A computing exercise using ROOT

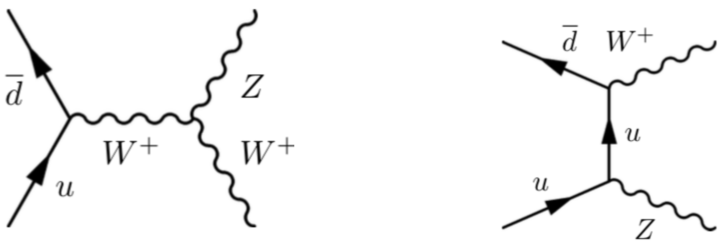
Aim: give a taste of data analysis @ LHC

- What is ROOT ?
  - ROOT is an object-oriented C++ analysis package
  - User-compiled code can be called to produce 1-d, 2-d, and 3-d graphics and histograms…

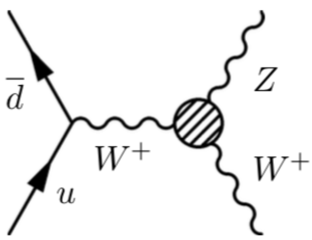
https://root.cern.ch

Hands-on on diboson physics  topic @ LHC

Test of SM



Discover New Physics



Outline

- Kinematic variables used in the analysis of *p - p* collisions
- Useful relations
- Concept of invariant mass (example: 'inclusive' Z boson production)

- Analysis in *p - p* collisions :
  - * Signal: Production of a W and a Z $p - p \rightarrow$ WZ X
  - * Background: Production of a pair of top-antitop

- Example: Macro.C

In all the following slides we assume the speed of the light

c=1

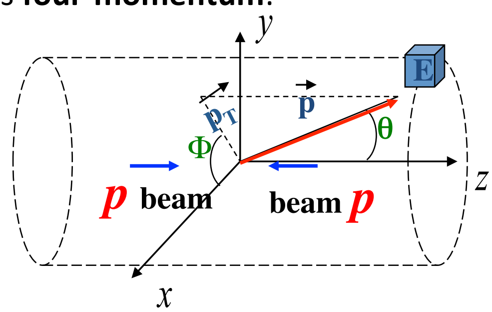
Variables used in the analysis of *p - p* collisions

A particle **(Z, W, e+, e-, etc …)** is described by its **four-momentum**:

$\tilde{p}$ = (E, px, py, pz)

The particle mass is $m = \sqrt{E^2 - p_x^2 - p_y^2 - p_z^2}$

When dealing with ***pp*** collisions the following variables are used:



For each particle (Z, W, e+, e-, etc …):

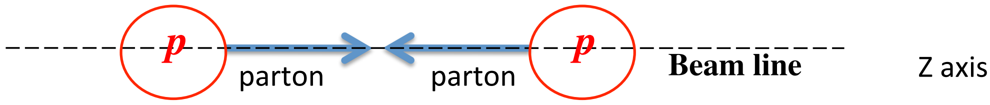
- **1. Transverse momentum/energy :** $p_T = p \sin\theta$ $E_T = E \sin\theta$
- **2. Rapidity** $Y = \frac{1}{2} \ln \frac{E+p_z}{E-p_z}$

  or Pseudorapidity $\eta = -\ln(\tan\frac{\theta}{2})$
- **3. Azimuthal angle** Φ

Why?

Variables used in the analysis of *p - p* collisions

Why p_T, Y ? Many reasons.



1. p_T and ΔY are invariants for Lorentz transformations along the z axis

2. The longitudinal momentum of an initial parton is 'unknown' while we know that $\vec{p}_T^{\text{ initial parton}} \sim 0$
   → Exploit momentum conservation
   use transverse quantities (in the plane ⊥ to the beam) → Example:

$$\Sigma^{\text{initial partons}}\, \vec{p}_T = \Sigma_{\text{vis}}^{\text{fin}}\, \vec{p}_T + \Sigma_{\text{invis}}^{\text{fin}}\, \vec{p}_T \approx 0$$

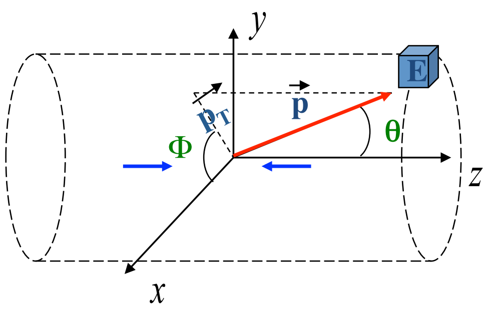
→ Allows to evaluate the p_T of not detected (ν) particles

$$\Sigma_{\text{invis}}^{\text{fin}}\, \vec{p}_T = - \Sigma_{\text{vis}}^{\text{fin}}\, \vec{p}_T$$

$|\Sigma_{\text{invis}}\, p_T|$ is the "missing E_T"

3. The "interesting" physics is due to hard scattering processes →
   high p_T particles (selection of high p_T particles assures "interesting" physics)

Useful relations



$$p_T = p \sin\theta$$

$$px = p_T * \cos(\Phi);$$
$$py = p_T * \sin(\Phi);$$
$$pz = E * \tanh(\eta);$$

$$\eta = -\ln\left(\tan\frac{\theta}{2}\right)$$

NB:

- $m \ll E$ → $Y \approx \eta$ (η doesn't require particle identification)
- $m \ll E$ → $p_T \approx E_T$ $E_T = E \sin\theta$

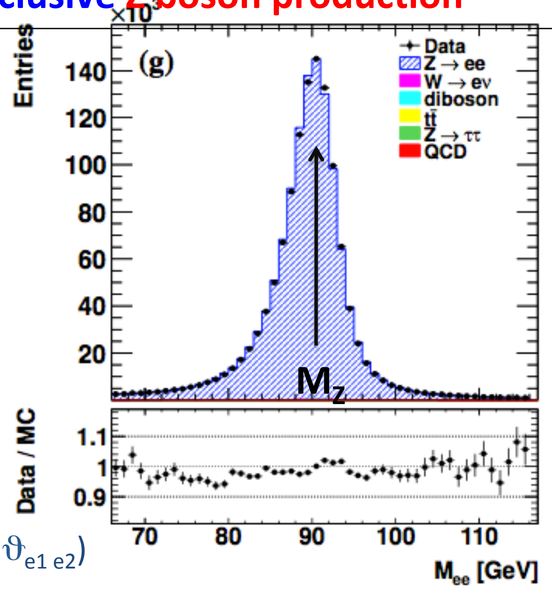
Concept of *invariant* mass: inclusive Z boson production

p - p --> Z (X)

With Z → e+e-

(X) = part1, part2, part3, …

Very 'clean' processes (low bkg)!!



Invariant mass M_{ee} of ee system from the 4-momentum conservation (it allows to measure the Z mass, $\mathbf{M_Z}$):

$$\tilde{p}_Z^2 = (\tilde{p}_{e1} + \tilde{p}_{e2})^2$$

$$M_{ee}^2 = (\tilde{p}_{e1} + \tilde{p}_{e2})^2 \approx 2\,(E_{e1} E_{e2} - |\vec{p}_{e1}|\,|\vec{p}_{e2}| \cos\vartheta_{e1\,e2})$$

$$M_{ee} \approx \sqrt{2\, E_{e1} E_{e2}\ (1-\cos\vartheta_{e1\,e2})}$$

(the electron mass is neglected)

Why M_{ee} gives a distribution and not a single value?

1. $\Delta E * \Delta t > \hbar/2$ $\Delta m * \tau > \hbar/2$

   $\Gamma * \tau > \hbar/2$

   width lifetime

2. Experimental resolution

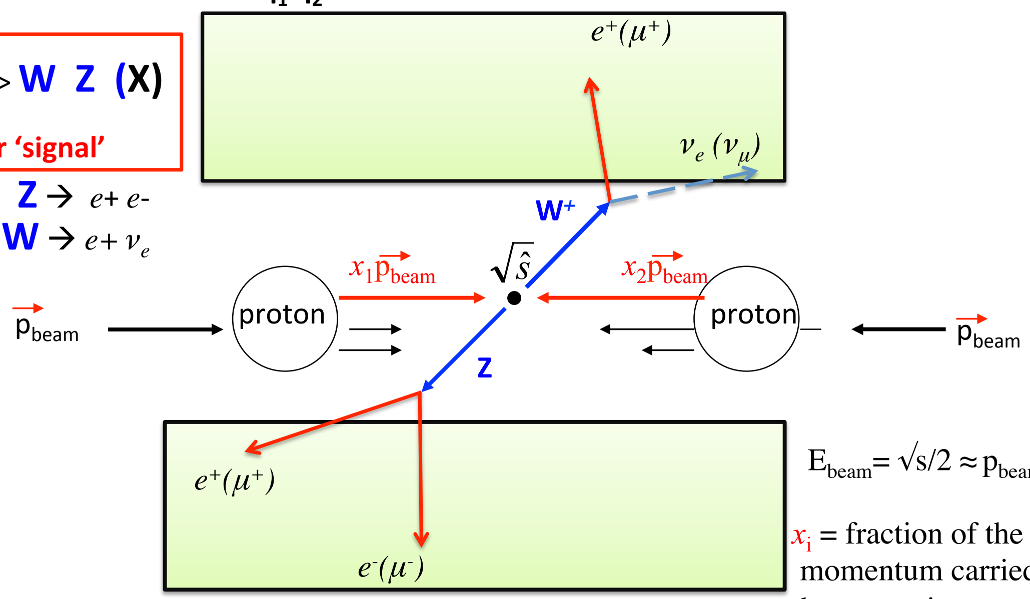
Our signal : production of a W and a Z

p- *p* 'hard' collisions in the $q_1\bar{q}_2$ center of mass:

p- p -> W Z (X)

Our 'signal'

With Z → $e+\ e-$
and W → $e+\ \nu_e$



$E_{beam} = \sqrt{s}/2 \approx p_{beam}$

x_i = fraction of the beam momentum carried by the parton i

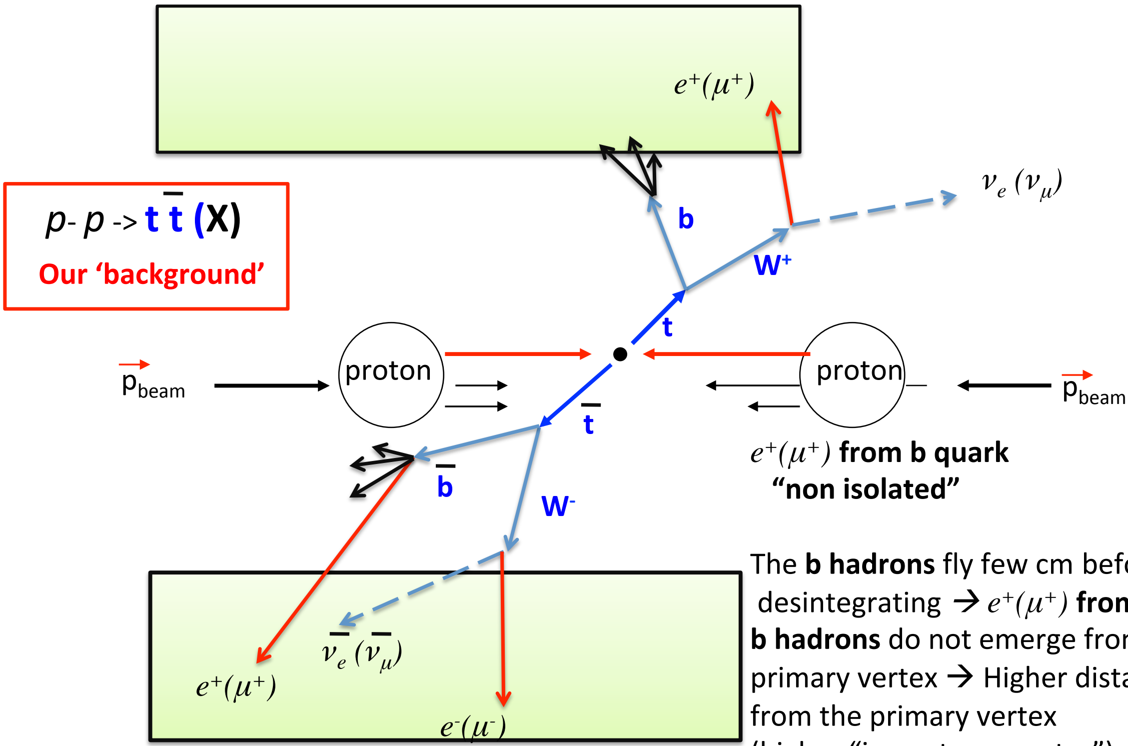
$0 < x_{1,2} < 1$

Kinematics of *p* - *p* collisions

✸ **4-mom of the initial partons** : $[\,(x_1+x_2)E_{beam}, 0, 0, (x_1-x_2)\, p_{beam}]$

Our background: Production of a pair of top-antitop

p- *p* 'hard' collisions. In the $q_1 \bar{q}_2$ center of mass:



The **b hadrons** fly few cm before desintegrating → $e^+(\mu^+)$ **from b hadrons** do not emerge from the primary vertex → Higher distance from the primary vertex (higher "impact parameter")

Aim of the exercise (note, this are steps of an analysis):

1) look at some important variables,
2) build the Z invariant mass,
3) how one can discriminate between the 'signal' and the 'background' ?

You will have:

GRASPA2022explanation.pptx.pdf (this slides)

Exercise2022.pdf (what we ask to do)
Selected_All_EEM.root (« data » (simulated data))
macro.C (draft of an analysis program)
macro_final.C (solution: final analysis program)

https://root.cern.ch/root/htmldoc/guides/primer/ROOTPrimer.html

1) an input file containing the physics: Selected_All_EEM.root

=== MOST ENERGETIC LEPTON FROM THE Z
Br 4 :pt1 : pt1
Br 5 :eta1 : eta1
Br 6 :phi1 : phi1
Br 7 :E1 : E1

=== SECOND ENERGETIC LEPTON FROM THE Z
Br 8 :pt2 : pt2
Br 9 :eta2 : eta2
Br 10 :phi2 : phi2
Br 11 :E2 : E2

=== LEPTON FROM W
Br 12 :pt3 : pt3
Br 13 :eta3 : eta3
Br 14 :phi3 : phi3
Br 15 :E3 : E3

List of variables given per each collision event (kinematics of the final state leptons)

2) Instructions to make the computing exercise : Exercise2022.pdf

COMPUTING EXERCISE

Study of the production of a pair of gauge bosons (W and Z) at the LHC

The data to analyse are organised into a 'Root n-tuple' which we will provide to you. The Root n-tuple is a file containaining information about the kinematics of "events", each resulting from a proton-proton interaction.
These events have three leptons (electrons or muons) and are of two kinds:

3) A skeleton of an analysis program using ROOT: macro.C

```
#include "TCanvas.h"
#include "TROOT.h"
#include "TFile.h"
#include "TTree.h"
#include "TBrowser.h"
#include "TH2.h"
#include "TRandom.h"

void tree1r()
{
  // Read Selected_All_EEM.root file
  //Root file
  TFile *f = new TFile("Selected_All_EEM.root");

  // Signal events
  TTree *sig = (TTree*)f->Get("WZSignal");
  Double_t pt1, eta1, phi1, E1;
  Double_t pt2, eta2, phi2, E2;
  Double_t pt3, eta3, phi3, E3;
  Double_t MZ, MET, trackd0cutWMu, TrackIsoWmu;
  Double_t Weight;

  //get some variables for SIGNAL EVENTS
  sig->SetBranchAddress("pt1",&pt1);
```


Example of analysis program


```
#include "TCanvas.h"
#include "TROOT.h"
#include "TFile.h"
#include "TTree.h"
#include "TBrowser.h"
#include "TH2.h"
#include "TRandom.h"
```

Header files

```
void tree1r()
{
  // Read Selected_All_EEM.root file
  //Root file
  TFile *f = new TFile("Selected_All_EEM.root");
```

Open the input file

```
  // Signal events
  TTree *sig = (TTree*)f->Get("WZSignal");
  Double_t pt1, eta1, phi1, E1;
  Double_t pt2, eta2, phi2, E2;
  Double_t pt3, eta3, phi3, E3;
  Double_t MZ, MET, trackd0cutWMu, TrackIsoWmu;
  Double_t Weight;
```

Access the Signal info

```
  //get some variables for SIGNAL EVENTS
  sig->SetBranchAddress("pt1",&pt1);
  sig->SetBranchAddress("eta1",&eta1);
  sig->SetBranchAddress("phi1",&phi1);
  sig->SetBranchAddress("E1",&E1);
  sig->SetBranchAddress("MZ",&MZ);
  sig->SetBranchAddress("Weight",&Weight);
  // add other variables ...
```

Define the name variables per each SIGNAL lepton

```
////get some variables for BACKGROUND EVENTS
TTree *ttbar = (TTree*)f->Get("ttbar");
Double_t pt1_bkg, eta1_bkg, phi1_bkg, E1_bkg;
Double_t MZ_bkg;
Double_t Weight_bkg;
```

Access the background info

```
//get some variables for ttbar
ttbar->SetBranchAddress("pt1",&pt1_bkg);
ttbar->SetBranchAddress("eta1",&eta1_bkg);
ttbar->SetBranchAddress("phi1",&phi1_bkg);
ttbar->SetBranchAddress("E1",&E1_bkg);
ttbar->SetBranchAddress("MZ",&MZ_bkg);
ttbar->SetBranchAddress("Weight",&Weight_bkg);
// add other variables ...
```

Define the name variables per each bkg lepton

```
//create two histograms (for sig and ttbar)
TH1F *h_MZ    = new TH1F("h_MZ","MZ distribution All events",40,65,115);
TH1F *h_MZ_bkg    = new TH1F("h_MZ_bkg","MZ distribution BKG",40,65,115);
TH1F *h_MZ_sig    = new TH1F("h_MZ_sig","MZ distribution SIG",40,65,115);

//read all SIGNAL entries and fill the histograms
Int_t nentries = (Int_t)sig->GetEntries();

for (Int_t i=0;i<nentries_bkg;i++) {
  ttbar->GetEntry(i);
  h_MZ_bkg->Fill(MZ_bkg,Weight_bkg);
  h_MZ->Fill(MZ_bkg,Weight);
 }
```

Loop on events

Page 1 of 2


```
// example how Draw and save histograms
 TCanvas *c = new TCanvas();
 c->cd();
 h_MZ_sig->Draw();
 h_MZ_bkg->SetLineColor(kRed);
 h_MZ_bkg->Draw("same");

 c->Print("test_MZ.eps");
}

void macro()
{
 tree1r();
}
```

Draw and save histograms

Main program

To start root you may type:

root –l
root [1] .x macro.C

and look at what you get ….

Useful in-line commands:

```
TFile f("Selected_All_EEM.root");
f.ls();
WZSignal->Scan( );
WZSignal->Print( );
```

https://root.cern.ch/root/htmldoc/guides/primer/ROOTPrimer.pdf

Have fun !!