

Contents

1. Theoretical framework	2
1.1. <i>Kinematics of $B \rightarrow K\pi\pi\gamma$ decay</i>	2
1.2. Resonances parametrization	6
1.3. Model of $1^+ \rightarrow (1^- \rightarrow 0^-0^-)0^-$ decays	6
1.3.1. <i>Model of quark pairs creation</i>	6
1.3.2. <i>Hadronic form-factors</i>	7
1.3.3. <i>Helicity currents</i>	7
1.4. Model of $1^- \rightarrow (1^- \rightarrow 0^-0^-)0^-$ decays	8
1.5. Model of $2^+ \rightarrow (1^- \rightarrow 0^-0^-)0^-$ decays	8
1.6. Model of $2^- \rightarrow (1^- \rightarrow 0^-0^-)0^-$ decays	9
1.7. Model of $2^- \rightarrow (1^- \rightarrow 0^-0^-)0^-$ decays	9
1.8. <i>Decay rate</i>	9
2. Generator	10
2.1. Notes about EvtGen	14
3. Fitter	16
3.1. Fitting	17

GamPola

Author:

Borys KNYSH

1. Theoretical framework

$$B^+ \rightarrow K^+ \pi^+ \pi^- \gamma \quad p = q \sum_{l=0}^{l=N-1} \frac{1}{l!} (-\ln q)^l \quad B^0 \rightarrow K^+ \pi^0 \pi^- \gamma$$

This section contains overview of theoretical expressions used in order to describe differential decay rate.

1.1. Kinematics of $B \rightarrow K\pi\pi\gamma$ decay

This subsection contains the expressions of the kinematical variables used to describe $B \rightarrow K\pi\pi\gamma$ decay. Majority of expressions are encapsulated in the *GKinematics* module, but some of them such as Lorenz's boosts and Euler's rotations are included in the modules *EvtGVector3D* and *EvtGVector4D*. Following are given building blocks which depend on each other:

- Defining Lorenz's transformation:

$$E = \gamma(E' + (\vec{\beta} \cdot \vec{p}')) \quad (1)$$

$$\vec{p} = \vec{p}' + (\gamma - 1) \vec{\beta} \frac{(\vec{\beta} \cdot \vec{p}')}{\beta^2} + \gamma \vec{\beta} E' \quad (2)$$

- Defining Euler's rotations:

$$v_x = v_{0x}(\cos \chi \cos \theta \cos \varphi - \sin \chi \sin \varphi) + v_{0y}(-\sin \chi \cos \theta \cos \varphi - \cos \chi \sin \varphi) + v_{0z} \sin \theta \cos \varphi \quad (3)$$

$$v_y = v_{0x}(\cos \chi \cos \theta \sin \varphi + \sin \chi \cos \varphi) + v_{0y}(-\sin \chi \cos \theta \sin \varphi + \cos \chi \cos \varphi) + v_{0z} \sin \theta \sin \varphi \quad (4)$$

$$v_z = v_{0x}(-\cos \chi \sin \theta) + v_{0y}(\sin \chi \sin \theta) + v_{0z} \cos \theta \quad (5)$$

- Calculating energy and momentum of final state particles in the rest frame of decaying particle in 2-body decay (*GKinematics::GMomentaA*, *GKinematics::GEnergyA*):

$$p(s_{ijk}, s_{ij}, m_b) = \sqrt{\frac{(s_{ijk} - (\sqrt{s_{ij}} + m_b)^2)}{4s_{ijk}}} \quad (6)$$

$$E(s_{ijk}, s_{ij}, m_b) = \frac{s_{ijk} - m_b^2 + s_{ij}}{2\sqrt{s_{ijk}}} \quad (7)$$

- Initialization the following kinematical constants using Eq. 7: $E_{K_{892}^*}, E_{\rho_{775}}, E_{K_{1430}^*}$. They will be required for helicity amplitudes calculation;
- Calculating lower and upper limits of the Dalitz plot (*GKinematics::GSjkMin* and *GKinematics::GSjkMax*):

$$E_j^* = E(s_{ij}, m_i^2, m_j) \quad (8)$$

$$E_k^* = \frac{s_{ijk} - s_{ij} - m_k^2}{2\sqrt{s_{ij}}} \quad (9)$$

$$s_{jk}^{low} = (E_j^* + E_k^*)^2 - (\sqrt{E_j^{2*} - m_j^2} + \sqrt{E_k^{2*} - m_k^2})^2 \quad (10)$$

$$s_{jk}^{up} = (E_j^* + E_k^*)^2 - (\sqrt{E_j^{2*} - m_j^2} - \sqrt{E_k^{2*} - m_k^2})^2 \quad (11)$$

- Defining the dot product of two pions' 4 momentum (*GKinematics::GPi1Pi2Vec4*), which will be used for calculating 1^+ amplitudes;

$$p_i \cdot p_j = \frac{s_{K\pi_1\pi_2} - (s_{K\pi_1} + s_{K\pi_2}) + m_K^2}{2} \quad (12)$$

- Defining the cross product of two pions' 3 momentum (*GKinematics::GPiPjVec3*). Note that this cross product is used to define 2^+ amplitude;
- Calculating dependent dalitz variable (*GKinematics::GSij*), which will be usefull for calculation of $\varphi_K, \varphi_{\pi_1}, \varphi_{\pi_2}$ angles:

$$s_{ik} = s_{ijk} - s_{ij} - s_{jk} + m_i^2 + m_j^2 + m_k^2 \quad (13)$$

- Calculating dot product of two pions' 3D vectors (*GKinematics::GPIpj*). Such product is used for calculating $\varphi_K, \varphi_{\pi_1}, \varphi_{\pi_2}$ angles, which will be required for 4-momentum calculation of K, π_1, π_2 in the K_{res} -rest frame:

$$(\vec{p}_{\pi_1} \cdot \vec{p}_{\pi_2}) = \frac{E_{\pi_1}E_{\pi_2} + (m_{\pi_1}^2 + m_{\pi_2}^2 - s_{\pi_1\pi_2})}{2} \quad (14)$$

- Calculating scalar product of two arbitrary 3D-vectors:

$$(\vec{v}_i \cdot \vec{v}_j) = v_{i1}v_{j1} + v_{i2}v_{j2} + v_{i3}v_{j3} \quad (15)$$

Required to define 2^+ amplitude;

- Calculating $[\vec{p}_{\pi_1} \times \vec{p}_{\pi_2}]$. Will be used for calculating 2^+ amplitude (*GKinematics::GPi1Pi2Vec3*):

$$[\vec{p}_{\pi_1} \times \vec{p}_{\pi_2}] = (p_{\pi_1 y}p_{\pi_2 z} - p_{\pi_1 z}p_{\pi_2 y})\vec{i} + (p_{\pi_1 z}p_{\pi_2 x} - p_{\pi_1 x}p_{\pi_2 z})\vec{j} + (p_{\pi_1 x}p_{\pi_2 y} - p_{\pi_2 x}p_{\pi_1 y})\vec{k} \quad (16)$$

- Generation of valid phase space of three body decay (*GKinematics::GGetPhaseSpace*):

Algorithm 1: Phase space generation

Result: $(\cos \theta, \varphi, s_{K\pi_1}, s_{\pi_1\pi_2}, s_{K\pi\pi})$

while 1 **do**

$s_{K\pi\pi} = \text{uniform}(s_{min}, s_{max});$
 $s_{\pi\pi} = \text{uniform}((m_{1\pi} + m_{2\pi})^2, (\sqrt{s_{K\pi\pi}} - m_K)^2);$
 $s_{K\pi} = \text{uniform}((m_K + m_\pi)^2, (\sqrt{s_{K\pi\pi}} - m_\pi)^2);$
 $s_{\pi\pi}^{min}$ and $s_{\pi\pi}^{max}$ calculations using Eqs. 10-11;
if $s_{\pi\pi} \geq s_{\pi\pi}^{min}$ and $s_{\pi\pi} \leq s_{\pi\pi}^{max}$ **then**
| break the loop;
end

end

$\cos \theta = \text{uniform}(-1, 1);$

$\varphi = \text{uniform}(0, 2\pi);$

- Following expressions are used to calculate all the kinematical variables used for helicity amplitudes calculation (*GKinematics::GCalculateKinematics*). These computations are performed for each event:

- Defining the $\cos \tilde{\theta}$ which is needed for neutral decay modes (without it, the integration of the Dalitz variables cancel the A_{ud} -asymmetry)

$$\cos \tilde{\theta} = \begin{cases} \cos \theta & \text{charge} = 1 \\ \cos \theta & \text{charge} = 0, s_{K\pi_1} \geq s_{K\pi_2} \\ -\cos \theta & \text{charge} = 0, s_{K\pi_1} < s_{K\pi_2} \end{cases} \quad (17)$$

where $\cos \theta$ is provided¹;

- Calculating momentum and energies of pion and kaon using Eqs.6-7;
- Calculating energy and momentum of photon in K_{res} rest frame:

$$E_\gamma = |\vec{p}_\gamma| = \frac{m_B - s_{K\pi\pi}}{2\sqrt{s_{K\pi\pi}}}; \quad (18)$$

- Calculating scalar product of two 3 and 4-dimensional momenta of pions using Eqs.12-14;
- Defining angle between two pions in the K_{res} rest frame using Eqs. 6,14:

$$\delta = \arccos\left(\frac{(\vec{p}_{\pi_1} \cdot \vec{p}_{\pi_2})}{|\vec{p}_{\pi_1}| \cdot |\vec{p}_{\pi_2}|}\right) \quad (19)$$

- Defining angle between K and π_1 using Eqs. 6,19:

$$\alpha = \arctan\left(\frac{|\vec{p}_{\pi_2}| \sin \delta}{|\vec{p}_{\pi_1}| + |\vec{p}_{\pi_2}| \cos \delta}\right) \quad (20)$$

¹Despite the fact that helicity amplitudes are calculated using modified $\cos \tilde{\theta}$, distribution of $\cos \theta$ is sampled from the p.d.f.

- Defining azimuthal angles between x-axis and corresponding final state particles in the $K\pi\pi$ rest-frame:

$$\varphi_{\pi_1} = \frac{2\varphi - \delta}{2} \quad (21)$$

$$\varphi_{\pi_2} = \frac{2\varphi + \delta}{2} \quad (22)$$

$$\varphi_K = \pi + \alpha + \varphi - \frac{\delta}{2} \quad (23)$$

- The pion's modified 3-momenta in the $\tilde{\theta}, \phi_{\pi,K}$ coordinates (used for calculation helicity currents):

$$\vec{p}_{\pi_1} = |\vec{p}_{\pi_1}|(\cos \tilde{\theta} \cos \varphi_{\pi_1}; \sin \varphi_{\pi_1}; -\sin \tilde{\theta} \cos \varphi_{\pi_1}) \quad (24)$$

$$\vec{p}_{\pi_2} = |\vec{p}_{\pi_2}|(\cos \tilde{\theta} \cos \varphi_{\pi_2}; \sin \varphi_{\pi_2}; -\sin \tilde{\theta} \cos \varphi_{\pi_2}) \quad (25)$$

$$\vec{p}_K = |\vec{p}_K|(\cos \tilde{\theta} \cos \varphi_K; \sin \varphi_K; -\sin \tilde{\theta} \cos \varphi_K) \quad (26)$$

- Calculating $([\vec{p}_{\pi_1} \times \vec{p}_{\pi_2}] \cdot \vec{\varepsilon}_0)$, $(\vec{p}_{\pi_1} \cdot \vec{\varepsilon}_0)$ and $(\vec{p}_{\pi_2} \cdot \vec{\varepsilon}_0)$ using Eqs.15-16;
- Calculating kinematical coefficients for 2^+ amplitude:

$$\vec{K}_{C4} = (\vec{p}_{\pi_1} \cdot \vec{\varepsilon}_0) \cdot [\vec{p}_{\pi_1} \times \vec{p}_{\pi_2}] + ([\vec{p}_{\pi_1} \times \vec{p}_{\pi_2}] \cdot \vec{\varepsilon}_0) \cdot \vec{p}_{\pi_1} \quad (27)$$

$$\vec{K}_{C5} = (\vec{p}_{\pi_2} \cdot \vec{\varepsilon}_0) \cdot [\vec{p}_{\pi_1} \times \vec{p}_{\pi_2}] + ([\vec{p}_{\pi_1} \times \vec{p}_{\pi_2}] \cdot \vec{\varepsilon}_0) \cdot \vec{p}_{\pi_2} \quad (28)$$

- Conversion of the generated phase space to the set of four 4-momenta of final state particles in the B rest frame:

- Defining the pion's and kaon's modified 3-momenta in the $\theta, \phi_{\pi,K}$ coordinates:

$$\vec{p}_{\pi_1} = |\vec{p}_{\pi_1}|(\cos \theta \cos \varphi_{\pi_1}; \sin \varphi_{\pi_1}; -\sin \theta \cos \varphi_{\pi_1}) \quad (29)$$

$$\vec{p}_{\pi_2} = |\vec{p}_{\pi_2}|(\cos \theta \cos \varphi_{\pi_2}; \sin \varphi_{\pi_2}; -\sin \theta \cos \varphi_{\pi_2}) \quad (30)$$

$$\vec{p}_K = |\vec{p}_K|(\cos \theta \cos \varphi_K; \sin \varphi_K; -\sin \theta \cos \varphi_K) \quad (31)$$

- Defining 4-momenta of final state particles in the K_{res} rest frame:

$$p_\gamma = (E_\gamma; 0; 0; -|\vec{p}_\gamma|); \quad (32)$$

$$p_{\pi_1} = (E_{\pi_1}; \vec{p}_{\pi_1}); \quad (33)$$

$$p_{\pi_2} = (E_{\pi_2}; \vec{p}_{\pi_2}); \quad (34)$$

$$p_K = (E_K; \vec{p}_K) \quad (35)$$

- Defining boost vector from K_{res} rest frame to B rest frame:

$$s_{K\pi_1\pi_2} = E_K + E_{\pi_1} + E_{\pi_2} \quad (36)$$

$$\vec{\beta} = (0; 0; \frac{M_B^2 - s_{K\pi_1\pi_2}}{M_B^2 + s_{K\pi_1\pi_2}}) \quad (37)$$

This generic representation allows flexible operating of well-known objects such as Lorenz vectors. In particular to have such conversion function will be extremely useful while plugging “GamPola” model to the *EvtGen*.

- Performing the boost and introducing new 4-momenta of final state particles in the B rest frame using Eq.;
- Synchronous randomizing of the directions of final state particles in the B rest frame using Euler's angles and Eq.:

$$\alpha^{rand} = 2\pi \cdot uniform(0, 1) \quad (38)$$

$$\beta^{rand} = \arccos(uniform(-1, 1)) \quad (39)$$

- Final state 4-momenta to phase space conversion (*GKinematics::G4VectorsToPhaseSpace*) is applied when given input final state particles with their 4-momenta in the B-rest frame:

- Defining 4-momentum of K_{res} :

$$p_{K_{res}} = p_K + p_{\pi_1} + p_{\pi_2} \quad (40)$$

- Calculating $s_{K\pi_1}, s_{\pi_1\pi_2}, s_{K\pi_1\pi_2}$ (Dalitz components of phase space):

$$s_{K\pi_1} = (p_K + p_{\pi_1})^2 \quad (41)$$

$$s_{\pi_1\pi_2} = (p_{\pi_1} + p_{\pi_2})^2 \quad (42)$$

$$s_{K\pi_1\pi_2} = p_{K_{res}}^2 \quad (43)$$

- Defining boosting vector from B rest frame to K_{res} rest frame:

$$\vec{\beta}_{B \rightarrow K_{res}} = -\frac{\vec{p}_{K_{res}}}{E_{K_{res}}} \quad (44)$$

- Performing boost of final state particles from B rest frame to K_{res} rest frame (will be marked as *) using Eq.;
- Defining normal vector to $K\pi\pi$ plane:

$$\vec{n} = \frac{[\vec{p}_{\pi_1} \times \vec{p}_{\pi_2}]}{||[\vec{p}_{\pi_1} \times \vec{p}_{\pi_2}]||} \quad (45)$$

- Defining the basis of the frame, where \vec{e}_y vector is in the $K\pi\pi$ plane and \vec{e}_x is in the same plane with \vec{e}_z and \vec{n} :

$$\vec{e}_z = -\frac{\vec{p}_\gamma^*}{E_\gamma^*} \quad (46)$$

$$\vec{e}_y = \frac{[\vec{e}_z \times \vec{n}]}{||[\vec{e}_z \times \vec{n}]||} \quad (47)$$

$$\vec{e}_x = [\vec{e}_y \times \vec{e}_z] \quad (48)$$

- Calculating the angle between photon and normal to the $K\pi\pi$ plane (this angle is one of the dimensions of the phase space):

$$\cos \theta = \vec{e}_z \cdot \vec{n} \quad (49)$$

- Defining \vec{e}'_x which lies in the $K\pi\pi$ plane:

$$\vec{e}'_x = \vec{e}_x \cos \theta - \vec{e}_z \sin \theta \quad (50)$$

- Since \vec{e}'_x and \vec{e}_y are in the same $K\pi\pi$ plane it is possible to calculate angles of two pions²:

$$\varphi_{\pi_1} = \arctan \frac{(\vec{e}_y \cdot \vec{p}_{\pi_1})}{(\vec{e}'_x \cdot \vec{p}_{\pi_1})} \quad (51)$$

$$\varphi_{\pi_2} = \arctan \frac{(\vec{e}_y \cdot \vec{p}_{\pi_2})}{(\vec{e}'_x \cdot \vec{p}_{\pi_2})} \quad (52)$$

Hence $\varphi_{\pi_1}, \varphi_{\pi_2} \in [-\pi; \pi]$;

- Calculating last but not least component of phase space by definition:

$$\phi = \frac{\varphi_{\pi_1} + \varphi_{\pi_2}}{2} \quad (53)$$

This variable stands for combined angle of two pions in the $K\pi\pi$ plane. The value of $\phi \in [-2\pi; 2\pi]$, however originally generator produces corresponding values in the range $[0; 2\pi]$. Hence, additional shifts to the conventional region are required;

- Calculating combined angle of two pions in the valid region using Eq. 19:

$$\varphi = \begin{cases} \phi + 2\pi & \delta = \varphi_{\pi_2} - \varphi_{\pi_1}, \phi < 0 \\ \phi & \delta = \varphi_{\pi_2} - \varphi_{\pi_1}, \phi \geq 0, \\ \phi + \pi & \delta \neq \varphi_{\pi_2} - \varphi_{\pi_1} \end{cases} \quad (54)$$

²in C++ arctan function is represented as atan2(y,x) function, which returns values in the range $[-\pi; \pi]$

1.2. Resonances parametrization

This subsection describes the parametrization of each resonance participating in the decay rate expression. Given below expressions are implemented within *GBreitWigner* module. Despite the fact that resonances incorporated in the current module are not isolated³ this parametrization brings simplicity to the model. List of implemented resonances, their masses and widths is given in the Table.

Resonance	nominal mass, GeV	nominal width, GeV
K^{*892}	0.896	0.047
ρ^{775}	0.775	0.149
K_1^{1270}	1.272	0.090
K_1^{1400}	1.403	0.174
K^{*1410}	1.414	0.232
K^{*1680}	1.717	0.322
K_2^{1430}	1.426	0.099

Table 1. Masses and widths of implemented resonances

- Individual K_{res} are parametrized as relativistic Breit-Wigner functions with fixed width:

$$BW_{K_{res}}(s_{K\pi\pi}) = \frac{1}{s_{K\pi\pi} - M_{K_{res}}^2 + i\Gamma_{K_{res}} M_{K_{res}}} \quad (55)$$

where $M_{K_{res}}$ and $\Gamma_{K_{res}}$ stand for nominal mass and width of K_{res} resonance respectively;

- K^{*892} resonance is parametrized in the same way:

$$BW_{K^{*892}}(s_{K\pi}) = \frac{1}{s_{K\pi} - M_{K^{*892}}^2 + i\Gamma_{K^{*892}} M_{K^{*892}}} \quad (56)$$

where M_{K^*} and Γ_{K^*} stand for nominal mass and width of K^* respectively;

- ρ^{775} shape accounts for energy dependent width:

$$\Gamma_{\rho^{775}}(s_{\pi_1\pi_2}) = \Gamma_{\rho^{775}} \left[\frac{s_{\pi_1\pi_2} - (m_{\pi_1} + m_{\pi_2})^2}{M_{\rho^{775}}^2 - (m_{\pi_1} + m_{\pi_2})^2} \right]^{3/2} \quad (57)$$

$$BW_{\rho^{775}} = \frac{1}{s_{\pi\pi} + M_{\rho^{775}}^2 + i\Gamma_{\rho^{775}}(s_{\pi_1\pi_2})M_{\rho^{775}}} \quad (58)$$

where M_ρ and Γ_ρ stand for nominal mass and width of ρ respectively;

1.3. Model of $1^+ \rightarrow (1^- \rightarrow 0^- 0^-) 0^-$ decays

1.3.1. Model of quark pairs creation

This subsection contains expressions of definition of spacial integrals of decays $A \rightarrow B + C$, S, D waves and decay amplitudes $A_{S/D}^{K_{res}}$, which were originally represented in the PhD thesis of A.Tayduganov:

- Author derives spacial integrals from harmonic oscillator meson wave functions and using Eq. 6:

$$p_{res} = p(s_{K\pi\pi}, s_{ij}, m_k^2) \quad (59)$$

$$I_0^{(ABC)} = -\frac{4\sqrt{3}}{\pi^{5/4}} \frac{R_A^{5/2}(R_B R_C)^{3/2}}{(R_A^2 + R_B^2 + R_C^2)^{5/2}} \left(1 - p_{res}^2 \frac{(2R_A^2 + R_B^2 + R_C^2)(R_B^2 + R_C^2)}{4(R_A^2 + R_B^2 + R_C^2)} \right) \times \exp \left[-p_{res}^2 \frac{R_A^2(R_B^2 + R_C^2)}{8(R_A^2 + R_B^2 + R_C^2)} \right] \quad (60)$$

$$I_1^{(ABC)} = \frac{4\sqrt{3}}{\pi^{5/4}} \frac{R_A^{5/2}(R_B R_C)^{3/2}}{(R_A^2 + R_B^2 + R_C^2)^{5/2}} \exp \left[-p_{res}^2 \frac{R_A^2(R_B^2 + R_C^2)}{8(R_A^2 + R_B^2 + R_C^2)} \right] \quad (61)$$

where $R_A = R_B = R_C = 0.4$ — meson function raduis;

³Usually case Breit-Wigner parametrization is chosen for isolated resonance.

- Using $I_0^{(ABC)}, I_1^{(ABC)}$, S and D waves expressions are calculated using Eqs. 6,60,61

$$p_{res}^{const} = p(M_{K_{res}}^2, m_{ij}^2, m_k) \quad (62)$$

$$S^{(ABC)} = \gamma \sqrt{\frac{3}{2}} \frac{2I_1^{(ABC)} - I_0^{(ABC)}}{18} \exp[(p_{res}^{const})^2 - p_{res}^2] f_2 \quad (63)$$

$$D^{(ABC)} = \gamma \sqrt{\frac{3}{2}} \frac{I_1^{(ABC)} + I_0^{(ABC)}}{18} \exp[(p_{res}^{const})^2 - p_{res}^2] f_2 \quad (64)$$

where γ, f_2 — model parameters defined in the Table. $M_{K_{res}}$ — nominal mass of resonance, m_{ij}, m_k — nominal masses of pseudo vector and pseudo scalar particles obtained in the 2 body decay: $K_{res} \rightarrow PV + PS$. Exponential term is introduced in order to account for the cut-off in the coupling vertices.

- Calculating decay amplitudes of K_1 into $K^* \pi$ or $K \rho$ final states:

$$EC_{(K\pi_i)\pi_j} = 8\sqrt{s\pi^3 E(s, s_{K\pi_i}, m_{\pi_j}) E(s, m_{\pi_j}^2, \sqrt{s_{K\pi_i}})} \quad (65)$$

$$EC_{K(\pi_i\pi_j)} = 8\sqrt{s\pi^3 E(s, s_{\pi_i\pi_j}, m_K) E(s, m_K^2, \sqrt{s_{\pi_i\pi_j}})} \quad (66)$$

$$\tilde{A}_S^{K_1(1270) \rightarrow (K\pi_i)\pi_j} = EC_{(K\pi_i)\pi_j} g_{1270,S}^{K\pi_i} S^{(K\pi_i\pi_j)} (\sqrt{2} \sin \theta_{K_1} - \cos \theta_{K_1}) \quad (67)$$

$$\tilde{A}_D^{K_1(1270) \rightarrow (K\pi_i)\pi_j} = EC_{(K\pi_i)\pi_j} g_{1270,D}^{K\pi_i} D^{(K\pi_i\pi_j)} (-\sin \theta_{K_1} - \sqrt{2} \cos \theta_{K_1}) \quad (68)$$

$$\tilde{A}_S^{K_1(1400) \rightarrow (K\pi_i)\pi_j} = EC_{(K\pi_i)\pi_j} g_{1400,S}^{K\pi_i} S^{(K\pi_i\pi_j)} (\sqrt{2} \cos \theta_{K_1} + \sin \theta_{K_1}) \quad (69)$$

$$\tilde{A}_D^{K_1(1400) \rightarrow (K\pi_i)\pi_j} = EC_{(K\pi_i)\pi_j} g_{1400,D}^{K\pi_i} D^{(K\pi_i\pi_j)} (-\cos \theta_{K_1} + \sqrt{2} \sin \theta_{K_1}) \quad (70)$$

$$\tilde{A}_S^{K_1(1270) \rightarrow K(\pi_i\pi_j)} = EC_{K(\pi_i\pi_j)} g_{1270,S}^{\pi_i\pi_j} S^{(K\pi_i\pi_j)} (\sqrt{2} \sin \theta_{K_1} + \cos \theta_{K_1}) \quad (71)$$

$$\tilde{A}_D^{K_1(1270) \rightarrow K(\pi_i\pi_j)} = EC_{K(\pi_i\pi_j)} g_{1270,D}^{\pi_i\pi_j} D^{(K\pi_i\pi_j)} (-\sin \theta_{K_1} + \sqrt{2} \cos \theta_{K_1}) \quad (72)$$

$$\tilde{A}_S^{K_1(1400) \rightarrow K(\pi_i\pi_j)} = EC_{K(\pi_i\pi_j)} g_{1400,S}^{\pi_i\pi_j} S^{(K\pi_i\pi_j)} (\sqrt{2} \cos \theta_{K_1} - \sin \theta_{K_1}) \quad (73)$$

$$\tilde{A}_D^{K_1(1400) \rightarrow K(\pi_i\pi_j)} = EC_{K(\pi_i\pi_j)} g_{1400,D}^{\pi_i\pi_j} D^{(K\pi_i\pi_j)} (-\cos \theta_{K_1} - \sqrt{2} \sin \theta_{K_1}) \quad (74)$$

In the original paper author introduces only θ_{K_1} — mixing angle between non-mass eigenstates, but in order to enable flexibility of the fit 8 additional complex parameters are introduced: $g_{1270,S}^{K^*/\rho}, g_{1270,D}^{K^*/\rho}, g_{1400,S}^{K^*/\rho}, g_{1400,D}^{K^*/\rho}$.

1.3.2. Hadronic form-factors

Once amplitudes are calculated one can compute hadronic form-factors for 1^+ :

$$A_{K^*}^{K_1(1270/1400) \rightarrow (K\pi_i)\pi_j} = \tilde{A}_S^{K_1(1270/1400) \rightarrow (K\pi_i)\pi_j} + \sqrt{2} \tilde{A}_D^{K_1(1270/1400) \rightarrow (K\pi_i)\pi_j} \exp(i\varphi_{K^*}^D) \quad (75)$$

$$A_{\rho}^{K_1(1270/1400) \rightarrow K(\pi_i\pi_j)} = (\tilde{A}_S^{K_1(1270/1400) \rightarrow K(\pi_i\pi_j)} + \sqrt{2} \tilde{A}_D^{K_1(1270/1400) \rightarrow K(\pi_i\pi_j)} \exp(i\varphi_{\rho}^D)) \exp(i\varphi_{\rho}^S) \quad (76)$$

$$B_{K^*}^{K_1(1270/1400) \rightarrow (K\pi_i)\pi_j} = \left[\left[1 - \sqrt{\frac{s_{K\pi_i}}{E(s, s_{K\pi_i}, m_{\pi_j})}} \right] \tilde{A}_S^{K_1(1270/1400) \rightarrow (K\pi_i)\pi_j} + \sqrt{2} \left[1 + \frac{2\sqrt{s_{K\pi_i}}}{E(s, s_{K\pi_i}, m_{\pi_j})} \right] \tilde{A}_D^{K_1(1270/1400) \rightarrow (K\pi_i)\pi_j} \exp(i\varphi_{K^*}^D) \right] \frac{E(s, s_{K\pi_i}, m_{\pi_j})}{\sqrt{sp(s, s_{K\pi_i}, m_{\pi_j})^2}} \quad (77)$$

$$B_{\rho}^{K_1(1270/1400) \rightarrow K(\pi_i\pi_j)} = \left[\left[1 - \sqrt{\frac{s_{\pi_i\pi_j}}{E(s, s_{\pi_i\pi_j}, m_K)}} \right] (\tilde{A}_S^{K_1(1270/1400) \rightarrow K(\pi_i\pi_j)} + \sqrt{2} \left[1 + \frac{2\sqrt{s_{\pi_i\pi_j}}}{E(s, s_{\pi_i\pi_j}, m_K)} \right] \tilde{A}_D^{K_1(1270/1400) \rightarrow K(\pi_i\pi_j)} \exp(i\varphi_{\rho}^D)) \exp(i\varphi_{\rho}^S) \right] \times \frac{E(s, s_{\pi_i\pi_j}, m_K)}{\sqrt{sp(s, s_{\pi_i\pi_j}, m_K)^2}} \quad (78)$$

Within the expressions three model parameters are introduced ($\varphi_S^{\rho}, \varphi_D^{\rho}, \varphi_D^{K^*}$). These parameters are called hadronic phases and they responsible for defining direction of S and D waves in the complex space.

1.3.3. Helicity currents

After calculation of hadronic form-factors one can obtain expressions for helicity currents:

- $C_{12}^{K^*(1270/1400)}, C_{12}^{\rho(1270/1400)}$:

$$\begin{aligned} C_1^{K^*(1270/1400)} &= CG_{K^*} g_{K^* K \pi} \left[\left[A_{K^*}^{K_1(1270/1400) \rightarrow (K \pi_2) \pi_1} \left(1 + \frac{m_{\pi_1}^2 - m_K^2}{s_{K \pi_2}} \right) + B_{K^*}^{K_1(1270/1400) \rightarrow (K \pi_2) \pi_1} \right. \right. \\ &\times \left. \left(2p_{\pi_1} p_{\pi_2} - \left(1 + \frac{m_{\pi_1}^2 - m_K^2}{s_{K \pi_2}} \right) (\sqrt{s} E_{\pi_1} - m_{\pi_1}^2) \right) \right] BW_{K^*}(s_{K \pi_2}) \\ &- \left. 2A_{K^*}^{K_1(1270/1400) \rightarrow (K \pi_1) \pi_2} BW_{K^*}(s_{K \pi_1}) \delta \right] \end{aligned} \quad (79)$$

$$\begin{aligned} C_2^{K^*(1270/1400)} &= CG_{K^*} g_{K^* K \pi} \left[\left[A_{K^*}^{K_1(1270/1400) \rightarrow (K \pi_1) \pi_2} \left(1 + \frac{m_{\pi_2}^2 - m_K^2}{s_{K \pi_1}} \right) + B_{K^*}^{K_1(1270/1400) \rightarrow (K \pi_1) \pi_2} \right. \right. \\ &\times \left. \left(2p_{\pi_1} p_{\pi_2} - \left(1 + \frac{m_{\pi_2}^2 - m_K^2}{s_{K \pi_1}} \right) (\sqrt{s} E_{\pi_2} - m_{\pi_2}^2) \right) \right] BW_{K^*}(s_{K \pi_1}) \\ &- \left. 2A_{K^*}^{K_1(1270/1400) \rightarrow (K \pi_2) \pi_1} BW_{K^*}(s_{K \pi_2}) \delta \right] \end{aligned} \quad (80)$$

$$C_1^{\rho(1270/1400)} = CG_{\rho} g_{\rho \pi \pi} \left[A_{\rho}^{K_1(1270/1400) \rightarrow K(\pi_1 \pi_2)} - B_{\rho}^{K_1(1270/1400) \rightarrow K(\pi_1 \pi_2)} \sqrt{s} (E_{\pi_1} - E_{\pi_2}) \right] BW_{\rho}(s_{\pi_1 \pi_2}) \quad (81)$$

$$C_2^{\rho(1270/1400)} = CG_{\rho} g_{\rho \pi \pi} \left[A_{\rho}^{K_1(1270/1400) \rightarrow K(\pi_1 \pi_2)} + B_{\rho}^{K_1(1270/1400) \rightarrow K(\pi_1 \pi_2)} \sqrt{s} (E_{\pi_1} - E_{\pi_2}) \right] BW_{\rho}(s_{\pi_1 \pi_2}) \quad (82)$$

where $CG_{K^*892} = \begin{cases} -\frac{2}{3}, & K_{res}^+ \rightarrow K^+ \pi^+ \pi^- \\ \frac{\sqrt{2}}{3}, & K_{res}^0 \rightarrow K^+ \pi^0 \pi^- \end{cases}$, $CG_{\rho} = \begin{cases} -\frac{1}{\sqrt{6}}, & K_{res}^+ \rightarrow K^+ \pi^+ \pi^- \\ \frac{1}{\sqrt{3}}, & K_{res}^0 \rightarrow K^+ \pi^0 \pi^- \end{cases}$ — Clebsh-Gordon coefficients,

$\delta = \begin{cases} 0, & K_{res}^+ \rightarrow K^+ \pi^+ \pi^- \\ 1, & K_{res}^0 \rightarrow K^+ \pi^0 \pi^- \end{cases}$ — coefficient which accounts for decay of $K^{*892} \rightarrow K^+ \pi^0$;

- $C_{12}^{(1270/1400)}$:

$$C_{1,2}^{(1270/1400)} = \left[(C_{1,2}^{K^*(1270/1400)} + C_{1,2}^{\rho(1270/1400)}) \right] BW^{(1270/1400)}(s) \quad (83)$$

- Helicity currents of 1270 and 1400 can be written as:

$$\vec{J}^{1270/1400} = f^{1270/1400} \left[C_1^{1270/1400} \vec{p}_{\pi_1} - C_2^{1270/1400} \vec{p}_{\pi_2} \right] \quad (84)$$

where $f^{1270/1400}$ stands for coupling constants of 1270/1400 resonances.

1.4. Model of $1^- \rightarrow (1^- \rightarrow 0^- 0^-) 0^-$ decays

- $C_3^{K^*(1410/1680)}, C_3^{\rho(1410/1680)}$:

$$C_3^{K^*(1410/1680)} = 2i\sqrt{s} CG_{K^*} g_{K^* K \pi} \tilde{g}_{K^*}^{1410/1680} \left[BW_{K^*}(s_{K \pi_2}) + \delta \cdot BW_{K^*}(s_{K \pi_1}) \right] \quad (85)$$

$$C_3^{\rho(1410/1680)} = 2i\sqrt{s} CG_{\rho} g_{\rho \pi \pi} \tilde{g}_{\rho}^{1410/1680} BW_{\rho}(s_{\pi_1 \pi_2}) \quad (86)$$

- $C_3^{(1410/1680)}$:

$$C_3^{1410/1680} = \left[C_3^{K^*(1410/1680)} + C_3^{\rho(1410/1680)} \right] BW^{1410/1680}(s) \quad (87)$$

- Helicity currents of 1410 and 1680 can be expressed as:

$$\vec{L}^{1410/1680} = f^{1410/1680} C_3^{1410/1680} [\vec{p}_{\pi_1} \times \vec{p}_{\pi_2}] \quad (88)$$

1.5. Model of $2^+ \rightarrow (1^- \rightarrow 0^- 0^-) 0^-$ decays

- $C_{4,5}^{K^*(1430)}, C_4^{\rho(1430)}$:

$$C_4^{K^*(1430)} = \sqrt{2} si CG_{K^*} g_{K^* K \pi} \tilde{g}_{K^*}^{1430} BW_{K^*}(s_{K \pi_2}) \quad (89)$$

$$C_5^{K^*(1430)} = \sqrt{2} si CG_{K^*} g_{K^* K \pi} \tilde{g}_{K^*}^{1430} BW_{K^*}(s_{K \pi_1}) \quad (90)$$

$$C_4^{\rho(1430)} = \sqrt{2} si CG_{\rho} g_{\rho \pi \pi} \tilde{g}_{\rho}^{1430} BW_{K^*}(s_{\pi_1 \pi_2}) \quad (91)$$

- $C_{4,5}^{1430}$:

$$C_{4,5}^{1430} = f^{1430}(m_B^2 - s) \left[C_4^{p(1430)} \mp C_4^{K^*(1430)} \right] BW^{1430}(s) \quad (92)$$

$$(93)$$

- Helicity current of 1430:

$$\begin{aligned} \vec{K}^{1430} &= C_4^{1430} \left[(\vec{\varepsilon}_0 \cdot \vec{p}_{\pi_1}) [\vec{p}_{\pi_1} \times \vec{p}_{\pi_2}] + \vec{p}_{\pi_1} (\vec{\varepsilon}_0 \cdot [\vec{p}_{\pi_1} \times \vec{p}_{\pi_2}]) \right] \\ &+ C_5^{1430} \left[(\vec{\varepsilon}_0 \cdot \vec{p}_{\pi_2}) [\vec{p}_{\pi_1} \times \vec{p}_{\pi_2}] + \vec{p}_{\pi_2} (\vec{\varepsilon}_0 \cdot [\vec{p}_{\pi_1} \times \vec{p}_{\pi_2}]) \right] \end{aligned} \quad (94)$$

1.6. Model of $2^- \rightarrow (1^- \rightarrow 0^- 0^-) 0^-$ decays

$$C_{\pi_1 \pi_2} = 2 \left((f_V + h_V \sqrt{s} * (E_{\pi_1} + E_{\pi_2})) (m_{\pi_1}^2 - m_{\pi_2}^2) / s_{\pi_1 \pi_2} - h_V \sqrt{s} (E_{\pi_1} - E_{\pi_2}) \right) \quad (95)$$

$$C_{K \pi_1} = 2 \left((f_V + h_V \sqrt{s} (E_{\pi_1} + E_K)) (m_{\pi_1}^2 - m_K^2) / s_{K \pi_1} - h_V \sqrt{s} (E_{\pi_1} - E_K) \right) \quad (96)$$

$$C_{K \pi_2} = 2 \left((f_V + h_V \sqrt{s} (E_{\pi_2} + E_K)) (m_{\pi_2}^2 - m_K^2) / s_{K \pi_2} - h_V \sqrt{s} (E_{\pi_2} - E_K) \right) \quad (97)$$

$$K_{\pi_1 \pi_2} = i \sqrt{\frac{2}{s}} C G_\rho B W_{\rho^{775}}(s_{\pi_1 \pi_2}) \quad (98)$$

$$K_{K \pi_1} = i \sqrt{\frac{2}{s}} C G_{K^*} B W_{K^*892}(s_{K \pi_1}) \quad (99)$$

$$K_{K \pi_2} = i \sqrt{\frac{2}{s}} C G_{K^*} B W_{K^*892}(s_{K \pi_2}) \quad (100)$$

$$(101)$$

$$c_{\pi_1 \pi_2} = K_{\pi_1 \pi_2} \left(C_{\pi_1 \pi_2} ((\vec{\varepsilon}_0 \cdot \vec{p}_{\pi_1}) + (\vec{\varepsilon}_0 \cdot \vec{p}_{\pi_2})) - 2 f_V^p (\vec{\varepsilon}_0 \cdot \vec{p}_{\pi_1}) \right) \quad (102)$$

$$c_{K \pi_1} = \delta K_{K \pi_1} \left(f_V^{K^*} (\vec{\varepsilon}_0 \cdot \vec{p}_{\pi_1}) + 2 C_{K \pi_1} ((\vec{\varepsilon}_0 \cdot \vec{p}_{\pi_1}) + (\vec{\varepsilon}_0 \cdot \vec{p}_{\pi_2})) \right) \quad (103)$$

$$c_{K \pi_2} = K_{K \pi_2} f_V^{K^*} (\vec{\varepsilon}_0 \cdot \vec{p}_{\pi_2}) \quad (104)$$

$$C_6 = c_{\pi_1 \pi_2} + c_{K \pi_1} + c_{K \pi_2} \quad (105)$$

$$\vec{M} = f^{1580} (C_6 \vec{p}_{\pi_1} - C_7 \vec{p}_{\pi_2}) \quad (106)$$

1.7. Model of $2^- \rightarrow (1^- \rightarrow 0^- 0^-) 0^-$ decays

1.8. Decay rate

Radiative $b \rightarrow s \gamma$ transition occurs either through $b \rightarrow s_L \gamma_L$ or $b \rightarrow s_R \gamma_R$. Since the final states are different, differential decay rate is the sum of two decay rates:

$$\frac{d\Gamma}{d \cos \theta d\varphi ds_{K \pi_1} ds_{\pi_1 \pi_2} ds} = \left[\frac{d\Gamma}{d \cos \theta d\varphi ds_{K \pi_1} ds_{\pi_1 \pi_2} ds} \right]_L + \left[\frac{d\Gamma}{d \cos \theta d\varphi ds_{K \pi_1} ds_{\pi_1 \pi_2} ds} \right]_R \quad (107)$$

Therefore additional modifications to the original helicity currents of $K_{res} \rightarrow K\pi\pi$ decay should be applied:

$$\vec{J}_{L,R}^{1270} = \vec{J}'^{1270} \quad (108)$$

$$\vec{J}_{L,R}^{1400} = \frac{m_B^2 - m_{K_1^{1400}}^2}{m_B^2 - m_{K_1^{1270}}^2} \vec{J}'^{1400} \quad (109)$$

$$\vec{L}_{L,R}^{1410} = \pm \left[\frac{m_B^2 - m_{K_1^{1410}}^2}{m_B^2 - m_{K_1^{1270}}^2} \right]^{\frac{3}{2}} \vec{L}'^{1410} \quad (110)$$

$$\vec{L}_{L,R}^{1680} = \pm \left[\frac{m_B^2 - m_{K_1^{1680}}^2}{m_B^2 - m_{K_1^{1270}}^2} \right]^{\frac{3}{2}} \vec{L}'^{1680} \quad (111)$$

$$\vec{K}_{L,R}^{1430} = \pm \left[\frac{m_B^2 - m_{K_1^{1430}}^2}{m_B^2 - m_{K_1^{1270}}^2} \right]^{\frac{3}{2}} \vec{K}'^{1430} \quad (112)$$

$$(113)$$

Then differential decay rate can be expressed as following:

$$\begin{aligned} \frac{d\Gamma}{d\cos\theta d\varphi ds_{K\pi_1} ds_{\pi_1\pi_2} ds} &= \frac{1-\lambda}{2} \left| \vec{\varepsilon}_L \cdot (\vec{J}_L^{1270} + \vec{J}_L^{1400} + \vec{L}_L^{1410} + \vec{L}_L^{1680} + \vec{K}_L^{1430}) \right|^2 \\ &+ \frac{1+\lambda}{2} \left| \vec{\varepsilon}_R \cdot (\vec{J}_R^{1270} + \vec{J}_R^{1400} + \vec{L}_R^{1410} + \vec{L}_R^{1680} + \vec{K}_R^{1430}) \right|^2 \end{aligned} \quad (114)$$

where λ — photon polarization parameter.

2. Generator

Decay rate produced by above theoretical model is a function of phase space and model parameters. The latter remain unchanged during events generation process, while phase space is sampled from allowed kinematical region. The events generation process of 5D data is the following:

Algorithm 2: Events sampling from the p.d.f

Result: Ω

$\Omega = []$

Find maximum decay rate (DR_{max}) value using hill climbing algorithm;

while ($i_{event} < N_{events}$) **do**

$\vec{\Omega}_{i_{event}} = (m_{K\pi\pi}^2, m_{\pi\pi}^2, m_{K\pi}^2, \cos\theta, \varphi) = \text{Generate Phase Space}();$

$DR_{i_{event}} = \text{CalculateDecayRate}(\vec{\Omega}_{i_{event}} | \vec{\alpha});$

$u_i = \text{uniform}(0, DR_{max});$

if $DR_{i_{event}} > u_i$ **then**

 Append $\vec{\Omega}_{i_{event}}$ to Ω ;

 Increase counter i_{event} by unit;

end

end

where Ω — phase space matrix with rows represented phase space vectors $\vec{\Omega}_{i_{events}}$, which passed hit-or-miss condition. Interface for using the generator in user code is the following:

- Create generator object with specifying the type of decay (either $B \rightarrow K^+\pi^+\pi^-\gamma$ or $B \rightarrow K^+\pi^0\pi^-\gamma$) by changing parameter *charge*;
- Set path to the file with model parameters;
- Generate events specifying number of events;
- Write results to root file;

File with model parameters looks like dictionary with key-value pairs. Where key — name of model parameter, value — value of corresponding parameter. Output of the generator is a root file containing 4-momenta of each final state particle for each generated event. Generator relies on:

- *GInterfaceForMinimization.cxx/GInterfaceForMinimization.h* — sets function, variables and parameters which will be used as input for generator core. It is also responsible for looking up the local maximum of decay rate using hill climbing algorithm;
- *GEventsGenerator.cxx/GEventsGenerator.h* — the core of events generation. Here event's decay rates are calculated according to the Algorithms 1-2.
- *GGenerator.cxx/GGenerator.h* — interface between user and generator engine, which defines functionality for setting up the generator, model couplings, events number and calls the Algorithm 2;
- *GGeneratorParser.cxx/GGeneratorParser.h* — utility parser of *.ge files, responsible for extracting (key: value) pairs from the file.

Example to reproduce:

Let us generate phase space distributed according to the probability density function of $K_1^{1270}, K_1^{1400}, K^{*1410}, K^{*1680}, K_2^{1430}$. The following configuration file has to be specified (the rest of parameters assigned to its default values):

```

1 thetaK1:      1.042; # mixing angle of K1(1270) and K1(1400)
2 phiDK*:      0.; # hadronic phase of K*(892) in K1
3 phiSRho:     0.; # hadronic phase of S-wave of rho(770) in K1
4 phiDRho:     0.; # hadronic phase of D-wave of rho(770) in K1
5 lambda:      0.7; # photon polarization parameter
6 ffRe_1400:   0.47; # K1(1400) coupling (real part)
7 ffIm_1400:   0.; # K1(1400) coupling (imaginary part)
8 gg3Kstr_1410: 1.; # K*(892) coupling in K*(1410), real part
9 gg3ImKstr_1410: 0.; # K*(892) coupling in K*(1410), imaginary part
10 gg3Rho_1410: 1.; # rho(770) coupling in K*(1410), real part
11 gg3ImRho_1410: 0.; # rho(770) coupling in K*(1410), imaginary part
12 gg4Kstr_1680: 1.; # K*(892) coupling in K*(1680), real part
13 gg4ImKstr_1680: 0.; # K*(892) coupling in K*(1680), imaginary part
14 gg4Rho_1680: 1.; # rho(770) coupling in K*(1680), real part
15 gg4ImRho_1680: 0.; # rho(770) coupling in K*(1680), imaginary part
16 gg5Kstr_1430: 1.; # K*(892) coupling in K2(1430), real part
17 gg5ImKstr_1430: 0.; # K*(892) coupling in K2(1430), imaginary part
18 gg5Rho_1430: 1.; # rho(770) coupling in K2(1430), real part
19 gg5ImRho_1430: 0.; # rho(770) coupling in K2(1430), imaginary part

```

Listing 1. Configuration file of generator (model.ge)

Below sample script is given:

```

1 #include "GGenerator.h"
2
3 using namespace Gamapola;
4
5 int main()
6 {
7     auto g1 = std::make_shared<GGenerator>(0); // generator object for neutral B-meson decay
8     g1->GSetCouplings("../additionalFiles/model.ge"); // set up the couplings from the configuration file
9     g1->GGenerate(1000); // set up number of events
10    g1->GWriteToFile("neutral.root"); // perform events generation and write results to the root-file
11
12    return 0;
13 }

```

Listing 2. User code of generator

After compiling and running the output root-file is produced. It has the following structure:

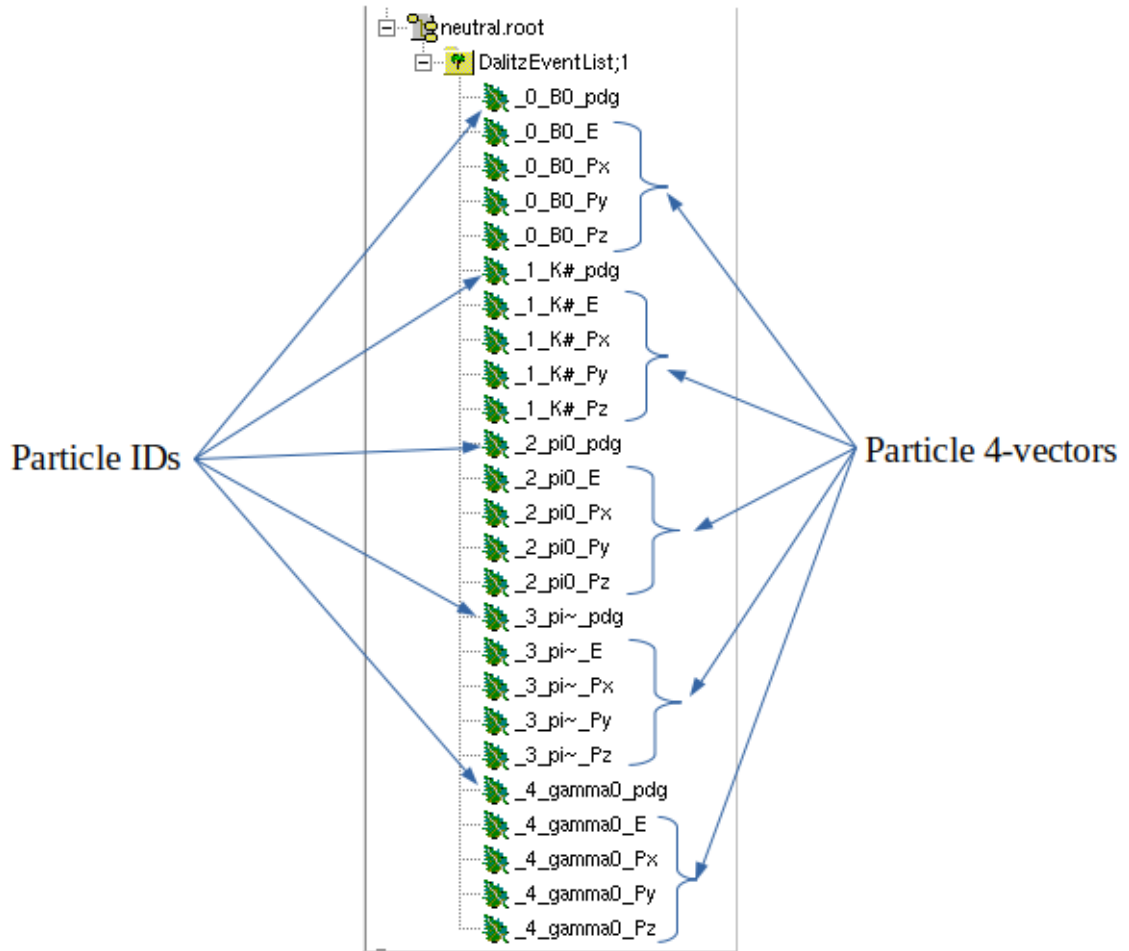
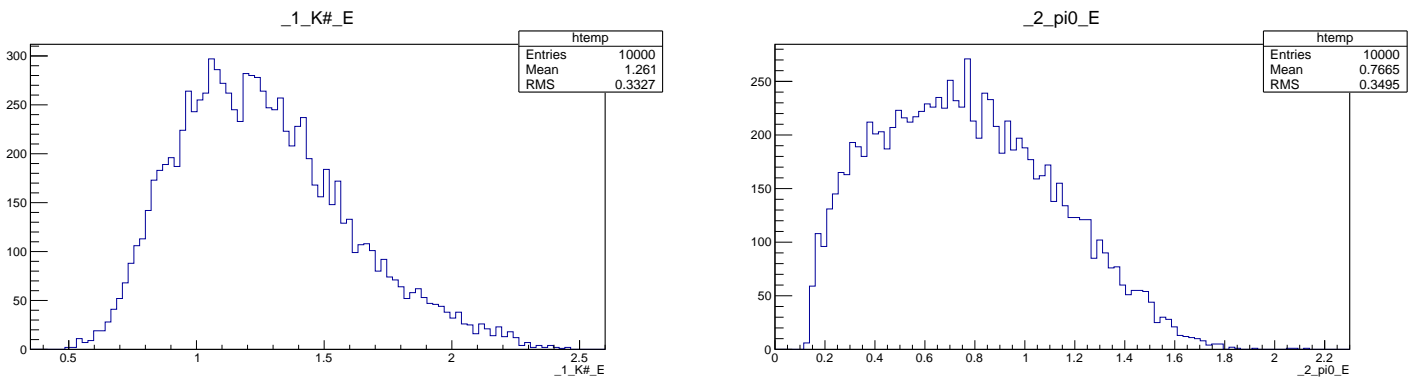


Figure 1. Output root-file structure

Produced 4-vectors are in the B-rest frame.



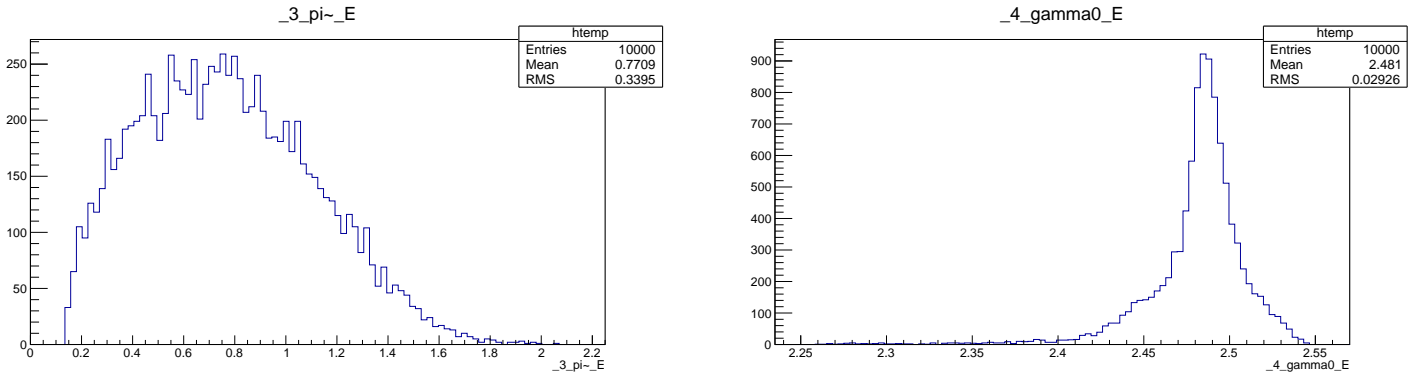


Figure 2. Energies of final state particles in the B-rest frame

Histograms drawing:

In order to obtain phase space distributions the following script can be used:

```

1 #include "GamPolaPlotter/GGampolaPlotter.h"
2
3 using namespace Gamapola;
4
5 int main()
6 {
7     auto&& plotter = std::make_shared<GGampolaPlotter>("../additionalFiles/plot.img"); // create plotter
8     // object with specifying drawing options in plot-file
9     plotter->plot("neutral.root", 0, "neutral_plot.root"); // specify input root file, charge and output
10    // root-file with histograms
11    return 0;
12 }

```

Listing 3. User code of generator

Configuration file for plotting projections of phase space contains:

```

1 #dimensionality, range flag: label: type, bins, low, up
2 1D, All: sKpipi: sKpipi, 100, 1, 2
3 1D, All: sKpi1: sKpi1, 100, 0, 2
4 1D, All: cosTheta: cosTheta, 100, -1, 1
5 1D, All: cosThetaBins10: cosTheta, 10, -1, 1
6
7 #dimensionality, range flag: label: type, bins, low, up
8 1D, cut: 1 < sKpipi < 1.4: cosTHetaCut: cosTheta, 100, -1, 1
9
10 #dimensionality, range flag: label: type, binsX, lowX, upX, binsY,
11 lowY, upY
12 2D, All: sKpi1Kpi2: sKpi1Kpi2, 100, 0.4, 1.4, 100,
13 0.4, 1.4
14 2D, All: sKpi1Kpi210x10: sKpi1Kpi2, 10, 0.4, 1.4, 10,
15 0.4, 1.4

```

Listing 4. Configuration flag for plotting (plot.img)

In general, three options are possible:

- One dimensional plotting in whole range of $M_{K\pi\pi}$;
- One dimensional plotting in specified range of $M_{K\pi\pi}$;
- Two dimensional plotting in whole range of $M_{K\pi\pi}$;

In the configuration file above:

- 1st column is dimensionality of plotted distributions;
- 2nd — cut-flag, which stands for cuts in $M_{K\pi\pi}$ while building the histograms (only cut in one dimension is available);
- 3rd — label, the name given to the distribution. It can be any name;
- 4th — type of distribution has predefined values: *cosTheta*, *sKpipi*, *sKpi1*, etc
- 5th — number of bins for building the histogram;
- 6th — low limit when building the histogram;

- 7th — up limit when building the histogram;

As result the following distributions of phase space can be obtained as specified in *plot.img*:

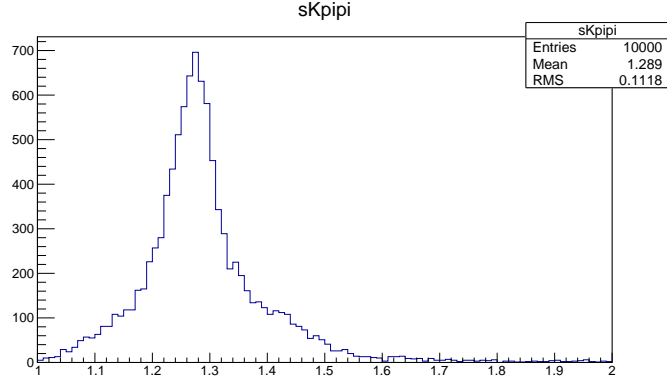


Figure 3. $M_{K\pi\pi}$

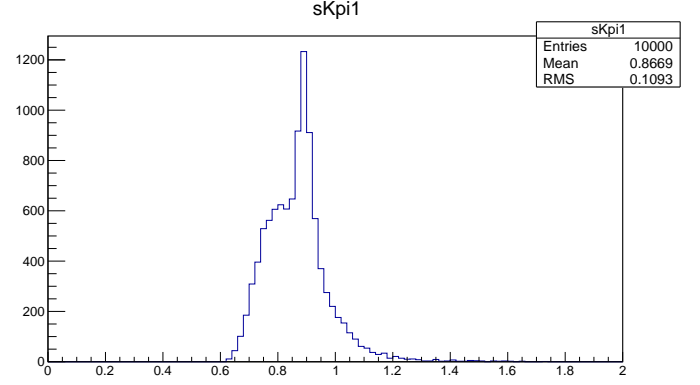


Figure 4. $s_{K\pi_1}$

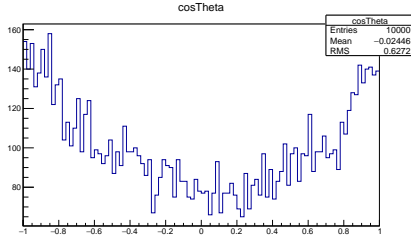


Figure 5. $\cos \theta$

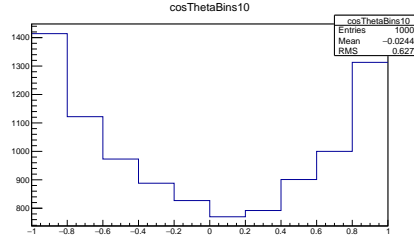


Figure 6. $\cos \theta$ with 10 bins

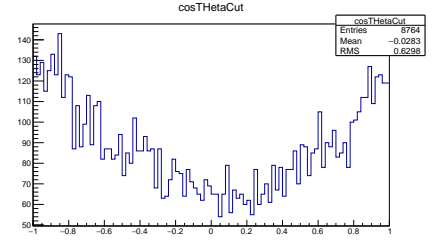


Figure 7. $1 < M_{K\pi\pi} < 1.4 \text{ GeV}$

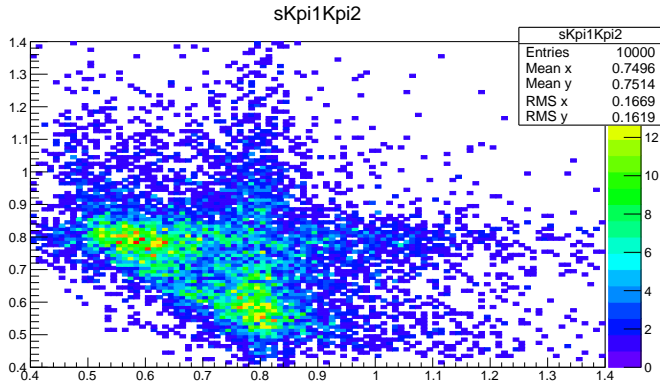


Figure 8. $s_{K\pi_1}$ vs $s_{K\pi_2}$

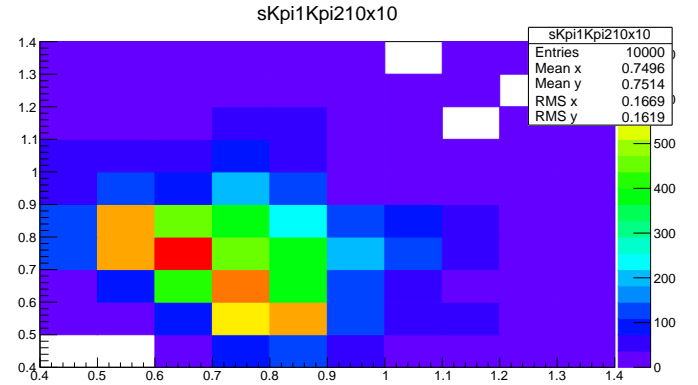


Figure 9. $s_{K\pi_1}$ vs $s_{K\pi_2}$ with 10 bins

2.1. Notes about EvtGen

In order to integrate the model in the EvtGen framework, one has to follow special conventional pattern of user-models insertion. It is achieved by inheriting parent's class *EvtGenAmplitude*. It allows to keep all public and protected methods of parent class, so new model class behaves as parent's one. In particular *EvtGamPolaBToKpipiGamma* class has to implement the following methods apart from constructor, destructor, clone⁴:

- `init()` — memory allocation, setting model parameters, mode;
- `getName()` — set name of the model;
- `initProbMax()` — set maximum value of decay rate for the given set of parameters;

⁴which means that developer of new model has to provide the following information

- `decay(EvtParticle *p)` — is executed for each loop iteration, where `p` stands for B-meson in our particular case;

The way “EvtGen” generates events is similar to the one used in “GamPola”:

- Set maximum decay rate value in the initialization step;
- Use hit-or-miss technique (EvtGen performs 10000 tries to get event, while GamPola has no limit on number of tries);

For clarity the pseudo code for decay looks as following:

```

1 decay
2 {
3   1) Create daughters of B meson decay according to the .DEC files
4   2) Get these daughters
5   3) Using GamPola, generate phase space with kinematical restrictions only
6   4) Substitute phase space point to the decay rate expression
7   5) Calculate PDF using decay rate
8   6) amplitude = sqrt(PDF)
9   7) send amplitude to vertex
10  8) send 4-vectors to daughters
11 }
```

Listing 5. Decay function from EvtGen

For completeness *.DEC* file is given:

```

1 #tion of dm still in some decay models
2 Define dm 0.507e12
3 Define lambdaPhoton 0.95
4 Alias gammaR      gamma
5
6
7 Decay Upsilon(4S)
8 0.5000 B-sig B+          VSS;
9 0.5000 B+sig B-          VSS;
10 Enddecay
11
12
13 Decay B+sig
14
15 1.0 gammaR      K+      pi-      pi+      GAMPOLAR
16 ##### photon polarization:
17      lambdaPhoton
18 ##### gammaQPC & f2:
19      4.          4.
20 ##### strong phases (thetaK1 & phiDK* & phiSRho & phiDRho & kappa):
21      1.0472      0.          0.          0.          0.0          0.0
22 ##### K1_1400 (coupling)#####:
23      0.          0.0
24 ##### K'*0 (coupling)#####:
25      0.          0.0
26 ##### K*0 (892)(couplings)#####:
27      1.0          0.0
28 ##### rho(775) (couplings)#####:
29      1.0          0.0
30 ##### K''*0(coupling)#####:
31      0.          0.0
32 ##### K*0 (892)(couplings)#####:
33      1.0          0.0
34 ##### rho(775)(couplings)#####:
35      1.0          0.0
36 ##### K_2*0 (coupling)#####:
37      0.          0.0
38 ##### K*0 (892)(coupling)#####:
39      1.0          0.0
40 ##### rho(775)(coupling)#####:
41      1.0          0.0;
42 Enddecay
43 #CDecay B-sig
44
45 Decay B-sig
46 Enddecay
47
48 End
```

Listing 6. Configuration decay file from EvtGen

The core of this file is the following string:

Listing 7. Core string

where:

- first entity — decay ratio;
- from second to fifth entity — final state particles of this particular decay;
- sixth entity — model name;

3. Fitter

Fitter utilizes functions from the theoretical framework for calculating decay rates. It also calculates logarithmic likelihood for each optimisation step as a part of gradient descent algorithm. Fitting requires decay rate factorization:

$$\frac{d\Gamma}{d\cos\theta d\varphi ds_{K\pi\pi} ds_{K\pi_1} ds_{\pi_1\pi_2}} = \frac{d\Gamma}{d\vec{\Omega}} = F(\vec{\Omega}, \vec{\alpha}) = \sum_{i=0}^{N_{terms}} f_i(\vec{\Omega}) \cdot g_i(\alpha) \quad (115)$$

where N_{terms} — number of terms within decay rate expression, $f_{i,i}$ — i -th term within decay rate, which depends only on kinematical variables, g_i — i -th term within decay rate, which depends only on model parameters, $\vec{\Omega}$ — point in 5D phase space, $\vec{\alpha}$ — set of model parameters, F — differential decay rate;

After dividing decay rate function on its integral over allowed phase space defined by Algorithm 1 and fixed set of model parameters the p.d.f expression is obtained:

$$p.d.f = \frac{F(\vec{\Omega}, \vec{\alpha})}{\int_{\Omega_0} \sum_{i=0}^{N_{terms}} f_i(\vec{\Omega}) \cdot g_i(\vec{\alpha}) d\vec{\Omega}} = \frac{F(\vec{\Omega}, \vec{\alpha})}{\sum_{i=0}^{N_{terms}} \left[\int_{\Omega_0} f_i(\vec{\Omega}) d\Omega \right] \cdot g_i(\vec{\alpha})} \quad (116)$$

On practice it means summation of the decay rates. It is also assumed that points of phase space are uniformly distributed.

$$\int_{\Omega_0} f_i(\vec{\Omega}) d\vec{\Omega} = \lim_{N \rightarrow \infty} \sum_{j=0}^N f_i(\vec{\Omega}_j) \frac{\Omega_0}{N} \approx \sum_{j=0}^{N_\Sigma} f_i(\vec{\Omega}_j) \frac{\Omega_0}{N_\Sigma} \quad (117)$$

where $\vec{\Omega}_0 = \int d\vec{\Omega}$ — integrated region of allowed phase space, N_Σ — total number of events used for calculation of normalization integrals. In the above formulas i — refers to term number within decay rate, j — refers to event number used for calculation normalization sum.

Then p.d.f is as following:

$$p.d.f = \frac{N_\Sigma}{\Omega_0} \frac{F(\vec{\Omega}, \vec{\alpha})}{\sum_{i=0}^{N_{terms}} \left[\sum_{j=0}^{N_\Sigma} f_i(\vec{\Omega}_j) \right] \cdot g_i(\vec{\alpha})} = \frac{N_\Sigma}{\Omega_0} \frac{F(\vec{\Omega}, \vec{\alpha})}{\sum_{i=0}^{N_{terms}} \tilde{f}_i \cdot g_i(\vec{\alpha})} \quad (118)$$

where $\tilde{f}_i = \left[\sum_{j=0}^{N_\Sigma} f_i(\vec{\Omega}_j) \right]$ — normalization integrals.

Likelihood function in general form can be written as:

$$L = \prod_{e=0}^{N_e} p.d.f(\vec{\Omega}_e | \vec{\alpha}) = \left[\frac{N_\Sigma}{\Omega_0} \right]^{N_e} \prod_{e=0}^{N_e} \frac{F(\vec{\Omega}_e, \vec{\alpha})}{\sum_{i=0}^{N_{terms}} \tilde{f}_i \cdot g_i(\vec{\alpha})} \quad (119)$$

where N_e — number of events used for estimating $\vec{\alpha}$, $\vec{\Omega}_e$ — point in the 5D phase space. After logarithmization:

$$-2 \log \mathcal{L} = -2 N_e \log \frac{N_\Sigma}{\Omega_0} - 2 \sum_{e=0}^{N_e} \log F(\vec{\Omega}_e, \vec{\alpha}) + 2 N_e \log \left[\sum_{i=0}^{N_{terms}} \tilde{f}_i \cdot g_i(\vec{\alpha}) \right] \quad (120)$$

The first term of Eq. 120 is constant with respect to $\vec{\alpha}$ and can be removed:

$$obj.func = -2 \sum_{e=0}^{N_e} \log F(\vec{\Omega}_e, \vec{\alpha}) + 2 N_e \log \left[\sum_{i=0}^{N_{terms}} \tilde{f}_i \cdot g_i(\vec{\alpha}) \right] \quad (121)$$

This function contains two terms:

- $-2 \sum_{e=0}^{N_e} \log F(\vec{\Omega}_e, \alpha)$ — computation of decay rate (see Eq. 114);
- $2 N_e \log \left[\sum_{i=0}^{N_{terms}} \tilde{f}_i \cdot g_i(\alpha) \right]$ — normalization part, which is computed by other means. In particular N_{terms} , \tilde{f}_i , $g_i(\alpha)$ should be known.

3.1. Fitting

Fitter works in the two regimes:

- Caching — stage, where normalization integrals \tilde{f}_i are calculated and correspondings of these integrals and couplings $g_i(\vec{\alpha})$ are created (done by creating dynamic library). In this stage fitter produces:
 - .gar files, which specify symbolic correspondance between kinematical terms and model terms within decay rate (see Eqn. 115);
 - .cxx files — represent the source code of implemented \tilde{f}_i and α_i ;
 - .so files — dynamic libraries, which intended to perform fast calls of precompiled function, which recalculates $g_i(\vec{\alpha})$ terms, for each step of minimization algorithm;
 - .txt — file with normalization integrals \tilde{f}_i ;

For example, user code for caching step looks as following:

```
1 #include "GFitter.h"
2
3 using namespace Gamapola;
4
5 int main()
6 {
7     auto f = std::make_shared<GFitter>(0); // create fitter object
8     f->GGenerateNormalizationIntegrals("norm_neutral.txt"); // calculate normalization integrals
9
10    return 0;
11 }
```

Listing 8. User code of caching step

- Fitting — using precompiled libraries, normalization integrals obtained on caching step, perform objective function minimization (gradient descent fully relies on ROOT minimizer).

Fitter relies on following classes:

- *GInterfaceForMinimization* — sets function, variabes and parameters which will be used as input for fitter core.
- *GEventsAnalyzer* — the core of fitting part, uses minuit minimizer to perform gradient descent, calculates normalization integrals;
- *GFitter* — interface between user and fitter engine, which defines functionality for setting up the function for optimization, model couplings and variables;
- *GSymbolicExpressionsAnalyzer* — responsible for defining the process of \tilde{f}_i calculation;
- *GExCompiler* — responsible for creation, compilation and linking of .cxx, .so, .gar files;
- *GSymbolicMathFunctions* — responsible for defyning the structure of symbolic decay rate;
- *GFitterParser* — utility class used for parsing .fit files and extraction the specification of model parameters to estimate;
- .fit files — provide parameter's description and specification. Represent key, tuple pairs, where key — name of parameter (the same as in .ge file), tuple — collection of objects of different types such as *initial value*, *type*, *low limit*, *up limit* and *minimization step*;

Example of user code for fitting step is given below:

```
1 #include "GFitter.h"
2
3 using namespace Gamapola;
4
5 int main()
6 {
7     auto f = std::make_shared<GFitter>(0); // create fitter object with neutral decay mode
8     f->GSetModelParameters("../additionalFiles/model.fit"); // specify configuration file with model
9     parameters
10    f->GSetNormalizationIntegrals("norm_neutral_5k.txt"); // specify normalization integrals obtained on
11    caching step
12    f->GSetData("neutral.root", "1 < sKpipi < 1.8"); // set data and cut where to fit
13    f->GFit(true); // fit the data and perform goodness of fit
14    return 0;
15 }
```

Listing 9. User code of fitting step

Goodness of fit is performed using Kolmogorov-Smirnov unbinned test. Using above mentioned script as a core one can perform sensitivity studies. The model used for generation of events for all cases contains: $K_1^{1270}, K_1^{1400}, K^{*1410}, K^{*1680}, K_2^{1430}$.

Fitting stage produces table in the *.csv* file which has the following structure:

Min val	fit status	GOF results	<i>pars</i>	<i>parerrs</i>
---------	------------	-------------	-------------	----------------

Table 2. Output format of the fitter