Tensorflow环境下开发的 FDC分波软件

王建雄，平荣刚

中国科学院高能物理研究所
pingrg@ihep.ac.cn

2023 年粒子物理实验计算软件与技术研讨会
2023年6月10-12日，
青岛，山东大学

FDC, FDC-PWA, FDC-Tensorflow

- FDC: Feynman Diagram Calculation
- FDC-PWA: FDC partial wave analysis
- FDC-Tensorflow tutorial

FDC Homepage

FDC is a package to do Feynman Diagram Calculation

The Project started in 1994 and aimed at developing a package to calculate Feynman Diagram automatically. The following parts have been finished already:

Construct the Lagrangian and deduce Feynman rules automatically
Generation of all Feynman diagrams and amplitudes for a given process.
Manipulate the amplitudes of these diagrams and generation of the expression of the total squared amplitude
Deal phase space integration automatically.

This page shows part of the results generated by FDC system.

This project is in part supported by the National Natural Science Foundation of China.

FDC-PWA homepage: FDC-PWA for Partial Wave Analysis method

FDCHQHP(FDC Heavy Quarkonium HadroProduction)

Manual

Progress in FDC project #2

Jian-Xiong Wang (Beijing, Inst. High Energy Phys.) (Jul, 2004)

Published in: *Nucl.Instrum.Meth.A* 534 (2004) 241-245 • Contribution to: ACAT 03 • e-Print: hep-ph/0407058 [hep-ph]

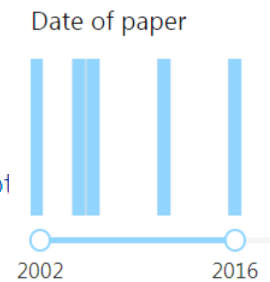
在BES/BEPC上研究重子谱的课题方案

Light hadron spectroscopy at BEPC

Bing-Song Zou (CCAST World Lab, Beijing and Beijing, Inst. High Energy Phys.) (Aug, 2000)

Published in: *Nucl.Phys.A* 692 (2001) 362-371 • Contribution to: Biennial Conference on Low-Energy Antiprot 2000) • e-Print: hep-ph/0011174 [hep-ph]

pdf DOI cite claim reference search 5 citations

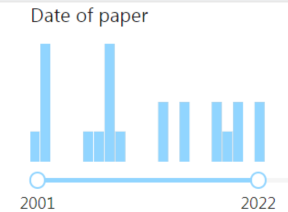


N*, Lambda*, Sigma* and Xi* resonances from J / Psi and Psi-prime decays

Bing-Song Zou (CCAST World Lab, Beijing and Beijing, Inst. High Energy Phys.) (Mar, 2000)

Published in: *Nucl.Phys.A* 684 (2001) 330-332 • Contribution to: FB16 • e-Print: hep-ph/0006039 [hep-ph]

pdf DOI cite claim reference search 23 citations

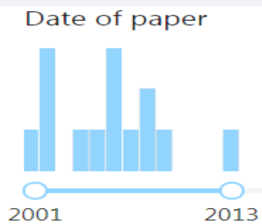


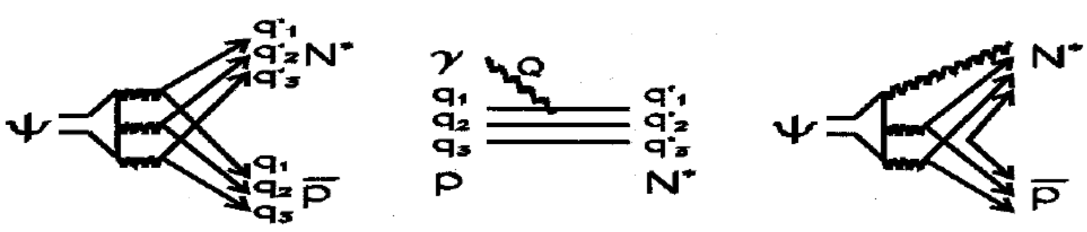
The baryon spectroscopy from J / psi decays

Bing-Song Zou (CCAST World Lab, Beijing and Beijing, Inst. High Energy Phys.) (2000)

Published in: *Nucl.Phys.A* 675 (2000) 167C-172C • Contribution to: Hadron 1999

DOI cite claim reference search 14 citations





(a) (b) (c)

FDC-PWA软件的研发起点

Theoretical formalism and Monte Carlo study of partial wave analysis for J / psi --> p anti-p omega

#1

W.H. Liang (CCAST World Lab, Beijing and Beijing, Inst. High Energy Phys. and Guangxi Normal U. and Natl. Lab. Heavy Ion Accel., Lanzhou), P.N. Shen (CCAST World Lab, Beijing and Beijing, Inst. High Energy Phys. and Guangxi Normal U. and Natl. Lab. Heavy Ion Accel., Lanzhou), J.X. Wang (CCAST World Lab, Beijing and Beijing, Inst. High Energy Phys. and Guangxi Normal U. and Natl. Lab. Heavy Ion Accel., Lanzhou), B.S. Zou (CCAST World Lab, Beijing and Beijing, Inst. High Energy Phys. and Guangxi Normal U. and Natl. Lab. Heavy Ion Accel., Lanzhou) (2002)

Published in: *J.Phys.G* 28 (2002) 333-343

2.3. *The effective vertices involved*

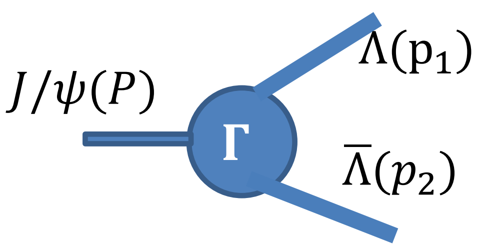
$$\mathcal{L} = \bar{\psi}_1 \Gamma \psi_2 A$$

Table 2. The transformation properties of some operators.

$\Gamma =$	i	γ_5	γ_μ	$\gamma_\mu\gamma_5$	$\sigma_{\mu\nu}$	$\sigma_{\mu\nu}\gamma_5$	$g_{\mu\nu}$
$\gamma_0\Gamma^+\gamma_0 =$	$-i$	$-\gamma_5$	γ_μ	$\gamma_\mu\gamma_5$	$\sigma_{\mu\nu}$	$-\sigma_{\mu\nu}\gamma_5$	$g_{\mu\nu}$
$C(\gamma_0\Gamma^+\gamma_0)^T C^{-1} =$	$-i$	$-\gamma_5$	$-\gamma_\mu$	$\gamma_\mu\gamma_5$	$-\sigma_{\mu\nu}$	$\sigma_{\mu\nu}\gamma_5$	$g_{\mu\nu}$
$\gamma_0\Gamma^P\gamma_0 =$	i	$-\gamma_5$	γ_μ	$-\gamma_\mu\gamma_5$	$\sigma_{\mu\nu}$	$-\sigma_{\mu\nu}\gamma_5$	$g_{\mu\nu}$

FDC分波软件中的协变张量振幅

- Tensor form of vertex generation by phenomenological Lagrangian (strong intera.)
  - conserve P, C parity, isospin, strangeness, charm, baryon and lepton numbers
  - an example of $J/\psi \to \Lambda\left(\frac{1}{2}^{+}\right)\bar{\Lambda}\left(\frac{1}{2}^{-}\right)$



Effective Lagrangian:

$$\mathcal{L} = \bar{u}(p_1)\, \Gamma\, v(p_2)\, \epsilon_\mu(P)$$

P, C, CPT symmetry transformation: $\mathcal{L}^P = \mathcal{L},\ \mathcal{L}^C = \mathcal{L}\ ,\ \mathcal{L} = \mathcal{L}^\dagger$

FDC分波软件的研发进展

□ 子曰：“工欲善其事，必先利其器”

✓ 费曼图自动计算(FDC)系统的GPU实现：FDC + Tensorflow

✓ FDC-TF 的物理分析

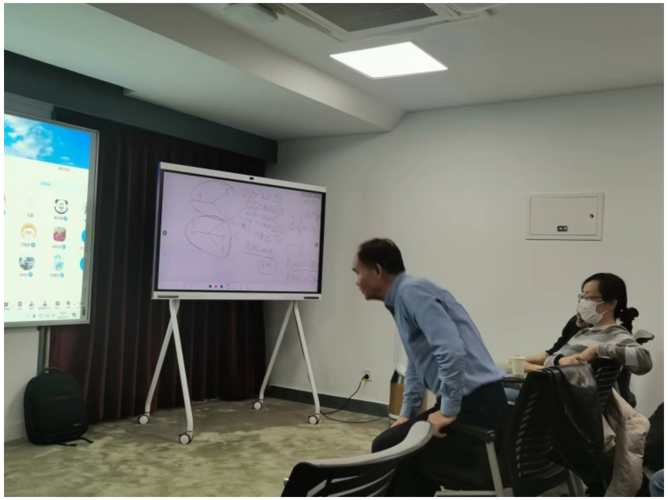
1. PWA $J/\psi \to \pi^+\pi^-\Sigma^+\bar{\Sigma}^-$
2. PWA $J/\psi \to p\bar{p}K^+K^-$
3. PWA $J/\psi \to \phi\eta\eta$
4. PWA $J/\psi \to \phi\eta\eta'$
5. PWA $\psi' \to \pi^0\Sigma^+\bar{\Sigma}^-$

……

✓ FDC版本升级和两大功能扩展

1. 超子的弱衰变
2. 辐射衰变

分波分析理论-实验联合讨论会，
2022-3-9,中国高等科学技术中心



FDC分波软件特征(武装到了牙齿！)

- ✓ 工作环境：
  Reduce Free PSL version, 9-Aug-2018，底层软件Rlisp
- ✓ 用户建立环境：
  - ➢ model_cp elsewhere_dir/model model
  - ➢ process_cp elsewhere_dir/process process
  - ➢ model/ process/
- ✓ 粒子态的描述:用户编辑文件model/add_vertices
- ✓ gmodel: 生成所有可能衰变的顶点，结果写入 model.tex
- ✓ lamodel: 生成顶点结构的model.ps文件
- ✓ diag： 生成所有可能的费曼数图
- ✓ amp: 产生各个费曼图对应的Fortran文件
- ✓ kine： 写出运动学文件和拟合文件

FDC分波软件包的拟合策略

- 用f77编译Fortran文件包，生成可执行文件
- 基于CPU单线程计算，算力有限
- MLLH 极小化程序包： Fumili （Fortran版本）
- Normalization factor calculated in the reduce amplitude (save more time)

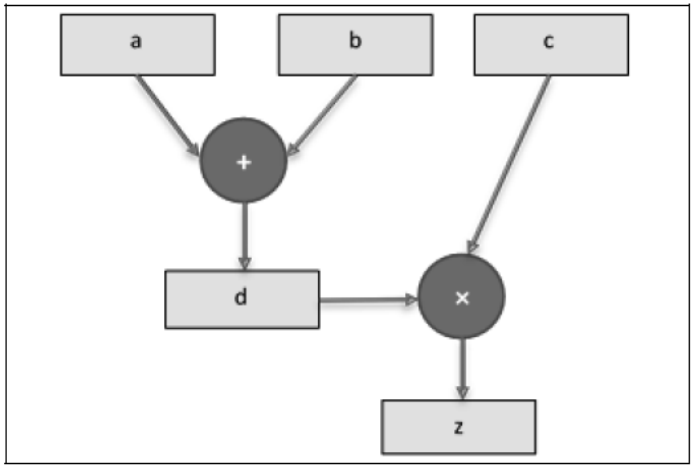
$$|\mathcal{M}|^2 = \sum_{j=1}^{n_{par}} \sum_{i=1}^{N_{mc}} c_j A_j = c_j a_j, \text{with } a_j = \sum_{i=1}^{N_{mc}} A_j$$

- But loop over the data events event by event

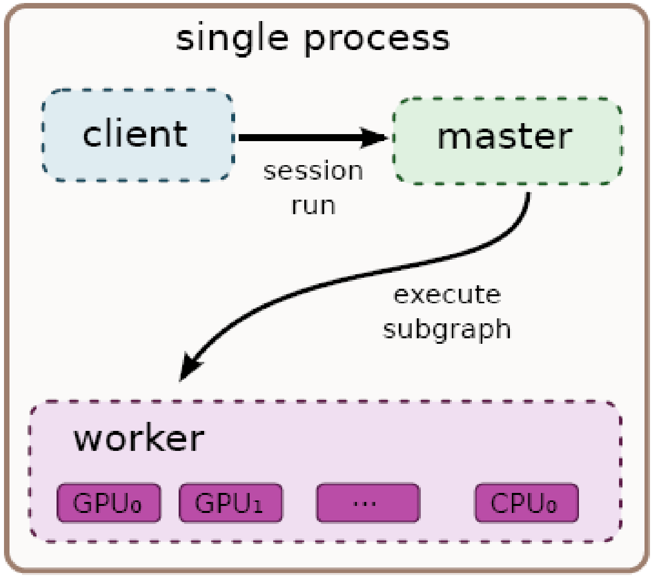
  (most cpu time consuming)
- Mass and width not enter into the fit parameter list
- Access the fit projection (dplot.hbook) and resonance ratio (mplot.info)

为何选择python环境下的Tensorflow?

- From Fortran to python environment.
- Tensorflow (tf) popular in AI community
- After Tensorflow2.x, it has eager execution mode, make it suitable for scientific computation
- Tensor data model applicable to amplitude calculation
- Autograph functionality



$$z = c \times (a + b)$$

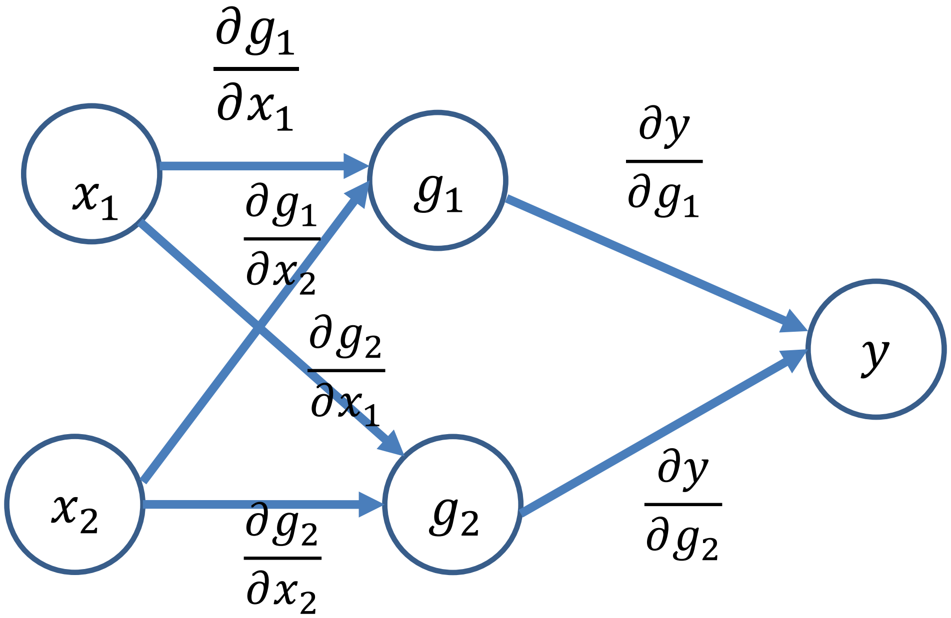


为何选择python环境下的Tensorflow?

- High efficiency to calculate Hessian matrix

For example: $y(x_1, x_2) = f(g_1, g_2) = \ln(g_1) + e^{g_2}$, with
$g_1(x_1, x_2) = x_1 + x_2$, $g_2(x_1, x_2) = x_1 x_2$

$$\begin{bmatrix} \frac{\partial y}{\partial x_1} & \frac{\partial y}{\partial x_2} \end{bmatrix} = \begin{bmatrix} \frac{\partial y}{\partial g_1} & \frac{\partial y}{\partial g_2} \end{bmatrix} \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} \end{bmatrix}$$



用GPU-Tensorflow加速FDC分波计算

- Compile amplitudes of Fortran codes into a python modules
- Calculate the event amplitude in Tensorflow framework
- MLLH minimized with Minuit in pyROOT (python version of Minuit)
- Access fit results (signal yields and statistical uncertainty calculated based on resultant covariance matrix)
- Allow user to add mass and width as hyper-parameters in the fit
- Allow for simultaneous fit to multiple samples

FDC振幅的张量算法

- $|\mathcal{M}(\text{event}_\text{v})|^2 = \bar{\Sigma}_{s_1,\ldots,s_j}\left|\Sigma_k c_k\, a_{\text{v},k}\right|^2$

  $= C_{k,l}\, A_{\text{v},k,l}$ (dumb index rule)

  with $C_{k,l} = c_k\, c_l^*,\ \ A_{\text{v},k,l} = \bar{\Sigma}_{s_1,\ldots,s_j}(a_{\text{v},k} a_{\text{v},l}^*)$

  c_k : k-th parameter,

  $a_{\text{v},k}$: k-th term of amplitude for event v
- $A_{\text{v},k,l}$ calculated by FDC, and stored in memory (limitation form GPU memory)
- Amplitude reduction in Tensorflow

Structure of FDC-TF package

FDC: generate amplitude

FDC python module

TensorFlow:
MLLH calculation

Minuit in pyROOT:
minimize MLLH

Signal yields, statistical uncertainty.

Matplotlib or pyROOT:
Fill histograms

Dependent modules:

- Numpy, matlibplot, iminuit, Scipy, Tensorflow2.0, pyROOT
- Python version: 3.7

例：用户脚本

```
from SMLfit import SMLfit
import tensorflow as tf
import numpy
import os
import warnings
import timeit as tt
import FDC
#here invoke FDC equivalent FDC in Amps

EvtPDL = Script()
EvtPDL.readPdtTable('reson.inp') # pdt.table
EvtPDL.readParaList('fpara.inp') # para.list
EvtPDL.setFinalState(['p','pbar','K^+','K^-'])
EvtPDL.setEvtFileMC(['p4.mc'])
EvtPDL.setEvtFileDT(['p4.dt'])
EvtPDL.setEvtFileBG(['p4.bg'])
EvtPDL.setAddWidth([['5','Gx0',0.005,0.004,0.08]])
#### SMLfit ####
with tf.device("/device:gpu:0"):
    myfit = SMLfit(EvtPDL)
    print('try scan 1 para....')
    myfit.scan(1)
    myfit.exec('migrad')
    myfit.writeParaList('myfit.list',myfit.exec('np_values'),myfit.exec('np_errors'))
    myfit.write_totMCamps('totAmps.npz',0)
    if myfit.exec('valid'): numpy.save('mycov',myfit.exec('np_covariance'))
    print('FVAL= ',myfit.exec('fval'))
    print('values ',myfit.exec('np_values'))
    print('errors ',myfit.exec('np_errors'))
    myfit.writeEvtMass('p4.mc','mij.mc')
    myfit.writeEvtMass('p4.dt','mij.dt')
    myfit.writeEvtMass('p4.bg','mij.bg')
    myfit.writeModeEvtAmps('mmEvt.npz')
    myfit.calSta(0)
```

分波软件Tensorflow应用开发

- **Signal yields and statistical errors**

For mode i: $N_i \pm \delta N_i$,

where $N_i = r_i \left(N_{obs} - N_{bkg}\right)$ with $r_i = \frac{\sigma_i}{\sigma_{tot}}$

$$\delta N_i = \sum_{m=1}^{N_{par}} \sum_{n=1}^{N_{par}} \left(\frac{\partial N_i \partial N_i}{\partial X_m \partial X_n}\right) V_{mn}$$

$\boldsymbol{V_{mn}}$: **Covariant matrix calculated by MINUIT. If failed, then calculated by Hessian matrix determined by tf.GradientTape.**

- **Mass resolution for narrow resonance**

$$|BW(x)|^2 = |BW(x')|^2 \otimes R(x', x)$$

$\boldsymbol{R(x'x)}$: **parametrized with 3 Breit-Wigner function, determined with zero-width resonance. Multi-Gaussian function parametrization is under developed.**

分波软件Tensorflow应用开发（续）

- **Simultaneous fit to multiple data sets**

  Object function for data set i: $S_i = -(ln\mathcal{L}^i_{dt} - ln\mathcal{L}^i_{bkg})$
  Minimized object function: $S = \sum_i S_i$
  where S_i calculated by one GPU card, dispatched by CPU muti-threads

- **One channel decay with running width**

$$BW(s, M_0, \Gamma_0) = \frac{1}{s - M_0^2 - iM_0\Gamma(s)} \quad \text{with} \quad \Gamma(m) = \Gamma_0 \left(\frac{q}{q_0}\right)^{2l+1} \frac{m_0}{m} B_l'^2(q, q_0, d).$$

$$B_0'(q, q_0, d) = 1,$$

$$B_1'(q, q_0, d) = \sqrt{\frac{1 + (q_0 d)^2}{1 + (qd)^2}},$$

$$B_2'(q, q_0, d) = \sqrt{\frac{9 + 3(q_0 d)^2 + (q_0 d)^4}{9 + 3(qd)^2 + (qd)^4}},$$

$$B_3'(q, q_0, d) = \sqrt{\frac{225 + 45(q_0 d)^2 + 6(q_0 d)^4 + (q_0 d)^6}{225 + 45(qd)^2 + 6(qd)^4 + (qd)^6}},$$

$$B_3'(q, q_0, d) = \sqrt{\frac{11035 + 1575(q_0 d)^2 + 135(q_0 d)^4 + 10(q_0 d)^6 + (q_0 d)^8}{11035 + 1575(qd)^2 + 135(qd)^4 + 10(qd)^6 + (qd)^8}},$$

- **Baryon resonance : couple channel running width**

For example: $N(1535)$:

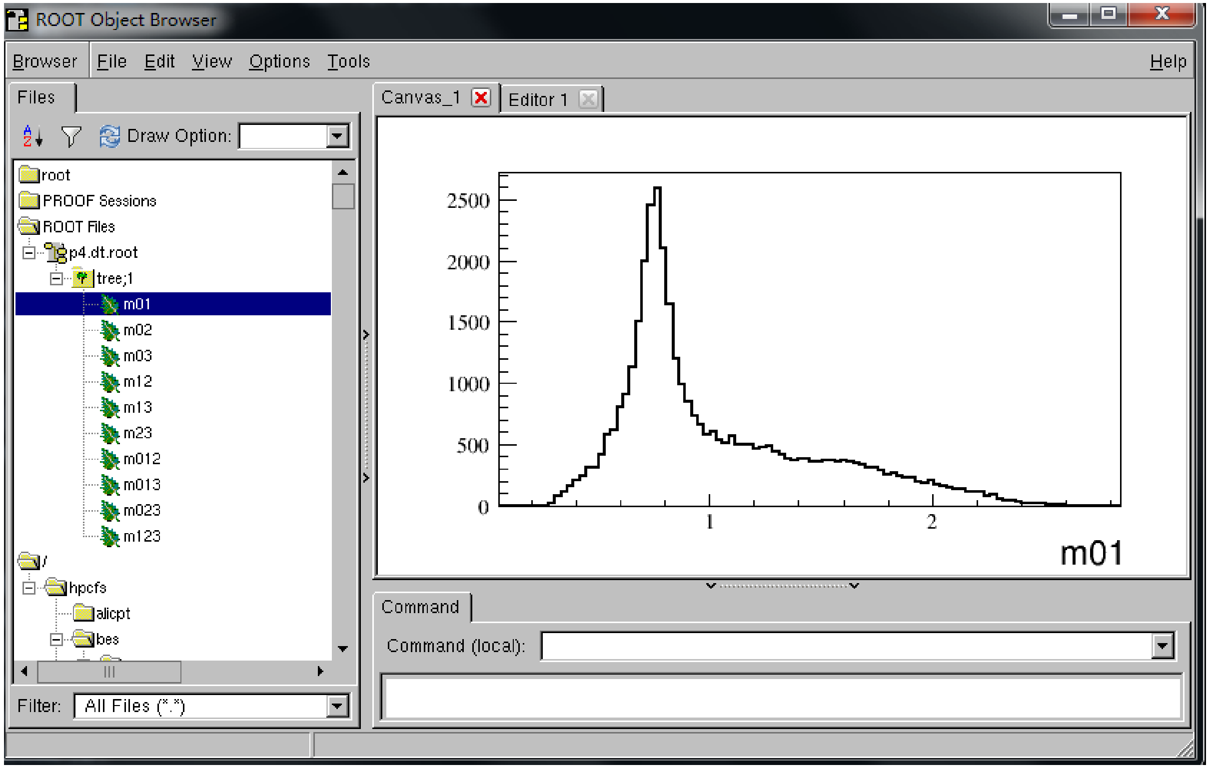
$$BW(s, M_0, \Gamma_0) = \frac{1}{s - M_0^2 - iM_0\Gamma(s)}, \text{ with } \Gamma(s) = \Gamma_0\left(0.5\ \frac{\rho_{\pi N}(s)}{\rho_{\pi N}(M_0^2)} + \frac{\rho_{\eta N}(s)}{\rho_{\eta N}(M_0^2)}\right)$$

关键词	N1440width	N1520width	N1535width	N1650width	N1700width
共振态	$N(1440)\ \frac{1}{2}^+$	$N(1520)\ \frac{3}{2}^-$	$N(1535)\ \frac{1}{2}^-$	$N(1650)\ \frac{1}{2}^-$	$N(1700)\ \frac{3}{2}^-$
关键词	N1710width	N1720width	L1380width	L1405width	L1520width
共振态	$N(1710)\ \frac{1}{2}^+$	$N(1720)\ \frac{3}{2}^+$	$\Lambda(1380)\ \frac{1}{2}^-$	$\Lambda(1405)\ \frac{1}{2}^-$	$\Lambda(1520)\ \frac{3}{2}^-$
关键词	L1600width	L1670width	D1232width	D1600width	D1620width
共振态	$\Lambda(1600)\ \frac{1}{2}^+$	$\Lambda(1670)\ \frac{1}{2}^-$	$\Delta(1232)\ \frac{3}{2}^+$	$\Delta(1600)\ \frac{3}{2}^+$	$\Delta(1620)\ \frac{1}{2}^-$
关键词	D1700width	S1385width	S1660width	S1670width	S1750width
共振态	$\Delta(1700)\ \frac{3}{2}^-$	$\Sigma(1385)\ \frac{3}{2}^+$	$\Sigma(1660)\ \frac{1}{2}^+$	$\Sigma(1670)\ \frac{3}{2}^-$	$\Sigma(1750)\ \frac{1}{2}^-$
关键词	S1910width	X1530width			
共振态	$\Sigma(1910)\ \frac{3}{2}^-$	$\Xi(1530)\ \frac{3}{2}^+$			

分波软件Tensorflow应用开发（续）

- npz to root format
  - In python environment, results are saved in the npz format file. Figures are filled with matplotlib.pyplot.
  - But most users like to use ROOT to fill histograms
  - transform npz file to root file

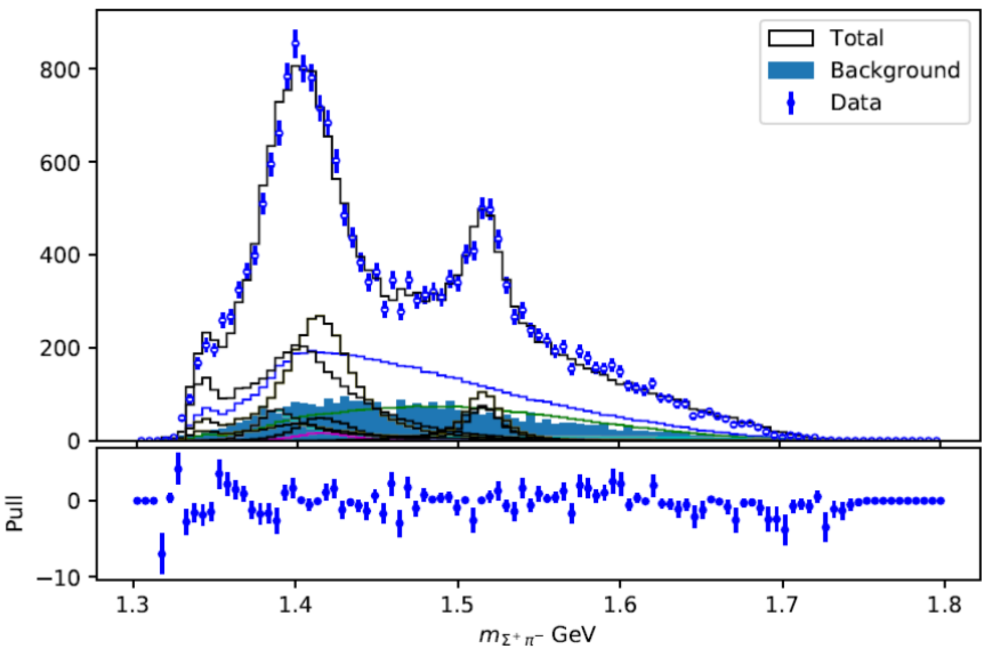
    npz2root:

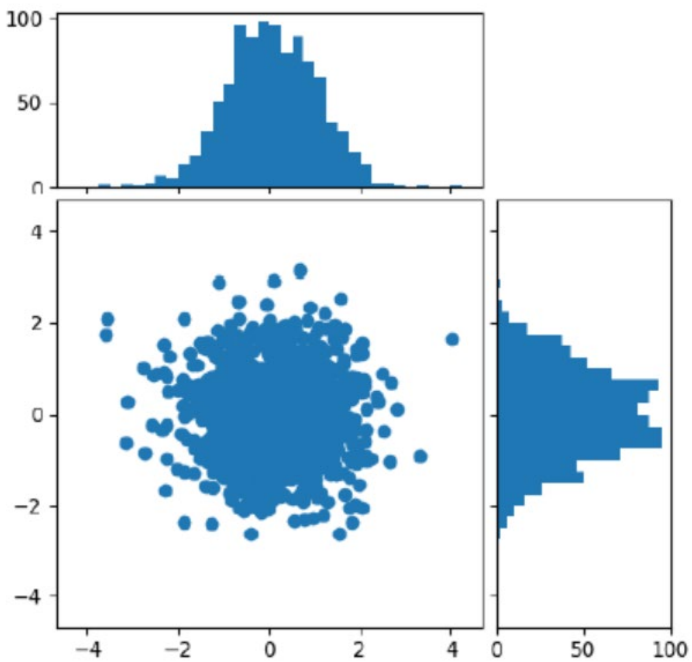


分波软件Tensorflow应用开发（续）

- Histogram: SMLfit.hist, SMLfit.phist

Using matplotlib.pyplot





分波软件Tensorflow应用开发（续）

- **FDC interface to event generator BesEvtGen**
  - **create an external FDC package**
  - **in BesEvtGen, FDC model is invoked with syntax:**

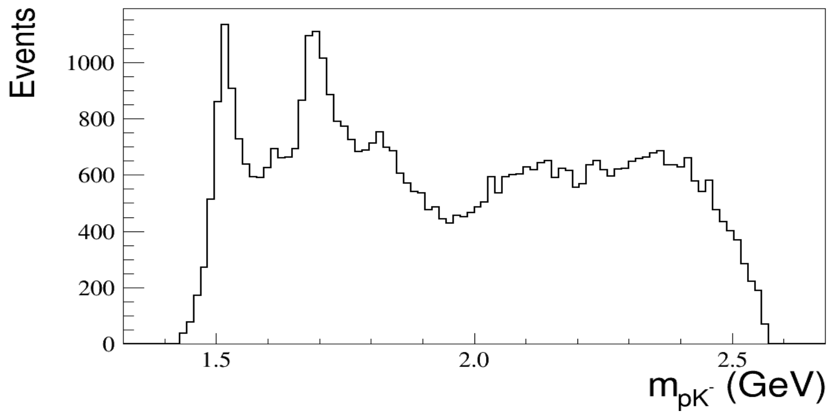
```
Decay parentParticle
Br  daughter particle lists  FDC modelIndex;
Enddecay
```

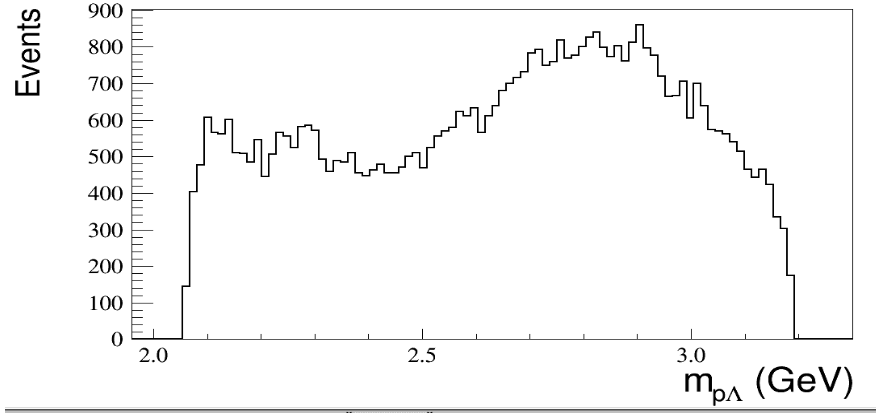
  - **If modelIndex is neglect, all modes are generated including interference effects.**

For instance: $J/\psi \to pK^-\Lambda$

```
Decay J/psi
1  p+ K- anti-Lambda0 FDC;
Enddecay
Decay Lambda0
1 p+ pi- PHSP;
Enddecay
End
```

分波软件Tensorflow应用开发（续）

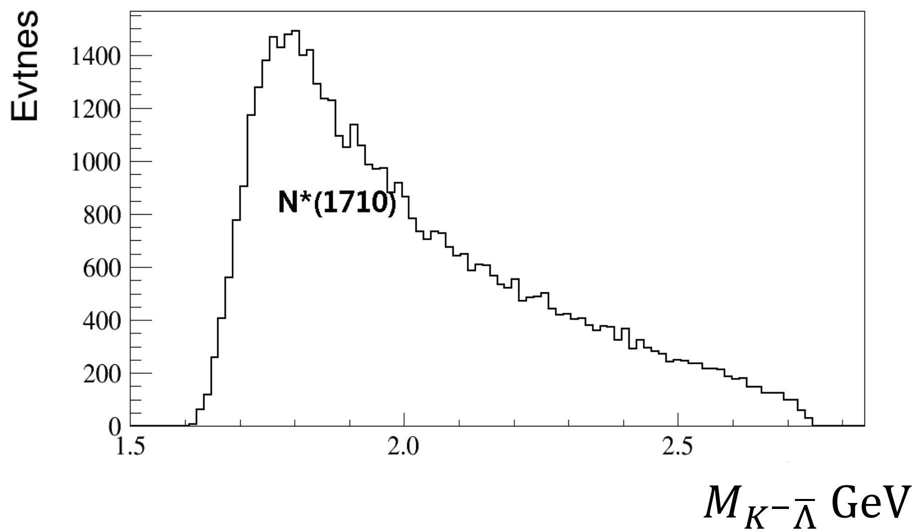




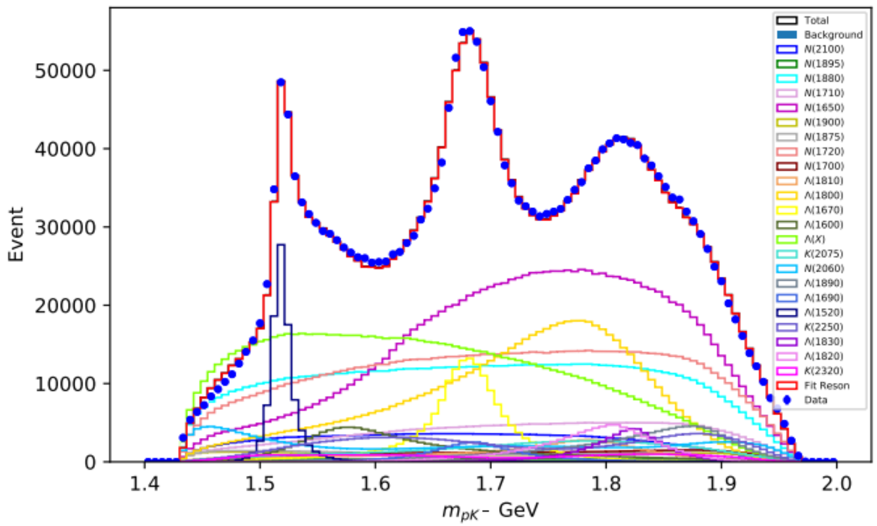
- **Event generation for a specified mode**

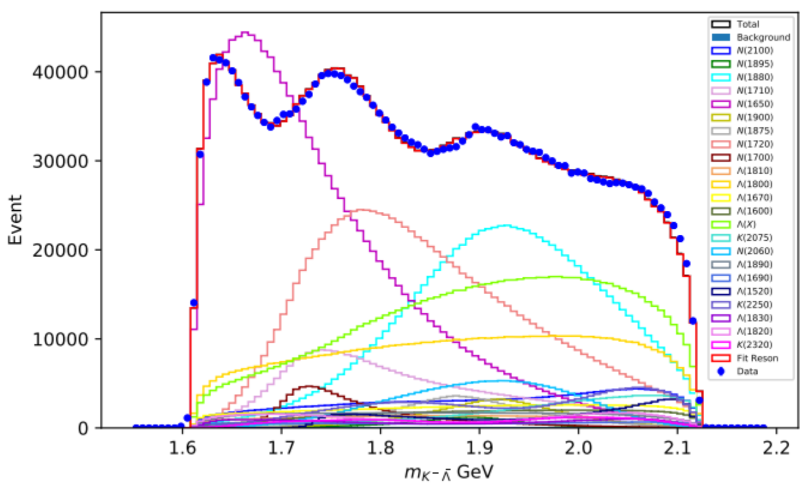
  For example, $J/\psi \to \bar{N}^*(1710)^- p$, $\bar{N}^*(1710)^- \to K^-\bar{\Lambda}$

```
Decay J/psi
1  p+ K- anti-Lambda0 FDC 1;
Enddecay
Decay Lambda0
1 p+ pi- PHSP;
Enddecay
End
```



大统计量的应对策略





$J/\psi \to pK^-\bar{\Lambda} + c.c.$

Using 10 billion J/ψ events:

Data: 2,537,989 ± 1593

MC: 20,000,000

Parameters: 156

Resonances: 18

- ✓ 支持多GPU卡的同时拟合
- ✓ 采用binned 似然函数

总结和展望

- 用 GPU加速计算的FDC-TF程序包已经用于BESIII的多个分析，将来会扩展到其他实验组。
- 在Tensorflow环境下开发了许多用于分波分析的应用程序，基本满足用户需求.
- FDC-PWA生成代码改进的展望:
  - 多级弱衰变,
  - χ_{cJ} 衰变
  - 辐射衰变

谢谢大家!

backup

Baryon production in 10 billion J/ψ and 2.7 billion $\psi(2S)$

X	$\mathrm{Br}(J/\psi \to X) \times 10^2$	$N(J/\psi \to X) \times 10^5$	$\mathrm{Br}(\psi' \to X) \times 10^3$	$N(\psi' \to X) \times 10^4$
$N\bar{N}\pi$	97.0 ± 6.0	970	76.0 ± 6.0	209
$p\bar{p}\pi^+\pi^-$	60.0 ± 5.0	600	60.0 ± 4.0	165
$N\bar{N}\eta$	41.8 ± 3.6	418	5.8 ± 1.3	16
$\Lambda\bar{\Lambda}\eta$	1.6 ± 0.2	16	2.5 ± 0.4	7
$pK^-\bar{\Lambda} + c.c.$	8.6 ± 1.1	86	10.0 ± 0.4	28
$pK^-\bar{\Sigma}^0$	2.9 ± 0.8	29	1.7 ± 0.2	5
$\Sigma\bar{\Lambda}\pi$	8.3 ± 0.7	83	15.4 ± 0.4	42