Accounting for the Asymmetry: Boost Vector Measurement in Experiment 26 Data Sets by Belle II Detector

St. Francis Xavier University

Evan Hunt

Supervised by:
Dr. Hossain Ahmed

In partial fulfillment for the
degree of Bachelor of Science with Honours
in the Faculty of Science,
Department of Physics

18 March 2024

Declaration of Authorship

I, Evan Hunt, declare that this thesis titled, 'Accounting for the Asymmetry: Boost Vector Measurement in Experiment 26 Data Sets by Belle II Detector' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have clarified exactly what was done by others and what I have contributed myself.

Signed: Evan Hunt
Date: 18 March 2024

Abstract

Belle II, an international collaboration, is a high-energy physics experiment that uses the SuperKEKB collider at KEK in Tsukuba, Japan. SuperKEKB is an electron-positron collider that acts as a B-meson factory to investigate physics beyond the Standard Model (SM) and the precision measurement of the SM parameters. An asymmetric beam of 7 GeV electrons and 4 GeV positrons circulates in opposite directions, colliding at the centre-of-mass (COM) of Upsilon (4S) resonance. A consequence of the asymmetry is a discrepancy between the frame of reference in the laboratory and that of the centre-of-mass, which is moved in the collision. A Lorentz boost vector must be introduced to account for this for the sake of performing physics analysis. The goal of this project is to create a way to improve upon the boost vector values of the most recent run of the Belle II project, experiment 26.

Acknowledgements

I would like to thank my supervising professor, Dr. Hossain Ahmed, for his help guiding me as a researcher and as a student.

I would also like to extend similar thanks to each of the other professors and fellow students who have made my time at StFX Physics a great one.

I'd like to show my appreciation for the many services I've used in for this project: KEK, DIRAC and GRID, and Compute Canada have each been indispensable in their own ways.

A big thank you to NSERC Belle II fund for making letting me do this research over the summer and making this possible. As well as for funding my travel to the 2023 Belle II Summer Workshop held in Duke University.

And a final thank you to my family for being so supportive and proud of my education.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Belle II Collaboration is one of the leading high energy physics projects in the world. With institutions in 37 different countries and a membership greater than a thousand physicists strong, it is a large and global experiment. It is an upgrade to the original Belle project at the SuperKEKB particle accelerator (fig 1.2 housed at the High Energy Accelerator Research Organization, or KEK, in Tsukuba, Ibaraki, Japan. The particle smashing is done with beams of electrons and positrons. The high energy ring, HER, uses electrons and is set to 7.007 GeV and the low energy ring, LER, uses positrons and is set to 4.000 GeV[16]. This is a substantial improvement to the beam energies of the Belle project, had the HER set to 8 GeV and LER at 3.5 GeV. In both cases, the two beam energies are specifically chosen so that the centre-of-mass energy, $\sqrt{s}$, will reach about 10.58 GeV[1], which is the energy required for Υ(4S) resonance to occur and create the B-mesons, as seen in figure 1.1. Predictably, this asymmetry in the beam energies causes issues that need to be accounted for, but it is done out of necessity.

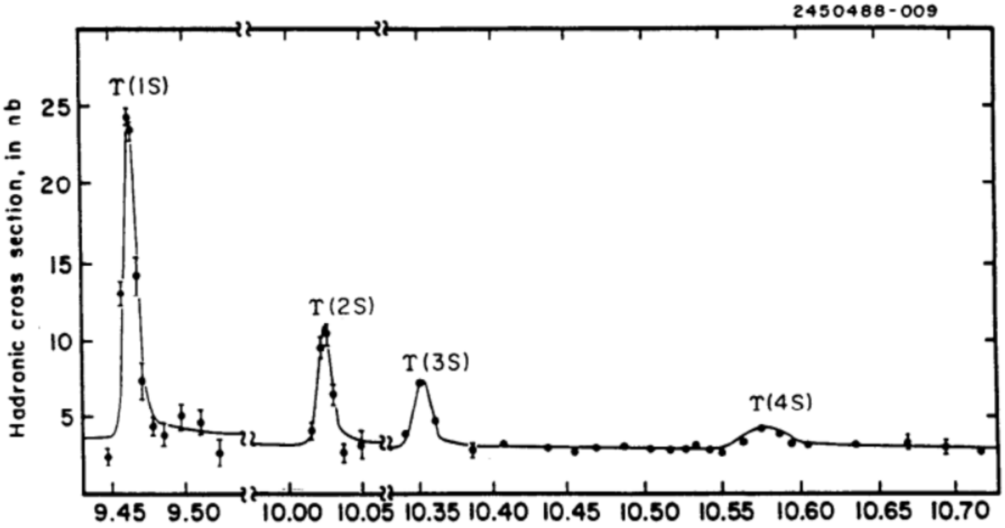

Figure 1.1: CLEO data showing the resonance energies in GeV required for Υ(nS) up to n=4[8]

Positrons, the antimatter counterpart of the electron, do not come quite as easily as electrons do. Creating antimatter involves an induced subatomic reaction. This is done through a process called β^+, or positron emission, where a high energy beam of electrons collides with radioactive material, like sodium-22. A proton is converted into a neutron, an electron neutrino, and most importantly, a positron[9]. It is worth noting that this is not proton decay. It involves a down quark decaying to an up quark, changing the proton into a neutron. The quark decay is what creates the positrons. SuperKEKB uses shoots a high energy electron beam at a 14-mm-thick piece of solid tungsten (fig 1.3) before being brought into the damping ring and cycled through until the positrons are the proper energy[21]. This process is costly, since it does not simply translate 1 GeV's worth of electron beam into 1 GeV of positron beam. There is substantial energy loss, and this must

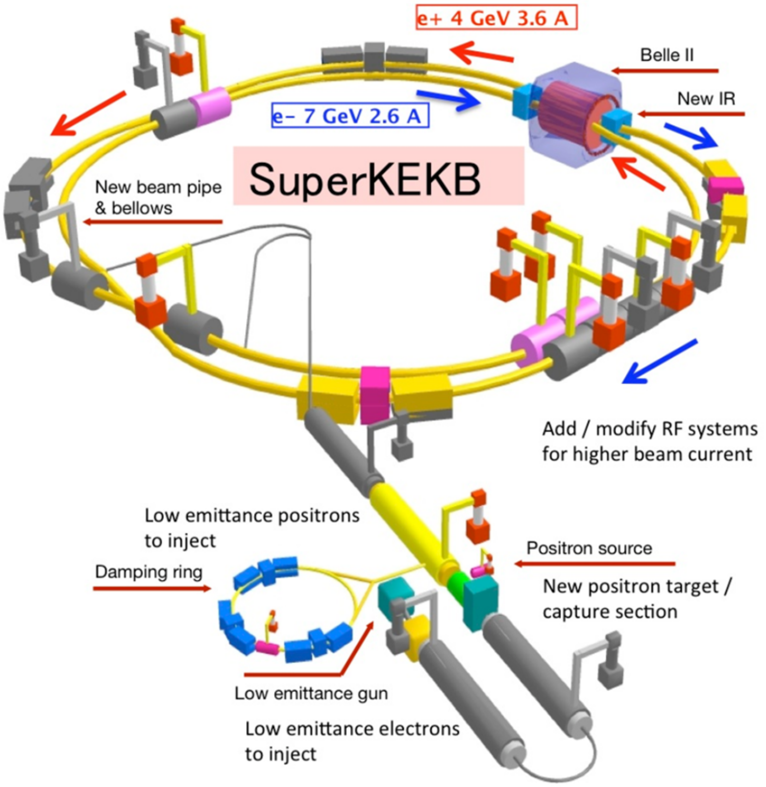

Figure 1.2: Diagram of the SuperKEKB accelerator[15]

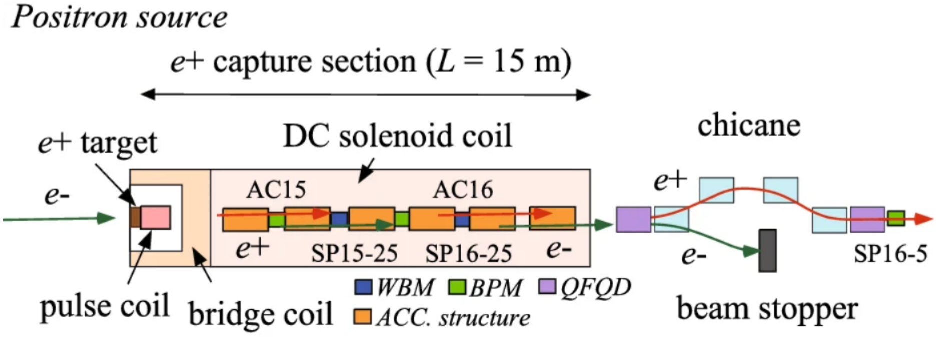

Figure 1.3: SuperKEKB's positron source[21]

be considered when designing an electron-positron collider.

Though in an ideal world the energies would be the same, the asymmetry is a necessity and must be handled for the sake of doing physics. The Belle II project, and other asymmetrical high energy experiments, keeps track of a special value to correct for this called the Lorentz boost vector, denoted by β. The goal of this research is to design a method of measurement and measure these boost vectors, with the purpose of improving upon the already present values.

Chapter 2

The Standard Model

The Standard Model of Particle Physics is a system of classification and organization of all known subatomic particles in the universe and three of the four fundamental forces that govern it, acting as a unification of those three[12]. One of the ultimate examples of collaboration in science, the Standard Model grew naturally and gradually over the last century of experimentation and discovery. Most of the account of the history of the development of the Standard Model is information taken from Abraham Pais[20].

Most of human history was a time without even an atomic model of matter. It was a theory posited by Democritus in the early centuries CE, according to the Greek historian Diogenes Laertius[19]. Coming from the Greek word atomos, meaning "indivisible," the atom was originally thought of as the fundamental piece of matter. One which cannot be split or broken apart. This was not a widely held belief by any stretch at the time of atomism's inception. Fast forward several centuries to the 1800s and an atomic form of matter was still not the consensus opinion. Modern particle physics and an atomic view of nature only truly began to take shape near the end of the 19th century and into the 1900s. In 1897, J. J. Thomson found the first reliable value for $\frac{e}{m}$ and two years later, the concept of the electron, the very first lepton, was formalized and considered discovered. The first real fundamental particle was finally found and thus the field of particle physics was inevitable

The Second particle to be discovered was the H-particle, H^+, later known as the proton. Now, unlike the first particle, the proton is not fundamental. Though not known at the time of discovery, by Ernest Rutherford, the proton is made of yet smaller particles. The same situation can be seen in the discovery of neutrons. Thus, the idea of the atomic nucleus is constructed in its modern form.

It is from here on that particle physics takes a sharp turn away from the realm of chemists and atomic physicists. Rather, there is an increased importance to 20th century innovations in quantum theory, high energy technology, and group theory. Though group theory is a mathematical field, it cannot be overstated how important it is to the Standard Model, primarily as it relates to gauge theory and symmetries. The mid to late 1900s were a busy time for subatomic physics. Quarks, more leptons, several other non-fundamental particles, and several vector bosons were all experimentally discovered with new high energy particle accelerators and detectors. This is also the point at which the Standard Model began formation as a codified model of the universe.

The Standard Model, as depicted in fig 2.1, separates each of the known fundamental particles into a few groups: The fermions, quarks and leptons, which comprise all matter and whose antimatter equivalents likewise comprise all antimatter. Along with the bosons, vector and scalar, provide the conduit by which matter and antimatter interact with itself and the other. Just as the Periodic Table comprises the building blocks for the universe, the particles of the Standard Model are the building blocks for the building blocks. They are, to current consensus, the bedrock of matter and antimatter. One may find it difficult to conceive of something not being "made" of something else, but fermions are simply bound states of energy and spin. And it is intuitive enough to understand how light particles, or photons, transfer energy. The photon is the vector boson that mediates electromagnetic interaction. The other bosons simply do the same for other types of interactions.

The first of the fermions, quarks, in traditional ordering are: Up, Down, Charm, Strange, Top, and Bottom, as seen in table 2.1. This order separates them into their generations, two to a single generation. The most common of the quarks are the first generation, the up and down. Two up with one down makes a proton and one up with two down makes a neutron. These are the components of the nucleolus of an atom of matter. Conversely, two anti-up with one anti-down makes an anti-proton. Likewise, the same pattern can be seen in the anti-neutron. The primary effective difference between these matter and antimatter equivalents is that the sign of the charge is flipped, though with the same absolute magnitude. The logic behind the generation separation has to do with the group theory mentioned earlier. SU(3) symmetry is directly related to the foundation of the quark model[14], but that is outside the scope of this description.

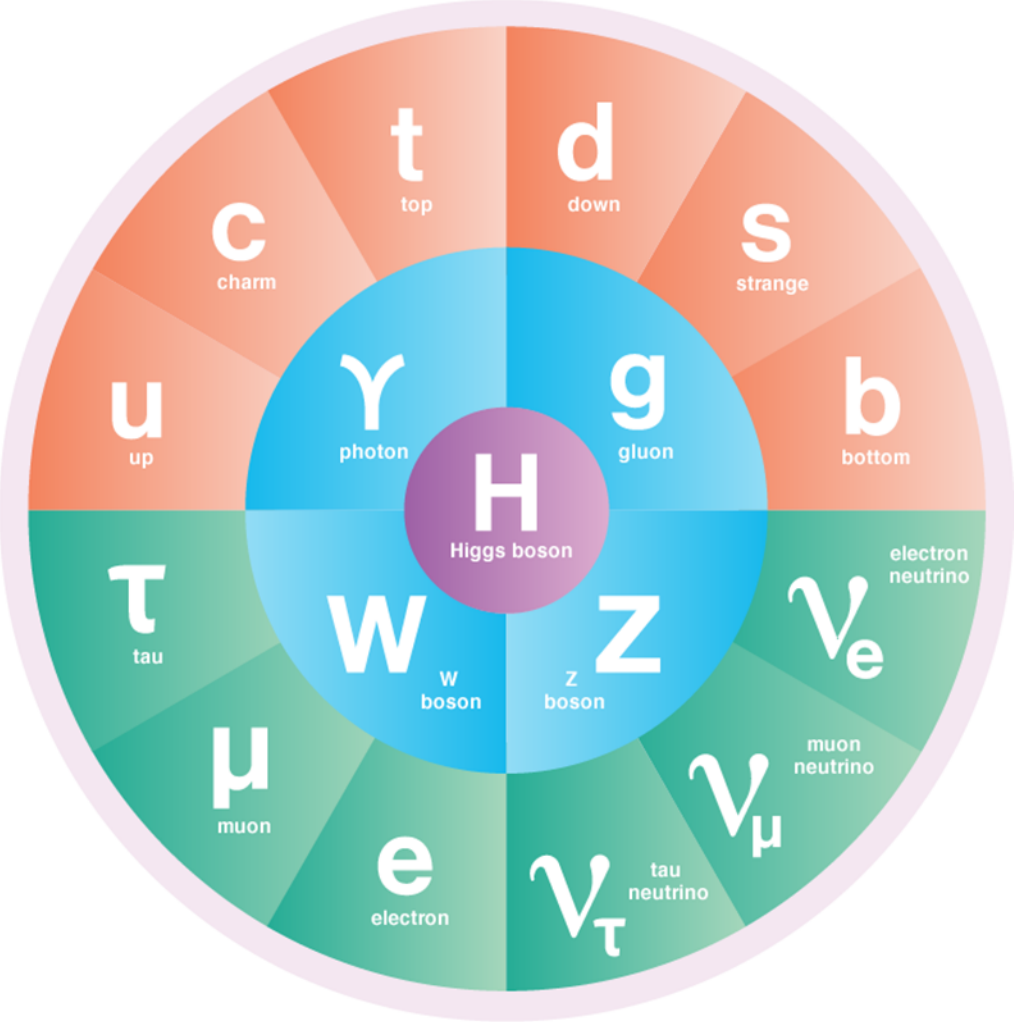

Figure 2.1: Visual representation of the different types of elementary particles in the Standard Model[10]

Table 2.1: Quark data[13]

Quark	Mass (GeV)	Charge (e)	Spin
Up (u)	0.00216	$+\frac{2}{3}$	$\frac{1}{2}$
Down (d)	0.00467	$-\frac{1}{2}$	$\frac{1}{2}$
Charm (c)	0.0934	$+\frac{2}{3}$	$\frac{1}{2}$
Strange (s)	1.27	$-\frac{1}{3}$	$\frac{1}{2}$
Top (t)	172.69	$+\frac{2}{3}$	$\frac{1}{2}$
Bottom (b)	4.18	$-\frac{1}{3}$	$\frac{1}{2}$

Table 2.2: Lepton data[13]

Lepton	Mass (MeV)	Charge (e)	Spin
Electron (e)	0.51099895000	-1	$\frac{1}{2}$
e Neutrino (ν_e)	< 0.0000008	0	$\frac{1}{2}$
Muon (μ)	105.6583755	-1	$\frac{1}{2}$
μ Neutrino (ν_μ)	< 0.19	0	$\frac{1}{2}$
Tau (τ)	1776.89	-1	$\frac{1}{2}$
τ Neutrino (ν_τ)	< 18.2	0	$\frac{1}{2}$

The other group of fermions are the leptons (table 2.2). It was mentioned before that the electron is a lepton, but so are the muon and the tau. Each of these have an associated generational family member called a neutrino: the electron neutrino, the muon neutrino, and the tau neutrino. Though the quarks tend to collect and create larger particles called hadrons, the leptons do no such thing. Because the positive matter leptons are all negatively charged or neutral, they do not have the tendency to attract. The lowest energy states of electrons are the ones in which they are the farthest apart.

The other side of the elementary particles are the bosons, starting with the vector bosons (table 2.3. They are the gluon, the photon, the Z^0 and the $W^\pm$. The gluon is the "glue" that holds the strong interaction, which is the force that maintains quark bound states, including the bound states of bound states, like the atomic nucleus. The photon mediates the electromag-

Table 2.3: Boson data[13]

Boson	Mass (GeV)	Charge (e)	Spin
Gluon (g)	0	0	1
Photon (γ)	0	0	1
Z^0	91.1876	0	1
$W^{\pm}$	80.377	±1	1
Higgs (H)	125.25	0	0

netic force, which is the force between electrically charged particles. The Z^0 and $W^{\pm}$ concern the weak interaction, which is what is seen during nuclear decay. The Standard Model acts as a unification of these three fundamental forces. This leaves out the fourth, gravity, to be a part of Physics Beyond the Standard Model, or BSM. The graviton is, as of now, the only purely theoretical mediation boson for the force of gravitation, though it is usually considered in its own tensor boson category. The other boson is the singular scalar boson, the Higgs boson. It facilitates the mass of everything in the universe. Being the most recent new subatomic particle, it was discovered in 2012 at CERN's Large Hadron Collider[2].

That was all of the fundamental subatomic matter particles. Each of these either have antimatter counterparts or are their own counterpart, such as the anti-Z boson being the matter Z. The Standard Model is the most complete model of the universe we have, but as already stated, there are blind spots. In other words, it is not the "theory of everything." Gravity, dark matter, dark energy, neutrino oscillations, and more are all things we can observe or know must exist but are simply unaccounted for in the Standard Model. Current experiments, such as the Belle II collaboration are being done in the effort to progress BSM theories.

Chapter 3

Belle II Detector

The Belle II project is an upgrade of the previous Belle project at the KEKB facility in Japan[11]. It speeds up a beam of electrons and positrons in the SuperKEKB accelerator until they are led to meet and collide at the interaction point at the centre of the Belle II detector (fig 3.1) for which the project was given its name. The intent of this section is to give an overview of the functionality of the Belle II detector and the data taking methods that it uses. All technical information is from either the internal Belle II wiki or the publicly accessible Belle II design document[4]. Keep in mind that throughout this section I will be talking about particle identification (PID) and the data acquisition (DAQ) process simultaneously. These processes do not happen at the same time. DAQ occurs during the accelerator run and PID occurs during analysis after the fact. Until the point of reconstruction during analysis, the data taken is simply raw event data and nothing more.

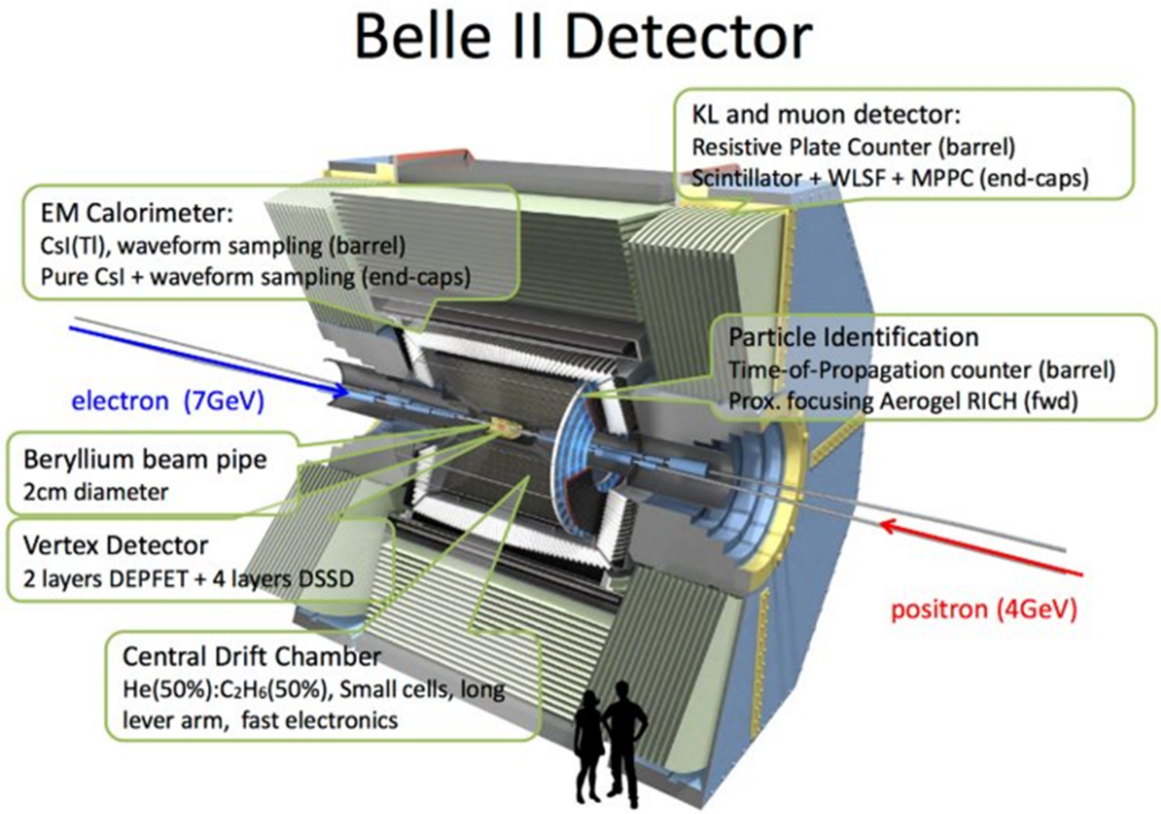
Belle II Detector
KL and muon detector:
Resistive Plate Counter (barrel)
Scintillator + WLSF + MPPC (end-caps)
EM Calorimeter:
CsI(Tl), waveform sampling (barrel)
Pure CsI + waveform sampling (end-caps)
electron (7GeV)
Particle Identification
Time-of-Propagation counter (barrel)
Prox. focusing Aerogel RICH (fwd)
Beryllium beam pipe
2cm diameter
Vertex Detector
2 layers DEPFET + 4 layers DSSD
positron (4GeV)
Central Drift Chamber
He(50%):C2H6(50%), Small cells, long
lever arm, fast electronics

Figure 3.1: The Belle II detector[6]

3.0 The detector structure

The following is an itemized list in order of innermost to outermost component of the detector:

1. Beam Pipe

2. Pixel Detector (PXD)

3. Silicon Vertex Detector (SVD)

4. Central Drift Chamber (CDC)

5. Time of Propagation (TOP)

6. Aerogel Ring Imaging Cherenkov (ARICH)

7. Electromagnetic Calorimeter (ECL)

8. K_L^0 and μ Indentification (KLM)

3.1 Beam Pipe

The beam pipe is the innermost chamber of the detector and where the detector connects to the SuperKEKB accelerator. Note that the detector exists separate from the accelerator and can be moved. The pipe is 20 millimeters across and double walled with a beryllium casing. This is the location in which the interaction point (IP) of the electron and positron beams is located. No data is taken within the beam pipe.

3.2 Pixel Detector (PXD)

The Pixel Detector is one of two components that comprise the Belle II Vertex Detector (VXD), along with the Silicon Vertex Detector. It is also the innermost subdetector with the Belle II detector. It is responsible, though not solely, for the reconstruction of "primary and decay vertices[7]." What that means is that when one is preforming an analysis, the PXD carries a large part of the credit for collecting the data about what exactly the particles in an event are.

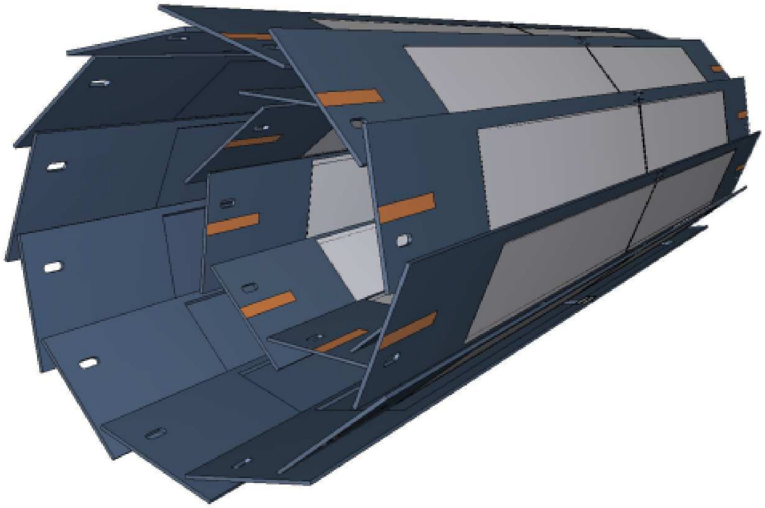

Figure 3.2: The Pixel Detector[4]

The PXD, as shown in figure 3.2, conducts the initial tracking of the resultant particles and finds which pixels in the tracking data are worth saving and which can be discarded. The PXD consists of two layers of DEPFETs, depleted field effect transistors. When a charged particle travels through the transistor it causes a modulation in the drain current and is considered a "hit." A collision of the magnitude of the Belle II project's will inevitably cause quite a bit of noise, though not as much as colliding two larger particles. The DEPFETs can do a reading several times per hit and weed out much of the noise. That is why DEPFETs were chosen: for the good signal-to-noise ratio.

The sections of layers of DEPFETs are called the PXD's "sensors" and each single transistor is one pixel of the sensor. The pixel sizes are (in μm) 55×50 and 60×50 on the inner layer, which consists of 8 sensors, and 70×50 and 8×50 in the outer layer, which consists of 12 sensors. All in all, there are a sum total of 7.68×10^6pixels in the PXD.

3.3 Silicon Vertex Detector (SVD)

The Silicon Vertex Detector is the other of the two components that make up the Belle II Vertex Detector. The SVD is directly outside of the radius of the PXD and is the next step in the particle tracking process. Its purpose is much the same as the PXD, but it uses 4 layers of increasingly more sensors. The sensors are Double-sided Silicon Strip Detectors. Strips of silicon are connected to a circuit for the purpose of readout. It keeps track of yet more hits of charged particles. During analysis it provides the use of being the intermediary placed between the inner VXD and outer CDC. Its hit finding efficiency is nearly 100%, meaning it is a reliable tool for analytical reconstruction of events.

By the outermost layer of the SVD, it is 270 mm across.

3.4 Central Drift Chamber (CDC)

The Central Drift Chamber, as seen in figure 3.3, is the next step in particle tracking and is also independently important for some particle identification even before reaching the PID. It provides four main functions. First, it concludes the generalized path reconstruction for charged particles that the Pixel Detector and Silicon Vertex Detector have been doing thus far. In doing this, the CDC measures the momenta of the particle. Second, it determines the energy lost in its gas and provides some crucial particle identification

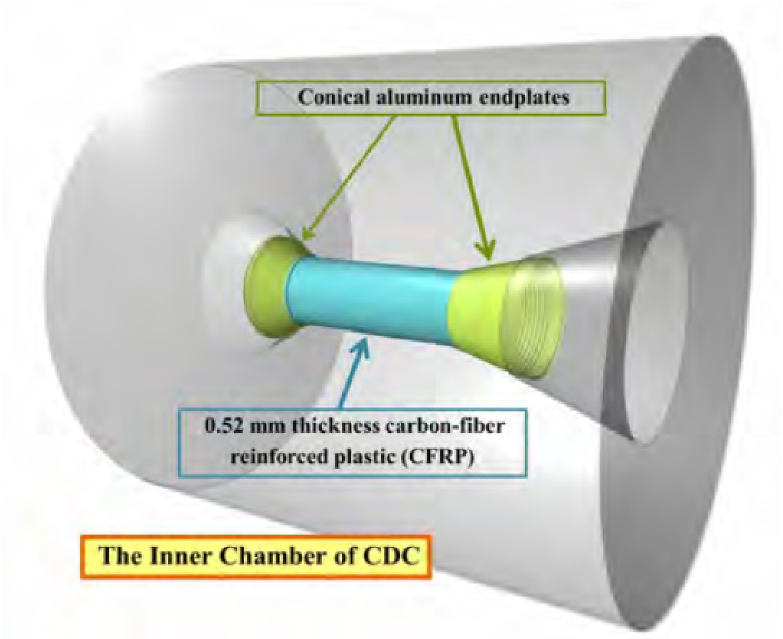
Conical aluminum endplates
0.52 mm thickness carbon-fiber reinforced plastic (CFRP)
The Inner Chamber of CDC

Figure 3.3: The Central Drift Chamber[5]

information. Third, some particles do not make it out of the CDC, but the CDC itself can provide enough information to accurately identify such particles. And finally, it provides signals that the relevant information is in fact useful. This last one is very important and is known as "trigger" signals. High Energy Physics generates vast seas of data, but not all of it is important. Despite the PXD's transistors' campaign against noise, which is very successful, more must be done to minimize the storage of useless data. The CDC's triggering minimizes it further.

The CDC's drift chamber uses the gas $He-C_2H_6$. In plain words, the gas used is a mixture of 50% helium and 50% ethane. The way the CDC records the track paths of the charged particles is through the particles "drifting" through the gas inside the chamber. Helium and Ethane is used because of the following advantages over all other gasses explored: “low radiation length, good position resolution, good energy loss resolution, low cross section for synchrotron radiation X-rays, and little radiation damage.”[4]

3.5 Time of Propagation (TOP)

The Particle Identification (PID) section of the hardware consists of two sections, the first being Time of Propagation Counter. The TOP is a Ring Imaging Cherenkov (RICH) detector. The purpose of a RICH detector is to identify charge subatomic particles. It uses the Cherenkov radiation emission of said particles to characterize and accurately identify them. Cherenkov radiation is the electromagnetic radiation a charged particle puts out when it travels through a dielectric radiator at a speed greater than that of the phase velocity of light through the material. The TOP uses a quartz radiator, specifically synthetic fused silica, situated in an array structure around the CDC. The Cherenkov photons are internally reflected through the quartz medium until it reaches a photon detector. It is then traced back to the point it hit in the CDC and the particle, identified through the path and the characteristic Cherenkov radiation, is given precise coordinates. In the PXD, the location of the particles is given on the pixel level. This new coordinate is significantly more exacting.

The TOP's statistics on discriminating between kaons and pions is 99% with a less than 0.5% chance of a pion false flag.

3.6 Aerogel Ring Imaging Cherenkov (ARICH)

The second of the Particle Identification section of the detector is another Ring Imaging Cherenkov detector, like the TOP. This RICH detector uses Aerogel as its radiator. The reason behind this second RICH detector has to do with the difference in the radiator used. The aerogel is a lower density than the quartz and therefore can be used to detect a much broader range of momenta of particles. The increase in range does unfortunately mean a slight decrease in accuracy of identification, justifying the presence of both RICH detectors.

The ARICH's statistics on discriminating between kaons and pions is 96% with a 1% change of a pion false flag.

3.7 Electromagnetic Calorimeter (ECL)

The Electromagnetic Calorimeter is a standard feature of any particle detector. Its uses are particularly numerous. It is responsible for detecting the many photons generated in the course of the collision and decays with high efficiency. Not only does it detect them, but the ECL also measures the energy and angular data of the photons. Aside from photons, it detects and identifies electrons as well. Much like the CDC, the ECL is also responsible for providing accurate triggering to filter out the electromagnetic background. The final particle it is responsible for is one in which the ECL shares detection and identification duties with the KLM subdetector, the K_L^0, which will be expanded upon in the KLM section. Aside from particle information, the ECL also collects data regarding the luminosity of the detector. Luminosity, in the context of high energy physics, is the amount of data acquired in units of inverse barns by inverse seconds, $b^{-1} \cdot s^{-1}$, though normally femtobarns is what is used. The unit barn is equivalent to 100 fm^2. The integrated luminosity, or the net amount of data obtained in a given amount of time is commonly used rather than the instantaneous luminosity and simply uses units b^{-1}. In different terms, luminosity is the number of events detected as related to time and cross-sectional area. Mathematically, this is expressed as the luminosity, $\mathcal{L} = \frac{N}{\sigma}$, where N is the number of events per second and σ is the cross-section.

The ECL detects through the use of a scintillator detector. Scintillation is when an incident particle goes through a scintillator material, it emits low energy photons. These photons are focused on a photocathode, which sends out photoelectrons by way of the photoelectric effect. The photoelectrons move through a photomultiplier tube, in which the electrons hit powered electrodes called dynodes. Every time a photoelectron hits a dynode, it creates more photoelectrons. Eventually, the many photoelectrons hit an anode which is connected to an electronic setup for reading the pulse of current that the final hit of photoelectrons gives. The net effect is that the current pulse strength is dependent on the energy of the incident particle and thus there is a way to identify the particles that go through the ECL.

The scintillation material that the ECL uses is cesium iodide, CsI, which is activated with thallium, Tl. The thallium, which is added to the CsI crystal lattice during production, enhances the scintillating ability of the material. CsI(Tl) was chosen because of its excellent abilities in photon output, the short radiation length of each photon, the cooperative mechanical properties, and of course the reasonable price.

3.8 K_L^0 and μ Indentification (KLM)

This final and outermost subdetector of Belle II serves a specific goal of identifying neutral kaon-long particles, K_L^0, and muons, μ. It consists of many alternating thick iron plates and detecting elements. These two particles are particularly difficult to catch and so there needs to be a lot of attempts made with very specialized equipment. The iron plates serve two main purposes: to act as a magnetic flux return for the solenoid between the ECL and KLM, and to get the kaons to shower, or decay into many smaller particles. The kaon can then be detected and identified by the resulting shower, as well as the data from the ECL scintillators. The muons are found through pockets of gas in resistive place chambers. When a muon passes through and interacts with the atoms in the gas, it knocks electrons out of their orbitals. It is possible to detect and identify muons because of this. The gas used is a mixture of "62% HFC-134a, 30% argon, and 8% butane-silver (the latter being a mixture of 70% n-butane and 30% isobutane)[4]." The KLM also has endcaps on both ends of the detector, with more scintillation detectors.

Chapter 4

Building Up to Belle II Analysis

This chapter is the equivalent to a description of building an apparatus in which to conduct experiments. Belle II is a relatively global endeavor, particularly when compared to non-high energy experiments. This means that there needs to be robust systems in place for many users all over the planet and affiliated with different institutions to interact with the Belle II data. On the other side of the coin, this also means that there must be cooperation, and automated cooperation between lots of different organizations which must retain the privacy of both user data and scientific data. What this ends up meaning for an aspiring user of the Belle II analytical systems is many steps of registration across several sites and services and many weeks of bureaucratic red tape before any analysis can begin.

First, registration for a KEK account is required for access to the KEK facilities in any work context and, more importantly for this, access to computing facilities. KEK is the High Energy Accelerator Research Organization. Sidenote: the name comes from the Japanese name of the organization, 高エネルギー加速器研究機構 [Kō Enerugī Kasokuki Kenkyū Kikō]. This is the physical place in Tsukasa, Ibaraki, Japan where the Belle II experiment is conducted. The signup process has a few complications, but a guide is provided on the DESY confluence page. With this in hand, the sign-up process is simple but requires several weeks of waiting for processing and confirmation for this first step alone. Unfortunately, nowhere does it say how long the wait will be, so there was a not insignificant amount of confusion involved.

While the weeks of confusion about the KEK account wears on, the next task is luckily completely separate from the KEK account. To be considered a working member of the Belle II collaboration one needs an account with the Belle II Membership Management System (B2MMS). This provides access to the Belle II collaborative services (B2CS), which are not hosted by KEK themselves in Japan, but rather by Deutsches Elektronen-Synchrotron, or DESY, in Germany. One can either enter this by invitation or by a provided certificate. As an undergraduate student, I was given an invitation by my supervising professor. A temporary username and password are provided in an email. When logging in with that, the new user is given the opportunity to create a proper username, but not a password and finish their registration. It is only later, when the new account is processed, which happens quite quickly, when there is another given password that is meant to be immediately replaced with a new password. After all of that, there is now a new Belle II user with access to all the internal services and documentation.

The next step is to apply for access rights to the Belle II computation and data servers, but that involves finishing the long wait for the KEK registration processing. But after that's resolved and the acceptance email has been received, it is time to move on to CCPortal. CCPortal is KEK's computation manager. In it, there are the options for requesting access to the required KEK access and work servers. The registration process for CCPortal is unfortunately quite complicated. Many fields are exclusively in Japanese and require a high level of reading skill to parse everything. Luckily, there is a helpful comment in the Belle II Newcomers page with the proper details for many of those fields. If the KEK account has been fully accepted into the member list for the year, then the CCPortal account should be accepted within a working day. After that, the CCPortal options will be completely opened. To be able to access the KEK servers, one needs to go to the KEKCC section of the Service List and apply for both the "KEKCC Access Server" and "KEKCC Work Server" with the "b2_belle2" working group selected. Once that has been accepted, finally the Belle II analysis framework will be open.

Though analysis is ready to happen at this point, it is only in a very limited form. The next piece of the puzzle is mass computing by way of the GRID and DIRAC. This is a long and winding path even more than everything that comes before it. The first thing to be done is obtain a GRID User Certificate. In Canada, this can be done from the GRID Canada website. After going through the slightly obtuse requesting process, there will be an email sent out saying that the requestee must sign up for a Digital Research Alliance of Canada (CCDB) account, which requires the account verification number of someone else's account. Once the account is accepted, the account number must be sent out to the GRID certificate representative, and they must manually verify and accept it. Eventually, they will get back with a link to the GRID certificate download page. Next, the certificate must be imported into the web browser. This is important for using the web-based GRID interface, DIRAC. The certificate comes in a .p12 file format. All web browsers should have an "import certificate" button or equivalent in the settings tab. It's as simple as selecting the .p12 file. The next step is yet more registration. To access DIRAC, an account with KEK VOMS is required. Another registration process ensues until another waiting period. After about a week, an email will be sent saying that the VOMS account is registered, and that DIRAC account has been registered under that VOMS account and certificate. Now, DIRAC is available for use. But without the ability to use the GRID in the KEK work server, its not particularly useful. So that is the next step, getting the GRID certificate into the work server and ensuring that it is in readable state for the GRID system. Keep in mind that the KEK servers are only accessible through the Linux kernel terminal, so all the following things described are done with bash commands. After getting the .p12 file into a reachable location by the Linux terminal, the "scp" command must be used to transfer the file into the access server and then the work server. The certificate file must be held in a hidden folder called "/.globus". Using the "openssl" command, the usercert.pem (GRID certificate data) and userkey.pem (proof of ownership) can be extracted from the .p12 certificate file. Now, finally the mass computing GRID is able to be used to the most of its abilities. Additionally, since getting a CCDB account is part of the process for gaining the ability to use the GRID, ACENET and

Compute Canada's servers come for free! Compute Canada has very robust systems in place, including ROOT analysis functionality, and none of the input lag of KEKCC, so it is a very useful tool in tandem with GRID and KEKCC.

Chapter 5

Belle II Analysis Software Framework

The Belle II Analysis Software Framework (basf2) is incredibly robust and infinitely deep. In the general scheme of high energy physics, basf2 is responsible for being the middleman between raw data and simulation to useful numbers for analysis[18]. Basf2 also includes the necessary tools for analysis, making it a fully complete machine for doing physics. Reconstruction of the events, complete with particle identification, and the process of skimming down the data unwanted by the user and the subsequent analysis of the data are all handled by the same framework. It is a wide web of choices that takes a lot of experience and concentration to get grip on. All this to say, I simply have not had the time to be able to claim “expert.” So, I can only give the general run down on my own basf2 experience.

There are several versions of basf2 to choose from. It is not simply a matter of which is the newest one, though that certainly plays a part. A pre-release version would only be useful for testing that particular version of the software, as it has been outclassed by future full release for the purpose of doing analysis. The two primary streams of basf2 versions are the full and light releases. For most people's everyday analysis needs, the light releases are the superior choice, as they are more generally stable and the lighter weight on the system means faster computation times. Though for large scale amounts of data or complex algorithms, the full version may suit the demands of the scientist better.

The usage of basf2 is done through command line. First, the version desired must be sourced. This is the bash code equivalent to extending a class in more modern programming languages, where all the functions of that sourced file become default usable commands in the text line. The primary functionality of basf2 at this pre-analysis stage is to facilitate the running of "steering files," which are written in a more standard coding language (usually Python 3). They are what determines what gets cut from the large, raw .root data files and processed into n-tuples also in the form of .root files, which are data tables that contain everything that passes the cuts in the steering file. The steering files use a "path" in which "modules" are attached and data files are passed, such as the Rest of Event (ROE). Steering files most often begin with imports followed by

```
main = b2.Path()
ma.inputMdstList(environmentType='default',
filelist=file_list, path=main)
```

Using the VariableManager library, which is a part of basf2, it is possible to do calculations using the "formula()" function, attach a variable name to them, and then read them out as part of the resulting n-tuple .root file. These files are a crucial part of the analysis process in high energy physics. ROOT is a C++ based analysis focused language created by the European Organization for Nuclear Research, more commonly known as CERN. ROOT is standard across the field and Belle II is certainly no exception. Though ROOT is conducted in C++, basf2 allows for the use of a Python based usage of it. Extremely robust, it can do everything the original C++ can do while mixing with the standard Python usage of basf2's own functions. This is not to say that using C++ ROOT is not useful. Some things, such as macros for plotting, are suited for C++ far more than PyROOT.

On to gbasf2, which is the version of basf2 dedicated to interacting with the GRID. This is where the truly big data comes into play. Using the DIRAC frontend UI, it is easy to find an astonishing amount of raw data just waiting to be used. KEK's locally hosted data is not the full dataset, though it is representative of the whole and is still useful. Gbasf2 is primarily what I use for processing very large amounts of MC simulation data. Rather than being done on KEK servers, like basf2 processes, gbasf2 takes advantage of the mass computing centres available for the Belle II project all over the world. Jobs are submitted using essentially the same steering file, though slightly modified for the sake of compatibility. Typically, in my experience, the jobs involve many data files and are processed en masse, and the processing responsibilities are spread around according to space and availability. After the processing is completed, the resulting .root files must be downloaded from the GRID. This normally takes quite a while if there are a lot of files.

Though not directly affiliated with KEK, I would be remiss to not mention

Compute Canada, a nationally local computing hub. It has proved integral to my workflow. With none of the input lag seen on the KEKCC servers, and access to C++ ROOT, I will normally send my processed files directly over to Compute Canada through ssh tunneling. All of the plots seen in the following chapters were made using Compute Canada services.

Chapter 6

Boost Vectors and Analysis

When the high energy electron beam and low energy positron beam collide, they are asymmetrical and create a shift in the center of mass system (CMS) frame of the collision. This decouples the CMS and lab frames of reference and make them unequal. This has the devastating consequence of incurring enough error in analysis results that they are effectively useless. The easy solution would be to just perform a simple basis transformation and get them back in order, but the relativistic speeds cause a wrinkle. The tool we have to remedy this is the Lorentz transformation. A relativistic frame of reference transformation that will come in the form of the matrix:

$$\begin{pmatrix} \gamma & -\beta\gamma & 0 & 0 \\ -\beta\gamma & \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{6.1}$$

The γ factor is

$$\gamma = \frac{1}{\sqrt{1-\beta^2}} \tag{6.2}$$

The more imperative value is the β, or the boost vector,

$$\beta = \frac{v}{c} \tag{6.3}$$

From this, we can conclude that ratio between the velocity of the center of mass's shift in the lab frame and the speed of light in a vacuum, which is the same in all inertial reference frames, represents the boost of the system. Now, v is something we do not have direct access to and so another method of finding this same value is required.

β can alternatively be expressed as the ratio between the total momentum of the system in the lab frame and the energy of the center of mass system.

$$\vec{\beta} = (\beta_x, \beta_y, \beta_z) = \frac{\sum \vec{p}_{Lab}}{\sum E_{CMS}} \tag{6.4}$$

Luckily, these are values that the Belle II detector can obtain.

The four-vector of the moment is of the following form:

$$p^{\mu}_{Lab} = (E, p_z, p_y, p_x) \tag{6.5}$$

Because the z-axis is the axis of propagation, the y- and x-axes can be largely neglected in this calculation.

$$p^{\mu}_{Lab} = (E, p_z, 0, 0) \tag{6.6}$$

E is the total energy of the center of mass, the two beam energies added together. The z-component of the momentum can be expressed as the difference between the energy of the high energy electron beam and that of the low energy positron beam.

$$p^{\mu}_{Lab} = (E_{HER} + E_{LER}, E_{HER} - E_{LER}, 0, 0) \tag{6.7}$$

This is the final form of the momentum in the lab frame.

The system energy in the center of mass system can be calculated using the formula:

$$E_{CMS} \approx \sqrt{2E_{-}E_{+}(1 + \cos\alpha)} \tag{6.8}$$

Where E_{-} and E_{+} are the low and high beam energies respectively and the α factor is a correctional term that accounts for the non-negligible curvature of the beam movement within the accelerator loop. For the SuperKEKb accelerator, $\alpha = \frac{83}{1000}$ radians, or about 4.7 deg. This is the crossing angle of SuperKEKB[3].

$$E_{CMS} \approx \sqrt{2(7GeV)(4GeV)(1 + \cos\frac{83}{1000})} \approx 10.583GeV \longrightarrow \Upsilon(4S) \tag{6.9}$$

What we get is the center of mass energy of $\Upsilon(4S)$ resonance, which is to be expected.

Armed with this new formulation of β, it is now possible to find some values, which requires data. The Belle II detector data is contained in the KEK databases and needs the Belle II Analysis Software Framework to be processed. The Belle II experiment creates B-meson pair which very quickly

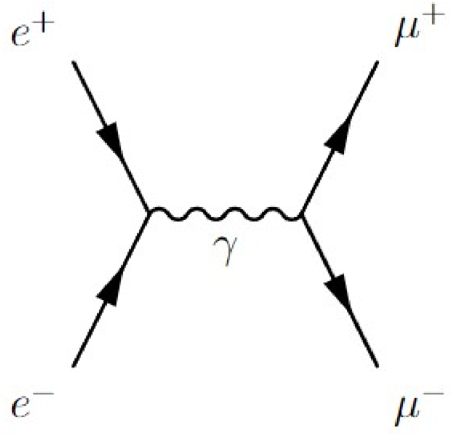

Figure 6.1: Feynman diagram of the $e^+e^- \longrightarrow \mu^+\mu^-$ reaction

decay into many different particles. To get the boost factors, it is beneficial to look at a simple decay reaction without much noise and that occurs as often as possible. For this reason, $e^+e^- \longrightarrow \Upsilon(4S) \longrightarrow \mu^+\mu^-$ is chosen as the ideal decay.

For further specification, the selection is limited to highly symmetrical reactions, where the resultant particles shoot off around 180 degrees back-to-back from each other. It is also detrimental to include the full energy spectrum, so the selection will cut the data down to only what is very near the 4S energy calculated above.

```
ma.reconstructDecay(’vpho:0 -> mu+:loose mu-:loose’,
’10.0 < M < 11.0 and nTracks == 2 and
sumThetaCMS > 3.13 and sumThetaCMS < 3.16’, path=main)
```

These cuts, as user defined in the basf2 code above, are fed many thousands of events from experiment 26, the most recent run of the Belle II detector.

The name “experiment” does not imply one separate from the larger Belle II experiment, as they are simply the numbered runs of the accelerator and detector.

Now that there are numbers to be used, it is useful to take a look at what it is saying about the kinematics of the situation. We can observe directly through the data in the figures 6.2, 6.3, 6.4, 6.5, and 6.6. The momentum of the center of mass system is a large peak at the zero mark. This says that in the center of mass frame, the system is stationary. The MC corroborates this conclusion from the data. On the other hand, the Lab frame tells a different story. It shows a sharp peak at just above $3\frac{\mathbf{GeV}}{\mathbf{c}}$, which almost entirely comes from the z-direction, or the axis that high energy beam propagates. Therefore, there is data proof of the asymmetry boost and confirmation that the z-axis is the dominating factor in the inequality of the reference frames.

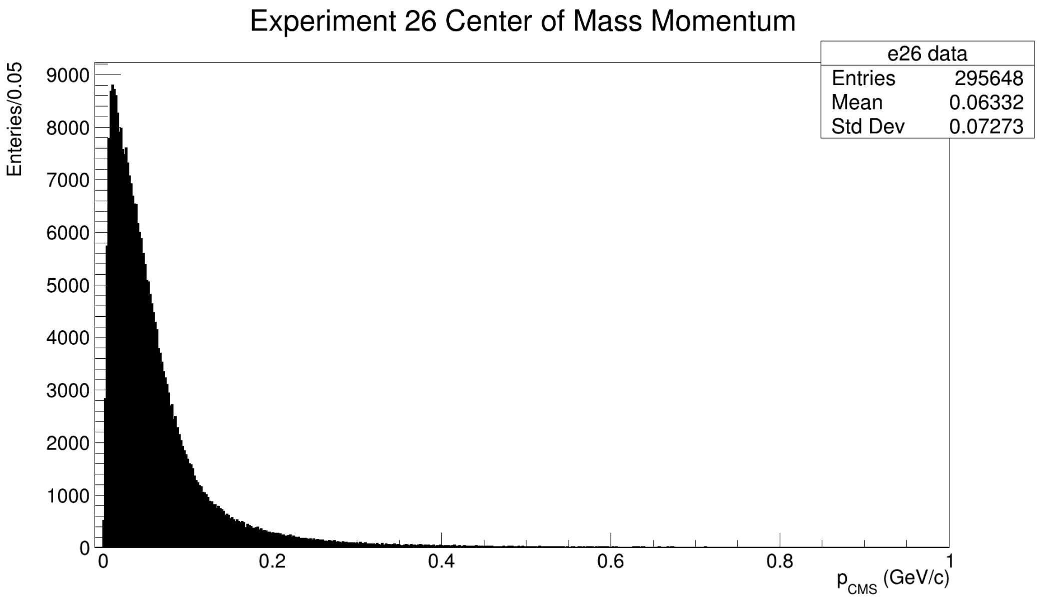

Figure 6.2: Histogram of the center of mass system momentum using experiment 26 data

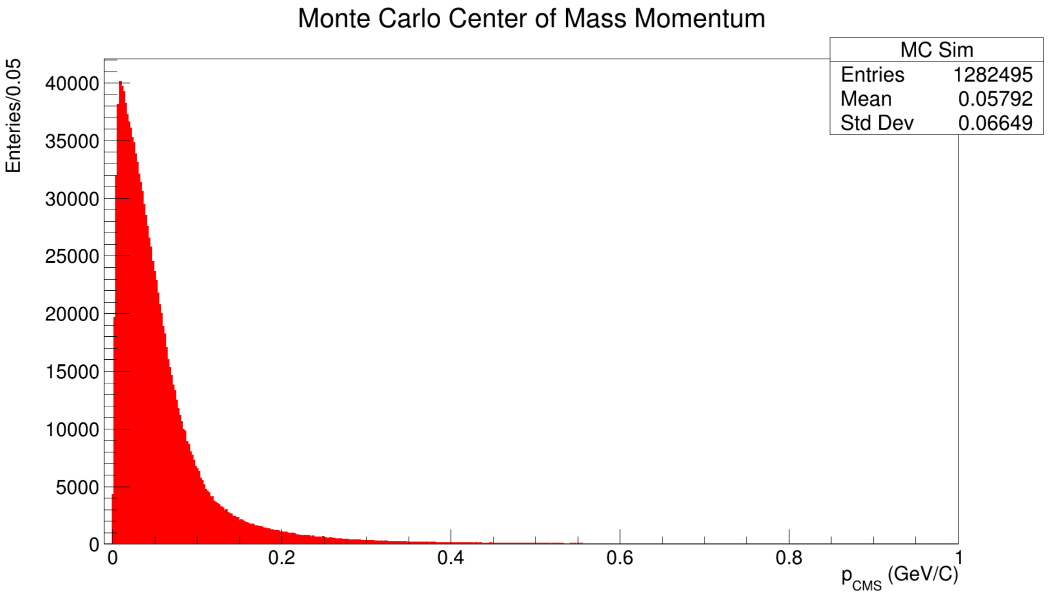

Figure 6.3: Histogram of the center of mass system momentum using MC simulation data

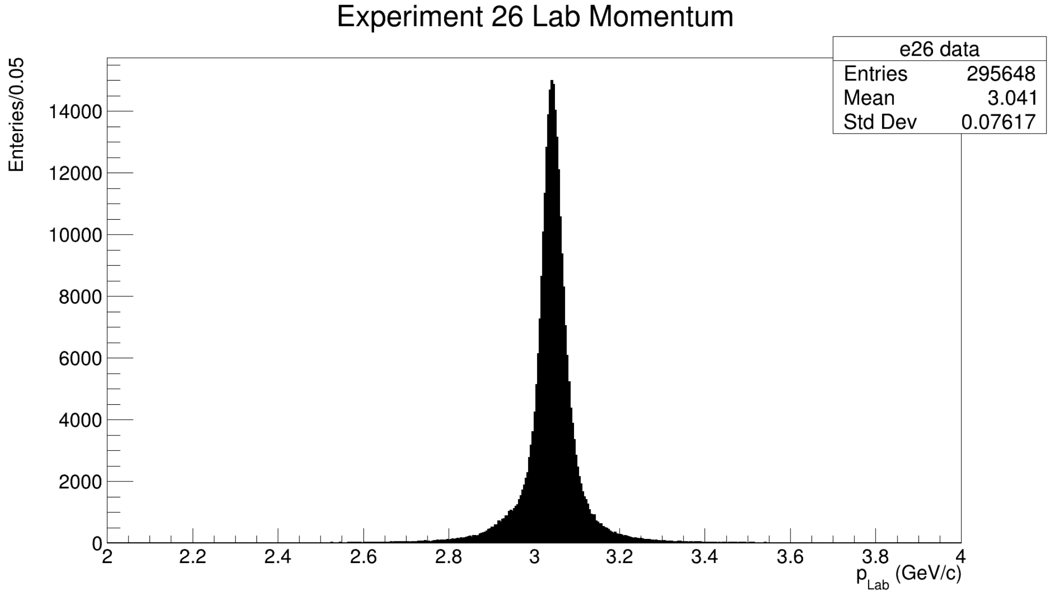

Figure 6.4: Histogram of the Lab frame momentum using experiment 26 data

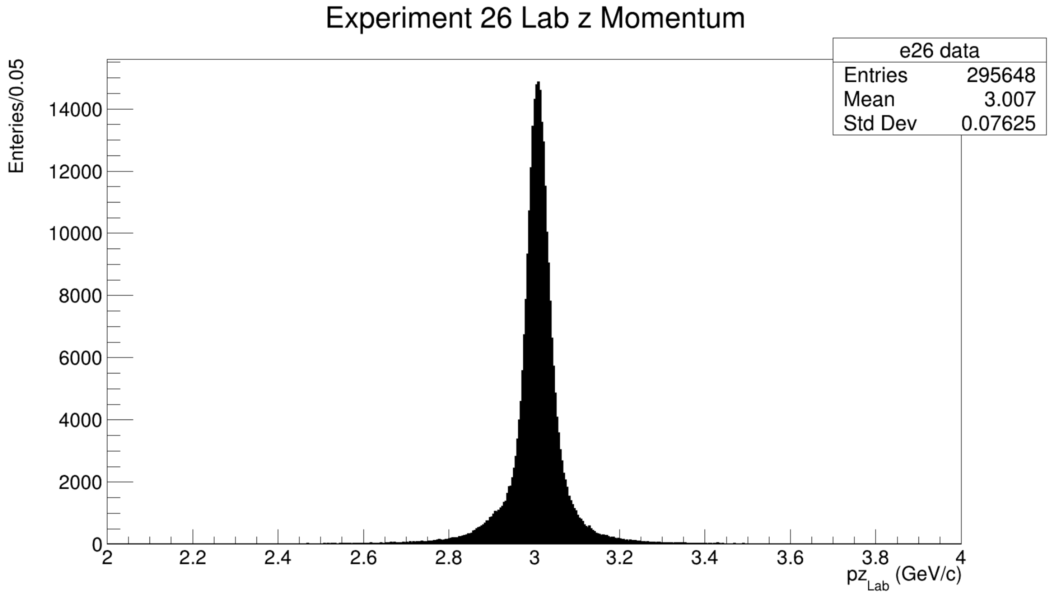

Figure 6.5: Histogram of the Lab frame momentum in the axis of propagation using experiment 26 data

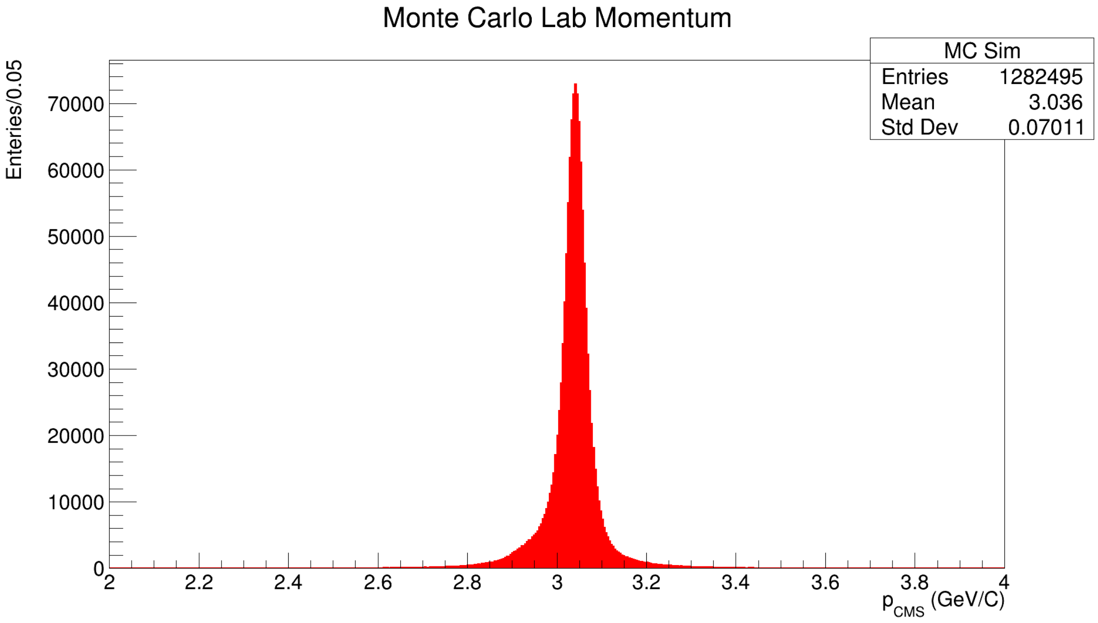

Figure 6.6: Histogram of the Lab frame momentum using MC simulation data

In regards to the choice of using the newest experiment over older ones, the choice is not solely for recency, but for the upgrade in quality of the detection. The luminosity of the detector, or the amount of data obtained, gets steadily better with time and constant improvements to the technology. The quality of old and new can be seen directly with a comparison of the mass distribution given the same or even stricter cuts compared to the new data.

Also consider the Monte Carlo, or MC, simulated distribution. The MC datasets are the theoretical ideals. The perfect version of the data without excess noise and exceptional behaviour held to the minimum. MC can be used to compare to the real datasets, such as in figures 6.7 and 6.8.

The fitting in these plots is performed by a Crystal Ball function. This is a piecewise function that fuses a Gaussian function with a low power. It takes 5 parameters, α and σ: which define the boundaries, N: a normalization constant, n: the number of events, and $\bar{x}$: the mean value of x, in this case the mean mass energy.

$$f(x;\alpha,n,\bar{x},\sigma) = N \cdot \begin{cases} \exp\left(-\frac{(x-\bar{x})^2}{2\sigma^2}\right), & \frac{x-\bar{x}}{\sigma} > -\alpha \\ A(n,\alpha)\cdot(B(n,\alpha) - \frac{x-\bar{x}}{\sigma})^{-n}, & \frac{x-\bar{x}}{\sigma} \leq -\alpha \end{cases} \tag{6.10}$$

Where $A = \frac{n}{\alpha}^n \cdot \exp(-\frac{\alpha^2}{2})$ and $B = \frac{n}{\alpha} - \alpha$. The fitting results in an improved mean value as seen in table 6.1, keeping in mind that the target is 10.5794 GeV. So the closer to 10.58 GeV, the better the result is.

It is evident that the experiment 26 dataset should be where we look to

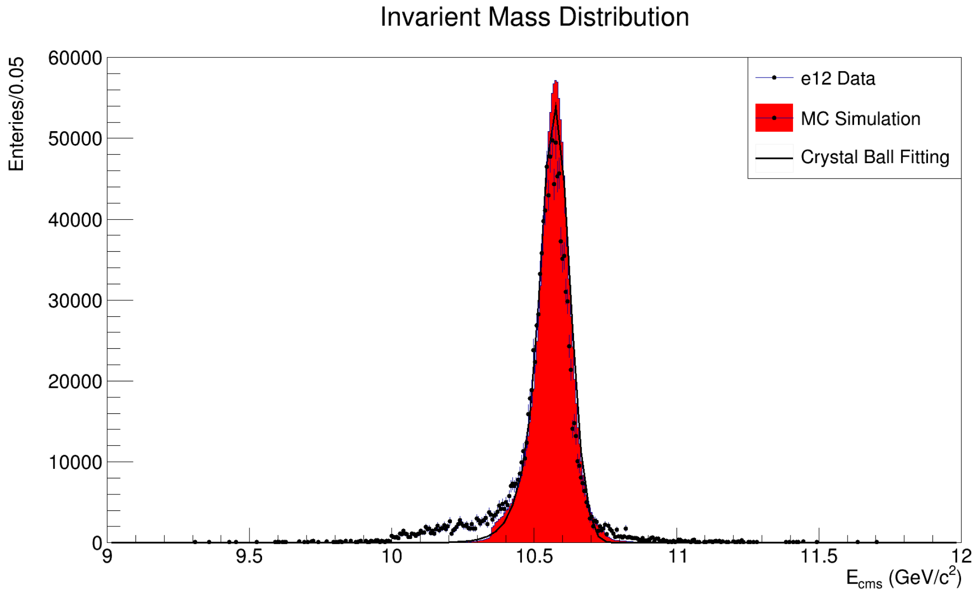

Figure 6.7: Comparison of the invariant mass energy of experiment 12 data and MC simulation

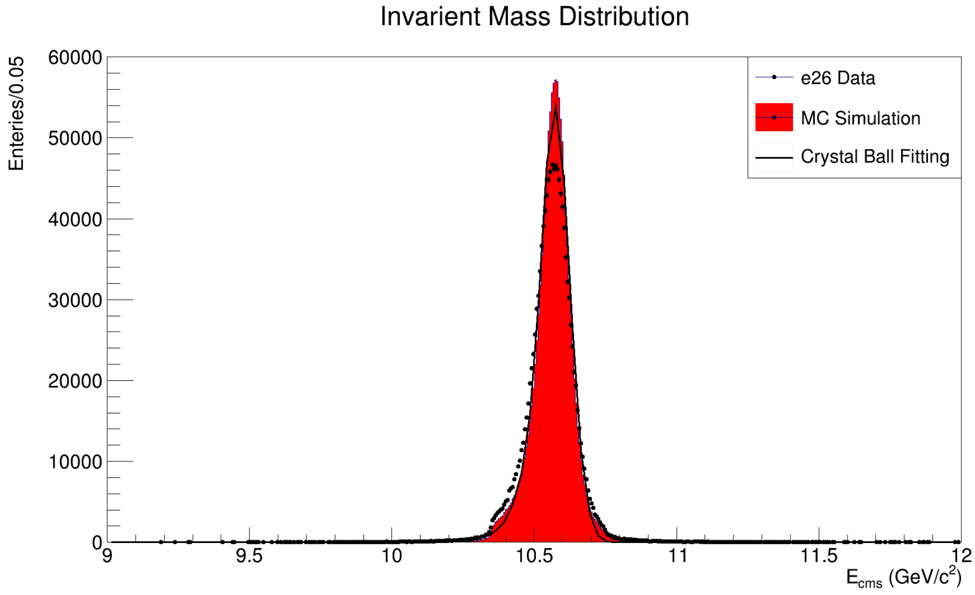

Figure 6.8: Comparison of the invariant mass energy of experiment 26 data and MC simulation

Table 6.1: The Mean Center of Mass energies of both tested experiments before and after fitting.

Experiment	Mean CMS Energy (GeV)	Fitted Mean Energy
12	10.5322	10.5644
26	10.5589	10.5726

get the most accurate boost values. There are a couple discrepancies between the experiment 12 data and the MC. First, there is the left tail, which tracks close near the mean mass energy, but can be seen trailing off at the end. The sudden dip in the MC comes from a tighter restriction in the cuts, but even before that, there is a clear divergence. Second thing is the right tail, which does not match very evenly around the center. It tends closer towards the middle rather than moving to hug the MC like the experiment 26 data does. Older experiments also have the consequence of being taken during a time when the luminosity of the Belle II detector was worse, resulting in far less data available. Using experiment 26 data results in a far greater resemblance to the situation depicted in the MC numbers, which means more reliable and accurate boost vectors, since the boost values of the MC are perfectly true by construction[17].

Chapter 7

Results

After matching the MC simulation and using formula 6.4. It is possible to calculate the boost vector. The following tables and plots are the results for the beta vector values for experiment 26 and the distributions of the x, y, z components of the vector.

The experimental values are in very close agreement with the simulation results, with the notable exception of the y component, in which the data undershoots compared to the MC. This could be due to the error introduced by the circular nature of the accelerator loop not being accounted for enough by the factor included in equation 6.8. Or by simple fact of being the smallest value by several orders of magnitude, the y component just ends up with more variance.

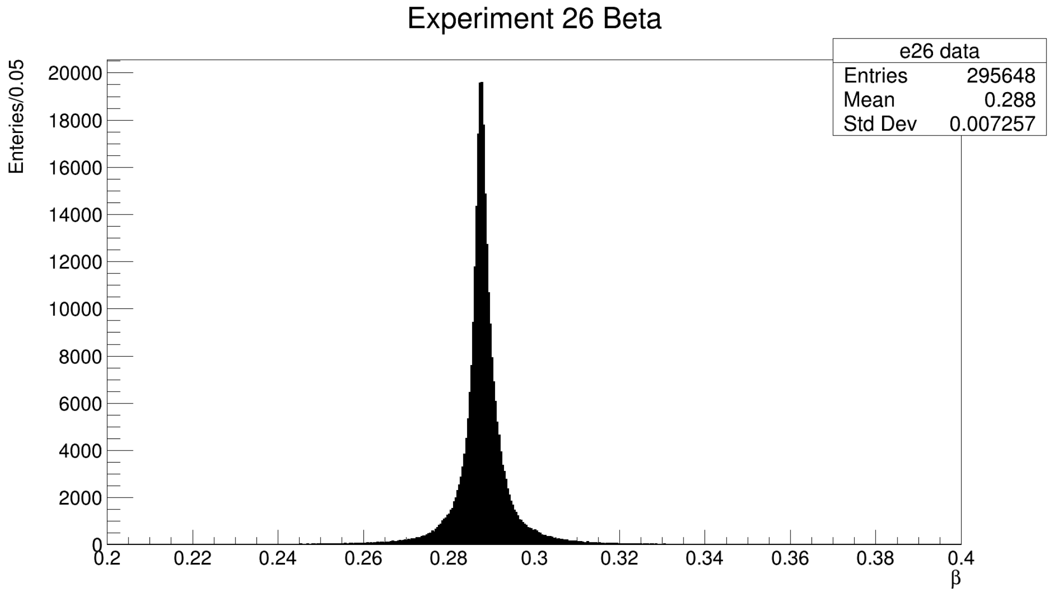

Figure 7.1: Histogram of the experiment 26 data β values

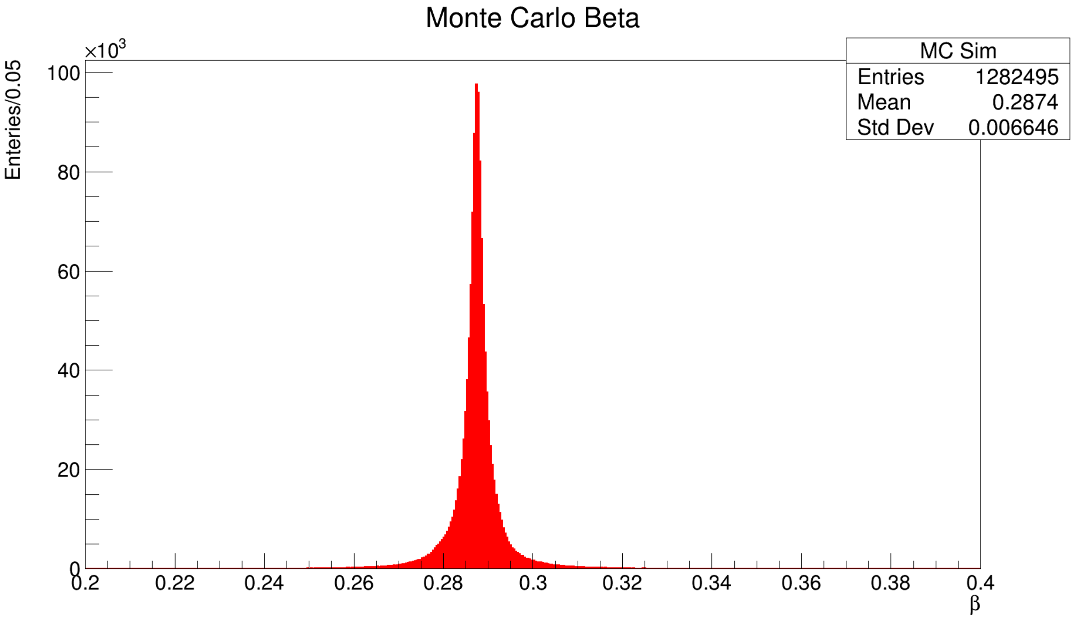

Figure 7.2: Histogram of the MC simulation β values

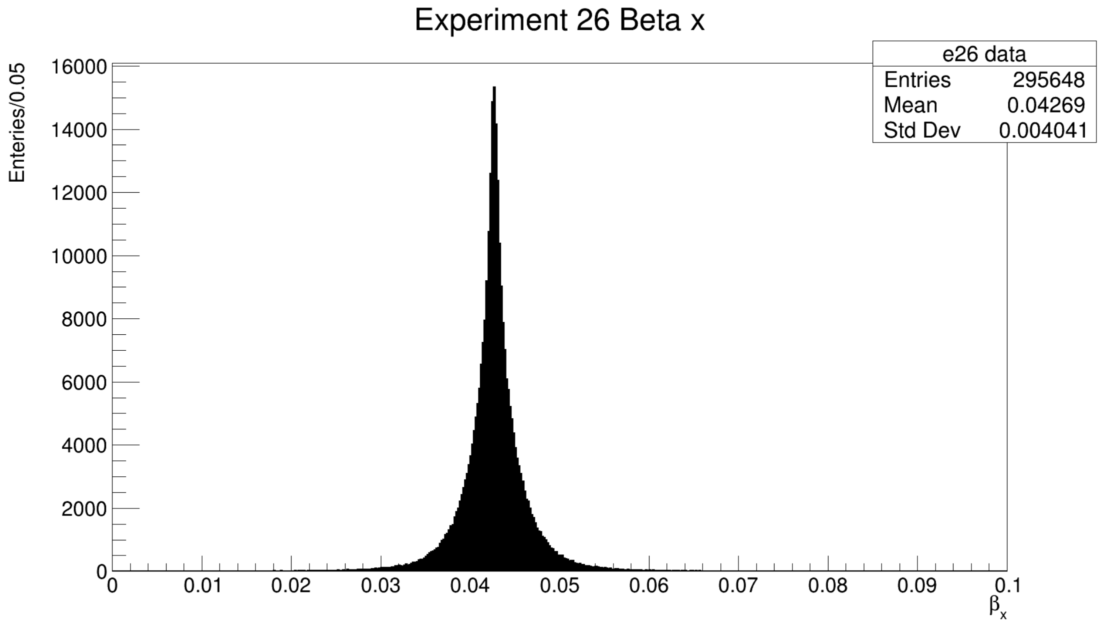

Figure 7.3: Histogram of the experiment 26 data β_x values

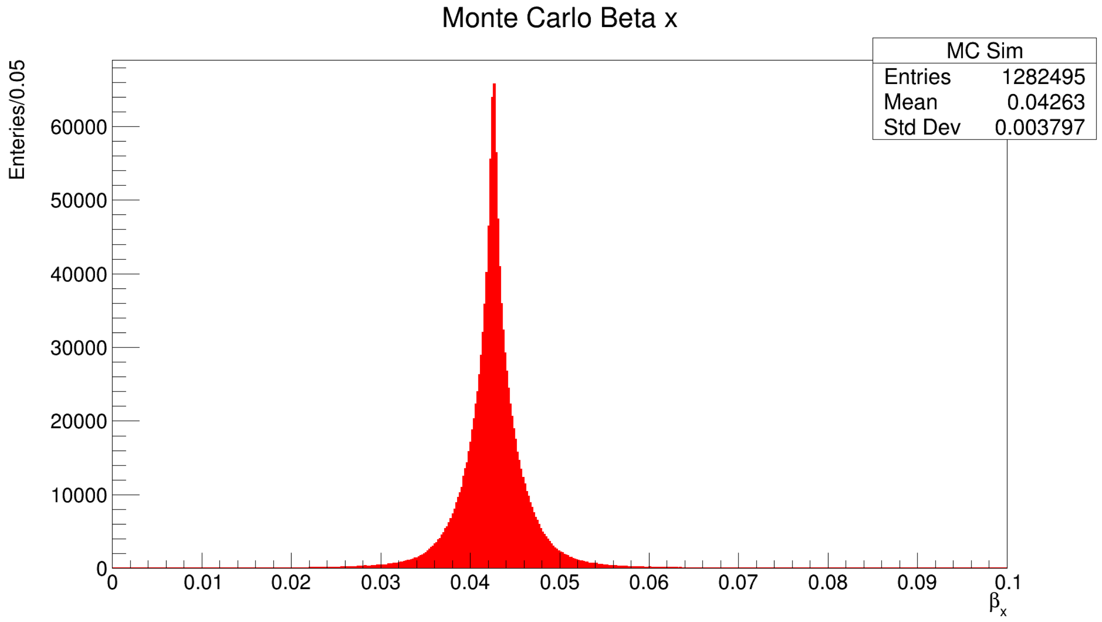

Figure 7.4: Histogram of the MC simulation β_x values

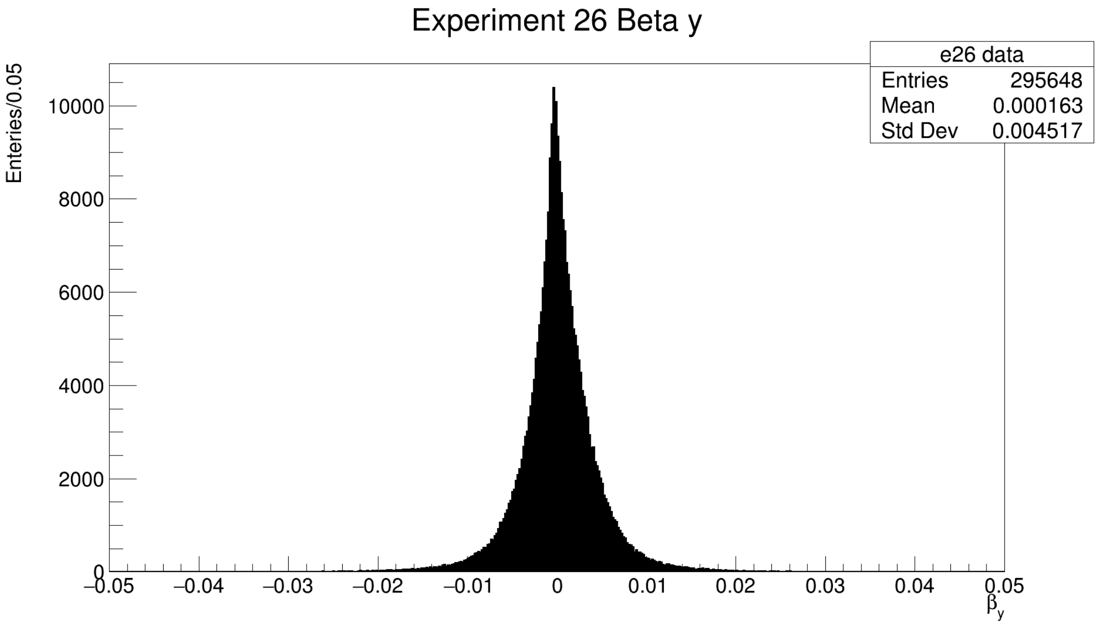

Figure 7.5: Histogram of the experiment 26 data β_y values

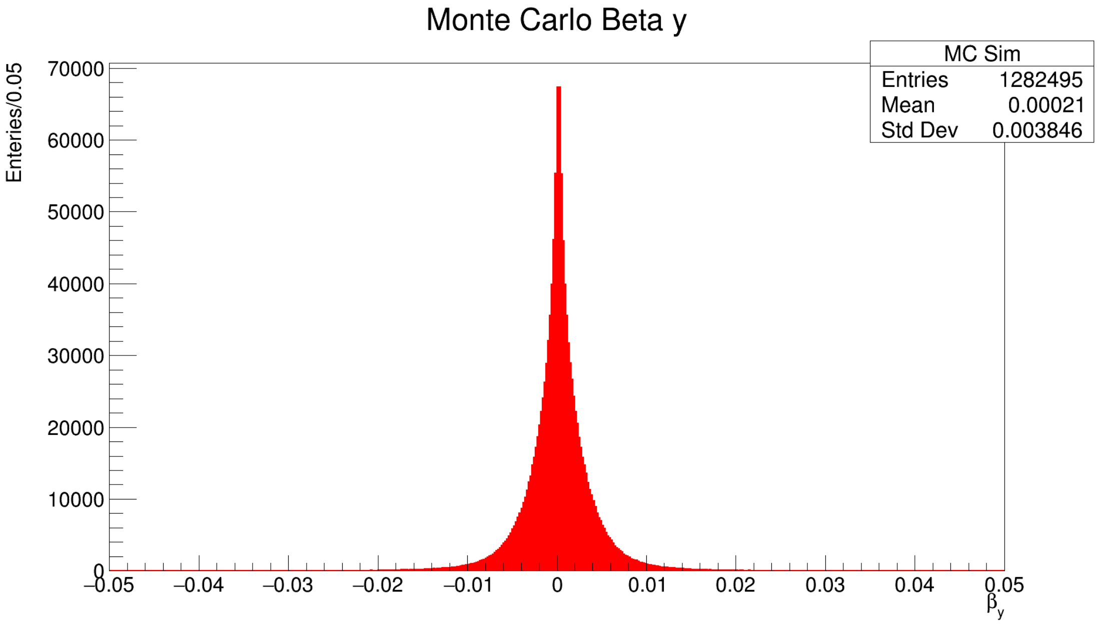

Figure 7.6: Histogram of the MC simulation β_y values

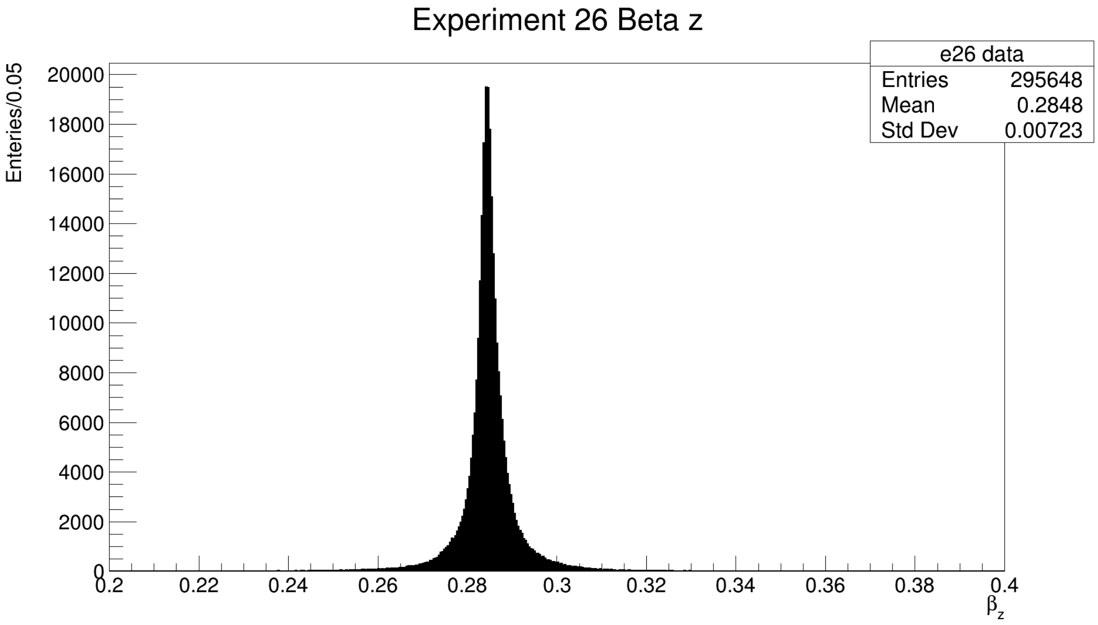

Figure 7.7: Histogram of the experiment 26 data β_z values

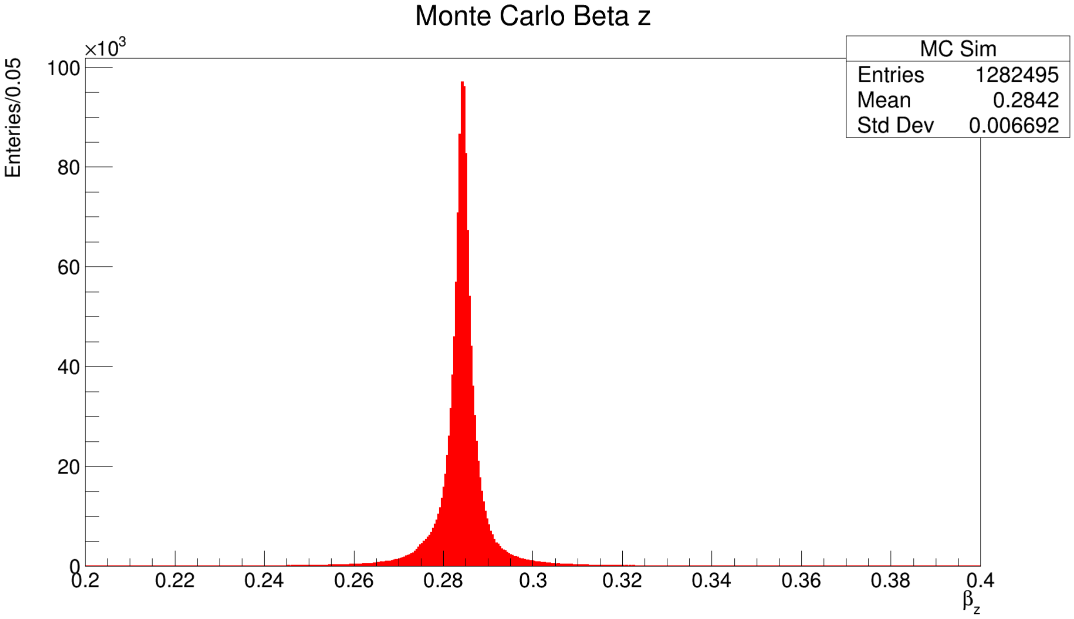

Figure 7.8: Histogram of the MC simulation β_z values

Table 7.1: A few values of experiment 26 β and its components

Event #	β	β_z	β_y	β_x
0	0.277906	0.274839	-0.006404	0.040672
1	0.287043	0.283896	0.000466	0.042383
2	0.288175	0.285044	0.000260	0.042364
3	0.288458	0.285219	0.000131	0.043109
4	0.287658	0.284425	-0.000830	0.042996

Table 7.2: A few values of MC simulation β and its components

Event #	β	β_z	β_y	β_x
0	0.275897	0.269165	-0.005136	0.060361
1	0.286837	0.283717	-0.000020	0.042192
2	0.287392	0.284012	-0.004032	0.043760
3	0.294054	0.291564	0.001107	0.038165
4	0.288407	0.284413	-0.002096	0.047783

Table 7.3: The average values of β and its components in both experiment 26 data and MC simulation data

	e26 data	MC simulation	% difference
β	0.288034	0.287416	0.215019
β_x	0.0426797	0.0426233	0.132322
β_y	0.000165277	0.000210452	21.4657
β_z	0.284783	0.284183	0.211132

Chapter 8

Ongoing Work

This formulation of the boost can be improved through the use of a quality factor function. By plotting distribution of the cosine of the polar angle, $\cos\theta$, as shown in figures 8.1 and 8.2, with fixed lesser components of beta, x and y, and varying the z component around the mean slightly, it is possible to plot out the χ^2 statistic. The χ^2 test has broad application in statistics, but Pearson's χ^2 test is its use in high energy physics. It takes the following form:

$$\chi^2(D, MC) = \sum_i \frac{(D_i - (MC)_i)^2}{\sigma^2_{D_i} + \sigma^2_{MC_i}} \tag{8.1}$$

Where i is the bins of the histogram, D and MC are the data and Monte Carlo histograms with the same binning, and σ_i is the bin resolution.

With several numbers collected from several β_z surrounding the mean found in the previous section, the χ^2 values are plotted against their associated β_z. The plot is given a quadratic fitting and the β_z value that lies at the minimum is the newly improved value for β_z. This is a process called

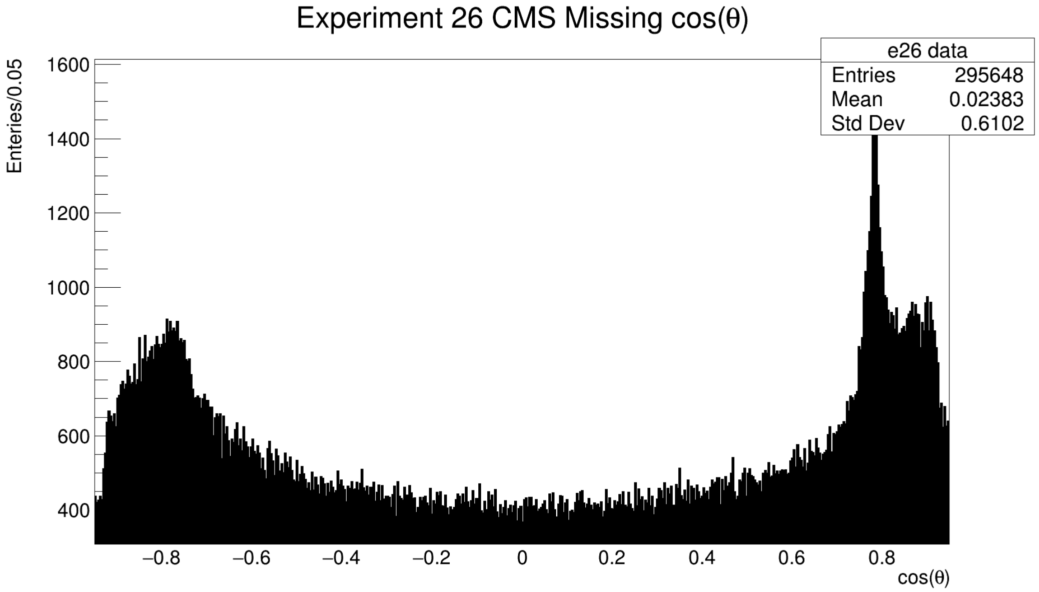

Figure 8.1: The missing cosine of the polar angle of the experiment 26 data.

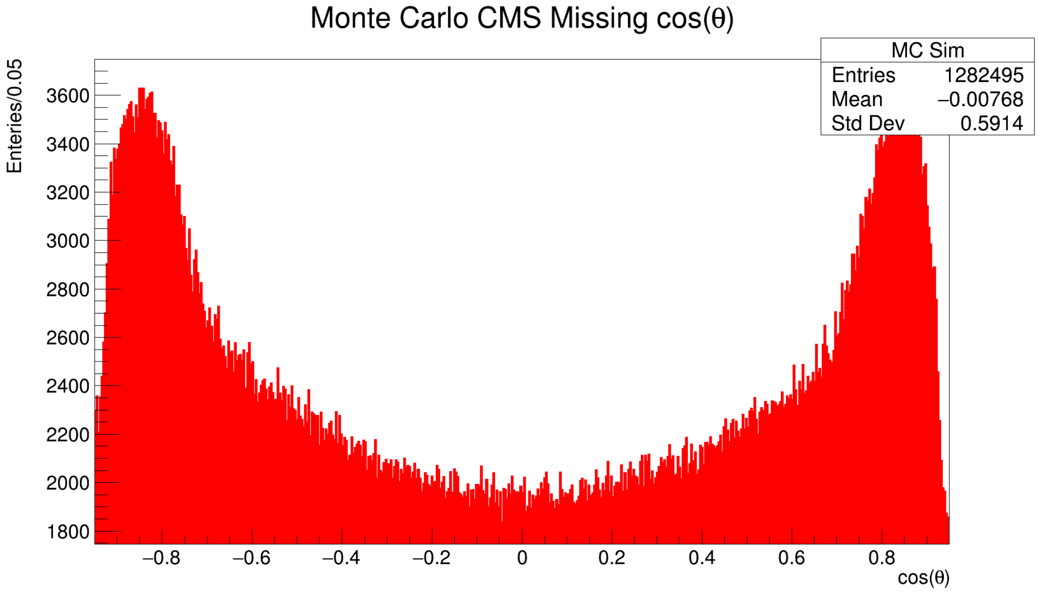

Figure 8.2: The missing cosine of the polar angle of the MC Simulation data.

minimization and it is the next step in my work. The outlier peak in the data (fig 8.1) should be resolved by controlling the β components during the χ^2 calculations.

Bibliography

[1] K.A. Olive et al. (Particle Data Group). "Υ(4S)". In: *Chinese Physics C* (2014). URL: https://pdg.lbl.gov/2014/listings/rpp2014-list-upsilon-4S.pdf.

[2] G. Aad et al. "Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC". In: *Physics Letters B* 716.1 (Sept. 2012), pp. 1–29. ISSN: 0370-2693. DOI: 10.1016/j.physletb.2012.08.020. URL: http://dx.doi.org/10.1016/j.physletb.2012.08.020.

[3] Kazunori Akai, Kazuro Furukawa, and Haruyo Koiso. "SuperKEKB collider". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 907 (2018), pp. 188–199. ISSN: 0168-9002.

[4] T. Abe et al. *Belle II Technical Design Report.* 2010. eprint: arXiv:1011.0352.

[5] Kullapha Chaiwongkhot et al. "Inner chamber of the Belle II Central Drift Chamber". In: *Nuclear Science Symposium and Medical Imaging Conference* (2013).

[6] Tadeas Bilka et al. "The track finding algorithm of the Belle II vertex detectors". In: *EPJ Web of Conferences* (2017).

[7] DESY. *PXD Basics*. Accessed: 24 November 2023. 2023. URL: https://confluence.desy.de/display/BI/PXD+Basics.

[8] Persis S. Drell. "Experimental Aspects Of B Physics". In: *SLAC Reports* (1991).

[9] Jason Engbrecht. *Introduction to Positron Research*. 2010. URL: https://web.archive.org/web/20100805002736/http://www.stolaf.edu/academics/positron/intro.htm.

[10] Fermilab/SLAC. *Symmetry Magazine*. URL: https://www.symmetrymagazine.org/.

[11] Moritz Gelb et al. "B2BII: Data Conversion from Belle to Belle II". In: *Computing and Software for Big Science* 2.1 (Nov. 2018). ISSN: 2510-2044. DOI: 10.1007/s41781-018-0016-x. URL: http://dx.doi.org/10.1007/s41781-018-0016-x.

[12] David Griffiths. *Introduction to Elementary Particles*. Wiley-VCH, 2008.

[13] Particle Data Group. URL: https://pdg.lbl.gov/.

[14] Francis Halzen and Alan D. Martin. *Quarks & Leptons: An Introductory Course in Modern Particle Physics*. John Wiley & Sons, 1984.

[15] KEK. *High Energy Accelerator Research Organization*. URL: https://kek.jp.

[16] KEK. *Machine Parameter*. Accessed: 24 november 2023. 2022. URL: https://www-superkekb.kek.jp/.

[17] Gregory King. "Search for Lepton Universality Violation Using Υ(3S) Decays". PhD thesis. University of Victoria, 2014.

[18] T. Kuhr et al. "The Belle II Core Software: Belle II Framework Software Group". In: *Computing and Software for Big Science* 3.1 (Nov. 2018). ISSN: 2510-2044. DOI: `10.1007/s41781-018-0017-9`. URL: `http://dx.doi.org/10.1007/s41781-018-0017-9`.

[19] Diogenes Laertius. *Lives of Eminent Philosophers*. Circa 200 AD.

[20] Abraham Pais. *Inward Bound*. 1986.

[21] Tsuyoshi Suwada. "Direct observation of positron capture process at the positron source of the superKEKB B-factory". In: *Scientific Reports* (2022).

Appendix A

Basf2 Steering File

```
    #!/usr/bin/env python3

from basf2 import *
import basf2 as b2
from ROOT import Belle2
from modularAnalysis import inputMdst
from modularAnalysis import inputMdstList
from modularAnalysis import fillParticleList
from modularAnalysis import applyEventCuts
from modularAnalysis import reconstructDecay
from modularAnalysis import variablesToNtuple
from modularAnalysis import variablesToHistogram
from modularAnalysis import buildEventKinematics
from variables import variables as vm
import modularAnalysis as ma
import vertex as vx
from stdCharged import stdMu
import os

main = b2.Path()

def list_files_in_directory(directory):
    file_list = []
    for root, dirs, files in os.walk(directory):
        for file in files:
            # Get the full path of the file
            file_path = os.path.join(root, file)
            # Append the file path to the list
            file_list.append(file_path)

    return file_list
ma.inputMdstList(environmentType=’default’, filelist=file_list, path=main)

ma.fillParticleList(’mu+:loose’,’’,path=main)
ma.fillParticleList(’mu-:loose’,’’,path=main)

#ma.reconstructDecay(’vpho:0 -> mu+:loose mu-:loose’,’10.0 < M < 11.0 and
nTracks == 2 and sumThetaCMS > 3.13 and sumThetaCMS < 3.16’, path=main)
ma.reconstructDecay(’vpho:0 -> mu+:loose mu-:loose’,’10.35 < M < 10.75 and
nTracks == 2 and sumThetaCMS > 3.13 and sumThetaCMS < 3.16’, path=main)
```

```
# ma.reconstructDecay(’vpho:0 -> mu+:loose mu-:loose’,’10.0 < M < 11.0 and
nTracks == 2 and sumThetaCMS > 3.139 and sumThetaCMS < 3.142’, path=main)
stdMu(’95eff’,path=main)
vx.treeFit(’vpho:0’, ipConstraint=True, path=main)

ma.buildEventKinematics(path=main)

vm.addAlias(’InvMass’, ’M’)
vm.addAlias(’InvMassError’, ’SigM’)
vm.addAlias(’d_0’, ’d0’)
vm.addAlias(’d_0Err’, ’d0Err’)
vm.addAlias(’z_0’, ’z0’)
vm.addAlias(’z_0Err’, ’z0Err’)
vm.addAlias(’sumInvM’, ’daughterInvM(0,1)’)
vm.addAlias(’deltaPhiLab’, ’daughterDiffOfPhi(0, 1)’) # lab
vm.addAlias(’deltaPhiCMS’,’daughterDiffOfPhiCMS(0, 1)’) #CMS
vm.addAlias(’deltaPhiLab1’, ’formula(daughter(0, useLabFrame(phi))
+ daughter(1, useLabFrame(phi)))’) # lab
vm.addAlias(’deltaPhiCMS1’,’formula(daughter(0, useCMSFrame(phi))
+ daughter(1, useCMSFrame(phi)))’) #CMS
vm.addAlias(’sumThetaLab’,’formula(daughter(0, useLabFrame(theta))
+daughter(1, useLabFrame(theta)))’)
vm.addAlias(’sumThetaCMS’,’formula(daughter(0, useCMSFrame(theta))
+daughter(1, useCMSFrame(theta)))’)
vm.addAlias(’combinationID’, ’extraInfo(decayModeID)’)
vm.addAlias(’deltaThetaLab’, ’formula(daughter(0, useLabFrame(theta))
- daughter(1, useLabFrame(theta)))’)
vm.addAlias(’deltaThetaCMS’, ’formula(daughter(0, useCMSFrame(theta))
- daughter(1, useCMSFrame(theta)))’)
vm.addAlias(’charge_0’, ’daughter(0, charge)’)
vm.addAlias(’charge_1’, ’daughter(1, charge)’)
vm.addAlias(’ELab’, ’useLabFrame(E)’)
vm.addAlias(’ECMS’, ’useCMSFrame(E)’)
vm.addAlias(’sumELab’,’formula(daughter(0, useLabFrame(E))
+ daughter(1, useLabFrame(E)))’)
vm.addAlias(’sumECMS’,’formula(daughter(0, useCMSFrame(E))
+ daughter(1, useCMSFrame(E)))’)
vm.addAlias(’clusterE_0’, ’daughter(0, clusterE)’)
vm.addAlias(’clusterE_1’, ’daughter(1, clusterE)’)
vm.addAlias(’sumclusterE’,’formula(daughter(0, clusterE)
+ daughter(1, clusterE))’) #lab
vm.addAlias(’sumclusterECMS’,’formula(daughter(0, useCMSFrame(clusterE))
+ daughter(1, useCMSFrame(clusterE)))’) #centre of mass
vm.addAlias(’pLab’, ’useLabFrame(p)’)
vm.addAlias(’pxLab’, ’useLabFrame(px)’)
vm.addAlias(’pyLab’, ’useLabFrame(py)’)
vm.addAlias(’pzLab’, ’useLabFrame(pz)’)
vm.addAlias(’pCMS’, ’useCMSFrame(p)’)
vm.addAlias(’pxCMS’, ’useCMSFrame(px)’)
vm.addAlias(’pyCMS’, ’useCMSFrame(py)’)
vm.addAlias('pzCMS', 'useCMSFrame(pz)')
vm.addAlias('clusterp_0', 'daughter(0, p)')
vm.addAlias('clusterp_1', 'daughter(1, p)')
vm.addAlias('clusterpCMS_0', 'daughter(0, useCMSFrame(p))')
vm.addAlias('clusterpCMS_1', 'daughter(1, useCMSFrame(p))')
vm.addAlias('sumpLab','formula(daughter(0,
useLabFrame(p)) + daughter(1, useLabFrame(p)))')
vm.addAlias('sumpCMS','formula(daughter(0,
useCMSFrame(p)) + daughter(1, useCMSFrame(p)))')
vm.addAlias('clusterTheta_0', 'daughter(0, clusterTheta)')
vm.addAlias('clusterTheta_1', 'daughter(1, clusterTheta)')
vm.addAlias('clusterPhi_0', 'daughter(0, clusterPhi)')
vm.addAlias('clusterPhi_1', 'daughter(1, clusterPhi)')
vm.addAlias('clusterECMS_0', 'daughter(0, useCMSFrame(clusterE))')
vm.addAlias('clusterECMS_1', 'daughter(1, useCMSFrame(clusterE))')
vm.addAlias('clusterThetaCMS_0', 'daughter(0, useCMSFrame(clusterTheta))')
vm.addAlias('clusterThetaCMS_1', 'daughter(1, useCMSFrame(clusterTheta))')
vm.addAlias('clusterPhiCMS_0', 'daughter(0, useCMSFrame(clusterPhi))')
vm.addAlias('clusterPhiCMS_1', 'daughter(1, useCMSFrame(clusterPhi))')
vm.addAlias('X_Angle', 'XAngle')
vm.addAlias('betax', 'formula(useLabFrame(px)/
[formula(daughter(0, useCMSFrame(E)) + daughter(1, useCMSFrame(E)))])')
vm.addAlias('betay', 'formula(useLabFrame(py)/
[formula(daughter(0, useCMSFrame(E)) + daughter(1, useCMSFrame(E)))])')
vm.addAlias('betaz', 'formula(useLabFrame(pz)/
[formula(daughter(0, useCMSFrame(E)) + daughter(1, useCMSFrame(E)))])')
vm.addAlias('beta', 'formula(useLabFrame(p)/
[formula(daughter(0, useCMSFrame(E)) + daughter(1, useCMSFrame(E)))])')
vm.addAlias('missing_theta_lab',
'useLabFrame(missingMomentumOfEvent_theta)')
vm.addAlias('missing_theta_CMS', 'useCMSFrame(missingMomentumOfEventCMS_theta)')
vm.addAlias('cosPolar', 'cosTheta')
vm.addAlias('cosPolarErr', 'cosThetaErr')
vm.addAlias('cosPolarCMS_miss',
'formula(missingMomentumOfEventCMS_Pz / missingMomentumOfEventCMS)')
vm.addAlias('cosPolarLab_miss',
'formula(missingMomentumOfEvent_Pz / missingMomentumOfEvent)')
vm.addAlias('Chi2', 'chi2')
vm.addAlias('MomDevChi2', 'momDevChi2')
vm.addAlias('cosPolarCMS', 'formula(useCMSFrame(pz) / useCMSFrame(p))')
vm.addAlias('cosPolarLab', 'formula(useLabFrame(pz) / useLabFrame(p))')
vm.addAlias('ChiProb', 'chiProb')
# variables

var_mu = ['M',
        'SigM',
        'cosPolar',
        'cosPolarErr',
        'cosPolarCMS_miss',
        'cosPolarLab_miss',
        ’cosPolarCMS’,
        ’cosPolarLab’,
        ’Chi2’,
        ’MomDevChi2’,
        ’ChiProb’,
        ’d_0’,
        ’d_0Err’,
        ’z_0’,
        ’z_0Err’,
        ’sumInvM’,
        ’deltaPhiLab’,
        ’deltaPhiCMS’,
        ’deltaPhiLab1’,
        ’deltaPhiCMS1’,
        ’sumThetaLab’,
        ’sumThetaCMS’,
        ’combinationID’,
        ’deltaThetaLab’,
        ’deltaThetaCMS’,
        ’charge_0’,
        ’charge_1’,
        ’ELab’,
        ’ECMS’,
        ’clusterE_0’,
        ’clusterE_1’,
        ’sumclusterE’,
        ’sumclusterECMS’,
        ’sumELab’,
        ’sumECMS’,
        ’pLab’,
        ’pxLab’,
        ’pyLab’,
        ’pzLab’,
        ’pCMS’,
        ’pxCMS’,
        ’pyCMS’,
        ’pzCMS’,
        ’clusterp_0’,
        ’clusterp_1’,
        ’clusterpCMS_0’,
        ’clusterpCMS_1’,
        ’sumpLab’,
        ’sumpCMS’,
        ’clusterTheta_0’,
        ’clusterTheta_1’,
        ’clusterPhi_0’,
        ’clusterPhi_1’,
        ’clusterECMS_0’,
        ’clusterECMS_1’,
        ’clusterPhiCMS_0’,
        'clusterPhiCMS_1',
        'clusterThetaCMS_0',
        'clusterThetaCMS_1',
        'X_Angle',
        'betax',
        'betay',
        'betaz',
        'beta'
        ]

ma.variablesToNtuple('vpho:0', var_mu, treename='vpho',
filename='vpho_data.root', path=main)

b2.process(main)

print(b2.statistics)
```

Appendix B

Comparative Mass Energy ROOT Macro

```
    #include "TMath.h"
    //Crystal ball function for signal, parameters are
    //0:alpha,1:n,2:mean,3:sigma,4:normalization
{Double_t CrystalBall(Double_t *x,Double_t *par) {

  Double_t t = (x[0]-par[2])/par[3];
    if (par[0] < 0) t = -t;

      Double_t absAlpha = fabs(par[0]);

        if (t >= -absAlpha) {
            return par[4]*exp(-0.5*t*t);
              }
```

```
                else {
                    Double_t a =  TMath::Power(par[1]/absAlpha,par[1])*
                    exp(-0.5*absAlpha*absAlpha);
                        Double_t b= par[1]/absAlpha - absAlpha;

                            return par[4]*(a/TMath::Power(b - t, par[1]));
                              }
   }
 gROOT->Reset();

TCanvas *c1 = new TCanvas("c1","histograms");

TFile *f1 = TFile::Open("MC.root");
TTree *treeMC;
f1->GetObject("vpho", treeMC);
Double_t M;
treeMC->SetBranchAddress("M", &M);
const double_t bins = 500;
const double_t min = 9;
const double_t max = 12;
TH1D *h1f = new TH1D("MC Sim", "Invarient Mass Distribution", bins, min, max);

for (Long64_t entry = 0; entry < treeMC->GetEntries(); entry++) {
    treeMC->GetEntry(entry);
    h1f->Fill(M);
```

```
}
h1f->SetMarkerStyle(8);
h1f->SetFillColor(kRed);

TFile *f2 = TFile::Open("vpho_data.root");
TTree *treeD;
f2->GetObject("vpho", treeD);
treeD->SetBranchAddress("M", &M);
TH1D *h1s = new TH1D("e26 data", "M Distribution", bins, min, max);

for (Long64_t entry = 0; entry < treeD->GetEntries(); entry++) {
    treeD->GetEntry(entry);
    h1s->Fill(M);
}

h1s->SetMarkerStyle(20);
h1s->SetMarkerColor(kBlack);

TF1 *crystal = new TF1("crystal", CrystalBall, 8, 12, 5);
crystal->SetParameters(1.11418,1.15420462,10.56,0.0515167,h1f->GetEntries());
crystal->SetParNames("#alpha","n","Mean","#sigma","N");
crystal->SetLineColor(kBlack);
crystal->SetLineWidth(3);
h1f->Fit("crystal","rv");
```

```
h1f->GetXaxis()->SetTitle ("E_{cms} (GeV/c^{2})");
h1f->GetYaxis()->SetTitle ("Enteries/0.05");

Double_t scaleFactor = h1f->GetEntries() / h1s->GetEntries();
h1s->Scale(scaleFactor);

TLegend* legend = new TLegend(0.7, 0.7, 0.9, 0.9);
legend->AddEntry(h1s, "e26 Data");
legend->AddEntry(h1f, "MC Simulation");

h1f->SetStats(false);
h1f->Draw();
h1s->Draw("SAME");
legend->Draw();

c1->Update();
c1->SaveAs("hm_mc.png");
}
```