Model-Independent Partial Wave Analysis using a Massively-parallel Fitting Framework

Rafael Aoude[1], Alberto C. dos Reis[1], Michael Sokoloff[2], **Liang Sun**[2,3]

1. Centro Brasileiro de Pesquisas Físicas
2. University of Cincinnati
3. Wuhan University

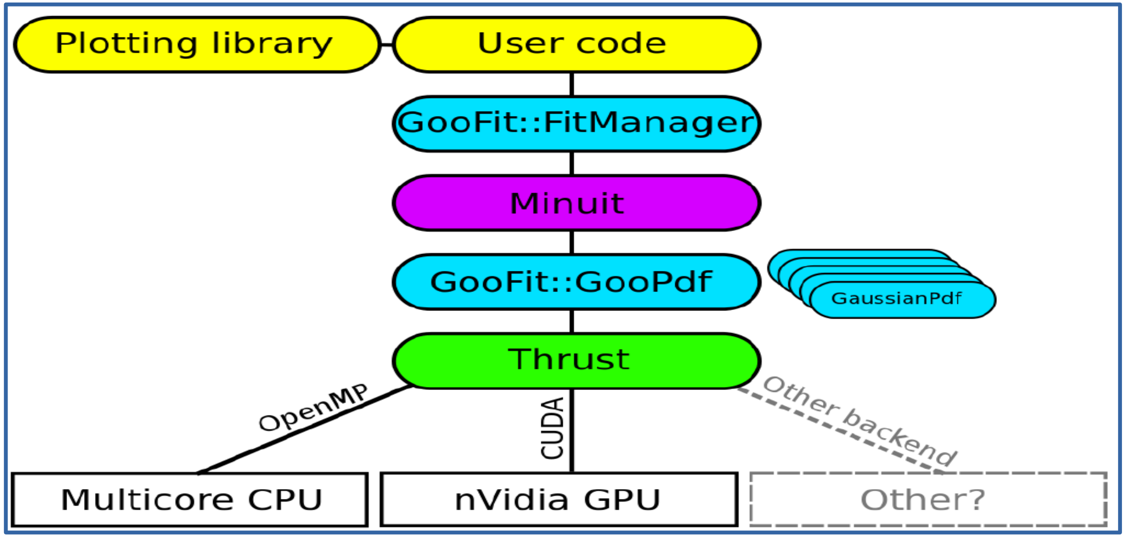
Outline

- Introduction on GooFit
- GPU based MIPWA method
- Performance
- Summary

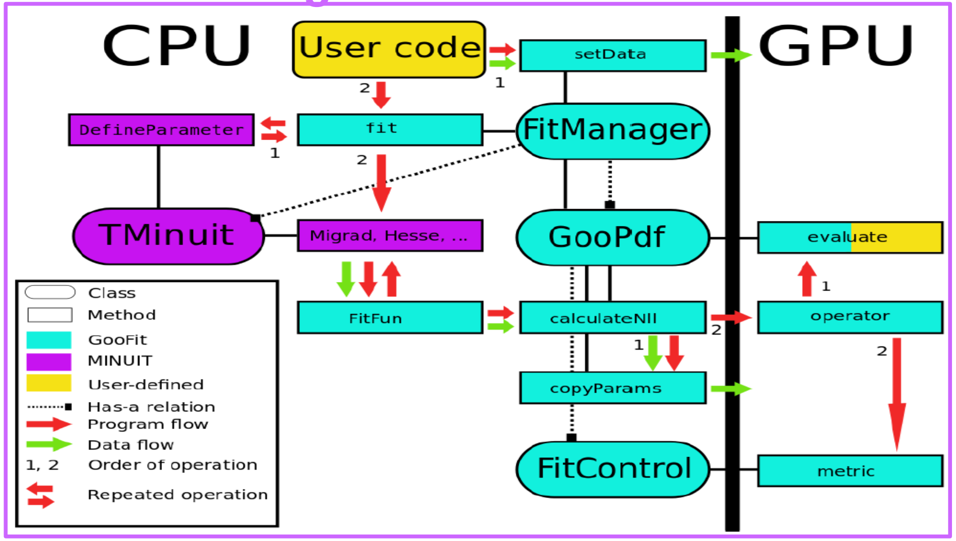
GooFit introduction

- GooFit: an open-source project originally developed by R. Andreassen and funded by NSF
  - **FitManager** object as the interface between MINUIT and a GPU which allows a PDF (**GooPDF** object) to be evaluated in parallel

Architecture:



Program flow:



GooFit: A library for massively parallelising maximum-likelihood fits
R. Andreassen et al., IEEE Access 2 (2014)

Analogy with RooFit

- Code structure similar to RooFit framework, the overall fit set-up and running are familiar for RooFit users

RooFit	GooFit
RooRealVar	Variable
RooDataSet	UnbinnedDataSet
RooDataHist	BinnedDataSet
RooAbsPdf	GooPdf
RooGaussian	GaussianPdf
RooArgSet	`vector<Variable*>`
RooPlot	None! Use ROOT plotting.
`myPdf->plotOn(foo)`	`myPdf->getCompProbsAtDataPoints(points)`
RooAbsTestStatistic	FitControl

GooFit PDFs

- Simple PDFs: ARGUS, correlated Gaussian, Crystal Ball, exponential, Gaussian, Johnson SU, polynomial, relativistic Breit-Wigner, scaled Gaussian, smoothed histogram, step function, Voigtian
- Composites:
  - Sum, $f_1 A(\vec{x}) + (1 - f_1) B(\vec{x})$.
  - Product, $A(\vec{x}) \times B(\vec{x})$.
  - Composition, $A(B(x))$ (only one dimension).
  - Convolution, $\int_{t_1}^{t_2} A(x - t) * B(t) \mathrm{d}t$.
  - Map,

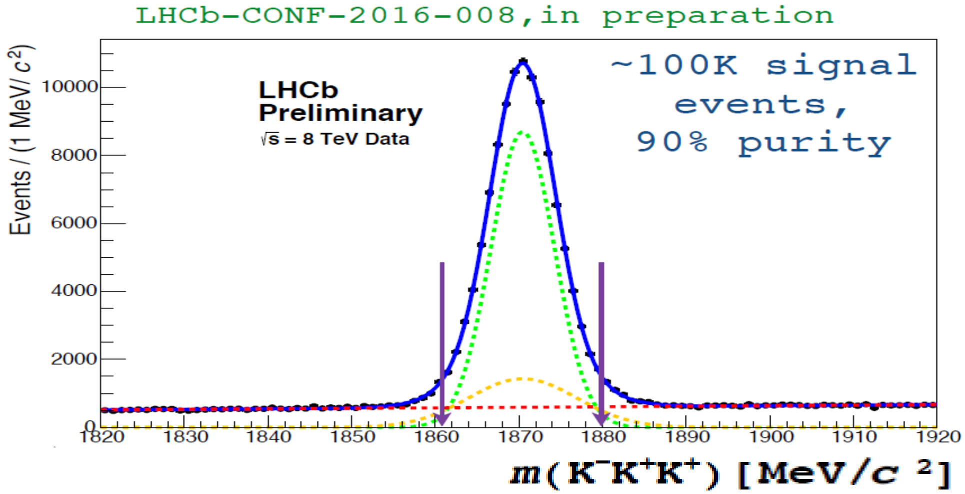
$$F(x) = \begin{cases} A(x) & \text{if } x \in [x_0, x_1) \\ B(x) & \text{if } x \in [x_1, x_2) \\ \dots & \\ Z(x) & \text{if } x \in [x_{N-1}, x_N] \end{cases}$$

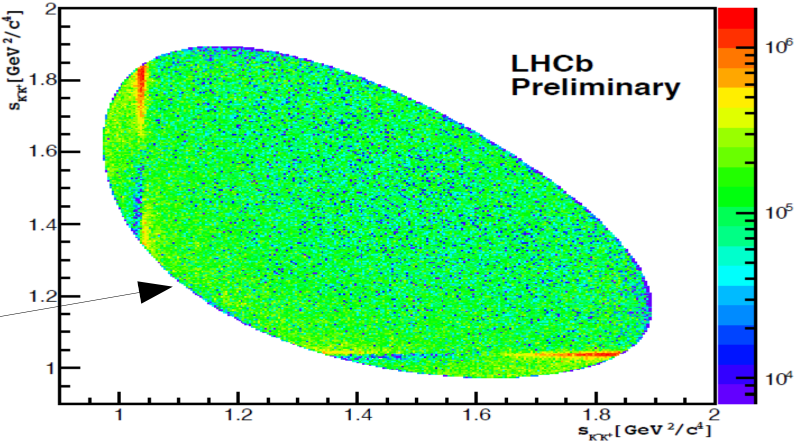
You can write your own PDFs based on the example PDF code with relative ease

- Specialized PDFs for amplitude analyses
  - **TddpPdf** (**DalitzPlotPdf**) as the main engine for time-dependent (-integrated) Dalitz plot fits, with a list of **ResonancePdf** objects as input to describe different (non-)resonance amplitudes

Amplitude analysis of $D^+ \rightarrow K^+K^+K^-$

- The goal: to identify the resonance structure & measure KK amplitude S-wave
- First amplitude analysis in this channel
- Using full 2012 data of LHCb, ~100K signal events after final selections with ~90% purity
- Dalitz-plot analysis using Isobar model & MIPWA with unbinned ML fit
  - Isobar model results presented during CHARM2016 workshop
  - MI-PWA with GooFit still under development





S-wave accounts for ~90% of the decay rate, phi makes up the rest

Model Independent Partial Wave Analysis

- Isobar Model is good for narrow resonances but it fails to describe the overlap of broad resonances
- An alternative is to describe scalar components in a model-independent approach
- This is done by dividing $s_{K\text{-}K\text{+}}$ in bins. The S-wave amplitude for bin j is given by:

$$\mathcal{A}_0(s_j) = c_j e^{i\phi_j}$$

- The total amplitude is the sum of all bins' contributions and spin-1 resonances

$$\mathcal{A} = \mathcal{A}_0(s_{12}, s_{23}) + \sum_{spin=1,2} a_i \; e^{i\delta_i} \times \mathcal{A}_i(s_{12}, s_{23})$$

- In a generic point of KK spectrum, the S-wave amplitude is obtained by a cubic spline interpolation

Cubic spline interpolation in MIPWA

Function: $y = f(x)$, $y_i = f(x_i)$, $\{x_i\}$ as bin limits, function parameters are to be determined in the spline fit

Basic algorithm*:

$$y = Ay_j + By_{j+1} + Cy''_j + Dy''_{j+1}$$

$$A \equiv \frac{x_{j+1} - x}{x_{j+1} - x_j} \qquad B \equiv 1 - A = \frac{x - x_j}{x_{j+1} - x_j}$$

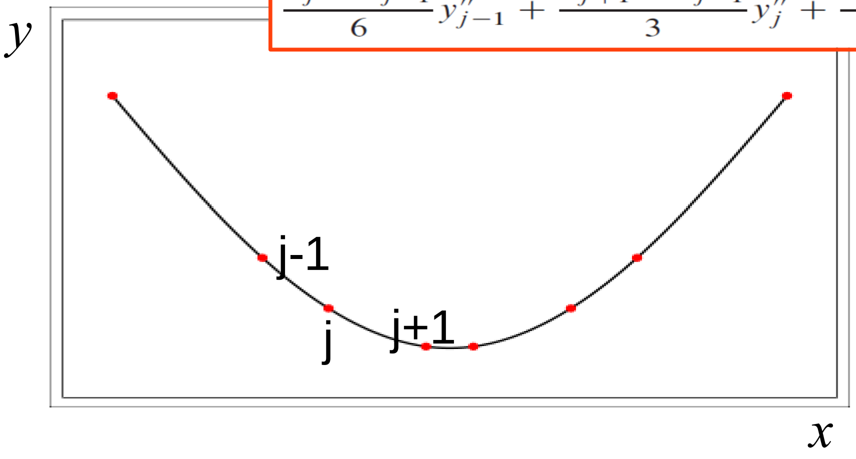
$$C \equiv \frac{1}{6}(A^3 - A)(x_{j+1} - x_j)^2 \qquad D \equiv \frac{1}{6}(B^3 - B)(x_{j+1} - x_j)^2$$

* From *Numerical Recipes 3rd ed.*

2nd derivatives are calculated by solving equations:

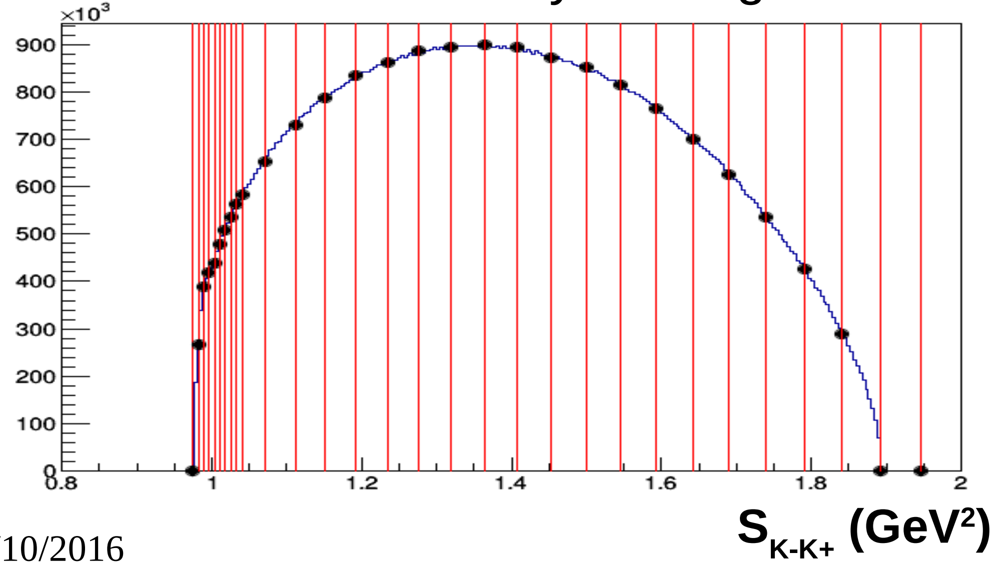
$$\frac{x_j - x_{j-1}}{6} y''_{j-1} + \frac{x_{j+1} - x_{j-1}}{3} y''_j + \frac{x_{j+1} - x_j}{6} y''_{j+1} = \frac{y_{j+1} - y_j}{x_{j+1} - x_j} - \frac{y_j - y_{j-1}}{x_j - x_{j-1}}$$

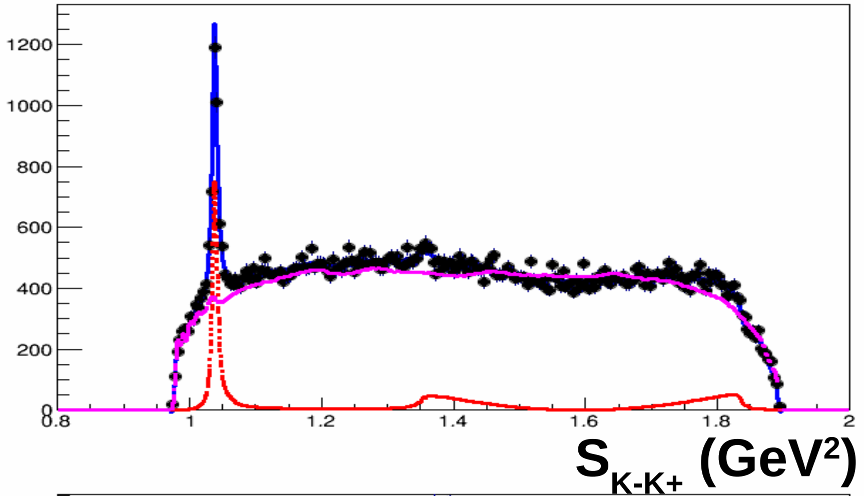
"Control points" are chosen as anchors for the spline fit

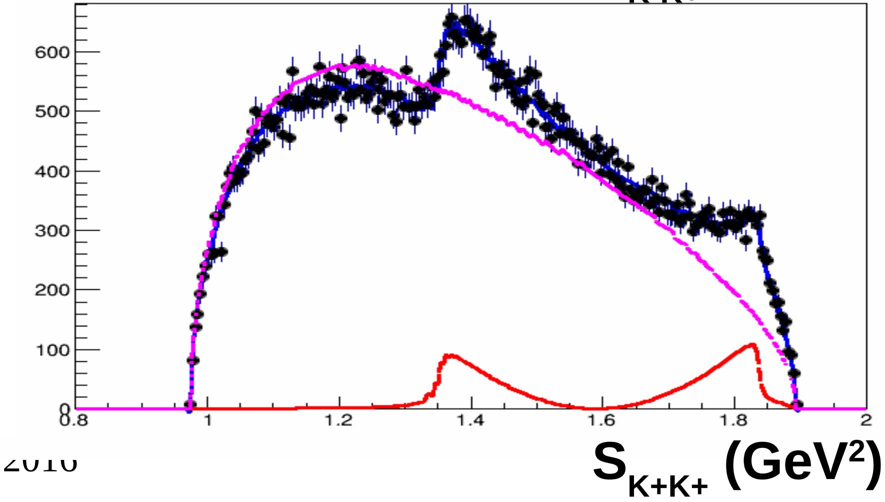


MIPWA fit on $D^+ \rightarrow K^+K^+K^-$ toy MC

- Tested on a sample of 100k toy events: phi resonance + s-wave
  - 30 control points
    - 60 S-wave shape parameters with arbitrary starting values







GPU Performance of PWA fit

With CUDA v7

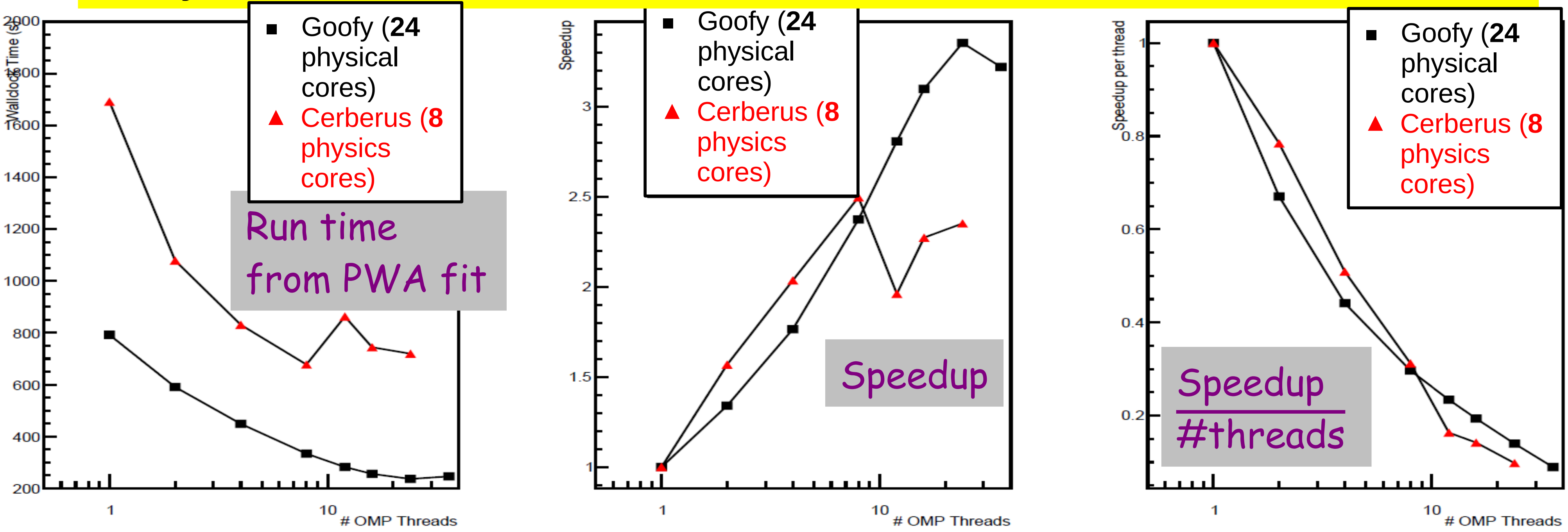
Platform	Model	Chip	CUDA cores	Run time (sec.)
Workstation	Tesla K40c	GK110BGL	2880	**76**
Desktop PC	GeForce GTX 980	2nd gen. Maxwell (GM204)	2048	**67**
Laptop ASUS N56V	GeForce GT 650M	GK107	384	**179**

- The speed-up wrt the original CPU version ("Rio") is estimated to be **>1000** on K40c
- OpenMP version of GooFit uses **792 s** for one thread, **236 s** for **24** threads with two Intel Xeon E5-2680 v3 CPUs
- A desktop-level GPU is capable of providing cost-efficient computing power

GooFit with openMP: PWA fit

- Besides GPU, GooFit also supports multiple CPUs by using openMP
- The speedup performance for different numbers of OMP threads used is not ideal – needing further investigation

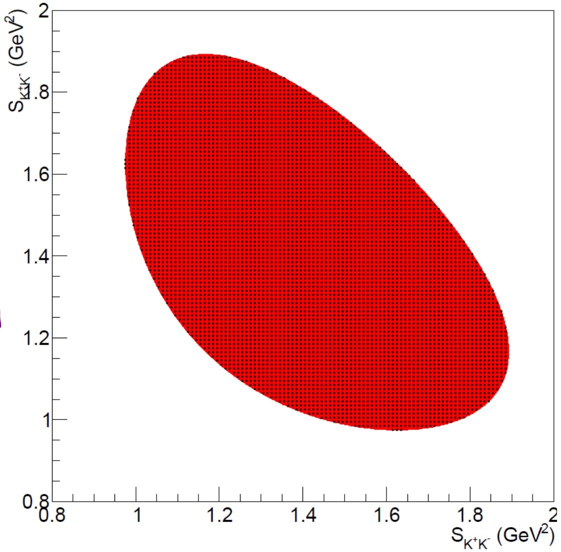
Goofy: Intel Xeon CPU E5-2680 v3 x 2 **Cerberus: Intel Xeon CPU E5520 x 2**

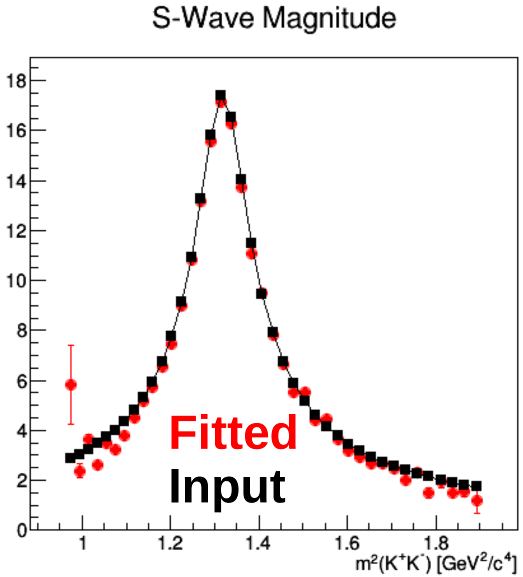


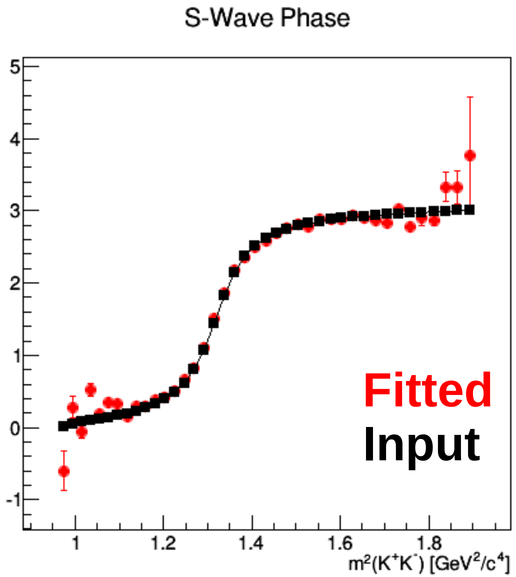
Fit stability test

- In each fit iteration, we normalize the total PDF to unity. The normalization is done by summing up PDF values of DP grid points
- We fit to a toy sample, and compare the **fitted** S-wave shape parameters to the **input values**

An example of DP grid points with spacing of 0.01 GeV^2 in either x or y axis



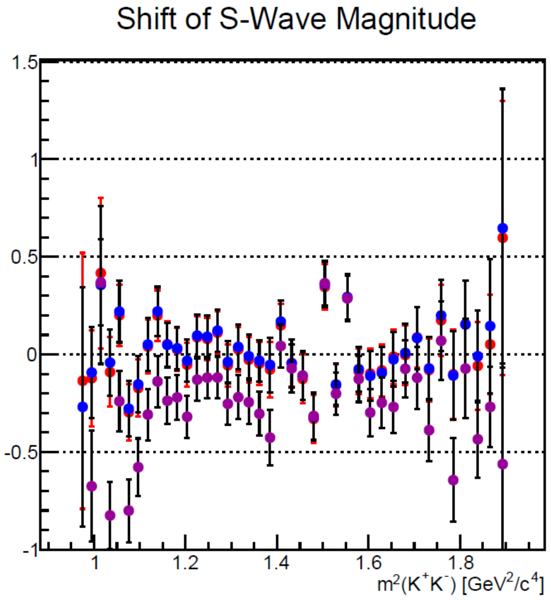


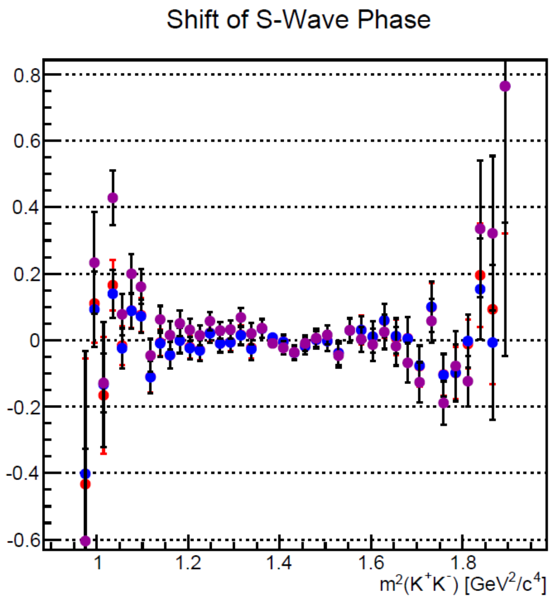


Fit stability test

- We test on different choices of grid spacing, and find that finer grid spacing (better normalization precision) leads to improved agreement between fitted and input values

Different grid spacing in x or y axis:
0.01 GeV²
0.004 GeV²
0.001 GeV²





Summary

- GooFit has now extended its support on MIPWA fits with vastly increased speed
- Code is accessible online:
  - https://github.com/GooFit/GooFit (main)
  - https://github.com/liang-sun/GooFit (personal branch, with PWA support, more frequent updates, WIP)
- Please feel free to contact us for any technical support!
  - Liang Sun (lsun@cern.ch)
  - Antonio Augusto Alves Jr (aalvesju@cern.ch)