

MultiDalitz简介

X.X Ma

July 17, 2019

1 前言

这个软件包“MultiDalitz”专为 $J/\psi \rightarrow e^+e^-K^+K^-\pi^0$ 而写。“Multi”意为多，“Dalitz”意为dalitz分析，整体的意义是多个dalitz的联合分波分析，这是由于 $\eta(1405)$ 和 $\eta(1475)$ 难分开，但是各自的动力学又迥然不同，必须同时对它们的dalitz plot进行联合拟合。这个软件包目前放在GitHub上，可以任意获取使用。

2 软件包的安装

在shell的命令行，可以用git直接获取源程序

```
1 git clone git@github.com:xxmawhu/MultiDalitz.git
```

唯一的依赖是ROOT的环境，配置好ROOT环境后，可以直接编译后使用。

```
1 cd MultiDalitz
2 make
```

编译完成后会自动在文件夹lib下生成一个.lib结尾的链接文件，名字为libKsKsK.so，在ROOT里面直接调用这个lib文件就可以使用这个包。具体的调用语法为

```
1 gSystem->Load("/path/to/lib/libKsKsK.so");
```

需要提供到libKsKsK.so的完整路径。

3 使用

3.1 准备工作

需要提供三个样本，a)data文件，b)PHSP MC文件，c)未经事例挑选的truth信息文件。需要注意的是样本b，c需要来自同一批MC样本，因为程序需要加权平均求效率。文件的格式为.txt，每三行是一个事例，存储的模板为

```
1 E1 px1 py1 pz1
2 E2 px2 py2 pz2
3 E3 px3 py3 pz3 weight
```

我们约定第一行表示第“1”个粒子的信息,记为A，依次类推。比如可以选择A、B、C分别为 K^+ 、 K^- 、 π^0 。weight为是该事例的权重，用来加权处理系统误差。约定a、b、c的文件名分别是Data.dat, PHSP.dat, Frac.dat。

3.2 声明

类的声明为

```

1 MultiDalitPdf ksksKpdf("ksksKpdf", "ksksKpdf",
2     v11, v12, v13, v14, v21, v22, v23, v24, v31, v32, v33, v34,
3     PHSPdat);

```

其中 v_{11} 等变量为四动量信息。

```

1 double high = 1.8865;
2 double low = 0-high;
3 const double pi = 3.1415926*4;
4 RooRealVar v11("v11", "v11", low, high);
5 RooRealVar v12("v12", "v12", low, high);
6 RooRealVar v13("v13", "v13", low, high);
7 RooRealVar v14("v14", "v14", low, high);
8 RooRealVar v21("v21", "v21", low, high);
9 RooRealVar v22("v22", "v22", low, high);
10 RooRealVar v23("v23", "v23", low, high);
11 RooRealVar v24("v24", "v24", low, high);
12 RooRealVar v31("v31", "v31", low, high);
13 RooRealVar v32("v32", "v32", low, high);
14 RooRealVar v33("v33", "v33", low, high);
15 RooRealVar v34("v34", "v34", low, high);

```

设置好Frac.dat

```

1 ksksKpdf.setFracDat(Fracdat);

```

添加母粒子

```

1 RooRealVar spin0("spin0", "spin0", 0);
2 spin0.setConstant();
3 RooRealVar rRes("rRes", "", 3);
4 rRes.setConstant();
5 RooRealVar meta1475("meta1475", "meta1475", 1475E-3);
6 meta1475.setConstant();
7 RooRealVar weta1475("weta1475", "weta1475", 90E-3, 10e-3, 200e-3);
8 //weta1475.setConstant();
9 ksksKpdf.configMother("eta(1475)", //the name of this resonance
10     "eta(1475)", // the title
11     LineShape::RBW, // the lineshape: you can choose from RBW, BW, GS, Flatte, Flat
12     RooArgList(spin0, rRes, meta1475, weta1475) // the configure number: spin,
13     effective radius, mass, width
14 );

```

添加共振态

```

1 const int S_WAVE(0);
2 ksksKpdf.addResonance("a0-980",
3     "eta(1405)", // this resonance from "eta(1405)"
4     LineShape::a980_0, // the lineshape
5     DecayType::AB, // this resonance decay into A,B
6     Rhoa0_980, Phia0_980, // magnitude and phase angle
7     RooArgList(spin0, rRes, ma0, gPiEtaa0, gKK2a0), //
8     configure
9     S_WAVE // the a_0 and pi0 form a S-wave
10 );

```

读取Data.dat

```

1 RooArgSet theSet1, theSet2, theSet3;
2 theSet1.add(RooArgSet(v11, v12, v13, v14, v21, v22, v23, v24));
3 theSet2.add(RooArgSet(v31, v32, v33, v34));

```

```

4   theSet3.add(RooArgSet(weight));
5   RooArgSet theSet4(theSet1, theSet2, "");
6   RooArgSet theSet(theSet4, theSet3, "");
7
8   RooDataSet *data = RooDataSet::read(Datadat, theSet);
9   data->Print();
10
11  RooDataSet *datWeight = new RooDataSet(data->GetName(), data->GetTitle(),
12      data,*data->get(), 0, weight.GetName());
13  datWeight->Print();

```

开始拟合

```

1  RooFitResult *result = ksksKpdf.fitTo(*datWeight, Save(kTRUE));

```

获取分支比，并得到拟合结果

```

1  ksksKpdf.fitFractions(cout);
2  ksksKpdf.project("project.root");

```