

**Interpolazione della distribuzione di  
massa invariante (segnale di una particella/risonanza)**

**con *ROOFIT***

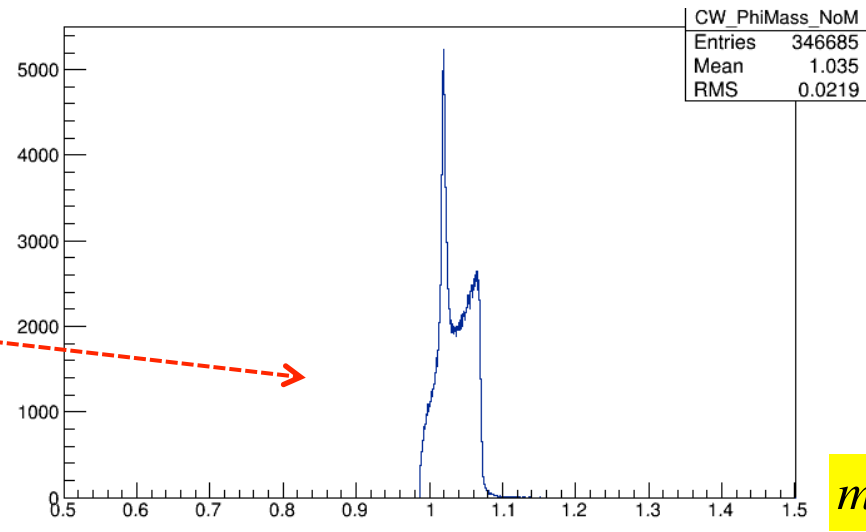
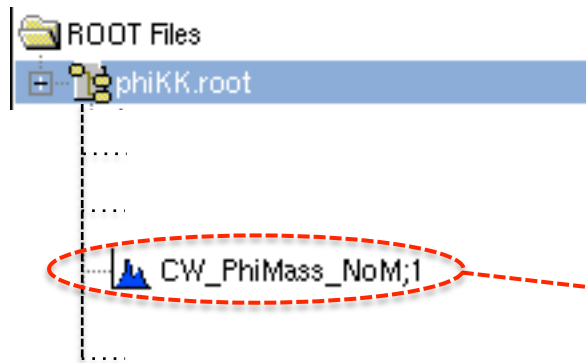
**nel caso la risoluzione in massa sia inferiore alla  
larghezza naturale della particella (caso di  $\phi \rightarrow K^+ K^-$ )**

**Esercitazione del Corso di *Laboratorio Analisi Dati***

2° anno Magistrale / 1° Semestre

**docente: Alexis Pompili**

Preliminarily let us visualize the distribution that should be fitted :

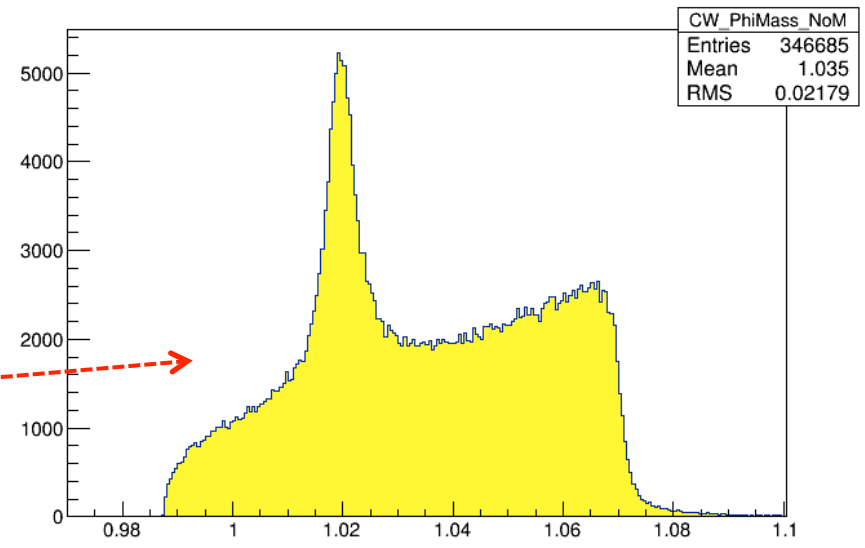


$m(K^+ K^-)$

This histogram represents the **invariant mass of a couple of tracks to which the Kaon mass is assigned** [these data are from CMS]. **The  $\phi$  signal is sitting on a mainly combinatorial background.**

It can be better seen with few settings given sequentially at the prompt:

```
-bash-3.2$ root -l phiKK.root
root [0]
Attaching file phiKK.root as _file0...
root [1] CW_PhiMass_NoM->SetAxisRange(0.97,1.1)
root [2] CW_PhiMass_NoM->SetTitle("")
root [3] CW_PhiMass_NoM->SetFillColor(5)
root [4] CW_PhiMass_NoM->Draw()
Info in <TCanvas::MakeDefCanvas>: created default TCanvas with name c1
root [5]
```



**Exercise:** Try by yourself to write a macro to execute the fit like previously done for the ...  
... signal  $\psi' \rightarrow \mu^+ \mu^-$  !

**For reference** here is some useful code to accomplish the task:

```
////////////////////////////////////  
// run with root: .x phiKK_fit.C  
////////////////////////////////////  
#include <vector>  
  
gROOT->Reset();  
gROOT->Clear();  
  
using namespace RooFit;  
  
void phiKK_fit() {  
    //  
    gROOT->ForceStyle();  
    gStyle->SetTitleOffset(1.4, "Y");  
    gStyle->SetOptFit(1);  
    //  
    TFile* f1 = TFile::Open("./phiKK.root","read");  
    //  
    //////////////////////////////////// PREPARE HISTO  
    //  
    TH1F* hPhiKK;  
    hPhiKK = (TH1F*) f1->Get("CW_PhiMass_NoM");  
    //  
    TCanvas *myC = new TCanvas("myC","PhiMassPlot", 1100, 800);  
    //  
    Double_t xmin = hPhiKK->GetXaxis()->GetXmin();  
    Double_t xmax = hPhiKK->GetXaxis()->GetXmax();  
    Int_t nBins = hPhiKK->GetNbinsX();  
    //  
    //RooRealVar xVar("xVar", "m(K^{+}K^{-}) [GeV/c^{2}]", xmin, xmax);  
    RooRealVar xVar("xVar", "m(K^{+}K^{-}) [GeV/c^{2}]", 0.987, 1.0645);  
    xVar.setBins((nBins/10)*0.2);  
    //  
    RooDataHist* KKHist = new RooDataHist("KK_hist", hPhiKK->GetTitle(), RooArgSet(xVar), Import(*hPhiKK,kFALSE));  
    //  
    //////////////////////////////////// CONFIGURE and MAKE FIT  
    //
```

```

////////////////////////////////////// CONFIGURE and MAKE FIT
//
// --- SIG
RooRealVar mean("mean","mean of voigtian", 1.0197, 1.019, 1.0205);
RooRealVar gamma("gamma","width of BW", 0.0045, 0.001, 0.01);
RooRealVar sigma("#sigma", "mass resolution", 0.0013); // to set at the MC value of 1.3MeV
RooVoigtian sigPDF("sigPDF", "sigPDF", xVar, mean, gamma, sigma);
//
// --- BKG
RooRealVar c0("c_{0}", "c0", 0.001, -15, 15);
RooRealVar c1("c_{1}", "c1", -0.1, -15, 15);
RooRealVar c2("c_{2}", "c2", -0.1, -15, 15);
RooRealVar c3("c_{3}", "c3", 0.1, -10, 10);
RooRealVar c4("c_{4}", "c4", 0.1, -10, 10);
RooRealVar c5("c_{5}", "c5", 0.05, -10, 10);
RooRealVar c6("c_{6}", "c6", 0.0, -1, 1);
//
RooChebychev bkgPDF("bkgPDF", "bkgPDF", xVar, RooArgSet(c0,c1,c2,c3,c4,c5) );
//
RooRealVar nSig("nSig", "Number of signal candidates ", 5e+5, 1000., 5e+6);
RooRealVar nBkg("nBkg", "Bkg component", 2e+6, 1000., 5e+7);
//
RooAddPdf* totalPDF = new RooAddPdf("totalPDF", "totalPDF", RooArgList(sigPDF, bkgPDF), RooArgList(nSig, nBkg));
//
totalPDF->fitTo(*KKHist, Extended(kFALSE));
//
////////////////////////////////////// PLOT RESULTS

```

Set the mass resolution to the value found in CMS Monte Carlo !

Chebyshev poly of enough high order

(\*) Voigtian (=BW⊗G) Voigtian: :v(x,mean,gamma,sigma)

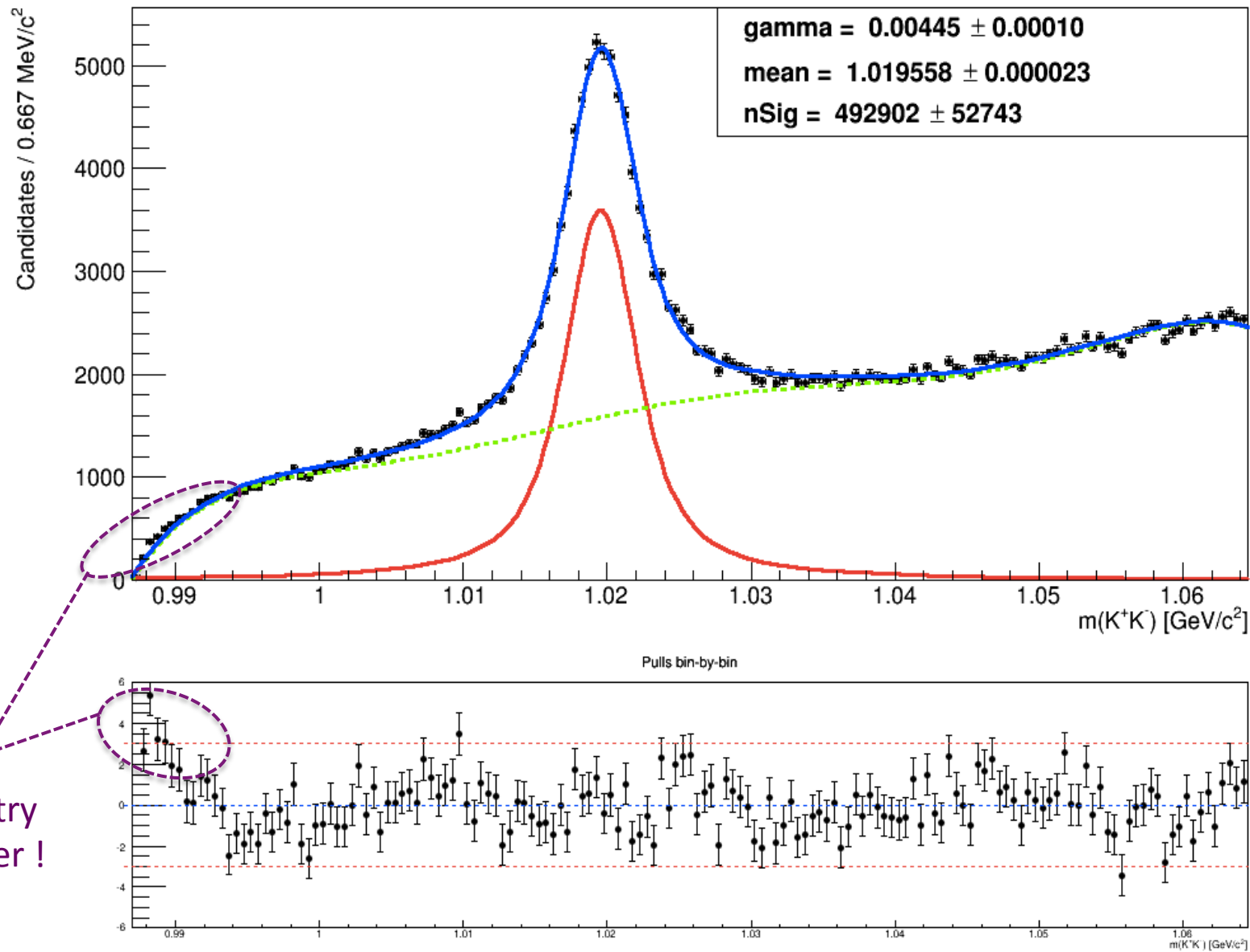
non-relativistic !

```

//////////////////////////////////// PLOT RESULTS
//
RooPlot* xframe = xVar.frame();
//xframe->SetTitle( hPhiKK->GetTitle() );
xframe->SetTitle("");
xframe->SetTitle("Candidates / 0.667 MeV/c^{2}");
//xframe->SetTitleOffset(1.45, "Y");
//
gStyle->SetMarkerSize(0.65);
gStyle->SetMarkerStyle(20);
KKHist->plotOn(xframe);
totalPDF->plotOn(xframe);
//
totalPDF->plotOn(xframe, Components(RooArgSet(sigPDF)), LineColor(kRed));
totalPDF->plotOn(xframe, Components(RooArgSet(bkgPDF)), LineColor(kGreen), LineStyle(kDashed) );
totalPDF->paranOn(xframe, Parameters(RooArgSet(mean,sigma,gamma,nSig)), Layout(0.52,0.9,0.9));
//
totalPDF->plotOn(xframe);
//
myC->cd();
//xframe->Draw();
//
// add the pulls bin-by-bin instead:
//
RooPlot* framePull = xVar.frame();
framePull->SetTitle("Pulls bin-by-bin");
framePull->addObject( (TObject*)xframe->pullHist(), "p" );
framePull->SetMinimum(-6);
framePull->SetMaximum(6);
//
myC->Divide(0,2);
myC->cd(2);
gPad->SetPad(0.,0.,1.,0.3);
framePull->Draw();
TLine *line = new TLine(0.9865, 0., 1.0645, 0.);
line->SetLineColor(kBlue);
line->SetLineStyle(kDashed);
line->Draw("same");
TLine *lineup = new TLine(0.9865, 3., 1.0645, 3.);
lineup->SetLineColor(2);
lineup->SetLineStyle(kDashed);
lineup->Draw("same");
TLine *linedown = new TLine(0.9865, -3., 1.0645, -3.);
linedown->SetLineColor(2);
linedown->SetLineStyle(kDashed);
linedown->Draw("same");
//
myC->cd(1);
gPad->SetPad(0.,0.3,1.,1.);
//
xframe->Draw();
//
//////////////////////////////////// SAVE IN OUTPUT PLOT
//
myC->SaveAs("../Plots/PhiKK-Voigtian.png");
//
////////////////////////////////////
}

```

This code provides the following plot:



## Check that the fit result is compatible with PDG world average :

Citation: C. Patrignani et al. (Particle Data Group), Chin. Phys. C, 40, 100001 (2016) and 2017 update

$\phi(1020)$

$$J^{PC} = 0^{-}(1^{-}-)$$

### $\phi(1020)$ MASS

<u>VALUE (MeV)</u>	<u>EVTS</u>	<u>DOCUMENT ID</u>	<u>TECN</u>	<u>COMMENT</u>
<b>1019.460 ± 0.016 OUR AVERAGE</b>				

### $\phi(1020)$ WIDTH

<u>VALUE (MeV)</u>	<u>EVTS</u>	<u>DOCUMENT ID</u>	<u>TECN</u>	<u>COMMENT</u>
<b>4.247 ± 0.016 OUR AVERAGE</b>				Error includes scale factor of 1.2.

The fit result is:  $m(\phi) \cong [1019.558 \pm 0.023] MeV$  (within  $3\sigma$ )

$\Gamma(\phi) \cong [4.45 \pm 0.10] MeV$  (within  $2\sigma$ )

**NOT BAD !**

Note : some enhancement can be obtained using a relativistic BW convoluted with a gaussian resolution function