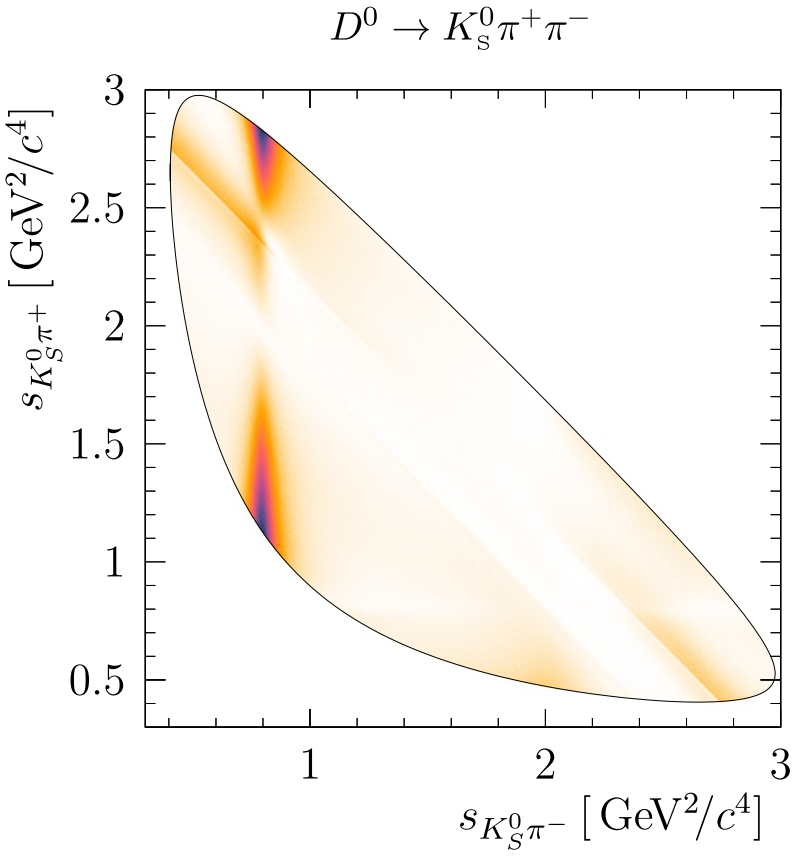
An Introduction to Amplitude Fitting in LHCb

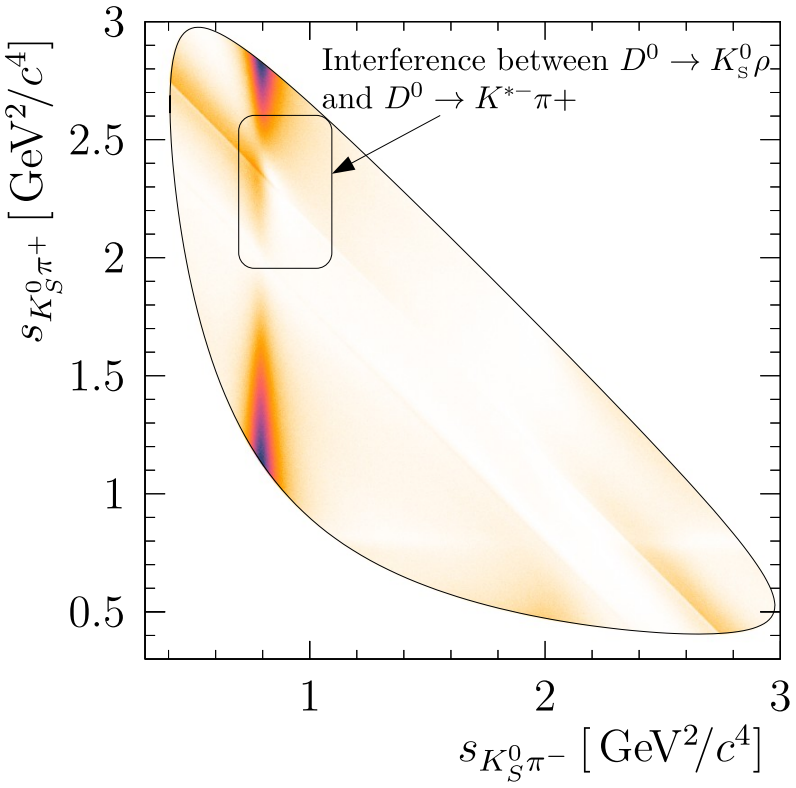
T.Evans, A. Mathad, M. Mikhasenko, N. Skidmore, Z. Xu

Amplitude analysis: In a nutshell



- Aim is to describe dynamics of multibody processes:
  - Measure CPV and mixing
  - Study intermediate hadronic systems (exotic + conventional)
  - As inputs to precision measurements
  - ...
- A multidimensional problem:
  $\implies$ For a spinless N-body decay: $d = 3N - 7$
- For a three body decay $d = 2$ - the famous 'Dalitz plot'
- Can have fairly enormous data sets (millions of signal candidates)

Amplitude analysis: In a nutshell



Hadronic systems are strongly coupled quantum mechanical entities
$\Longrightarrow$ Interference and phase behaviour are key

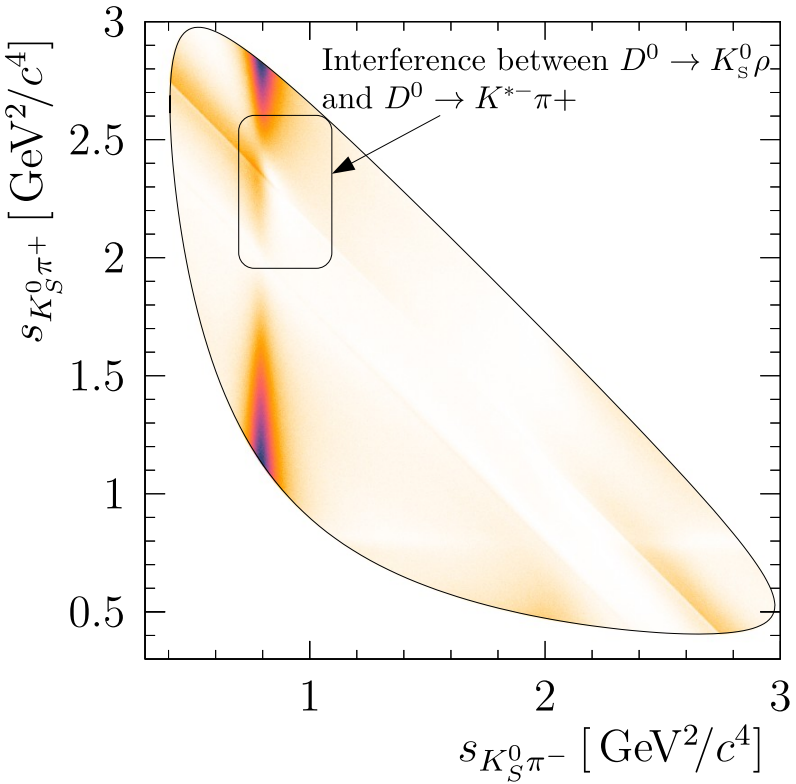
- The observed distribution of signal events for some process $X \to Y_1 Y_2$..

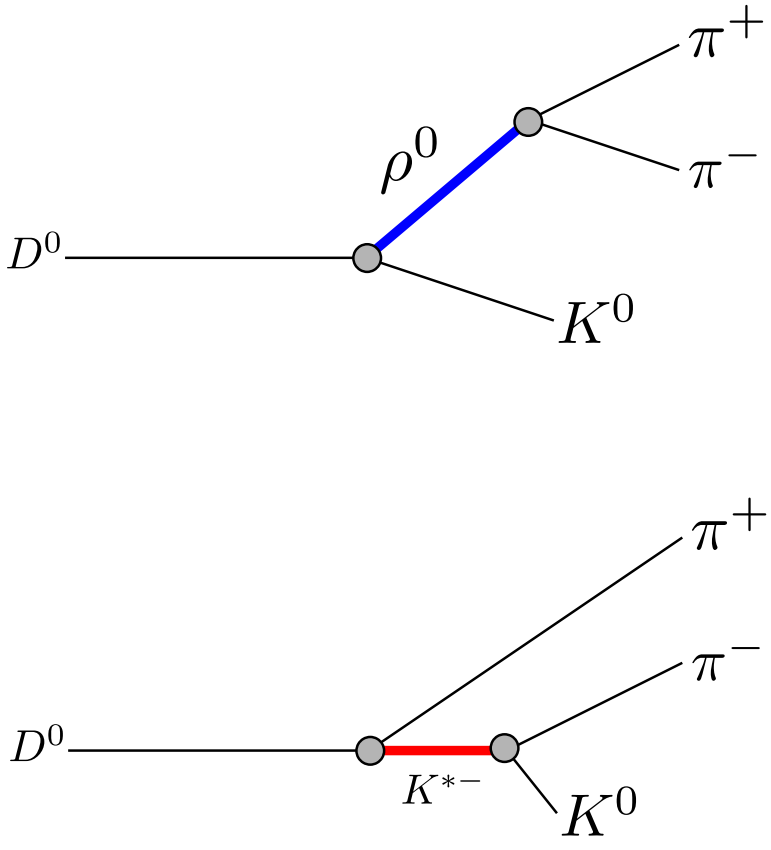
$$\mathcal{P}(\psi) d\psi \propto \rho_N(\psi) \left| \mathcal{M}_{X \to Y_1 Y_2 \ldots} \right|^2 d\psi$$

- Often work within the 'isobar' approximation:

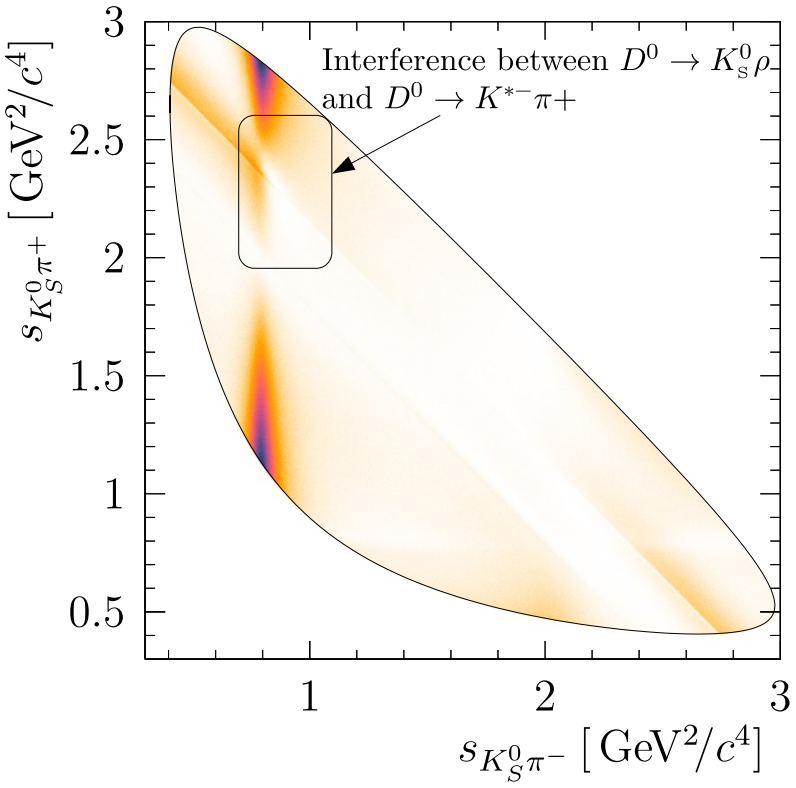
$$\mathcal{M}_{X \to Y_1 Y_2 \ldots} = \sum g_{X \to R(Y_1 Y_2) \ldots} \mathcal{T}_{X \to R(Y_1 Y_2) \ldots}$$

Amplitude analysis: In a nutshell

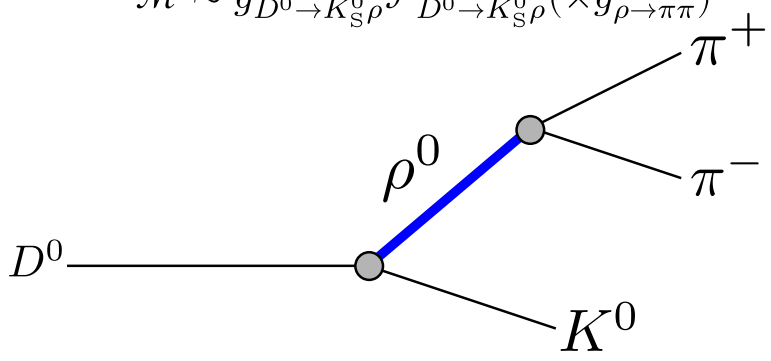




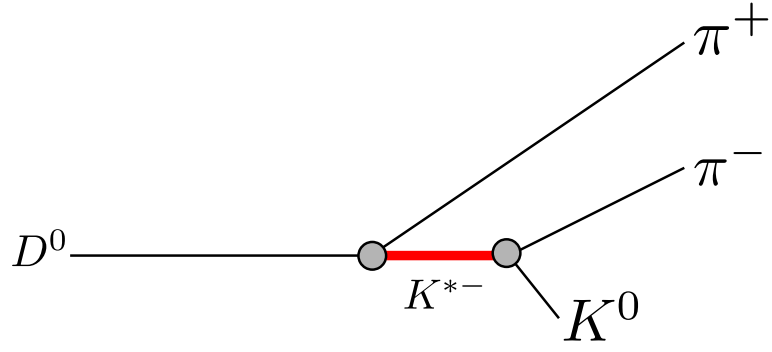
Amplitude analysis: In a nutshell



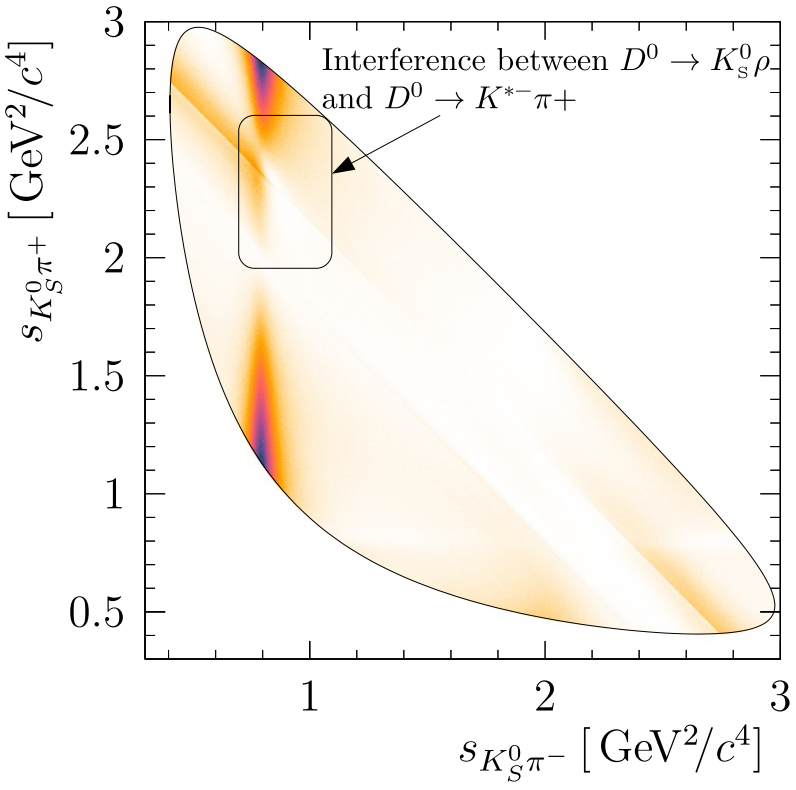
$\mathcal{M} \sim g_{D^0 \to K^0_S \rho} \mathcal{T}_{D^0 \to K^0_S \rho} (\times g_{\rho \to \pi\pi})$



$\mathcal{M} \sim g_{D^0 \to K^{*-}\pi^+} \mathcal{T}_{D^0 \to K^{*-}\pi^+} (\times g_{K^{*-} \to K^0_S \pi^-})$



Amplitude analysis: In a nutshell



Hadronic systems are strongly coupled quantum mechanical entities
$\Longrightarrow$ Interference and phase behaviour are key

- The observed distribution of signal events for some process $X \to Y_1 Y_2$..

$$\mathcal{P}(\psi)d\psi \propto \rho_N(\psi)\left|\mathcal{M}_{X\to Y_1Y_2\ldots}\right|^2 d\psi$$

- Often work within the 'isobar' approximation:

$$\mathcal{M}_{X\to Y_1Y_2\ldots} = \sum g_{X\to R(Y_1Y_2)\ldots}\mathcal{T}_{X\to R(Y_1Y_2)\ldots}$$

- (Complex) decay constants $g_{X\to R(Y_1Y_2)\ldots}$ are often the primary output of an amplitude analysis
  - Can also measure properties of intermediate states (masses, widths, J^P, ...)
  - 'Decay constants' in most AmAn not really meaningful $\Longrightarrow$ should only be considered a parameter of a model

A first look at the Likelihood Function

Multidim fits often doing by maximising likelihood (minimising -2LL of some observed data set $\{\psi_i\}$:

$$\log \mathcal{L} = \sum_i \log \mathcal{P}(\psi_i)$$

A first look at the Likelihood Function

Multidim fits often doing by maximising likelihood (minimising −2LL of some observed data set $\{\psi_i\}$:

$$\log \mathcal{L} = \sum_i \log \left(f_{sig} \mathcal{P}_{sig}(\psi_i) + (1 - f_{sig}) \mathcal{P}_{bkg}(\psi_i) \right)$$

- But there is also background

A first look at the Likelihood Function

Multidim fits often doing by maximising likelihood (minimising −2LL of some observed data set $\{\psi_i\}$:

$$\log \mathcal{L} = \sum_i \log \left(\varepsilon(\psi_i)(f_{sig}\mathcal{P}_{sig}(\psi_i) + (1 - f_{sig})\mathcal{P}_{bkg}(\psi_i) \right)$$

- But there is also background
- And the probability to reconstruct decays generally varies over $\{\psi\}$

* this view is biased: can also deal with these effects using event-weighting.

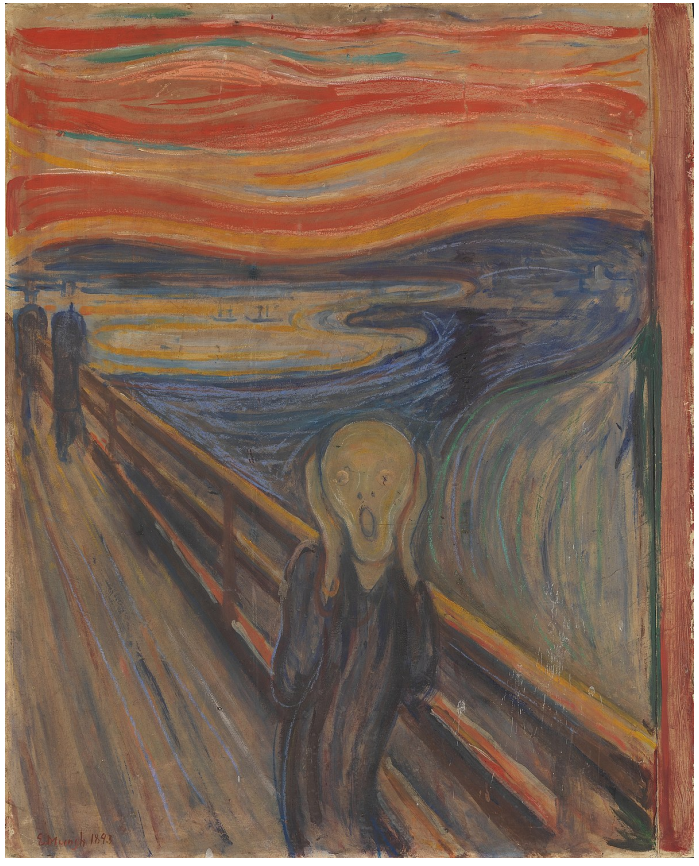
A first look at the Likelihood Function

Multidim fits often doing by maximising likelihood (minimising -2LL of some observed data set $\{\psi_i\}$:

$$\log \mathcal{L} = \sum_i \log \left(\varepsilon(\psi_i) \left(\frac{f_{sig}}{N_{sig}} \mathcal{P}_{sig}(\psi_i) + \frac{(1 - f_{sig})}{N_{bkg}} \mathcal{P}_{bkg}(\psi_i) \right) \right)$$

- But there is also background
- And the probability to reconstruct decays generally varies over $\{\psi\}$
- We also need to normalise our PDFs to use in a likelihood function

A first look at the Likelihood Function



Multidim fits often doing by maximising likelihood (minimising −2LL of some observed data set $\{\psi_i\}$:

$$\log \mathcal{L} = \sum_i \log \left(\int d\psi' \, \mathcal{G}(\psi', \psi_i) \varepsilon(\psi') \left(\frac{f_{sig}}{N_{sig}} \mathcal{P}_{sig}(\psi') + \frac{(1 - f_{sig})}{N_{bkg}} \mathcal{P}_{bkg}(\psi') \right) \right)$$

- But there is also background
- And the probability to reconstruct decays generally varies over $\{\psi\}$
- We also need to normalise our PDFs to use in a likelihood function.
- What we measure is also convolved with the detector resolution.

It's not as bad as it looks

$$\log \mathcal{L} = \sum_i \log \left(\int d\psi' \mathcal{S}(\psi', \psi_i) \varepsilon(\psi') \left(\frac{f_{sig}}{N_{sig}} \mathcal{P}_{sig}(\psi') + \frac{(1 - f_{sig})}{N_{bkg}} \mathcal{P}_{bkg}(\psi') \right) \right)$$

$$\log \mathcal{L} = \sum_i \log \left(\frac{f_{sig}}{N_{sig}} \mathcal{P}_{sig}(\psi_i) + \frac{(1 - f_{sig})}{N_{bkg}} \mathcal{P}_{bkg}(\psi_i) \right) + \log \varepsilon(\psi_i)$$

$$\log \mathcal{L} = \sum_i \log \left(\frac{f_{sig}}{N_{sig}} \mathcal{P}_{sig}(\psi_i) + \frac{(1 - f_{sig})}{N_{bkg}} \mathcal{P}_{bkg}(\psi_i) \right) + \log \varepsilon(\psi_i)$$

It's not as bad as it looks

$$\log \mathcal{L} = \sum_i \log \left(\int d\psi' \mathcal{S}(\psi', \psi_i) \varepsilon(\psi') \left(\frac{f_{sig}}{N_{sig}} \mathcal{P}_{sig}(\psi') + \frac{(1 - f_{sig})}{N_{bkg}} \mathcal{P}_{bkg}(\psi') \right) \right)$$

$$\log \mathcal{L} = \sum_i \log \left(\frac{f_{sig}}{N_{sig}} \mathcal{P}_{sig}(\psi_i) + \frac{(1 - f_{sig})}{N_{bkg}} \mathcal{P}_{bkg}(\psi_i) \right) + \log \varepsilon(\psi_i)$$

$$\log \mathcal{L} = \sum_i \log \left(\frac{f_{sig}}{N_{sig}} \mathcal{P}_{sig}(\psi_i) + \frac{(1 - f_{sig})}{N_{bkg}} \mathcal{P}_{bkg}(\psi_i) \right) + \log \varepsilon(\psi_i)$$

These are PDFs that we want to use in likelihood fitting: needs normalisation

$$N = \int d\psi \varepsilon(\psi)\mathcal{P}(\psi) = \int d\psi \varepsilon(\psi)\rho_N(\psi)\left|\sum_i g_i \mathcal{T}_i(\psi)\right|^2$$

Everyone (mostly) uses the same trick* :

$$\int d\psi \rho_N(\psi)\left|\sum_i g_i \mathcal{T}_i(\psi)\right|^2 = \sum_{ij} g_i g_j^* \int d\psi \rho_N(\psi)\mathcal{T}_i(\psi)\mathcal{T}_j(\psi)^*$$

Integral(s) are independent of $g_{\{i\}}$: can evaluate once (slow) and reuse each iteration of the fit.

* slightly more involved for spinful initial / final states but idea is the same

For problems with $d > 2$, common to evaluate using MC integration:

$$\mathcal{J} = \int d\psi \rho_N(\psi) f(\psi) \sim \frac{1}{N} \sum_{j=0}^{N} f(\psi_j) \rho_N(\psi_j)$$

Suppose we sample the points with distribution $g(\psi)$ w.r.t. phase space rather than uniformly:

$$\mathcal{J} \sim \frac{1}{N} \sum_{j=0}^{N} \frac{f(\psi_j)}{g(\psi_j)}$$

For problems with $d > 2$, common to evaluate using MC integration:

$$\mathcal{I} = \int d\psi \rho_N(\psi) f(\psi) \sim \frac{1}{N} \sum_{j=0}^{N} f(\psi_j) \rho_N(\psi_j)$$

Suppose we sample the points with distribution $g(\psi)$ w.r.t. phase space rather than uniformly:

$$\mathcal{I} \sim \frac{1}{N} \sum_{j=0}^{N} \frac{f(\psi_j)}{g(\psi_j)}$$

Suppose we want to integrate $\varepsilon(\psi) f(\psi)$ and have a simulated sample distributed as $\varepsilon(\psi) g(\psi)$ ..

$$\mathcal{I} = \int d\psi \rho_N(\psi) f(\psi) \varepsilon(\psi) \sim \frac{1}{N} \sum_{j=0}^{N} \frac{f(\psi_j)}{g(\psi_j)}$$

A Quick Summary

Amplitude analysis might look a bit scary

- Multidimensional fits
- Lots of data
- Complex PDFs
- Complex likelihoods + experimental effects

But lots of tricks have been learnt to make it more managable

- Aggressive caching
- Wise normalisation strategies
- ML tools for dealing with background

Lots of amplitude frameworks written to tackle these problems

- AmpGen
- cFit
- Craft
- GooFit
- Hydra
- Ipanema
- Laura++
- RooFit(+CUDA)
- Mint2
- TensorFlowAnalysis (TFA)
- zFit

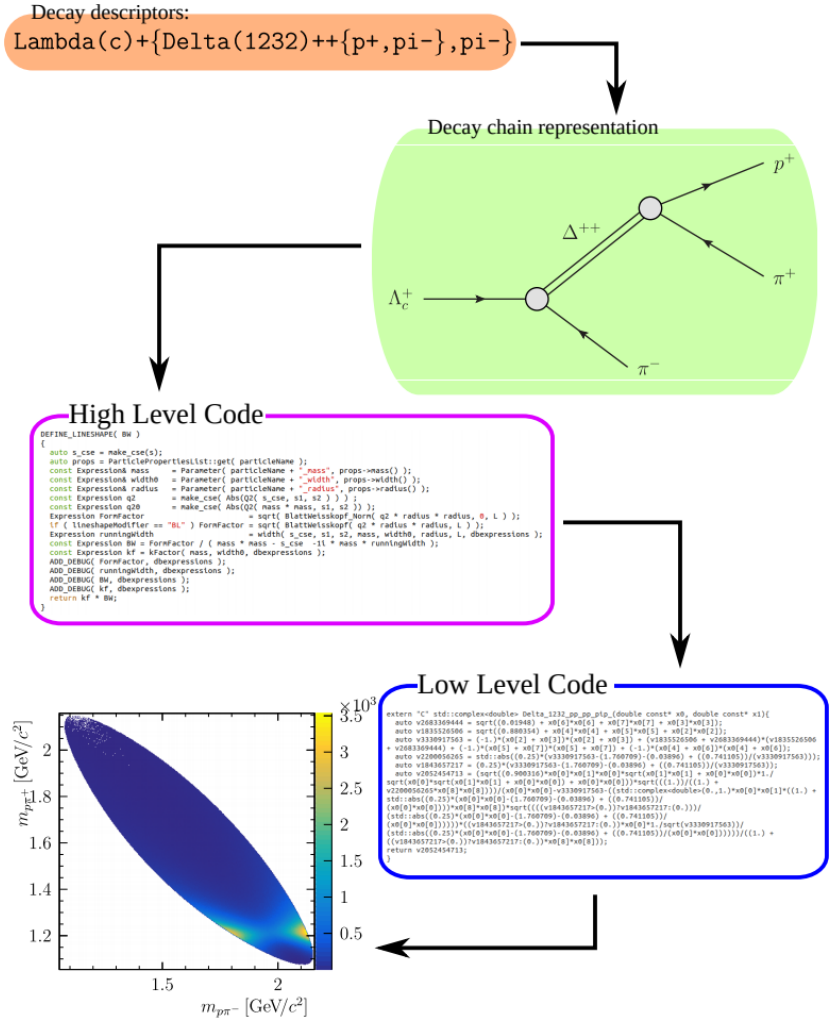
+ Standalone code

Amplitude Fitting Tools

Lots of amplitude frameworks written to tackle these problems

- AmpGen
- cFit
- Craft
- GooFit
- Hydra
- Ipanema
- Laura++
- RooFit(+CUDA)
- Mint2
- TensorFlowAnalysis (TFA)
- zFit

+ Standalone code

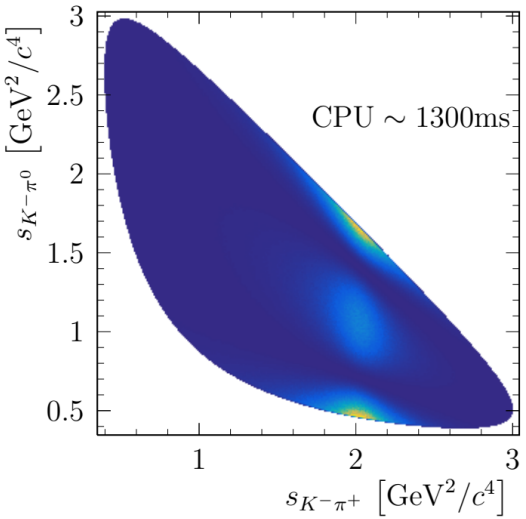


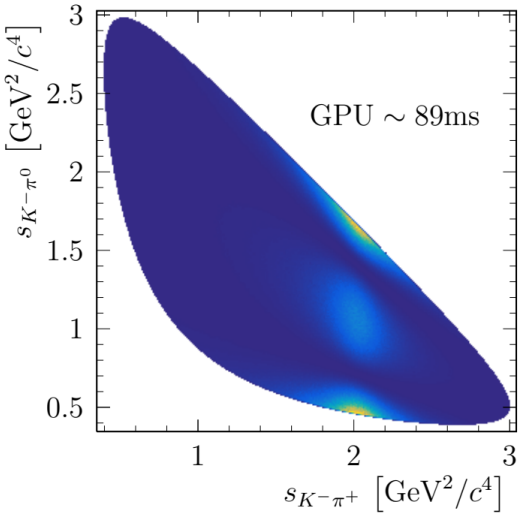
- Started as a divergent branch of Mint, redesigned to deal with charm data sets ($\sim 10^6$ signal candidates)
- Originally used for four-body decays, but now much more
  - Spin $\frac{1}{2}, 1$ initial/final state(s) (including multifermion final states)
  - Indeterminate initial states (quantum-correlated decays, mixing, ...)
  - Supports both canonical and covariant spin formulations.
  - Very flexible support for different propagators (Breit-Wigners, multi-body running widths, K-matrices, 'Dalitz' models i.e. $\eta^0 \to \pi^+\pi^-\pi^0$, ...)
  - Can export models for publication + integration with other frameworks (*i.e.* EvtGen)

GooFit
CUDA/OpenMP
Fitting Framework
for C++ & Python

Part of the GooFit organisation
https://github.com/goofit/ampgen

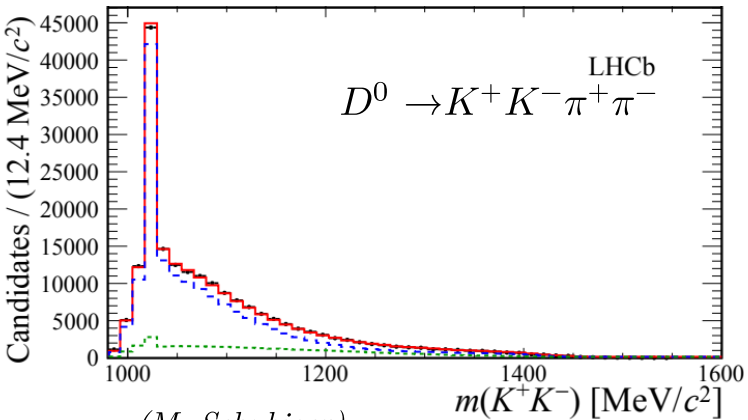
On the more technical side



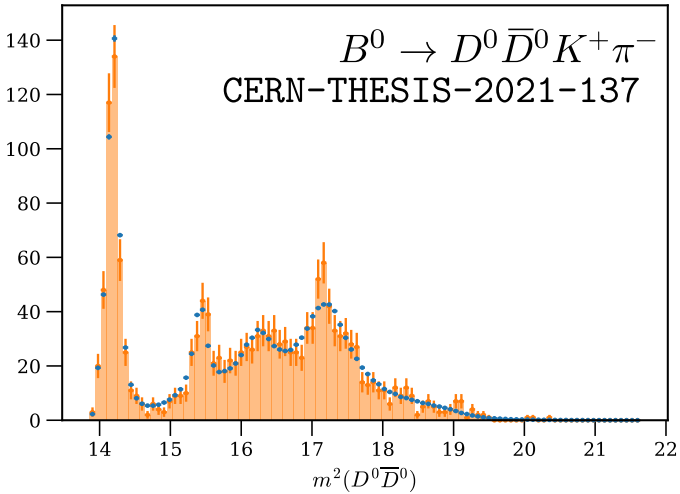


- Default target architecture is AVX2 + OpenMP
- Can also target other architectures:
  - Scalar, AVX512 ... (latter is untested)
  - Can target GPUs but not integrated into the project.
- This (+ design) means AmpGen is very fast
  - From $D^0 \to K^0_S\pi^+\pi^-$ example, takes $\sim$ 3s to do FCN evaluation 'from scratch' with 1 million signal + 2.5 million integration events. $\implies$ about 1 minute to do the fit (20 amplitudes, 38 free parameters)
- Example compares CPU (scalar, single core) vs GPU for $D^0 \to K^{*0}\pi^0$

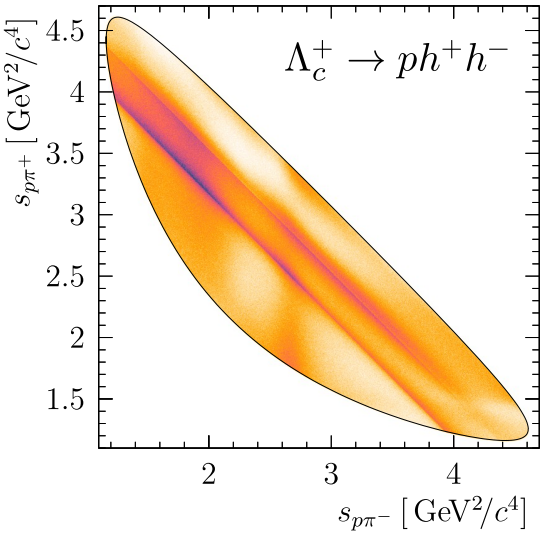
Some Example Analyses

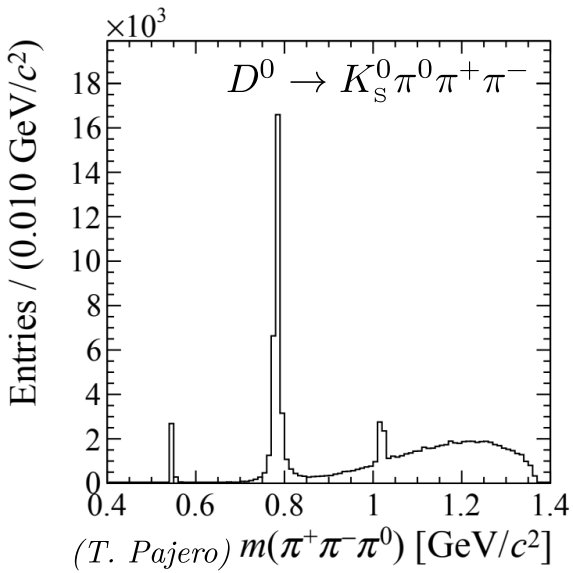


(M. Schubiger)

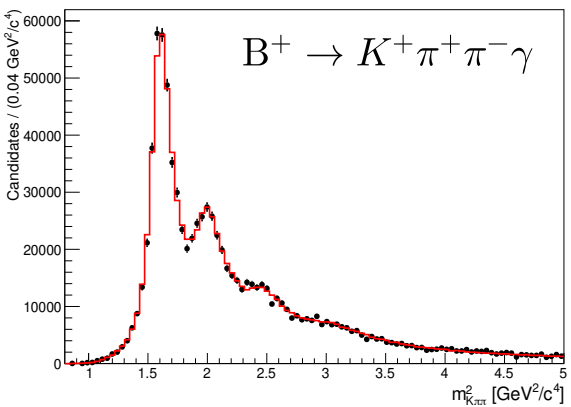


(S. Bhasin, J. Amey, M. Whitehead)

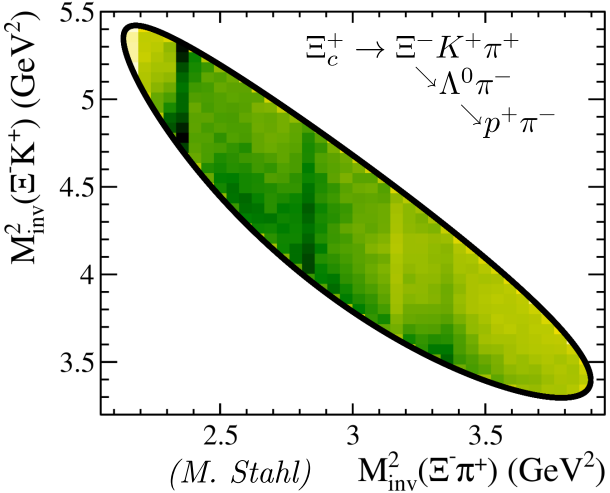




(T. Pajero)



(V.Bellee, M. Diaz, P. Pais)



(M. Stahl)

Unit Tests and Publication of Models

It is **infamously** difficult to reproduce the results of amplitude analyses with code/framework other than whatever was used for the paper

- What we call a 'decay constant' vs what is encoded by the dynamic functions is *almost* entirely arbitrary

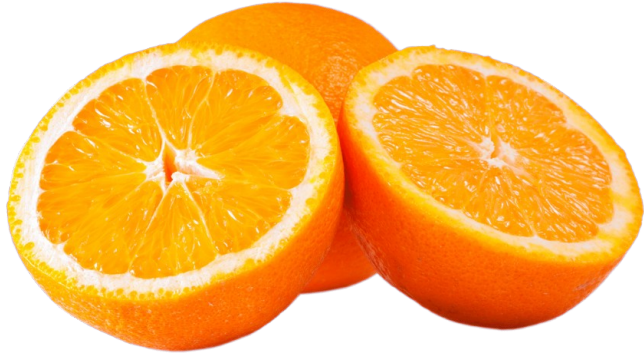
$$\mathcal{M} \sim g_{D^0 \to K^0_S \rho} \mathcal{T}_{D^0 \to K^0_S \rho} (\times g_{\rho \to \pi\pi})$$

- (Almost) no-one agrees on a common set of conventions: tends to be reasonable choices for each analysis / framework
- Also makes it virtually impossible to have sensible 'unit tests' that are supposed to agree between frameworks

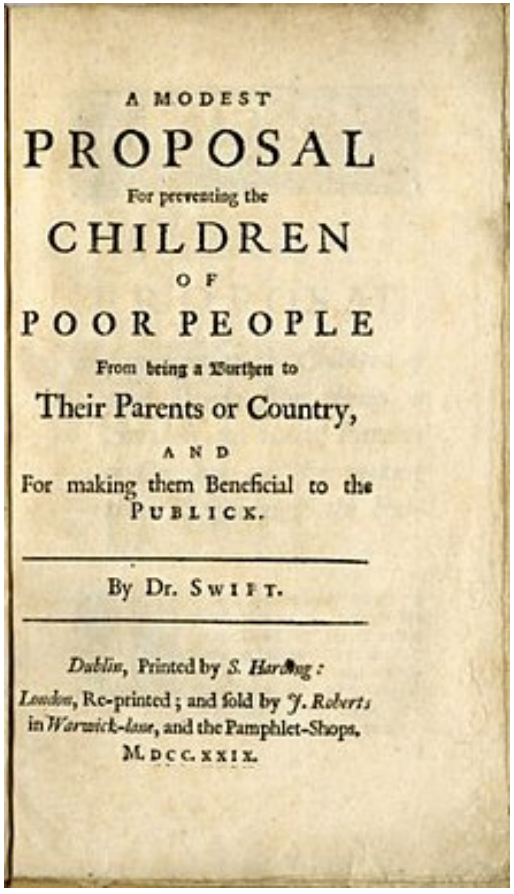
'Solved' problem for Dalitz models: publish high resolution histograms of amplitude / phase of models

$\implies$ Generally doesn't work for $> 2d$ problems due to curse-of-dimensionality

$\implies$ $> 2d$ requires some mechanism to publish code (preferably relatively independent of frameworks)



Benchmarks



A MODEST

PROPOSAL

For preventing the

CHILDREN

OF

POOR PEOPLE

From being a Burthen to

Their Parents or Country,

AND

For making them Beneficial to the

PUBLICK.

By Dr. SWIFT.

Dublin, Printed by *S. Harding*:

London, Re-printed; and sold by *J. Roberts* in *Warwick-lane*, and the Pamphlet-Shops.

M.DCC.XXIX.

Can compare results from a few key decays (no specific proposals here)
$\implies$ Less about comparing results / performance and more about showing some common examples for those starting an analysis to be able to compare different options

- Time to do fits is relevant *if* it becomes a bottleneck (or we start limiting what we do such that can be done in reasonable compute time)
- Readability
- Length of code
- Something else?

Basis can be a set of signal-only simulation samples different frameworks can try and fit in their own way
$\implies$ Critical that these are generated with whatever framework/code was used to write the model in the first place..