4-th семестр, Lesson 8, Examples with heavy quarks

- Up family, charge +2/3 : u c t
- Down family, charge -1/3 d s b
- Masses at large momentum transfer, $|t| \to \infty$ («current mass»)
- Mass m(u) ≈0.0023 GeV m(c) ≈ 1.275 Gev m(t) ≈ 173 GeV
- m(d) ≈ 0.0048 GeV m(s) ≈ 0.095 GeV m(b) ≈ 4.18 GeV
- Quarks in light hadrons (connected withother quarks, with low momentum transfer) are significantly more heavy
- Proton p contains (uud), it is stable
- Neutron n contains (udd) -> decays n -> p e^- $\hat{\nu}$
- Quark lifetime decreases with mass, for c and b-quarks the lifetime is of order of 10^{-12} or 10^{-13} sec
- t-quark decays very quickly, Γ≈1.4 GeV (too short to form hadrons)

Photon and vector mesons

- Photon γ has spin 1, negative parity and hegative charge parity (it means that the wave function changes sign in the case of change the sign of space axes or the signof electric charges), γ has $J^{pc} = 1^{--}$
- There are so called vector mesons with $J^{pc} = 1^{--}$:
- ρ and ω mesons (a mixture of $u\bar{u}$ and $d\bar{d}$ quarks)
- ϕ(1020) meson, $s\bar{s}$ system, $BR(\phi \rightarrow \mu^+ \mu^-) = 2.87 \cdot 10^{-4}$
- J/ψ(3096) , $c\bar{c}$ system, $BR(J/\psi \rightarrow \mu^+ \mu^-) = 5.96\%$, also ψ(2S)
- Y(1S), m=9.450 GeV, also Y(2S) Y(3S)

Plan

- We are going to look for new resonance reported by D0 experiment, new (B_s π^{+-}) state with mass 5.668 GeV
- http://arxiv.org/pdf/1602.07588v2.pdf
- ATLAS data acquired in 2015 at 13 TeV will be used
- $B_s(5366)$ meson is reconstructed in decay chains: $B_s \rightarrow J/\psi\phi$
- $J/\psi \rightarrow \mu^+\mu^-$
- $\phi \rightarrow K^+ K^-$ (without kaon identification)
- Extra π^{+-} are taken from tracks which compose the selected primary vertex PV, after cuts on track impact parameters with respect to the PV, and in a cone around the B_s direction with radius R=0.3 ($R = \mathrm{sqrt}(\int \Delta\eta^2 + \Delta\phi^2)$)

D0 result

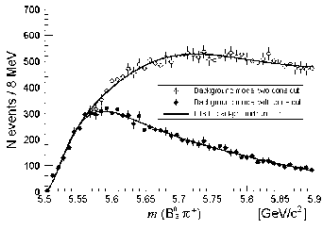

FIG. 2: The combined background for the $m(B_s^0\pi^\pm)$ distribution described in the text and the fit to that distribution with the cone cut and without the cone cut.

The $B_s^0\pi^\pm$ invariant mass spectrum is shown in Fig. 3(a) with the cone cut and (b) without the cone cut. An enhancement is seen near 5.57 GeV/c^2. To extract the signal parameters, the distributions are fitted with a function F (Eq. 2) that includes two terms: the background term $F_{\rm bgr}(m_{B\pi})$ with fixed shape parameters as in Fig. 2 and the signal term $F_{\rm sig}(m_{B\pi}, M_X, \Gamma_X)$, modeled by a relativistic Breit-Wigner function convolved with a Gaussian detector resolution function and with the mass-dependent efficiency of the cone cut [10]. Here M_X and Γ_X are the mass and the natural width of the resonance. The Gaussian width parameter $\sigma_{res} = 3.8$ MeV/c^2 is taken from simulations.

The fit function has the form:

$$F = f_{\rm sig} \times F_{\rm sig}(m_{B\pi}, M_X, \Gamma_X) + f_{\rm bgr} \times F_{\rm bgr}(m_{B\pi}), \quad (2)$$

where $f_{\rm sig}$ and $f_{\rm bgr}$ are normalization factors.

We use the Breit-Wigner parametrization appropriate for an S-wave two-body decay near threshold:

$$BW(m_{D\pi}) \propto \frac{M_X^2 \Gamma(m_{B\pi})}{(M_X^2 - m_{D\pi}^2)^2 + M_X^2 \Gamma^2(m_{D\pi})}. \quad (3)$$

The mass-dependent width $\Gamma(m_{B\pi}) = \Gamma_X \cdot (q_1/q_0)$ is proportional to the natural width Γ_X, where q_1 and q_0 are three-vector momenta of the D_s^0 meson in the rest frame of the $B_s^0\pi^\pm$ system at the invariant mass equal to $m_{D\pi}$ and M_X, respectively.

In the fit shown in Fig. 3a, the normalization parameters $f_{\rm sig}$ and $f_{\rm bgr}$ and the Breit-Wigner parameters M_X and Γ_X are allowed to vary. The fit yields the mass and width of $M_X = 5567.8 \pm 2.9$ MeV/c^2, $\Gamma_X = 21.9 \pm 6.4$ MeV/c^2, and the number of signal events of $N = 133 \pm 31$. As the measured width is significantly larger than the experimental mass resolution, we infer that $X(5568) \to B_s^0\pi^\pm$ is a strong decay. The statistical significance of the signal is defined as $\sqrt{-2\ln(\mathcal{L}_0/\mathcal{L}_{\rm max})}$,

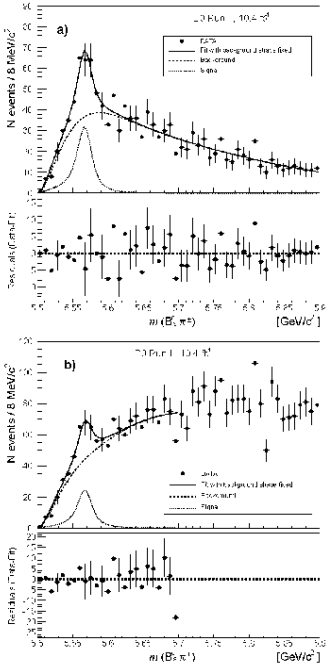

FIG. 3: The $m(B_s^0\pi^\pm)$ distribution together with the background distribution and the fit results (a) after applying the cone cut and (b) without the cone cut.

where $\mathcal{L}_{\rm max}$ and $\mathcal{L}_0$ are likelihood values at the best-fit signal yield and the signal yield fixed to zero. The obtained local statistical significance is 6.6σ for the given mass and width values. With the look-elsewhere effect [11] taken into account, the global statistical significance is 6.1σ. The search window is taken as the interval between the $D_s^0\pi^\pm$ threshold (5508 MeV/c^2) and the D_d^0K mass threshold (5774 MeV/c^2).

We also extract the signal from the $m(B_s\pi^\pm)$ distribution without the ΔR cone cut, fixing the mass and natural width of the signal and the background mass shape to their default values. We see a tendency for data to exceed background for $m(B_s\pi^\pm) > M_X$ [10]. We perform a fit in the restricted range $m(D_s^0\pi^\pm) < 5.7$ GeV/c^2

Look for J/ψ in ATLAS data

- New Ntuple format, needed MakeClass:
- TFile * _file0 = TFile::Open("/nfs/lfi.mipt.su/data/nikola/atlas/2015/BPHY5_bs_plus_pv/2015J/user.nikola.8011000._000074.DefaultOutput.root");
- TTree * read_tree = (TTree*)gROOT->FindObject("tree");
- read_tree -> MakeClass("read_bs");
- Example in /nfs/lfi.mipt.su/data/nikola/example/20160404/makeClass_1.C

- Files: read_bs.h and read_bs.C generated from TTree: tree

Read several Ntuple files, needed «chain»

- Edit read_bs.C , create command file bs_chain.C
- {
- TChain * chain = new TChain("tree","");
- chain->

Add("/nfs/lfi.mipt.su/data/nikola/atlas/2015/BPHY5_bs_plus_pv/2015J/user.nikola.8011000._000001.DefaultOutput.root");

… (many lines, copy from …/example/makeClass_2.C)

…

read_bs t(chain);

t.Loop();

}

Add executable statements to read_bs.C

- At least add in read_bs.C, in the event Loop:
- if(jentry%10000 == 1) { cout << " jentry = " << jentry << endl; }
- And run the simple example:
- root -l
- .L read_bs.C
- .X bs_chain.C
- event count is printed

Next step: look for J/ψ mass

- Add in suitable places:
- TH1D * hist_jpsi_mass;
- hist_jpsi_mass = new TH1D("hist_jpsi_mass", "hist_jpsi_mass", 120, 2800., 3400.);
- if(Jpsi_mass->size() > 0) {
- for(int Njpsi=0; Njpsi<Jpsi_mass->size(); Njpsi++) {
- hist_jpsi_mass -> Fill(Jpsi_mass -> at(Njpsi));
- }
- }
- TFile *_filewOut= TFile::Open("./histogram.root", "recreate");
- hist_jpsi_mass -> Write();
- _filewOut -> Close();

Now run ROOT again:

- root -l
- .L read_bs.C
- .X bs_chain.C
- .q
- And look for distribution in histogra.root file

More histograms

- TH1D * hist_jpsi_eta;
- TH1D * hist_jpsi_pt;
- hist_jpsi_eta = new TH1D("hist_jpsi_eta", "hist_jpsi_eta", 70, -3.5, 3.5);
- hist_jpsi_pT = new TH1D("hist_jpsi_pT", "hist_jpsi_pT", 100, 0., 50000.);
- hist_jpsi_eta -> Fill(Jpsi_rapidity -> at(Njpsi));
- hist_jpsi_pT -> Fill(Jpsi_pT -> at(Njpsi));
- hist_jpsi_eta -> Write();
- hist_jpsi_pT -> Write();
- Example in /nfs/lfi.mipt.su/data/nikola/example/20160404/makeClass_3.C

Backup : 4-й семестр, lesson 7, Reading and Writing NTuples

- Corrections to previous exercise, Reading:
- 1) in Header section
- static int Itape;
- static int Ispln;
- static int Ievtn;
- static TBranch * b_Itape;
- static TBranch * b_Ispln;
- static TBranch * b_Ievtn;

Corrections for reading

- 2) static TBranch * b_Itape;
- static TBranch * b_Ispln;
- static TBranch * b_Ievtn;
- 3) after opening of input Ntuple, but before the Event Loop:
- fChain->SetBranchStatus("Itape", 1);
- fChain->SetBranchStatus("Ispln", 1);
- fChain->SetBranchStatus("Ievtn", 1);
- 4) fChain->SetBranchAddress("Itape", &Itape, &b_Itape);
- fChain->SetBranchAddress("Ispln", &Ispln, &b_Ispln);
- fChain->SetBranchAddress("Ievtn", &Ievtn, &b_Ievtn);
- 5) Itape = 0; Ispln = 0; Ievtn = 0;

Sequence for Writing

- 1) in Header section:
- static int nrun_t3;
- static int nspl_t3;
- static int nevt_t3;
- static double m3pi;
- 2) before the Event Loop
- TFile *newfile = new TFile("small.root","recreate");
- t3ev= new TTree("t3ev", " short ntuple ");

Writing NTuple

- 3) t3ev->Branch("nrun_t3", &nrun_t3, "nrun_t3/I");
- t3ev->Branch("nspl_t3", &nspl_t3, "nspl_t3/I");
- t3ev->Branch("nevt_t3", &nevt_t3, "nevt_t3/I");
- t3ev->Branch("m3pi_t3", &m3pi_t3, "m3pi_t3/D");
- 4) nrun_t3 = 0;
- nspl_t3 = 0;
- nevt_t3 = 0;
- m3pi_t3 = 0;

Writing NTuple

- 5)In Event Loop, fill variables
- nrun_t3 = ltape;
- nspl_t3 = lspln;
- nevt_t3 = levtn;
- m3pi_t3 = fm3;
- 6) and copy the event data to new Ntuple
- t3ev -> Fill();
- 7) After Event Loop
- t3ev->AutoSave();
- newfile->Close();

Complete previous exercise – produce new Tree

- Starting from previous example in /nfs/lfi.mipt.su/data/nikola/ves/run42/example_4_edited.C
- Produce new tree «omega» with 4 branches:
- Itape, Ispln, Ievtn, m(pi+pi-pi0)
- Useful comand : l9024->Print(), it helps to find the type of objects
- Please do not use the identifiers from input Tree for definition of new structure.
- Write new Tree to file.

MakeClass from produced file

- How to prepare reading of produced file « small.root»
- There is Tree called «t3ev» in this file
- Needed the following command file called setMakeClass.C :
- {
- TFile *_file0 = TFile::Open("small.root");
- TTree * new_tree = (TTree*)gROOT->FindObject("t3ev");
- new_tree->MakeClass("read_tree");
- }
- Then in ROOT :
- .X setMakeClass.C
- Files: read_tree.h and read_tree.C generated from TTree: t3ev

Look for produced read_tree.h and read_tree.h

- Created class read_tree with different methods and statements like
- 1) Double_t m3pi_t3;
- 2) TBranch *b_m3pi_t3;
- 3) fChain->SetBranchAddress("m3pi_t3", &m3pi_t3, &b_m3pi_t3);
- In ROOT section user can do:
- 1) compilation, Root > .L read_tree.C
- 2) declare class Root> read_tree t;
- 3) open input file, find input Tree «input» and call t.Loop(input) with

An example of reading sequence

- root -l
- TFile *_file0 = TFile::Open("small.root");
- TTree * input_tree = (TTree*)gROOT->FindObject("t3ev"   );
- .L read_tree.C
- read_tree t;
- t.Init(input_tree);
- t.Loop( );

- The input Tree name is written in read_tree.h by MakeClass
- If the same Tree is used in reading, then t.Init(…) is not needed

Work within read_tree.C

- For example, user can declare a histogram, book it (before the Loop call), fill it in the Loop. Then after the Loop user can open an output file, write the produce histogram and close the output file. An example of corresponding statements :
- 1) TH1D * hist_m3pi;
- 2) hist_m3pi = new TH1D("hist_m3pi", "hist_m3pi", Nbins, Xlow, Xhigh);
- 3) double value = … ;
- hist_m3pi->Fill(value);
- 4) TFile *_filewOut= TFile::Open("hist_file.root", "recreate");
- hist_m3pi->Write();
- _filewOut->Close();

- Run the t.Loop() (see previous page)

Work with hist_file.root (resonances in ($\pi^+ \pi^- \pi^0$) system

- root –l hist_file.root
- .ls
- hist_m3pi -> Draw();

- Try to fit the distribution like the fit hist_m3pi by a Gauss function in mass range (0.65, 0.90) GeV like the example in /nfs/lfi.mipt.su/data/nikola/ves/run42/example_6_edited.C :
- {
- TFile *_file0 = TFile::Open("hist_file.root");
- TF1 *myfit2 = new TF1("myfit2", "gaus(0)",0.65, 0.9);
- hist_m3pi ->Draw();
- gStyle-> SetOptFit();
- //myfit2 -> SetParameter(0, xxx.);
- //myfit2 -> SetParameter(1, x.x);
- //myfit2 -> SetParameter(2, xxxx0);
- hist_m3pi->Fit("myfit2","eR+");
- }
- needed edition: please set the starting values for 3 parameters in the Gauss

Exercises cont.

- Please compare the fitted curve with histogram.
- 3) In order to improve agreement between data and fit result, let's include a Background term (linear function). An example available in /nfs/lfi.mipt.su/data/nikola/ves/run42/example_7_edited.C{
- TFile *_file0 = TFile::Open("hist_file.root");
- TF1 *myfit5 = new TF1("myfit5","gaus(0)+pol1(3)",0.65, 0.9);
- hist_m3pi->Draw();
- gStyle-> SetOptFit() ;
- //myfit5 -> SetParameter(0, xxxx);
- //myfit5-> SetParameter(1, xxx);
- //myfit5-> SetParameter(2, xxxxx);
- //myfit5-> SetParameter(3, xxxxx);
- //myfit5-> SetParameter(4, xxxxx);
- hist_m3pi->Fit("myfit5","eR+");
- }

Exercises cont.

- Parameters 0, 1, 2 are associated with Gauss
- Parameters 3,4 are associated with the linear fubction (pol1).
- Again, needed a starting point for 3+2 parameters (which might be very approximate)
- Run the fit and look for result. Sttill the χ^2 is bad, but the agreement between the data and the fit result is significantly better.