Partial-Wave Analysis of $B \to J/\psi K\pi$ at Belle and Belle II

DEMOS Hadron Models Meeting

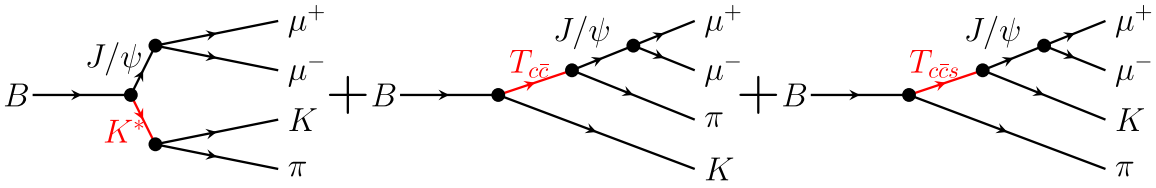
Martin Bartl, Hans-Günther Moser, Stefan Wallner
(bartl@mpp.mpg.de)

Max Planck Institute for Physics

May 29, 2026

Analysis Basics

- $B \to J/\psi(\to \mu^+\mu^-)K\pi$ is a 4 body decay
  $\Rightarrow$ isobar model $\Rightarrow$ subsequent 2-body decays with resonance X
- For our decay channel resonances X can appear in three different topologies



($K\pi$) topology ($J/\psi\pi$) topology ($J/\psi K$) topology

Implementation

- Code base is in `python` using mostly publicly available packages

Partial Wave Analysis Model I

Basic Intensity Model

- Same initial and final state for all resonances
  $\Rightarrow$ interference of all resonances on amplitude level
- We only measure the distribution in the phase-space variables τ, i.e. the intensity
  $\Rightarrow$ coherently sum amplitudes A_i for resonances i to form measured intensity $I(\tau)$

$$I(\tau) = \left|A_{\text{tot}}(\tau)\right|^2 = \left|\sum_i A_i(\tau)\right|^2$$

- Factorize amplitude into 2 parts:
  - decay amplitude $\Psi_i(\tau) \Rightarrow$ angular and energy dependence
  - complex coupling amplitude C_i of resonance $\Leftarrow$ we want to measure that
- Phase space τ is four-dimensional

Partial Wave Analysis Model I

Basic Intensity Model

- Same initial and final state for all resonances
  $\Rightarrow$ interference of all resonances on amplitude level
- We only measure the distribution in the phase-space variables τ, i.e. the intensity
  $\Rightarrow$ coherently sum amplitudes A_i for resonances i to form measured intensity $I(\tau)$

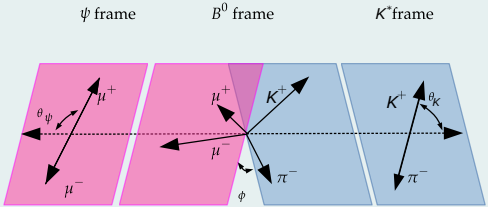
$$I(\tau) = \left|A_{\text{tot}}(\tau)\right|^2 = \left|\sum_i A_i(\tau)\right|^2 = \left|\sum_i C_i \Psi_i(\tau)\right|^2$$

- Factorize amplitude into 2 parts:
  - decay amplitude $\Psi_i(\tau) \Rightarrow$ angular and energy dependence
  - complex coupling amplitude C_i of resonance $\Leftarrow$ we want to measure that
- Phase space τ is four-dimensional

Phase-Space Variables

Angular Dependencies

- First consider only resonances in the $K\pi$ subsystem, i.e. K^* resonances



- θ_K: helicity angle of $K \Rightarrow$ angle between K and B^0 direction in the K^* rest frame
- θ_ψ: helicity angle of $\psi \Rightarrow$ angle between μ^+ and B^0 direction in the ψ rest frame
- ϕ: angle between K^* and ψ decay planes in the B^0 rest frame

Energy Dependence

- $m_{K\pi}$: invariant mass of the $K^+\pi^-$ subsystem

All phase space variables can be calculated directly from the 4-momenta p of all final state particles

Phase-Space Variables Calculation I

Energy Dependence

- Invariant masses are straight forward using the sum of p for topologies' particles
  e.g. $m^2_{K\pi} = (p_K + p_\pi)^2 = (E_K + E_\pi)^2 - (\mathbf{p}_K + \mathbf{p}_\pi)^2$

```
# Minkoski Metric
eta = _np.array([[1., 0., 0., 0.], [0., -1., 0., 0.],
           [0., 0., -1., 0.], [0., 0., 0., -1.]])
# Diagonal of the metric to be used in einsum:
et = _np.array([eta[0, 0], eta[1, 1], eta[2, 2], eta[3, 3]])

def lp(a, b):
   """Lorentz product of lorentz vectors a and b.
   The contraction is performed along the first axis of a and b
   """
   return _math.einsum('i...,i...,i->...', a, b, et)
```

Phase-Space Variables Calculation II

Angular Dependencies

- Angular variables are more difficult to calculate, need to boost the final state momenta in 3 frames: B rest frame, isobar rest frame, and J/ψ rest frame
  $\Rightarrow$ custom code to calculate all angles for given topology
  $\Rightarrow$ needs matching for individual topology, e.g. $K\pi$ topology

```
angles_12['cosTheta_1__12'],angles_12['phi_12_34__1234'],angles_12['cosTheta_3__34']
```

- Calculation of each angle still relatively simple using:
  - boost using `lorentz` package
  - standard math operations:
    - dot product and cross product
    - arctan2, arccos
    - sqrt

Partial Wave Analysis Model II

Detailed Intensity Model

- Reminder of the Basic idea

$$I(\tau) = \left|A_{\text{tot}}(\tau)\right|^2 = \left|\sum_i A_i(\tau)\right|^2 = \left|\sum_i C_i \Psi_i(\tau)\right|^2$$

- We choose the helicity formalism for the partial-wave amplitudes:
  - Coherently sum over all internal helicities $\lambda \Rightarrow$ intermediate resonance states and J/ψ
  - Incoherently sum over all external helicities $\xi \Rightarrow \mu^+$ and μ^- from the J/ψ decay

$$I(\tau) = \sum_\xi \left|\sum_{i,\lambda} C_\lambda^i \Psi_{i,\lambda,\xi}(\tau)\right|^2$$

- Add $(J/\psi\pi)$-topology resonances j and $(J/\psi K)$-topology resonances k
  $\Rightarrow$ each topology with different phase space variables, i.e. τ' and τ''

Detailed Intensity Model

- Reminder of the Basic idea

$$I(\tau) = \left|A_{\text{tot}}(\tau)\right|^2 = \left|\sum_i A_i(\tau)\right|^2 = \left|\sum_i C_i \Psi_i(\tau)\right|^2$$

- We choose the helicity formalism for the partial-wave amplitudes:
  - Coherently sum over all internal helicities $\lambda \Rightarrow$ intermediate resonance states and J/ψ
  - Incoherently sum over all external helicities $\xi \Rightarrow \mu^+$ and μ^- from the J/ψ decay

$$I(\tau) = \sum_\xi \left| \sum_{i,\lambda} C_\lambda^i \Psi_{i,\lambda,\xi}(\tau) + \sum_{j,\lambda} C_\lambda^j \Psi_{j,\lambda,\xi}(\tau'(\tau)) + \sum_{k,\lambda} C_\lambda^i \Psi_{k,\lambda,\xi}(\tau''(\tau)) \right|^2$$

- Add $(J/\psi\pi)$-topology resonances j and $(J/\psi K)$-topology resonances k
  $\Rightarrow$ each topology with different phase space variables, i.e. τ' and τ''

Partial Wave Analysis Model III

Amplitude Parametrization

- The amplitude for partial wave i with spin J_i, e.g. in the $(K\pi)$-topology is:

$$\Psi_{i,\lambda,\xi}(\tau) = d^{J_i}_{\lambda 0}(\theta_K)\, e^{i\lambda\phi} d^1_{\lambda\xi}(\theta_\psi)\, R(m_{K\pi})$$

- Small Wigner-d functions $d^J_{\lambda\lambda'}$ describe how a resonance with spin J and helicity λ decays into particles with summed helicity $\lambda' \Rightarrow$ **model independent**
  - Use general Wigner-d functions, e.g. from `sympy`
    $\Rightarrow$ simplify and evaluate to get condensed functions of θ

```
d = Wigner.d(j, l1, l2, theta).doit().evalf()
```

- R is the resonance line shape $\Rightarrow$ dependency on the invariant mass $m_{K\pi} \Rightarrow$ **model dependent**

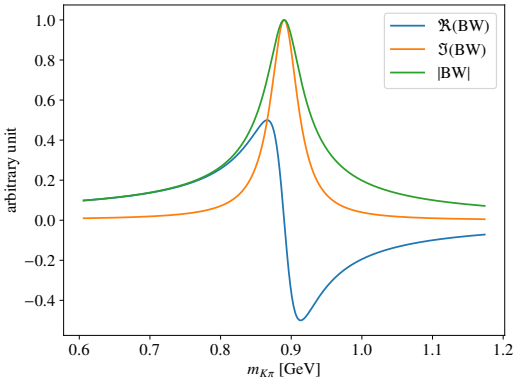
$\Rightarrow$ split up angular and energy dependence when setting up amplitudes

Example Line Shape

- Take spin-1 resonance $K^*(892)$
- Lineshape is relativistic Breit-Wigner

$$\mathrm{BW}(m_{K\pi}) = \frac{m_0 \Gamma_0}{m_0^2 - m_{K\pi}^2 - i m_0 \Gamma(m_{K\pi})}$$

- Multiplied by barrier factors
  $\Rightarrow$ simple to calculate from phase space variables using standard math operations




```
shape = sphysics.amplitudes.resonance.breitWignerS(m_K_pi,mass_pi,mass_K,L,mass,width)
```

Putting Everything Together I

Wave Set

- Decide which resonances are in the final model in a dictionary

```
"K*0(1430)":{"mass":1.425,"width":0.27,"spin":0,"model":"LASS","topology":(1, 2)}
"Tcc1(4200)":{"mass":4.196,"width":0.37,"spin":1,"model":"BW","topology": (3, 4, 2)}
```

Definition of the Individual Amplitudes

- For each possible resonance, λ, and ξ combination build one amplitude

```
x, y, z = symbols("theta_resonance phi theta_jpsi", real=True)
# Dependence on the angles using sympy
angular_resonance = Wignerd(x, resonance_spin, res_hel, 0)
plane_angular = sympy.exp(S(sympy.I*res_hel))
angular_jpsi = Wignerd(z, 1, res_hel, muon_helicity)
total_angular_dependence = angular_resonance*plane_angular*angular_jpsi
# produce callable funtion
angular_dependence = lambdify((x,y,z,v), total_angular_dependence, "numpy")
# Then get the lineshape, i.e. the mass dependence with numpy/tensorflow backend
mass_dependence = lineshape_Kpi(model,mass,width,resonance_spin)
```

Putting Everything Together II

Total Intensity

$$\sum_{\xi} \left| \sum_{i,\lambda} C_{\lambda}^{i} \Psi_{i,\lambda,\xi}(\tau) + \sum_{j,\lambda} C_{\lambda}^{j} \Psi_{j,\lambda,\xi}(\tau'(\tau)) + \sum_{k,\lambda} C_{\lambda}^{i} \Psi_{k,\lambda,\xi}(\tau''(\tau)) \right|^2$$

- Combine couplings C_{λ}^{i} with model amplitudes
  $\Rightarrow$ match C_{λ}^{i} to $\Psi_{i,\lambda,\xi}$ and corresponding τ

Further Application

- In some models one needs to further combined the amplitudes
  $\Rightarrow$ i.e. build linear combinations with given parameters
- For fitting we need to do some matrix multiplication
  $\Rightarrow$ possibility to this with complex valued amplitudes and matrices

Summary and Outlook

Summary

- Partial-wave analysis of $B \to J/\psi K\pi \Rightarrow$ multiple decay topologies
- Amplitude model using helicity formalism written in `python`
- Angular part is written in `sympy`
- Lineshapes build using `numpy/tensorflow`
  $\Rightarrow$ angular dependencies combined with model dependent line shape as callable function
- Full model is build from config file with resonance definitions

Backup

Partial Wave Analysis Model IV

ℓs Basis

- Reminder: Couplings $C_{i,\lambda}$ depend on helicity $\Rightarrow$ mixture of different parity states
- Isobar states have well-defined quantum number J^P $\Rightarrow$ utilize this by extension into ℓs basis
  - $\vec{s} = \vec{s_1} \oplus \vec{s_2}$ with total intrinsic spin $\vec{s}$ and spin $\vec{s_i}$ of the decay particles
  - $\vec{J} = \vec{s} \oplus \vec{\ell}$, with total spin $\vec{J}$ of mother particle and orbital angular momentum $\vec{\ell}$

  $\Rightarrow$ well defined parity P, conserved in strong decays of $T_{c\bar{c}}$: $P(\pi)P(J/\psi)(-1)^\ell = P(T_{c\bar{c}})$
  $\Rightarrow$ potential test of different quantum number hypotheses

Maximum-Likelihood Fits I

Likelihood

- For given model and model parameters likelihood of data sample τ is product of probability P for each event k in τ

$$\mathcal{L}(\{\tau\}) = \prod_k P(\tau_k)$$

- For easier numerical handling convert into log-likelihood

Maximum-Likelihood Fits I

Likelihood

- For given model and model parameters likelihood of data sample τ is product of probability P for each event k in τ

$$\mathcal{L}(\{\tau\}) = \prod_k P(\tau_k) \Rightarrow \log \mathcal{L}(\{\tau\}) = \sum_k \log P(\tau_k)$$

- For easier numerical handling convert into log-likelihood

Maximum-Likelihood Fits I

Likelihood

- For given model and model parameters likelihood of data sample τ is product of probability P for each event k in τ

$$\mathcal{L}(\{\tau\}) = \prod_k P(\tau_k) \Rightarrow \log \mathcal{L}(\{\tau\}) = \sum_k \log P(\tau_k)$$

- For easier numerical handling convert into log-likelihood

Event Probability

- PDF is intensity of an event normalized with the number of expected events $\hat{N}_{\text{ev}}$

$$P(\tau) = \frac{\sum_{a,b} C_a C_b^* \Psi_a(\tau) \Psi_b(\tau)^* \Phi(\tau)}{\hat{N}_{\text{ev}}}$$

with phase space element $\Phi(\tau)$

Maximum-Likelihood Fits II

Integral Matrix

- The normalization $\hat{N}_{\text{ev}}$ is

$$\hat{N}_{\text{ev}} = \int d\tilde{\tau}\Phi(\tilde{\tau}) \sum_{a,b} C_a C_b^* \Psi_a(\tilde{\tau})\Psi_b(\tilde{\tau})^* = \sum_{a,b} C_a C_b^* \int d\tilde{\tau}\Phi(\tilde{\tau})\Psi_a(\tilde{\tau})\Psi_b(\tilde{\tau})^* = \sum_{a,b} C_a I_{a,b} C_b^*$$

- Integral matrix $I_{a,b} \Rightarrow$ no free parameters $\Rightarrow$ can be pre-calculated

Maximum-Likelihood Fits III

Likelihood

- Insert PDF and integral matrix into the likelihood and add Poisson term $\Rightarrow$ treat the number of events N_{ev} as random variable

$$\mathcal{L}(\{\tau\}) = \frac{\prod_k^{N_{\text{ev}}} \sum_{a,b} C_a C_b^* \Psi_a(\tau_k) \Psi_b(\tau_k)^*}{\left[\sum_{a,b} C_a I_{a,b} C_b^*\right]^{N_{\text{ev}}}} \prod_k^{N_{\text{ev}}} \Phi(\tau_k) \frac{\left[\hat{N}_{\text{ev}}\right]^{N_{\text{ev}}}}{N_{\text{ev}}!} e^{-\hat{N}_{\text{ev}}}$$

- Take log and drop parameter-free terms and cancel equal terms

$$\log \mathcal{L}(\{\tau\}) = \sum_k^{N_{\text{ev}}} \log \left[\sum_{a,b} C_a C_b^* \Psi_a(\tau_k) \Psi_b(\tau_k)^* \right] - \sum_{a,b} C_a I_{a,b} C_b^*$$