# RECONSTRUCTION OF W AND Z BOSONS MASS

NATASHA MAHMOOD
M.PHIL ($3^{RD}$ SEMESTER)
**SUPERVISIOR: DR. ASHFAQ AHMAD**
**CO-SUPERVISIOR: DR. TEEBA RASHID SAAD**

# CONTENTS

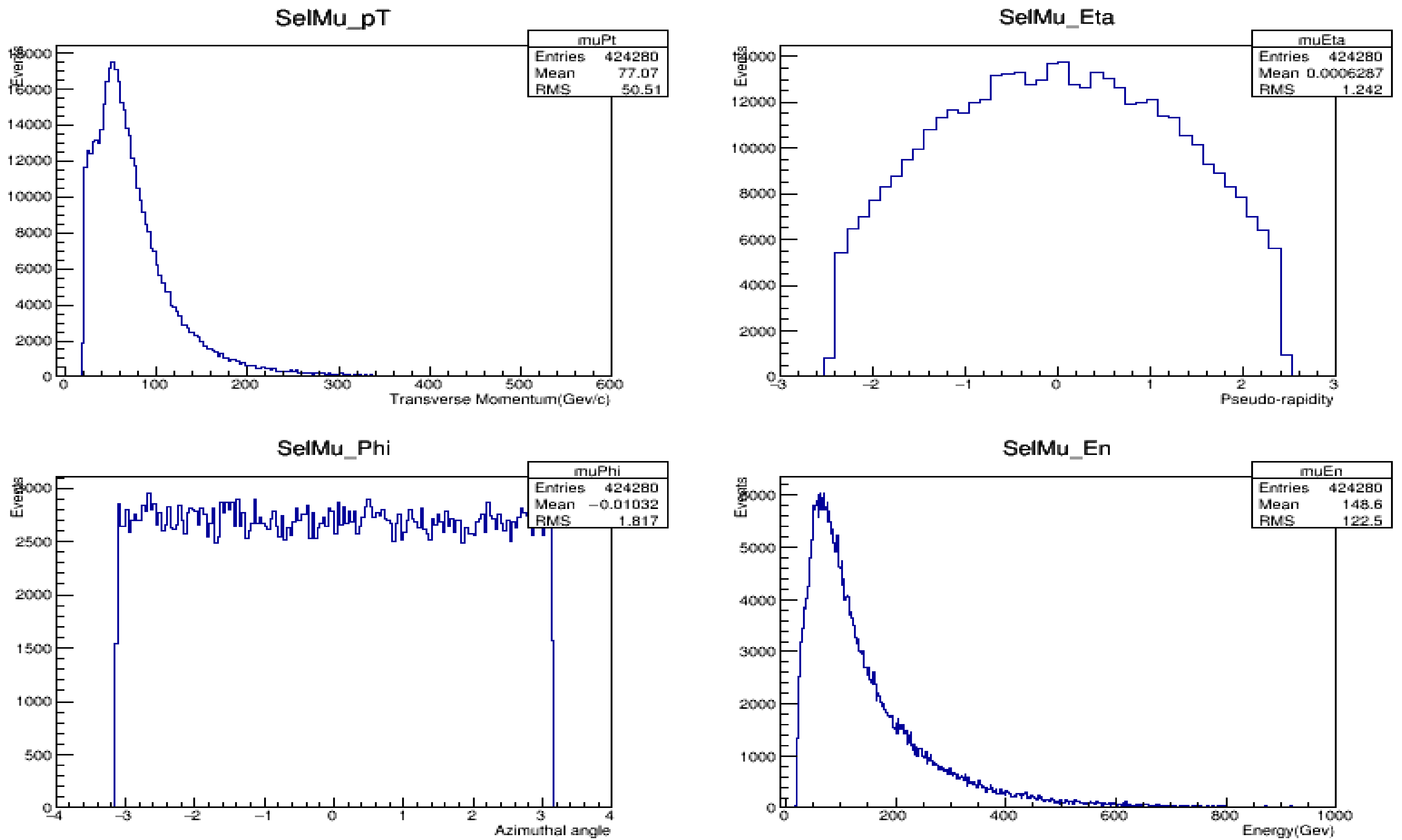
# READ AND ANALYZE ROOT FILE

1. Load Root File
2. Get Tree from Root File
3. Define Variables
4. Set the Tree’s Branches
5. Declare Histograms
6. Fill Histograms using loop
7. Draw Histogram

# READ AND ANALYZE ROOT FILE

```
// Header files
# include "TFile.h"
# include "TTree.h"
# include "TH1F.h"
# include "TBranch.h"
# include "TCanvas.h"
# include <iostream>
void SelMu()
{
// load root file
TFile *f = new TFile ("May26_2.root","READ");
//get the tree AnaTree from the file
TTree *t =(TTree*) f->Get("demo/AnaTree");
//Define variables
Float_t SelMu_pt, SelMu_eta, SelMu_phi, SelMu_en;
//Set the tree's branches to the local variables
t->SetBranchAddress("SelMu_pt", &SelMu_pt);
t->SetBranchAddress("SelMu_eta", &SelMu_eta);
t->SetBranchAddress("SelMu_phi", &SelMu_phi);
t->SetBranchAddress("SelMu_en", &SelMu_en);
//Declare histograms
TH1F *muPt =new TH1F("muPt", "SelMu_pT",200,-10,600);
TH1F *muEta =new TH1F("muEta", "SelMu_Eta",50,-3,3);
TH1F *muPhi =new TH1F("muPhi", "SelMu_Phi",200,-4,4);
TH1F *muEn =new TH1F("muEn", "SelMu_En",500,-10,1000);
//loop over each entry (event) in the tree
Long64_t nentries = t->GetEntries();
for(int i=0; i<nentries; i++)
{
t-> GetEntry(i);
//Filling Histograms
muPt->Fill(SelMu_pt);
```

```
//Filling Histograms
muPt->Fill(SelMu_pt);
muEta->Fill(SelMu_eta);
muPhi->Fill(SelMu_phi);
muEn->Fill(SelMu_en);
}
// Draw Histograms
TCanvas *c1 =new TCanvas("c1","Selected_Muons",800,600);
   c1->Divide(2,2);
   c1->cd(1);
   muPt->GetXaxis()->SetTitle("Transverse Momentum(Gev/c)");
   muPt->GetYaxis()->SetTitle("Events");
   muPt->Draw();

   c1->cd(2);
   muEta->GetXaxis()->SetTitle("Pseudo-rapidity");
   muEta->GetYaxis()->SetTitle("Events");
   muEta->Draw();

   c1->cd(3);
   muPhi->GetXaxis()->SetTitle("Azimuthal angle");
   muPhi->GetYaxis()->SetTitle("Events");
   muPhi->Draw();

   c1->cd(4);
   muEn->GetXaxis()->SetTitle("Energy(Gev)");
   muEn->GetYaxis()->SetTitle("Events");
   muEn->Draw();
}
```


SelMu_pT
muPt
Entries 424280
Mean 77.07
RMS 50.51
Events
Transverse Momentum(Gev/c)
SelMu_Eta
muEta
Entries 424280
Mean 0.0006287
RMS 1.242
Events
Pseudo-rapidity
SelMu_Phi
muPhi
Entries 424280
Mean −0.01032
RMS 1.817
Events
Azimuthal angle
SelMu_En
muEn
Entries 424280
Mean 148.6
RMS 122.5
Events
Energy(Gev)

# Pseudo-Rapidity, Azimuthal Angle and Transverse Momentum

**Pseudo-rapidity** η

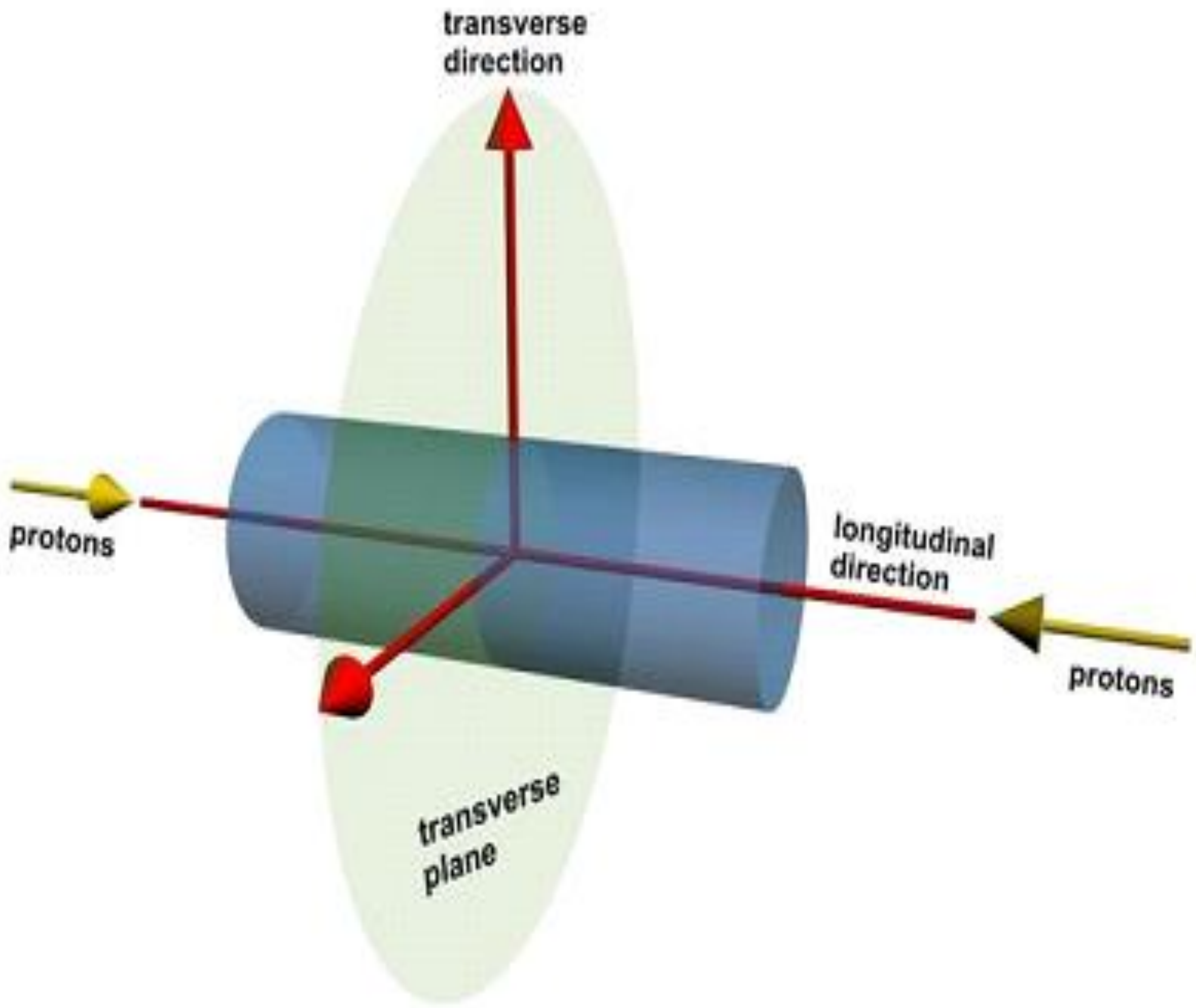
- $\eta$ is spatial co-ordinate describing the angle of particle relative to beam axis (z-axis). It is defined as
- $\eta = -\ln\left[\tan\left(\frac{\theta}{2}\right)\right]$
- Where, θ is the polar angle between the particle three momentum and the positive direction of beam axis

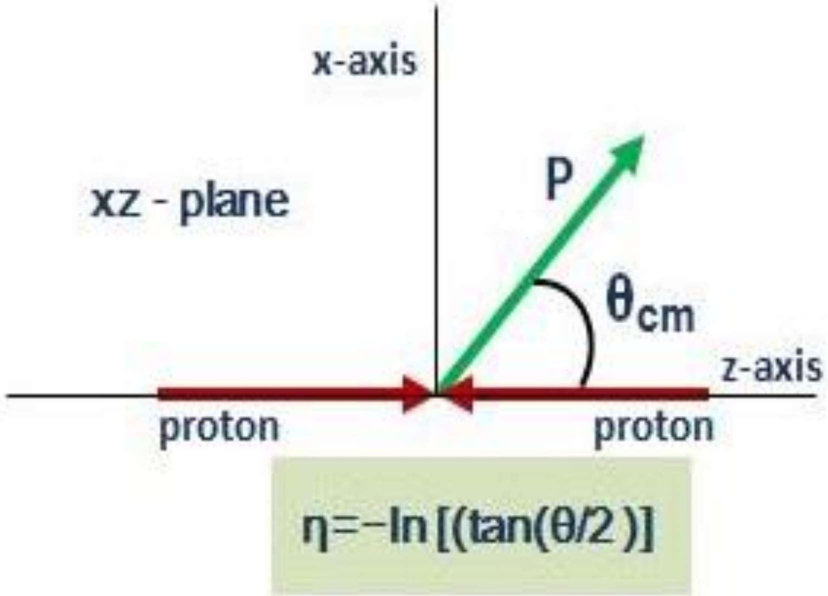
**Azimuthal angle**

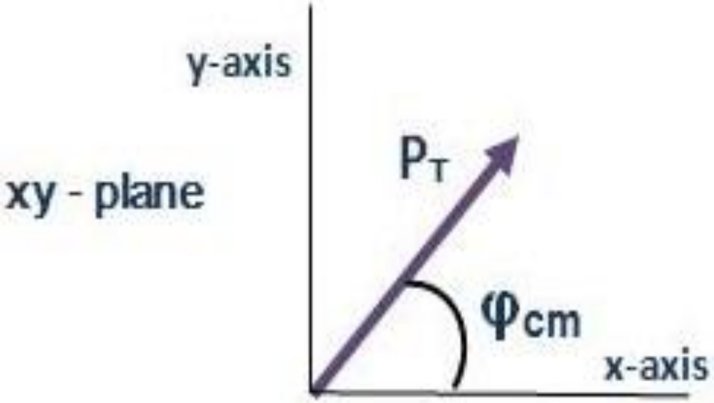
- φ is the angle between transverse momentum and x-axis.

**Transverse Momentum** Pt

- Pt is the momentum perpendicular to the path of the colliding particles.


transverse direction
protons
longitudinal direction
protons
transverse plane

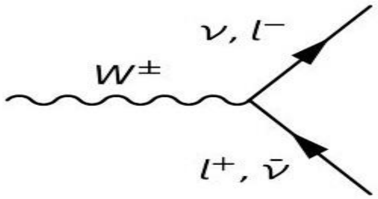

x-axis
xz - plane
P
$\theta_{cm}$
z-axis
proton
proton
$\eta = -\ln[(\tan(\theta/2))]$


y-axis
xy - plane
$P_T$
$\varphi_{cm}$
x-axis

# W and Z Bosons

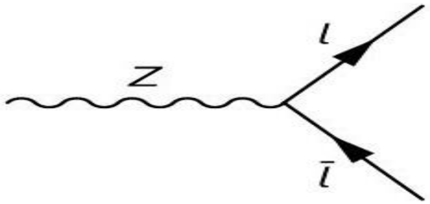
## W Boson

- Mediate weak interaction
- Electric charge $\pm 1$
- Mass 80.385±0.015 GeV/$c^2$
- Spin 1
- Half-life $3 \times 10^{-25}$ s
- Decay
- W boson decay into two quarks or lepton and its corresponding neutrino




## Z Boson

- Mediate weak interaction
- Neutral Particle
- Mass 91.1876±0.0021 GeV/$c^2$
- Spin 1
- Half-life $3 \times 10^{-25}$ s
- Decay
- Z boson decay into pair of quarks or leptons of the same flavor



# Invariant Mass of Z Boson

- In particle physics, the invariant mass is equal to the mass in rest frame of particle.

In Laboratory frame:

According to Law of Conservation of 4-Momentum,

$$P^{\mu} = P_1^{\mu} + P_2^{\mu}$$

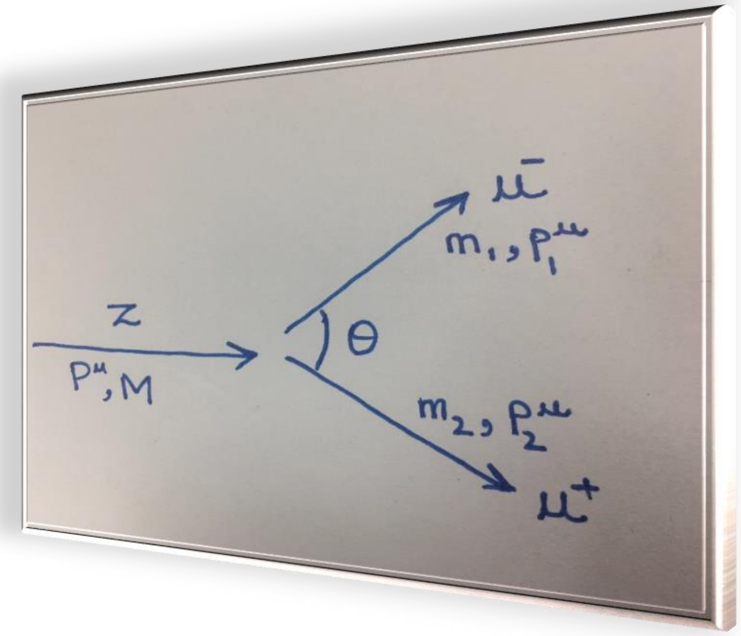
Taking Square on both sides, we have

$$(P^{\mu})^2 = (P_1^{\mu} + P_2^{\mu})^2$$

$$P^{\mu}.P_{\mu} = \left(P_1^{\mu} + P_2^{\mu}\right).(P_{1\mu} + P_{2\mu})$$

Where, $P^{\mu} = (E, 0)$, $P_1^{\mu} = (E_1, \vec{p}_1)$ and $P_2^{\mu} = (E_2, \vec{p}_2)$

$$E^2 = (E_1 + E_2)^2 - (\vec{p}_1 + \vec{p}_2)^2$$



# Invariant Mass of Z Boson

$$E^2 = E_1^2 + E_2^2 + 2E_1E_2 - \vec{p}_1^2 - \vec{p}_2^2 - 2p_1p_2\cos\theta$$

Using Relativistic Energy-Momentum relation, $E^2 = |\vec{p}|^2 + m^2$

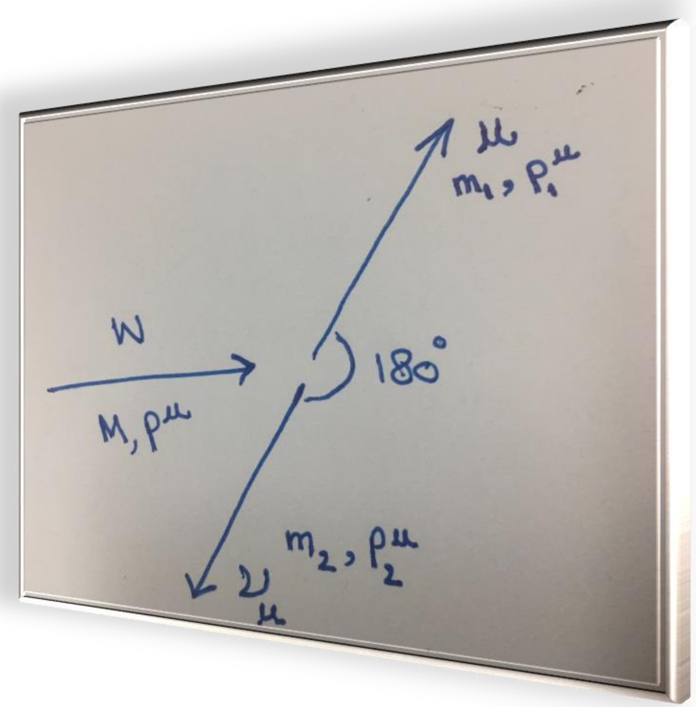
$$M^2 = \vec{p}_1^2 + m_1^2 + \vec{p}_2^2 + m_2^2 + 2E_1E_2 - \vec{p}_1^2 - \vec{p}_2^2 - 2p_1p_2\cos\theta$$

$$\boxed{M^2 = m_1^2 + m_2^2 + 2(E_1E_2 - p_1p_2\cos\theta)}$$

# Transverse Mass of W Boson

- Transverse mass used when one particle can not be detected directly but is only indicated by missing transverse momentum and energy.

According to Law of Conservation of 4-Momentum,




$$P^{\mu} = P_1^{\mu} + P_2^{\mu}$$

Taking Square on both sides, we have

$$(P^{\mu})^2 = (P_1^{\mu} + P_2^{\mu})^2$$

$$P^{\mu}.P_{\mu} = \left(P_1^{\mu} + P_2^{\mu}\right).(P_{1\mu} + P_{2\mu})$$

Where, $P^{\mu} = (E, 0)$, $P_1^{\mu} = (E_{1T}, \vec{p}_{1T})$ and $P_2^{\mu} = (E_{2T}, \vec{p}_{2T})$

# TRANSVERSE MASS OF W BOSON

$$E^2 = (E_{1T} + E_{2T})^2 - (\vec{p}_{1T} + \vec{p}_{2T})^2$$

$$E^2 = E_{1T}^2 + E_{2T}^2 + 2E_{1T}E_{2T} - \vec{p}_{1T}^{\,2} - \vec{p}_{2T}^{\,2} - 2p_{1T}p_{2T}\cos\theta$$

Using Relativistic Energy-Momentum relation, $E_T^2 = |\vec{p}_T|^2 + m^2$

$$M_T^2 = \vec{p}_{1T}^{\,2} + m_1^2 + \vec{p}_{2T}^{\,2} + m_2^2 + 2E_{1T}E_{2T} - \vec{p}_{1T}^{\,2} - \vec{p}_{2T}^{\,2} - 2p_{1T}p_{2T}\cos\theta$$

$$M_T^2 = m_1^2 + m_2^2 + 2(E_{1T}E_{2T} - p_{1T}p_{2T}\cos\theta)$$

For massless particles, $m_1 = m_2 = 0$ and $E_T = p_T$

$$\boxed{M^2 = 2E_{T1}E_{T2}(1 - \cos\theta)}$$

# LORENTZ VECTOR

- TLorentzVector is a general four-vector class, which can be used either for the description of position and time (x, y, z, t) or momentum and energy $(p_x, p_y, p_z, E)$.
- Motion of particles can be described with $(p_x, p_y, p_z, E)$ in Cartesian co-ordinates
- More common used: $(p_T, \eta, \phi, m)$ or $(p_T, \eta, \phi, E)$
- Conversion:

$$p_x = p_T \cos\phi$$
$$p_y = p_T \sin\phi$$
$$p_z = \frac{p_T}{\tan\theta} = p_T \sin h\eta$$
$$|p| = p_T \cos h\eta$$

# LORENTZ VECTOR

- Lorentz vector corresponding to final state particles

TLorentzVector muon;

$(p_x, p_y, p_z, E)$ + Several built in functions

- Using TLorentzVector

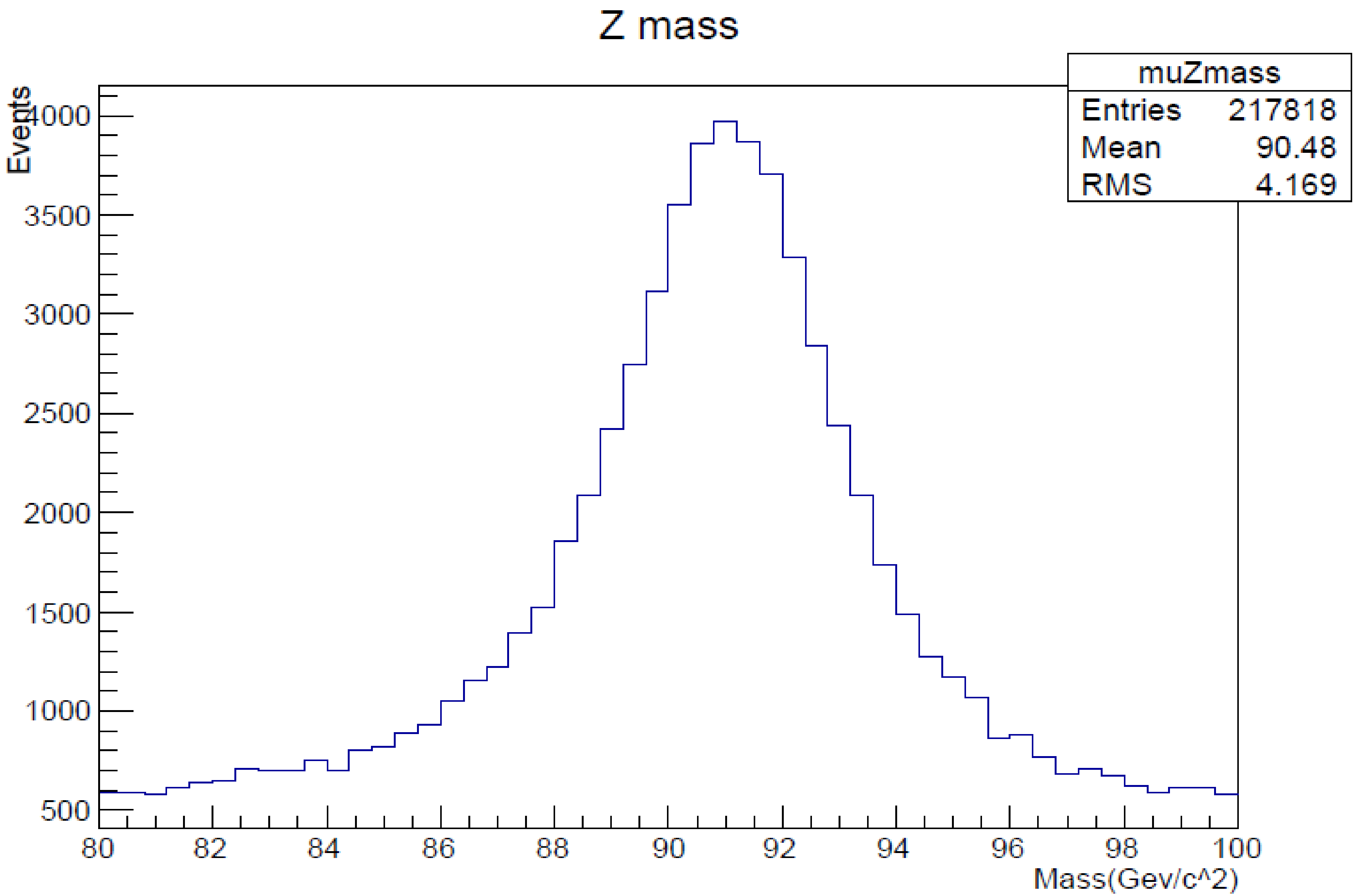
muon.SetPtEtaPhiE( );

Built-in function that defines $(p_x, p_y, p_z, E)$ from $(p_T, \eta, \phi, E)$

# RECONSTRUCTION OF Z MASS

```
# include "TFile.h"
# include "TTree.h"
# include "TCanvas.h"
# include "TH1F.h"
# include "TBranch.h"
# include <iostream>
# include "TLorentzVector.h"
void Zmass()
{
//Load the file
TFile *f = new TFile("May26_2.root","READ");
f->cd();
//get AnaTree from the file
TTree *t1 =(TTree* )f->Get("demo/AnaTree");

//create local variables for the branches
Int_t SelMu;
Float_t SelMu_pt[4],SelMu_eta[4],SelMu_phi[4],SelMu_en[4];

 TLorentzVector M1(0,0,0,0);
 TLorentzVector M2(0,0,0,0);

//set the branches to the local variables
t1->SetBranchAddress("SelMu_pt", &SelMu_pt);
t1->SetBranchAddress("SelMu_eta", &SelMu_eta);
t1->SetBranchAddress("SelMu_phi", &SelMu_phi);
t1->SetBranchAddress("SelMu_en", &SelMu_en);
t1->SetBranchAddress("SelMu", &SelMu);
```

```
//declare Histograms
TH1F *Zmass =new TH1F("muZmass", "Z mass",50,80,100);

//get entries
Long64_t nEntries= t1->GetEntries();

//loop over each entry n in the event
for (int i=0;i<nEntries;i++)
{
t1->GetEntry(i);

for (int m=0;m<SelMu;m++)
{
// by applying cuts
if (fabs(SelMu_pt[m]<20)) continue;
if (SelMu_en[m]<25) continue;
if (SelMu_eta[m]<-2) continue;

if (m==0) M1.SetPtEtaPhiE(fabs(SelMu_pt[m]),SelMu_eta[m],SelMu_phi[m],SelMu_en[m]);
if (m==1) M2.SetPtEtaPhiE(fabs(SelMu_pt[m]),SelMu_eta[m],SelMu_phi[m],SelMu_en[m]);

double ZM = (M1+M2).M();
Zmass->Fill(ZM);
}
}
Zmass->GetXaxis()->SetTitle("Mass(Gev/c^2)");
Zmass->GetYaxis()->SetTitle("Events");

Zmass->Draw();
}
```

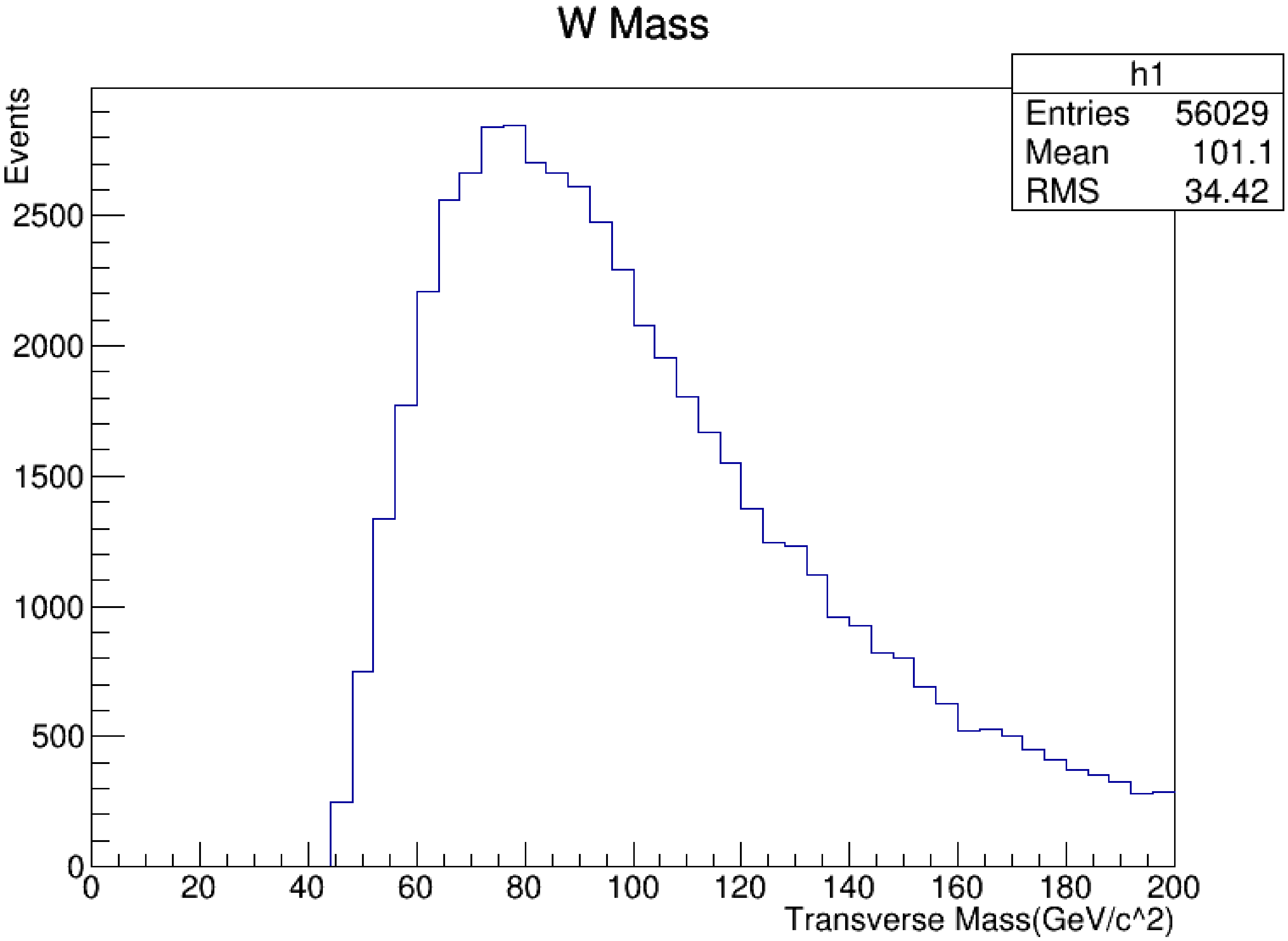


Z mass
muZmass
Entries 217818
Mean 90.48
RMS 4.169
Events
4000
3500
3000
2500
2000
1500
1000
500
80
82
84
86
88
90
92
94
96
98
100
Mass(Gev/c^2)

# RECONSTRUCTION OF W MASS

```cpp
# include "TFile.h"
# include "TTree.h"
# include "TCanvas.h"
# include "TH1F.h"
# include "TBranch.h"
# include <iostream>
# include "TLorentzVector.h"
void Wmass()
{
//Define Canvas
TCanvas *c1=new TCanvas("c1","Transverse Mass",800,600);
c1->cd();
//Load the file
TFile *f = new TFile("May26_2.root","READ");
f->cd();
//get AnaTree from the file
TTree *t= (TTree*)f->Get("demo/AnaTree");
//create local variables for the branches
TLorentzVector mu;
TLorentzVector neu;
Float_t met_pt,met_phi,SelMu_pt[4],SelMu_eta[4],SelMu_phi[4],SelMu_en[4];
Int_t SelMu;
//set the branches to the local variables
t->SetBranchAddress("met_phi", &met_phi);
t->SetBranchAddress("met_pt", &met_pt);
t->SetBranchAddress("SelMu", &SelMu);
t->SetBranchAddress("SelMu_pt", &SelMu_pt);
t->SetBranchAddress("SelMu_eta", &SelMu_eta);
t->SetBranchAddress("SelMu_phi", &SelMu_phi);
t->SetBranchAddress("SelMu_en", &SelMu_en);
//declare Histograms
TH1F *h1 = new TH1F("h1", "W Mass",50,0,200);
```

```cpp
//get entries
Long64_t nentries = t->GetEntries();
//loop over each entry n in the event
for (int i=0;i<nentries;i++)
{
t->GetEntry(i);
//by applying cut
if (met_pt<20) continue;
if (SelMu < 2) continue;
neu.SetPtEtaPhiM(met_pt,0,met_phi,0);
neu.SetPz(0);
neu.SetE(neu.Pt());

for(int m=0;m<SelMu;m++)
{
if (SelMu_en[m]<25) continue;
if (m==2) mu.SetPtEtaPhiE(fabs(SelMu_pt[m]),SelMu_eta[m],SelMu_phi[m],SelMu_en[m]);
}
Double_t TM = (neu+mu).Mt();
h1->Fill(TM);
}
h1->GetXaxis()->SetTitle("Transverse Mass(GeV/c^2)");
h1->GetYaxis()->SetTitle("Events");

// Draw histogram
h1->Draw();
}
```


W Mass
h1
Entries 56029
Mean 101.1
RMS 34.42
Events
2500
2000
1500
1000
500
0
0
20
40
60
80
100
120
140
160
180
200
Transverse Mass(GeV/c^2)

# CONCLUSION

- Read Events $\longrightarrow$ Identify final state particles and get their kinematics information
- Reconstruction algorithm is used to identify different short-lived particles which decay without being directly observed
- Missing energy and momentum are used to infers the presence of non-detectable particles


Thank you