Fitting background Pdf check

2D background is estimated by:

$$PDF_{bkg}(x,y) = \mathrm{Argus}(x; m_x, z_x, \rho_x) \times \text{ Argus } (y; m_y, z_y, \rho_y) \times \text{ Student } (x-y; \mu, \sigma(x+y), N).$$

$$\text{Student } (Q; \mu, \sigma, N) = \frac{\Gamma\left(\frac{N+1}{2}\right)}{\sigma\sqrt{N\pi}\Gamma\left(\frac{N}{2}\right)}\left[1+\frac{1}{N}\left(\frac{Q-\mu}{\sigma}\right)^2\right]^{-\frac{N+1}{2}} .$$

$$\mathrm{Argus}(x; m, z, \rho) = Cx\left(1-\frac{x^2}{m^2}\right)^{\rho} e^{z\left(1-\frac{x^2}{m^2}\right)}$$

bkgFunction :

```
const double Igamma0 = -(degreeF+1.0)/2.0;//degreeF:N
const double Igamma1 = (degreeF+1.0)/2.0;
const double Igamma2 = degreeF/2.0;
const double Ipar3 = degreeF*TMath::Pi();
const double Ipar4 = ((x-y-mean)/Isigma0) * ((x-y-mean)/Isigma0);
const double Ipar5 = 1.0 + Ipar4/degreeF;

const double gamma1 = TMath::Gamma(Igamma1);
const double gamma2 = TMath::Gamma(Igamma2);
const double par3 = TMath::Sqrt(Ipar3);
const double par5 = TMath::Power(Ipar5,Igamma0);

double StudentT;

if((degreeF > 0) && (mt > (x+y))){
  StudentT = (gamma1 * par5) / (Isigma0 * gamma2 * par3);
}else{
  StudentT = 0.0;
}

double Argus_x = 0;
if(x<mx){
  Argus_x =  x*(TMath::Power((1-(x/mx)*(x/mx)),rx))*exp(zx*(1-(x/mx)*(x/mx)));
}
double Argus_y = 0;
if(y<my){
  Argus_y =  y*(TMath::Power((1-(y/my)*(y/my)),ry))*exp(zy*(1-(y/my)*(y/my)));  }

double value = StudentT * Argus_x * Argus_y;

return value;
```

Student

Argus x

Argus y

Argus × Argus × Student

```
bkgPdf[g][y] = new bkgFunc(namebkgPdf[g][y],namebkgPdf[g][y], *Valx[g][y], *Valy[g][y], mu, degreeF, sigma0, *mt_xy[g][y], gs, gs0, *mt_x[g][y], *mt_y[g][y], kx, ky, rho_x, rho_y);
```

RooGenericPdf :

```
const double Igamma0 = -(degreeF+1.0)/2.0;//degreeF:N
const double Igamma1 = (degreeF+1.0)/2.0;
const double Igamma2 = degreeF/2.0;
const double Ipar3 = degreeF*TMath::Pi();
const double Ipar4 = ((x-y-mean)/Isigma0) * ((x-y-mean)/Isigma0);
const double Ipar5 = 1.0 + Ipar4/degreeF;

const double gamma1 = TMath::Gamma(Igamma1);
const double gamma2 = TMath::Gamma(Igamma2);
const double par3 = TMath::Sqrt(Ipar3);
const double par5 = TMath::Power(Ipar5,Igamma0);

double StudentT;

if((degreeF > 0) && (mt > (x+y))){
  StudentT = (gamma1 * par5) / (Isigma0 * gamma2 * par3);
}else{
  StudentT = 0.0;
}

return StudentT;
```

Student

```
StudentT[g][y] = new RooStudentT(namestudentT[g][y],namestudentT[g][y], *Valx[g][y], *Valy[g][y], mu, degreeF, sigma0, *mt_xy[g][y], gs, gs0);
```

```
argusx[g][y] = new RooArgusBG(nameargusx[g][y], nameargusx[g][y], *Valx[g][y], *mt_x[g][y], kx, rho_x);
argusy[g][y] = new RooArgusBG(nameargusy[g][y], nameargusy[g][y], *Valy[g][y], *mt_y[g][y], ky, rho_y);
```

Argus x

Argus y

```
bkgpdf[g][y] = new RooGenericPdf(namebkgpdf[g][y], "@0*@1*@2", RooArgList(*argusx[g][y],*argusy[g][y],*StudentT[g][y]));
```

Multiplying PDF components

demo3.cc

RooProdPdf constructs the product of N PDFs:

$$P = P_0(x_1, x_2) \cdot P_1(y_1, y_2, ..) \cdot P_2(z_1, z_2, ...) \cdot ... P_n(w_1, w_2, ...)$$

Build 2 Gaussian PDFs

```
// Build two Gaussian PDFs
RooRealVar x("x","x",-5,5) ;
RooRealVar y("y","y",-5,5) ;
RooGaussian gaussx("gaussx","gaussx",x,meanx,sigmax);
RooGaussian gaussy("gaussy","gaussx",y,meany,sigmay);

// Multiply the components
RooProdPdf prod("gaussxy","gaussx*gaussy",
                RooArgList(gaussx,gaussy)) ;
```

Component PDFs may not share dependents
e.g. $pdf_1(\mathbf{x},y)*pdf_2(\mathbf{x},z)$ not allowed

Such forms are not very common,
but can be performed with RooGenericPdf
Shared parameters no problem

Normalization more complicated

Comparison of the fitting result

	inclusive $\mathbf{1 \times 10^{-4}}$(偏高，实际>$\mathbf{1 \times 10^{-4}}$)	data
my result	-	$\mathbf{(1.72 \pm 0.67) \times 10^{-4}}$
bkg Function value = Argus × Argus × Student → Pdf	$\mathbf{(1.30 \pm 0.10) \times 10^{-4}}$	$\mathbf{(1.63 \pm 0.65) \times 10^{-4}}$
RooGenericPdf	$\mathbf{(3.71 \pm 265) \times 10^{-9}}$	$\mathbf{10^{-10}}$ at limit

backup

My fitting code

定义六个（组）变量

```
RooRealVar* Valx[TOTENERGY];
RooRealVar* Valy[TOTENERGY];
```

```
for(int k=INIENERGY; k<TOTENERGY; k++)
{
  int ENERGY = LambdacENESidx(k);
  double ecm = LambdacEcms(ENERGY);


  Valx[k] = new RooRealVar("sigmBC_d", "sigmBC_d", MinValue[k], MAXMbcVal);
  Valy[k] = new RooRealVar("tagmBC_d", "tagmBC_d", MinValue[k], MAXMbcVal);

}
```

```
data[k] = new RooDataSet(MbcDataStr[k], MbcDataStr[k], RooArgSet(*Valx[k], *Valy[k]), Import(*t_data[k]));
```

```
combDatas[k] = new RooDataSet(ComDataStr[k], "combined data", RooArgSet(*Valx[k], *Valy[k]),
Index(sample), Import(SamplesStr[k], *data[k]));
```

```
Signal[k] = new RooHistPdf(signal[k], "sigPdf", RooArgSet(*Valx[k], *Valy[k]), *SigHistMC[k], 1);
```

```
BifStudent2Dpdf[k] = new RooBifStudent2D(bifStudent2Dpdf[k], bifStudent2Dpdf[k], *Valx[k], *Valy[k],
m, g1, g2, fg, N1, N1, RooConst(MaxValue[k]), RooConst(MaxValue[k]), argus_parx, argus_pary, argus_expx,
argus_expy);
```

```
MySumPdf[k] = new RooAddPdf(mysumPdf[k], mysumPdf[k], RooArgList(*Signal[k], *BifStudent2Dpdf[k]),
RooArgList(*NSignal[k], *NBifStudent2Dpdf[k]));
```

```
RooSimultaneous simPdf("simPdf", "simultaneous pdf", sample);
for(int k=INIENERGY; k<TOTENERGY; k++)
{
  simPdf.addPdf(*MySumPdf[k], SamplesStr[k]);
}
```

```
RooFitResult *result = simPdf.fitTo(*combData, Save(kTRUE));
```

定义一个全局变量

```
RooRealVar* Valx = new RooRealVar("sigmBC_d", "sigmBC_d", 2.2, MAXMbcVal);
RooRealVar* Valy = new RooRealVar("tagmBC_d", "tagmBC_d", 2.2, MAXMbcVal);
```

```
data[k] = new RooDataSet(MbcDataStr[k], MbcDataStr[k], RooArgSet(*Valx, *Valy), Import(*t_data[k]))
```

```
combDatas[k] = new RooDataSet(ComDataStr[k], "combined data", RooArgSet(*Valx, *Valy),
Index(sample), Import(SamplesStr[k], *data[k]));
```

```
Signal[k] = new RooHistPdf(signal[k], "sigPdf", RooArgSet(*Valx, *Valy), *SigHistMC[k], 1);
```

```
BifStudent2Dpdf[k] = new RooBifStudent2D(bifStudent2Dpdf[k], bifStudent2Dpdf[k], *Valx, *Valy,
m, g1, g2, fg, N1, N1, RooConst(MaxValue[k]), RooConst(MaxValue[k]), argus_parx, argus_pary, argus_
expx, argus_expy);
```

```
MySumPdf[k] = new RooAddPdf(mysumPdf[k], mysumPdf[k], RooArgList(*Signal[k], *BifStuden
t2Dpdf[k]), RooArgList(*NSignal[k], *NBifStudent2Dpdf[k]));
```

```
RooSimultaneous simPdf("simPdf", "simultaneous pdf", sample);
for(int k=INIENERGY; k<TOTENERGY; k++)
{
  simPdf.addPdf(*MySumPdf[k], SamplesStr[k]);
}
```

```
RooFitResult *result = simPdf.fitTo(*combData, Save(kTRUE));
```

My fitting code

定义六个（组）变量

```
COVARIANCE MATRIX CALCULATED SUCCESSFULLY
FCN=-102.172 FROM HESSE     STATUS=OK            83 CALLS          504 TOTAL
                     EDM=3.8692e-06    STRATEGY= 1      ERROR MATRIX ACCURATE
 EXT PARAMETER                                INTERNAL      INTERNAL
 NO.   NAME      VALUE            ERROR       STEP SIZE       VALUE
  1  Brfpi0       1.70047e-04   6.65442e-05   1.21753e-03  -7.22907e-01
  2  N1           2.32322e-01   5.50958e-01   3.73933e-03  -1.13627e+00
  3  g2           4.39167e-02   3.00510e-02   1.00952e-03  -1.14854e+00
  4  m           -1.62060e-03   6.13531e-04   4.27280e-05  -1.62067e-02
  5  nstu_4600    1.31275e+00   1.29050e+00   7.89190e-04  -1.34114e+00
  6  nstu_4640    2.82509e+00   1.88228e+00   7.85131e-04  -1.23303e+00
  7  nstu_4660    4.94613e+00   2.23196e+00   7.21164e-04  -1.12225e+00
  8  nstu_4680    3.95564e+00   2.01949e+00   7.25755e-04  -1.17035e+00
  9  nstu_4700    1.82437e+00   1.51941e+00   7.89858e-04  -1.29983e+00
                               ERR DEF= 0.5
```

定义一个全局变量

```
COVARIANCE MATRIX CALCULATED SUCCESSFULLY
FCN=-104.732 FROM HESSE     STATUS=OK            91 CALLS          808 TOTAL
                     EDM=8.46888e-05    STRATEGY= 1      ERROR MATRIX ACCURATE
 EXT PARAMETER                                INTERNAL      INTERNAL
 NO.   NAME      VALUE            ERROR       STEP SIZE       VALUE
  1  Brfpi0       1.44926e-04   5.94818e-05   2.38478e-04  -7.92144e-01
  2  N1           1.22675e-03   7.37980e-03   6.48896e-03  -1.53947e+00
  3  g2           4.71251e-01   6.59251e-01   1.94753e-01  -5.75291e-02
  4  m           -1.54494e-03   5.67780e-04   1.25957e-03  -1.54500e-02
  5  nstu_4600    1.38092e+00   1.33984e+00   1.61562e-04  -1.33523e+00
  6  nstu_4640    3.42261e+00   1.95915e+00   3.04661e-05  -1.19865e+00
  7  nstu_4660    5.00718e+00   2.23459e+00   1.45434e-04  -1.11944e+00
  8  nstu_4680    4.05028e+00   2.02275e+00   1.45619e-04  -1.16552e+00
  9  nstu_4700    2.20743e+00   1.55400e+00   1.49793e-04  -1.27254e+00
                               ERR DEF= 0.5
```

my result 6 variables	$(\mathbf{1.72} \pm \mathbf{0.67}) \times \mathbf{10^{-4}}$ **4.1σ**
my result global variable	$(\mathbf{1.47} \pm \mathbf{0.60}) \times \mathbf{10^{-4}}$ **3.9σ**

高扬$\Lambda_c^+ \to n\pi^+$例子

定义两个变量

```
RooRealVar val("mrecsignB","#font[12]{M}_{rec} (GeV/#it{c}^{2})",0.85,1.34) ;
RooRealVar val2("mrecsignB","#font[12]{M}_{rec} (GeV/#it{c}^{2})",0.85,1.34) ;
RooDataSet data("data","data",t_d,val);
```

```
TTree *t_s = t_sideband->CopyTree("mbc>2.235&&mbc<2.27&&mrecsignB>0.85&&mrecsignB<1.34");

RooDataSet sigsetS("sigsetS", "sigsetS", t_s, val2);
```

```
RooChebychev bkgnL("bkgnL", "background no LcLc p.d.f", val, RooArgList(cn0,cn1));
RooChebychev bkgnL2("bkgnL2", "background no LcLc p.d.f2", val2, RooArgList(cn0,cn1));
```

```
RooAddPdf sum0("sum0", "sum0", RooArgList(sigA,sigB,sigC,bkgPdfLcLc,bkgnL), RooArgList(nsigA,nsigB,nsigC,nbkgL,nbkgNLS));
```

```
RooAddPdf sumNL("sumNL", "sumNL", RooArgList(bkgnL2), RooArgList(nbkgNL));
```

```
RooAbsReal *ll1=sum0.createNLL(data);
RooAbsReal *ll2=sumNL.createNLL(sigsetS);
//RooAbsReal *ll3=sumcN.createNLL(toymc);
//RooAbsReal *ll3=sumcN.createNLL(*cNDistribution);

RooAddition* ll = new RooAddition("ll", "ll", RooArgSet(*ll1,*ll2));

RooMinuit m(*ll);

m.migrad();
m.hesse();
```

定义一个变量

```
RooRealVar val("mrecsignB","#font[12]{M}_{rec} (GeV/#it{c}^{2})",0.85,1.34) ;
RooDataSet data("data","data",t_d,val);
```

```
TTree *t_s = t_sideband->CopyTree("mbc>2.235&&mbc<2.27&&mrecsignB>0.85&&mrecsignB<1.34");

RooDataSet sigsetS("sigsetS", "sigsetS", t_s, val);
```

```
RooChebychev bkgnL("bkgnL", "background no LcLc p.d.f", val, RooArgList(cn0,cn1));
```

```
RooAddPdf sum0("sum0", "sum0", RooArgList(sigA,sigB,sigC,bkgPdfLcLc,bkgnL), RooArgList(nsigA,nsigB,nsigC,nbkgL,nbkgNLS));
```

```
RooAddPdf sumNL("sumNL", "sumNL", RooArgList(bkgnL), RooArgList(nbkgNL));
```

```
RooAbsReal *ll1=sum0.createNLL(data);
RooAbsReal *ll2=sumNL.createNLL(sigsetS);
//RooAbsReal *ll3=sumcN.createNLL(toymc);
//RooAbsReal *ll3=sumcN.createNLL(*cNDistribution);

RooAddition* ll = new RooAddition("ll", "ll", RooArgSet(*ll1,*ll2));

RooMinuit m(*ll);

m.migrad();
m.hesse();
```

高扬$\Lambda_c^+ \to n\pi^+$例子

定义两个变量

```
FIRST CALL TO USER FUNCTION AT NEW START POINT, WITH IFLAG=4.
COVARIANCE MATRIX CALCULATED SUCCESSFULLY
FCN=-10894.3 FROM HESSE     STATUS=OK            110 CALLS          555 TOTAL
                     EDM=5.96328e-05     STRATEGY= 1      ERROR MATRIX ACCURATE
 EXT PARAMETER                                   INTERNAL      INTERNAL
 NO.   NAME        VALUE            ERROR        STEP SIZE       VALUE
  1  cN           1.01354e+00   constant
  2  cn0          8.24808e-01   7.51161e-02   4.85778e-04   8.25746e-02
  3  cn1          1.75835e-01   7.27001e-02   4.81604e-04   1.75844e-02
  4  meanA        2.56591e-03   3.03472e-03   2.29366e-02   2.59494e-01
  5  meanB        1.00351e-03   8.32516e-04   5.99908e-03   1.00520e-01
  6  meanC        5.34747e-05   8.24902e-04   5.93087e-03   5.34750e-03
  7  nbkgL        2.67153e+02   2.72077e+01   3.80482e-03  -4.84418e-01
  8  nbkgNL       2.51517e+02   1.50555e+01   5.09054e-03   2.60521e-01
  9  nsigA        4.97805e+01   8.97765e+00   1.25326e-02  -4.38946e-03
 10  nsigB        3.76115e+02   2.20679e+01   7.76117e-03   4.66094e-01
 11  nsigC        3.42516e+02   2.23295e+01   7.25174e-03   3.04766e-01
 12  sigma        3.42511e-03   1.39043e-03   2.07653e-02  -3.20434e-01
```

定义一个变量

```
FIRST CALL TO USER FUNCTION AT NEW START POINT, WITH IFLAG=4.
COVARIANCE MATRIX CALCULATED SUCCESSFULLY
FCN=-10894.3 FROM HESSE     STATUS=OK            110 CALLS          555 TOTAL
                     EDM=5.96328e-05     STRATEGY= 1      ERROR MATRIX ACCURATE
 EXT PARAMETER                                   INTERNAL      INTERNAL
 NO.   NAME        VALUE            ERROR        STEP SIZE       VALUE
  1  cN           1.01354e+00   constant
  2  cn0          8.24808e-01   7.51161e-02   4.85778e-04   8.25746e-02
  3  cn1          1.75835e-01   7.27001e-02   4.81604e-04   1.75844e-02
  4  meanA        2.56591e-03   3.03472e-03   2.29366e-02   2.59494e-01
  5  meanB        1.00351e-03   8.32516e-04   5.99908e-03   1.00520e-01
  6  meanC        5.34747e-05   8.24902e-04   5.93087e-03   5.34750e-03
  7  nbkgL        2.67153e+02   2.72077e+01   3.80482e-03  -4.84418e-01
  8  nbkgNL       2.51517e+02   1.50555e+01   5.09054e-03   2.60521e-01
  9  nsigA        4.97805e+01   8.97765e+00   1.25326e-02  -4.38946e-03
 10  nsigB        3.76115e+02   2.20679e+01   7.76117e-03   4.66094e-01
 11  nsigC        3.42516e+02   2.23295e+01   7.25174e-03   3.04766e-01
 12  sigma        3.42511e-03   1.39043e-03   2.07653e-02  -3.20434e-01
```

结果完全一致，猜测是因为代码结构导致我的结果不同，需要进一步研究