4-й семестр, lesson7. Exercise with ATLAS B_s data

- **B_s meson**
- **Contains (b antis) or (antib s)**
- **Mass 5366 MeV**
- **Like the K0, there are 2 lifetimes:**
- **Tau(Bs_H) = 1620 +- 0.007 ps**
- **Tau(Bs_L) = 1423 +- 0.005 ps**
- **ATLAS experiment reconstructs Bs mesons in $B_s \rightarrow J/\psi\phi \rightarrow \mu^+ \mu^- K^+ K^-$ mode**
- **Used trigger based on $J/\psi \rightarrow \mu^+ \mu^-$ decays**
- **ATLAS statistics in Run2 contains nearly 900000 reconstructed events**
  Today we'll try to see oscillations between (b antis)↔(antib s) states

Experimental procedure

- For observation of oscillations needed information about B_s components at the production time, was it (b antis) or (antib s) - so called Tagging. There are several methods for Tagging, using decays of 2-nd B-object in the event: semileptonic decays of 2-nd b-quark to muons, electrons of charge of opposite-side jets.
- Today we'll try new method, selecting candidates for fast charged kaons in narrow cone around Bs direction, with requirement that the K-candidate found in forward hemisphere after Lorentz transformation to the (Bs+K) rest frame. Only cases in one and only one kaon candidate selected for this tagging.

Event selection

- B_s candidates are copied to /nfs/lfi.mipt.su/data/nikola/atlas/Bs_run2_same-side directory, file data15_data16_data17_data18_Combined_p3601.root.
- Reading is arranged using the MakeClass procedure:

- root data15_data16_data17_data18_Combined_p3601.root
-
- TTree * inputBs = (TTree*)gROOT->FindObject("BsBestChi");
- inputBs->MakeClass("oscillation")
- Files: oscillation.h and oscillation.C generated from TTree: BsBestChi
- There are Branches created during tests, it should be disactivated:
- Please comment lines with " same-side " in oscillation.h

Next step: select only needed Branches

- This selection significantly accelerates the reading
- Please look in oscilation.C file and include in the following statements:
- fChain->SetBranchStatus("*",0); // disable all branches
- fChain->SetBranchStatus("B_mass", 1);
- fChain->SetBranchStatus("B_chi2_ndof", 1);
- fChain->SetBranchStatus("B_A0xy_MinA0", 1); // Bs lifetime
- fChain->SetBranchStatus("new_v2", 1);// tagging

You can look for information in Branches

root –l /nfs/lfi.mipt.su/data/nikola/atlas/Bs_run2_sameside/data15_data16_data17_data18_Combined_p3601.root

Tbrowser b

BsBestChi->Draw("B_mass");

BsBestChi->Draw("B_chi2_ndof");

BsBestChi->Draw("B_tau_MinA0"); // Bs lifetime

BsBestChi->Draw(“new_v2“);   // Tagging

.q

Edit the oscillation.C code

- 1) add Branch selection:
- fChain->SetBranchStatus("*",0);
- fChain->SetBranchStatus("B_mass", 1);
- fChain->SetBranchStatus("B_chi2_ndof", 1);
- fChain->SetBranchStatus("B_A0xy_MinA0", 1);
- fChain->SetBranchStatus("new_v2", 1);
- 2) add historrams:
- TH1D * hist_tau_tagpos;
- TH1D * hist_tau_tagneg;
- TH1D * hist_tau_tagdif;
- hist_tau_tagpos = new TH1D("hist_tau_tagpos|, "hist_tau_tagpos", 104, -0.707725, 3.89249);
- hist_tau_tagneg = new TH1D("hist_tau_tagneg|, "hist_tau_tagpos", 104, -0.707725, 3.89249);
- hist_tau_tagdif = new TH1D("hist_tau_tagdif|, "hist_tau_tagdif", 104, -0.707725, 3.89249);
- hist_tau_tagpos->Sumw2();
- hist_tau_tagneg->Sumw2();
- hist_tau_tagdif->Sumw2();

Writing the code

- 3) after Event Loop add
- TFile *_filewOut= TFile::Open("output.root", "RECREATE");
- hist_tau_tagpos->Write();
- hist_tau_tagneg->Write();
- _filewOut->Close();
- 4) add in Event Loop
- if(jentry%10000 ==1) cout << " jentry = " << jentry << endl;

5) Add preselection:

```
if( B_chi2_ndof < 3.0) {
 if( 5316.<B_mass && B_mass<5416.) {
  if( new_v2 != 0) {

  }
 }
}
```

Writing the code

- 6) fill hist_tau_tagpos and hist_tau_tagneg in previous loop
- if(new_v2 > 0.99) {
- hist_tau_tagpos->Fill(B_tau_MinA0);
- }
- if(new_v2 < -0.99) {
- hist_tau_tagneg->Fill(B_tau_MinA0);
- }
- after Event Loop add hist_tau_tagdif->Add(hist_tau_tagpos, hist_tau_tagneg, 1., -1.);
- 8) fill difference
- hist_tau_tagdif->Add(hist_tau_tagpos, hist_tau_tagneg, 1., -1.);

Histograms

- 9) write histograms to file
- TFile *_filewOut= TFile::Open("output.root", "RECREATE");
- hist_tau_tagpos->Write();
- hist_tau_tagneg->Write();
- hist_tau_tagdif->Write();
- _filewOut->Close();
- 10) The histograms are written to "output.root“ file.
- 11) The hist_tau_tagdif histogram will be used for search of oscillation, starting slightly above from tau=0, make fit of hist_tau_tagdif
-

Produce histograms

- root –l
- chain = new TChain("BsBestChi");
- .L oscillation.C
- oscillation t
- t.Loop()
- Produced file ‘output.root’ with histograms
- hist_tau_tagpos
- hist_tau_tagneg
- hist_tau_tagdif

Final fit

- Create file with with fit function
- vim fit_oscillation.C or write from terminal:
- root -l output.root
- { TFile *_file0 = TFile::Open("output.root");
- TF1 *f12= new TF1("f12", "[0]*(0. + TMath::Cos((x *[1])+ [2]))*TMath::Exp([3]*x)", 0.47, 3.91);
-
- f12->SetParameter(0, 1200.00);
- f12->SetParameter(1, 17.7);
- f12->SetParameter(2, 0.009);
- f12->SetParameter(3, -0.78);
- f12->SetParLimits(0,-0.35 , 290000.0);
- f12->SetParLimits(1, 17.5, 17.9);
- f12->SetParLimits(2, -3.555, 3.002);
- hist_tau_tagdif->Fit("f12","eR+same");

Result

- c1->SaveAs(“hist_tau_tagdif.pdf”, “pdf”);
- C1->SaveAs(“hist_tau_taudif.png”,”png”);
- Parameter p1 gives frequency of oscillation (in inverse ps).
- In Table of particle properties: oscillation frequency 17.749 +- 0.020 ps^{-1}