NP10: Search for mesons containing b-quarks report

James Bate
orie3568
Oriel College

21/01/19

1 Abstract

In this project, data from Collider Detector Facility (CDF) at Fermilab is used to find the mass of the J/ψ, B^+ and B_s^0 mesons. The mass of the J/ψ meson is found to be $3.09 \pm 0.08\ GeV/c^2$, which is consistent with the known mass of $3.096916 \pm 0.011\ GeV/c^2$. The mass of the B^+ meson if found to be $5.28 \pm 0.05\ GeV/c^2$ which is consistent with the known mass of $5.27929 \pm 0.15\ GeV/c^2$. The mass of the B_s^0 is found to be $5.25 \pm 0.30\ GeV/c^2$ which is consistent with the known mass of $5.36679 \pm 0.23\ GeV/c^2$. The lifetime of the B^+ was also calculated to be $(6.95 \pm 4.0)\ 10^{-14} s$. The known value of the mean lifetime is 1.64×10^{-12}s. We therefore predict a branching ratio of approximately $4.2 \times 10^{-2} \pm 2.4 \times 10^{-2}$ for the decay mode examined in this paper.

2 Introduction

The aim of this project was to calculate the mass and lifetime of three mesons; the J/ψ, B^+ and B_s^0 mesons. These mesons are produced by high energy proton-antiproton collisions. The data used in this project is from the CDF at Fermilab. All three of these mesons decay on time scales that are far too short to allow direct measurement of these properties, since they decay before reaching any of the detectors. Therefore, the properties have to be inferred from measurements of the more stable muons and kaons that they decay into.

Table 1 shows the lifetime of each of the mesons relevant to this project. It is noticeable how significantly the lifetimes of the particles vary. This is dictated by which interaction causes the decay. The three forces that can cause nuclear decay are the strong, weak, and electromagnetic force. To each of these forces there is associated an exchange particle which propagates it. Decays via the weak interaction occur on longer time scales, due to the larger mass of the propagator. The mesons on the top half of table 1 with longer lifetimes of $\tau \approx 10^{-8}\ s \rightarrow 10^{-12} s$ decay via the weak force. The mesons on the bottom half of table 1 decay via the strong force which occurs at much shorter time scales. The decay modes that we will be analysing in project this are the following.

$$B^+ \rightarrow J/\psi + K^+ \tag{1}$$

$$J/\psi \rightarrow \mu^+ + \mu^- \tag{2}$$

$$B_s^0 \rightarrow J/\psi + \phi \tag{3}$$

$$\phi \rightarrow K^+ + K^- \tag{4}$$

The detector at Fermilab can detect muons, pions and kaons, but all the other mesons are not directly detectable.

To analyse particle decays, we use the 4-momentum defined in the following way.

$$P^\mu = \left(\frac{E}{c}, \overline{p}\right) \tag{5}$$

The n-body decay process can be written in terms of 4-momenta as follows.

$$P^\mu_{parent} = \sum_i P^\mu_i \tag{6}$$

Meson	Quark content	Lifetime
B^+	$u\bar{b}$	1.64×10^{-12}s
B_s^0	$s\bar{b}$	1.51×10^{-12}s
K^+	$u\bar{s}$	1.24×10^{-8}s
K^-	$s\bar{u}$	””
J/ψ	$c\bar{c}$	7.2×10^{-21}s
ϕ	$s\bar{s}$	1.55×10^{-22}s

Table 1: Table of lifetime and quark content of relevant mesons, separated into weak (top half) and strong (bottom half)

where P_i^μ are the products of the decay. This is a statement of the conservation of energy and momentum. By taking the invariant, we can find the mass of a parent particle given the rest masses and momenta of the particles it decays into.

$$M^2c^2 = \sum_{i=1}^{n} m_i^2c^2 + 2 \sum_{i,j;i<j}^{n} \left\{ \frac{\sqrt{p_i^2c^2 + m_i^2c^4}\sqrt{p_j^2c^2 + m_j^2c^4}}{c^2} - \vec{p_i}.\vec{p_j} \right\} \tag{7}$$

Where M is the mass of the parent particle, and m_i and $\vec{p_i}$ are mass and momentum respectively of the particles the parent decays into. This is a very useful formula as the parent mass is expressed purely in terms of variables which are measurable.

The relativistic effect of time dilation is also important in our analysis of the lifetime of a particle. A particle moving at speed $\beta = v/c$ travels a distance d in time t where

$$d = \beta c \gamma t \tag{8}$$

where $\gamma = \frac{1}{\sqrt{1-\beta^2}}$ is the gamma factor.

The probability of decay for a particle is related to its mean lifetime by

$$P_{decay} = e^{-t/\tau} \tag{9}$$

where τ is the mean lifetime averaged over all possible decay modes. The lifetime for a specific decay mode can then be found using the branching ratio

$$\tau_i = \frac{\Gamma}{\Gamma_i}\tau \tag{10}$$

where Γ_i is known as the partial width for a specific decay mode.

3 Method

The collider consists of several co-axial cylindrical detectors of different radii. The outermost layer is the muon chamber. The only particles which make it to this chamber are muons since they do not interact with the strong force. It is therefore very easy to determine whether data from a given track came from a muon by finding whether it was detected by the outer muon chamber, or one of the inner detectors.

The overall structure of the program used to collect the particle data worked in the following way. The outer loop of the program ran over collision events. Within a given collision, data on the properties of all the detected particles was read. The muon tracks in this data had already been separated due to the fact that muons are easily distinguished. The mesons that are being detected in this project have very short lifetimes, much too short to be directly detected. The method used to find them therefore is to detect the more stable particles they decay into. The difficulty then is finding which of these stable products came from the desired parent, and which came from other decay modes. This is done using an inner loop over all combinations of the detected stable products which are expected from a particular decay mode. These are then treated as candidate products. The relativistic decay formula, equation 7, is then used to calculate what the mass of the parent would be if these products came from the decay mode, and this is then plotted on a histogram. This is called a 'candidate mass' since the particles used in equation 7 might not have actually come from the desired parent, in which case it would contribute to the background noise of the mass histogram. This histogram should then have a gaussian peak at the value of the mass of the unstable parent, and roughly uniform background from the incorrect combinations. The structure of the program for detecting the J/ψ meson is displayed in figure 5 in appendix A as an example. The programs for detecting the other two mesons have a very similar structure, so comparisons can be made from the J/ψ meson flow chart.

When detecting less common mesons like the B mesons, the mass peak can often be buried in the data. The peak can be made more clear by disregarding candidate product tracks whose momenta could not have come from the desired parent on kinematic grounds. This is called making 'cuts' to the data. 'cuts' can also be applied to candidate masses for more complicated decay chains, as will be described later.

3.1 The J/ψ meson

The J/Ψ meson decays into two muons as described in equation 2. The combinations that were looped over in the program were therefore (μ^+, μ^-) muons. The charge of the muon can be easily determined from its track by looking at the curvature. Since the muons are travelling through a constant external B field, muons of opposite charge have tracks of opposite curvature. In the geometry of this detector, positive curvature corresponds to a positive charge. The structure of the program is displayed in the form of a flow chart in figure 5. This is a very common decay mode, and the peak came out very clearly without any cuts.

3.2 The B^+ meson

In contrast, the B^+ meson undergoes the rarer decay mode given in equation 1. It was therefore searched for by looping over J/Ψ candidate and kaon candidate pairs. The tracks are expected to be a mix of kaons, pions and muons, but in this program we treated them all as a candidate kaon, since kaons and pions are not as easily distinguishable as muons. This therefore creates much more noise in the signal than for the J/ψ peak. In order to reduce background noise and get a better defined peak at the B^+ mass, cuts were used on the candidate K^+ and J/Ψ particles. A cut was made on the J/Ψ mass to be between 3 and 3.2 GeV/c^2. A cut was made on the transverse momentum of the muons to be $> 0.8GeV/c$. A cut was made on the transverse momentum of the kaons to be $> 2GeV/c$.

3.3 The B_s^0 meson

The more rare B_s^0 meson was searched for by using the longer decay chain (equation 2,3,4). This was done by considering all pairs $(\mu^+, \mu^-), (K^+, K^-)$ as candidate products of J/ψ and ϕ decay respectively. The candidate B_s^0 meson mass was then found from the four particle decay formula (equation 7). This particle is less common than the B^+ meson and so the signal to noise ratio is much higher. Extra data was therefore imported to boost the signal. A cut was made on the transverse momentum of the kaon candidates to be $> 2GeV/c$. A cut was made on the candidate ϕ meson mass to be in the range $1.01 < m_\phi < 1.04$ GeV/c^2.

3.4 Fitting the mass histograms

After finding the mass histograms of the mesons, a fitting function was used to analyse it. The fitting function used was a gaussian function with uniform background noise, which has the following form:

$$f(x) = p_0 exp(\frac{-(x-p_1)^2}{2p_2^2}) + p_3 \tag{11}$$

This fitting function has four parameters. p_3 controls the level of the uniform background noise. p_0 controls the height of the gaussian peak above the noise. p_1 controls the mean, and p_2 controls the standard deviation. The stability of each of these parameters in the fitting function was tested by trying a range and observing whether this affected the final fit. After fitting the function, the mean mass of the particle was found from the value of p_1. There are two errors present in this value, one from the FWHM of the gaussian, and one from the error on the fit. In all cases in this project, the fitting error was negligible in comparison to the FWHM, and so the latter was taken as the error on the mass.

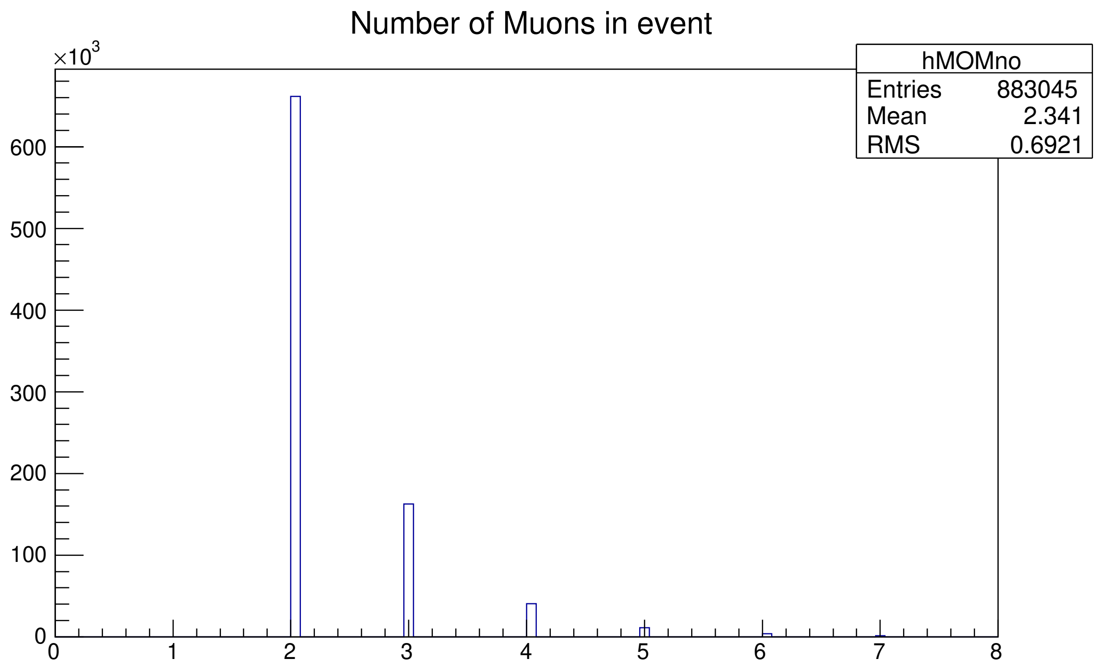


Figure 1: Histogram plot of number of muons detected per event. y axis is number of entries.

3.5 Lifetime

Another property that could be calculated from the data was the particle lifetime. The lifetime of the B^+ meson was found by finding the spatial location of the vertex at which the muon pair and kaon tracks converge, and calculating the length of this point from the collision event, known as the primary vertex, to get the path length of the B^+ meson. In practice, the tracks do not converge exactly to a vertex due to a limit on measurement precision, and so an algorithm was used which used a least squares method to find the vertex of best fit. In this calculation it is assumed that the B^+ meson travels in a straight line, which is reasonable over such a small length scale.

$$\tau = \frac{md}{p} \tag{12}$$

Where τ is the proper time of the particle, m is its mass and d is the distance travelled before decay. The probability of decay is expected to be exponential, as described in equation 9. When τ is plotted on a logarithmic scale, it will be a straight line. The gradient of this is proportional to the inverse of the lifetime.

4 Results and Analysis

Throughout this analysis, the z axis is defined as being along the line of the detector. It is useful to have an idea of the average number of muons that are detected in each event. The plot of number of muons per event is shown in figure 1. The mean number of muons in each event is 2.34, and so we expect events with more than one meson to be very rare. It is expected that no event records only an single muon, because by the conservation of charge most decay modes will product muon anti-muon pairs. As a reference, the detected muon momenta are shown in figures 6a,6b in appendix A.

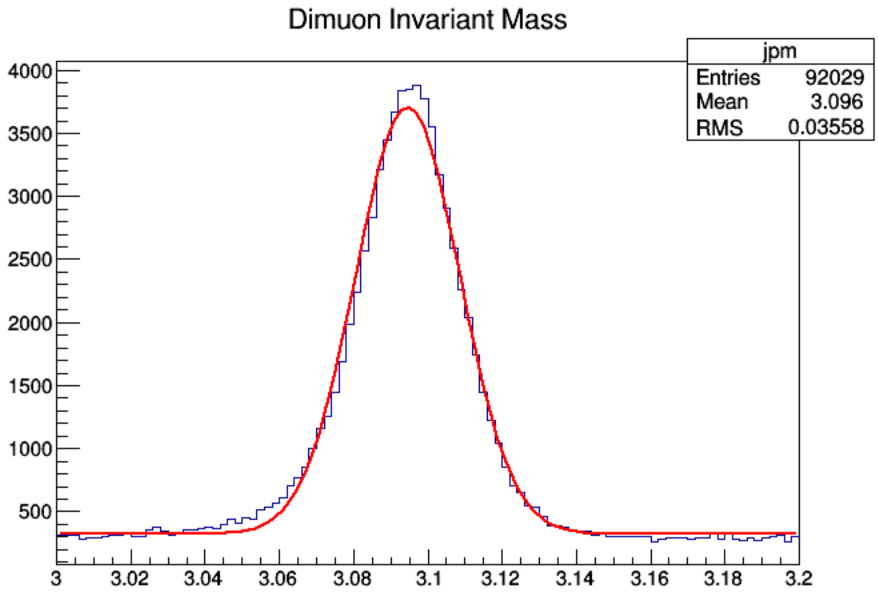


Figure 2: Histogram plot of invariant mass (equation 7) for the J/ψ meson. x axis is the mass given in GeV/c^2. y axis is number of entries.

Parameters	Value	Error
p0	3.38×10^3	1.97×10^1
p1	3.09	6.99×10^{-5}
p2	1.40×10^{-2}	6.84×10^{-5}
p3	3.20×10^2	2.34

Table 2: Values of the parameters for the fitting function (equation 11) of the J/Ψ meson histogram.

4.1 The J/ψ meson

The histogram of the mass of the J/ψ meson is shown in figure 2. The histogram has a gaussian shape. The peak is very clear, the uniform background noise is about 6.3% of the signal peak, and so we can be very confident that a peak does exist here. This peak was fitted with a gaussian fitting function as described in equation 11, and the the fit parameters are outlined in table 2. There are two independent sources of error in the value of the mass. One comes from the FWHM of the gaussian curve, which is 0.016 GeV/c^2. The other is from the fitting error the gaussian mean, which is $6.988\,95 \times 10^{-5}$ GeV/c^2. The fitting error is much smaller than that of the FWHM, and so we will ignore it. The mass of the J/ψ meson is found to be 3.09 ± 0.08 GeV/c^2. This is consistent with the known mass of 3.096916 ± 0.011 GeV/c^2. The fitting function was very stable, and no different fit was found after reasonable perturbations in all of the parameters. The transverse and z momenta can be seen in figure 7 in appendix A.

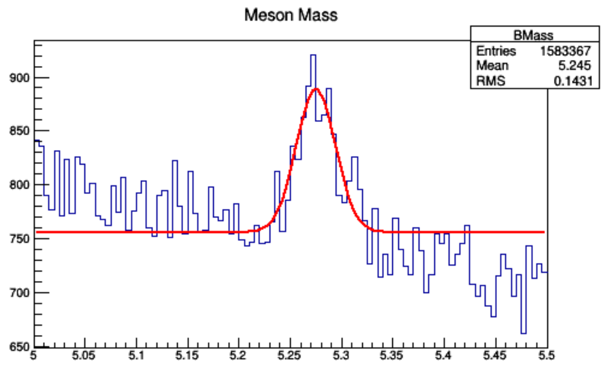
4.2 The B^+ meson

The histogram of the mass of the B^+ meson is shown in figure 3a. It locally follows an approximate gaussian shape, although there is a lot more noise in the signal than for the J/ψ meson. The peak was fit with a gaussian fitting functions as specified in equation 11. The parameters of the fit are shown in table 3. The contribution to the error on the mass is much larger from the FWHM of the histogram than from the error on the gaussian fit, and so the FHWM is used as the error. The mass is found from the mean of the gaussian fit to be 5.28 ± 0.05 GeV/c^2 which is consistent with the known mass of 5.27929 ± 0.1 GeV/c^2. This fit was made for a range of starting points to test its stability, and it was found that the fit was very stable.

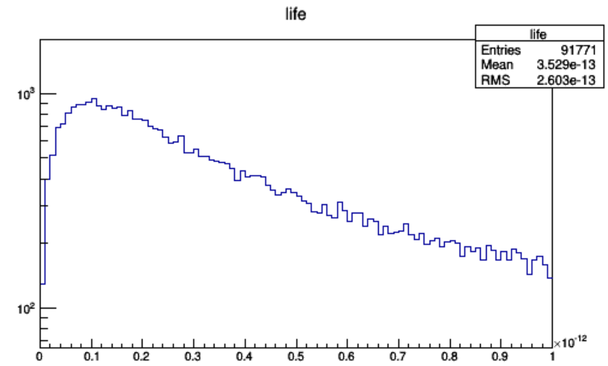
The lifetime of the B^+ meson was then found from the gradient of the approximate straight line from figure 3b, as described in section 3.5. The gradient of this curve was taken by eye, as well as the error bars. This predicted a lifetime of (6.95 ± 4.0) $10^{-14}s$. The error on this value is large due to gradient being taken by eye. The mean lifetime is known to be 1.64×10^{-12}s. The mean lifetime takes into account

Parameters	Value	Error
p0	1.33×10^{2}	1.45×10^{1}
p1	5.28	2.28×10^{-3}
p2	1.90×10^{-2}	2.49×10^{-3}
p3	7.56×10^{2}	3.09

Table 3: Values of the parameters for the fitting function (equation 11) of the B^+ meson histogram.



(a) B^+ meson mass histogram with the fitting function overlayed. x axis is mass in GeV/c^2



(b) B^+ meson lifetime histogram. x axis in s.

Figure 3: B^+ meson mass and lifetime histograms.

all the decay modes. The decay mode we are interested in is equation 1. We can therefore predict the branching ratio of this decay mode to be approximately $4.2 \times 10^{-2} \pm 2.4 \times 10^{-2}$.

4.3 The B_s^0 meson

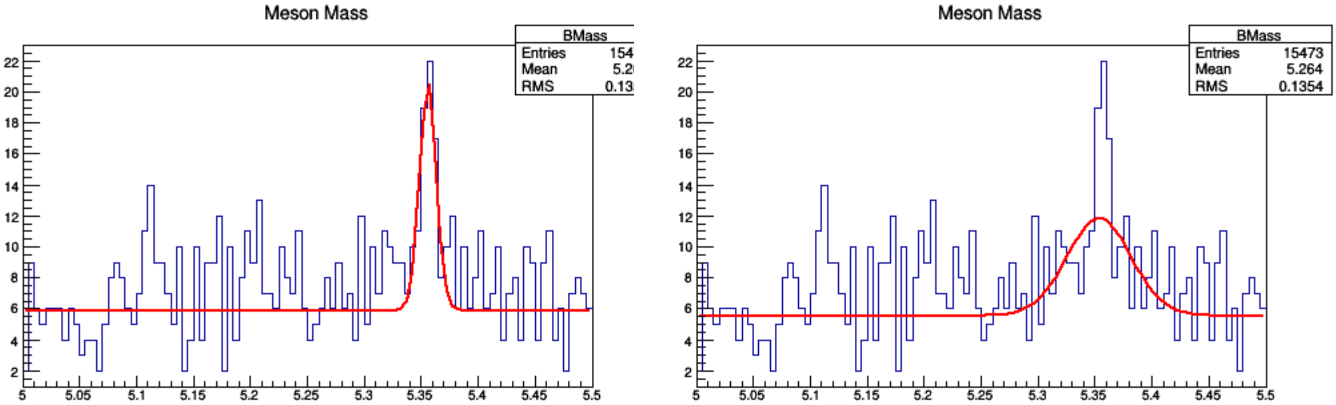
Figure 4 shows the mass histogram for the B_s^0 meson. The peak can be modelled as a gaussian distribution above uniform background noise. Using the fit described in equation 11, the parameters are shown in table 4. There is a high level of noise in this signal from all the incorrect pairings, however the peak is still clear. Taking the uniform background noise to be the mean for no signal, and taking the detection of a particle to have poisson probability, we get

$$\sigma = \sqrt{\mu} = \sqrt{p_0} = 2.42$$
$$\mu + 3\sigma = 13.15$$

The peak is at 20.48, which is well above three sigmas of the uniform background noise. The fit gives a value of $5.25 \pm 0.30\ GeV/c^2$ for the mass, where again the error has been calculated from the FWHM. This value is consistent with the known mass of $5.36679 \pm 0.23\ GeV/c^2$. The gaussian fit that was used was not completely stable. In particular, the parameters p_0 and p_2 were quite unstable, and in some cases a new set of fitting parameters were found, shown in figure 4b.

Parameters	Value	Error
p0	1.47×10^{1}	4.53
p1	5.36	1.53×10^{-3}
p2	7.53×10^{-3}	2.66×10^{-3}
p3	5.89	2.57×10^{-1}

Table 4: Values of the parameters for the fitting function (equation 11) of the B_s^0 meson histogram.



(a) Histogram plot of B_s^0 meson mass. x axis in GeV/c^2. Fitting function (equation 11) overlayed

(b) Same mass plot but with second gaussian fit overlayed, showing instability of fit.

Figure 4: Histogram plots of B_s^0 meson mass

5 Conclusion

The aim of this project was to calculate the mass of the J/ψ, B^+ and B_s^0 mesons, and also to calculate the lifetime of the B^+ meson, with data collected by the CDF at Fermilab. The masses calculated in this project were all in agreement with known values. The mass of the J/ψ meson is found to be 3.09 ± 0.08 GeV/c^2, which is consistent with the known mass of 3.096916 ± 0.011 GeV/c^2. The mass of the B^+ meson if found to be 5.28 ± 0.05 GeV/c^2 which is consistent with the known mass of 5.27929 ± 0.15 GeV/c^2. The mass of the B_s^0 is found to be 5.25 ± 0.30 GeV/c^2 which is consistent with the known mass of 5.36679 ± 0.23 GeV/c^2. This method of calculating the mass of unstable particles has therefore proved to be very effective. The lifetime of the B^+ was calculated to be (6.95 ± 4.0) $10^{-14}s$. The error on this value is quite significant since the gradient was found by eye. The known lifetime is 1.64×10^{-12}s, allowing a prediction of the branching ratio for this decay mode to be approximately $4.2 \times 10^{-2} \pm 2.4 \times 10^{-2}$.

References

[1] *INTRODUCTION TO THE CDF DETECTOR AND THE PARTICLES WE OBSERVE.* `https://www-cdf.fnal.gov/events/detintro.html`. 2005

[2] K. Nakamura et al., `http://pdg.lbl.gov/2010/listings/rpp2010-list-J-psi-1S.pdf`. 2010

[3] Anon, `https://en.wikipedia.org/wiki/B_meson`. 2018

Appendix A Extra Figures

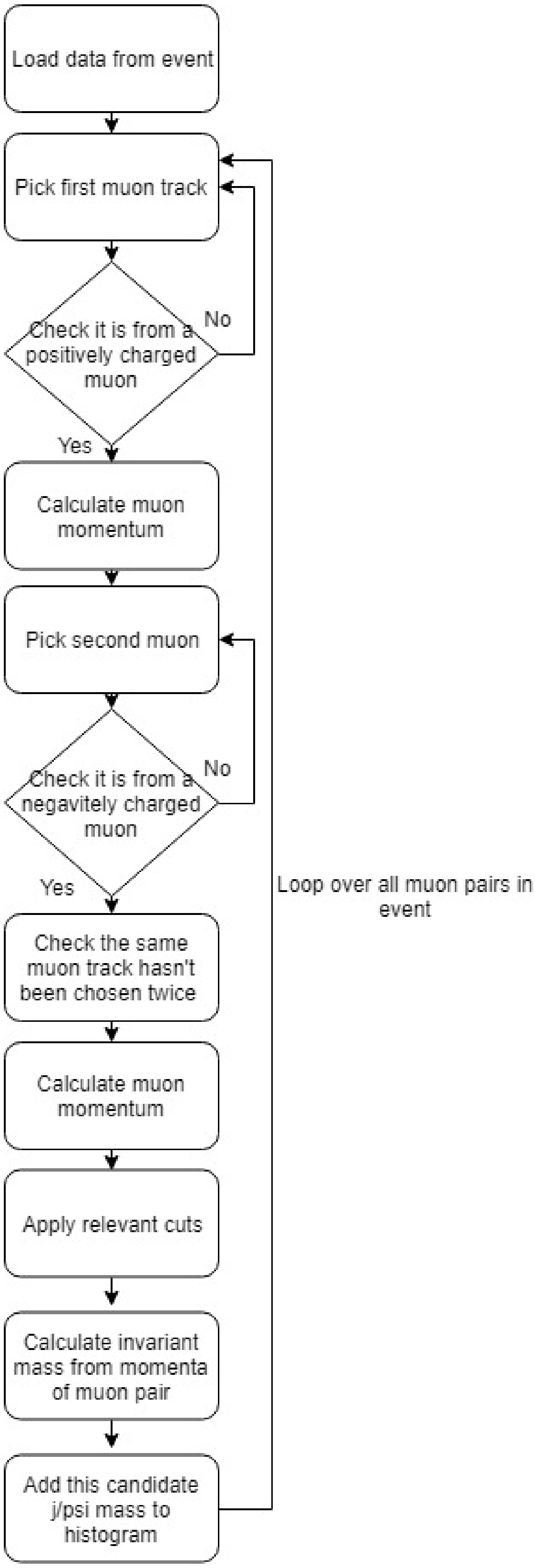
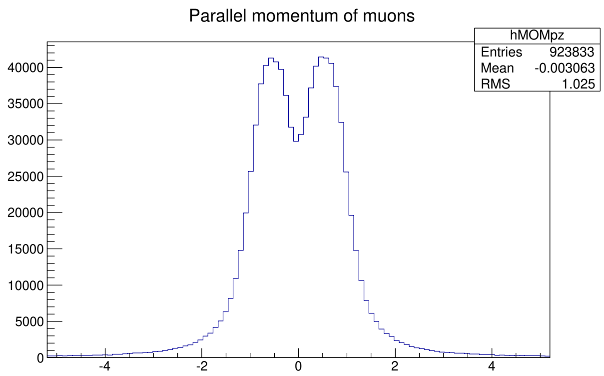
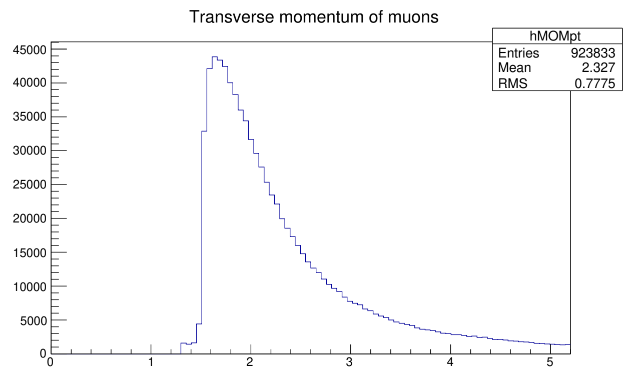


Figure 5: Flow chart of program calculating mass of J/ψ meson.

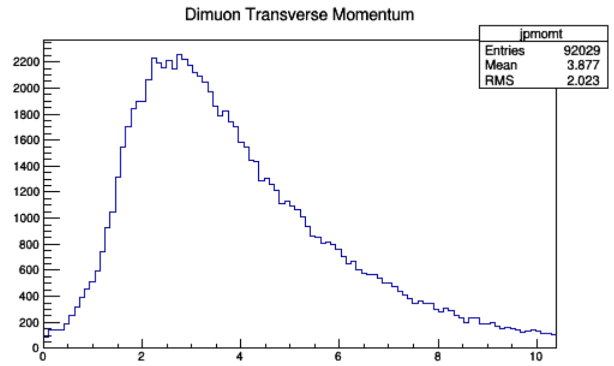


(a) Parallel momentum of muons. x axis in GeV/c. y axis total entries over all events.

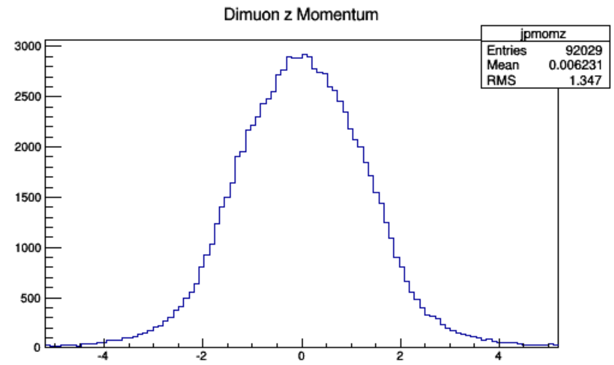


(b) Transvers momentum of muons. Again x axis in GeV/c^2. cut off due to the minimum transverse required for a muon to reach the outer muon chamber.

Figure 6: Histogram plots of muon momenta.

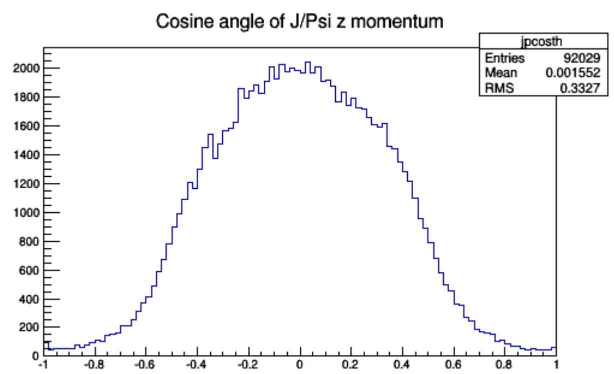


(a) Parallel momentum of J/ψ mesons. x axis in GeV/c. y axis total entries over all events.

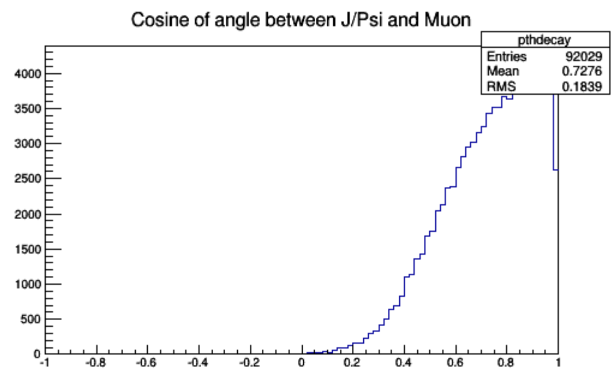


(b) Transverse momentum of J/ψ mesons. x axis in GeV/c. y axis total entries over all events.

Figure 7: Histogram plots of J/ψ meson momenta.



(a) Cosine of the angle of the J/ψ meson with the z axis



(b) Cosine of the angle between the J/ψ meson and the muons it decays into

Figure 8: Angular distribution of J/ψ meson

Appendix B Source Code for Project

```
#define JPsiLab_cxx
#include "JPsiLab.h"
#include <TH2.h>
#include <TStyle.h>
#include <TCanvas.h>
#include <TMath.h>
#include <iostream>
#include "LsqFit.hh"
#include "LsqFit.cc"
double MU_Mass_Gev = 0.105658369;
double K_Mass = 0.493677;
double Mu_Mass_Gev = 0.105658369;
double K_mass = 0.493677;
void JPsiLab::Loop()
{
  //---
  //--- The next section contains details of creating the output tree
  //---
  TFile *f2 = new TFile("phimble.root", "RECREATE");  //make a new root file
  f2->cd();
  TTree *output = new TTree("output","calculated values");  //make a new tree inside
      the file

  struct outvar {          //here we are creating data structures of new variables.
    Float_t jpmass;        //having an existing structure makes it MUCH simpler to
        create new branches

  };                       //and leaves in the tree.  the variables will be filled and
       written to the
  outvar out;              //tree later, but for now we just create space in memory
      for them.

  struct james {
    Float_t B_mass;
    Float_t phimass;
    double vx;
    double vy;
  };
  james bertie;
  struct vtxvar {
    Float_t xintersect;
    Float_t yintersect;
    Float_t primx;
    Float_t primy;
    Float_t jpmomz;
    Float_t jpmomt;
    Float_t jpcosth;
    Float_t pthdecay;
  };
  vtxvar outvtx;

  //make new branches and leaves in the "output" tree:
  TBranch *jpmass = output->Branch("jpmass",&out.jpmass,"jpmass/F");
  TBranch *Bmeson = output->Branch("BMass",&bertie.B_mass,"B_mass/F:phimass/F");
  TBranch *vertex = output->Branch("vertex",&outvtx.xintersect,"xintersect/F:
      yintersect:primx:primy:jpmomz/F:jpmomt/F:jpcosth/F:pthdecay/F:vx/d:vy/d");
  if (jpmass) ;     // This just avoids a 'variable not used' message
  if (vertex) ;     // so does this
  if (Bmeson) ;

  //---
  //--- Create new histograms here, by copy-pasting the example
  //--- below and changing the jpm variable to something unique
  //---
  //TH1F *hMOMpt = new TH1F("hMOMpt", "Transverse momentum of muons",100,0.0,5.20);
  //TH1F *hMOMpz = new TH1F("hMOMpz", "Parallel momentum of muons",100,-5.20,5.20);
```

```
//TH1F *hMOMth = new TH1F("hMOMth", "Angle from z of muon momentum",100,-1,1);
//TH1F *hMOMthp = new TH1F("hMOMthp", "Angle from z of Positive muon momentum",100,-1,1);
//TH1F *hMOMthn = new TH1F("hMOMthn", "Angle from z of Negative muon momentum",100,-1,1);
//TH1F *hMOMno = new TH1F("hMOMno", "Number of Muons in event",100,0.0,8.0);
//TH1F *jpm = new TH1F("jpm", "Dimuon Invariant Mass",100,3.00,3.20); //make a new histogram too
//TH1F *jpmomz_hist = new TH1F("jpmomz","Dimuon z Momentum",100,-5.20,5.20);
//TH1F *jpmomt_hist = new TH1F("jpmomt","Dimuon Transverse Momentum",100,0,10.40);
//TH1F *jpcosth_hist = new TH1F("jpcosth", "Cosine angle of J/Psi z momentum",100,-1,1);
//TH1F *pthdecay_hist = new TH1F("pthdecay","Cosine of angle between J/Psi and Muon",100,-1,1);

TH1F *BMass_hist = new TH1F("BMass", "Meson Mass",100,5,5.5); //make a new histogram too

//now retrieve the input data and start looping over entries:
 if (fChain == 0) return;

 Long64_t nentries = fChain->GetEntriesFast();

 Int_t nrunnew = 1;
 Int_t nrunold = 0;

 Long64_t nbytes = 0, nb = 0;
 for (Long64_t jentry=0; jentry<nentries;jentry++) {
    Long64_t ientry = LoadTree(jentry);
    if (ientry < 0) break;
    nb = fChain->GetEntry(jentry);   nbytes += nb;
    if (nrunnew != nrunold) {  //this output helps you be sure your code is running over long intervals!
      std::cout<<"run number ="<<GLB_nrun<<"  entry number ="<<jentry<<std::endl;
      nrunold = nrunnew;
    }
    nrunnew = Int_t(jentry/100000);



    if (MU_nMuon<2.0) continue; //do not bother if fewer than two muons in the event

    //hMOMno->Fill(MU_nMuon);

    Float_t primarysvxx = VERTEX_vtx_svx_beamx+ZVTX_zvtx_zPos[0]*VERTEX_vtx_svx_slopex; //find primary vertex
    Float_t primarysvxy = VERTEX_vtx_svx_beamy+ZVTX_zvtx_zPos[0]*VERTEX_vtx_svx_slopey;
    outvtx.primx = primarysvxx;  //store the primary vtx coordinates in leaves
    outvtx.primy = primarysvxy;

    for (Int_t nMup=0;nMup<MU_nMuon;nMup++){ //muon+ loop
      Int_t trkp = MU_mu_trkIndex[nMup];    //find track corresponding to the muon
      if (trkp>299 || trkp<0) continue;    //eliminating unphysical track numbers
      Int_t mtypep = 0;  //classify muons by detector
      if (MU_mu_fidBMU[nMup]) continue;
      if (MU_mu_fidCMX[nMup]) mtypep = 2;
      if (MU_mu_fidCMU[nMup]) mtypep = 3;
      if (MU_mu_fidCMP[nMup]) mtypep = 5;
      if (MU_mu_fidCMU[nMup]&& MU_mu_fidCMX[nMup]) mtypep = 4;
      if (MU_mu_fidCMU[nMup]&& MU_mu_fidCMP[nMup]) mtypep = 6;
      if (mtypep == 0) continue;
      if (mtypep < 5) continue;  //use only muons that made it to the most shielded detector
      if (TRACK_trk_nonbc_nsvxHits[trkp] < 1) continue;  //SVX tracks only
```

```
      if (TRACK_trk_nonbc_curv[trkp]<0) continue; //Positive muon, uncomment to
          count all muons

      double costh=-1.02;  //Invalid in case p=0
      double p = TMath::Sqrt(MOM_pt[trkp]*MOM_pt[trkp] + MOM_pz[trkp]*MOM_pz[trkp])
          ;
      if (p > 0.) costh=MOM_pz[trkp]/p;


      //if (TRACK_trk_nonbc_curv[trkp]<0) hMOMthp->Fill(costh); //positive muon

      //if (TRACK_trk_nonbc_curv[trkp]>0) hMOMthn->Fill(costh); //negative muon



      //---
      //--- This is the place to insert histograms on single-muon quantities
      //---
      //hMOMpt->Fill(MOM_pt[trkp]);  //Plot the muon transverse momentum
      //hMOMpz->Fill(MOM_pz[trkp]);  //Plot the muon transverse momentum
      //hMOMth->Fill(costh);




      for (Int_t nMum=0;nMum<MU_nMuon;nMum++){ //muon- loop
        Int_t trkm = MU_mu_trkIndex[nMum];
        if (trkm>299 || trkm<0) continue;
        if (trkp == trkm) continue;//probably not needed
        Int_t mtypem = 0;
        if (MU_mu_fidBMU[nMum]) continue;
        if (MU_mu_fidCMX[nMum]) mtypem = 2;
        if (MU_mu_fidCMU[nMum]) mtypem = 3;
        if (MU_mu_fidCMP[nMum]) mtypem = 5;
        if (MU_mu_fidCMU[nMum]&& MU_mu_fidCMX[nMum]) mtypem = 4;
        if (MU_mu_fidCMU[nMum]&& MU_mu_fidCMP[nMum]) mtypem = 6;
        if (mtypem == 0) continue;
        if (mtypem <5) continue;
        if (TRACK_trk_nonbc_nsvxHits[trkm] < 1) continue;
        if (TRACK_trk_nonbc_curv[trkm]>0) continue; //negative muon

        //---
        //--- This is the place to insert histograms on paired muon quantities
        //--- and calculate variables for the output tree
        //---



        double p_m = TMath::Sqrt(MOM_pt[trkm]*MOM_pt[trkm] + MOM_pz[trkm]*MOM_pz[
            trkm]);
        double p_p = TMath::Sqrt(MOM_pt[trkp]*MOM_pt[trkp] + MOM_pz[trkp]*MOM_pz[
            trkp]);
        //Float_t psimass=0;
        Float_t Mu_term_m =  2*MU_Mass_Gev*MU_Mass_Gev;
        Float_t Mu_term_p = MOM_px[trkp]*MOM_px[trkm]  + MOM_py[trkp]*MOM_py[trkm]+
             MOM_pz[trkp]*MOM_pz[trkm];
        Float_t psimass2 = Mu_term_m + 2*(TMath::Sqrt((MU_Mass_Gev*MU_Mass_Gev +
            p_p*p_p)*(MU_Mass_Gev*MU_Mass_Gev + p_m*p_m)) - Mu_term_p);//calculate
            mass of jpsi candidate
        Float_t psimass = TMath::Sqrt(psimass2);


        if (psimass>3.2 || psimass<3.0) continue;  //loose jpsi mass cut
        out.jpmass=psimass;  //store calculated values in leaves


        Float_t ppz = MOM_pz[trkm] + MOM_pz[trkp];
        outvtx.jpmomz = ppz;
```

```
          Float_t ppx = MOM_px[trkm] + MOM_px[trkp];
          Float_t ppy = MOM_py[trkm] + MOM_py[trkp];
          Float_t ppt = TMath::Sqrt(ppx*ppx + ppy*ppy);
          Float_t ppp = TMath::Sqrt(ppx*ppx + ppy*ppy + ppz*ppz);
          outvtx.jpmomt = ppt;

          double costh = ppz/TMath::Sqrt((ppz*ppz + ppt*ppt));
          outvtx.jpcosth = costh;

          double pangle = (MOM_pz[trkp]*ppz + MOM_px[trkp]*ppx + MOM_py[trkp]*ppy) /
              (p_p * ppp);

          outvtx.xintersect=0;    //change these to calculate muon track intersection
          outvtx.yintersect=0;
          outvtx.pthdecay = pangle;

          if(MOM_pt[trkm] <0.8 || MOM_pt[trkp] <0.8) continue; //limiting lower bound
               trans mom


          if(trkp > 75||trkm > 75)
            {
            std::cout<<"James why"<<std::endl;
            return;
            }
          LsqFit l;
          l.push(TRACK_trk_nonbc_phi[trkp] , TRACK_trk_nonbc_d0[trkp] ,TMath::Sqrt(
              TRKDET_trkdet_nonbc_sigD02[trkp]),MOM_pt[trkp]);

          l.push(TRACK_trk_nonbc_phi[trkm], TRACK_trk_nonbc_d0[trkm],TMath::Sqrt(
              TRKDET_trkdet_nonbc_sigD02[trkm]),MOM_pt[trkm]);

          l.fit();

          bertie.vx = l.x()- outvtx.primx;
          bertie.vy = l.y() - outvtx.primy;



          for (Int_t nKa=0;nKa<TRACK_ntrk;nKa++){ //kaon loop
            if(nKa == trkp)
              continue;
               if(nKa == trkm)
              continue;
          if (TRACK_trk_nonbc_nsvxHits[nKa] < 1) continue;  //SVX tracks only
           if (TRACK_trk_nonbc_curv[nKa]<0) continue; //Positive Kaons

          Float_t momt_1 = TMath::Sqrt(MOM_px[nKa]*MOM_px[nKa] + MOM_py[nKa]*MOM_py[
              nKa]);


          Float_t mom_1 = TMath::Sqrt(momt_1*momt_1 + MOM_pz[nKa]*MOM_pz[nKa]);


          //Float_t Bmtr = TMath::Sqrt((ppx+MOM_px[nKa])*(ppx+MOM_px[nKa]) + (ppy+
              MOM_py[nKa])*(ppy+MOM_py[nKa]));
          //if (Bmtr < 6) continue;

          for (Int_t mKa=0;mKa<TRACK_ntrk;mKa++){ //kaon loop
            if(mKa == trkp)
              continue;
             if(mKa == trkm)
              continue;
                if(mKa == nKa)
              continue;
          if (TRACK_trk_nonbc_nsvxHits[mKa] < 1) continue;  //SVX tracks only
           if (TRACK_trk_nonbc_curv[mKa]>0) continue; //Negative kaons
          Float_t momt_2 = TMath::Sqrt(MOM_px[mKa]*MOM_px[mKa] + MOM_py[mKa]*MOM_py[
```

```
            mKa]);
        Float_t mom_2 = TMath::Sqrt(momt_2*momt_2 + MOM_pz[mKa]*MOM_pz[mKa]);
        Float_t dot_Ka = MOM_px[mKa]*MOM_px[nKa] +MOM_py[mKa]*MOM_py[nKa] + MOM_pz[
            mKa]*MOM_pz[nKa];
        Float_t Phimass2 = 2*K_Mass*K_Mass + 2*TMath::Sqrt((K_Mass*K_Mass + mom_1*
            mom_1)*(K_Mass*K_Mass + mom_2*mom_2)) - 2*dot_Ka;
        Float_t Phimass = TMath::Sqrt(Phimass);
        if(0>Phimass || 1.15<Phimass) continue;
        Float_t PhiMomx = MOM_px[mKa]+MOM_px[nKa];
        Float_t PhiMomy = MOM_py[mKa]+MOM_py[nKa];
        Float_t PhiMomz = MOM_pz[mKa]+MOM_pz[nKa];
        Float_t PhiMOM = TMath::Sqrt(PhiMomx*PhiMomx +PhiMomy*PhiMomy +PhiMomz*
            PhiMomz);
        Float_t dot_Phi = PhiMomx*ppx +PhiMomy*ppy + PhiMomz*ppz;
        //Float_t BSmass = Phimass*Phimass + psimass*psimass + 2*TMath::Sqrt((
            Phimass*Phimass + PhiMOM*PhiMOM)*(psimass*psimass + ppp*ppp)) - 2*
            dot_Phi;
        Float_t BS_Ent1 =  2*TMath::Sqrt((K_mass*K_mass + mom_1*mom_1)*(Mu_Mass_Gev
            *Mu_Mass_Gev + p_m*p_m)) + 2*TMath::Sqrt((K_mass*K_mass + mom_1*mom_1)*(
            Mu_Mass_Gev*Mu_Mass_Gev + p_p*p_p));
        Float_t BS_Ent2 =  2*TMath::Sqrt((K_mass*K_mass + mom_2*mom_2)*(Mu_Mass_Gev
            *Mu_Mass_Gev + p_m*p_m)) + 2*TMath::Sqrt((K_mass*K_mass + mom_2*mom_2)*(
            Mu_Mass_Gev*Mu_Mass_Gev + p_p*p_p));
        Float_t BS_dot1 = (MOM_px[mKa]*MOM_px[trkm]  + MOM_py[mKa]*MOM_py[trkm]+
            MOM_pz[mKa]*MOM_pz[trkm]) +  (MOM_px[mKa]*MOM_px[trkp]  + MOM_py[mKa]*
            MOM_py[trkp]+ MOM_pz[mKa]*MOM_pz[trkp]);
          Float_t BS_dot2 = (MOM_px[nKa]*MOM_px[trkm]  + MOM_py[nKa]*MOM_py[trkm]+
              MOM_pz[nKa]*MOM_pz[trkm]) +  (MOM_px[nKa]*MOM_px[trkp]  + MOM_py[nKa]*
              MOM_py[trkp]+ MOM_pz[nKa]*MOM_pz[trkp]);
          Float_t BSmass2 = Phimass2 + psimass2 + BS_Ent1 + BS_Ent2 - 2*BS_dot1 -
              2*BS_dot1;
                Float_t BSmass = TMath::Sqrt(BSmass2);


        if(BSmass>200)
          continue;

        bertie.phimass = Phimass;
        bertie.B_mass = BSmass;
        BMass_hist->Fill(BSmass);


        //Jpcosth_hist->Fill(costh);
        //jpmomz_hist->Fill(ppz);
        //jpmomt_hist->Fill(ppt);
        //pthdecay_hist->Fill(pangle);
        //jpm->Fill(psimass); //fill the histogram with mass values
        output->Fill();  //fill the tree with the calculated leaf values
              } //K2 loop
       }//K1 loop
      } //muon- loop
    }// muon+ loop
  }
  f2->cd();
  output->Write();  //once the tree has been fully filled, write to file

  //---
  //--- Insert Write() statements for each of the histograms you add here
  //---
  //jpm->Write();     //write mass plot histogram
  //jpmomz_hist->Write();
  //jpmomt_hist->Write();
  //jpcosth_hist->Write();
  //pthdecay_hist->Write();
  //hMOMpt->Write(); //write pt histogram
  //hMOMpz->Write(); //write pz histogram
  //hMOMth->Write(); //write theta histogram
  //hMOMthp->Write(); //write theta histogram
```

```
    //hMOMthn->Write(); //write theta histogram
    //hMOMno->Write(); //write theta histogram
    BMass_hist->Write();
    f2->Close();       //close the file
    std::cout<<"loop finished"<<std::endl;
}
```