describing various magnetic and electromagnetic properties of the structure. The structure functions are essential for calculating the cross section, or can be acquired by measuring the cross section if you're working the other way. In fact, for inelastic scatterings the differential cross section can be defined as:

$$\frac{d^2\sigma}{dxdQ^2} = \frac{4\pi\alpha^2}{Q^4}[(1-y)\frac{F_2(x,Q^2)}{x} + y^2F_1(x,Q^2]$$

This equation describes the differential cross section during an inelastic collision of the type $ep \rightarrow eX$ [17] but demonstrates well the importance that these structure functions have for the cross sections of inelastic scattering, a process which is deeply relevant for strong interaction calorimetry, since direct collisions are the only case where the strong interaction plays a major role, due to its short range.

For larger composite particles, such as a nucleus, the structure function for the whole should be able to be written out as either a structure function of the whole system, or rather a structure function for the composite nucleons, one could naively imagine that there wouldn't be any difference, since the structure functions help describe an inelastic collision as merely an elastic collision with one of the nucleons. However by measuring cross sections at a wide range of energy transfers and values of the bjorken x, it has been observed that for low x interactions, the whole nuclear structure function $F_2^A(x,Q^2)$ will see a reduction in comparison with the corresponding free nucleon structure function:$F_2^N(x,Q^2)$. In fact, one can write out the ratio between these two functions and describe it as the following[3]:

$$R_{F_2}^A(x,Q^2) = \frac{F_2^A(x,Q^2)}{AF_2^N(x,Q^2)}$$

Here A is defined as the nuclear mass number, and it is experimentally shown that for varying bjorken x and fixed Q^2 this ratio will go through phases increasing or decreasing above or below 1, defined as[3]:

1. For $x \geq 0.8$ the ratio goes as: $R_{F_2}^A > 1$
2. For x between 0.3 and 0.8 $0.3 \leq x \leq 0.8$ the ratio goes below 1: $R_{F_2}^A < 1$
3. At the interval of $0.1 \leq x \leq 0.3$ the ratio briefly goes above 1 again: $R_{F_2}^A > 1$
4. Finally at the lowest values of x $x \leq 0.1$ it reaches thenuclear shadowing region as the ratio goes: $R_{F_2}^A < 1$ and stays below 1, continuing to decrease as the bjorken x goes toward zero.

This is illustrated here within this figure:

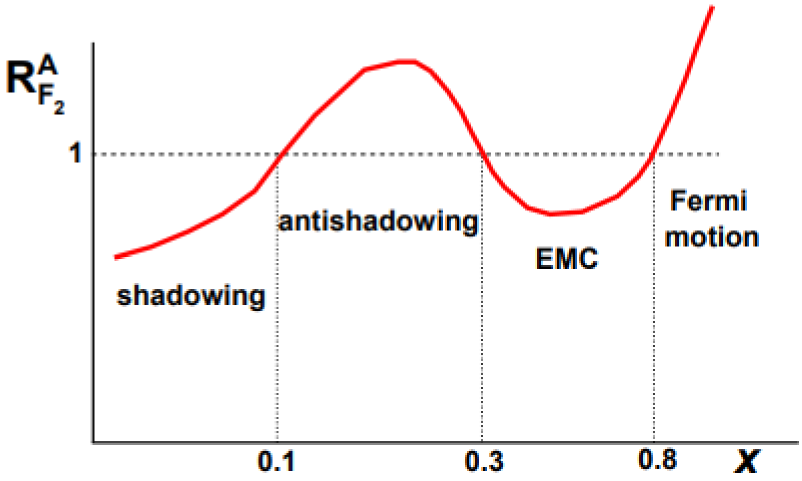


Figure 8: Illustration of the relationship between the nuclear ratio and the bjorken x, figure taken from [4]

The FoCal upgrade is, as stated, intended to study the low x regime and the shadowing effect, because of this the shadowing region is of greatest interest to this project
Now to start there are 3 main properties of nuclear shadowing that it fulfills:

1. Shadowing decreases with increasing bjorken x
2. Shadowing increases for increasing mass number A
3. Shadowing decreases with increasing values of Q^2

Now to describe some of these properties it will be prudent to look at a lab frame deep inelastic interaction wherein a virtual photon is exchanged. In this case there are a couple of options as to what can occur the first would be a direct transfer of virtual photon between the incoming particle and a constituent quark, alternatively the virtual photon could instead become a quark-antiquark pair that then proceeds to interact with the constituent quarks. This distinction can in part be used to explain the phenomena of nuclear shadowing.[10]

The ratio between the two options is described by:

$$|\frac{A_a}{A_b}| \sim |\frac{\Delta E_b}{\Delta E_a}| = \frac{Mx}{< p_q^2 >^{1/2}}(1 + \frac{\mu^2}{Q^2})$$

Where $< p_q^2 >$ is the average quark momentum inside of a nucleon, M is the nucleon mass, the A values are the amplitudes of the two different interaction methods, and μ is the invariant quark-antiquark pair mass.[10]

As can be observed in the above equation, the ratio between the two cases: (a) direct transfer of virtual photon and (b) the virtual photon turns into quark-antiquark pair that then subsequently interacts, is directly proportional to the bjorken x, and for low values of x the ratio becomes small and process (b) begins to dominate. This then leads to investigating the coherence length of interaction (b), which is defined as the inverse of the energy transfer ΔE_b and as such is given by[10]:

$$\lambda \sim \frac{1}{\Delta E_b} = \frac{2\nu}{\mu^2 + Q^2} \xrightarrow{\mu^2 \sim Q^2} \frac{1}{2xM}$$

For very small values of x this coherence length will exceed the average distance between nuclei, and as a consequence multiple interactions with nuclei will become coherent. This then gives an explanation for the nuclear shadowing effect, as it would then occur due to the destructive interference between multiple coherent interactions with nuclei. This also fits with the criteria mentioned earlier, as this shows that for large values of x this coherence length decreases, leading to the interactions becoming incoherent, which results in no interference between scatterings and as such no nuclear shadowing effect. Further it also showcases the dependence on Q^2 as the coherence lenght will likewise decrease as Q^2 increases.

In the case of what FoCal seeks to look at we also want to take the gluon density of the particle into account, for low x particles only the minority of the momentum within the particle is carried by the partons. In this case thus the gluon density and its impact begins to rise dramatically, this manifests as a growing amount of sea quarks in the particle, however for low x the main degrees of freedom are tied to the gluons and as such they have the main impact.

This means that the quark density of the particle will have a direct impact on the cross section of the interaction, and as such a direct impact on the shadowing effect that would occur, as the virtual photons' quark-antiquark pairs' interaction length will easily extend beyond the average distance between the valence and sea quarks.

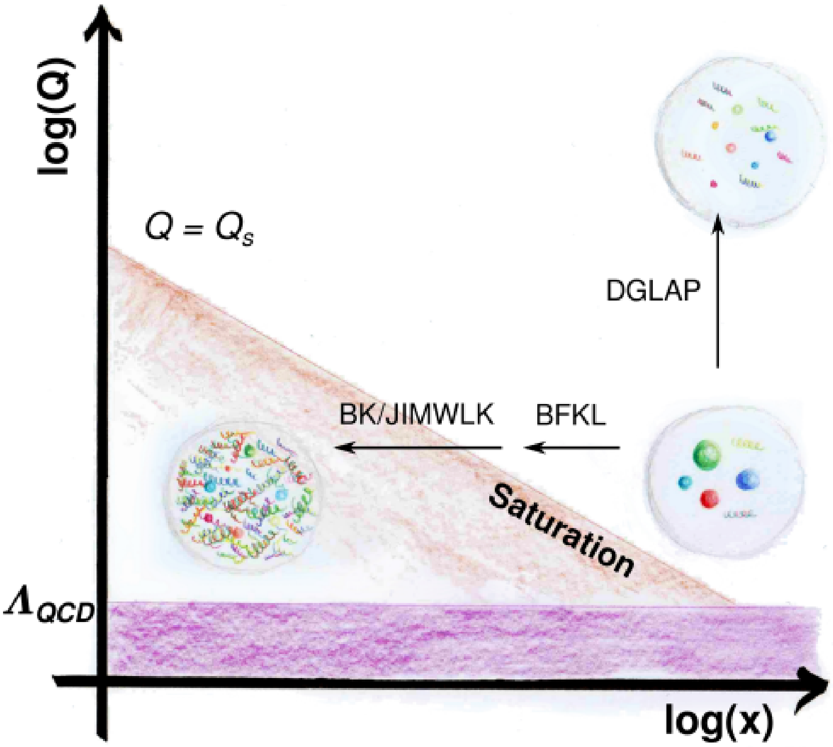


Figure 9: Illustration of QCD at various values of Q and x, demonstrating the growing gluon density for low x [9]

Thus from measuring the output of particle collisions and reconstructing the cross sections of these collisions, comparing them to theoretically calculated versions of them using different PDFs will allow us to gain a better understanding of the impact that the gluon density has on the nuclear shadowing effect.

2.2.2 Jet-quenching

Jet quenching is a phenomena that occurs in ultra high energy collisions of heavy ions, which is precisely the type of events that take place in the ALICE experiment. It is the phenomenon of jets having a markedly reduced energy compared to what would be expected, indeed occationally a two-jet pair can have one of them at a much reduced energy compared to the partner jet, which should have had equal energy in accordance with conservation of momentum. However due to interactions that can occur to the jets, this might not be the case.

This quenching occurs due to strong interactions by the jet with *Quark-Gluon plasma* which occurs in extremely high energy collisions with heavy ions. This medium mimics the properties of the universe in its first millionths of a second. Thus jets are quenched, and sometimes even wholly absorbed in this medium as they traverse it on their way out of the initial collision. This medium only maintains itself for a very short amount of time, as it will then cool and collect into hadrons.

The medium, Quark-Gluon plasma, is a plasma of quarks and gluons, ordinarily gluons and quarks are bound by colour confinement, which prevents them from existing on their own, as the colour of a state has to be 0. However in quark-gluon plasma the medium behaves somewhat like a plasma, hence the name, except instead of free electrons and ions, it is free quarks and gluons intermixing.

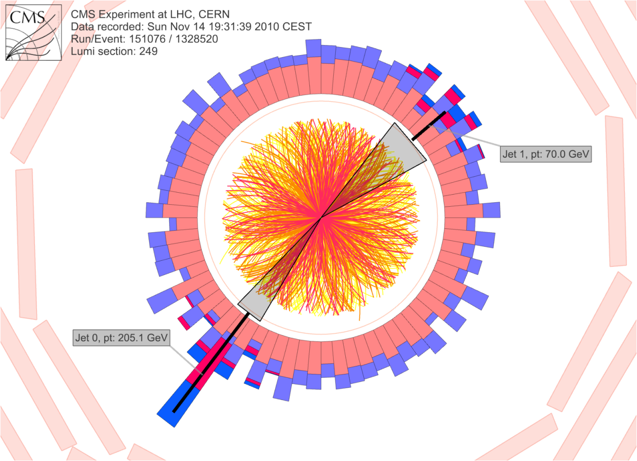

CMS
CMS Experiment at LHC, CERN
Data recorded: Sun Nov 14 19:31:39 2010 CEST
Run/Event: 151076 / 1328520
Lumi section: 249
Jet 1, pt: 70.0 GeV
Jet 0, pt: 205.1 GeV

Figure 10: Data from the CMS experiment at CERN showing two back to back jets demonstrating asymmetric energies, indicating jet quenching[2]

The jet quenching phenomena occurs due to a variety of interactions within the quark-gluon plasma of the high energy heavy ion collisions. The two main interactions by which this energy loss mechanism takes place through are: **Radiatiative energy losses** and **Collisional energy losses** [14].

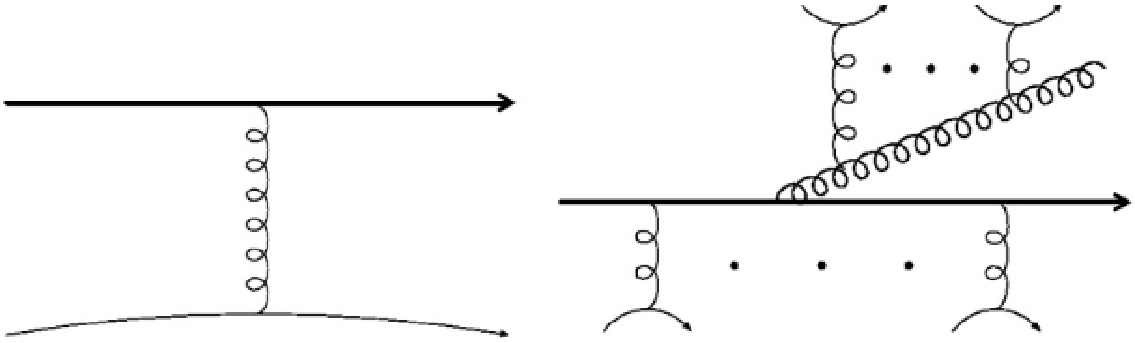


Figure 11: Illustration of Collisional energy loss (left) and radiative energy loss (rigth)

Radiative energy loss

The most significant contribution to jet quenching is the radiative energy loss, this occurs due to recurring rescatterings with the shower development inside of the Quark-Gluon plasma (henceforth referred to as QGP). Similarly to Nuclear shadowing, quantum coherence can affect these scatterings too, since if they take place close enough to one another (within the coherence length) then they have to treated as coherent, which will cause an effective suppression in the radiation spectrum.

As such radiative energy loss mainly manifests through the emission of gluons, specifically single or multiple gluons. This is done throu parton splitting and subsequent hadronization to maintain colour confinement.

Collisional energy loss

Collisional energy loss overall plays a smaller, but still significant part in the jet quenching effect, and are defined as elastic collisions between jet components during the jets. This is particularly important for heavy quarks where elastic collisions are considered the primary mechanism by which they will reduce their energy.

Overall jet quenching has a significant impact on high energy heavy ion collisions and through measuring jets and the quenching effect on them we can get information on the QGP that causes this jet quenching.

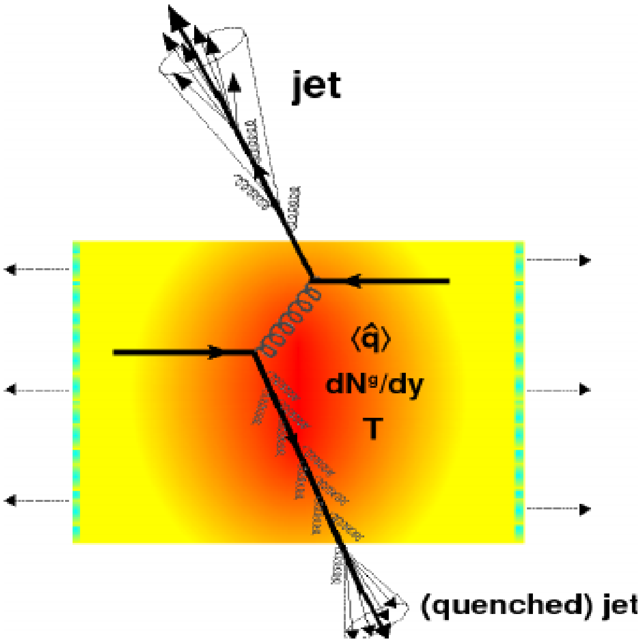


Figure 12: Figure showing the process of jet quenching from a back to back jet pair.

It is expected that the density of QGP is reduced at forward rapidities, while the density of quark jets would increase. This puts the FoCal Calorimeters in a prime position to measure jets at these forward rapidities allowing greater insight into the properties and distribution of QGP by measuring the forward jets, and the strength of the quenching phenomena on them.

2.2.3 Long range correlations

In p-Pp collisions a complication in the study of gluon density has been observed, a correlation between the Azimuthal angle over a large range in rapidity, this correlation manifests as a "double ridge" in the yield per trigger.

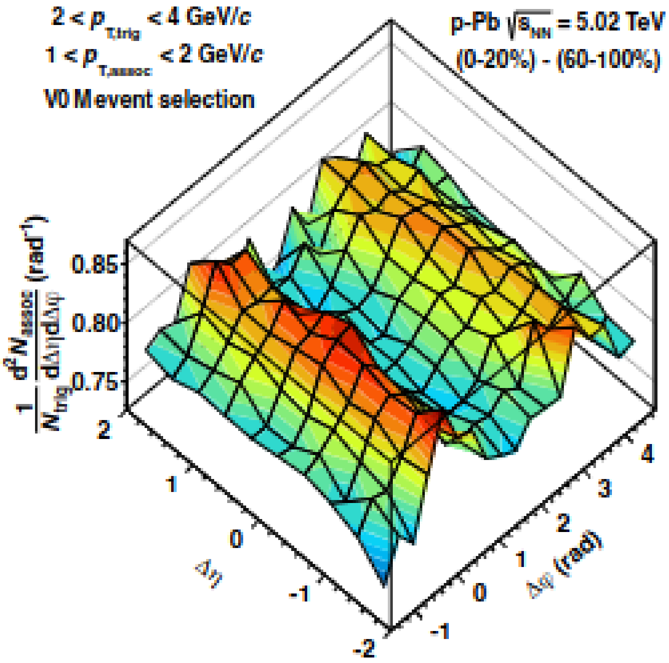


Figure 13: The double ridge, demonstrating the correlation between azimuthal angle and the range in rapidity[6]

The cause of this phenomena is as of yet unknown, however the large range of the correlation does seem to indicate that the phenomena originates in the early states of the collision, either the momentum distributions of the inititial state itself or the immediate states after scattering. This could imply collective motions could significantly impact the energy yield and hadron mass.

The FoCal upgrade seeks to study this through the measurement of these correlations of the Azimuthal angle at various rapidities and ranges, even able to test with various heavier mesons in order to check for mass dependancies in this phenomena.

3 The 2nd prototype, construction and the beamtests

3.1 The prototype

The second FoCal-H prototype is a sampling calorimeter constructed from tubes of copper with scintillating fiber straws inside the tubes meant to act as light conductors, ensuring that the light hit the silicon photo-multipliers (SiPMs) attached at the back of the calorimeter and connected to the readout system. The final FoCal-H calorimeter is intended to likewise be a sampling calorimeter. For FoCal-H the copper tubes act as the dense material to produce showers, and the scintillating fiber straws will be the main method of producing a signal within the calorimeter.

The physical setup of the second prototype of FoCal-H consists of 9 modules containing the copper tubes and optical fiber straws, the modules are each approx. 66 mm by 66 mm adding up to dimensions of 20 cm by 20 cm for the whole setup, as the modules are placed in a 3 by 3 setup.

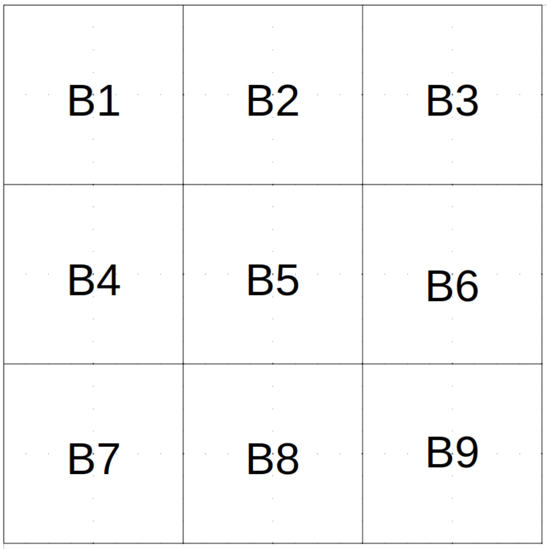


Figure 14: The layout of the 9 modules of the second prototype, collectively they add up to a larger square. The modules are each named from B1-B9.

In terms of the length of the calorimeter it has a length of 110 cm, so the total dimensions of the prototype is 20 cm by 20 cm by 110 cm.

The first beamtest for the prototype was the september 2022 PS beamtest, and as such gave a deadline for finishing the prototype, as described the protype consists of nine modules each consisting of copper tubes and fiber straws. To be exact each module consists of 668 tubes and fiber straws, each straw has to be inserted into the tubes and guided to the correct bundle at the end. '

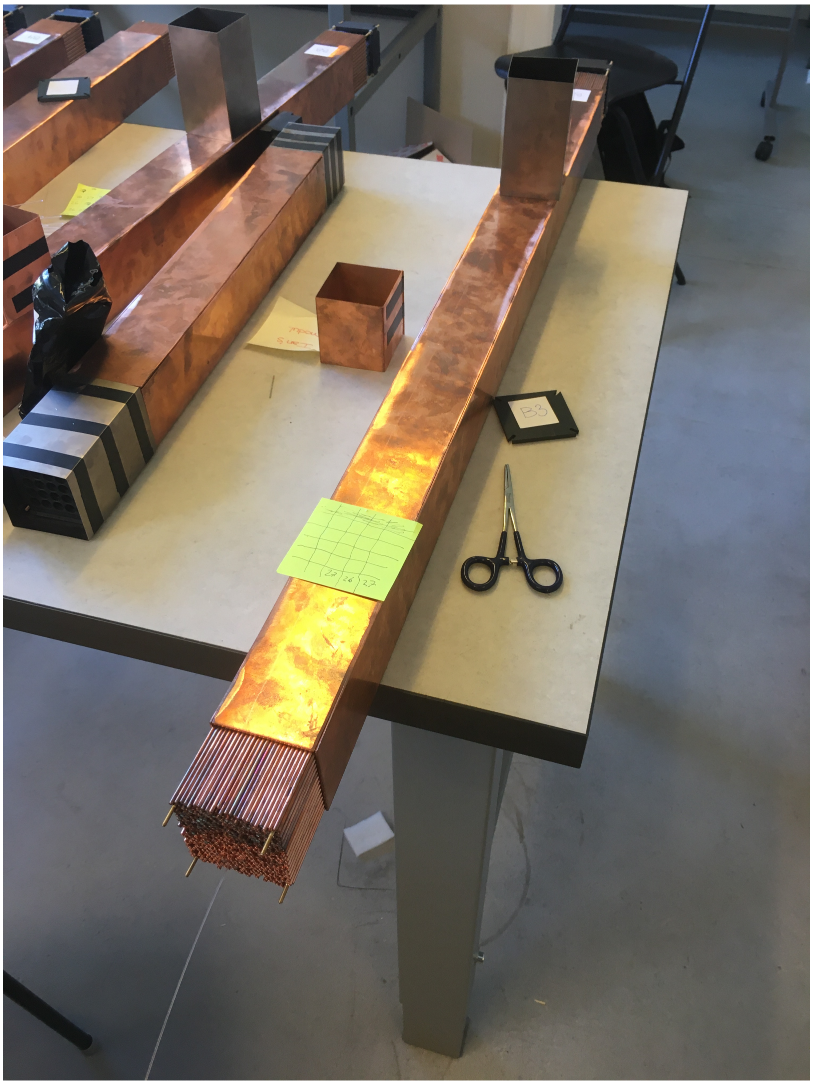
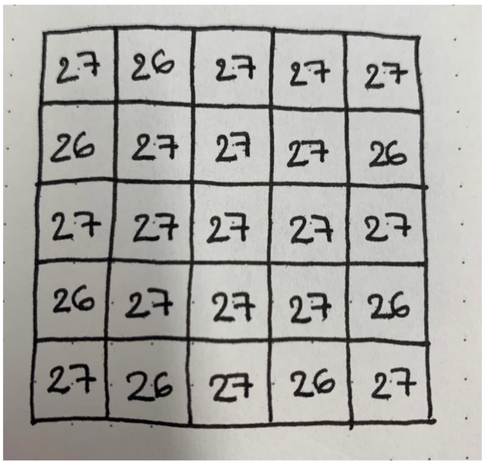


Figure 15: One of the 9 modules of the second prototype during the process of inserting the fiber straws

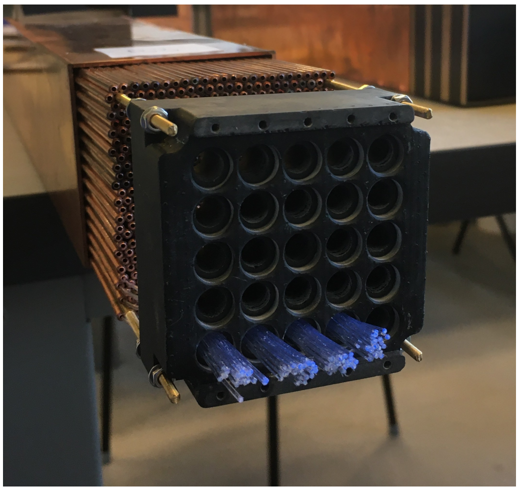
When I started my project in early-mid august, my first contribution was to assist with inserting the straws into the copper tubes of the modules, since this task needed as many hands as it could get, as this was a rather time consuming task since you could only really insert one straw at a time, and it could take several minutes to get one straw through properly in the more difficult cases.

As mentioned earlier the straws had to be guided int o specific bundles, this is due to the silicon photo-multipliers used to record hits within the detector, as it isn't practical to have 668 channels per module within the detector, we would instead bundle the various fiber straws into 25 bundles consisting of 26-27 straws. These bundles would then be the main attachment point of the photo-multipliers (SiPMs from here on out).



27	26	27	27	27
26	27	27	27	26
27	27	27	27	27
26	27	27	27	26
27	26	27	26	27

(a) The layout of the various bungles within a given module, image credt goes to Laura Marie Dufke.



(b) The actual module during assembly, with a couple of bundles filled out.

In order guide the straws into the correct bundles we had to use tools in order to gently draw them in the right direction, which became notably more difficult to do as each module neared completion as the straws became harder to distinguish and more closely and tightly bunched together. Of course care also had to be taken to not damage the straws, gloves had to be worn while touching them, since sweat and grease from skin contact would reduce their ability to send signals to the SiPMs, therefore care also had to be taken to not scratch or snap them in any way, as this would also reduce their effectiveness, if any of the above occured the straw would then have to be manually removed again and then subsequently replaced.

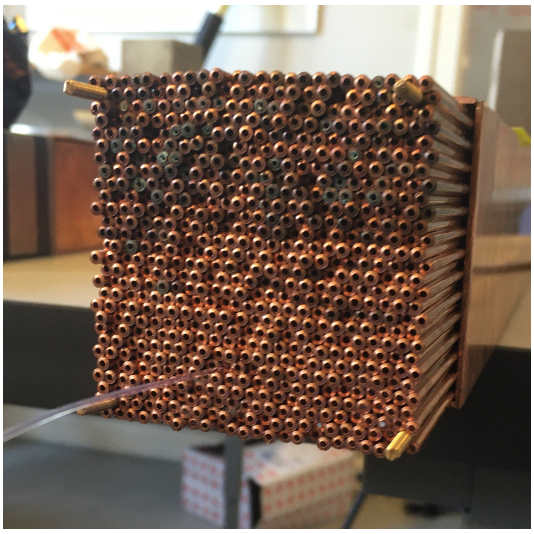


Figure 17: The front of single module, giving a close view on the makeup of the module, and the process of inserting each straw into the module.

Once each individual module had been assembled, and the straws guided through into the bundles, the assembly of the entire prototype could begin, with the 9 modules being stacked into the 3 by 3 shape and then locked in together, with preparations then being made for the transport of the prototype to CERN.

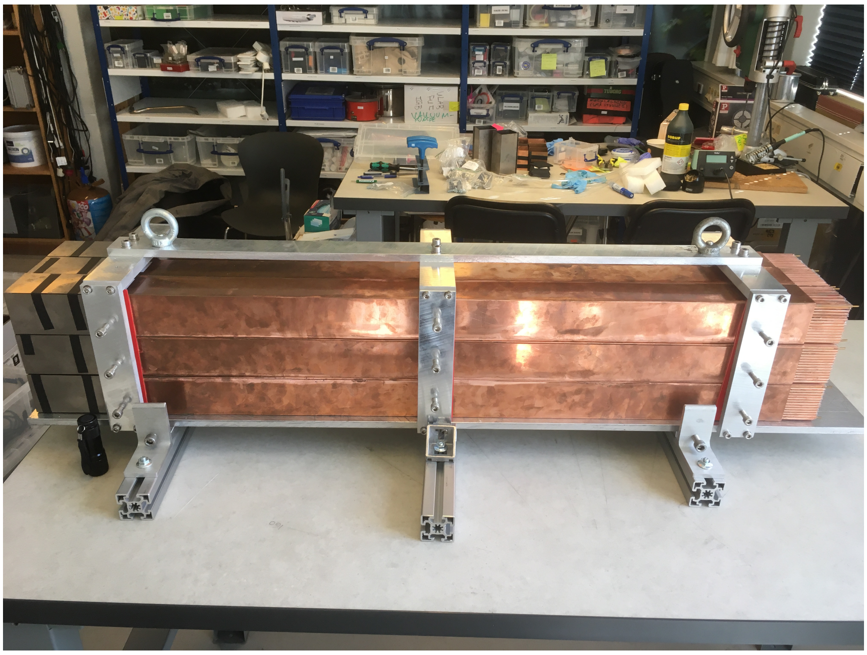
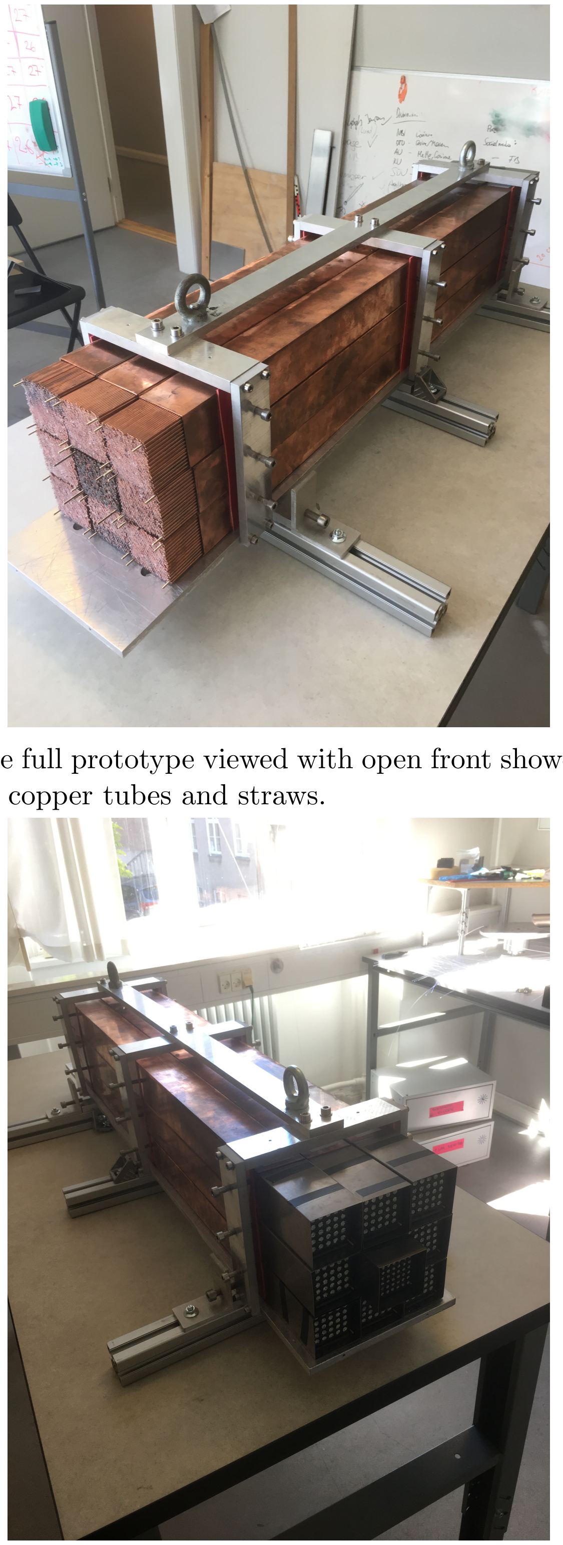


Figure 18: The full prototype, seen from the side within the assembly room where the straws were inserted into the modules.



(a) The full prototype viewed with open front showcasing all the copper tubes and straws.

(b) The full prototype as viewed from the back, giving a view of all the bundled straws.

3.2 The beamtests

As stated, once the prototype was finished preperations were underway for it to be transported to CERN, where the beamtest would take place, as such it and several of us from the poject group travelled to Geneva, Switzerland to attend the september beamtest at CERN. In the beamtest the prototype would be tested under circumstances closer to what the final calorimeter would be used for, as it would be placed behind the partnered electromagnetic calorimeter FoCal-E, this would have an impact upon the resulting energy distributions of events, due to the shower developments that could start within FoCal-E.

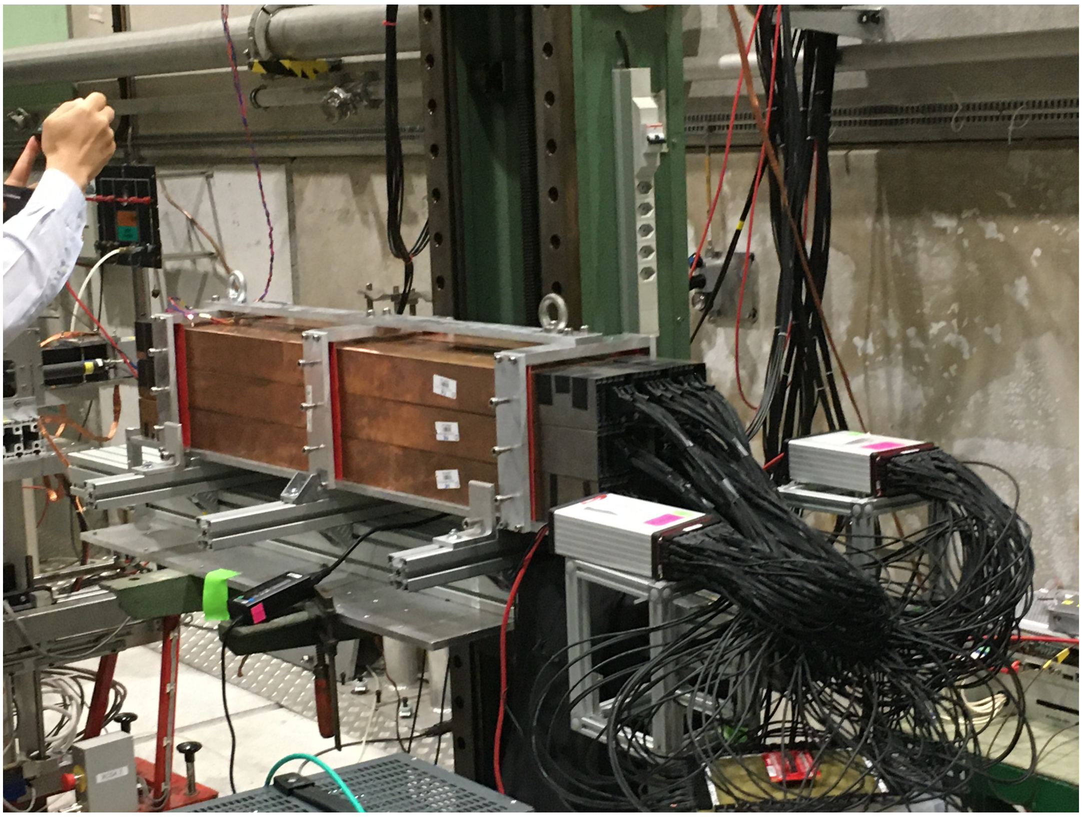


Figure 20: The setup of the prototype at the September beamtest, seen from an angle that shows off the data collecting CAEN boards and their connections to the SiPMs

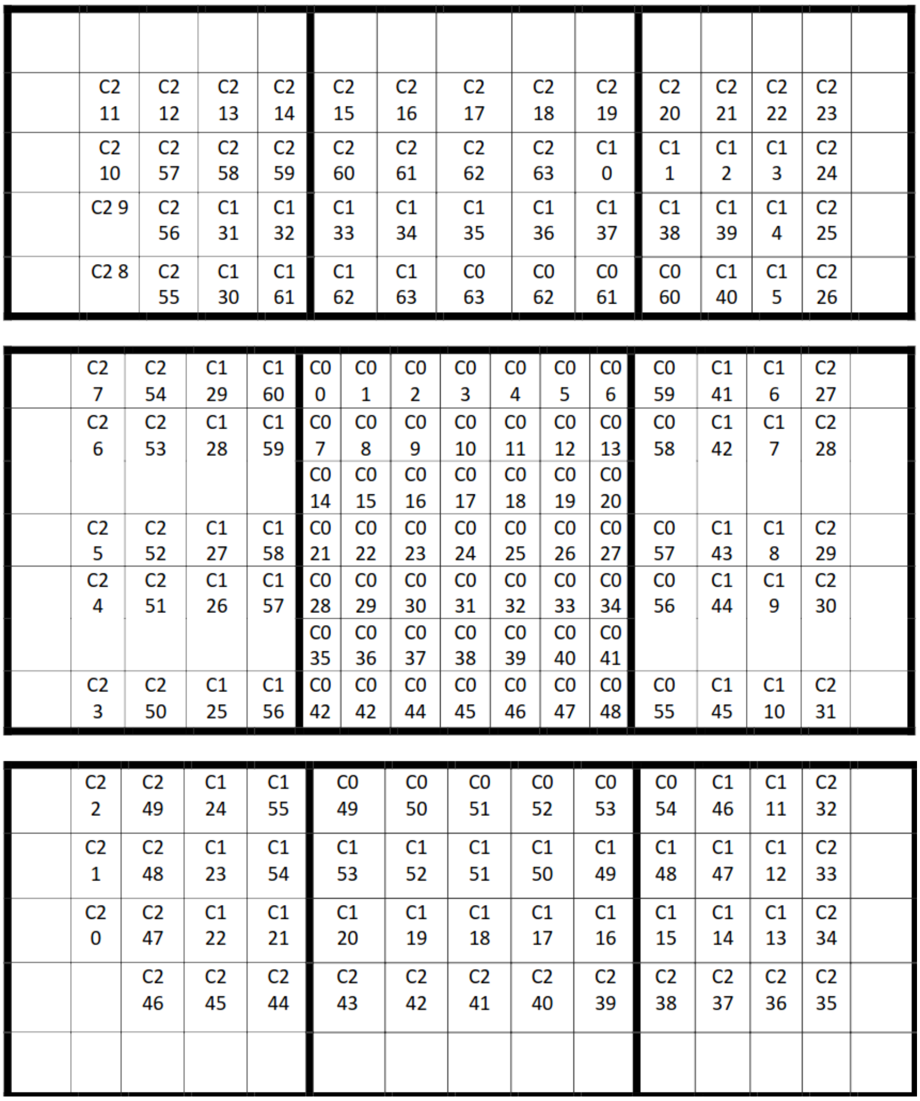
The beamtest itself involved the prototype being placed in a similar lineup to how the final calorimeter will be placed within ALICE, with FoCal-E in front of it, the two calorimeter prototypes were placed together in front of a beam that would send out particles of various types and at various energies that we had the ability to determine.

Since the particles that the final calorimeter is meant to capture wont be coming in at a perfect parallel angle with the detector, the beamtests were done at an angle, the September beamtest was done at an angle of 1 degree, whereas the November beamtest was done at a 2.085 degree angle, this does have a benefit for the calorimeter, since it means the beam particles are more likely to pass through multiple tubes, allowing for potentially more energylosses within the calorimeter, and minimizes the probability of a beam particle passing in between the tubes and just continuing through the calorimeter losing no energy to the calorimeter in the process.

For the setup during the beamtests the calorimeter was connected to CAEN boards and a VMM setup, both used for recording of the data, however the specific nature of these boards and the software associated with them is beyond the scope of this project. However the connection between the boards and the calorimeter is important to note, both to accurately report on the process of setting up the beamtest, but also to illustrate how the datatakin works in greater detail.

Essentially, the calorimeter is connected to the recording instruments through the SiPMs, the SiPMs are attached to parts of the various modules, and then for the most connected to the CAEN boards (The majority of the beamtests were done with CAEN, and the testbeam data used in this project was collected by the CAEN boards, as a consequence I will be assuming the usage of the CAEN boards for the rest of the project).

The calorimeter is connected to the CAEN boards through the SiPMs, part of the modules is connected to specific channeles within the CAEN boards, and as such results in each channel within the board correspond to hits inside of the calorimeter itself. The channels were placed in a spiral pattern around the detector, as we didn't have enough CAEN boards to fully cover the entire calorimeter, as a consequence the innermost channels of the prototype were prioritized, since that was where getting hits was most likely.



	C2 11	C2 12	C2 13	C2 14	C2 15	C2 16	C2 17	C2 18	C2 19	C2 20	C2 21	C2 22	C2 23	
	C2 10	C2 57	C2 58	C2 59	C2 60	C2 61	C2 62	C2 63	C1 0	C1 1	C1 2	C1 3	C2 24	
	C2 9	C2 56	C1 31	C1 32	C1 33	C1 34	C1 35	C1 36	C1 37	C1 38	C1 39	C1 4	C2 25	
	C2 8	C2 55	C1 30	C1 61	C1 62	C1 63	C0 63	C0 62	C0 61	C0 60	C1 40	C1 5	C2 26	

	C2 7	C2 54	C1 29	C1 60	C0 0	C0 1	C0 2	C0 3	C0 4	C0 5	C0 6	C0 59	C1 41	C1 6	C2 27	
	C2 6	C2 53	C1 28	C1 59	C0 7	C0 8	C0 9	C0 10	C0 11	C0 12	C0 13	C0 58	C1 42	C1 7	C2 28	
					C0 14	C0 15	C0 16	C0 17	C0 18	C0 19	C0 20					
	C2 5	C2 52	C1 27	C1 58	C0 21	C0 22	C0 23	C0 24	C0 25	C0 26	C0 27	C0 57	C1 43	C1 8	C2 29	
	C2 4	C2 51	C1 26	C1 57	C0 28	C0 29	C0 30	C0 31	C0 32	C0 33	C0 34	C0 56	C1 44	C1 9	C2 30	
					C0 35	C0 36	C0 37	C0 38	C0 39	C0 40	C0 41					
	C2 3	C2 50	C1 25	C1 56	C0 42	C0 42	C0 44	C0 45	C0 46	C0 47	C0 48	C0 55	C1 45	C1 10	C2 31	

	C2 2	C2 49	C1 24	C1 55	C0 49	C0 50	C0 51	C0 52	C0 53	C0 54	C1 46	C1 11	C2 32	
	C2 1	C2 48	C1 23	C1 54	C1 53	C1 52	C1 51	C1 50	C1 49	C1 48	C1 47	C1 12	C2 33	
	C2 0	C2 47	C1 22	C1 21	C1 20	C1 19	C1 18	C1 17	C1 16	C1 15	C1 14	C1 13	C2 34	
		C2 46	C2 45	C2 44	C2 43	C2 42	C2 41	C2 40	C2 39	C2 38	C2 37	C2 36	C2 35	

Figure 21: The mapping of the various bundles within the calorimeter to the various channels of the CAEN boards. Image credit to Laura Marie Dufke

Finally it should be noted that the SiPMs that are used for recording hits from the calorimeter are highly light sensitive, and that the optical fiber straws focuse ambient light extremely well, as a consequence the prototype has to be covered up during runs, as the amount of noise from ambient light would drown out any actual events, and would probably also crash the software.

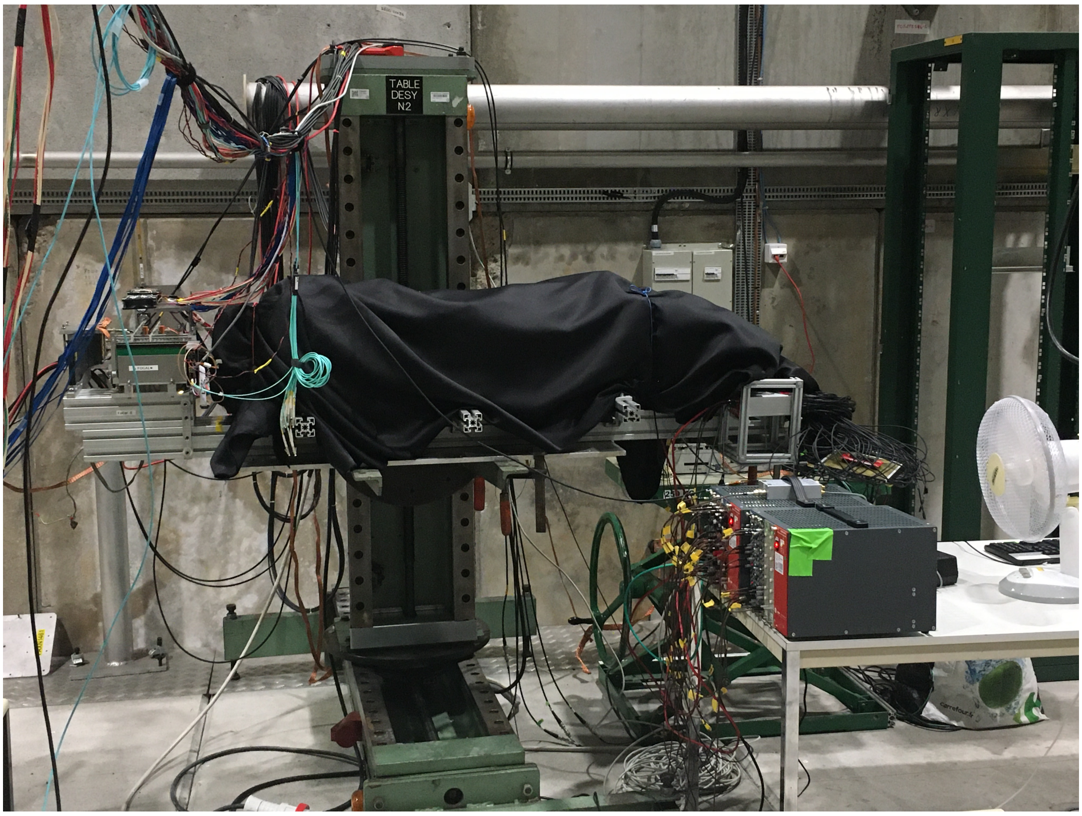


Figure 22: The setup of the prototype at the September beamtest, covered up in order remove as much noise as possible. In addition also showcasing the FoCal-E prototype in front of FoCal-H.

With this setup in place the main beamtests could take place, this would be over the course of several days, as the beamstests we run continously until they ended. As for the datataking itself, this would be done through a program used to record the data. Since the beamtests ran continously for their durations, this meant that data collection took place in shifts, we ran at 8 hour shifts: a day shift, evening shift and a night shift, during these shifts we would record many runs at various energies in collaboration with FoCal-E and in some cases without them.

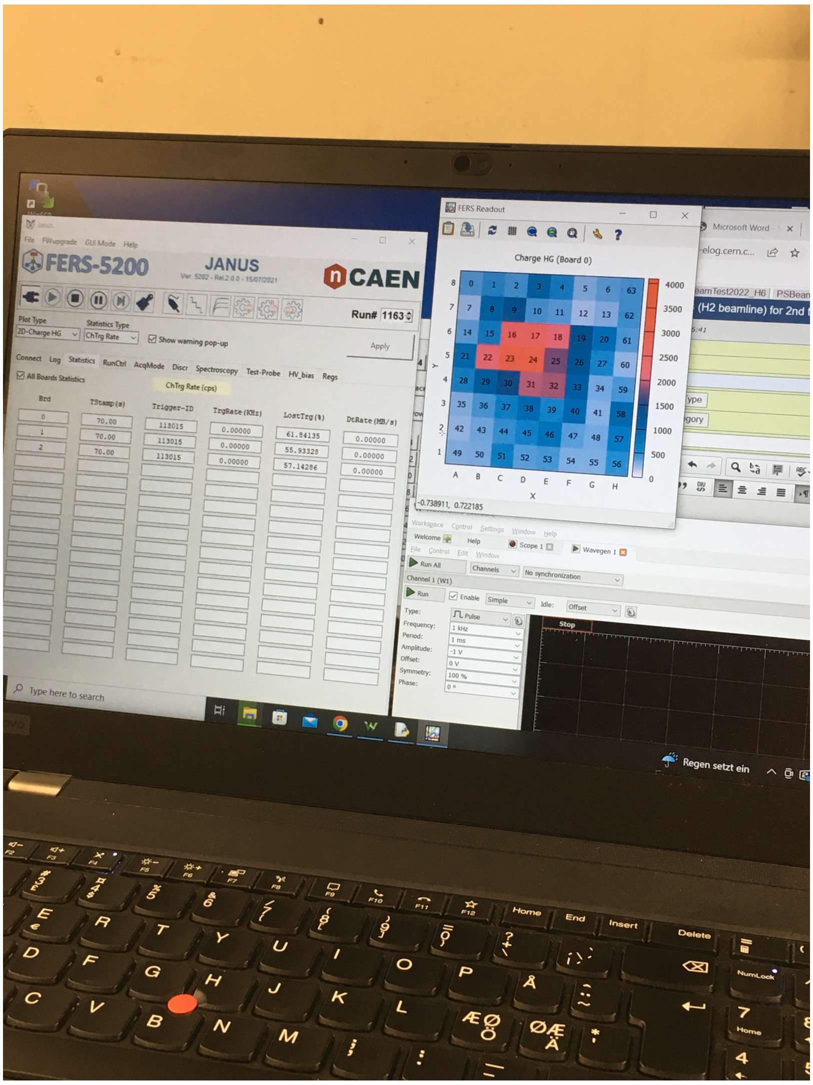


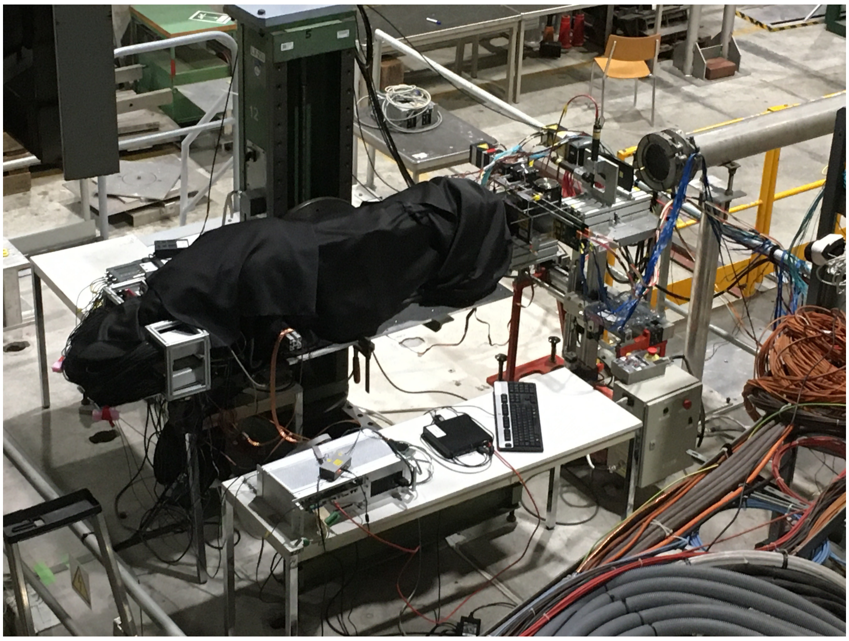
Figure 23: The actual process of data collecting was mainly done by starting and stopping runs through this interface, while also keeping an eye on the heatmap to make sure everything is working as intended.

The procedure for doing a run varied slightly depending on whether we had FoCal-E taking data as well, or even if they were there, versus if they were not. In the case of FoCal-E not being present, or not taking data we would start a run by starting the recording software, making sure to save the config file alongside it, and ensure its being saved in the correct folder for the given energy/angle/position of the overall setup. Then after starting the run we would turn off the busy signal.

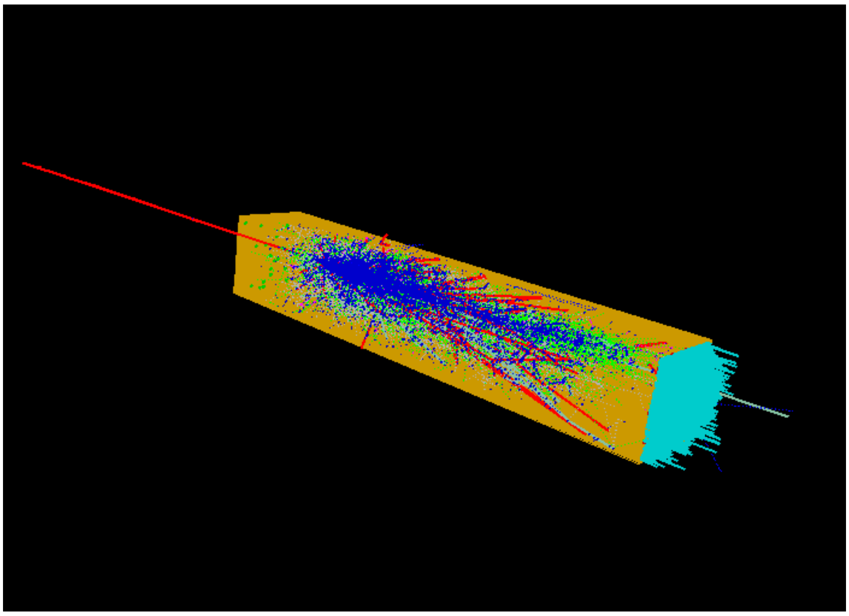
The busy signal This is a signal run through an oscilloscope attached to the overall setup of FoCal-H and FoCal-E prototypes, the main function of it is to allow us to create a busy signal which would allow us to ”cut off” the signal, this is done to prevent desync between the multiple CAEN boards, as during a spill, ie when the particles from the beam are being sent at the calorimeters, the different boards will desync during the spill itself, as their rate of counting out the hits during the spill will vary. However at the end of the spill they will have reached the same number, syncing up again, however if we ended a run right as a spin began for instance, it could lead to a major desync of the run, rendering the data useless. In order for the software to only record one event at a time it will send a busy signal to prevent simultaneous hits. By having another busy signal we could turn on or off at will, we were able to prevent this desync.

After turning off the busy signal the run would begin. Once the run ends we would just turn on the busy signal, and then end the run. In the case of takind data alongside FoCal-E, there was a couple steps added, we also had to coordinate our run starts and stops with them, essentially it became: ”turn on run + Focal-E turns on run $- >$ Turn off busy signal” to start a run and ”turn on busy signal$- >$ End run + Focal-E ends run” to end a run.

In between runs was the period in which we would adjust beam energy, or beam makeup, runs were done throughout the two beamtests with energies going from a few GeV to 350 GeV, specifically the SPS beamtests that I took part in was for energies goin from 60 GeV to 350 GeV. Specifically during the November beamtest we had the opportunity for the last day to take data without FoCal-E in front, which would become the main comparison point, and motivation between most of the simulations done in this project.



(a) The full setup for the November beamtest, showing both FoCal-H and FoCal-E as well as the beampipe in front.



(b) A simulated 350GeV event, meant to emulate some of the runs during the november beamtest.

4 The simulations

4.1 The testbeamsimulation simulation - geant4 version tests

For the initial part of this project the main simulation software was the "Testbeamsimulation" package, in addition to a custom ReadHCalData script written by Laura Marie Dufke to analyse the results of the simulation runs. Over the course of the project I worked with several different versions of this simulation program and several configurations of it, which will be covered in the final results section further in the report.

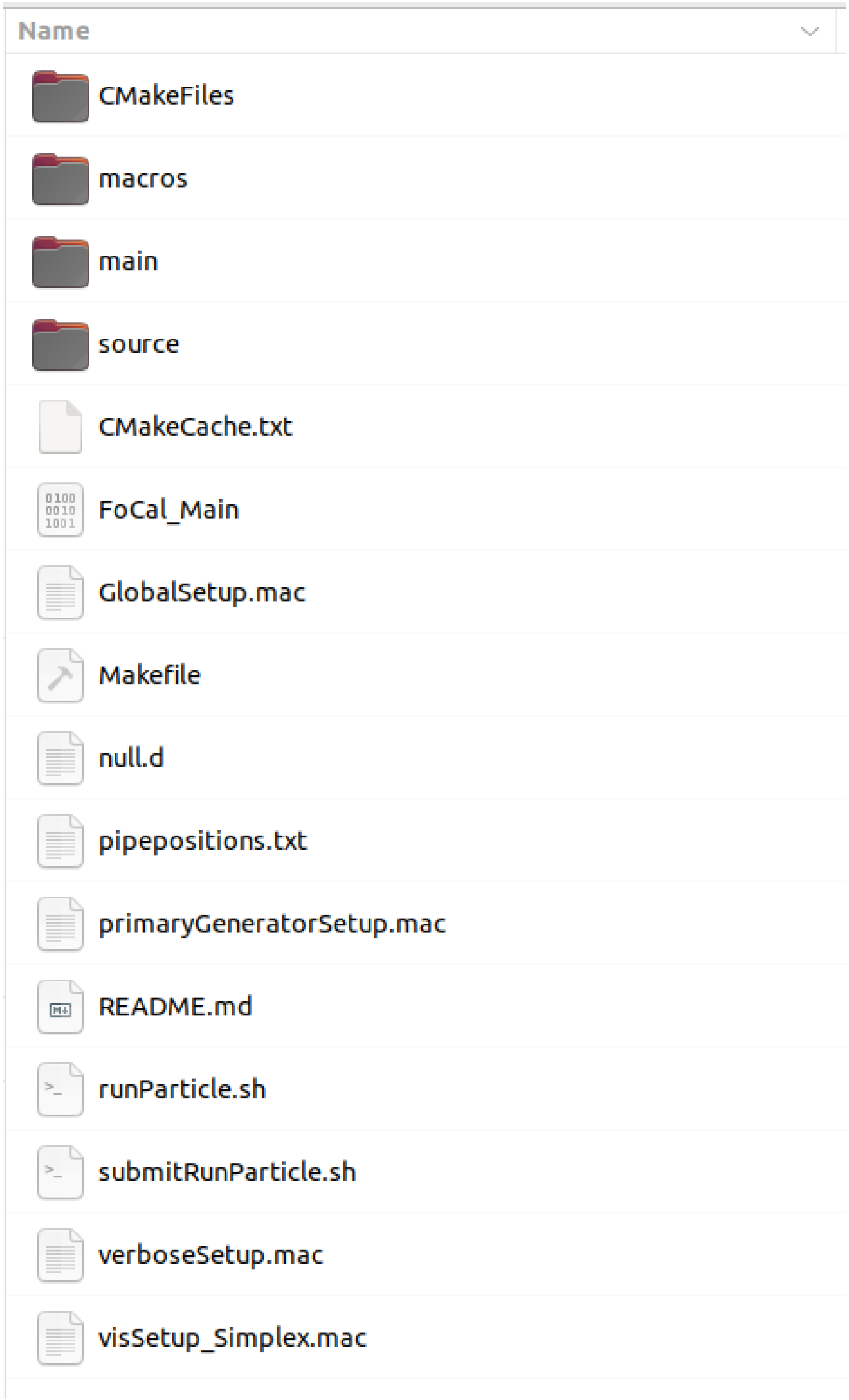
This simulation runs of geant4 exclusively, and uses specific versions of geant4 as well, though part of this project involved testing the differences between two versions of geant4 as well as two versions of the simulation package. Compared to the other simulation package described after the current one, this one is quite simple, but also quite effective.

This was the simulation package I used for initial comparisons with data from the beam tests as well as initial heatmaps and general tests of the setup, however I ran into a major issue when it came to the latter parts of the project, specifically the longitudinal energy loss distribution within the calorimeter, as that proved rather challenging to implement. The beginnings of a solution was being developed, however it proved much simpler to just swap to the The Focal_sim simulation package and just use that moving foward, since it could achieve my desired results baseline, as well as having solid built in analysis tools that rendered it quite effective.

Below I will detail the important parts of this simulation package as well as the tests betweeen the various versions of geant4 and update versions of the simulation package itself, I will also go over the initial steps I took to attempt the implementation of the longitudinal energy loss distribution before swapping to the other simulation package.

4.1.1 The contents of the main folder

To start with, an overview of the simulation package is in order:



Name

CMakeFiles

macros

main

source

CMakeCache.txt

FoCal_Main

GlobalSetup.mac

Makefile

null.d

pipepositions.txt

primaryGeneratorSetup.mac

README.md

runParticle.sh

submitRunParticle.sh

verboseSetup.mac

visSetup_Simplex.mac

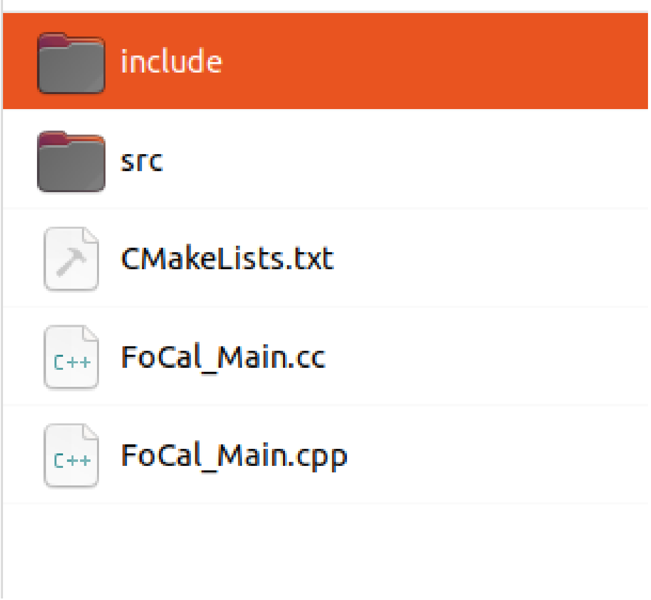
The main files of relevance to the running of this simulation would be the contents of the **source** folder and **FoCal_Main** and the output file.

Source is the main folder which contains the files that determine the setup of the simulation from whether it should have a GUI illustrating the hits or not, to what energies it should be running at, or whether FoCal-E would be in front. In essense this is where the bulk of the process of editing the parameters of the simulation take place.

FoCal_main is the main file for running the program, and is used to run the program.

GlobalSetup.mac is another important file for this simulation, it is specificall used to set the random seed for the simulation, as well as determine the particle type for the beam, as well as the energy for the beam, and is one of the steps in applying the GUI if that is what you want.

If we then look closer at the source folder of the simulation we can find two additional folders as well as some files:



include

src

CMakeLists.txt

FoCal_Main.cc

FoCal_Main.cpp

Taking a closer look at these:

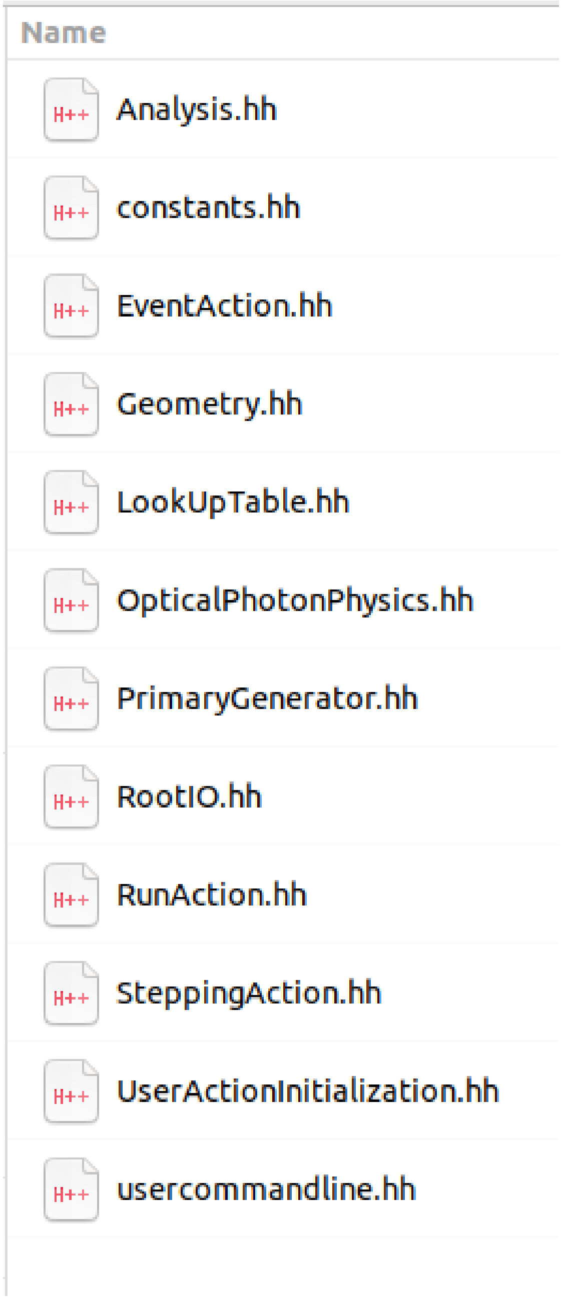
include is the main folder containing all the files that build the simulation, from the geometry to the steps taking during an event. Specifically this contains all the .hh files for the setup. It is specifically these .hh files which are used to alter the setup of the simulation if one needs to make changes, though any changes needs to be followed by cmaking the package again with the new changes.

src is much like the **include** folder in that it is the main setup of the simulation that is located here, however whereas **include** contained all the .hh files, **src** contains all the .cc files.

FoCal_Main.cc is the editable main file and can be altered to turn various features on or off, specifically is edited as part of the process of turning on the GUI.

CMakeLists.txt is used in the process of building the simulation through cmake, make and make install commands.

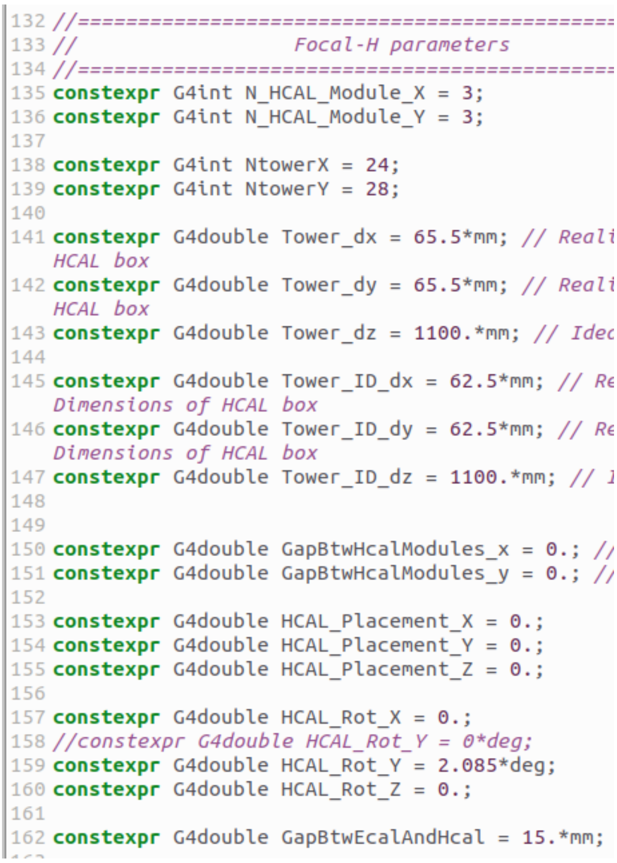
To go into further detail we take a closer look at the **include** folder, since the **src** folder looks nigh identical, save for the .cc files instead of .hh I will skip over that folder.



Name
Analysis.hh
constants.hh
EventAction.hh
Geometry.hh
LookUpTable.hh
OpticalPhotonPhysics.hh
PrimaryGenerator.hh
RootIO.hh
RunAction.hh
SteppingAction.hh
UserActionInitialization.hh
usercommandline.hh

Analysis.hh links the constants of defined in the **constants.hh** file with the root structure used in the simulation.

constants.hh is the main file for determining the geometry and physical setup of the simulated beamtest, in here you can setup the version geant4 you're using, the angle that the beam should come in at, rotation of FoCal-H, whether FoCal-E is in front of FoCal-H or not and more.



```
//==============================================
//                     Focal-H parameters
//==============================================
constexpr G4int N_HCAL_Module_X = 3;
constexpr G4int N_HCAL_Module_Y = 3;

constexpr G4int NtowerX = 24;
constexpr G4int NtowerY = 28;

constexpr G4double Tower_dx = 65.5*mm; // Reali
HCAL box
constexpr G4double Tower_dy = 65.5*mm; // Reali
HCAL box
constexpr G4double Tower_dz = 1100.*mm; // Ide

constexpr G4double Tower_ID_dx = 62.5*mm; // Re
Dimensions of HCAL box
constexpr G4double Tower_ID_dy = 62.5*mm; // Re
Dimensions of HCAL box
constexpr G4double Tower_ID_dz = 1100.*mm; // I


constexpr G4double GapBtwHcalModules_x = 0.; //
constexpr G4double GapBtwHcalModules_y = 0.; //

constexpr G4double HCAL_Placement_X = 0.;
constexpr G4double HCAL_Placement_Y = 0.;
constexpr G4double HCAL_Placement_Z = 0.;

constexpr G4double HCAL_Rot_X = 0.;
//constexpr G4double HCAL_Rot_Y = 0*deg;
constexpr G4double HCAL_Rot_Y = 2.085*deg;
constexpr G4double HCAL_Rot_Z = 0.;

constexpr G4double GapBtwEcalAndHcal = 15.*mm;
```

As can be observed in this image, the **constants.hh** file includes the methods to change the entire setup, here showcasing specifically the setup of FoCal-H, defining the number of modules in each direction to get the 3x3 structure of the prototype, as well as the size of each module and the placement of the prototype in the ”world”. Finally it even allows you to change the rotation of the FoCal-H prototype, to more closely match the results of the beam tests.

EvenAction.hh is used to define an event and get the results from said event by summing up the energies registered in both FoCal-E pads and pixels (if they are enabled) and FoCal-H modules.

Geometry.hh determines the volume and materials of the calorimeters according to what has been defined in other files.

LookUpTable.hh and **OpticalPhotonPhysics.hh** are not particularly interesting for this project. **Lookuptable.hh** seems to call upon the number of layers defined for the calorimeters, and **OpticalPhotonPhysics.hh** is used specifically if optical photons are enabled for the simulation.

PrimaryGenerator calls upon the particle gun and is used to generate the events that the simulation runs.

RootIO.hh is used to define various parameters, and **RunAction.hh** is mainly focused on the start and end of a run.

SteppingAction.hh is the main file wherein the individual hits of the events are determined, and are where the energy loss of these hits are initially calculated, before they're registered by other files. This is also the primary file that I worked in for the purpose of modifying the Testbeam simulation package to log the longitudinal energy loss distribution.

4.1.2 Running the Testbeam simulation package

Upon determining the energy for this run, the number of hits as well as the configuration of the setup. You put in a new random seed in the **GlobalSetup.mac** file, after which you can now begin the run.

Starting the simulation is very simple, all you do is: While in the main folder for the simulation, simply type

```
./FoCal_Main
```

Doing this will start the simulation for the specificed setup in **constants.hh** and beam energy, particle type and number of hits defined within **GlobalSetup.mac**. After the simulation has run its course, it will give a file called

```
hits.root
```

This is the output file from the simulation software, and will need to be moved or renamed before running the simulation again, as it will simply overwrite the **hits.root** file.

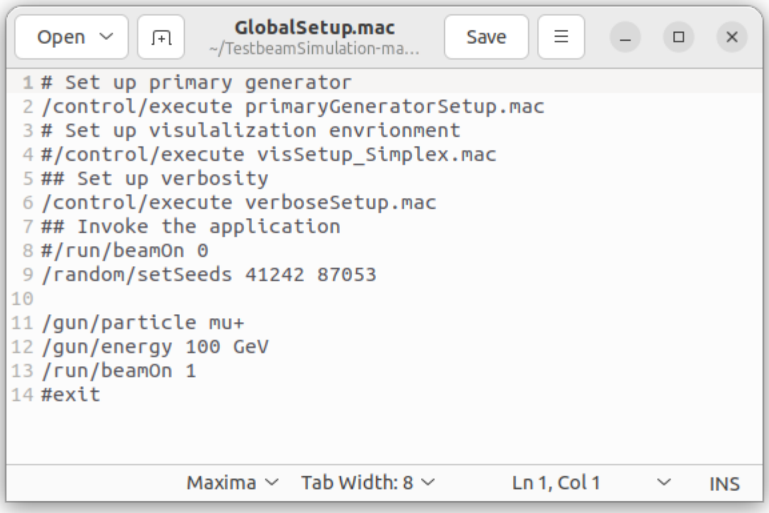
After this process it is then probably smart to move the renamed **hits.root** file into the folder in which the **ReadHCalData.cc** file is located. In the **ReadHCalData.cc** you then make sure to adjust it to target the correct filename of your simulation output file, as well as specify what you want the treated file to be called, most likely something similar to the raw output file in order to make organization easier.

Upon doing this you are then left with a treated output file where you can view the number of events at various energies and the transversive heatmap of the energy loss distribution within the calorimeter. These results are covered in the results section of the report further down.

4.1.3 Visualizing the setup

In order to visualize the setup within the Testbeam Simulation package it will be necessary to make a few changes to two files with in the folders.

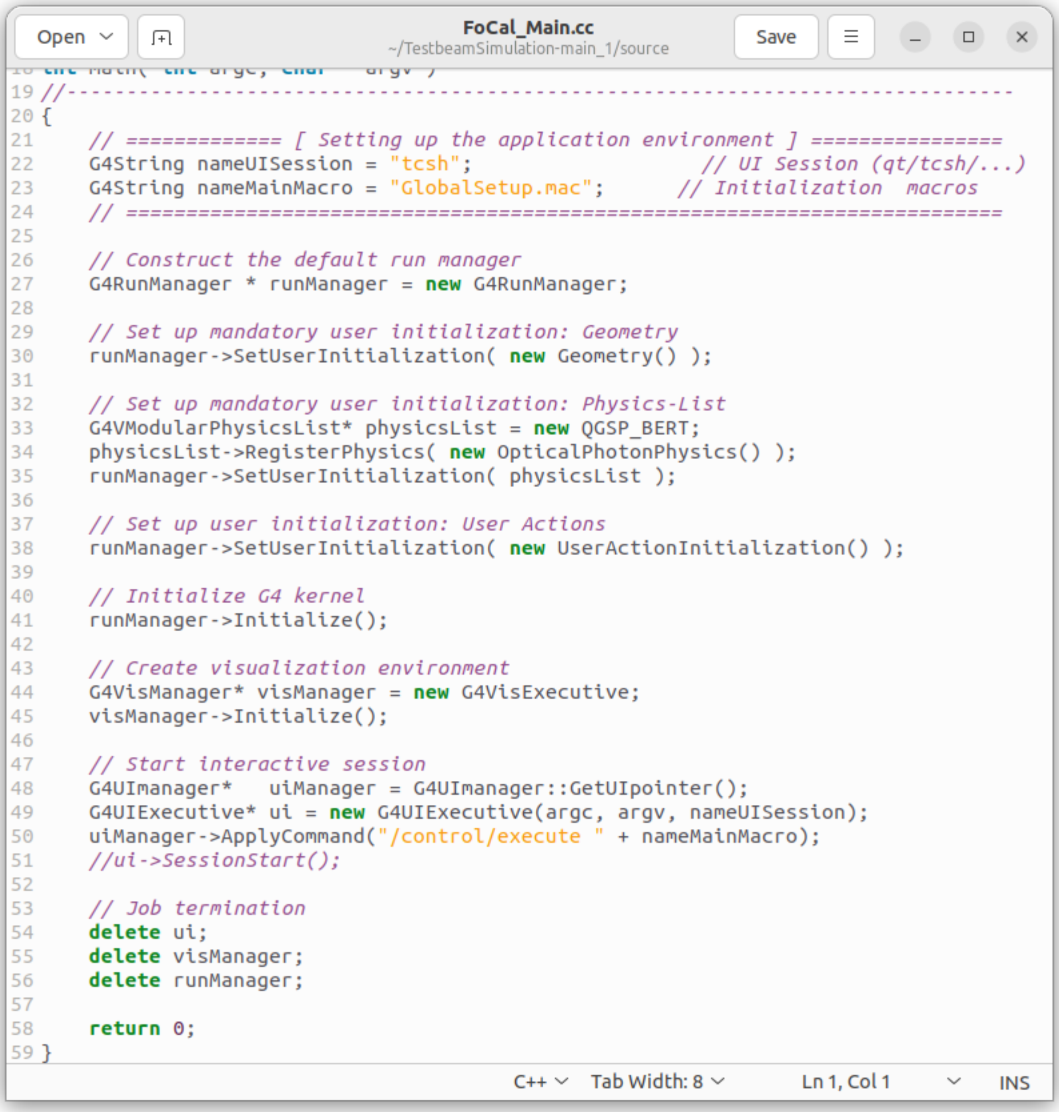
First, within the **GlobalSetup.mac** file:



```
 1 # Set up primary generator
 2 /control/execute primaryGeneratorSetup.mac
 3 # Set up visulalization envrionment
 4 #/control/execute visSetup_Simplex.mac
 5 ## Set up verbosity
 6 /control/execute verboseSetup.mac
 7 ## Invoke the application
 8 #/run/beamOn 0
 9 /random/setSeeds 41242 87053
10 
11 /gun/particle mu+
12 /gun/energy 100 GeV
13 /run/beamOn 1
14 #exit
```

You uncomment line 4: "/control/execute visSetup_Simplex.mac"
It is also recommended to make sure that line 13 says: /run/beamOn 1

Then enter the **FoCal_Main.cc** file within the **source** folder:

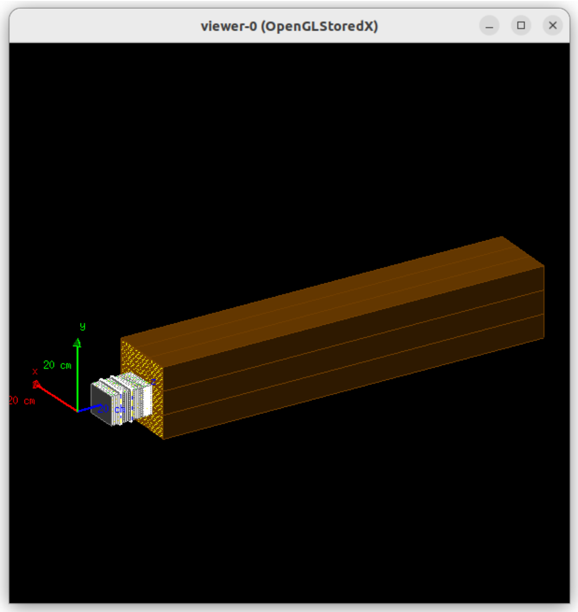


```
18 int main( int argc, char** argv )
19 //-------------------------------------------------------------------------------
20 {
21     // ============= [ Setting up the application environment ] ================
22     G4String nameUISession = "tcsh";                    // UI Session (qt/tcsh/...)
23     G4String nameMainMacro = "GlobalSetup.mac";       // Initialization  macros
24     // ======================================================================
25 
26     // Construct the default run manager
27     G4RunManager * runManager = new G4RunManager;
28 
29     // Set up mandatory user initialization: Geometry
30     runManager->SetUserInitialization( new Geometry() );
31 
32     // Set up mandatory user initialization: Physics-List
33     G4VModularPhysicsList* physicsList = new QGSP_BERT;
34     physicsList->RegisterPhysics( new OpticalPhotonPhysics() );
35     runManager->SetUserInitialization( physicsList );
36 
37     // Set up user initialization: User Actions
38     runManager->SetUserInitialization( new UserActionInitialization() );
39 
40     // Initialize G4 kernel
41     runManager->Initialize();
42 
43     // Create visualization environment
44     G4VisManager* visManager = new G4VisExecutive;
45     visManager->Initialize();
46 
47     // Start interactive session
48     G4UImanager*   uiManager = G4UImanager::GetUIpointer();
49     G4UIExecutive* ui = new G4UIExecutive(argc, argv, nameUISession);
50     uiManager->ApplyCommand("/control/execute " + nameMainMacro);
51     //ui->SessionStart();
52 
53     // Job termination
54     delete ui;
55     delete visManager;
56     delete runManager;
57 
58     return 0;
59 }
```

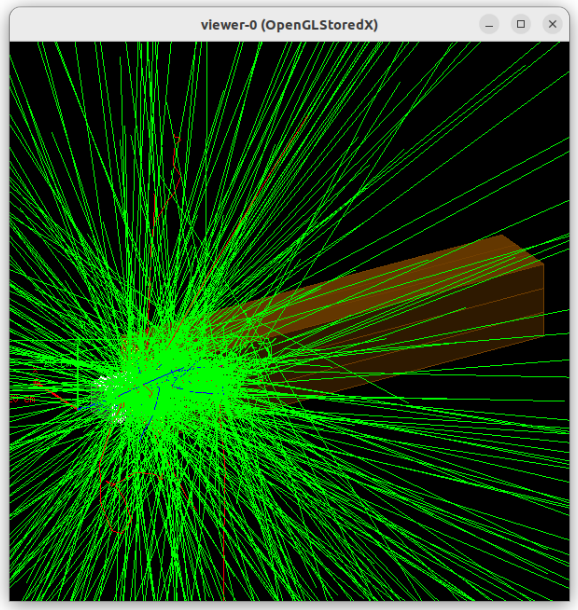
Start by changing line 22 to "G4String nameUISession = "qt";" instead of "G4String nameUISession = "tcsh";"

Then you simply uncomment line 51 "ui− > SessionStart();"

Finally you can run the simulation to get a visualization of the setup:



(a) The baseline setup of FoCal-E and FoCal-H visualized through the Testbeam Simulation package.



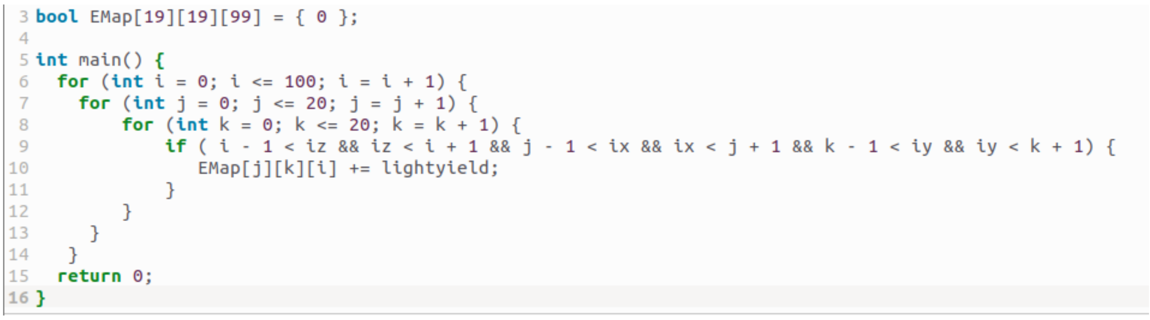
(b) A 20 GeV event visualized in the same setup through the Testbeam Simulation package.

4.1.4 The longitudinal problem

Now, the Testbeam Simulation package worked great initially, to get energy distributions of the total events, the transverse heatmaps, for initial comparisons with the recorded data from the actual test beam runs it worked great. However a problem would soon emerge, the longitudinal distributions. While the simulation recorded the distribution and location of the energy deposits in the transverse plane, it didn't record them in longitudinal plane, thus when I sought to get this distribution I ran into this problem.

The core part is, that the simulation package *does* actually calculate these energy losses over the course of a given event, the issue is that it doesn't save this information, instead just summing out all the energy losses instead. In addition it wouldn't be practical for it to register and log the exact Z positions per hit, as that would create an enormous amount of different coordinate points. It would be much simpler to instead just have it register and log if the hit was within a specific range in Z and then just save all those energy deposits to this range instead.

While the first problem of the simulation software saving this information was tough to solve the second problem was comparatively simple. It is a simple case of making a 3 dimensional array of a desired size then implement certain if statements to check the location of the hit in the calorimeter, an example code was created to do this:

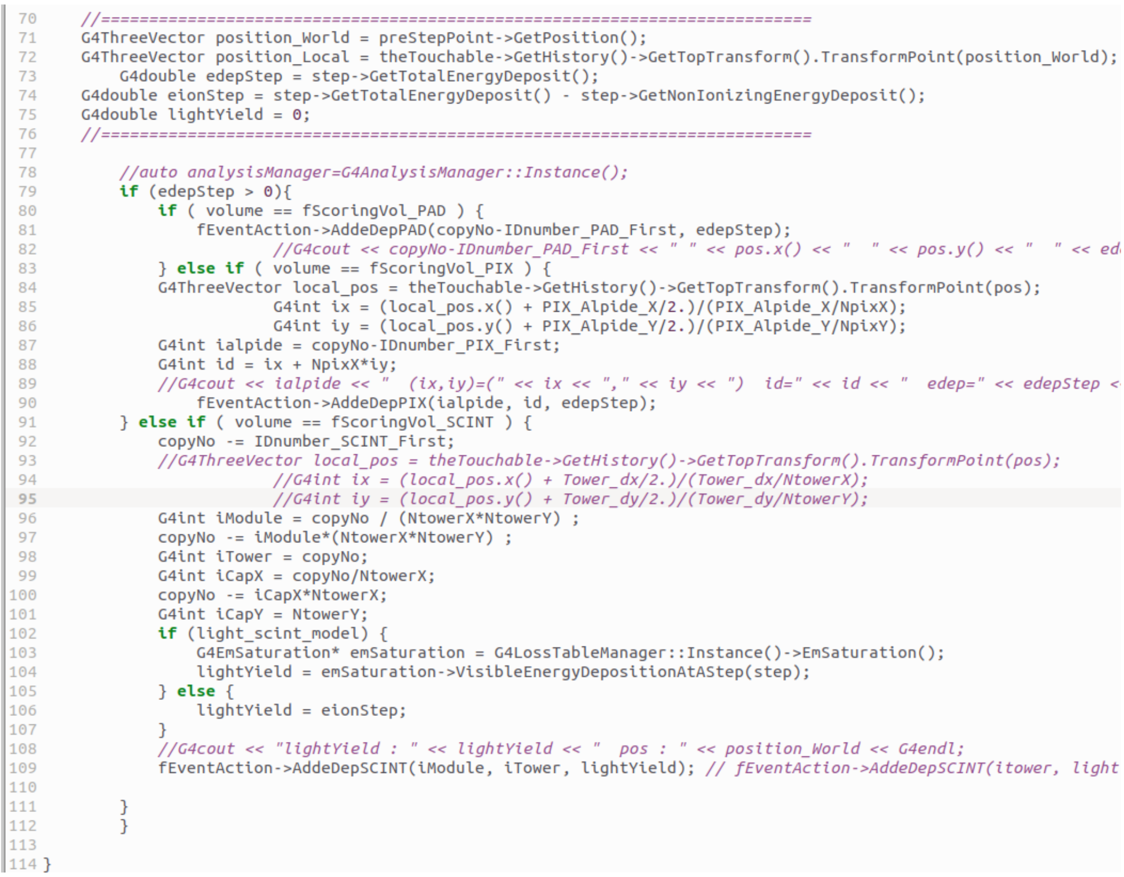


```
bool EMap[19][19][99] = { 0 };

int main() {
  for (int i = 0; i <= 100; i = i + 1) {
    for (int j = 0; j <= 20; j = j + 1) {
        for (int k = 0; k <= 20; k = k + 1) {
            if ( i - 1 < iz && iz < i + 1 && j - 1 < ix && ix < j + 1 && k - 1 < iy && iy < k + 1) {
               EMap[j][k][i] += lightyield;
            }
        }
     }
   }
  return 0;
}
```

Figure 26: In the above image: A method to place the energy at location X,Y,Z within a particular spot of a 3d array, using if loops to check whether the hit would be closer to the given spot in the array or any of its neighbors. As shown not using the proper dimensions for the actual calorimeter, but rather test values.

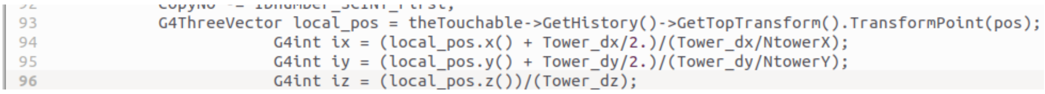
With this issue solved, the next step was to identify where the hits were calculated, which turned out to be, logically, the **SteppingAction.hh** file:



```
 70     //=====================================================================
 71     G4ThreeVector position_World = preStepPoint->GetPosition();
 72     G4ThreeVector position_Local = theTouchable->GetHistory()->GetTopTransform().TransformPoint(position_World);
 73         G4double edepStep = step->GetTotalEnergyDeposit();
 74     G4double eionStep = step->GetTotalEnergyDeposit() - step->GetNonIonizingEnergyDeposit();
 75     G4double lightYield = 0;
 76     //=====================================================================
 77 
 78         //auto analysisManager=G4AnalysisManager::Instance();
 79         if (edepStep > 0){
 80             if ( volume == fScoringVol_PAD ) {
 81                 fEventAction->AddeDepPAD(copyNo-IDnumber_PAD_First, edepStep);
 82                         //G4cout << copyNo-IDnumber_PAD_First << " " << pos.x() << "  " << pos.y() << "  " << ed
 83             } else if ( volume == fScoringVol_PIX ) {
 84             G4ThreeVector local_pos = theTouchable->GetHistory()->GetTopTransform().TransformPoint(pos);
 85                         G4int ix = (local_pos.x() + PIX_Alpide_X/2.)/(PIX_Alpide_X/NpixX);
 86                         G4int iy = (local_pos.y() + PIX_Alpide_Y/2.)/(PIX_Alpide_Y/NpixY);
 87             G4int ialpide = copyNo-IDnumber_PIX_First;
 88             G4int id = ix + NpixX*iy;
 89             //G4cout << ialpide << "  (ix,iy)=(" << ix << "," << iy << ")  id=" << id << "  edep=" << edepStep <
 90                 fEventAction->AddeDepPIX(ialpide, id, edepStep);
 91         } else if ( volume == fScoringVol_SCINT ) {
 92             copyNo -= IDnumber_SCINT_First;
 93             //G4ThreeVector local_pos = theTouchable->GetHistory()->GetTopTransform().TransformPoint(pos);
 94                         //G4int ix = (local_pos.x() + Tower_dx/2.)/(Tower_dx/NtowerX);
 95                         //G4int iy = (local_pos.y() + Tower_dy/2.)/(Tower_dy/NtowerY);
 96             G4int iModule = copyNo / (NtowerX*NtowerY) ;
 97             copyNo -= iModule*(NtowerX*NtowerY) ;
 98             G4int iTower = copyNo;
 99             G4int iCapX = copyNo/NtowerX;
100             copyNo -= iCapX*NtowerX;
101             G4int iCapY = NtowerY;
102             if (light_scint_model) {
103                 G4EmSaturation* emSaturation = G4LossTableManager::Instance()->EmSaturation();
104                 lightYield = emSaturation->VisibleEnergyDepositionAtAStep(step);
105             } else {
106                 lightYield = eionStep;
107             }
108             //G4cout << "lightYield : " << lightYield << "  pos : " << position_World << G4endl;
109             fEventAction->AddeDepSCINT(iModule, iTower, lightYield); // fEventAction->AddeDepSCINT(itower, light
110 
111         }
112         }
113 
114 }
```

Figure 27: The relevant part of the file, determining the position of hits within the two calorimeters, of particular note the X an Y position values (line 94 and 95) for FoCal-H are commented out.

So the first step was to uncomment those two and add a Z position value in addition to them.



```
 92             copyNo -= IDnumber_SCINT_First;
 93             G4ThreeVector local_pos = theTouchable->GetHistory()->GetTopTransform().TransformPoint(pos);
 94                         G4int ix = (local_pos.x() + Tower_dx/2.)/(Tower_dx/NtowerX);
 95                         G4int iy = (local_pos.y() + Tower_dy/2.)/(Tower_dy/NtowerY);
 96                         G4int iz = (local_pos.z())/(Tower_dz);
```

Figure 28: X and Y positions uncommented and Z position value added.

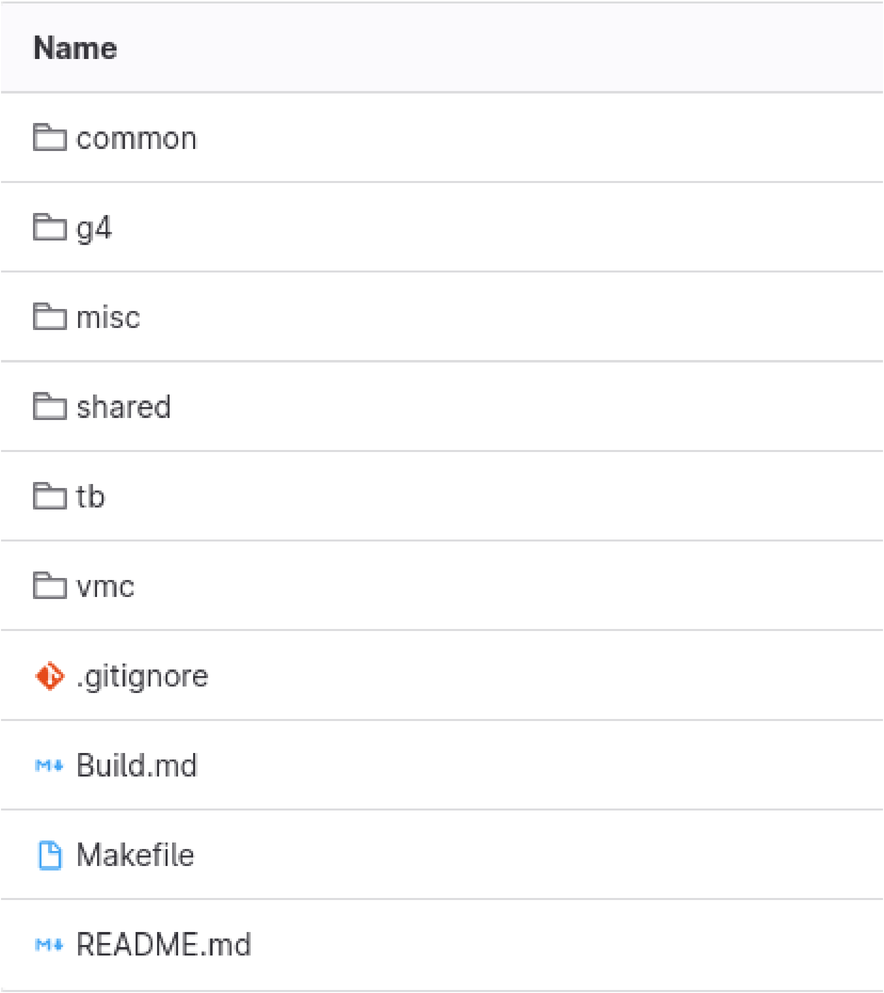
However the main issue preventing this would be the final lines where the eventaction is saved, as can be observed in the above image, it sums the values up in the end, discarding the individual hits for the sake of the sums, and as such the next problem to solve would be to get it to save this information. Or rather it would have been the next step if a more convenient solution hadn't come up.

4.2 The Focal_sim simulation

For the final goals and comparisons with data I used the "Focal_sim" package developed by Christian Holm Christensen [5] to achieve my results. This simulation package utilizes two different methods to do the simulation, a pure geant4 version which uses geant4 backend, and a vmc version that can use either geant3 or geant4, in this project I use both the pure geant4 and the vmc version to compare results. This simulation package also has several advantages over the previous used one, including its default analysis setup, which even allows it to analyse actual test beam data, due to the format of the simulated results being of the same form as the test beam data. Further more it has a strong ability for visualising the data, which will be demonstrated in the results section.

The main issue with the previous simulation package was that it didnt record the longitudinal event distribution, ie how the energy was deposited over the z axis, it gave a transverse viewpoint with no problem, however it didnt save the data from the z coordinates, and as a consequence it just summed them all up for the final result. Now of course the simulation would have to make the calculation and take the z position into account in order to even be remotely accurate, but to make it record and store this information would require changes in the Steppingaction file which dictate the individual events, which I also covered earlier. However while it would have been possible to change the previous package to make it capable of saving this data, it proved much simpler to change to this better alternative, which of course has the advantage to directly compare with test beam results.

Now as to showcase the simulation package properly, it will be prudent to go through the folders for the simulation, to explain the relevant parts of it and grant an overview of how it simulates and records the data.



Name
common
g4
misc
shared
tb
vmc
.gitignore
Build.md
Makefile
README.md

Figure 29: The contents of the primary folder of the Focal_sim package

Common is code shared between the two primary version of the simulation: vmc and g4 this is mainly just data and parameter classes that both of the versions will need to access in order to run the simulation.

Shared is similar to common this is code shared between the two versions, however in this case the code within here needs to be compiled seperately, as a consequence the two versions will have symbolic links pointing to this folder, to make sure that compiled code stays seperate between the two versions.

Misc is mainly just for storing various miscellaneous files, such as notes and similar. This folder isn't really relevant to this project and will be ignored moving foward.

VMC is one of the two main versions of the simulation package, this is specifically an implementation of the simulation using a virtual monte-carlo setup with root, using root vmc as well as either geant4 or geant3 for backend.

G4 is the other version of the simulation, this implementation uses a pure geant4 implementation and can as such not be run with geant3.

Tb is the folder for test beam data, this folder is used to analyse the actual test beam data from the test beam runs in September and November 2022

4.2.1 The VMC folder

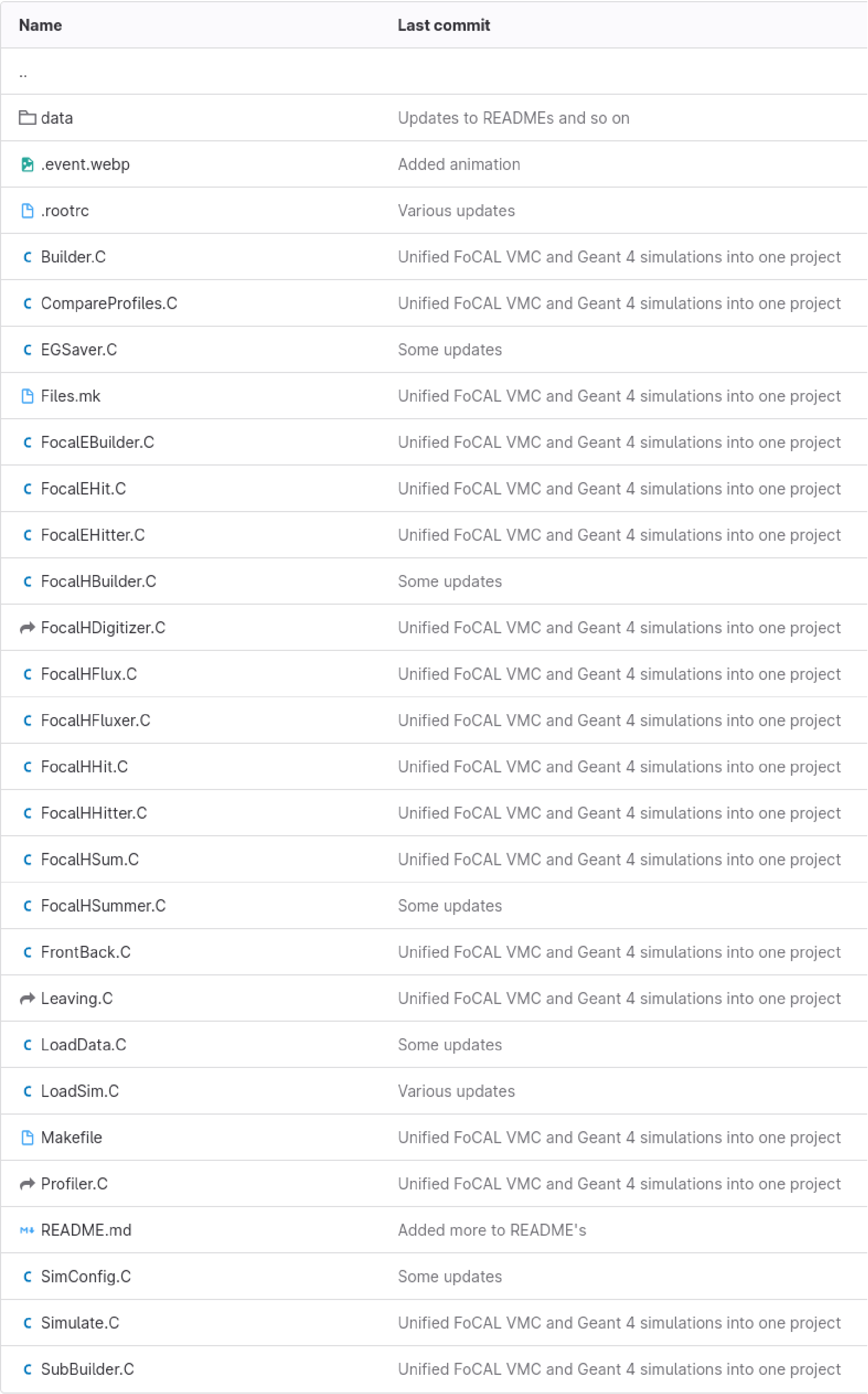
Starting our with a look at the vmc implementation it will be useful to take a look into the folder of this implementation in order to get a better view of it.
To start with:

Builder.C essemtialy builds the setup, it sets up the dimensions of the calorimeter, the materials used for it and their properites, it does not consider the particle beam or hit detection or anything, it just builds the setup.

FoCalHBuilder.C and **FoCalEBuilder.C** are the files that builds the geometry and properties of the FoCal-H and FoCal-E detectors respecively, again they do not take the hit detection into account as that is done by other files.

FoCalHHit.C and **FoCalEHit.C** both define what a hit within FoCal-H and FoCal-E are, including position, momentum, energy loss, and the specific straw volume the hit took place in for FoCal-H, and the pad or pixel volume for FoCal-E.

FoCalHSum.C is a summing mechanism for hits within FoCal-H, it will essentially sum up the number of hits and energy loss within the individual straw volumes inside of FoCal-H allowing us to see the total energy within each straw at the end of a given event.



Name	Last commit
..	
data	Updates to READMEs and so on
.event.webp	Added animation
.rootrc	Various updates
Builder.C	Unified FoCAL VMC and Geant 4 simulations into one project
CompareProfiles.C	Unified FoCAL VMC and Geant 4 simulations into one project
EGSaver.C	Some updates
Files.mk	Unified FoCAL VMC and Geant 4 simulations into one project
FocalEBuilder.C	Unified FoCAL VMC and Geant 4 simulations into one project
FocalEHit.C	Unified FoCAL VMC and Geant 4 simulations into one project
FocalEHitter.C	Unified FoCAL VMC and Geant 4 simulations into one project
FocalHBuilder.C	Some updates
FocalHDigitizer.C	Unified FoCAL VMC and Geant 4 simulations into one project
FocalHFlux.C	Unified FoCAL VMC and Geant 4 simulations into one project
FocalHFluxer.C	Unified FoCAL VMC and Geant 4 simulations into one project
FocalHHit.C	Unified FoCAL VMC and Geant 4 simulations into one project
FocalHHitter.C	Unified FoCAL VMC and Geant 4 simulations into one project
FocalHSum.C	Unified FoCAL VMC and Geant 4 simulations into one project
FocalHSummer.C	Some updates
FrontBack.C	Unified FoCAL VMC and Geant 4 simulations into one project
Leaving.C	Unified FoCAL VMC and Geant 4 simulations into one project
LoadData.C	Some updates
LoadSim.C	Various updates
Makefile	Unified FoCAL VMC and Geant 4 simulations into one project
Profiler.C	Unified FoCAL VMC and Geant 4 simulations into one project
README.md	Added more to README's
SimConfig.C	Some updates
Simulate.C	Unified FoCAL VMC and Geant 4 simulations into one project
SubBuilder.C	Unified FoCAL VMC and Geant 4 simulations into one project

FoCalHFlux.C is used to detect particles entering and leaving FoCal-H, it does this by "creating" a box volume that just exactly covers the entirety of FoCal-H, and then has detection for whether something enters or leaves the calorimeter, it then records the information about the particles entering or leaving, such as their momentum, position and whether they enter or leave the detector.

FoCalHHitter.C and **FoCalEHitter.C** are the specific detection of the hits, whereas the above files were used to define *what* a hit is, these are the mechanisms by which the simulation recognizes and records the actual hits within FoCal- H and FoCal-E

FoCalHSummer.C is like the two hitters above in that it is what records the hits, this one just sums up the energyloss within the specific fibres and then takes into account how long it takes for all these energy losses to occur. This time taking is to seperate distinct events from one another, as if it takes too long it will discard the hit for this event.

FoCalHFluxer.C is the corresponding file to record the hits defined by **FoCalHFlux.C**, it specifically is only concerned with the particles that enter or exit the calorimeter, it is not concerned with what occurs within the calorimeter, only what occurs at the edge, this is the method by which the particles that exit the calorimeter are found, which can be seen further below in the results chapter.

SimConfig.C is the file in which the simulation itself is set up, it is here that the backend options are set up (geant3 and geant4) as well as the type of particles used for the beam, by default pi+ mesons, it also introduces standard deviation in width of the beam to ensure that not all the beam particles hit exactly the same spot in X and Y. It also determines whether the GUI should be active.

Simulate.C is the script for running the actual simulation, it is here where you determine the properties of the beam, such as number of events, the Z momentum of the beam particles, as well as which backend to use. Even determine the name of the file generated. However this file is not really interacted with directly to run the simulation.

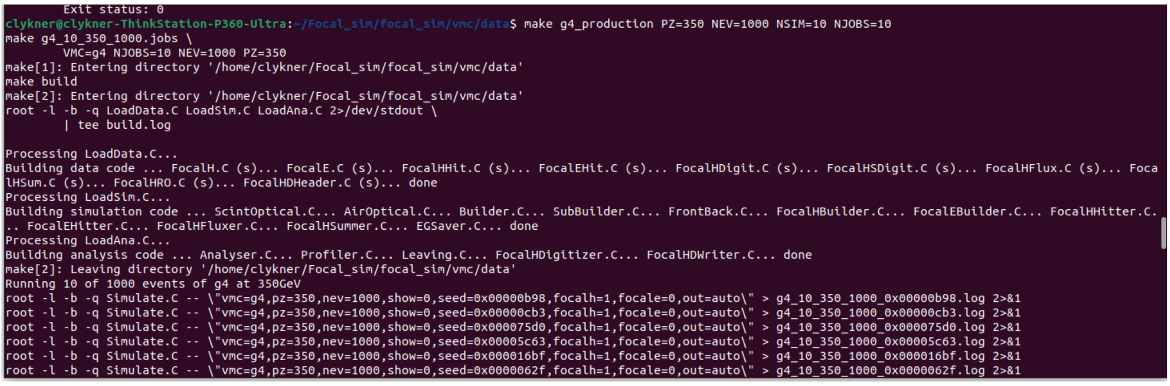
Some extra information

The simulation stores the information about the row and column of the detector through what is called the copy number, this number stores various bits of information about the location of hits within the detector. The number is a 32 bit string, wherein the various bits contain different pieces of information on the location. The first 8 bits (bit 0 to 7) contain the information of the row that the tube of the hit is in, the next 8 bits (8 to 15) contain the information on the column that the tube is in. The second half of the number contains the infor on the row and column of the whole module vs just the tube. So next 8 (16-23) contain the row of the module, and the final bits (24-31) contain the info on the column of the module.

4.2.2 Running the VMC implementation

In order to run the simulation I would use the ”make production” command, this command allows the user to specify the number of events, the energy of these events, and also the number of jobs the simulation should run simultaneously, this allows it to run several times faster than if it were to take it one event at a time. This does mean it creates several files for a given run, but it ensures that the seed between each of these files is different, making sure that the files can just be combined into one run, which is what is then done for the analysis of these runs.

Example of the process of running a simulation using the vmc implementation:

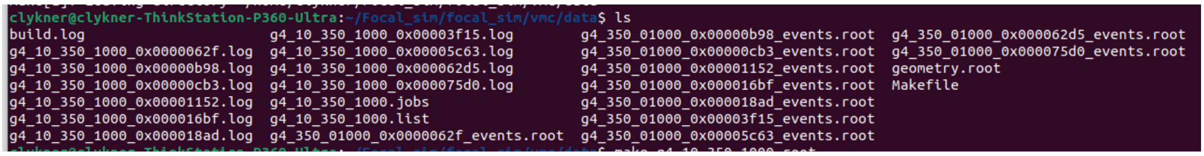


```
        Exit status: 0
clykner@clykner-ThinkStation-P360-Ultra:~/Focal_sim/focal_sim/vmc/data$ make g4_production PZ=350 NEV=1000 NSIM=10 NJOBS=10
make g4_10_350_1000.jobs \
        VMC=g4 NJOBS=10 NEV=1000 PZ=350
make[1]: Entering directory '/home/clykner/Focal_sim/focal_sim/vmc/data'
make build
make[2]: Entering directory '/home/clykner/Focal_sim/focal_sim/vmc/data'
root -l -b -q LoadData.C LoadSim.C LoadAna.C 2>/dev/stdout \
        | tee build.log

Processing LoadData.C...
Building data code ... FocalH.C (s)... FocalE.C (s)... FocalHHit.C (s)... FocalEHit.C (s)... FocalHDigit.C (s)... FocalHSDigit.C (s)... FocalHFlux.C (s)... Foca
lHSum.C (s)... FocalHRO.C (s)... FocalHDHeader.C (s)... done
Processing LoadSim.C...
Building simulation code ... ScintOptical.C... AirOptical.C... Builder.C... SubBuilder.C... FrontBack.C... FocalHBuilder.C... FocalEBuilder.C... FocalHHitter.C.
.. FocalEHitter.C... FocalHFluxer.C... FocalHSummer.C... EGSaver.C... done
Processing LoadAna.C...
Building analysis code ... Analyser.C... Profiler.C... Leaving.C... FocalHDigitizer.C... FocalHDWriter.C... done
make[2]: Leaving directory '/home/clykner/Focal_sim/focal_sim/vmc/data'
Running 10 of 1000 events of g4 at 350GeV
root -l -b -q Simulate.C -- \"vmc=g4,pz=350,nev=1000,show=0,seed=0x00000b98,focalh=1,focale=0,out=auto\" > g4_10_350_1000_0x00000b98.log 2>&1
root -l -b -q Simulate.C -- \"vmc=g4,pz=350,nev=1000,show=0,seed=0x00000cb3,focalh=1,focale=0,out=auto\" > g4_10_350_1000_0x00000cb3.log 2>&1
root -l -b -q Simulate.C -- \"vmc=g4,pz=350,nev=1000,show=0,seed=0x000075d0,focalh=1,focale=0,out=auto\" > g4_10_350_1000_0x000075d0.log 2>&1
root -l -b -q Simulate.C -- \"vmc=g4,pz=350,nev=1000,show=0,seed=0x00005c63,focalh=1,focale=0,out=auto\" > g4_10_350_1000_0x00005c63.log 2>&1
root -l -b -q Simulate.C -- \"vmc=g4,pz=350,nev=1000,show=0,seed=0x000016bf,focalh=1,focale=0,out=auto\" > g4_10_350_1000_0x000016bf.log 2>&1
root -l -b -q Simulate.C -- \"vmc=g4,pz=350,nev=1000,show=0,seed=0x0000062f,focalh=1,focale=0,out=auto\" > g4_10_350_1000_0x0000062f.log 2>&1
```

In this case I used the make production command, this command defines the various properties of the simulation run before running it, whereas if I were to just type make production, I would need to specify what the parameters were in the SimConfig.C file, however this becomes unnecessary as it is simpler to define the parameters in the run command instead. Here I ask it to run the g4 backend, with 10 parallel runs at 350 GeV and 1000 events, for a total 10000 event run.

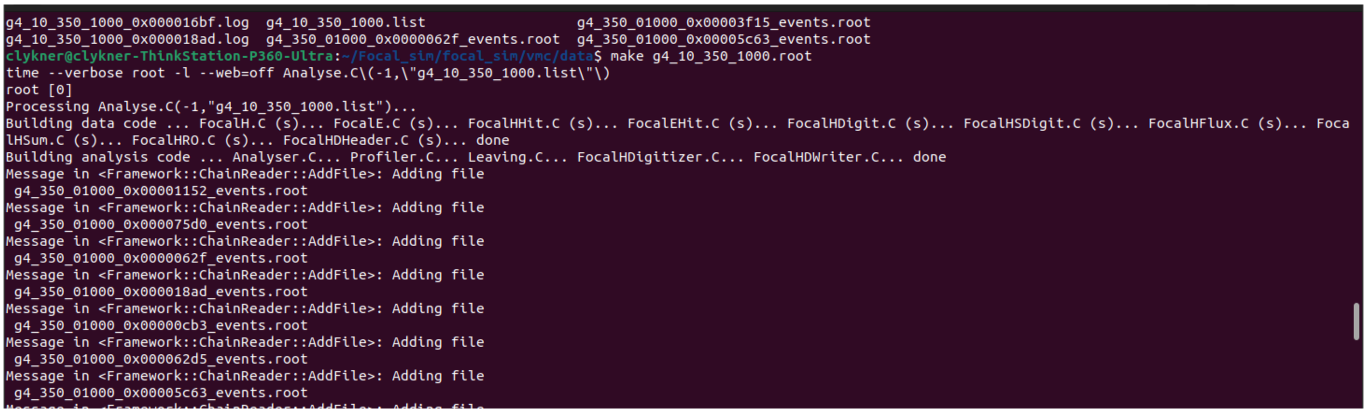
Doing this results in several files:



```
clykner@clykner-ThinkStation-P360-Ultra:~/Focal_sim/focal_sim/vmc/data$ ls
build.log                     g4_10_350_1000_0x00003f15.log       g4_350_01000_0x00000b98_events.root  g4_350_01000_0x000062d5_events.root
g4_10_350_1000_0x0000062f.log  g4_10_350_1000_0x00005c63.log       g4_350_01000_0x00000cb3_events.root  g4_350_01000_0x000075d0_events.root
g4_10_350_1000_0x00000b98.log  g4_10_350_1000_0x000062d5.log       g4_350_01000_0x00001152_events.root  geometry.root
g4_10_350_1000_0x00000cb3.log  g4_10_350_1000_0x000075d0.log       g4_350_01000_0x000016bf_events.root  Makefile
g4_10_350_1000_0x00001152.log  g4_10_350_1000.jobs                 g4_350_01000_0x000018ad_events.root
g4_10_350_1000_0x000016bf.log  g4_10_350_1000.list                 g4_350_01000_0x00003f15_events.root
g4_10_350_1000_0x000018ad.log  g4_350_01000_0x0000062f_events.root  g4_350_01000_0x00005c63_events.root
```

These files are the various root files containing the results of the simulation, they are named with their main properties, the backend version in front, followed by the number of events and finally the random seeds it used for this particular file. from here you could use the cat command to combine them into one file, however that will not be necessary for these runs.

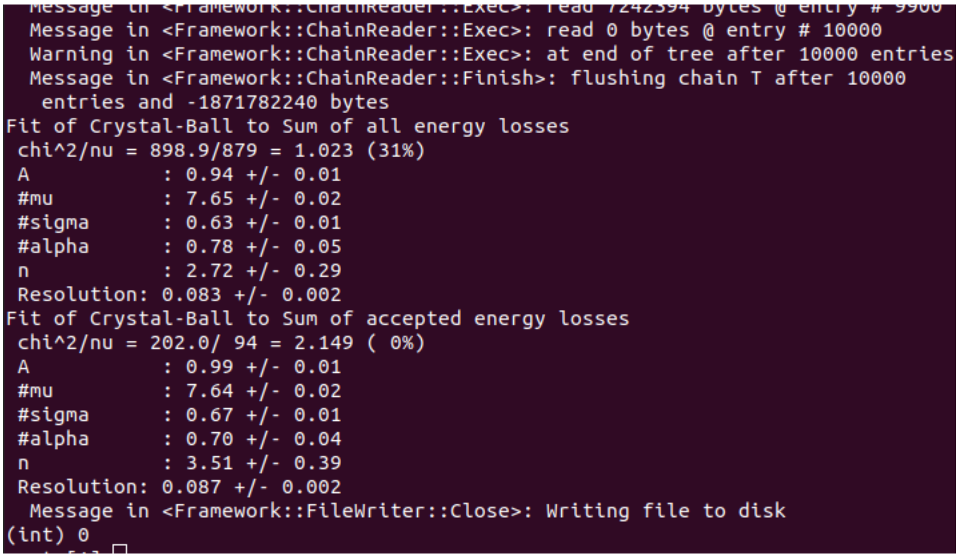
Instead it is simpler to use the built-in analysis functions, this is done by using the list file generated. The list file is rather simple, it just contains a list of the various root files, however this is enough, by typing in "make g4_10_350_1000.root" note here that I replaced the .list with .root, this runs the analysis script on the list file, and using the list file it then finds all the root files and begins to analyse them collectively, it should be noted that it is very much possible to look at each root file independently by opening it in root. Analysing them together gives the following output:



```
g4_10_350_1000_0x000016bf.log  g4_10_350_1000.list                  g4_350_01000_0x00003f15_events.root
g4_10_350_1000_0x000018ad.log  g4_350_01000_0x0000062f_events.root  g4_350_01000_0x00005c63_events.root
clykner@clykner-ThinkStation-P360-Ultra:~/Focal_sim/focal_sim/vmc/data$ make g4_10_350_1000.root
time --verbose root -l --web=off Analyse.C\(-1,\"g4_10_350_1000.list\"\)
root [0]
Processing Analyse.C(-1,"g4_10_350_1000.list")...
Building data code ... FocalH.C (s)... FocalE.C (s)... FocalHHit.C (s)... FocalEHit.C (s)... FocalHDigit.C (s)... FocalHSDigit.C (s)... FocalHFlux.C (s)... Foca
lHSum.C (s)... FocalHRO.C (s)... FocalHDHeader.C (s)... done
Building analysis code ... Analyser.C... Profiler.C... Leaving.C... FocalHDigitizer.C... FocalHDWriter.C... done
Message in <Framework::ChainReader::AddFile>: Adding file
 g4_350_01000_0x00001152_events.root
Message in <Framework::ChainReader::AddFile>: Adding file
 g4_350_01000_0x000075d0_events.root
Message in <Framework::ChainReader::AddFile>: Adding file
 g4_350_01000_0x0000062f_events.root
Message in <Framework::ChainReader::AddFile>: Adding file
 g4_350_01000_0x000018ad_events.root
Message in <Framework::ChainReader::AddFile>: Adding file
 g4_350_01000_0x00000cb3_events.root
Message in <Framework::ChainReader::AddFile>: Adding file
 g4_350_01000_0x000062d5_events.root
Message in <Framework::ChainReader::AddFile>: Adding file
 g4_350_01000_0x00005c63_events.root
```

As can be seen here it then adds the various root files together in order to create a single file containing the information of the run.

In the end the analysis will return with a fit of the various energylosses and the associated values for this fit:



```
  Message in <Framework::ChainReader::Exec>: read 7242394 bytes @ entry # 9900
  Message in <Framework::ChainReader::Exec>: read 0 bytes @ entry # 10000
  Warning in <Framework::ChainReader::Exec>: at end of tree after 10000 entries
  Message in <Framework::ChainReader::Finish>: flushing chain T after 10000
   entries and -1871782240 bytes
Fit of Crystal-Ball to Sum of all energy losses
 chi^2/nu = 898.9/879 = 1.023 (31%)
 A           : 0.94 +/- 0.01
 #mu         : 7.65 +/- 0.02
 #sigma      : 0.63 +/- 0.01
 #alpha      : 0.78 +/- 0.05
 n           : 2.72 +/- 0.29
 Resolution: 0.083 +/- 0.002
Fit of Crystal-Ball to Sum of accepted energy losses
 chi^2/nu = 202.0/ 94 = 2.149 ( 0%)
 A           : 0.99 +/- 0.01
 #mu         : 7.64 +/- 0.02
 #sigma      : 0.67 +/- 0.01
 #alpha      : 0.70 +/- 0.04
 n           : 3.51 +/- 0.39
 Resolution: 0.087 +/- 0.002
  Message in <Framework::FileWriter::Close>: Writing file to disk
(int) 0
```

The full output of this analysis will be covered further into the report, in the FoCal_sim part of the results section.

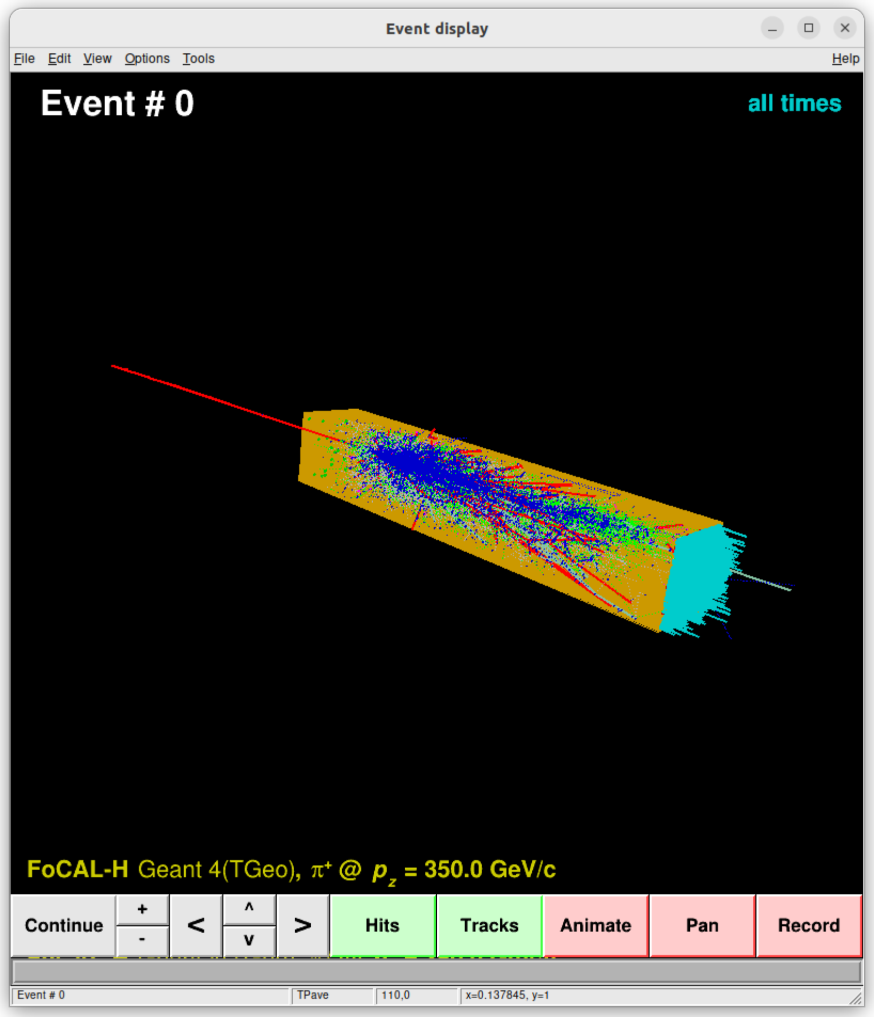
This is however not the only way to run the simulation, indeed there is another way that is of relevance to the report, and is generally useful for the sake of prudence. That method would be the interactive session of the simulation. This can be done by using the expected "make interactive" command, however this will run with the default settings of the simulation, 50 GeV, and would require to edit the SimConfig.C file in order to change this. Fortunately it is also set to by default use the interactive format, so therefore I can in this case simply use the

```
make    vr_E_n_events.root
```

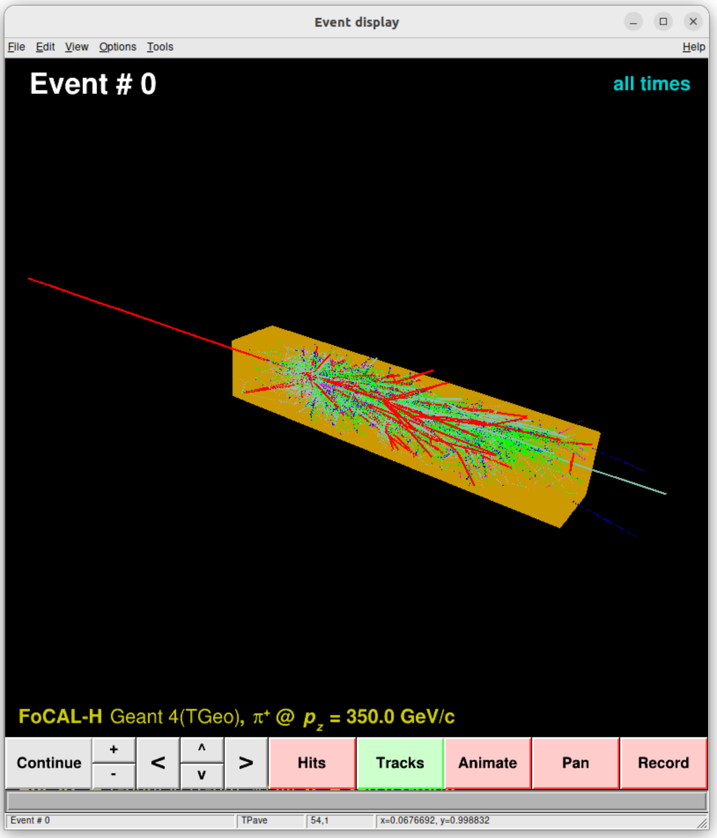
command, where:

1. vr is the backend version I intend to use, in this case g4
2. n is the number of events that I want it to run, for the sake of viewing a single event this will be set to 1
3. E is the energy of these events in GeV, in the case of what I will cover this will be 350.

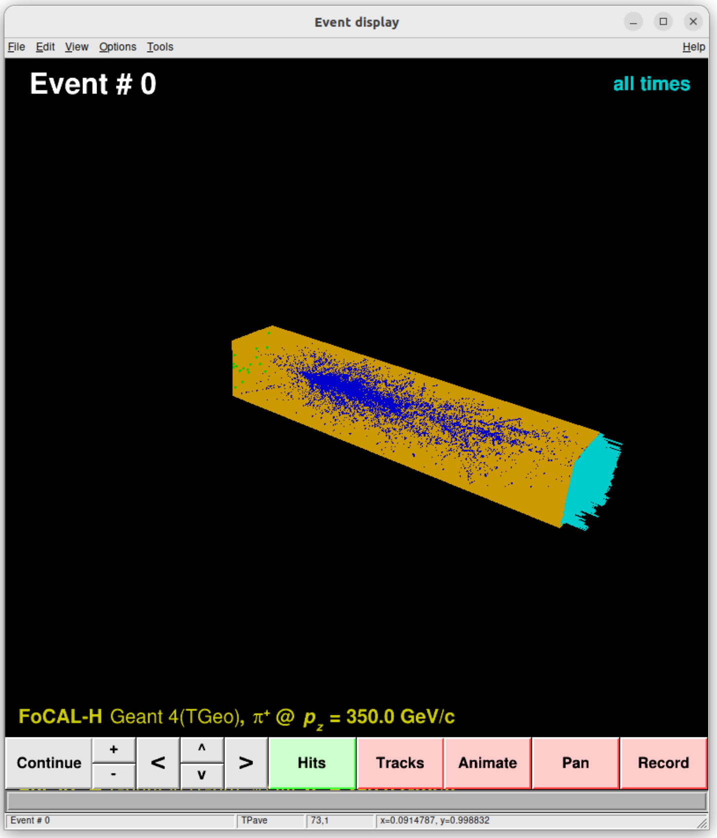
This then opens the simulation in interactive mode, which allows us to view the hit distribution througout the calorimeter in each event.



Here is the full interactive view of one event at 350 GeV within FoCal-H, as can be seen it showcases both the hits and the particle tracks for this event. It is possible to turn the display of these on or off. This interactive version is of interest because it allows us to see the various parts of the simulation at work, we can see the builder files having constructed the geometry of the setup, and we can see the hit files showing the hits and tracks within the calorimeter, finally at the edge we can see the summed energy loss within each straw demonstrated by the cyan lines at the end.



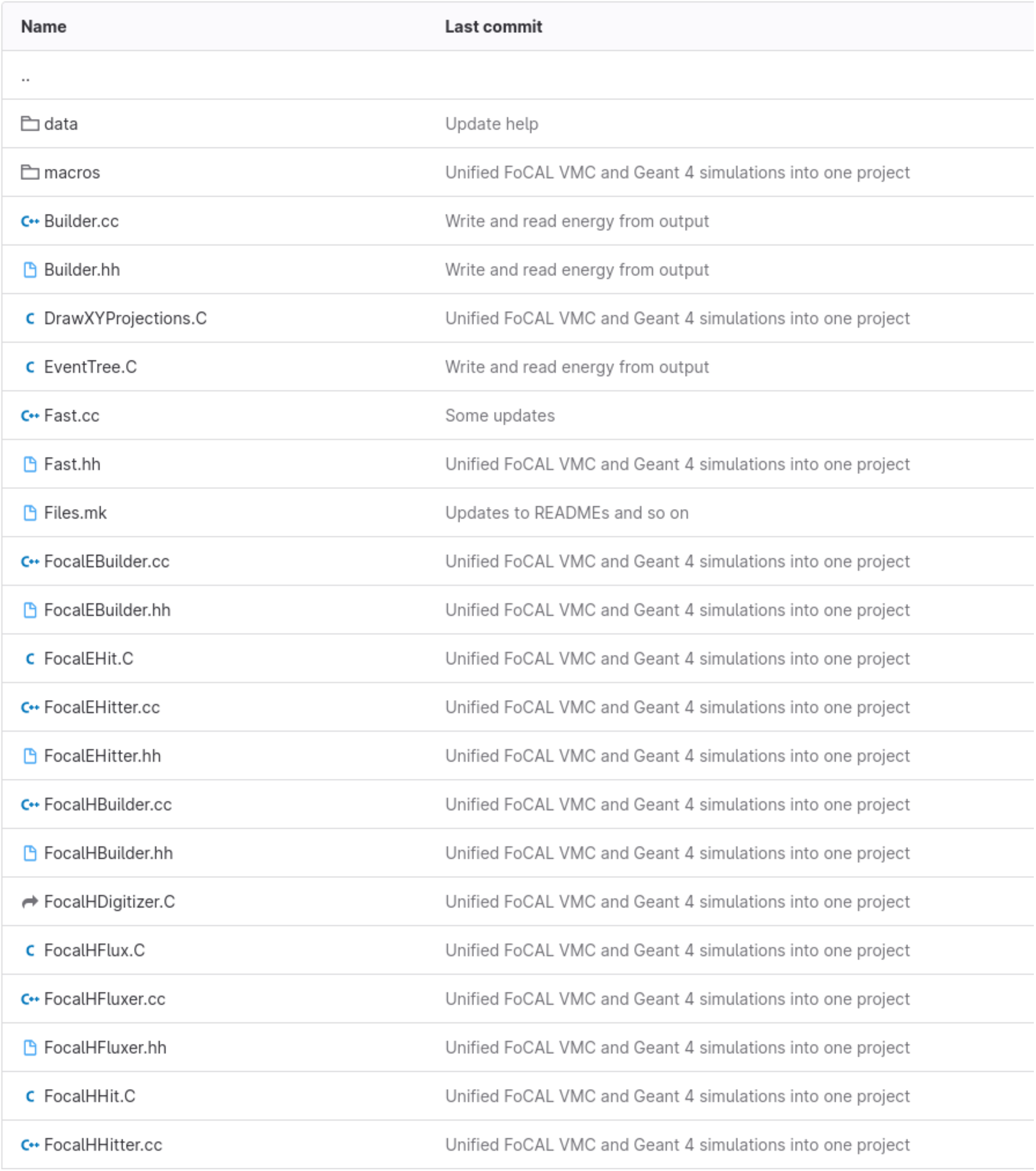
(a) Showcased here: The interactive display, however only showing the hits within the calorimeter, thus we just show the hit distribution.



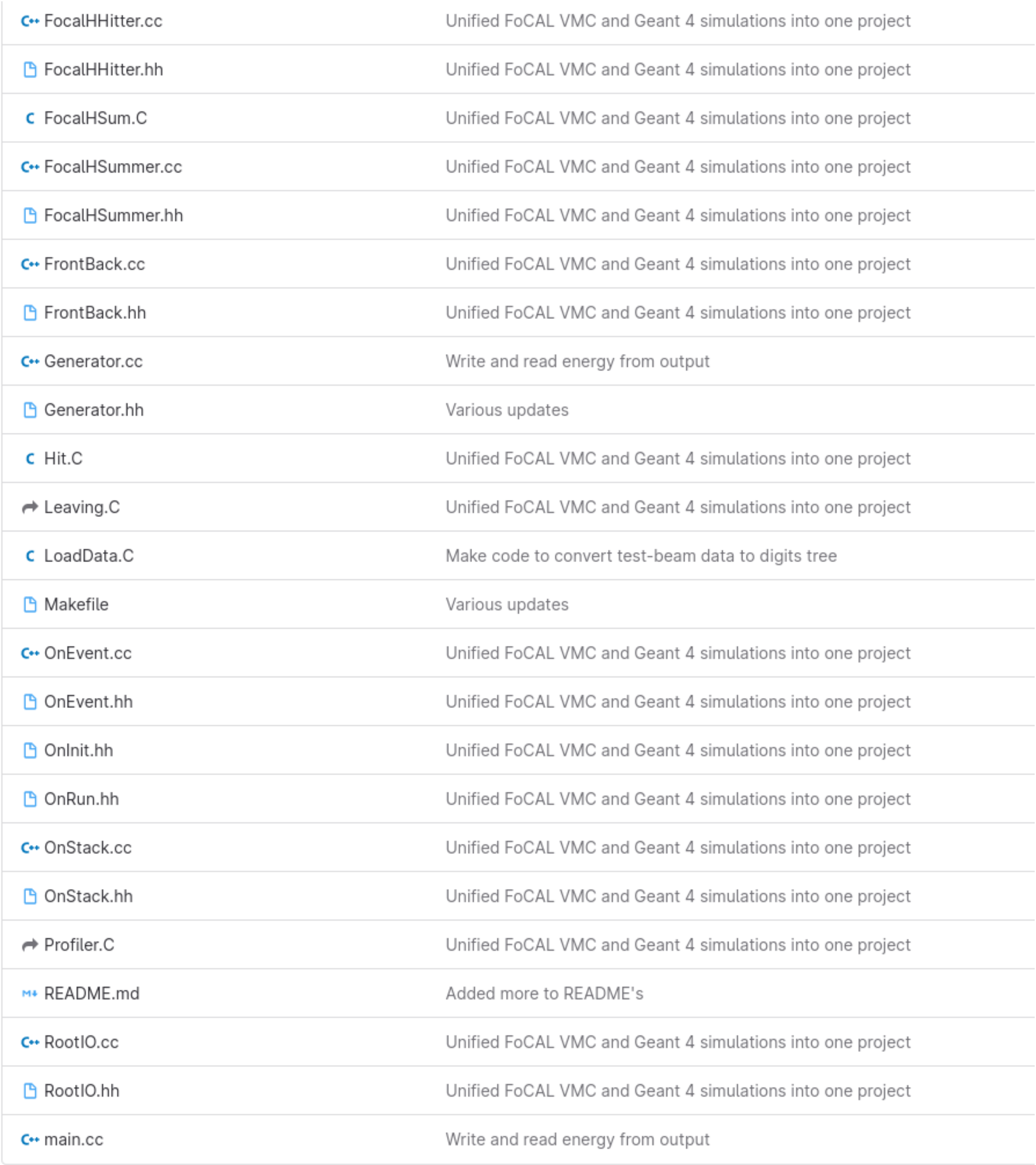
(b) Showcased here: The interactive display, however in this case the hits have been removed leaving only the particle tracks, showcasing the distribution within the calorimeter.

4.2.3 The Geant4 implementation

The VMC implementation is however not the only implementation of the simulation, the other is the purely geant4 version, whereas the VMC implementation uses geant4 or geant3 for backend, this implementation only uses geant4, and is therefore kept seperate from the other implementation. Taking a look into its folder we can see:



Name	Last commit
..	
data	Update help
macros	Unified FoCAL VMC and Geant 4 simulations into one project
Builder.cc	Write and read energy from output
Builder.hh	Write and read energy from output
DrawXYProjections.C	Unified FoCAL VMC and Geant 4 simulations into one project
EventTree.C	Write and read energy from output
Fast.cc	Some updates
Fast.hh	Unified FoCAL VMC and Geant 4 simulations into one project
Files.mk	Updates to READMEs and so on
FocalEBuilder.cc	Unified FoCAL VMC and Geant 4 simulations into one project
FocalEBuilder.hh	Unified FoCAL VMC and Geant 4 simulations into one project
FocalEHit.C	Unified FoCAL VMC and Geant 4 simulations into one project
FocalEHitter.cc	Unified FoCAL VMC and Geant 4 simulations into one project
FocalEHitter.hh	Unified FoCAL VMC and Geant 4 simulations into one project
FocalHBuilder.cc	Unified FoCAL VMC and Geant 4 simulations into one project
FocalHBuilder.hh	Unified FoCAL VMC and Geant 4 simulations into one project
FocalHDigitizer.C	Unified FoCAL VMC and Geant 4 simulations into one project
FocalHFlux.C	Unified FoCAL VMC and Geant 4 simulations into one project
FocalHFluxer.cc	Unified FoCAL VMC and Geant 4 simulations into one project
FocalHFluxer.hh	Unified FoCAL VMC and Geant 4 simulations into one project
FocalHHit.C	Unified FoCAL VMC and Geant 4 simulations into one project
FocalHHitter.cc	Unified FoCAL VMC and Geant 4 simulations into one project



FocalHHitter.cc	Unified FoCAL VMC and Geant 4 simulations into one project
FocalHHitter.hh	Unified FoCAL VMC and Geant 4 simulations into one project
FocalHSum.C	Unified FoCAL VMC and Geant 4 simulations into one project
FocalHSummer.cc	Unified FoCAL VMC and Geant 4 simulations into one project
FocalHSummer.hh	Unified FoCAL VMC and Geant 4 simulations into one project
FrontBack.cc	Unified FoCAL VMC and Geant 4 simulations into one project
FrontBack.hh	Unified FoCAL VMC and Geant 4 simulations into one project
Generator.cc	Write and read energy from output
Generator.hh	Various updates
Hit.C	Unified FoCAL VMC and Geant 4 simulations into one project
Leaving.C	Unified FoCAL VMC and Geant 4 simulations into one project
LoadData.C	Make code to convert test-beam data to digits tree
Makefile	Various updates
OnEvent.cc	Unified FoCAL VMC and Geant 4 simulations into one project
OnEvent.hh	Unified FoCAL VMC and Geant 4 simulations into one project
OnInit.hh	Unified FoCAL VMC and Geant 4 simulations into one project
OnRun.hh	Unified FoCAL VMC and Geant 4 simulations into one project
OnStack.cc	Unified FoCAL VMC and Geant 4 simulations into one project
OnStack.hh	Unified FoCAL VMC and Geant 4 simulations into one project
Profiler.C	Unified FoCAL VMC and Geant 4 simulations into one project
README.md	Added more to README's
RootIO.cc	Unified FoCAL VMC and Geant 4 simulations into one project
RootIO.hh	Unified FoCAL VMC and Geant 4 simulations into one project
main.cc	Write and read energy from output

Some general comments for this folder: Many of the files have .hh and .cc pairs that are named the same except for the filetypes. All files with these endings (even single files with .hh filetype) are all part of the simulation code and will be compiled together into a main file.

Now, onto the files themselves:

Builder.cc, **Builder.hh** are the main files for building the geometry of the simulation, much like the vmc implementation, they use the **FoCalHBuilder.cc**, **FoCalHBuilder.hh**, **FoCalEbuilder.cc** and **FoCalEBuilder.hh** files to construct the various parts of the calorimeter setup.

Hit.C is the main class for a hit and serves as the basis of the datastructure.

FoCalHHit.C and **FoCalEHit.C** is a hit within FoCal-H or FoCal-E respectively, much like in the vmc implementation it encodes all the relevant info, position, energy loss, momentum and the specific straw in the case of FoCal-H or the specifics pads and/or pixels for FoCal-E.

FoCalHSum.C and **FoCalHFlux.C** both function like in the VMC implementation, so the sum encodes the sum of energy losses within each straw and the Flux creates this "box" around the calorimeter in order to detect what enters and leaves the calorimeter.

EventTree.C sets up the main class eventtree which serves as the main structure of the simulation, it has the main methods for generating hits and similar processes for the simulation.

RootIO.hh and **RooTIO.cc** are singleton wrappers for the EventTree, their purpose is to allow global access to the EventTree and as such will be called upon by the OnEvent files and the main detectors.

FoCalHHitter.cc, **FoCalHHitter.hh**, **FoCalEHitter.cc** and **FoCalEHitter.hh** all serve to create the hits in FoCal-H and FoCal-E respectively.

FoCalHSummer.cc and **FoCalHSummer.hh** creates the sum output in each straw, including a modelled light yield and number of photons entering through the back of the calorimeter, this is again used to model total energy loss per straw.

FoCalHFluxer.cc and **FoCalHFluxer.hh** both serve to detect whenever a particle exits through the back of the calorimeter, this is used to determine the type of particles that leave and potentially also the flux of those particles through the calorimeter.

FrontBack.cc and **FrontBack.hh** are used as part of the Fluxers and the Summers to determine whether a particle, crossing the boundary defined by the flux files, is leaving or entering the FoCal-H volume.

The **OnRun.hh** file does nothing for this simulation setup, in other instances it is used to execute certain things upon starting a run.

OnEvent.cc and **OnEvent.hh** both serve to create the OnEvent class, which fulfills various functions whenever an event begins, mainly cleans out event storage and moves the event data to disk. Objects of this class use the RootIO singleton for these purposes.

OnStack.cc and **OnStack.hh** records the tracks within a given branch of the output TTree.

Generator.cc and **Generator.hh** both generate the primary particle of any given event, this is done with a particle cannon with a certain transverse smearing that is determined by a standard distribution to make sure that hits aren't identical.

OnInit.hh is the main file for initializing the simulation, this file will call up and initialize the Generator, OnRun, OnStack and OnEvent objects, once it has done this it does nothing more.

main.cc is the main program and can be used to initialize the simulation directly, as such it can take a number of configurations as argument upon launching it, however it is also not going to be the main way in which this simulation will be run in this project.

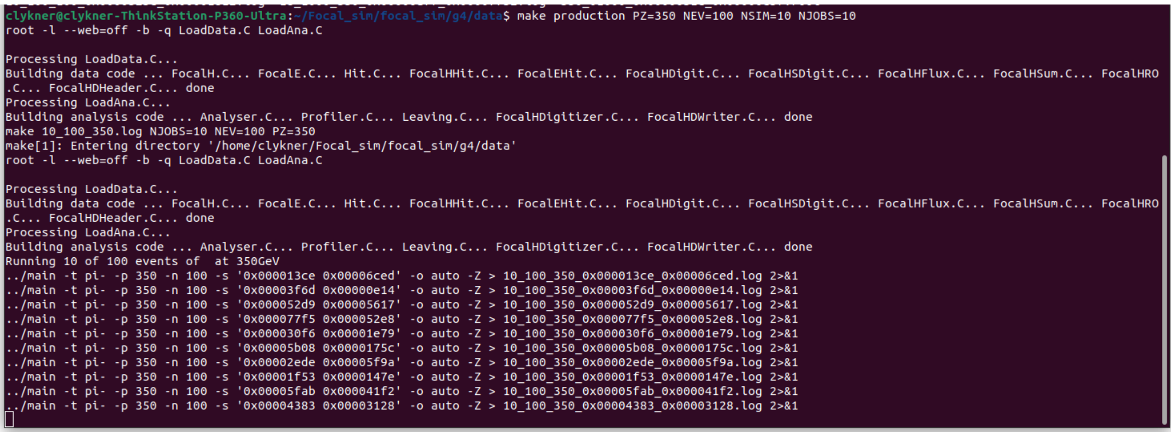
Finally the folder also has a folder for macros which can be used upon launching the simulation to launch it with predetermined configurations.

4.2.4 Running the Geant4 implementation

To run this simulation implementation it is again beneficial to use the ”make production” command, there are however a difference between this and the VMC implementation, in that you no longer need to specify which backend you use, as it can only use geant4. so the command just becomes:

```
make production PZ=350 NEV=100 NSIM=10 NJOBS=10
```

which again specifies how many simultaneous jobs I want it to run, and how many events per job, in this case its 10 jobs of 100 events at 350 GeV for a total of 1000 events at 350 GeV. The process of this and the output is again quite similar to the VMC version.

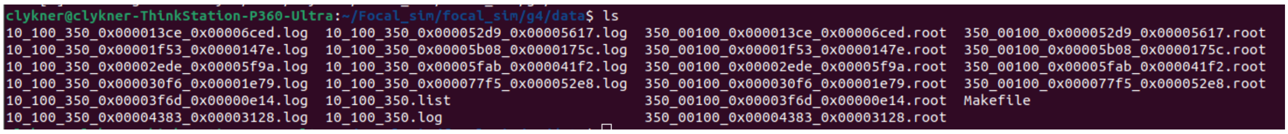


```
clykner@clykner-ThinkStation-P360-Ultra:~/Focal_sim/focal_sim/g4/data$ make production PZ=350 NEV=100 NSIM=10 NJOBS=10
root -l --web=off -b -q LoadData.C LoadAna.C

Processing LoadData.C...
Building data code ... FocalH.C... FocalE.C... Hit.C... FocalHHit.C... FocalEHit.C... FocalHDigit.C... FocalHSDigit.C... FocalHFlux.C... FocalHSum.C... FocalHRO
.C... FocalHDHeader.C... done
Processing LoadAna.C...
Building analysis code ... Analyser.C... Profiler.C... Leaving.C... FocalHDigitizer.C... FocalHDWriter.C... done
make 10_100_350.log NJOBS=10 NEV=100 PZ=350
make[1]: Entering directory '/home/clykner/Focal_sim/focal_sim/g4/data'
root -l --web=off -b -q LoadData.C LoadAna.C

Processing LoadData.C...
Building data code ... FocalH.C... FocalE.C... Hit.C... FocalHHit.C... FocalEHit.C... FocalHDigit.C... FocalHSDigit.C... FocalHFlux.C... FocalHSum.C... FocalHRO
.C... FocalHDHeader.C... done
Processing LoadAna.C...
Building analysis code ... Analyser.C... Profiler.C... Leaving.C... FocalHDigitizer.C... FocalHDWriter.C... done
Running 10 of 100 events of  at 350GeV
../main -t pi- -p 350 -n 100 -s '0x000013ce 0x00006ced' -o auto -Z > 10_100_350_0x000013ce_0x00006ced.log 2>&1
../main -t pi- -p 350 -n 100 -s '0x00003f6d 0x00000e14' -o auto -Z > 10_100_350_0x00003f6d_0x00000e14.log 2>&1
../main -t pi- -p 350 -n 100 -s '0x000052d9 0x00005617' -o auto -Z > 10_100_350_0x000052d9_0x00005617.log 2>&1
../main -t pi- -p 350 -n 100 -s '0x000077f5 0x000052e8' -o auto -Z > 10_100_350_0x000077f5_0x000052e8.log 2>&1
../main -t pi- -p 350 -n 100 -s '0x000030f6 0x00001e79' -o auto -Z > 10_100_350_0x000030f6_0x00001e79.log 2>&1
../main -t pi- -p 350 -n 100 -s '0x00005b08 0x0000175c' -o auto -Z > 10_100_350_0x00005b08_0x0000175c.log 2>&1
../main -t pi- -p 350 -n 100 -s '0x00002ede 0x00005f9a' -o auto -Z > 10_100_350_0x00002ede_0x00005f9a.log 2>&1
../main -t pi- -p 350 -n 100 -s '0x00001f53 0x0000147e' -o auto -Z > 10_100_350_0x00001f53_0x0000147e.log 2>&1
../main -t pi- -p 350 -n 100 -s '0x00005fab 0x000041f2' -o auto -Z > 10_100_350_0x00005fab_0x000041f2.log 2>&1
../main -t pi- -p 350 -n 100 -s '0x00004383 0x00003128' -o auto -Z > 10_100_350_0x00004383_0x00003128.log 2>&1
```

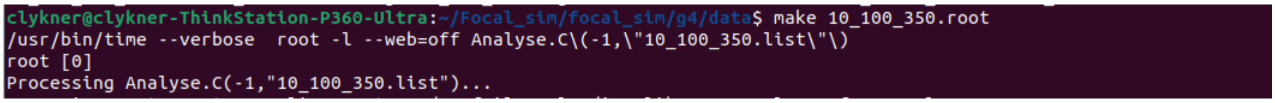
With the output:



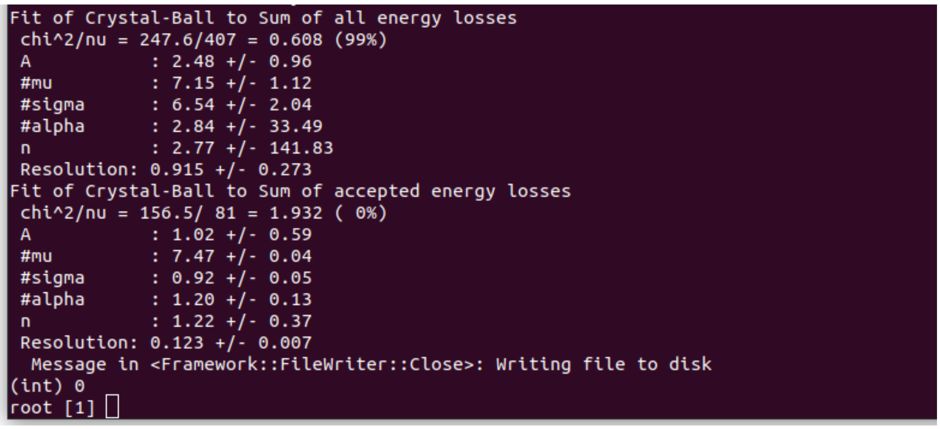
```
clykner@clykner-ThinkStation-P360-Ultra:~/Focal_sim/focal_sim/g4/data$ ls
10_100_350_0x000013ce_0x00006ced.log  10_100_350_0x000052d9_0x00005617.log  350_00100_0x000013ce_0x00006ced.root  350_00100_0x000052d9_0x00005617.root
10_100_350_0x00001f53_0x0000147e.log  10_100_350_0x00005b08_0x0000175c.log  350_00100_0x00001f53_0x0000147e.root  350_00100_0x00005b08_0x0000175c.root
10_100_350_0x00002ede_0x00005f9a.log  10_100_350_0x00005fab_0x000041f2.log  350_00100_0x00002ede_0x00005f9a.root  350_00100_0x00005fab_0x000041f2.root
10_100_350_0x000030f6_0x00001e79.log  10_100_350_0x000077f5_0x000052e8.log  350_00100_0x000030f6_0x00001e79.root  350_00100_0x000077f5_0x000052e8.root
10_100_350_0x00003f6d_0x00000e14.log  10_100_350.list                       350_00100_0x00003f6d_0x00000e14.root  Makefile
10_100_350_0x00004383_0x00003128.log  10_100_350.log                        350_00100_0x00004383_0x00003128.root
```

Again this output is treated quite similarly to the output of the VMC version, as the package uses the same analysis tools to treat both versions giving essentially an identical format between the two versions.

As can be observed here the process of running the analysis, and the format of the output is identical:



```
clykner@clykner-ThinkStation-P360-Ultra:~/Focal_sim/focal_sim/g4/data$ make 10_100_350.root
/usr/bin/time --verbose  root -l --web=off Analyse.C\(-1,\"10_100_350.list\"\)
root [0]
Processing Analyse.C(-1,"10_100_350.list")...
```



```
Fit of Crystal-Ball to Sum of all energy losses
 chi^2/nu = 247.6/407 = 0.608 (99%)
 A           : 2.48 +/- 0.96
 #mu         : 7.15 +/- 1.12
 #sigma      : 6.54 +/- 2.04
 #alpha      : 2.84 +/- 33.49
 n           : 2.77 +/- 141.83
 Resolution: 0.915 +/- 0.273
Fit of Crystal-Ball to Sum of accepted energy losses
 chi^2/nu = 156.5/ 81 = 1.932 ( 0%)
 A           : 1.02 +/- 0.59
 #mu         : 7.47 +/- 0.04
 #sigma      : 0.92 +/- 0.05
 #alpha      : 1.20 +/- 0.13
 n           : 1.22 +/- 0.37
 Resolution: 0.123 +/- 0.007
  Message in <Framework::FileWriter::Close>: Writing file to disk
(int) 0
root [1]
```

Again, full results will be covered later in the report.

Like for the VMC implementation there is also an interactive version for this implementation, here however I use the standard

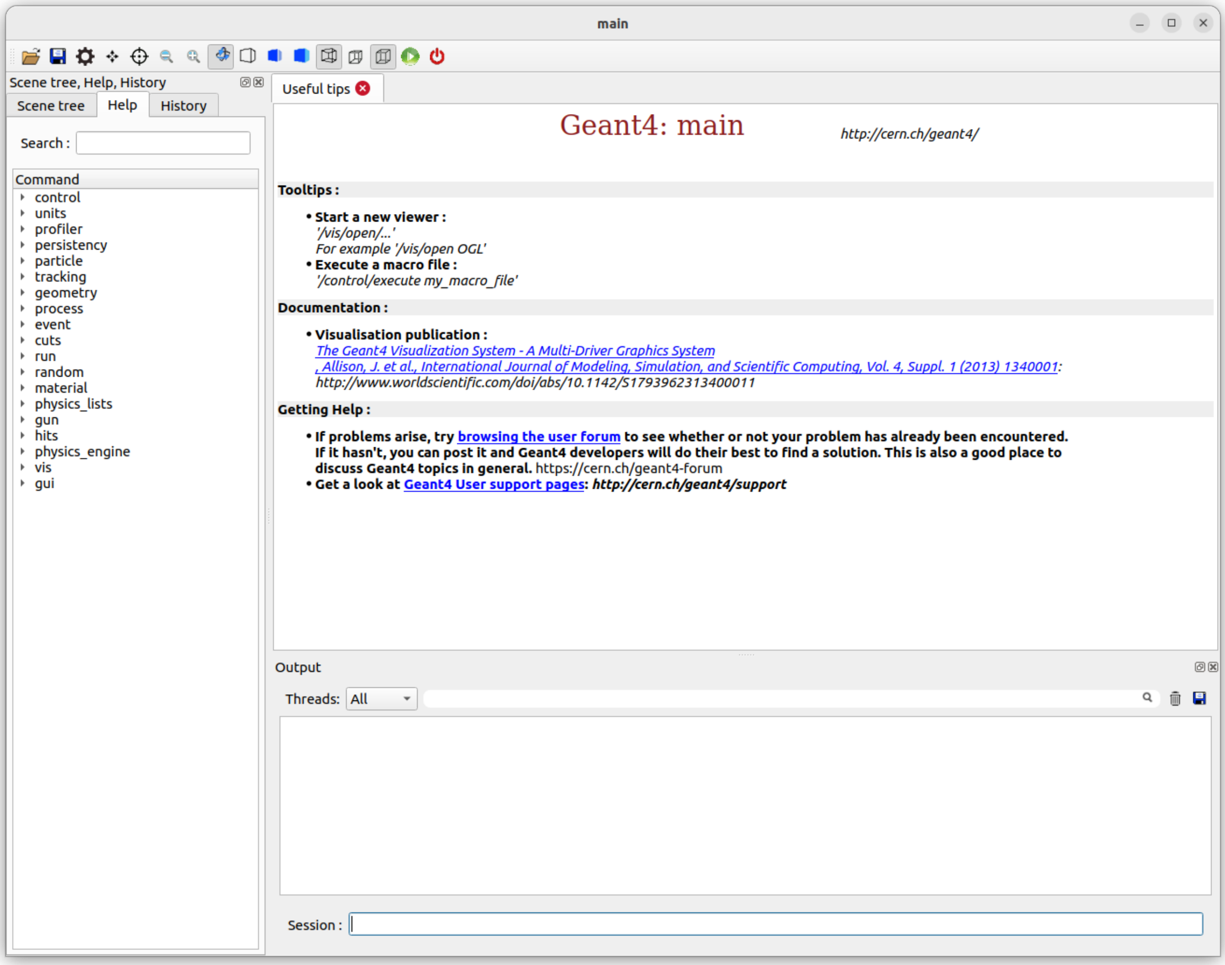
make interactive

command to generate the GUI, if you specify the number of events and the energy by writing for instance:

make 350_1_events.root interactive

to generate the interactive version for 350 GeV.

To start it returns the following page:



Geant4: main http://cern.ch/geant4/

Tooltips :

- **Start a new viewer :**
  '/vis/open/...'
  For example '/vis/open OGL'
- **Execute a macro file :**
  '/control/execute my_macro_file'

Documentation :

- **Visualisation publication :**
  The Geant4 Visualization System - A Multi-Driver Graphics System
  , Allison, J. et al., International Journal of Modeling, Simulation, and Scientific Computing, Vol. 4, Suppl. 1 (2013) 1340001:
  http://www.worldscientific.com/doi/abs/10.1142/S1793962313400011

Getting Help :

- **If problems arise, try browsing the user forum to see whether or not your problem has already been encountered. If it hasn't, you can post it and Geant4 developers will do their best to find a solution. This is also a good place to discuss Geant4 topics in general.** https://cern.ch/geant4-forum
- **Get a look at Geant4 User support pages: *http://cern.ch/geant4/support***

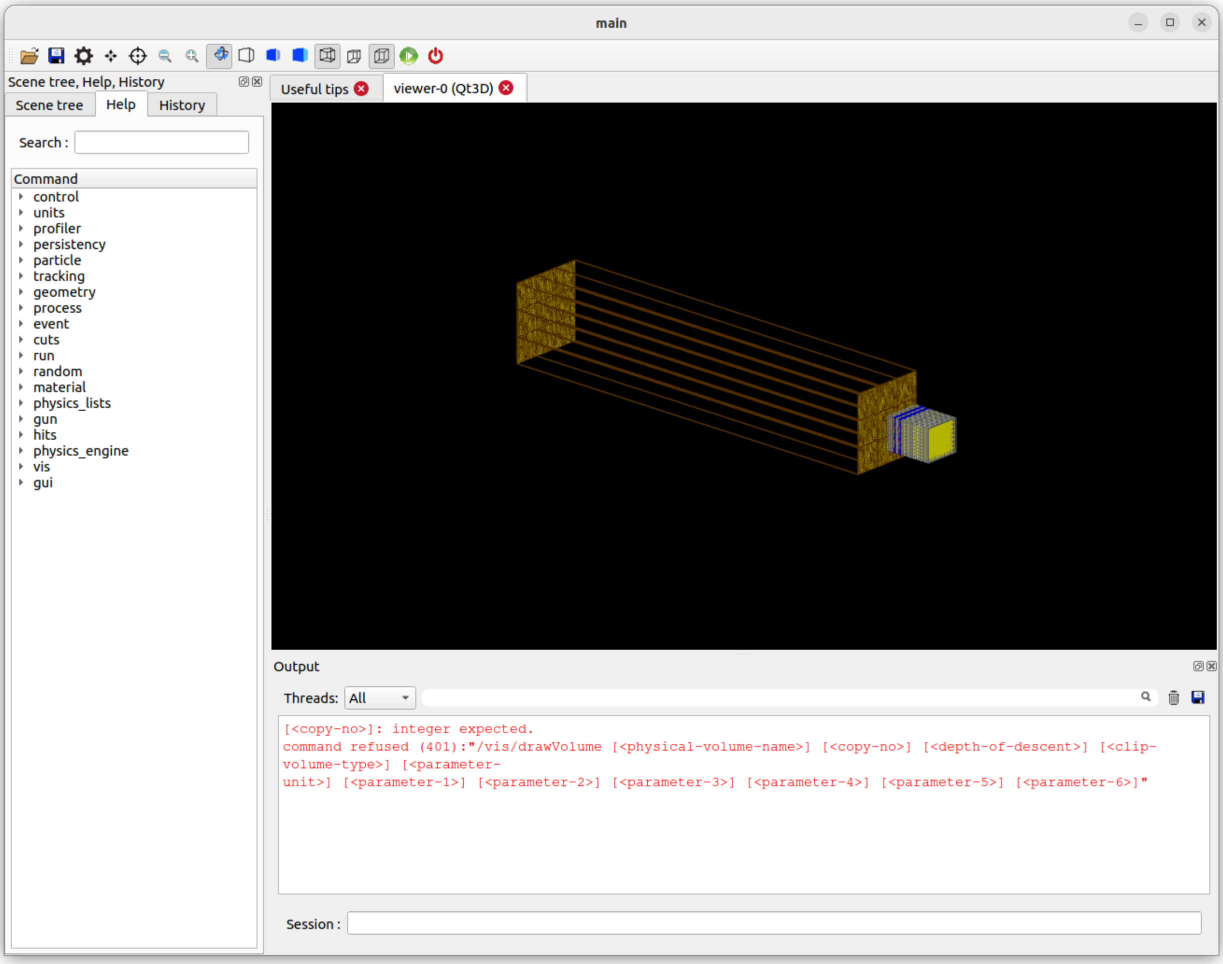
From this page I can setup the view of the calorimeter by implementing the

```
/vis/open Qt3D
```

command into the session searchbar. Qt3D being the gui program used to visualise the setup. Then I follow up with

```
/vis/drawVolume world
```

in the searchbar, which allows a view of the calorimeter setup.



```
[<copy-no>]: integer expected.
command refused (401):"/vis/drawVolume [<physical-volume-name>] [<copy-no>] [<depth-of-descent>] [<clip-
volume-type>] [<parameter-
unit>] [<parameter-1>] [<parameter-2>] [<parameter-3>] [<parameter-4>] [<parameter-5>] [<parameter-6>]"
```

In this case it shows the setup with FoCal-E in front, it has the benefit of allowing us to view the details of the calorimeter, dowm to the individual straws of the modules.

4.2.5 Final notes on the Focal_sim simulation package

It should be noted that this simulation package also has certain methods for ensuring that the simulation matches the actual calorimeter, as mentioned the builder files construct the geometry of the setup and define the properties of the various materials, such as the air surrounding it or the copper in the tubes, or the silicone of the straws. To ensure that these materials match what is in the actual calorimeter, it can also calculate the total weight of the setup for instance, this allows us to compare with the weight of the actual setup, where they hopefully match.

5 Results

5.1 Expected results and their physical interpretations

As an aside before the discussion on the results begins, when it comes to the setup of the FoCal prototypes has a significant impact upon the results, specifically the presence of FoCal-E in front of FoCal-H has a significant impact upon the results that would return from the beamtests, and likewise also has a significant impact on the simulated results, as they will vary depending on whether FoCal-E is simulated in front or not.

The origin of these variations is due to the shower developments within the overall setup. Specifically it depends on the type of shower and when it begins to develop. Broadly speaking there are four main options that a run is expected to fall in under. It can either be: a) an electromagnetic shower starting in FoCal-E, b) a hadronic shower starting in FoCal-E, c) a hadronic shower starting in FoCal-H or d) a muon traversing through the detector setup. These can then have the influences on results as described.

5.1.1 Electromagnetic showers within FoCal-E

The electromagnetic shower case is fairly simple to explain, considering that FoCal-E is an electromagnetic calorimeter, which means that it must necessarily contain the electromagnetic showers within it. Due to this it will need to necessarily absorb all the energy of this shower, as if it didn't there wouldn't be proper shower containment.

Because of this there will be nothing really visible within FoCal-H and you'll get is just noise. It is strictly speaking possible that something might be detected within FoCal-H, but that would in most likelyhood be stray atmospheric muons, or lacking shower containment within FoCal-E.

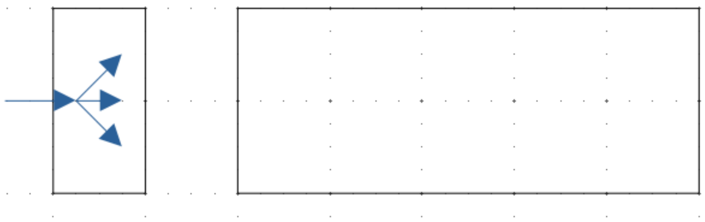


Figure 31: Simplified illustration of an electromagnetic shower contained within FoCal-E, the small box in front is FoCal-E and the larger box behind is FoCal-H. As can be seen the shower remains within FoCal-E.

5.1.2 Muons passing through the calorimeters

This case, like the electromagnetic showers, are not overly interesting for FoCal-H, since they won't result in any useful data for the FoCal-H prototype. Any case of muons will likely just passs straight through both calorimeters, only slightly depositing a small amount of energy in the calorimeters, indeed this will be demonstrated in the simulations, since they have the capacity to identify the particles leaving the calorimeters.

As the muons just travel straight through the calorimeters, no showers will develop, meaning that there will be very little to see, it might give a single blip on the calorimeter, however it might also just not leave any real trail. As a consequence this will also result in an output mainly consisitng of noise, and as such not any useful information.

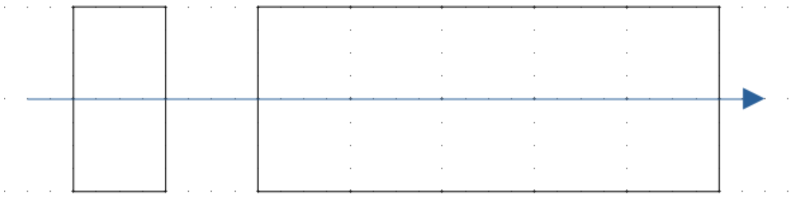


Figure 32: Simplified illustration of a muon passing through both calorimeters, as it passes through it will only lose a little energy to the calorimeter and will not develop any showers.

5.1.3 Hadronic showers starting in FoCal-H

Now reaching the physically interesting cases for FoCal-H, the first of which is a hadronic shower exclusively in FoCal-H, this mainly occurs in runs without FoCal-E in front, but can also occur in cases of particles that only interact strongly, as they may pass through FoCal-E without making any collisions.

In this case the resulting output is quite clear, as will be demonstrated later. Here the energy deposits will be quite clear, with a narrow maximum around the beam energy, as the beam will deposit all of it's energy within FoCal-H, thus giving this clear peak. These cases are useful to see how well FoCal-H contains showers.

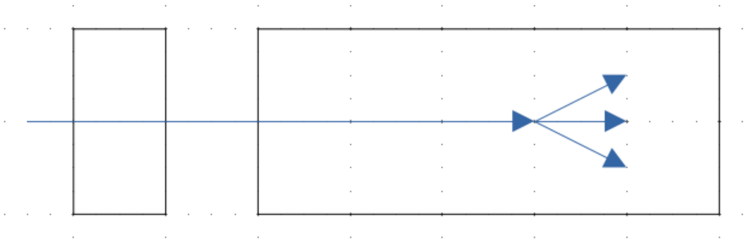


Figure 33: Simplified illustration of a hadronic shower starting in FoCal-H. As can be seen the shower exclusively develops in FoCal-H, ensuring that FoCal-H contains the full energy of the event

5.1.4 Hadronic showers starting in FoCal-E

However, the actual setup will have FoCal-E in front and as such we'll have to take the development of showers inside of FoCal-E into consideration. These showers will have already deposited an amount of their energy insider of FoCal-E before they hit FoCal-H, and as a consequence will have a lower energy when measured by FoCal-H.

This results in a smearing of the energy distribution, as the clear peak around the beam energy will be joined with a second smaller "shoulder" at a lower energy, this shoulder then represents the showers starting in FoCal-E as they come in with lower energy.

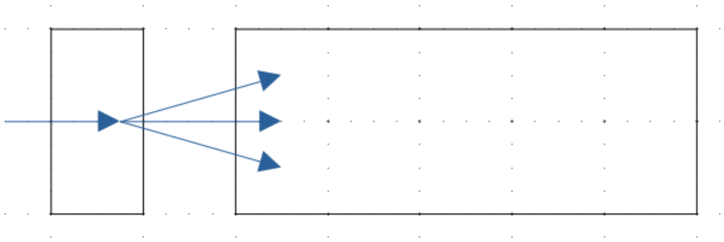
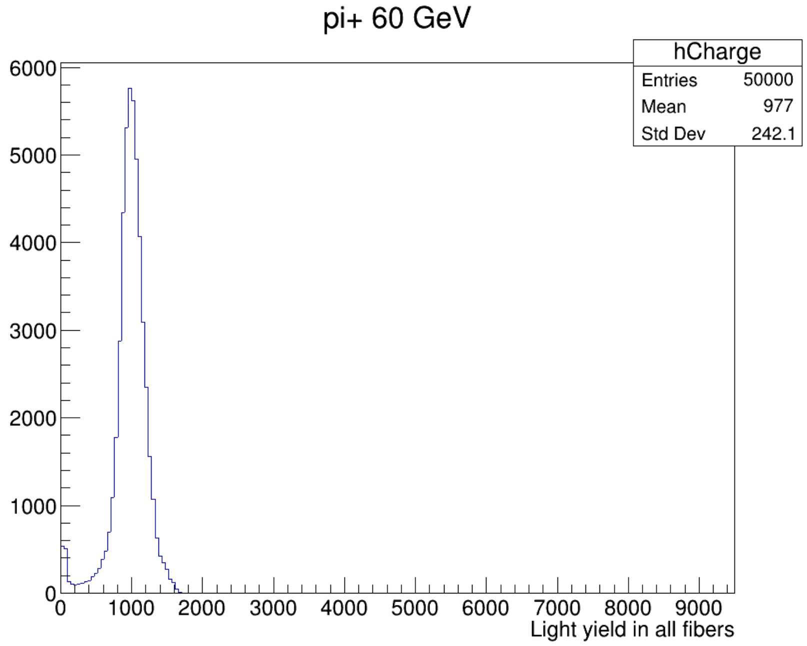


Figure 34: Simplified illustration of a hadronic shower starting in FoCal-E. As can be observed, the earlier shower start results in the shower already being in progress by the time it enters FoCal-H, which affects the results and measured energies in FoCal-H.

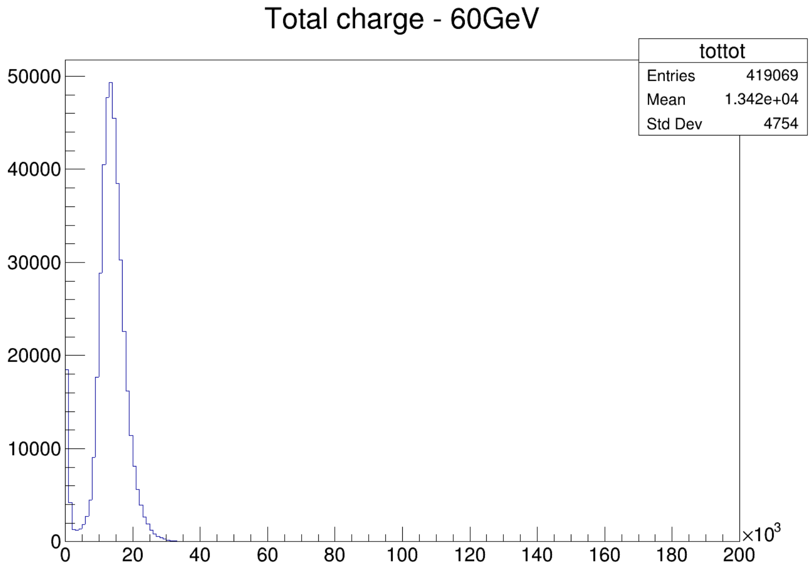
5.2 Early comparisons between beamtest results and simulations

Early simulations were done using the Testbeam simulation package in order to get energy distributions at various energies with and without FoCal-E in front, these simulations were mainly used to get familiar with the simulation package and to get baseline distributions to expect from various energies ranging from 60-200 GeVs. Similar distributions were also acquired from the actual testbeam data and viewed through ROOT.

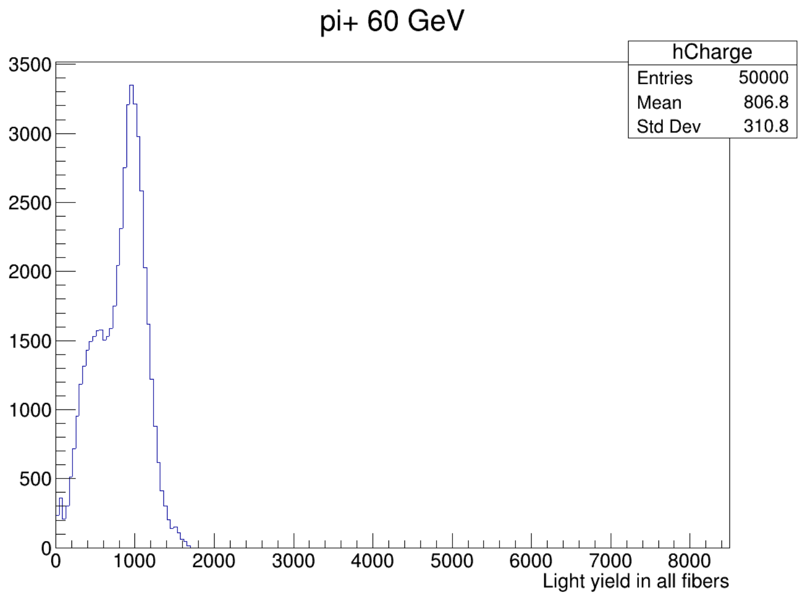
The specific energies compared for these early comparison runs were 60, 100 and 200 with and without FoCal-E in front.



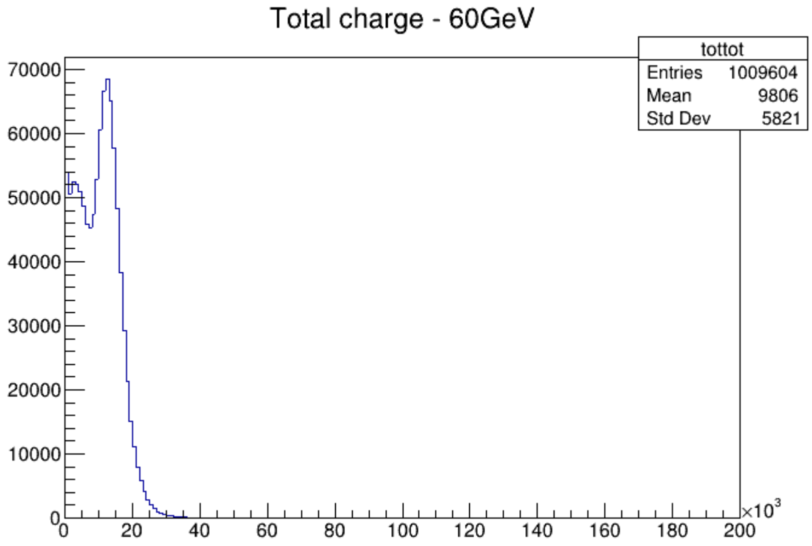
(a) Initial simulations runs done at 60 GeV for 50k events without FoCal-E in front.



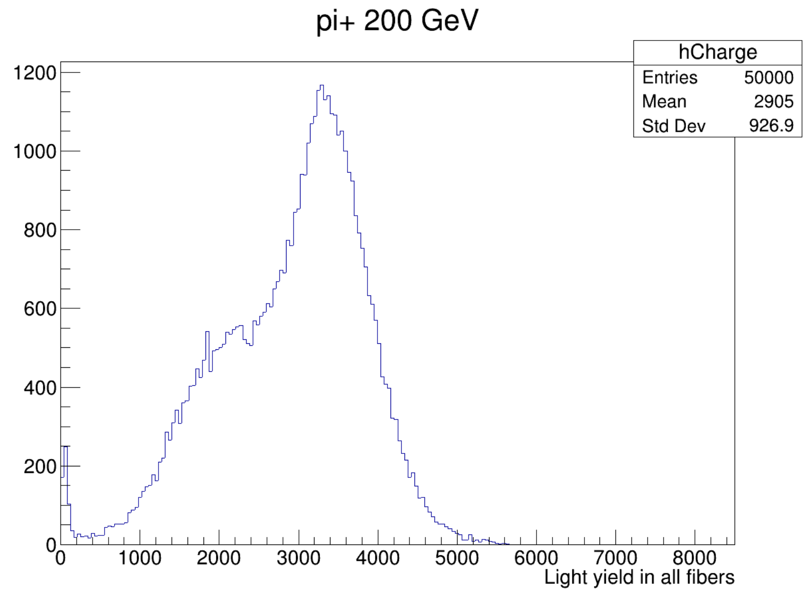
(b) Initial distribution of data taken from FoCal-H beamtest without FoCal-E in front and at 60 GeV



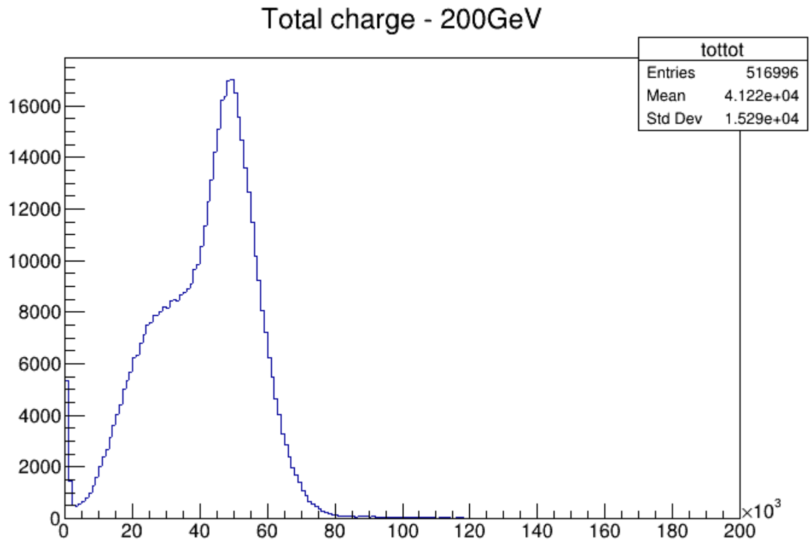
(a) Initial simulations runs done at 60 GeV for 50k events with FoCal-E in front.



(b) Initial distribution of data taken from FoCal-H beamtest with FoCal-E in front and at 60 GeV



(a) Initial simulations runs done at 200 GeV for 50k events with FoCal-E in front.

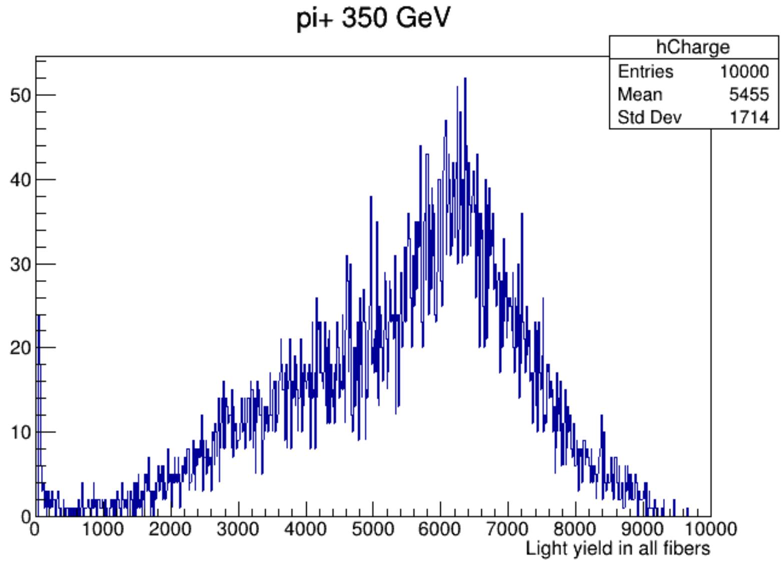


(b) Initial distribution of data taken from FoCal-H beamtest with FoCal-E in front and at 200 GeV

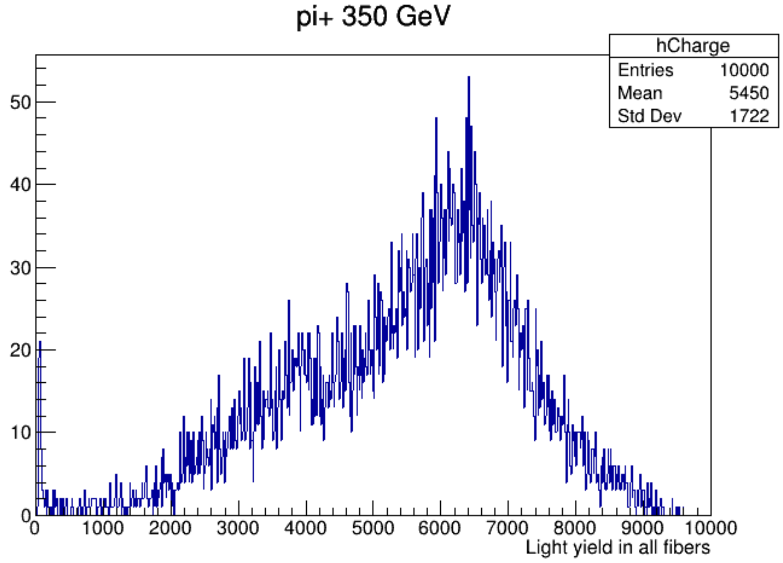
The objective for these was to show initial comparisons between the simulations and testbeam data, seeing how they compared to actual results from beamtests, of course as can be seen the scale that the compared graphs show are different between beamtest data and the simulations, fortunately since we know what the beam energy of the given simulations and experiments were, we then know that their peaks correspond to the respective beam energies of 60 GevV and 200 GeV, further results ranging from 100 GeV to 200 GeV without FoCal-E are included in the appendix.

5.3 The 350GeV simulations

Now the main results of this project are the various 350 GeV simulations that have been made, and have allowed for the comparisons between various versions of geant4 and even between updates to the Testbeamsimulation package, which came about in order to achieve the goal of acquiring longitudinal distributions of the energy within FoCal-H. This additionally showcased the simulated energy distributions for higher energy than the previous runs.



(a) Energy distribution from a testbeamsimulation run with geant4 v.10.7.3 at 350GeV with FoCal-E in front.



(b) Similar distribution done for 350GeV with FoCal-E in front, this time run using the simulation package using geant4 v11.1.1

The initial reasoning for these comparisons was that due to updating the software between the previously used v10.7.3 and the more recent v11.1.1 necessitated comparison simulations to make any differences between the back-end clear.

This did however also help to illustrate another point, the various shower developments and their influences on results. As was observed in the initial comparisons, a run without FoCal-E in front has a noticeably sharper peak around the beam energy, which is due to the fact that all the hadronic showers begin and end within FoCal-H, whereas with FoCal-E in front we also have to take the hadronic showers that begin in FoCal-E into account as well, and the electromagnetic showers that begin and end in FoCal-E, resulting in the shoulder on the distribution.

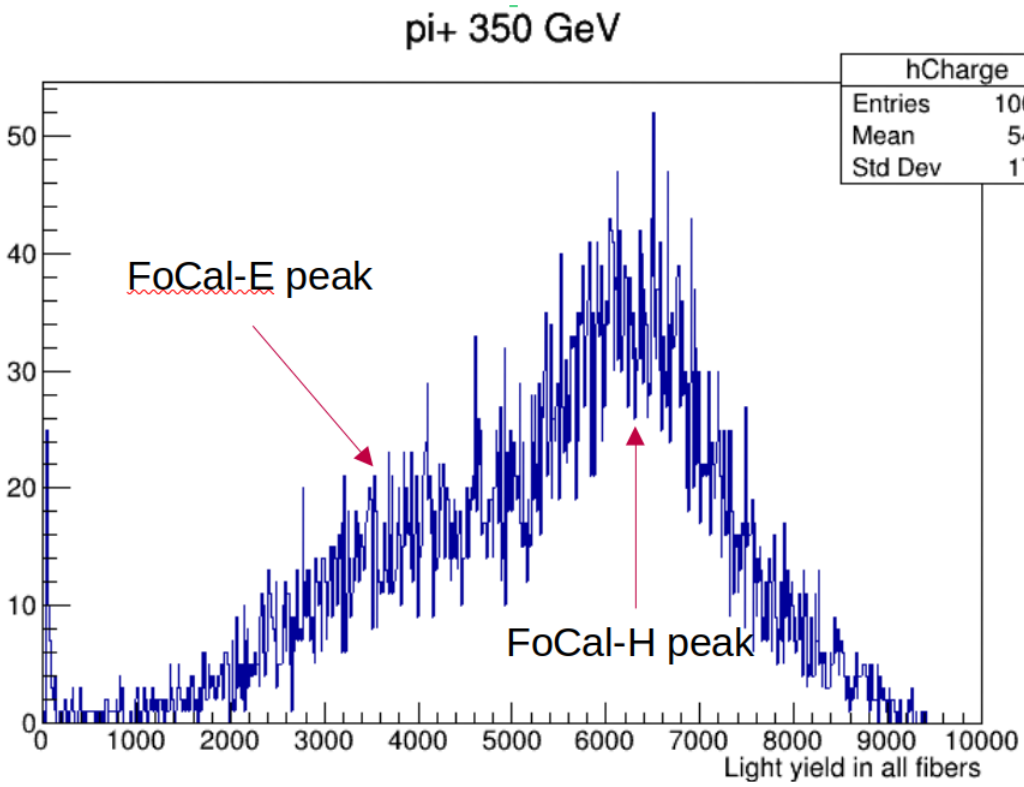
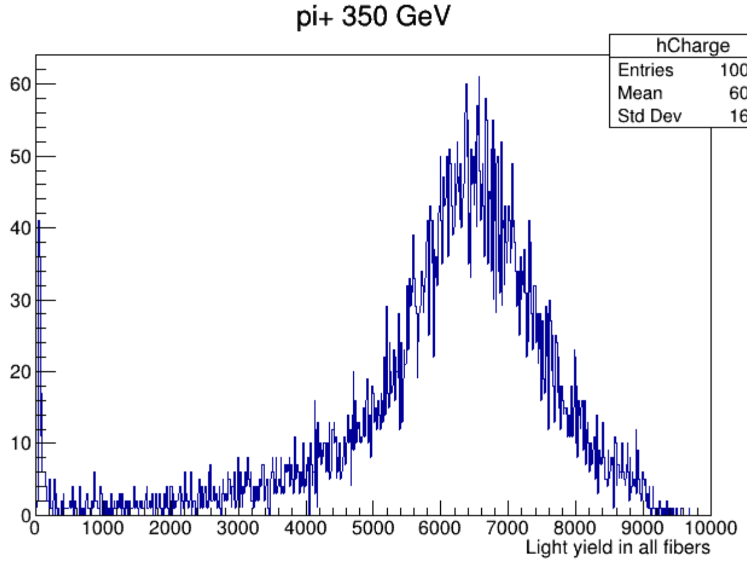
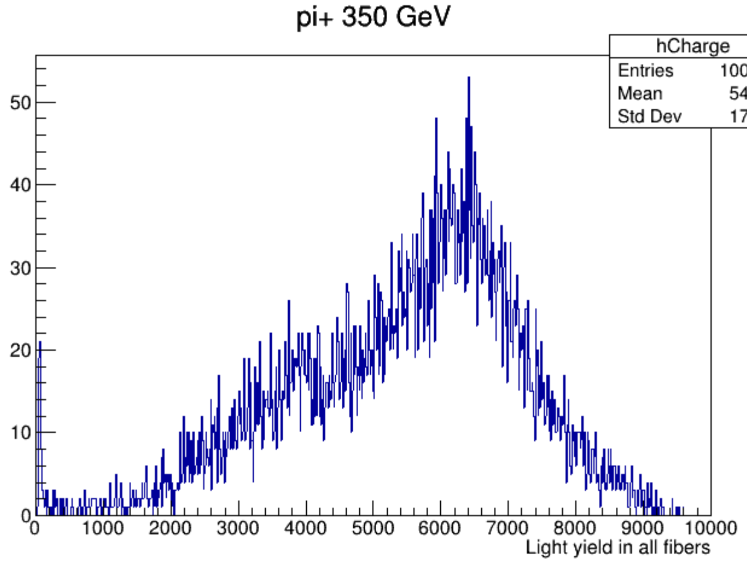


Figure 39: Demonstration of the various parts of the shower development in the setup of FoCal-E + FoCal-H, the FoCal-E peaks refer to the shoulder, which occurs due to the showers that start within FoCal-E and as such have a lower energy upon entering FoCal-H, whereas the FoCal-H peak refers to the fraction of showers that begin and end in FoCal-H

A comparison between a run in the v11.1.1 implementation done with FoCal-E and without is illustrated in these results.

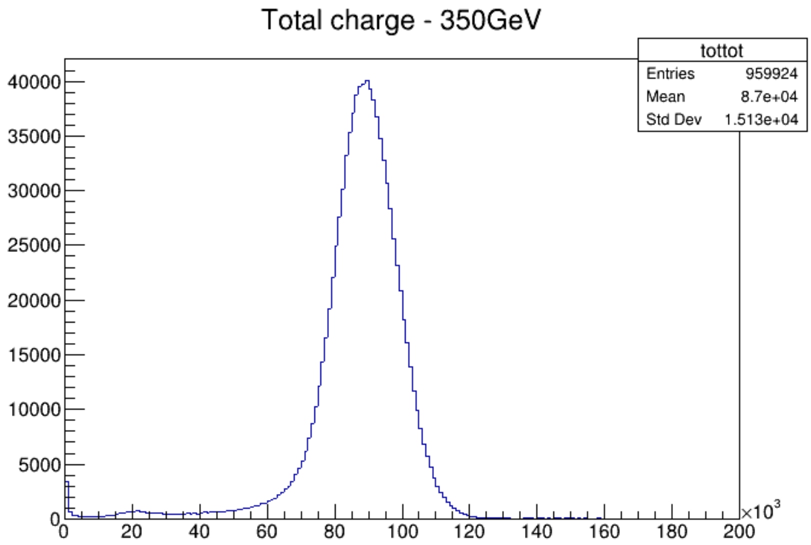


(a) Simulation run done with the testbeamsimulation package at 350GeV for 10k events without FoCal-E in front

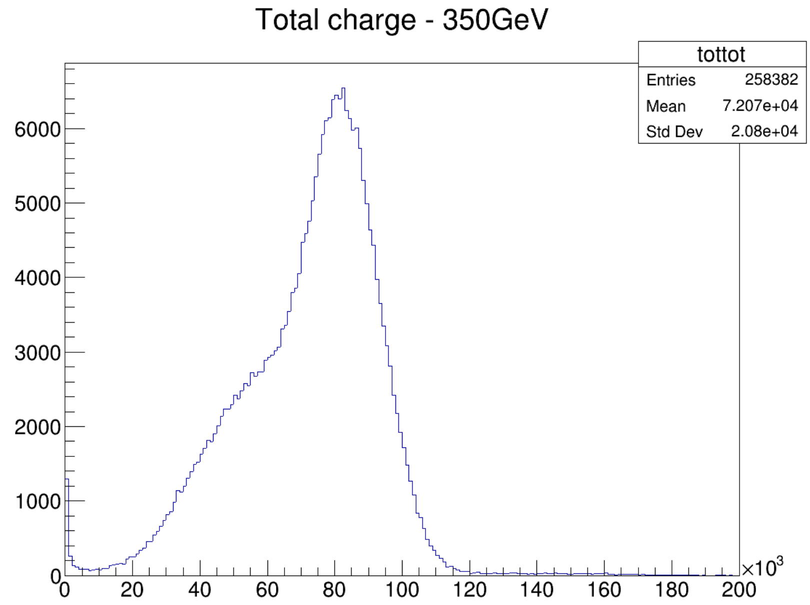


(b) Similar distribution done for 350GeV with FoCal-E in front.

Now from this one would perhaps like to ask how this would compare to the actual data from FoCal-H, particularly considering how weakly defined the shoulder is in the simulations compared to the lower energy cases. Well the comparable data from the beamtests looks like the following:



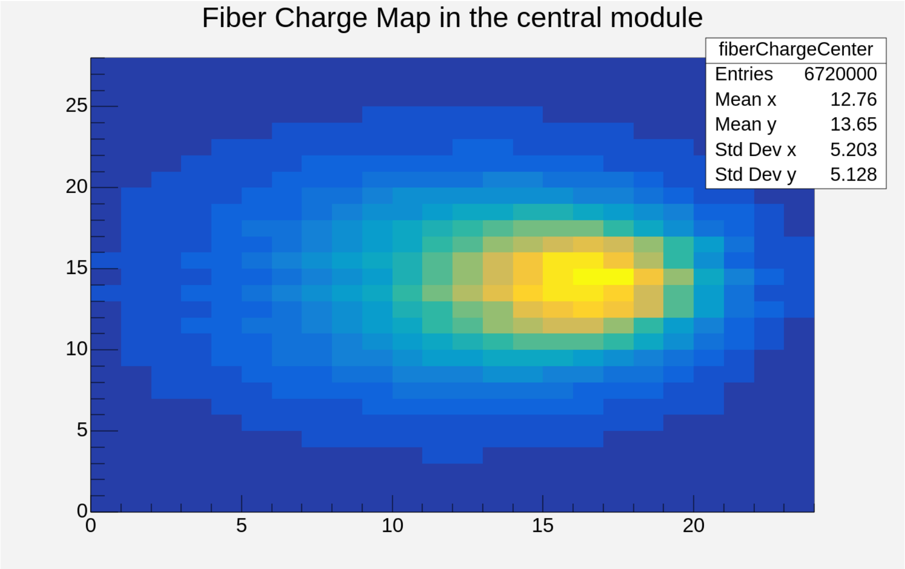
(a) Results from the beamtest run done without FoCal-E in front at 350 GeV beam energy.



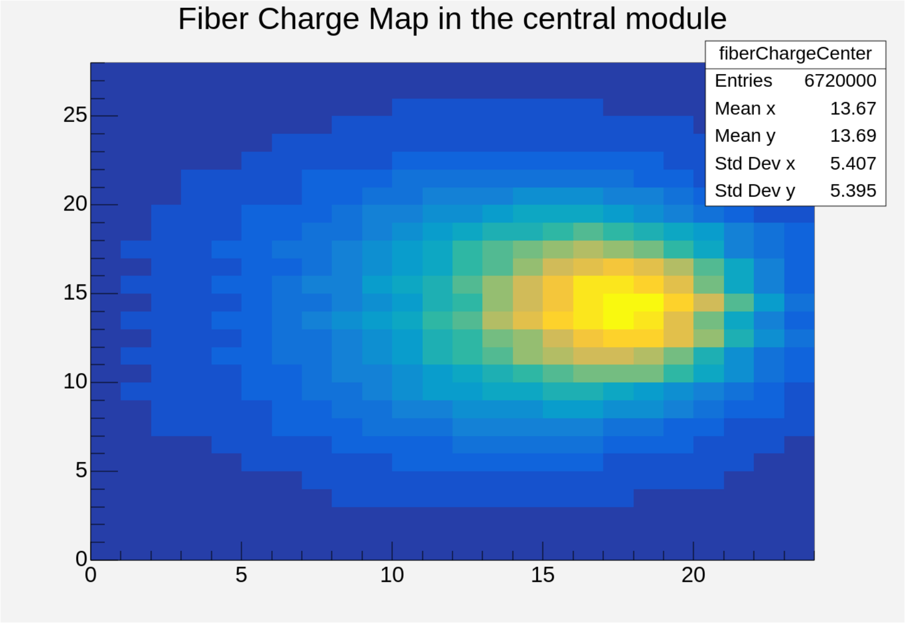
(b) Similar distribution done for 350GeV with FoCal-E in front.

As can be observed here, the weak shoulder seems to be in agreement with the experimental data, and could arguably be explained by the higher energy runs resulting in larger instability in possible energies of the showers, as it allows showers to start earlier and still enter FoCal-H with higher energies, and as such it goes some way to explain the wider peaks overall at higher energies.

For the sake of consistency transverse heatmaps were also made in the simulations to demonstrate the energy deposits within various channels, it also helped to check whether the simulation was correctly configured to be at a predefined angle in comparison to the beam.



(a) Heatmap of the central module within FoCal-H in a run without FoCal-E, demonstrating the angle of the beam in the simulation, meant to replicate the circumstances of the November 2022 beamtest.



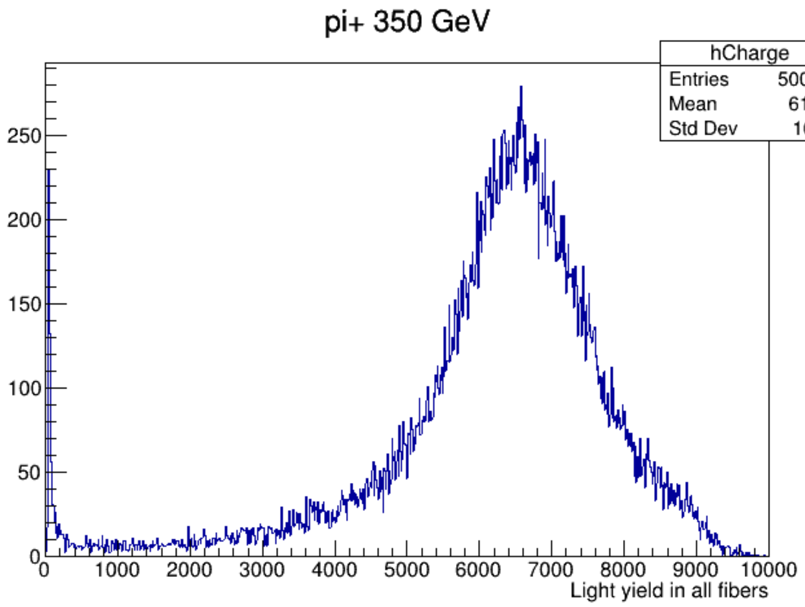
(b) Heatmap demonstrating the same for a simulated run at the same beam energy but this time with FoCal-E in front

As can be observed, while the two heatmaps are quite similar, there is however a difference in the standard deviations along the x and y axis in comparison to the mean point. In the case without FoCal-E this deviation is 5.15-5.2, whereas in the case of a simulation run with FoCal-E in front this deviation is larger, at 5.39-5.4 instead. To ensure this deviation isn't just a random case another simulation was done at the same energy with the same setups, and a similar difference can be seen there. These second heatmaps can be seen in the appendix. In fact heatmaps like these were also done for the two distinct 350GeV runs with FoCal-E done with geant4 v10.7.3, which can also be seen in the appendix.

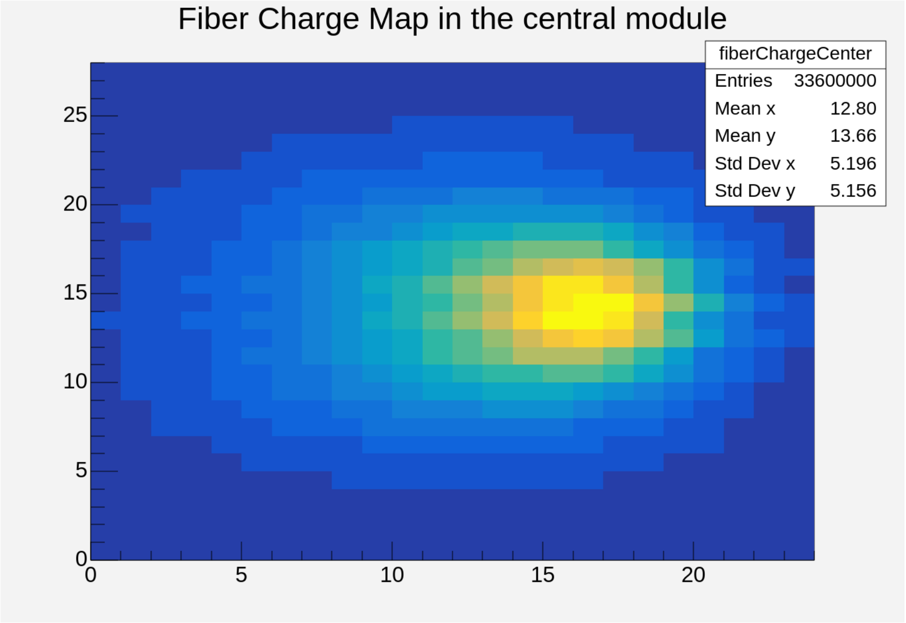
This all lead to an update for the simulation package itself, leading to further comparisons. Since the main focus was on the performance of FoCal-H without FoCal-E in front, these comparisons would be done without FoCal-E.

An important note: As it can be observed in the simulated plots for the various energies the x axis lists itself as "lightyield" which might make one think that it corresponds to the number of photons that hit the detector, however this is not the case. Lightyield, somewhat unintuitively, refers to the charge that the detectors would recieve, not the amount of photons hitting the detector.

And to clear out any last possible confusion, for these plots the y axis refers to the number of events with the given charge output.



(a) A 50 thousand event run done with the updated simulation package at 350 GeV beam energy and without FoCal-E in front.



(b) Heatmap of the central module from this particular run. Illustrating the correct angle and beamwidth.

Similarly to the previous version runs we can see the single peak with no shoulder as one would expect, we also see that the mean is located slightly higher compared to the earlier version, though this is in all likelihood just a random variance and not indicative of major changes within the simulation package itself. An additional 50k event run was done, the results of which will be in the appendix, however they showed similar outcomes as the above.

5.4 Energy distribution within FoCal-H

The main objective for this section of the project was to use the simulation software to draw a distribution of the energy losses within the detector along the Z-axis. We know in theory that the distribution should peak at around 1 interaction length within the given material, and shouldn't be really dependant on the energy of the particular event. For this purpose I have utilized the Focal_sim package developed by Christian Holm Christensen with some slight modifications on my own part for a particular part of the results.

5.4.1 350GeV - VMC implementation

The first results to demonstrate here are the results acquired through the Virtual Monte Carlo implementation of the simulation software. As a point of note all results within this report, with previous and upcoming methods utilize geant4, so despite a geant3 version of this implementation being available, it was not used in favour of the geant4 method.

The first important thing one would wish to look at is a similar plot to those seen above:

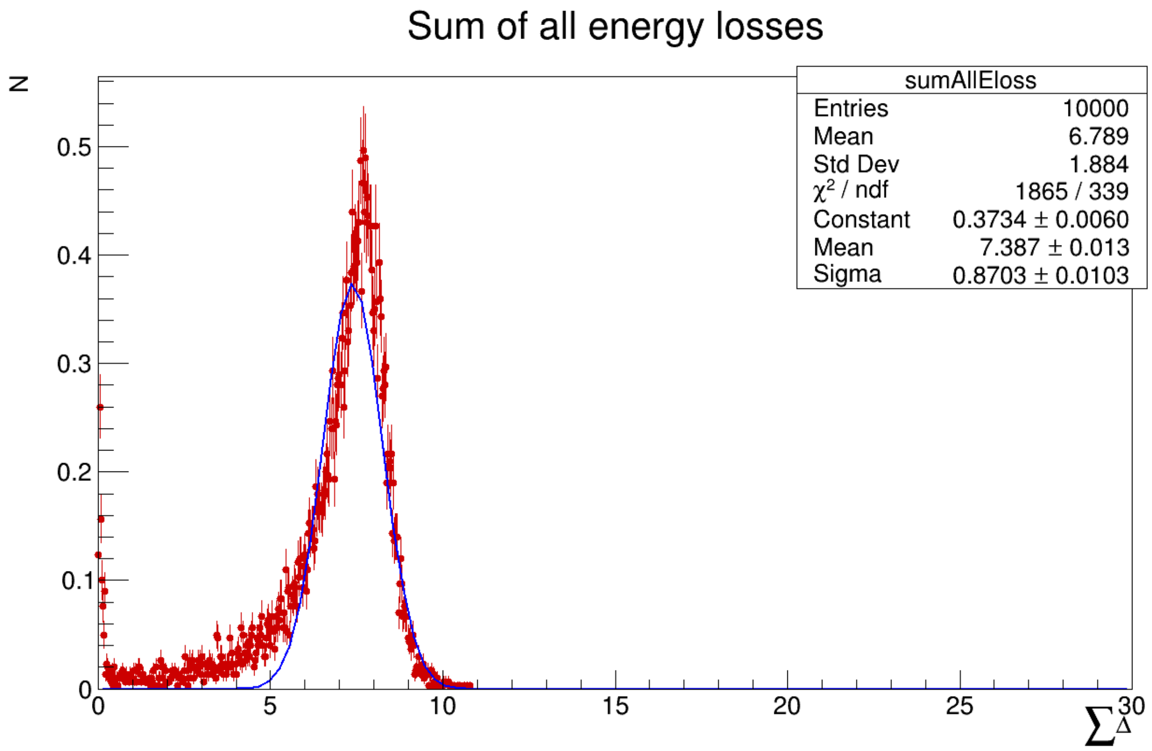


Figure 44: Illustration of the number of events at a given sum energyloss within the detector. Fit to a gaussian function shown in blue.

It should be noted, that whereas the previous simulation packages simulated the charge within the detectors for these plots, this instead takes the sum energyloss over the course of a whole event and compares it, as can also be seen it is normalized along the Y-axis with N = 1 representing all events.

As we would expect from a run without FoCal-E in front of FoCal-H, the energy is distributed around a single sharp peak, representing the showers that start and end in FoCal-H, however it is not all that we can get from this, one might for instance be interested in noting what leaves the calorimeter, now it should ideal be almost just muons, but is that actually the case? Fortunately the simulation package has the FocalHFlux classes, which refers to a "box" around the simulated detector, that has the specific purpose of noting what enters and leaves it. And the simulation package has specific scripts for plotting this:

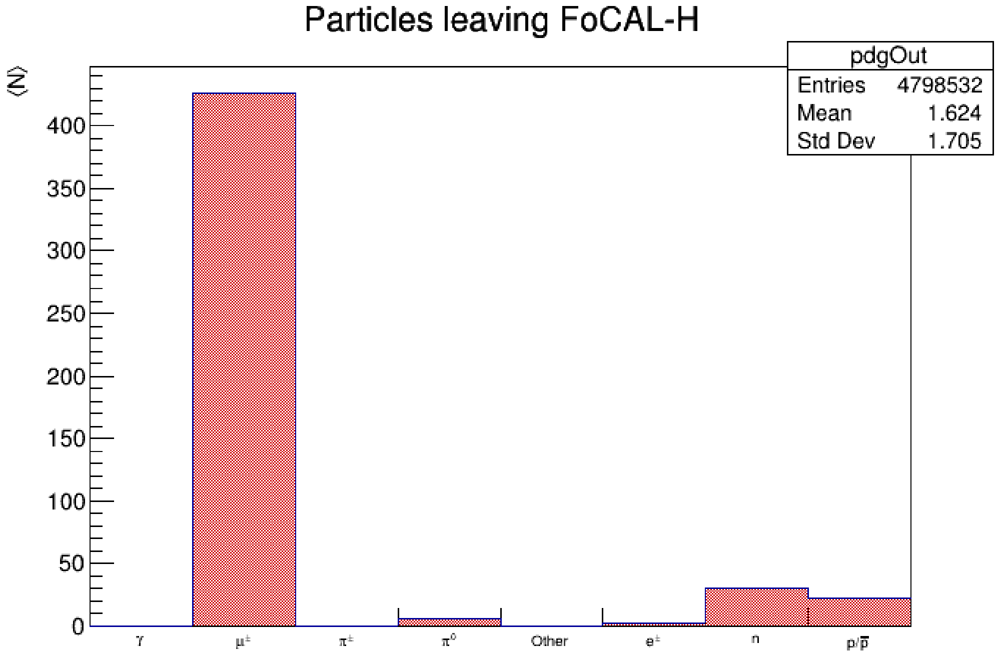


Figure 45: Plot showcasing the particle types leaving FoCal-H and the relative amounts

As can be observed the vast majority of particles leaving the FoCal-H detector are muons and neutrons, muons fit with what was expected as mentioned earlier, and neutrons can be explained due to be a chargeless particle, therefore there is no electromagnetic shower development from neutrons, therefore all their energyloss has to come from collisions, which could theoretically occur, given the finite length of the calorimeter.

However one might also wish to see further visualization of the data, including simulated ADC counts, to see the risk of saturation, as well as viewing the beamwidth and angle.

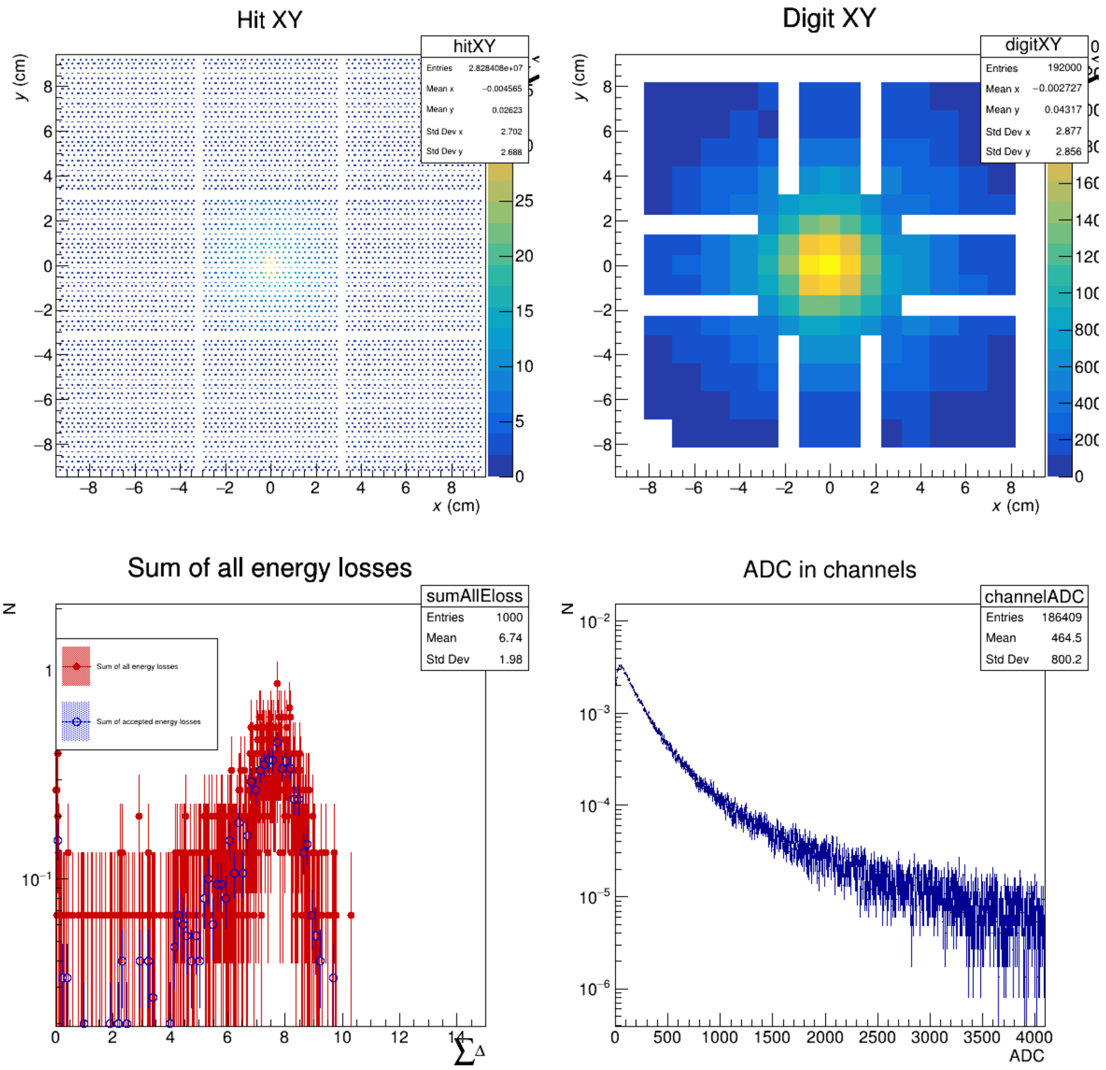


Figure 46: Numerous plots showcasing various data. The top two plots showcase the energy deposited in heatmaps. Top left: specifically showcases the hits within each straw of the simulated detector,and can be tough to read as a consequence. Top Right: the same heatmap but mapped to channels more similar to what we would actually see from the beamtests. Bottom left: illustration of the sum energyloss like the sum of all energylosses showed two figures up. Bottom right is an illustration of the ADC counts for the number of events within the simulation. Note that both bottom figures have a logarithmic scaling along the y-axis.

Now I mentioned earlier that the sum of all energy losses distribution corresponded to the expected peak at the beam energy, but in order to confirm this it would be prudent to compare data. If we were to assume that the peaks correspond to the beam energies, then it would make sense to assume that the ratios between the various components of these graphs match the ratios between beam energies. For instance the mean values in these plots should have a similar ration to that of the beam energies, so by using multiple runs at varying energies I can put this claim to the test. Thus for a baseline we establish the ratios between various beam energies, for this I will compare the 350GeV run with a 200GeV run and a 1 TeV run:

$$\frac{350GeV}{200GeV} = 1.75 \quad \text{and} \quad \frac{1000GeV}{350GeV} = 2.86 \quad \text{and} \quad \frac{1000GeV}{200GeV} = 5$$

This can then be compared to the ratios between the means of gaussian fits made to the energy sum distributions from the simulations, if the peaks within the simulations indeed correspond to the beam energies, then these ratios should be equal to the ratios of beam energies.

Now the means from the various distributions are as follows:

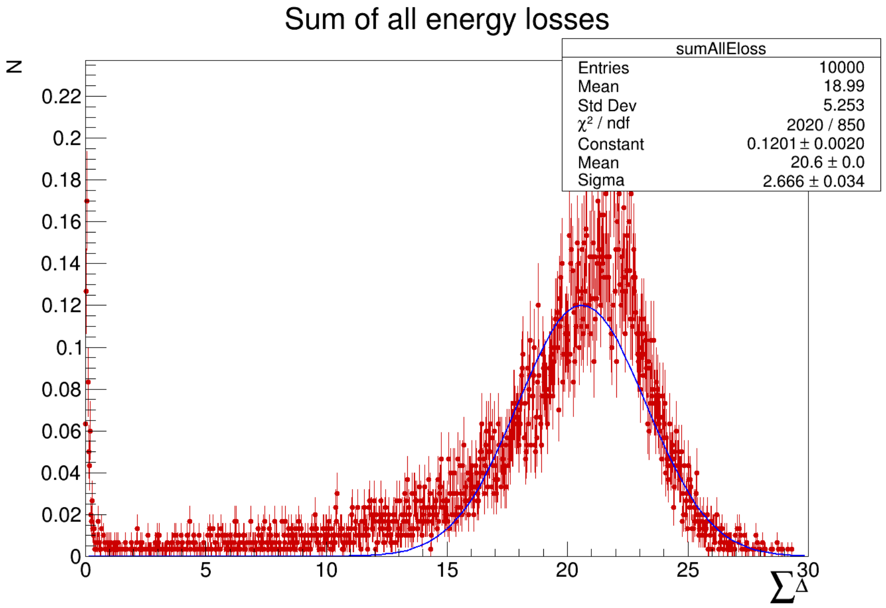
1. Mean of fit: 350GeV distribution = 7.387
2. Mean of fit: 200GeV distribution = 4.287
3. Mean of fit: 1 TeV distribution = 20.6

Taking the ratios between these 3 means we get the following results:

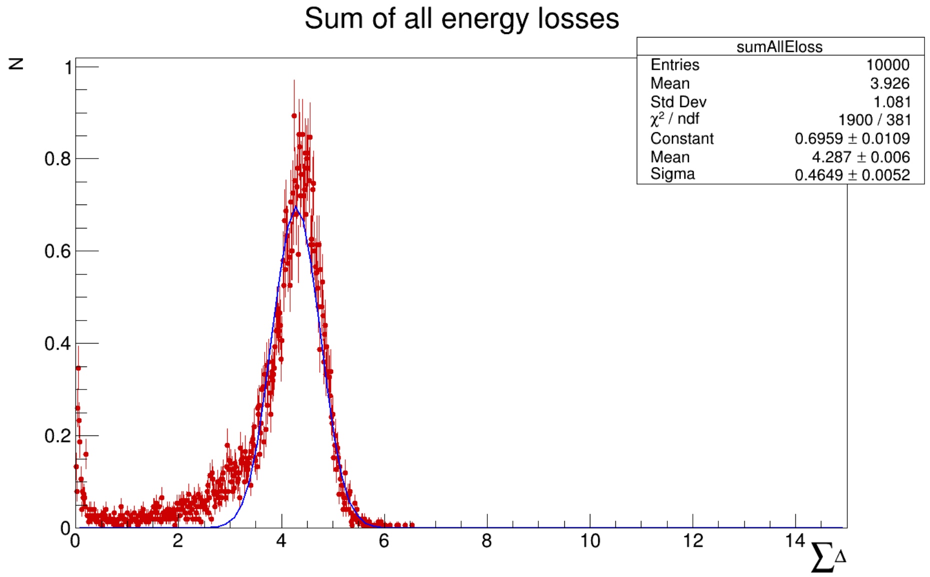
$$\frac{7.387}{4.287} = 1.72 \quad \text{and} \quad \frac{20.6}{7.387} = 2.79 \quad \text{and} \quad \frac{20.6}{4.287} = 4,81$$

As can be seen here that to within about 0.2 these match up with what one would expect compared to beam energies, and as a consequence these energyloss distributions are a reasonable tool to visualize the consequnces of these shower developments within the calorimeter. Finally it should be noted that the gaussian fit isn't ideal for describing these distributions, however it does give a decent idea of the shape of everything.

As a final note on the graphs showcased so far, in the collection of four graphs, it also gives the sum of all accepted energy losses alongside the sum of energy losses, the difference between these two is that the accepted energy losses are acquired by simulating the integration time of the SiPMs within the detectors, so any hits within an event taking place after a certain amount of time are discredited.



(a) Results from a 1 TeV run with 10 thousand events and without FoCal-E in front. Fit to a Gaussian function shown in blue.



(b) Results from a 200 GeV run with 10 thousand events and without FoCal-E in front. Fit to a Gaussian function shown in blue

For the further graphs showcasing among other things the ADC counts of the 1 TeV and 200 GeV runs see the appendix.

Finally for the reason why these simulations were run in the first place: the longtitudinal energy distribution. As stated earlier the simulation records the position and energy loss for every hit during an event, which due to shower generation, there are a lot of. However it does record the exact positions in the X, Y and Z axis for every hit, and the simulation package comes with an analysis script that specifically plots this:

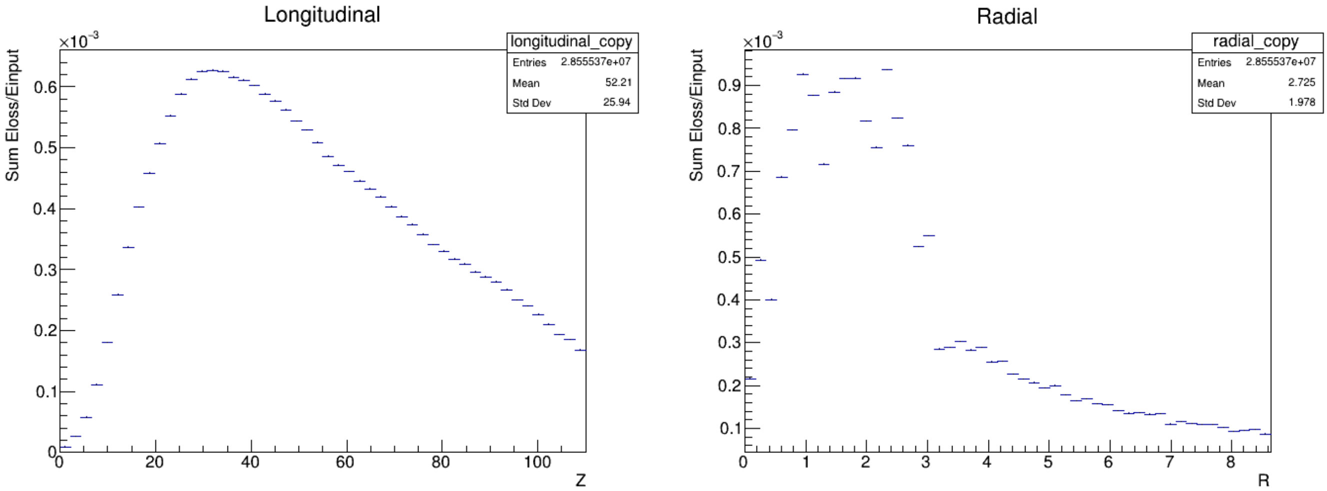
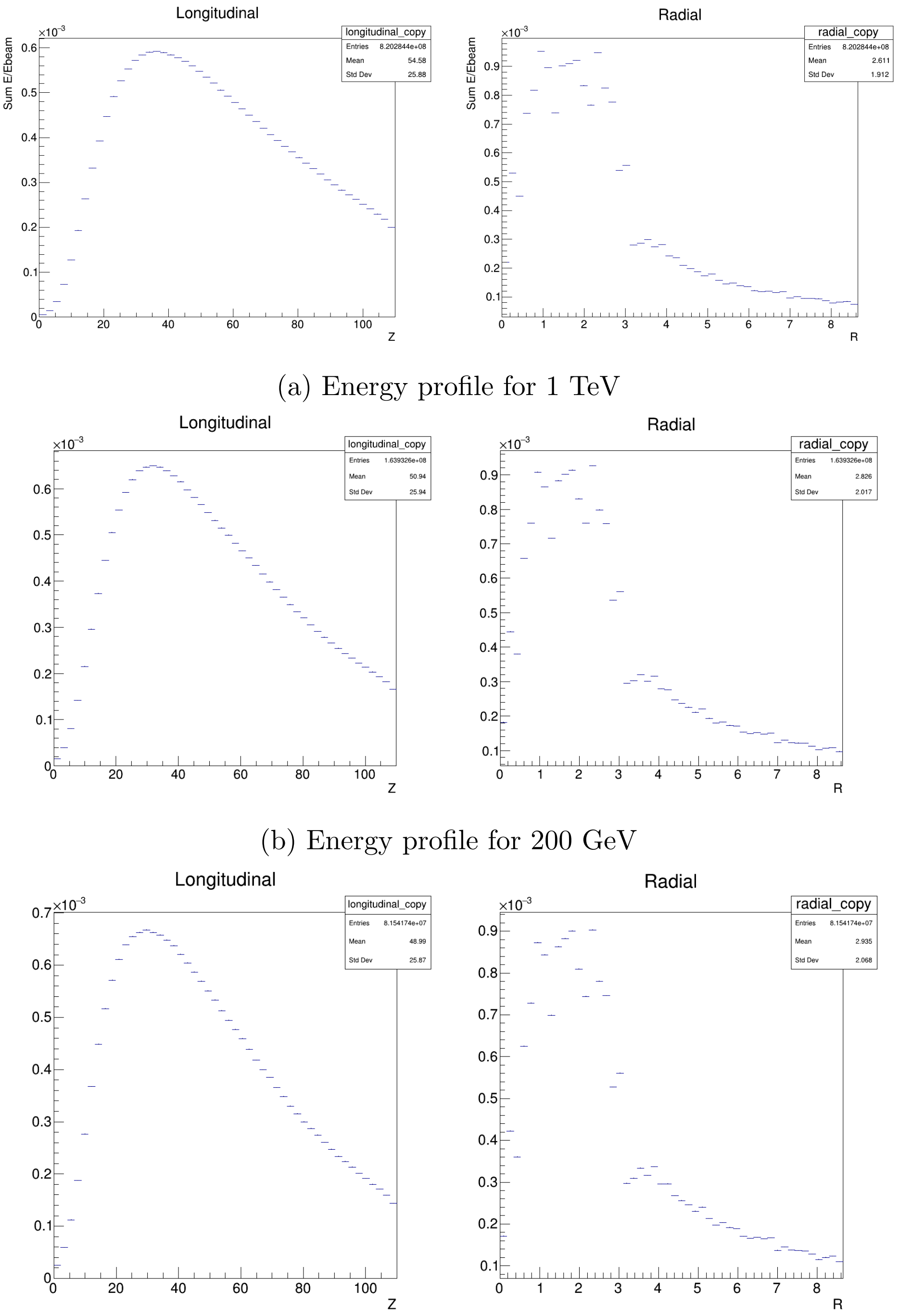


Figure 48: The energy loss profiles for the 350GeV simulation, illustrating the distribution of energy deposits over the Z axis, and the radial distribution of these energy losses.

Now from this a curious quirk of the vmc implementation shows itself, the radial distribution seemingly fluctuates in the center of the calorimeter, however as will be seen later this appears to be a specific quirk of the vmc implementation and not a physical phenomenon. Despite this quirk the profiling still showcases the distribution of energy deposits along both the longitudinal axis and the radial axis. Most notably as one would expect the majority of the energy deposited along the radial axis is in the middle of calorimeter, which makes sense as thats the location of the beam, particularly since for this simulation it is not angled and fires straight into the calorimeter. As for the longitudinal distribution we can see that the majority of the energy deposits occur around the 1-2 interaction length point in the calorimeter, which fits with what one would expect from a hadronic shower.

5.4.2 Other energies - VMC

Other simulations were run, specifically at 100, 200 and 1 TeV beam energies, just to compare the profiles, since we would expect no significant change in the longitudinal distribution across the varying energies.



(a) Energy profile for 1 TeV

(b) Energy profile for 200 GeV

(c) Energy profile for 100 GeV

As can be seen from all these 3 profiles, the energy distribution remains consistent, the majority of the energy is deposited within the 1-2 interaction length section of the calorimeter, and the majority of energy is deposited in the center of the radial axis.

5.4.3 Energy distribution of various parts of the calorimeter - VMC

This all begs the further question: How does the radial distribution change the further you go into the calorimeter? We would expect it to remain heavily focused in the center in the early parts of the calorimeter, however the further in we go, the more equally distributed it would become. Fortunately this could be showcased with a minor change to the profiler script. By simply changing it to demand a certain range of Z in order to fill into the radial distribution, it is possible to leave out anything beyond a specific range of Z.

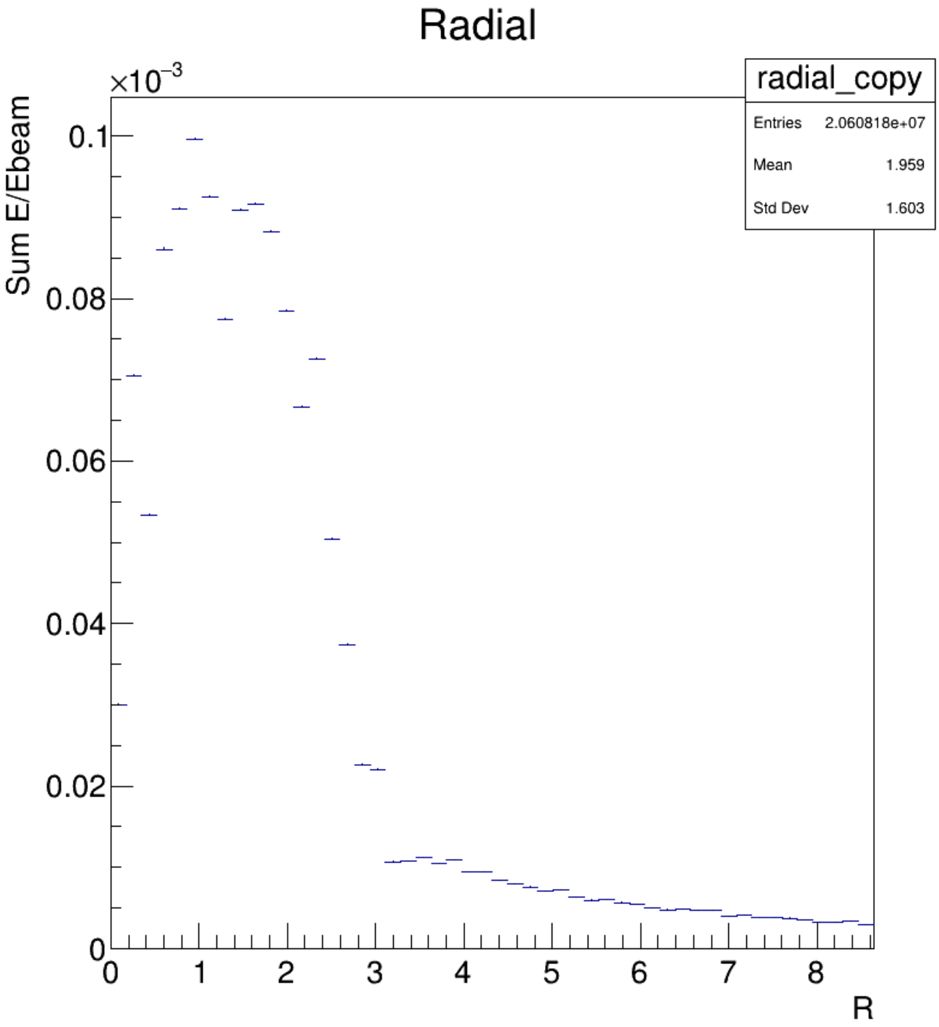
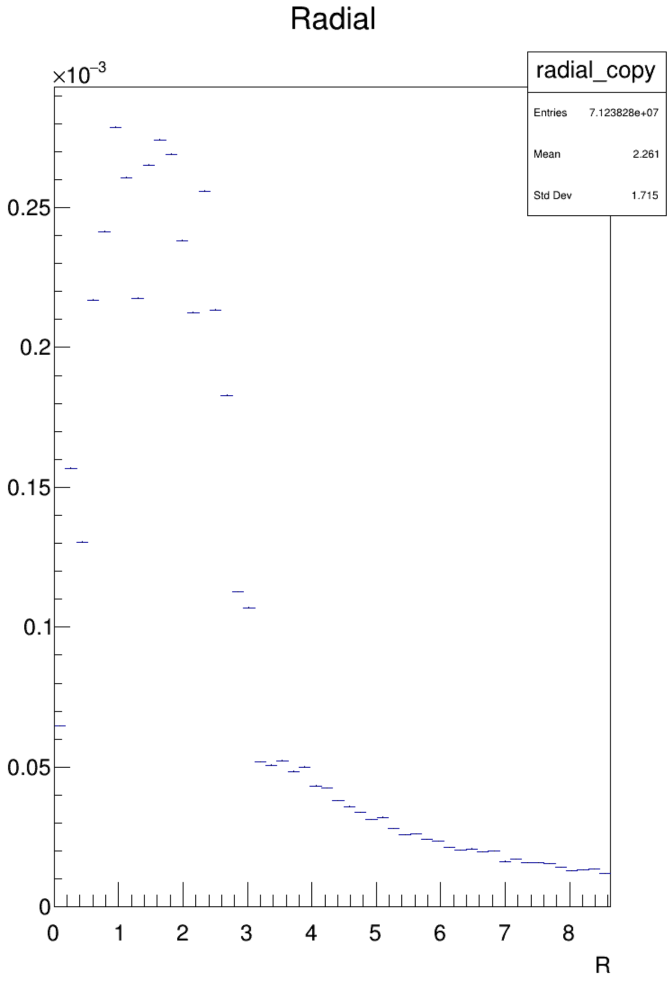
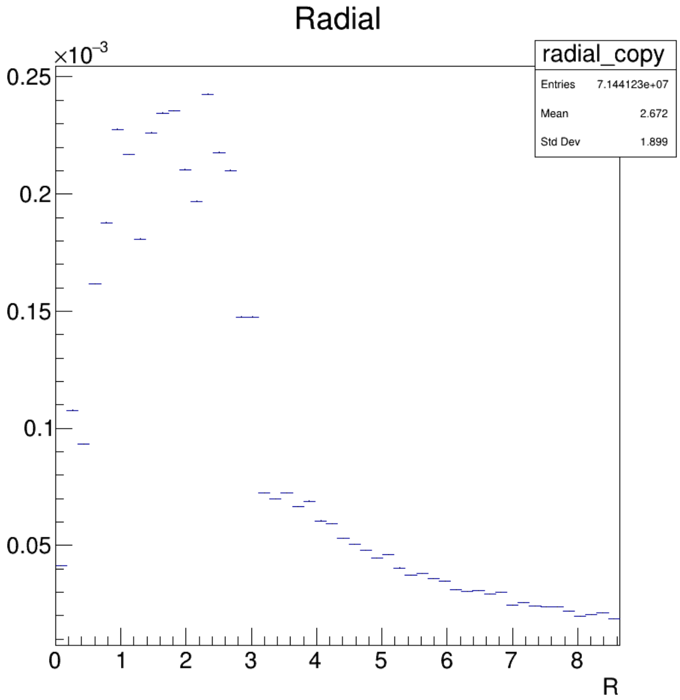


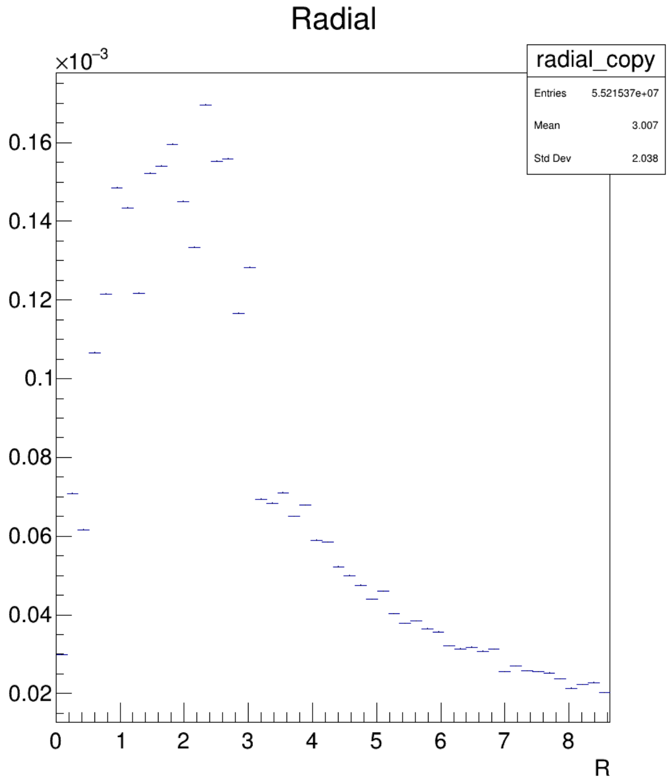
Figure 50: Energy profile for the first interaction length: 0-18 cm



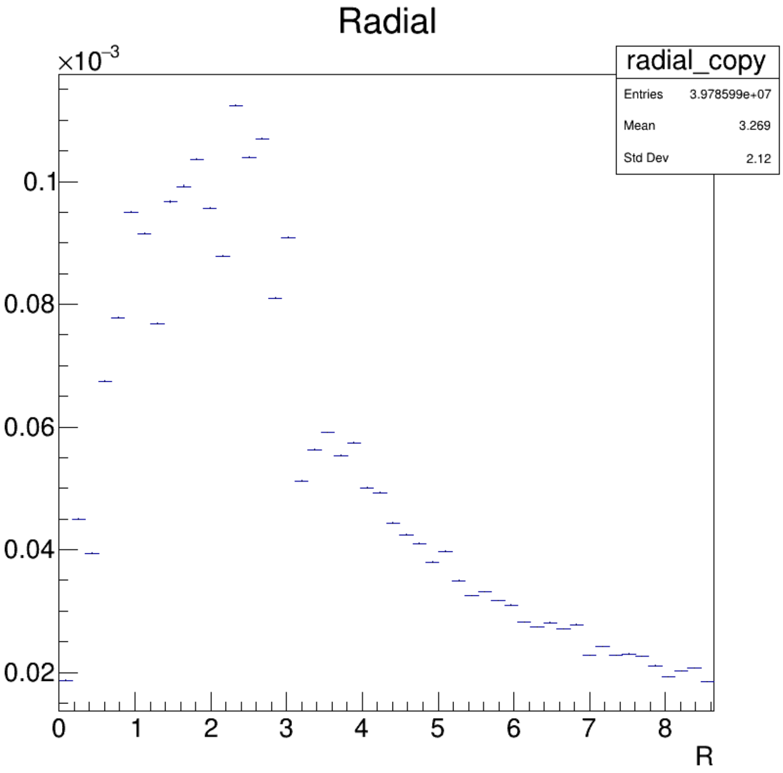
(a) Energy profile for the first interaction length: 18-36 cm



(b) Energy profile for the third interaction length: 36-54 cm



(a) Energy profile for the fourth interaction length: 54-72 cm



(b) Energy profile for the fifth interaction length: 72-90 cm

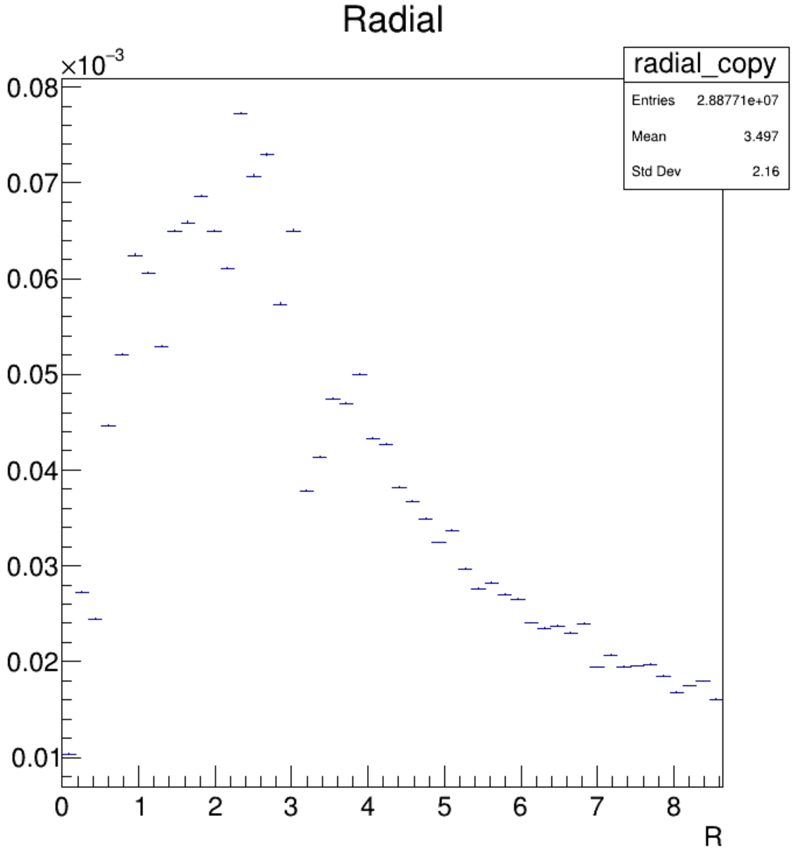


Figure 53: Energy profile for the remainder of the calorimeter: 90-110 cm

From these profiles we can see that indeed, the vast majority of the energy deposited within the first interaction length is indeed deposited in the center of the calorimeter, as the edges of the radius see almost no energy deposits. And as we move further along the calorimeter this angle between the center and the edge becomes less and less steep, until it in the end looks like a perfectly smooth curve.

Now regarding the axis on the profiles, they use the fractional energy deposits along the y-axis and then of course the z and radial distance in cm along the x-axis. The fractional energy deposits are defined as the energy lost within the particular hit, divided by the energy of the beam itself, in this case: 350 GeV.

Finally for this part, it would be interesting to perhaps view a single event, in order to see the visualization of this event profile and what it actually looks like, and for this we have the interactive format of the vmc implementation to showcase this:

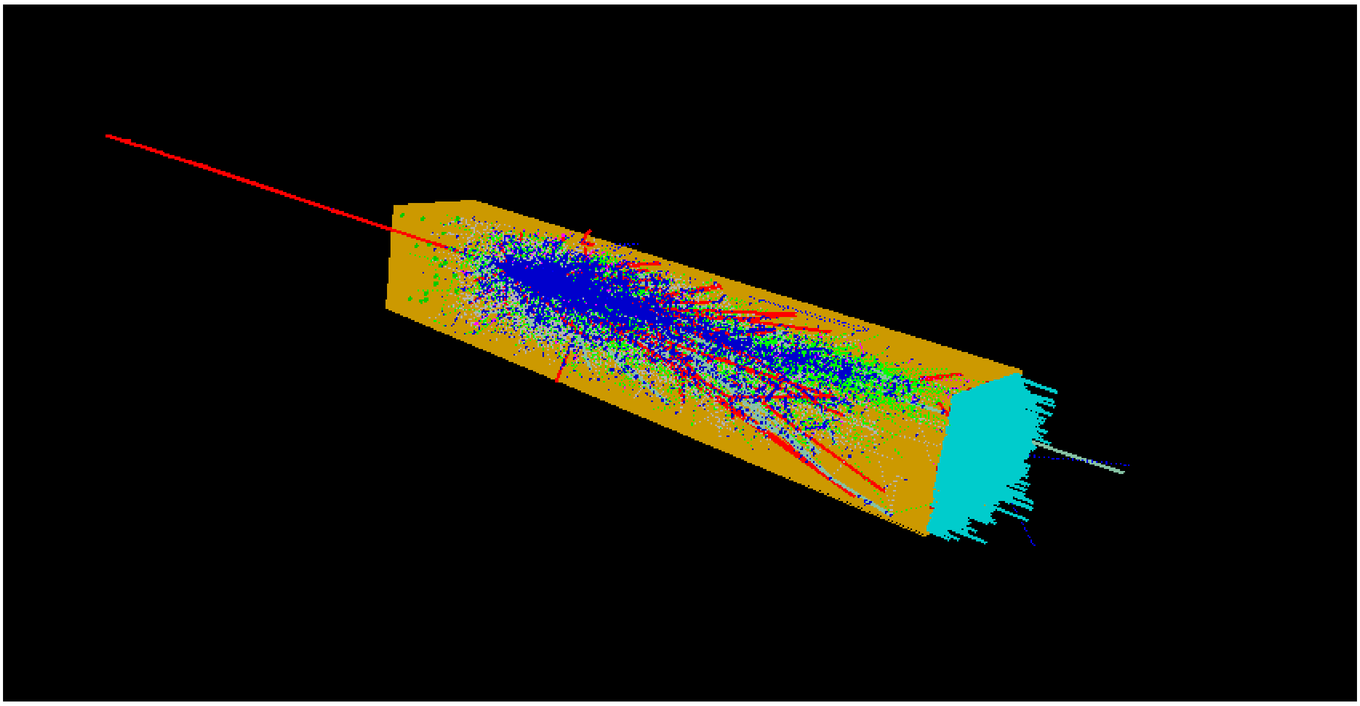


Figure 54: The visualization of a single 350GeV events, showcasing the hits in blue and the tracks in green and red, with the sum energy per straw located at the end in cyan

As can be seen as well in the visualization the majority of the hits take place within the center of the calorimeter at about 1-2 interaction lengths depth, and the spreads out wider after that into a less dense distribution.

5.4.4 Runs with FoCal-E - g4

Now while strictly speaking not a main goal it seemed quite prudent to do a similar simulation of the calorimeter, however this time it would be with FoCal-E in front to see how it would impact the distribution of energy deposits as well as the general results. This was done with the g4 implementation, as it also doubled to check whether this implementation had the same quirk as the vmc implementation.

So starting out simple with the sum energy loss plot:

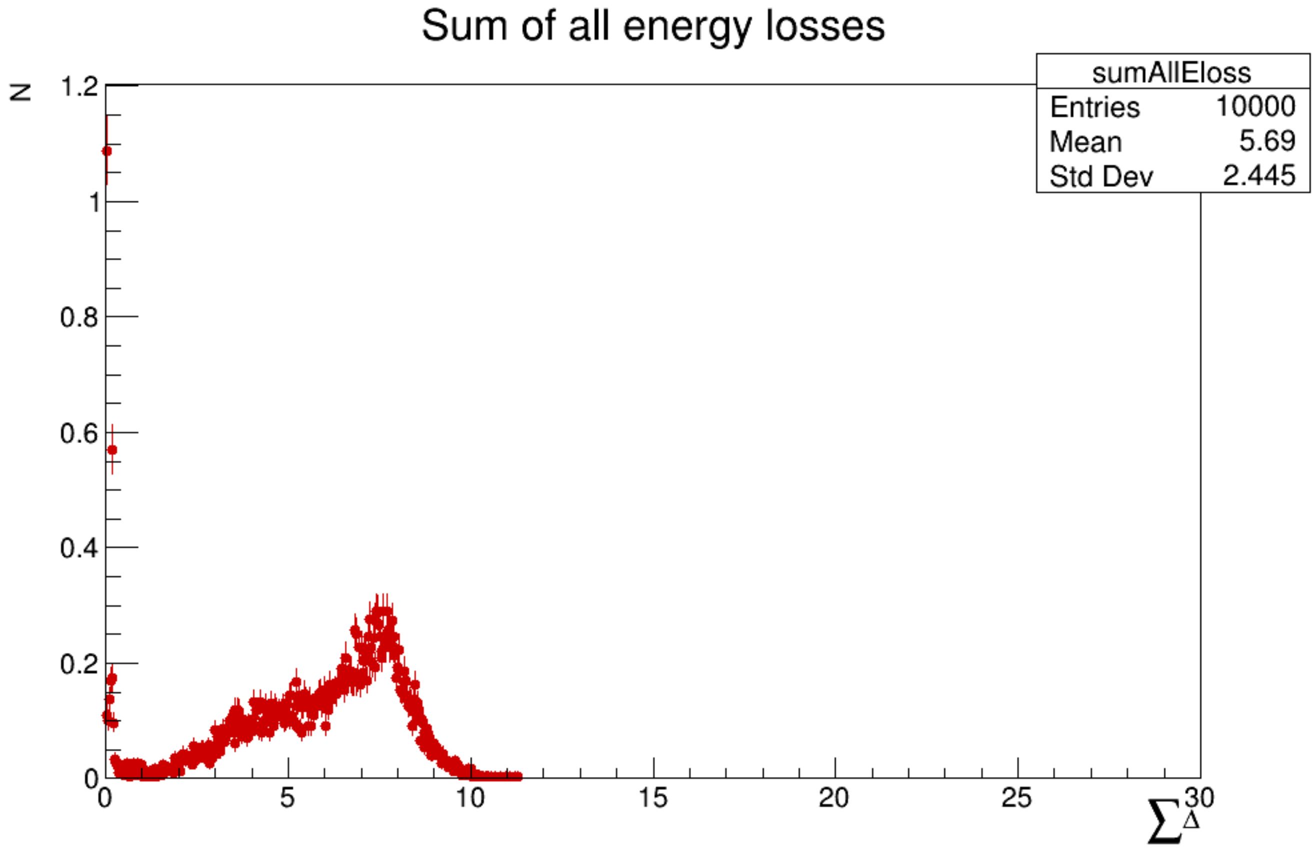


Figure 55: Sum energyloss plot for FoCal-H, this time with FoCal-E in front

Immediately we do notice the shoulder from FoCal-E in front which fits with what is expected, however here we can see a quirk for the g4 implementation, which also shows in other runs done with it, there are a large amount of 0 or almost 0 events that completely skew the scale of the plots, while there are ways that somewhat allieviate these issues, they do appear to be a consistent quirk for this type of plot with the g4 implementation.

If we look at the more general collection of plots:

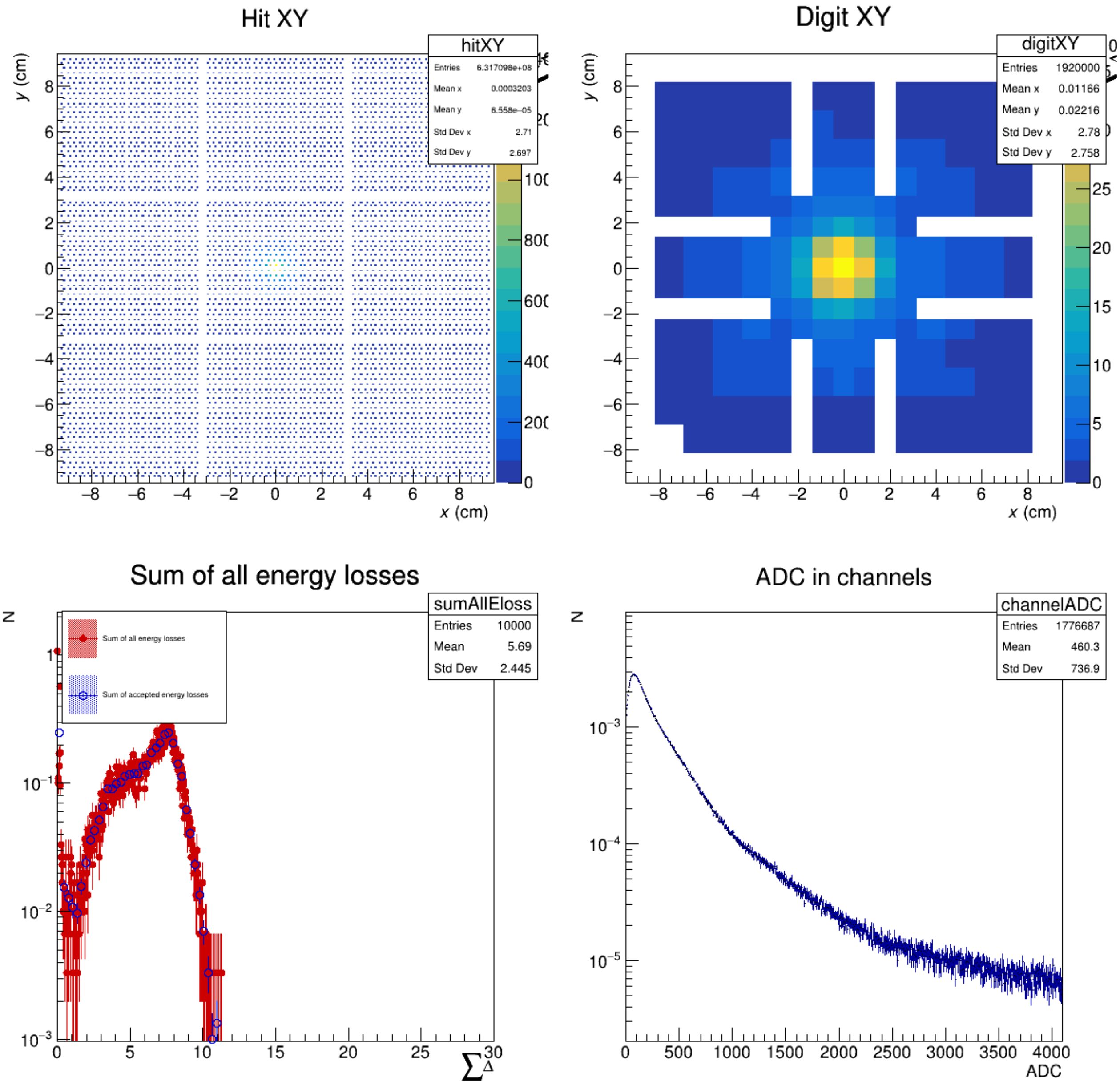
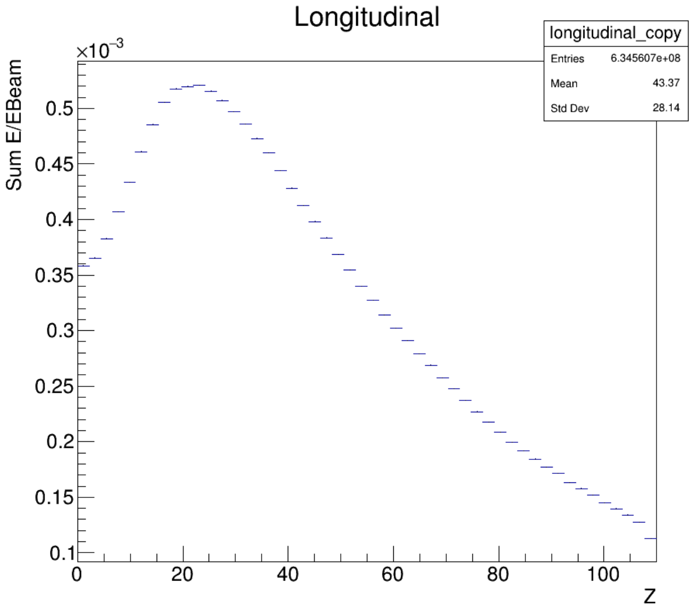


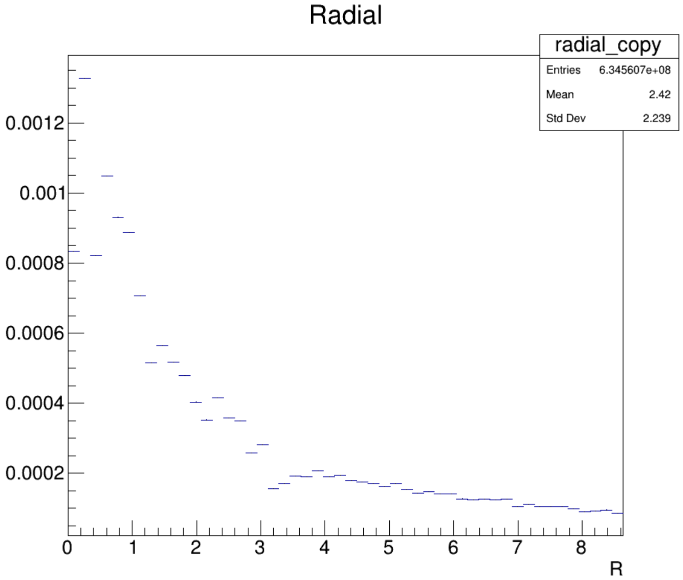
Figure 56: The collection of plots similar to what was shown for 350GeV without FoCal-E earlier.

This collection shows largely what one would expect again, however the logarithmic scale for the bottom two plots comes in handy in the sum of energy losses plot, as it diminishes the visual impact of the extreme 0 points and makes the shoulder more visible.

Finally we would of course like to see the distribution of energylosses throughout the calorimeter, which gives the following:



(a) Longitudinal energy distribution at 350GeV with FoCal-E

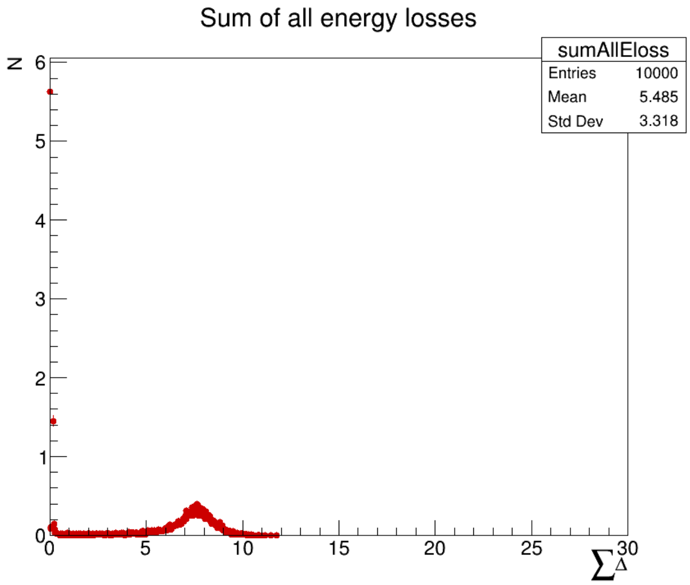


(b) Longitudinal energy distribution at 350GeV with FoCal-E

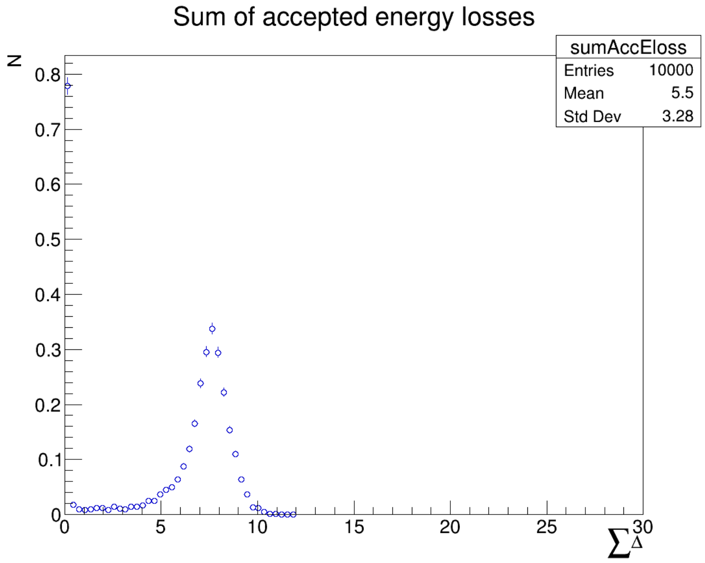
Here we can see that while the overall trends of the distribution remains much like it was for the case without FoCal-E in front. However there is one major difference, in the longitudinal distribution it starts noticeably higher on the energy scale, whereas without FoCal-E in front it starts from 0, this does fit with the idea that some showers have already begun by the time that they enter the FoCal-H calorimeter.

5.4.5 Comparison g4 vs. vmc

An additional number of runs were made with the g4 implementation, of note a run at 350 GeV without FoCal-E in front was also made, specifically with the purpose of comparing the two simulation implementations. The results of the comparison run of the g4 implementation is the following:



(a) Sum of energy losses distribution



(b) Sum of accepted energy losses distribution

These first plots of the energy loss distributions illustrate the main quirk of the g4 implementation well, as well as the use that the accepted energy loss distribution can provide. As can be seen there are these massive 0 energy event data points that massively skew the scale of these plots, however with the accepted energy loss plot we can eliminate the most egregious of them, suggesting that they are a type of ”false events” that just serve to mess with the scale of the plots.

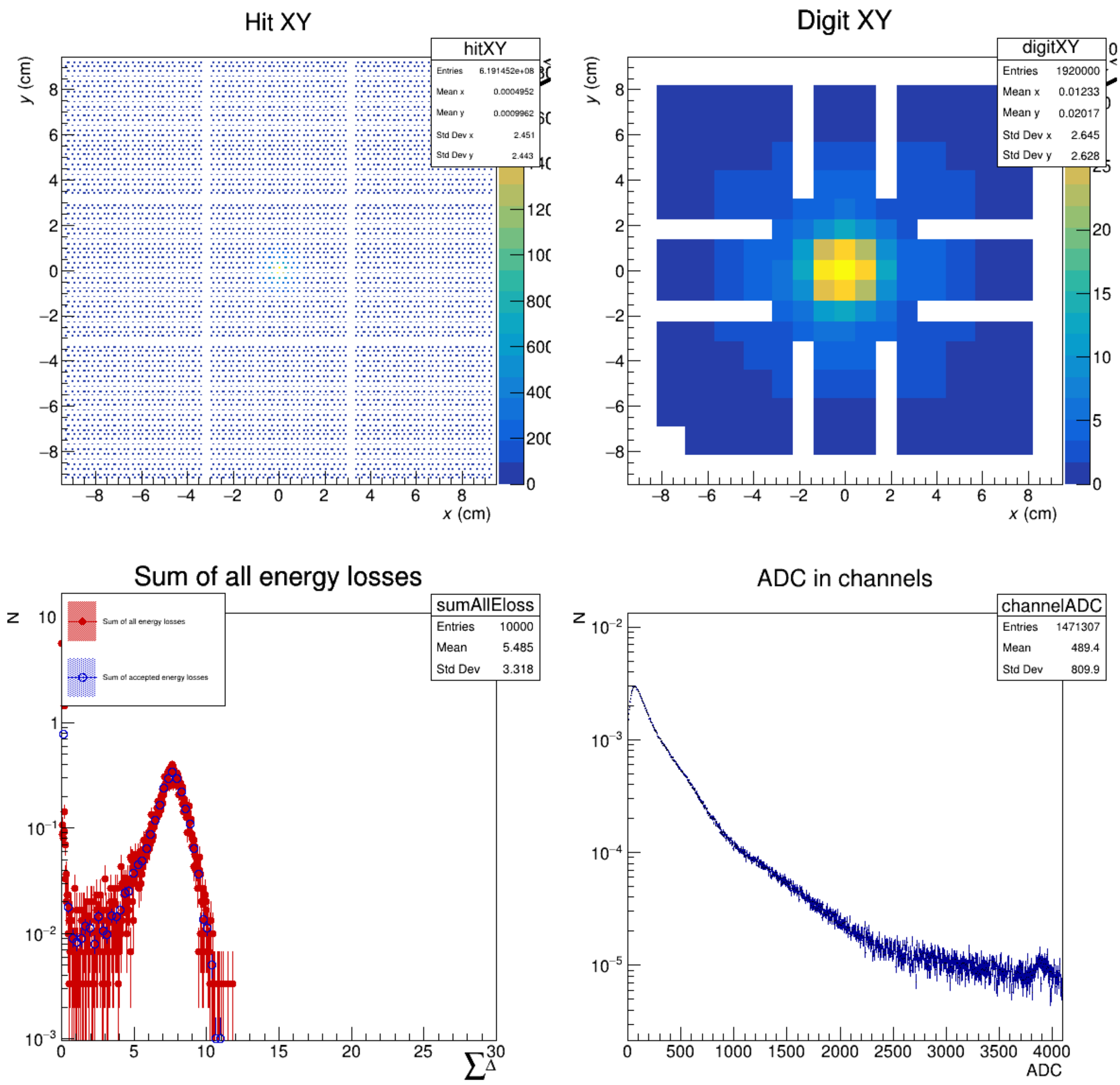


Figure 59: The collection of plots, this time for a g4 simulation done without FoCal-E in front

Here we seen another difference between the g4 and vmc implementation, the amount of noise for the ADC values, overall the g4 simulation show a much lower amount of uncertainty and variance in the number of hits at a given ADC count.

Finally to demonstrate the energy distribution profile of this simulation:

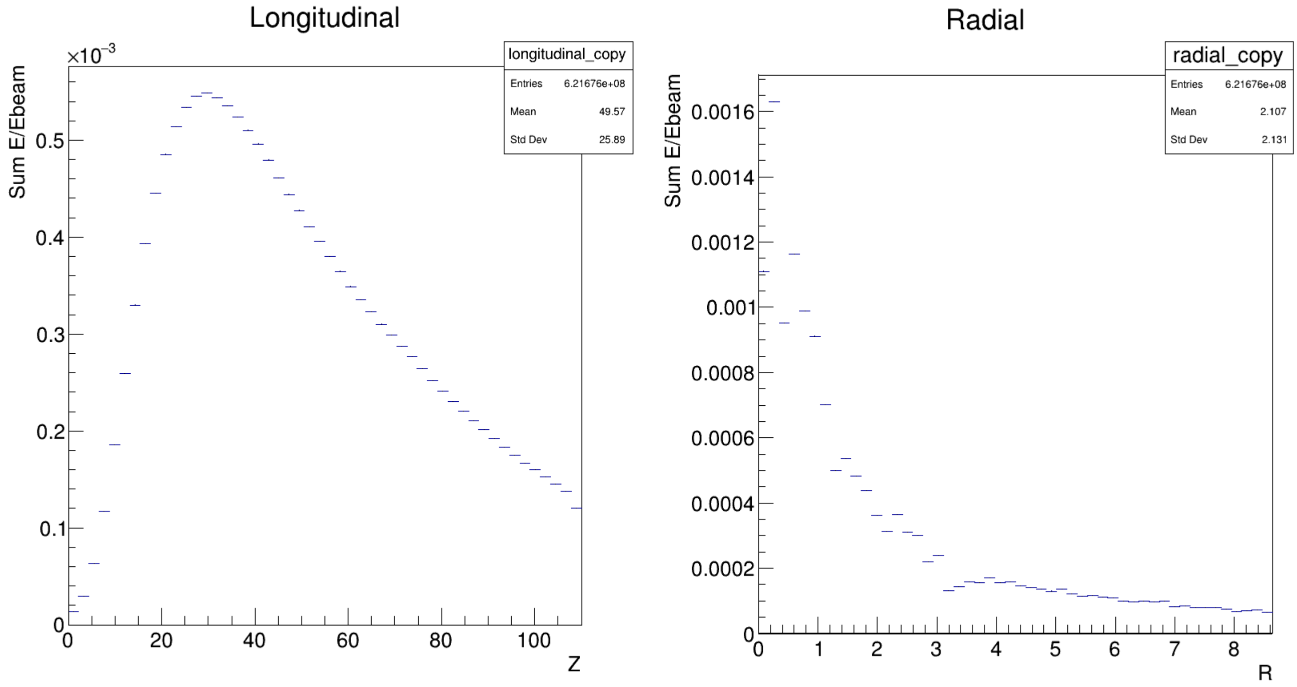
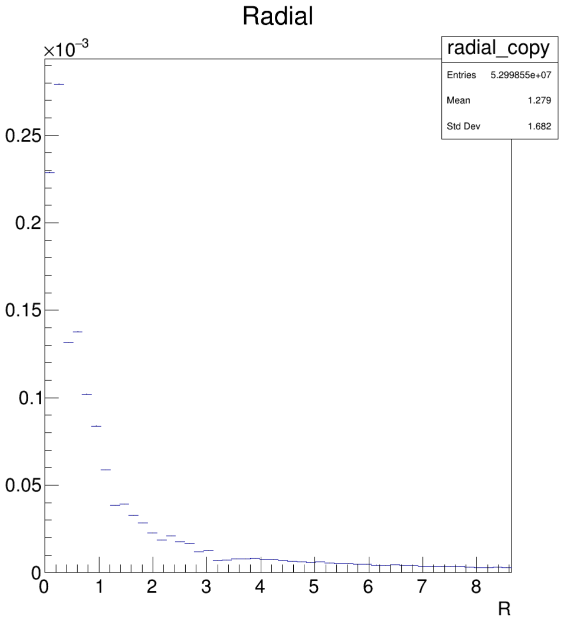
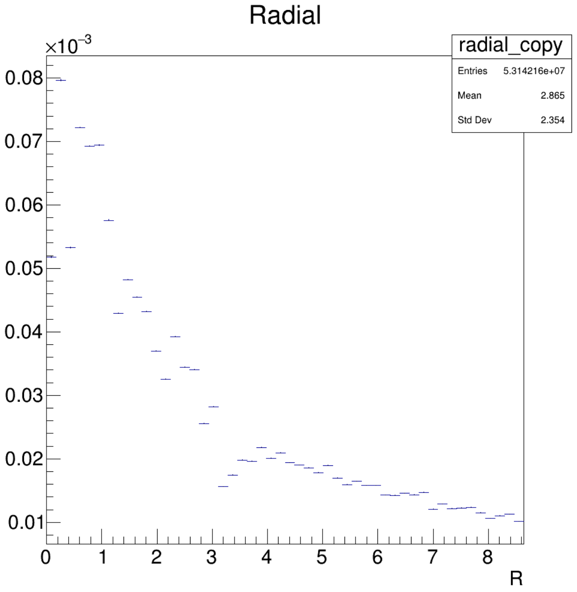


Figure 60: Energy distribution profile for a 350GeV simulation run with the g4 implementation.

Here we can see a very familiar picture along the Z-axis, and a more expected result in the radial direction, seemingly demonstrating that the fluctuations in the vmc implementation were just a quirk of the simulation. For a full view of the profiles for various interaction lengths for the g4 implementation, see the appendix, however here we will just review the extreme points of it.



(a) Radial profile for the first interaction length: 0-18 cm



(b) Radial profile for the last interaction length of the calorimeter: 90-110 cm

This all serves to again illustrate the difference in the distribution between the start of the calorimeter and the end, showing the much more centrally focused start, and the more broadened end distribution.

5.5 Final beamtest data

Finally, as a last demonstration of the benefits of this simulation package is its ability to analyse the actual beamtest data, due to the simulation saving its data in a similar style to the way that the actual data collecting from the beamtests was saved.

First by running the data from the beamtest through the analysis code we can get a view of the energy event distribution. Here is a 350GeV run without FoCal-E in front recorded during the November 2022 beamtest (run 1500):

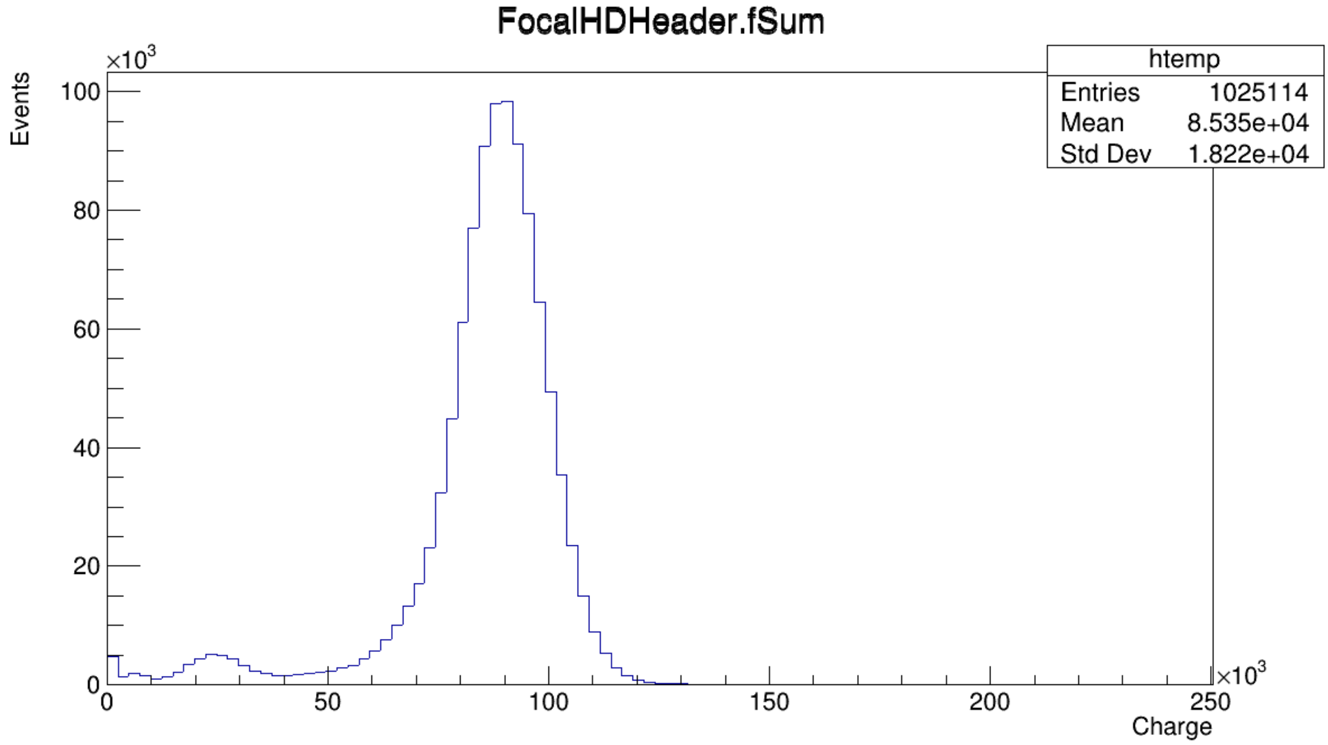


Figure 62: Charge distribution of run 1500 from the November 2022 beamtest

While this is rather simple, the more interesting analysis is by looking at the ADC counts for the run, which it can also do for the simulation.

Here is an ADC distribution analysis of the beamtest data (run 1500):

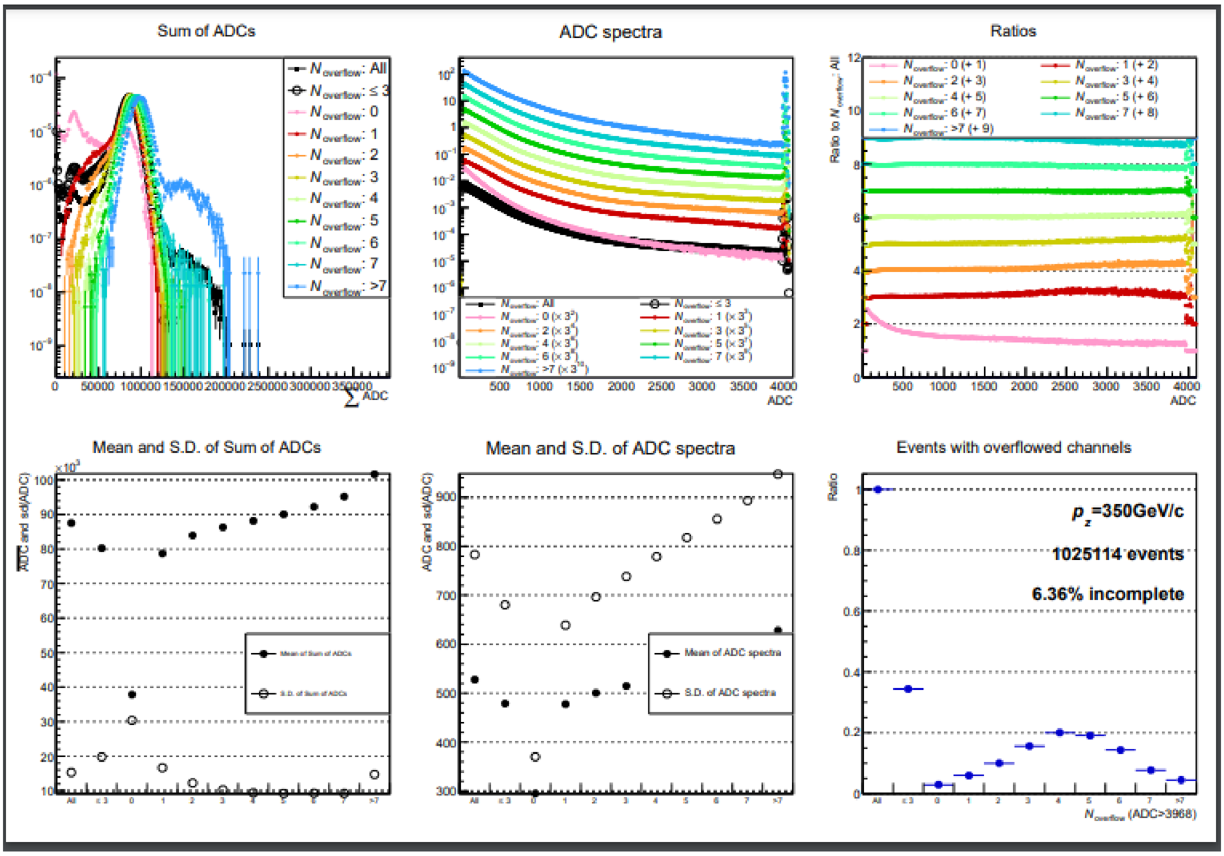


Figure 63: Illustration of ADC distribution as well as number of saturated channels

And here is a similar distribution analysis done on a g4 simulated 350GeV event:

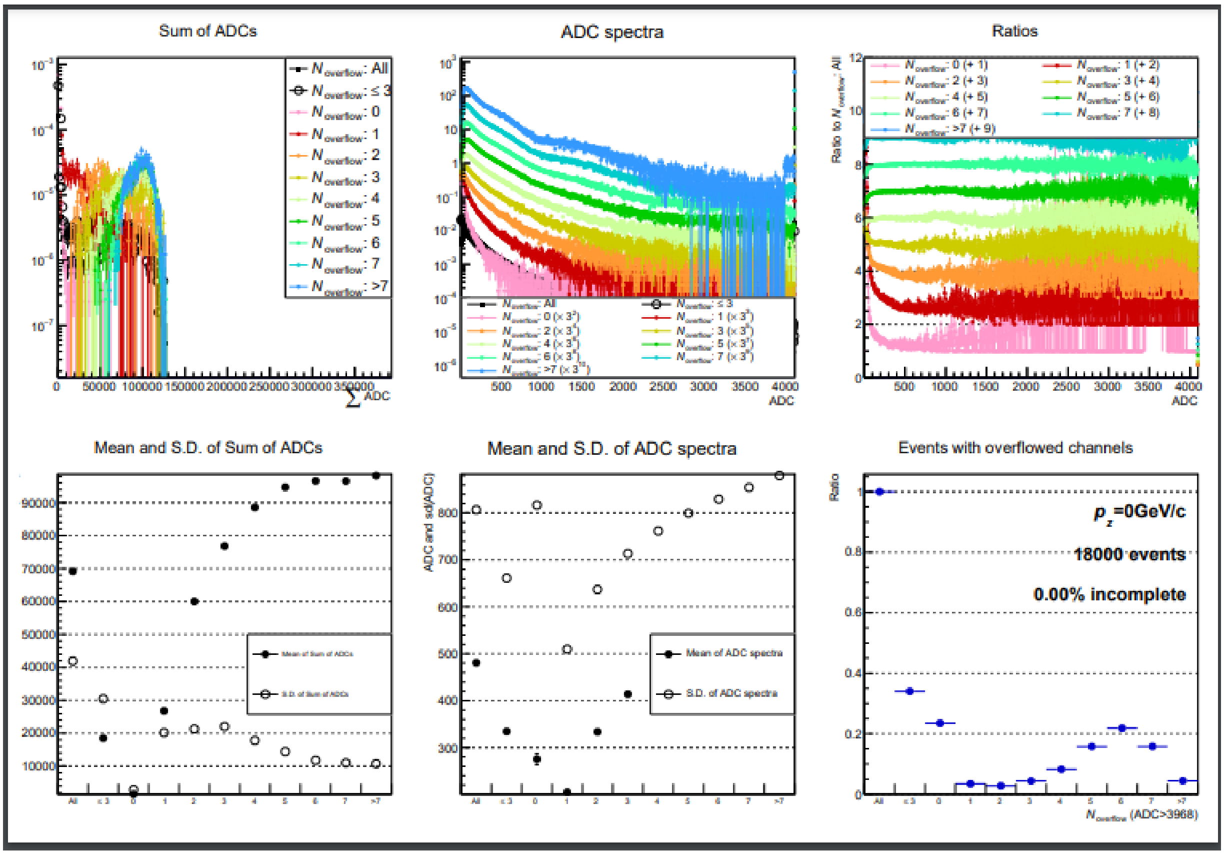


Figure 64: Illustration of ADC distribution as well as number of saturated channels

In all plots the various colours refers to the number of overflows, so it plots ADC counts and spectra for different number of overflows, as well as just counting the number of events with specific numbers of overflow.

6 Conclusion

This project has given me personally a great insight into the construction and setup of the Foward hadronic calorimeter, as well as the physics that it seeks to study. Further the use of the various simulation packages has also given me the ability to use the ROOT analysis program and the various simulation packages that have been used for the simulations. Practically I have also gained greater insight into how the data taking process at the beamtests function, as well as more generally how to read and interpret the data from these simulations and the various beamtest runs.

For the FoCal upgrades in general, the longitudinal and radial profiles and how they develop throughout the calorimeter will allow a better idea of how the proportions of the final upgrade should look, with both the g4 and vmc implementations of the simulation software acting as great tools for simulation, their small kinks notwithstanding.

Beyond the second prototype, these simulation packages should be very able to further simulate the third prototype, though given its much larger size, the simulations will probably be noticeably slower, and with much larger resulting files from them.

The beamtests themselves have also been essential in demonstrating the capabilities and flaws of the second prototype, and that information has then been used to prepare for the third prototype, which will be much larger compared to the second prototype.

References

[1] Alice frontpage description. https://home.cern/science/experiments/alice.

[2] Jet quenching observed by cms in heavy-ion collisions. https://cms.cern/news/jet-quenching-observed-cms-heavy-ion-collisions.

[3] N. Armesto. *Nuclear shadowing*, page 2. 2006.

[4] N. Armesto. *Nuclear shadowing*, page 3. 2006.

[5] C. H. Christensen. Focal simulation code. https://gitlab.com/cholmcc/focal_sim.

[6] A. collaboration. *Letter of Intent: A Forward Calorimeter (FoCal) in the ALICE experiment*. The CERN collaboration, 2019.

[7] A. collaboration. *Letter of Intent: A Forward Calorimeter (FoCal) in the ALICE experiment*, page 6. The CERN collaboration, 2019.

[8] A. collaboration. *Letter of Intent: A Forward Calorimeter (FoCal) in the ALICE experiment*, page 7. The CERN collaboration, 2019.

[9] A. collaboration. *Letter of Intent: A Forward Calorimeter (FoCal) in the ALICE experiment*, page 9 Fig.1. The CERN collaboration, 2019.

[10] W. R. G. Piller and W. Weise. *Phenomenology of Nuclear Shadowing in Deep-Inelastic Scattering*, page 3. 1995.

[11] M. Livan and R. Wigmans. *Calorimetry for collider physics, an introduction*, page 42. Springer, 2019.

[12] M. Livan and R. Wigmans. *Calorimetry for collider physics, an introduction*, page 45 Fig. 2.7. Springer, 2019.

[13] M. Livan and R. Wigmans. *Calorimetry for collider physics, an introduction*, pages 53–55. Springer, 2019.

[14] G.-Y. Qin and X.-N. Wang. Jet quenching in high-energy heavy-ion collisions. https://arxiv.org/pdf/1511.00790.pdf, 2015.

[15] M. Thomson. *Modern particle physics*, page 179. Cambridge University Press, 2013.

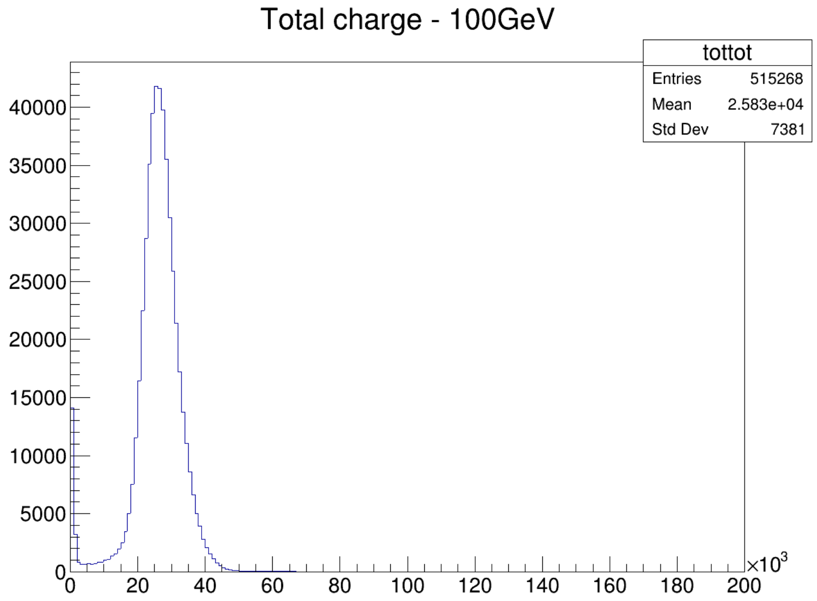
[16] M. Thomson. *Modern particle physics*, page 185. Cambridge University Press, 2013.

[17] M. Thomson. *Modern particle physics*, page 184. Cambridge University Press, 2013.

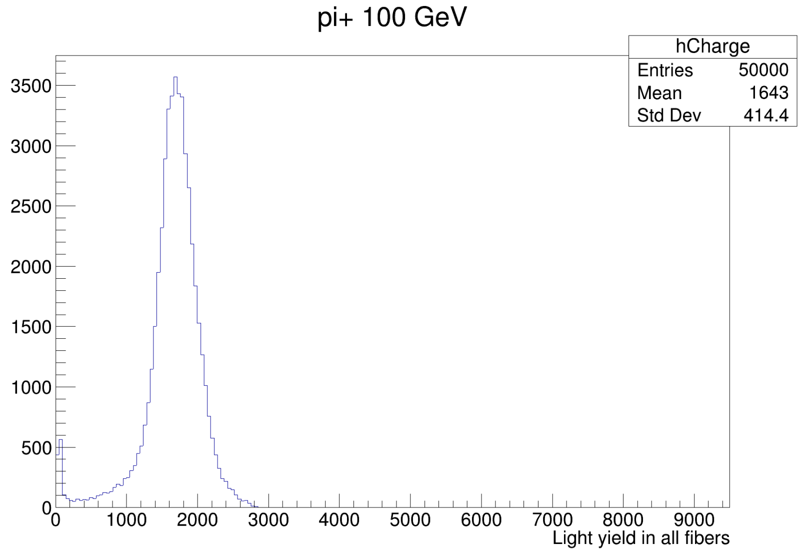
A Appendix

A.1 Further results

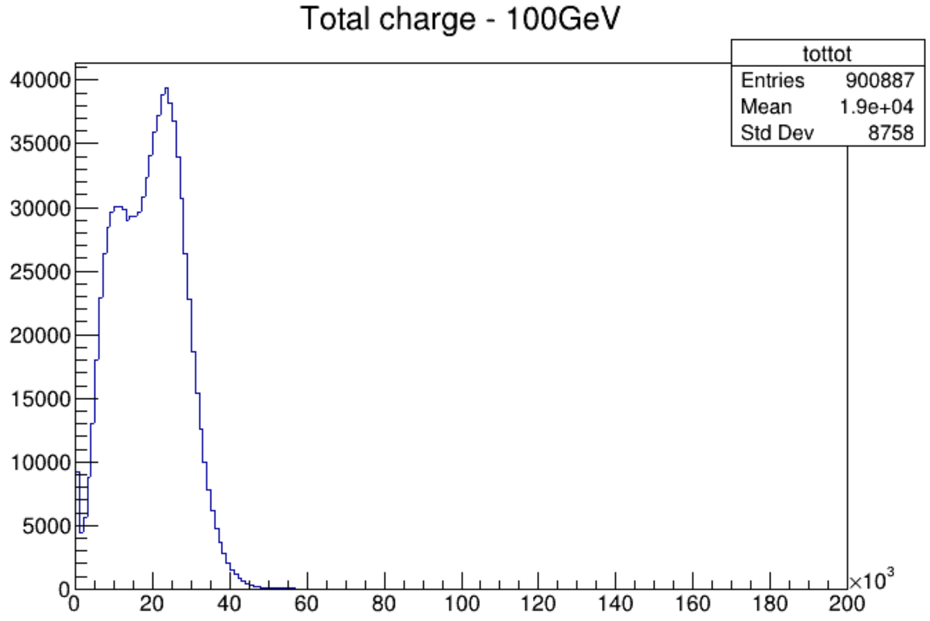
A.1.1 Further comparisons between testbeam package and recorded data



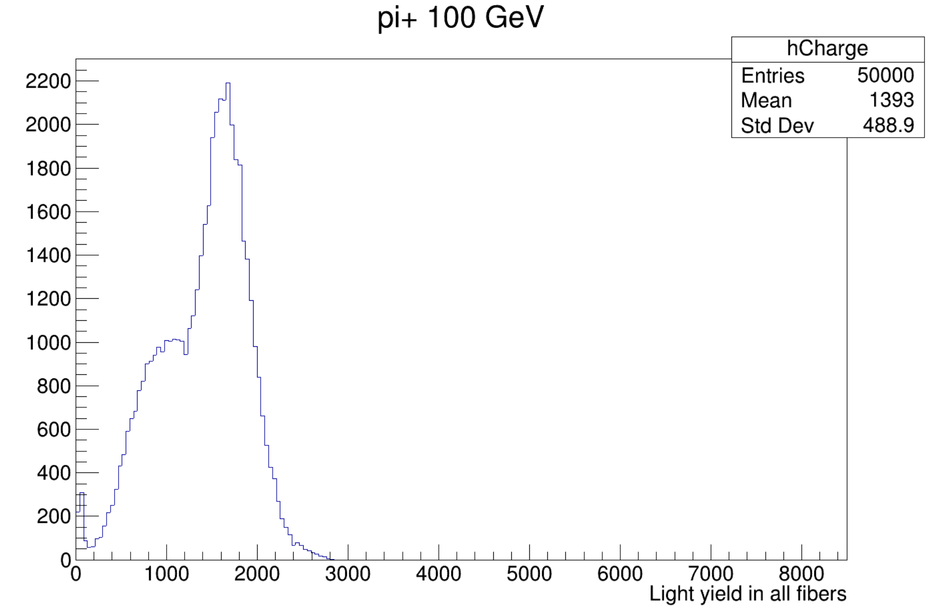
(a) Data from November beamtest at 100 GeV without FoCal-E in front



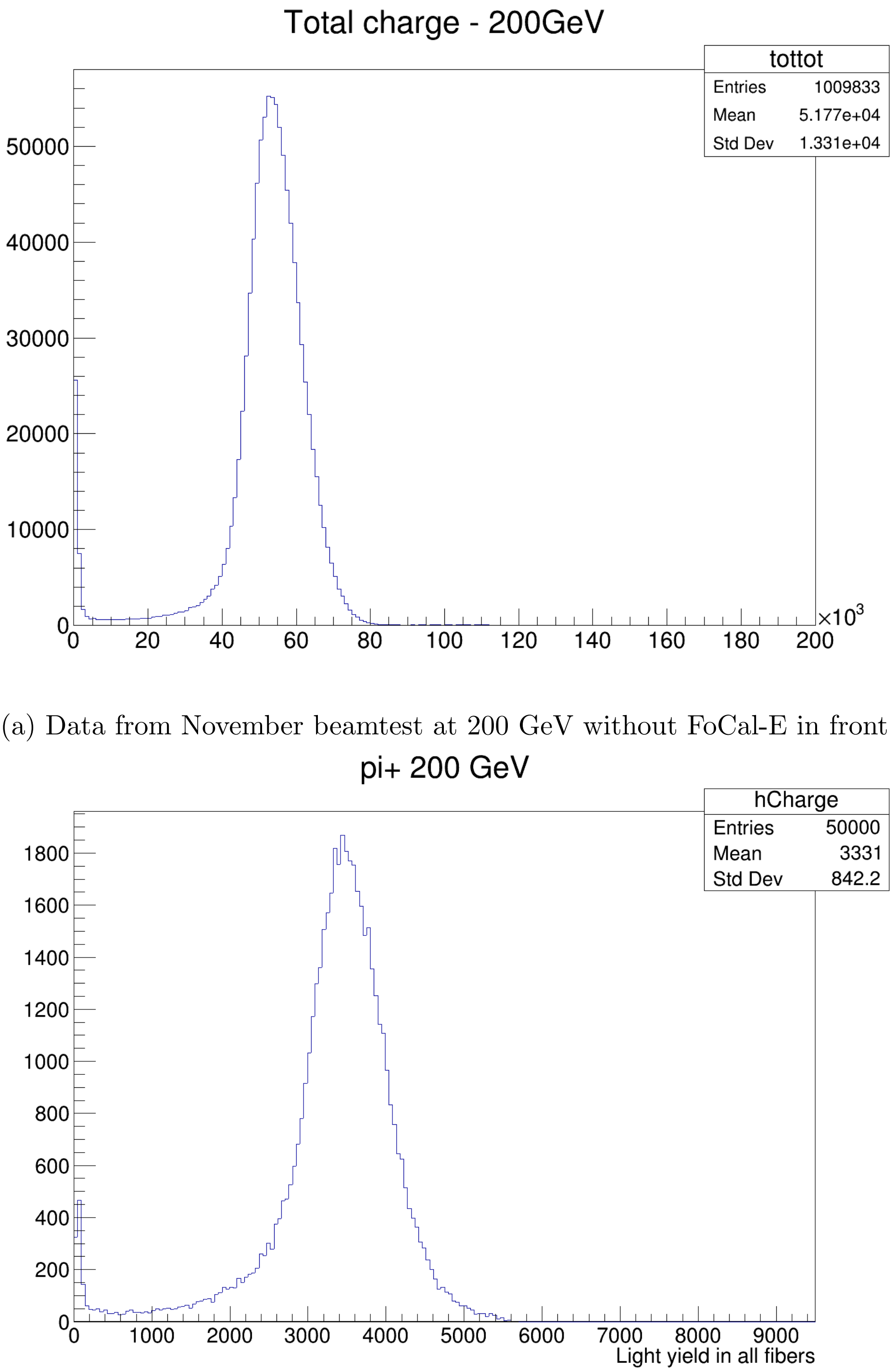
(b) Comparison simulation of 100 GeV without FoCal-E in front



(a) Data from November beamtest at 100 GeV with FoCal-E in front

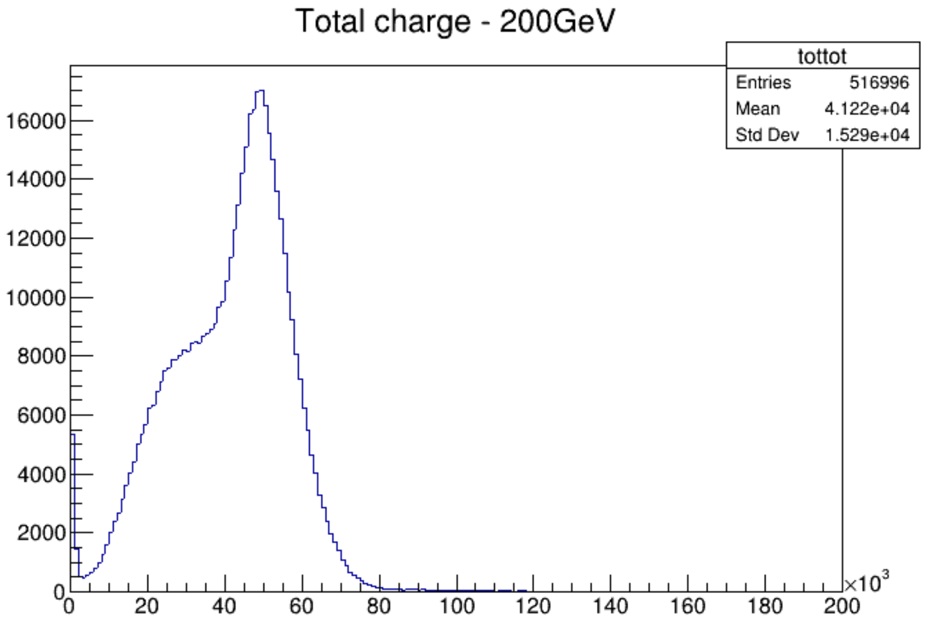


(b) Comparison simulation of 100 GeV with FoCal-E in front

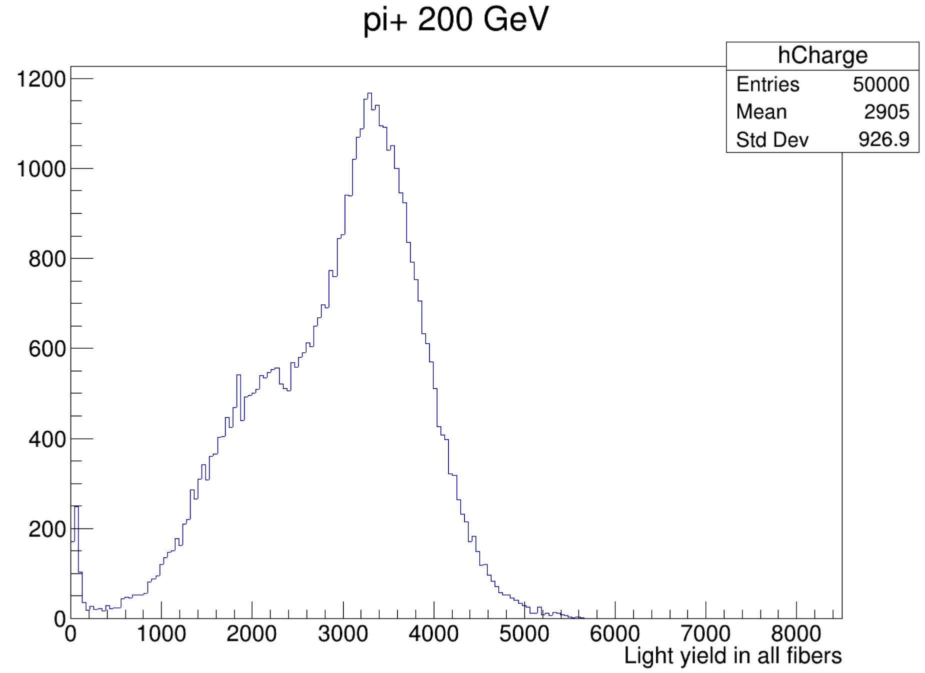


(a) Data from November beamtest at 200 GeV without FoCal-E in front

(b) Comparison simulation of 200 GeV without FoCal-E in front

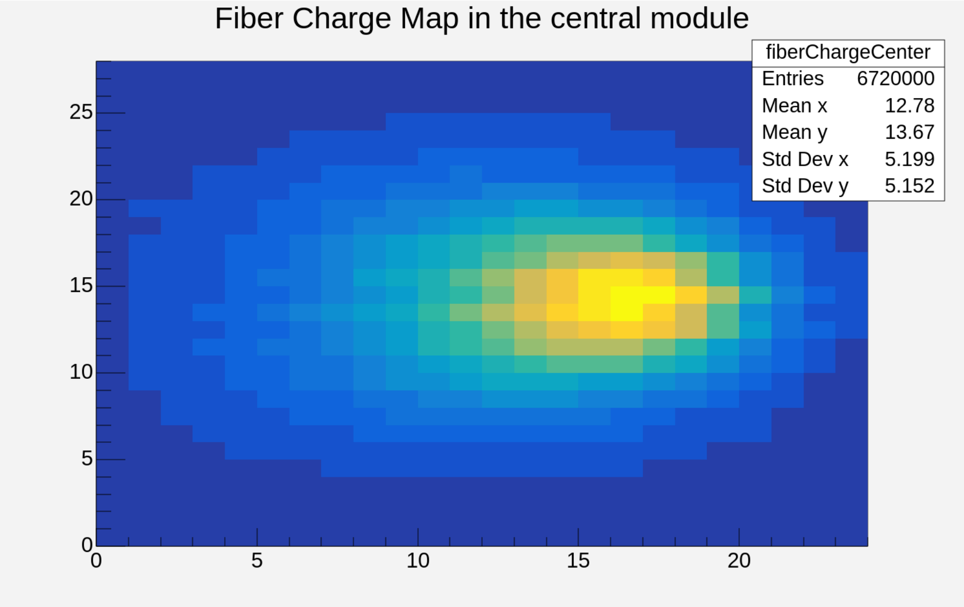


(a) Data from November beamtest at 200 GeV with FoCal-E in front

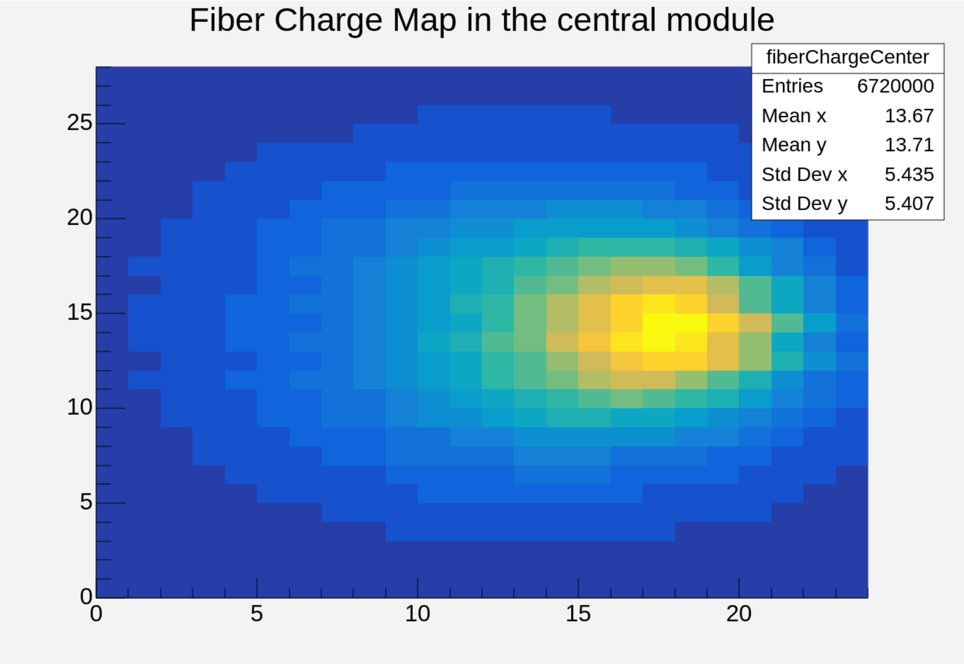


(b) Comparison simulation of 200 GeV with FoCal-E in front

A.1.2 Additional heatmaps from testbeam simulation with geant4 v11.1.1

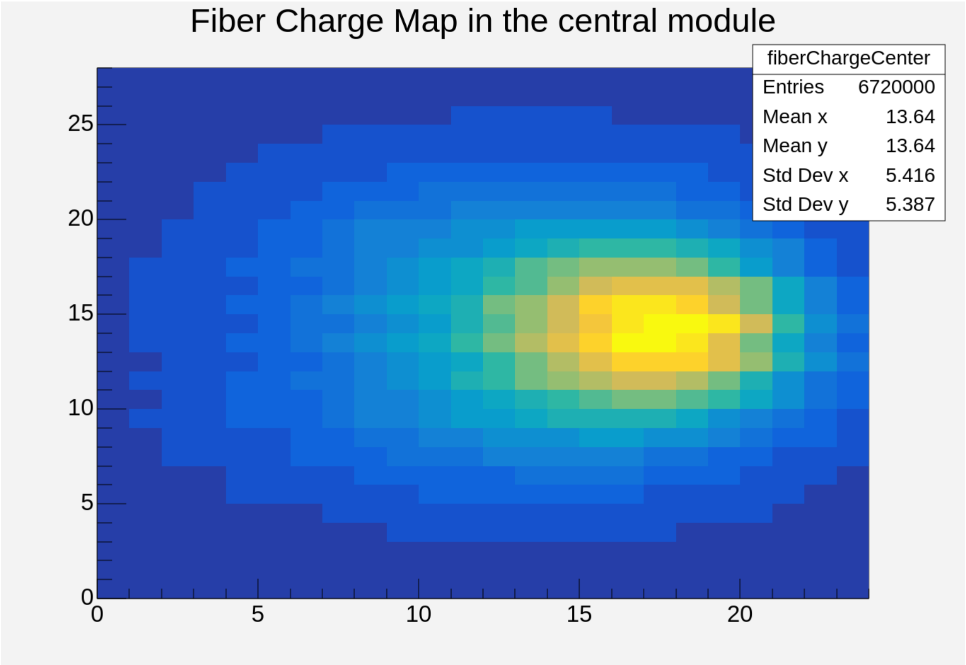


(a) Second run at 350 GeV without FoCal-E

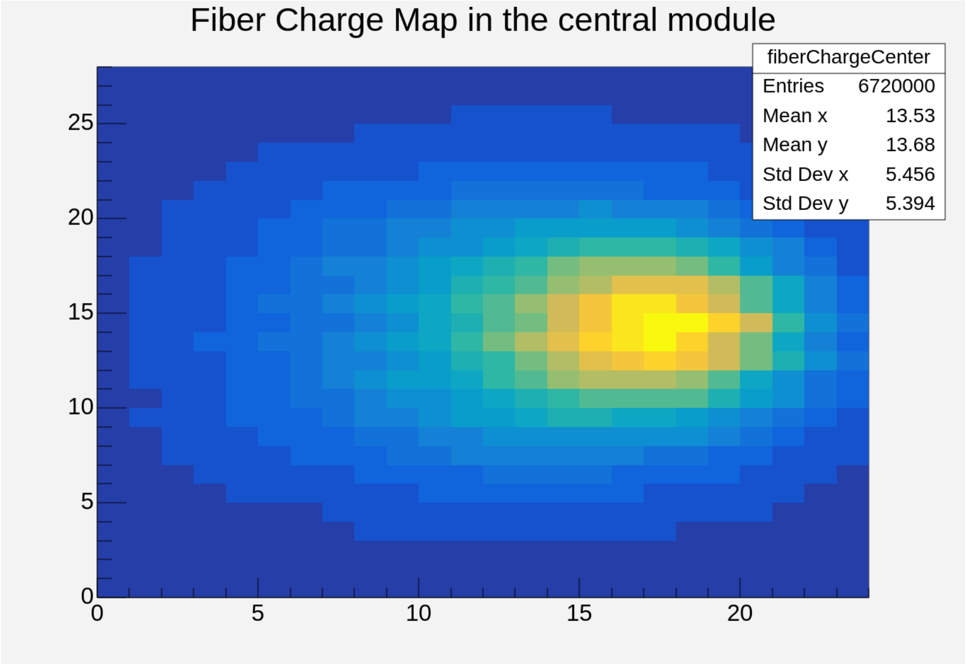


(b) Second run at 350 GeV with FoCal-E

A.1.3 Additional heatmaps from testbeam simulation with geant4 v10.7.3

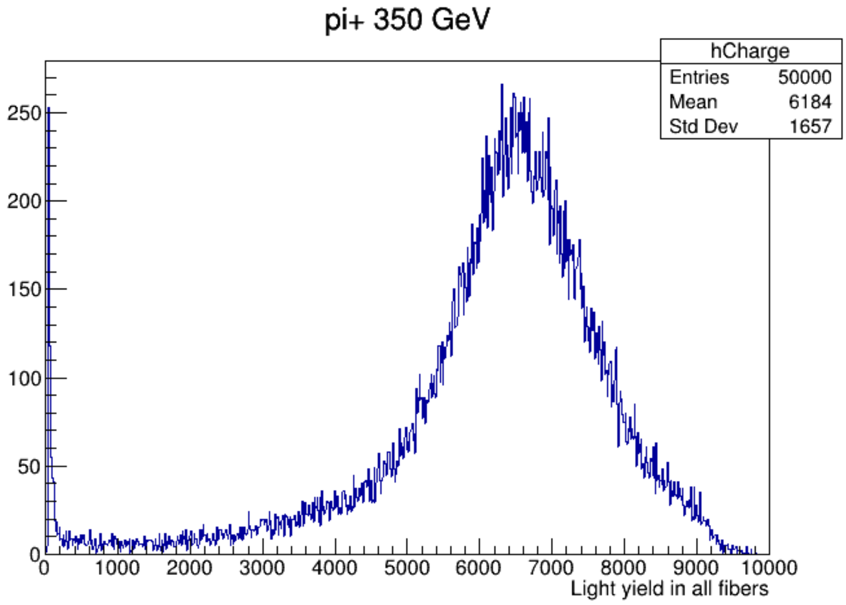


(a) First run for 10.7.3 at 350 GeV with FoCal-E

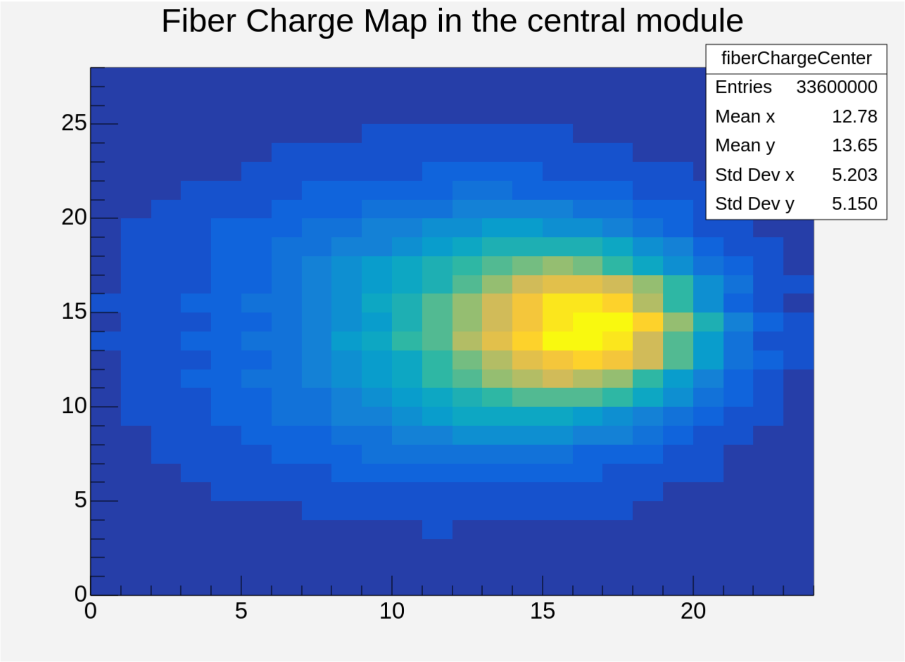


(b) Second run for 10.7.3 at 350 GeV with FoCal-E

A.1.4 Extra runs with updated testbeam simulation package



(a) Second run of 50k events at 350 GeV without FoCal-E with the update to the base package



(b) Heatmap of the same 50k event run

A.1.5 Extra digitized outputs

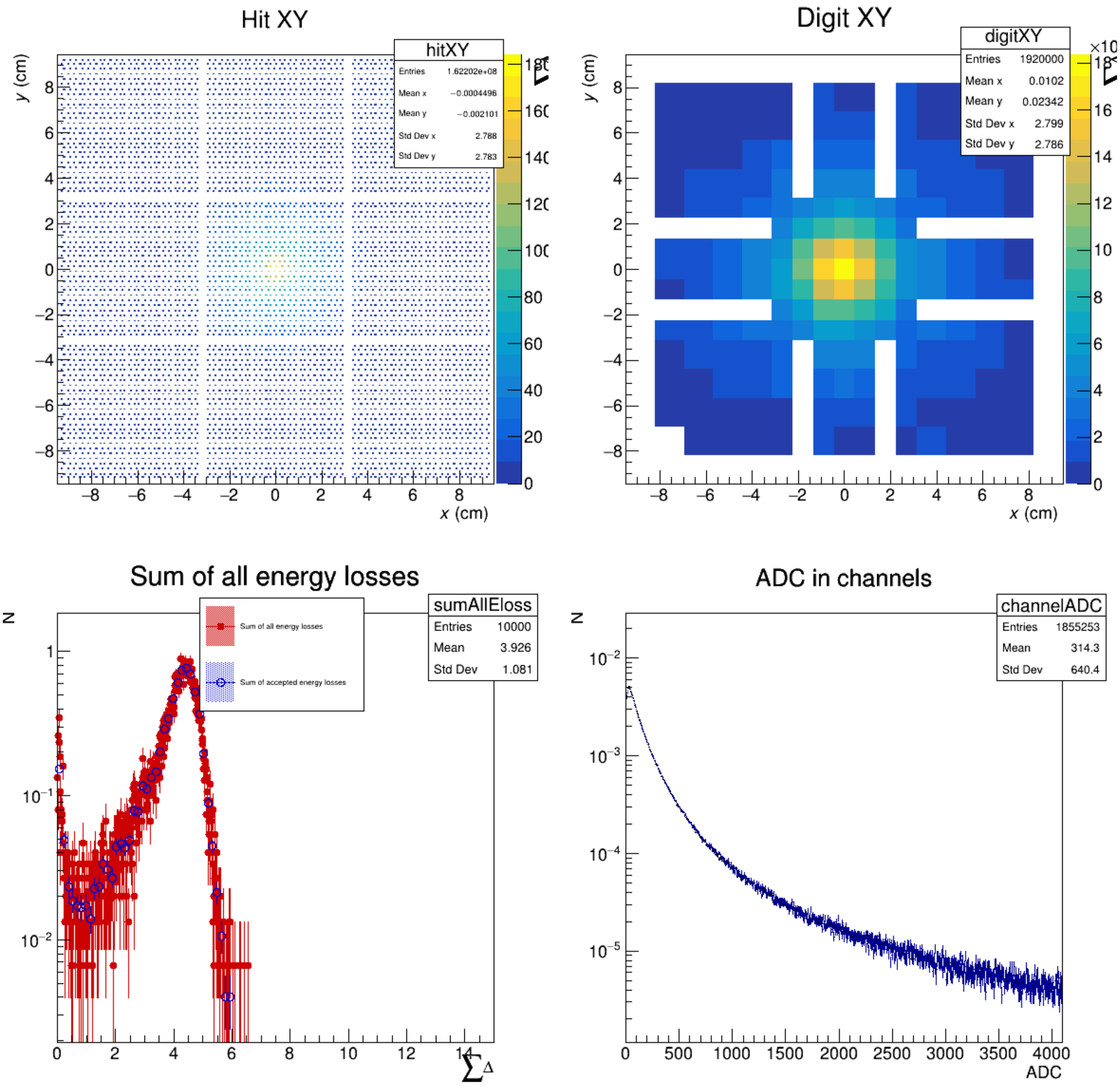


Figure 72: Digitized output for a 200 GeV run with the vmc implementation

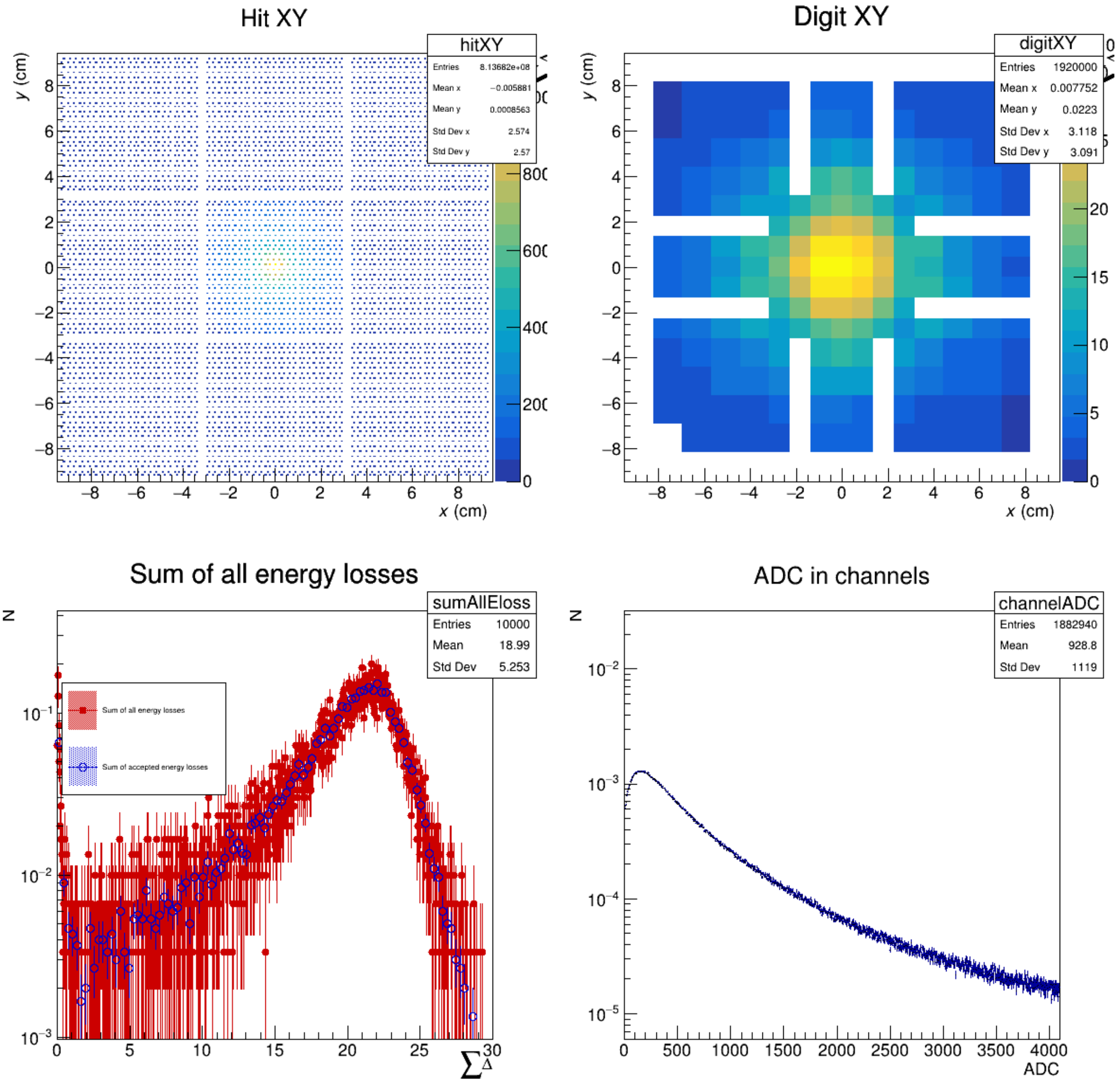


Figure 73: Digitized output for a 1 TeV run with the vmc implementation

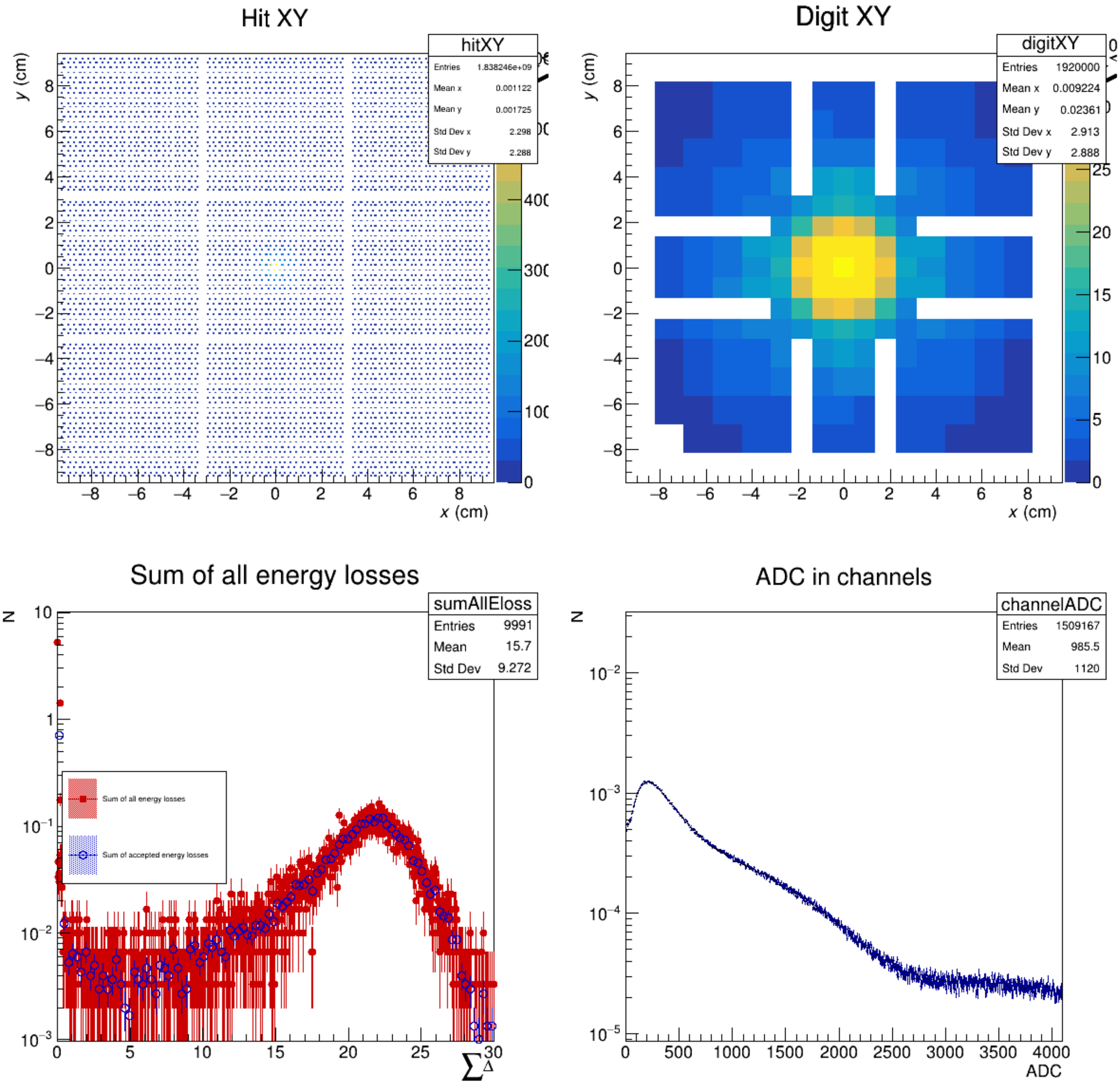


Figure 74: Digitized output for a 1 TeV run with the g4 implementation

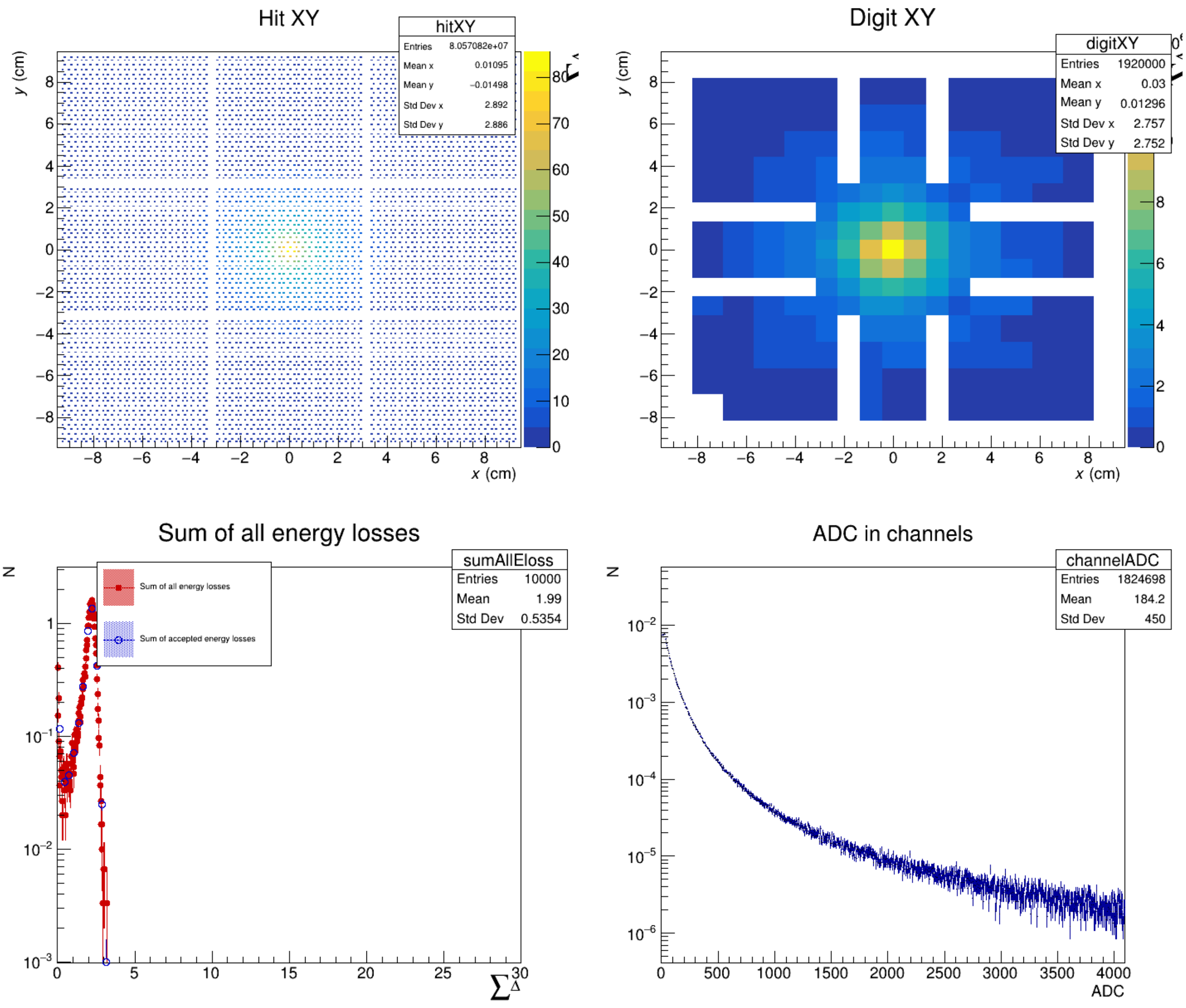


Figure 75: Digitized output for a 100 GeV run with the vmc implementation

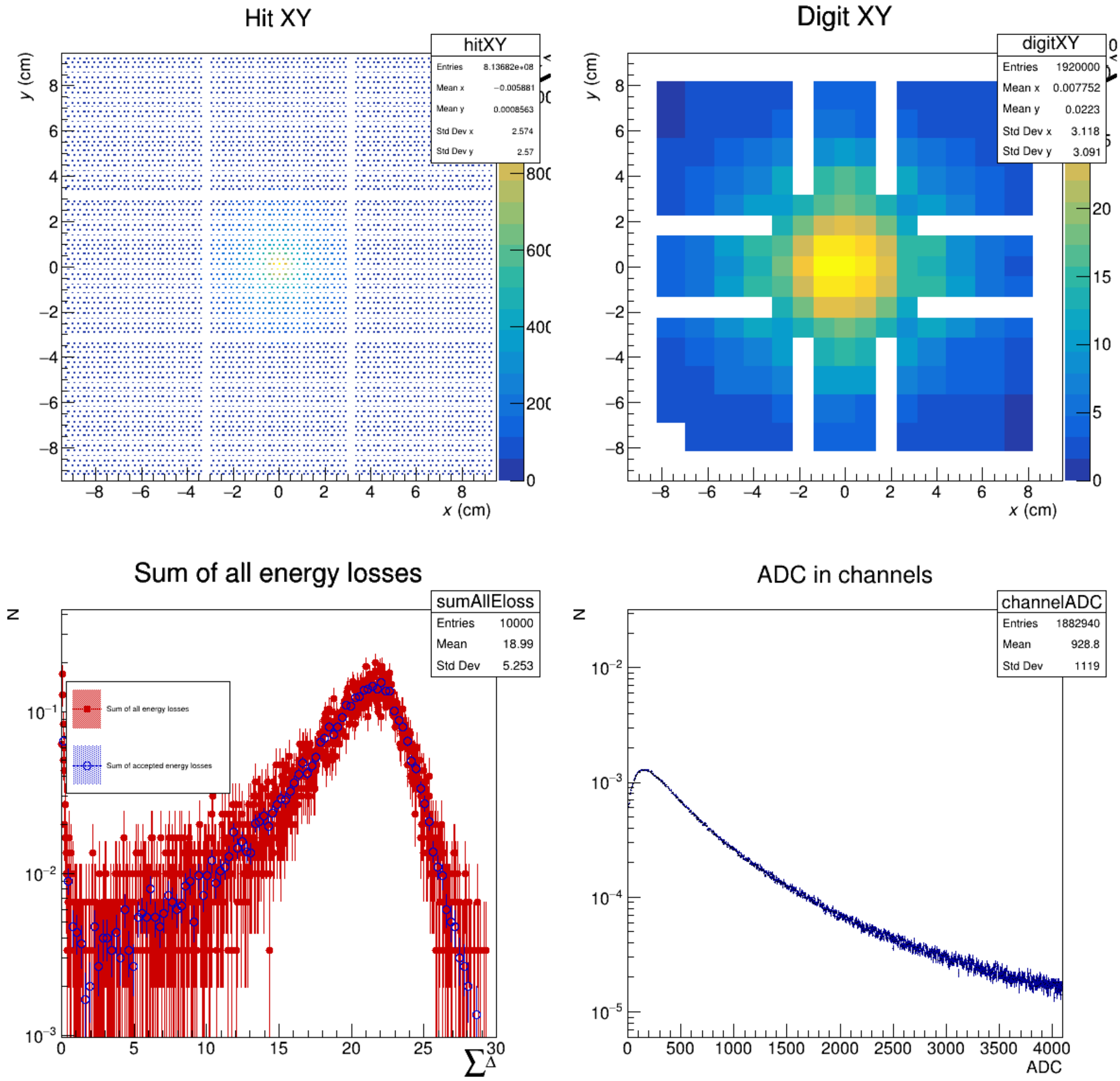
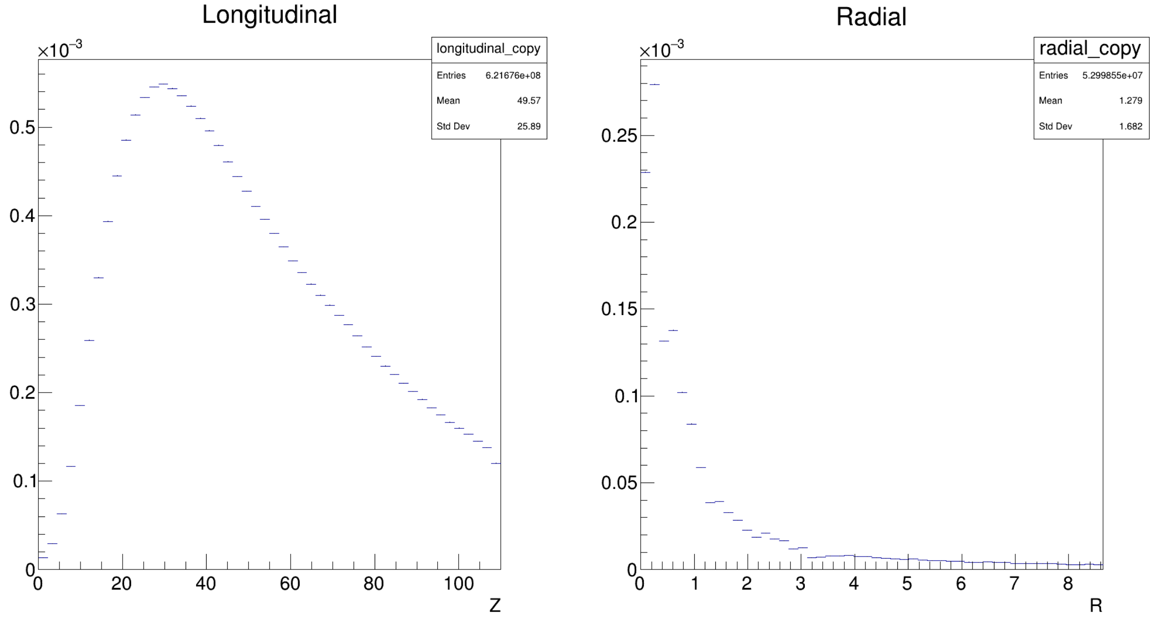
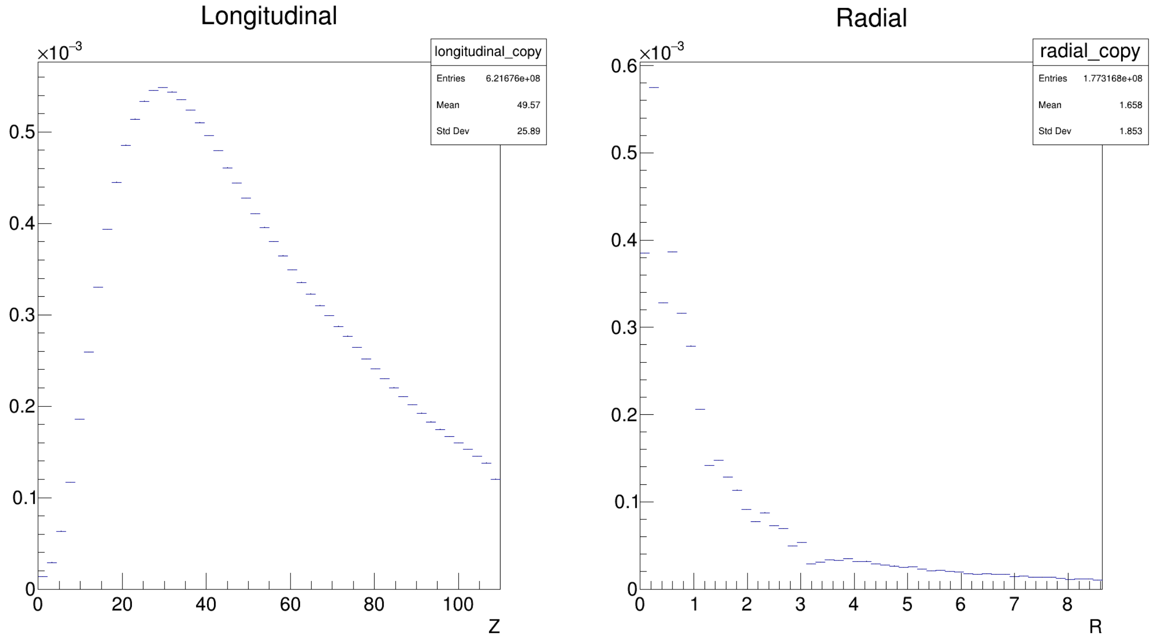


Figure 76: Digitized output for a 1 TeV run with the vmc implementation

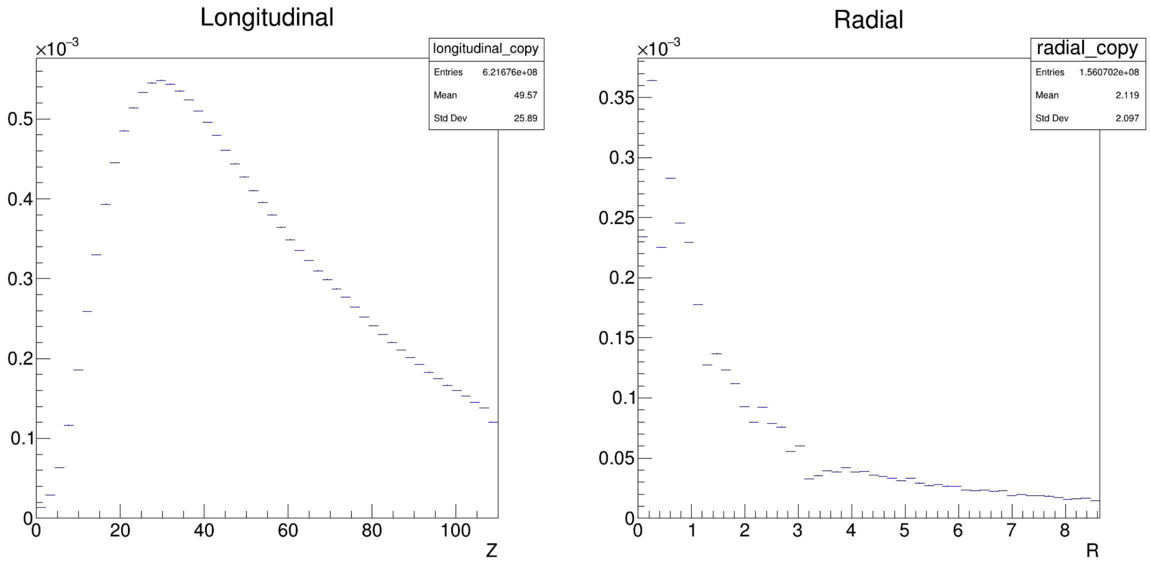
A.1.6 Full profile 350 GeV without FoCal-E - g4 implementation



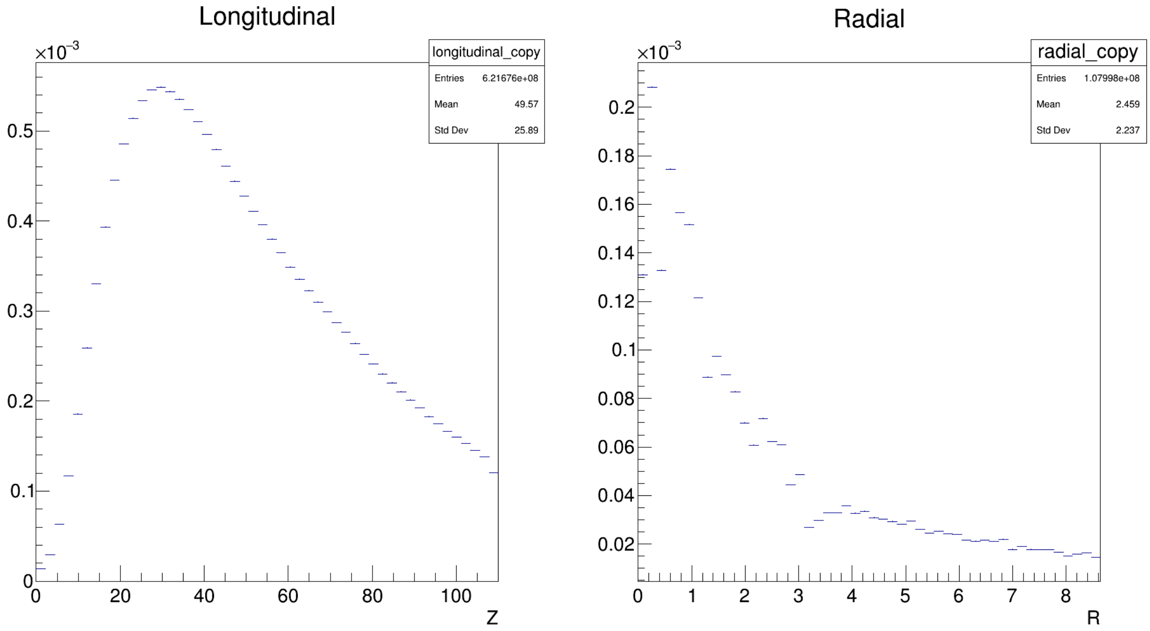
(a) Profile for first interaction length: 0-18 cm



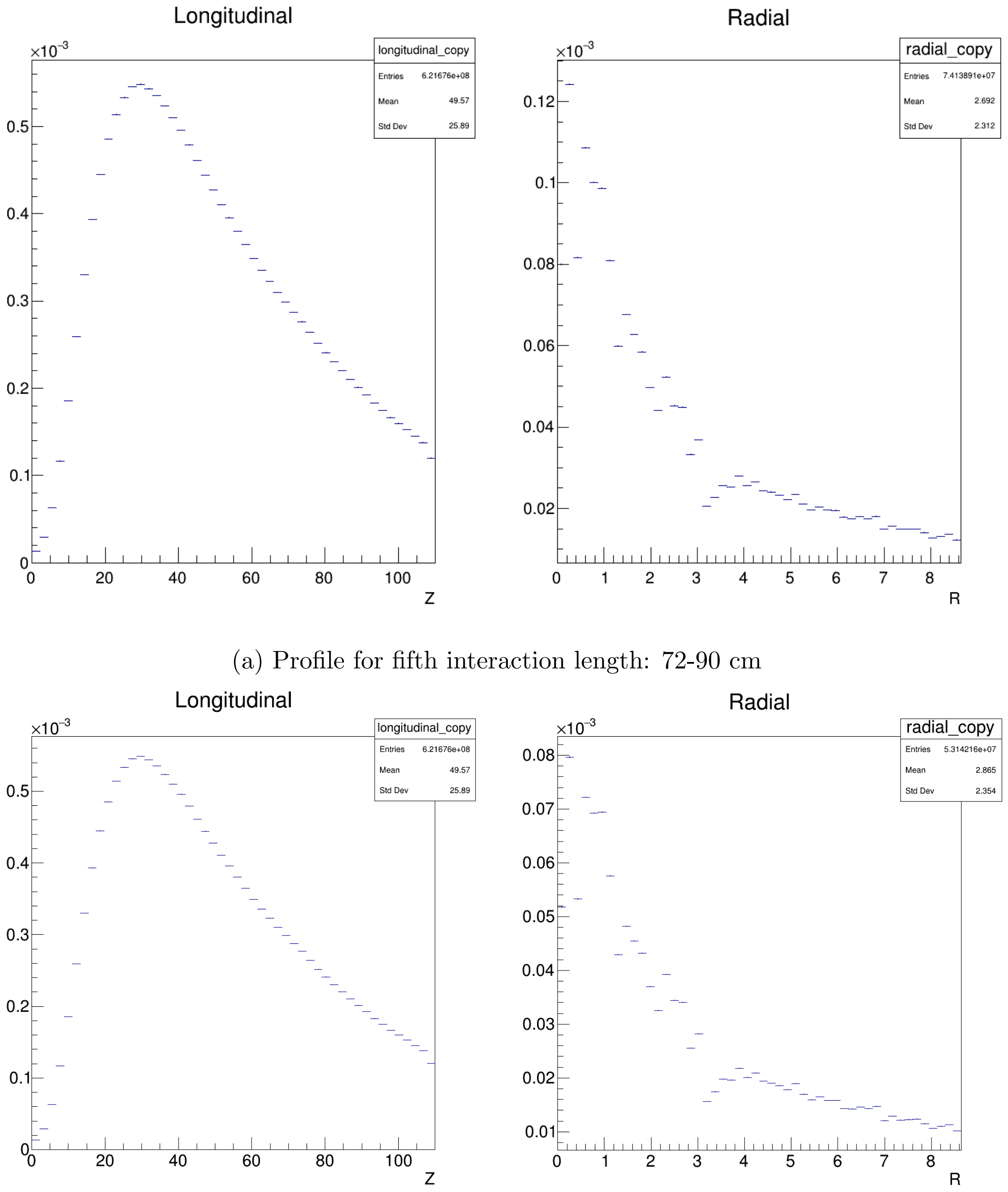
(b) Profile for second interaction length: 18-36 cm



(a) Profile for third interaction length: 36-54 cm



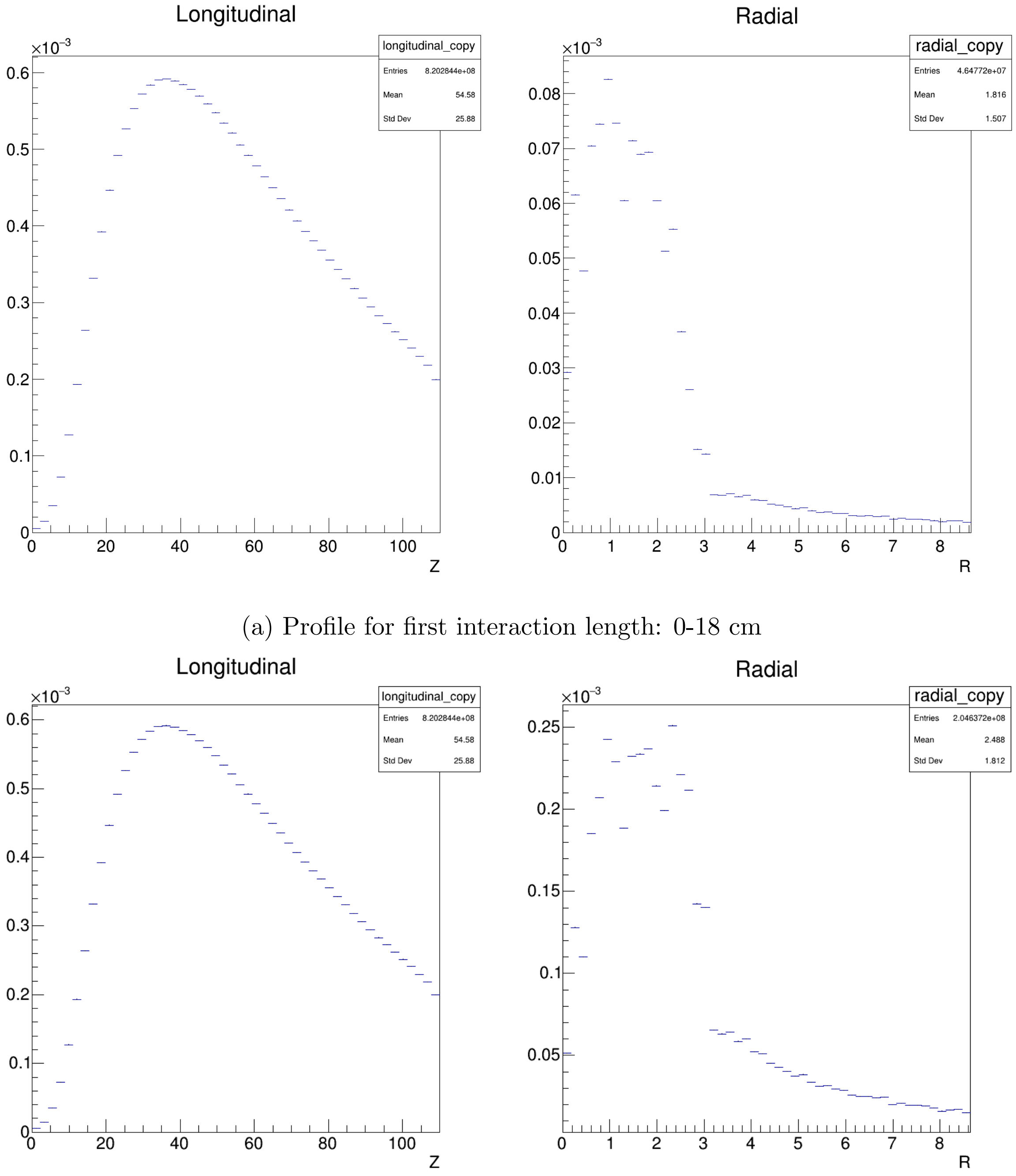
(b) Profile for fourth interaction length: 54-72 cm



(a) Profile for fifth interaction length: 72-90 cm

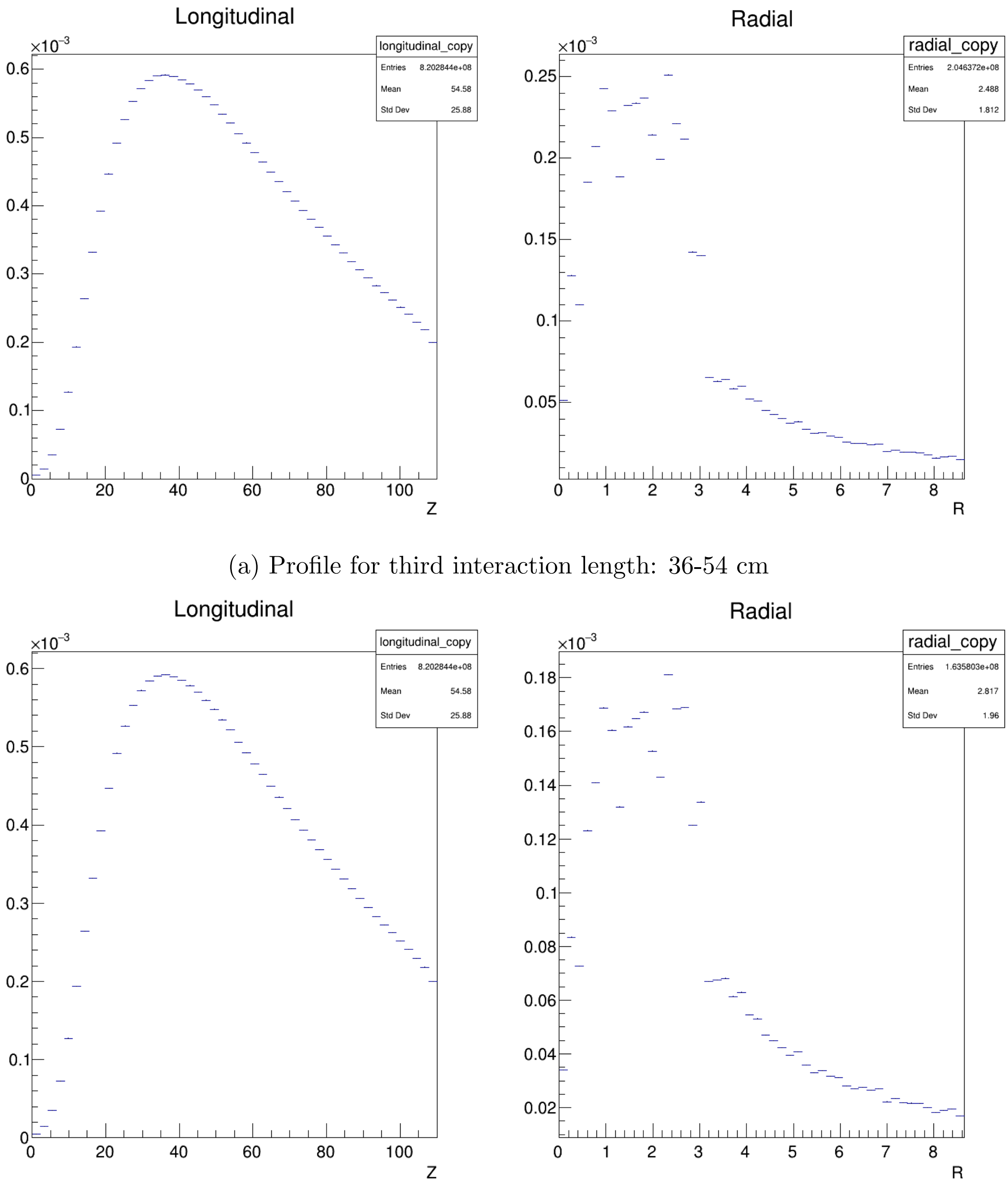
(b) Profile for final interaction length: 90-110 cm

A.1.7 Full profile 1 TeV without FoCal-E - vmc implementation



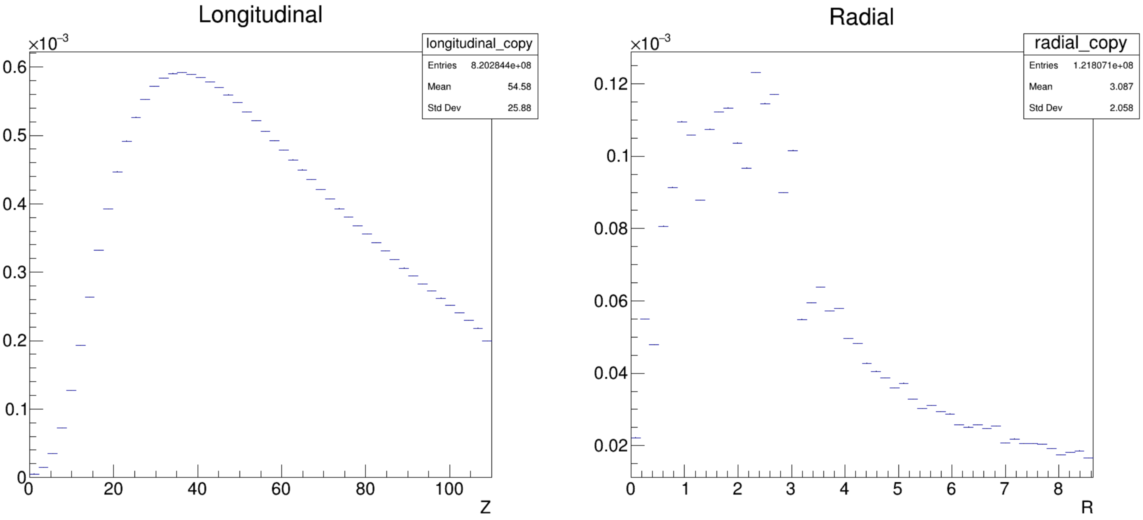
(a) Profile for first interaction length: 0-18 cm

(b) Profile for final interaction length: 18-36 cm

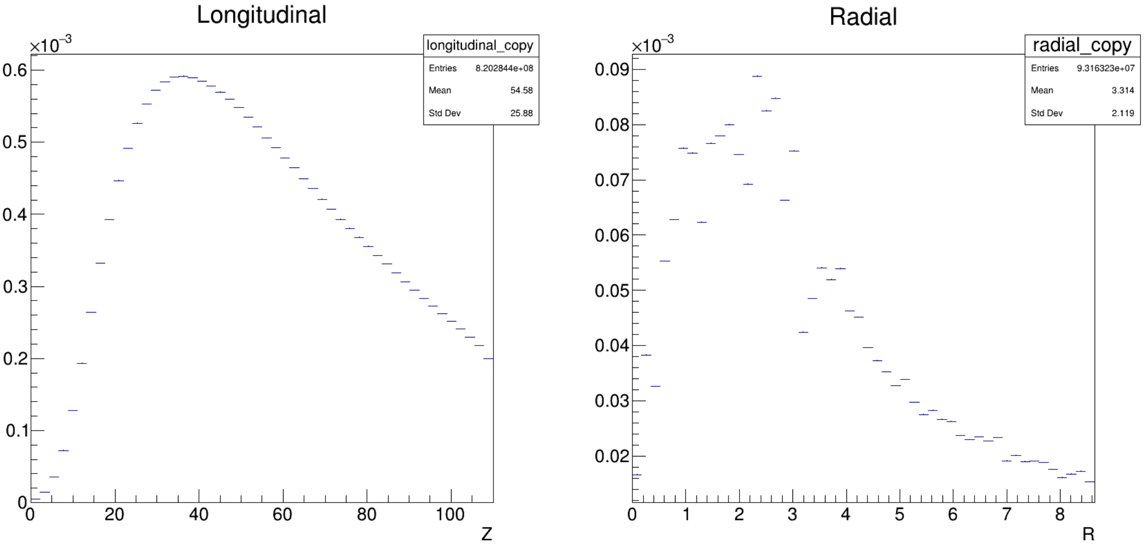


(a) Profile for third interaction length: 36-54 cm

(b) Profile for fourth interaction length: 54-72 cm



(a) Profile for fifth interaction length: 72-90 cm



(b) Profile for final interaction length: 90-110 cm

A.2 Visualized setups

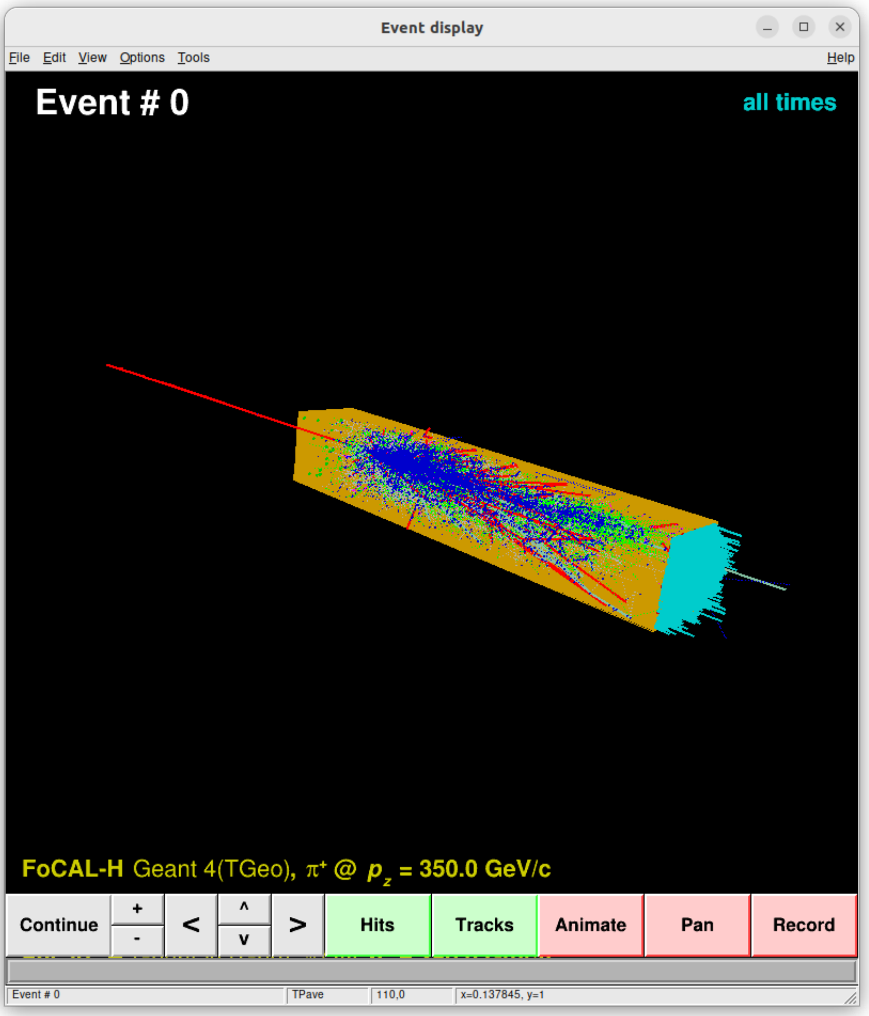
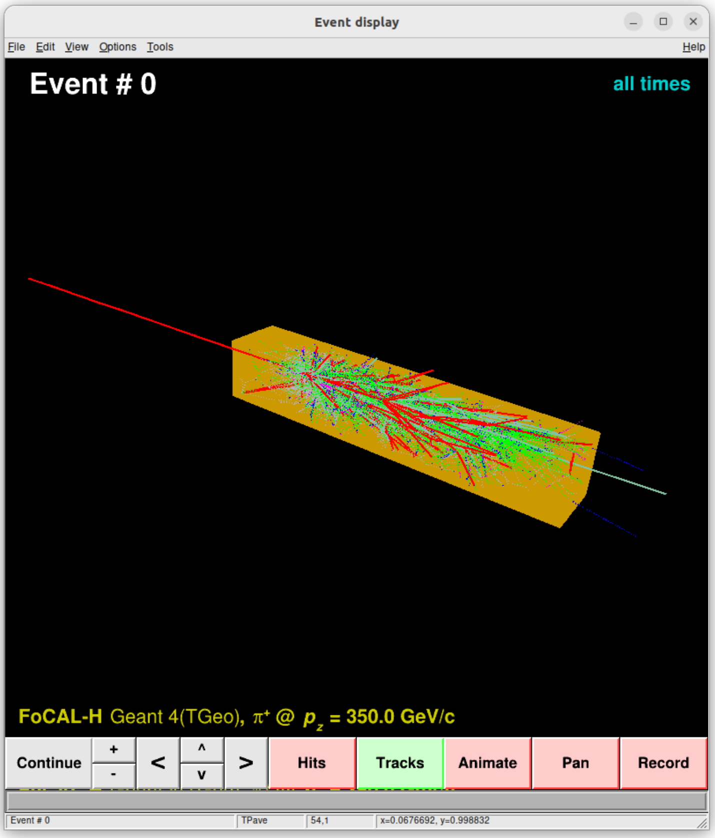
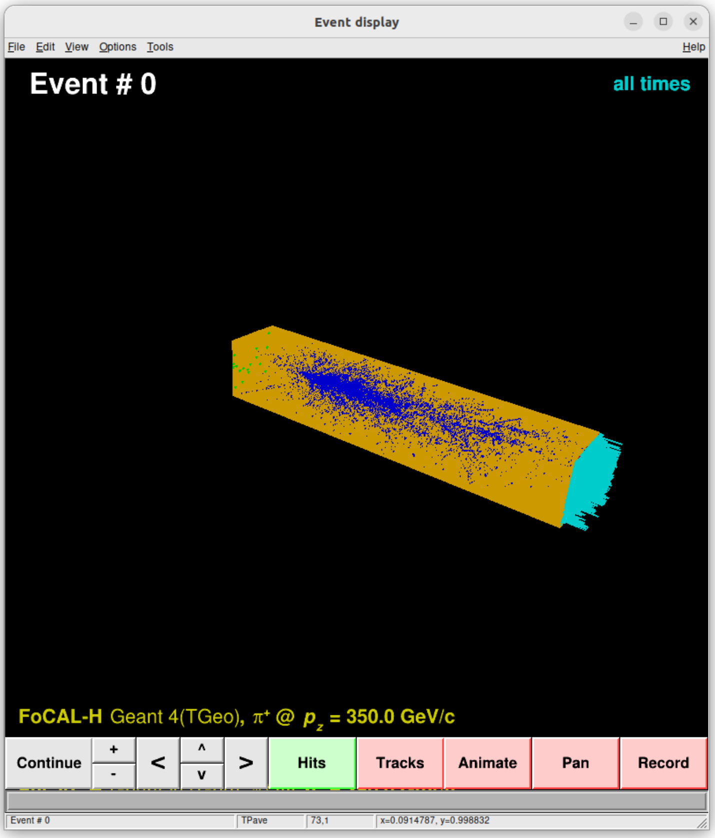


Figure 83: Full visualization of a 350 GeV event in the vmc simulation



(a) Visualization of the particle tracks within a 350 GeV event in the vmc simulation.



(b) Visualization of the hit detections within a 350 GeV event in vmc simulation

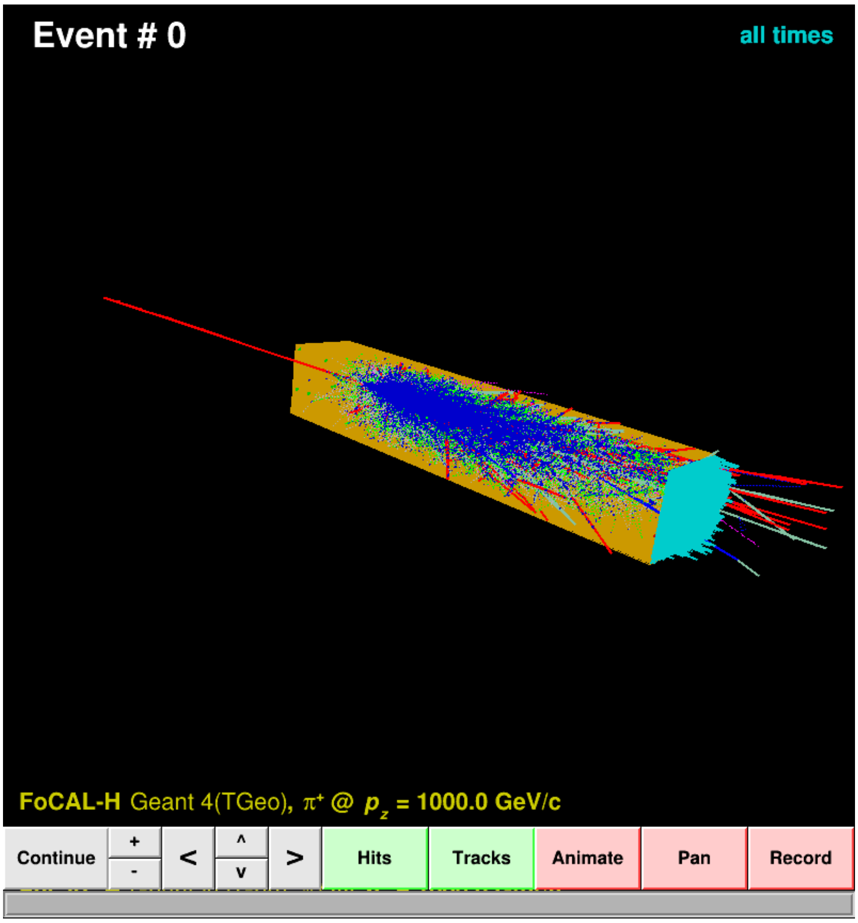
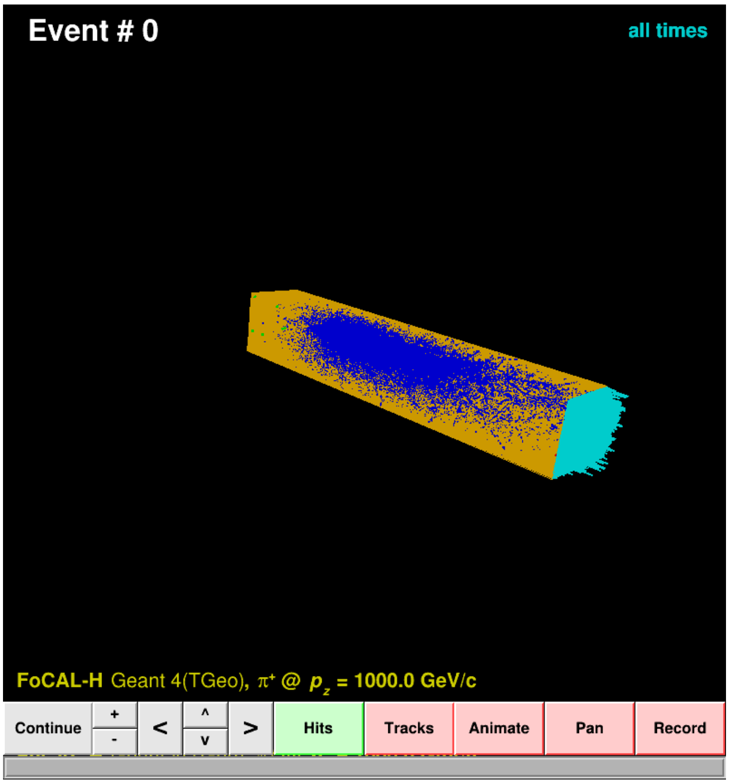
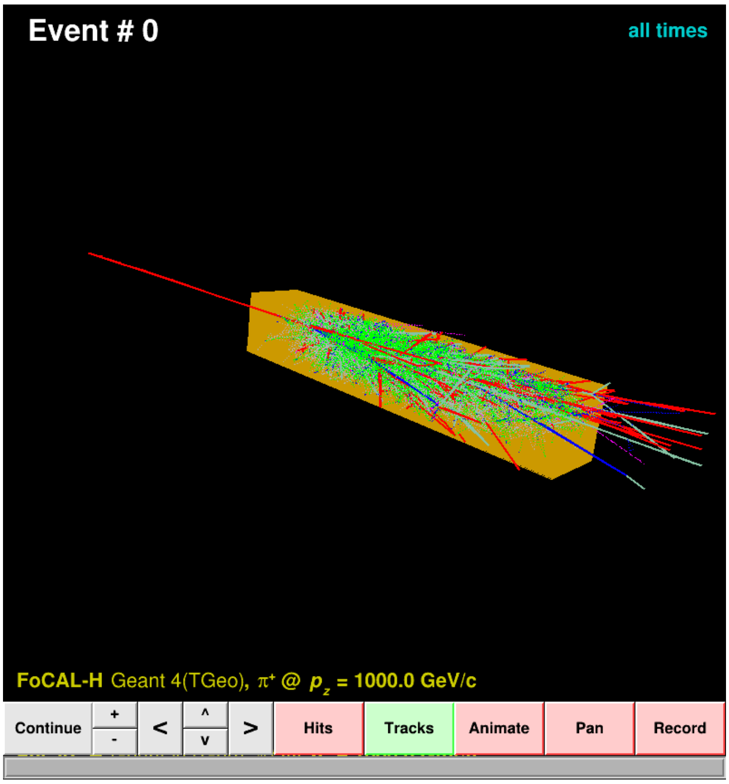


Figure 85: Full visualization of a 1 TeV event in the vmc simulation



(a) Visualization of the hit detections within a 1 TeV event in the vmc simulation.



(b) Visualization of the particle within a 1 TeV event in vmc simulation