

Outline

- Quasipotential approximation
 - Partial-wave expansion
 - Fixed parity
 - Transformation to a matrix equation
 - Code
 - Physical observable
 - The cross section for the channel considered
 - Argand plot
- Three body decay
 - Kinematics
 - Lorentz boost
 - Amplitude
 - Decay width
 - The case with one rescattering
 - Phase space
 - Differential decay width
 - The case with more than one rescattering
- qBSE package
 - Data Structures for the Dimensions
 - structChannel
 - structIndependentHelicity
 - structDimension
 - Data Structures for the Interactions
 - structSys
 - structInterAction
 - Additional Data Structures
 - structMomentum
 - structHelicity
 - structParticle
 - Functions for the qBSE
 - preprocessing
 - res
 - fV
 - propFF
 - simpleXsection
 - lambda
 - Decay
 - setTGA
 - TGA
 - Vertex14
 - Auxiliary function
 - Additional functions
 - ch
 - LorentzBoostRotation

Quasipotential approximation

The general form of the Bethe-Salpeter equation (BSE) for the scattering amplitude can be written as follows:

$$\mathcal{M}(k'_1 k'_2, k_1 k_2; P) = \mathcal{V}(k'_1 k'_2, k_1 k_2; P) + \int \frac{d^4 k''_2}{(2\pi)^4} \mathcal{V}(k'_1 k'_2, k''_1 k''_2; P) G(k''_1 k''_2; P) \mathcal{M}(k''_1 k''_2, k_1 k_2; P), \quad (1)$$

where \mathcal{V} is the potential kernel and G is the propagator for the two constituent particles. The total momentum of the system is denoted by $P = k_1 + k_2 = k'_1 + k'_2 = k''_1 + k''_2$.

The Bethe-Salpeter equation can be succinctly expressed as

$$\mathcal{M} = \mathcal{V} + \mathcal{V} G \mathcal{M},$$

The Bethe-Salpeter (BS) equation for the amputated scattering matrix with external legs, denoted as $\mathcal{M}_{[\mu_f][\mu_i]}$, is given by:

$$\mathcal{M}_{[\mu_f][\mu_i]} = \mathcal{V}_{[\mu_f][\mu_i]} + \mathcal{V}_{[\mu_f][\mu']}] G^{[\mu'][\mu]} \mathcal{M}_{[\mu][\mu_i]}, \quad (2)$$

where the propagator for the two constituent particles is given by

$$G^{[\mu'][\mu]} = G_1^{[\mu'_1][\mu_1]} G_2^{[\mu'_2][\mu_2]} = \frac{-P^{[\mu'][\mu]}}{(k_1^2 - m_1^2)(k_2^2 - m_2^2)} = P^{[\mu'][\mu]} \tilde{G}_0, \quad (3)$$

where $P^{[\mu'][\mu]}$ is a general tensor structure. For example, for two vector mesons, $P^{[\mu'][\mu]} = (-g^{\mu'_1 \mu_1} + k_1^{\mu'_1} k_1^{\mu_1} / m_1^2)(-g^{\mu'_2 \mu_2} + k_2^{\mu'_2} k_2^{\mu_2} / m_2^2)$; for two spin-1/2 baryons, $P^{[\mu'][\mu]} = (\gamma \cdot k_1 + m_1)(\gamma \cdot k_2 + m_2)$.

In general, such propagators are difficult to handle because the potential \mathcal{V} and amplitude \mathcal{M} cannot be factorized. However, since the dominant contribution comes from the region where both constituents are near their mass shells, a form factor or cutoff is usually introduced to restrict the propagator to the near on-shell region. Therefore, it is reasonable to approximate $P^{[\mu'][\mu]}$ by its on-shell value, which can be expressed as a sum over polarization vectors or spinors: $P^{[\mu'][\mu]} \approx \sum_{\lambda_1, \lambda_2} A_{\lambda_1}^{[\mu'_1]} A_{\lambda_1}^{[\mu_1]} A_{\lambda_2}^{[\mu'_2]} A_{\lambda_2}^{[\mu_2]}$, where A denotes the polarization vector or spinor.

The Gross form of proposed quasipotential propagators for particles 1 and 2 with mass m_1 and m_2 written down in the center of mass frame where $P = (W, \mathbf{0})$ with particle 2 being on shell are

$$\tilde{G}_0 = \frac{1}{(k_1^2 - m_1^2)(k_2^2 - m_2^2)} \rightarrow g = 2\pi i \frac{\delta^+(k_2^2 - m_2^2)}{k_1^2 - m_1^2} = 2\pi i \frac{\delta(k_2^0 - E_2)}{2E_2[(W - E_2)^2 - E_1^2]}, \quad (4)$$

where $k_1 = (k_1^0, \mathbf{k}) = (E_1, \mathbf{k})$, $k_2 = (k_2^0, -\mathbf{k}) = (W - E_1, -\mathbf{k})$ with $E_1 = \sqrt{m_1^2 + |\mathbf{k}|^2}$.

With the define of $G_0 = g/(2\pi i)$, the four-dimensional BSE can be reduced to a three-dimensional equation in center of mass frame

$$i\mathcal{M}(\mathbf{k}', \mathbf{k}) = i\mathcal{V}(\mathbf{k}', \mathbf{k}) + \int \frac{d\mathbf{k}''}{(2\pi)^3} i\mathcal{V}(\mathbf{k}', \mathbf{k}'') G_0(\mathbf{k}'') i\mathcal{M}(\mathbf{k}'', \mathbf{k}), \quad (5)$$

Note: the $i\mathcal{M}$ and $i\mathcal{V}$ are usually real. In the center of mass frame. We choose $\mathbf{k}_2 = \mathbf{k}$ and $\mathbf{k}_1 = -\mathbf{k}$.

Partial-wave expansion

To reduce the equation to one-dimensional equation, we apply the partial wave expansion,

$$\begin{aligned}
\mathcal{V}_{\lambda'\lambda}(\mathbf{k}', \mathbf{k}) &\equiv \langle \theta' \phi', \lambda' | \mathcal{V} | \theta \phi, \lambda \rangle = \sum_{J' M' J M} \langle \theta' \phi' | J' M', \lambda' \rangle \langle J' M' | \mathcal{V} | J M, \lambda \rangle \langle J M, \lambda | \theta \phi, \lambda \rangle \\
&= \sum_{J M} N_J^2 D_{M\lambda'}^{J*}(\phi', \theta', 0) \mathcal{V}_{\lambda'\lambda}^{J M}(\mathbf{k}', \mathbf{k}) D_{M\lambda}^J(\phi, \theta, 0) = \sum_J N_J^2 d_{\lambda\lambda'}^J(\theta_{k,k'}) \mathcal{V}_{\lambda'\lambda}^J(\mathbf{k}', \mathbf{k}). \\
\mathcal{V}_{\lambda'\lambda}^{J M}(\mathbf{k}', \mathbf{k}) &= N_J^2 \int d\Omega' d\Omega D_{M,\lambda'}^J(\phi', \theta', 0) \mathcal{V}_{\lambda'\lambda}(\mathbf{k}', \mathbf{k}) D_{M,\lambda}^{J*}(\phi, \theta, 0).
\end{aligned} \tag{6}$$

where $N_J = \sqrt{\frac{2J+1}{4\pi}}$, $\int d\Omega D_{\lambda_1, \lambda_2}^{J*}(\phi, \theta, 0) D_{\lambda_1', \lambda_2'}^{J'}(\phi, \theta, 0) = N_J^{-2}$.

To calculate $\mathcal{V}_{\lambda'\lambda}^{J M}(\mathbf{k}', \mathbf{k})$, we adopt a special CMS frame. The momenta are chosen as $k_2 = (E_2, 0, 0, \mathbf{k})$, $k_1 = (W - E_2, 0, 0, -\mathbf{k})$ and $k'_2 = (E'_2, \mathbf{k}' \sin \theta_{k,k'}, 0, \mathbf{k}' \cos \theta_{k,k'})$, $k = (W - E_2, -\mathbf{k}' \sin \theta_{k,k'}, 0, -\mathbf{k}' \cos \theta_{k,k'})$ with $\mathbf{k} = |\mathbf{k}|$ and $\mathbf{k}' = |\mathbf{k}'|$.

$$\begin{aligned}
\rightarrow \mathcal{V}_{\lambda'\lambda}(\mathbf{k}', \mathbf{k}) &= \sum_{J M} N_J^2 D_{M\lambda'}^{J*}(0, \theta_{k',k}, 0) \mathcal{V}_{\lambda'\lambda}^{J M}(\mathbf{k}', \mathbf{k}) D_{M\lambda}^J(0, 0, 0). \\
&= \sum_{J M} N_J^2 d_{M\lambda'}^J(\theta_{k',k}) \mathcal{V}_{\lambda'\lambda}^{J M}(\mathbf{k}', \mathbf{k}) \delta_{M\lambda} = \sum_J N_J^2 d_{\lambda\lambda'}^J(\theta_{k',k}) \mathcal{V}_{\lambda'\lambda}^J(\mathbf{k}', \mathbf{k}) \\
\mathcal{V}_{\lambda'\lambda}^J(\mathbf{k}', \mathbf{k}) &= 2\pi \int d\cos \theta_{k,k'} d_{\lambda\lambda'}^J(\theta_{k',k}) \mathcal{V}_{\lambda'\lambda}(\mathbf{k}', \mathbf{k}).
\end{aligned} \tag{7}$$

where $\int_{-1}^1 d\cos \theta' d_{\lambda\lambda'}^{J'}(\theta') d_{\lambda\lambda'}^J(\theta') = N_J^{-2} / 2\pi \delta_{JJ'}$ are used.

NOTE: Which particle is chosen to parallel to z axis is related to the order of λ and λ' in $d_{\lambda'\lambda}^J(\theta_{k,k'})$, so it can not be chosen arbitrarily. And the definition of helicity is also dependent of the definition of $\mathbf{k}_{1,2}$. Here, $\lambda = \lambda_2 - \lambda_1$ and $\lambda_1 = -s_1$, $\lambda_2 = s_2$. The scattering amplitudes \mathcal{M} has analogous relations.

Now we have the partial wave BS equation,

$$\begin{aligned}
i\mathcal{M}_{\lambda'\lambda}^J(\mathbf{k}', \mathbf{k}) &= N_J^2 \int d\Omega' d\Omega D_{\lambda_R, \lambda'}^{J*}(\phi', \theta', 0) D_{\lambda_R, \lambda}^J(\phi, \theta, 0) \\
&\cdot \left[i\mathcal{V}_{\lambda'\lambda}(\mathbf{k}', \mathbf{k}) + \int \frac{d\mathbf{k}''}{(2\pi)^3} i\mathcal{V}_{\lambda'\lambda''}(\mathbf{k}', \mathbf{k}'') G_0(\mathbf{k}'') i\mathcal{M}_{\lambda''\lambda}(\mathbf{k}'', \mathbf{k}) \right] \\
&= i\mathcal{V}_{\lambda'\lambda}^J(\mathbf{k}', \mathbf{k}) + \int \frac{\mathbf{k}''^2 d\mathbf{k}''}{(2\pi)^3} i\mathcal{V}_{\lambda'\lambda''}^J(\mathbf{k}', \mathbf{k}'') G_0(\mathbf{k}'') i\mathcal{M}_{\lambda''\lambda}^J(\mathbf{k}'', \mathbf{k})
\end{aligned} \tag{8}$$

Fixed parity

For a helicity state $|J, \lambda\rangle = |J, \lambda_1 \lambda_2\rangle$ fulfill the party property,

$$P|J, \lambda\rangle = P|J, \lambda_1 \lambda_2\rangle = \eta_1 \eta_2 (-1)^{J-s_1-s_2} |J, -\lambda_1 - \lambda_2\rangle \equiv \tilde{\eta} |J - \lambda\rangle \tag{9}$$

The construction of normalized states with parity \pm is now straightforward:

$$\begin{aligned}
|J, \lambda; \pm\rangle &= \frac{1}{\sqrt{2}} (|J, +\lambda\rangle \pm \tilde{\eta} |J, -\lambda\rangle) \\
\Rightarrow P|J, \lambda; \pm\rangle &= \frac{1}{\sqrt{2}} (\tilde{\eta} |J, -\lambda\rangle \pm |J, +\lambda\rangle) = \pm \frac{1}{\sqrt{2}} (\pm \tilde{\eta} |J, -\lambda\rangle + |J, +\lambda\rangle) = \pm |J, \lambda; \pm\rangle
\end{aligned} \tag{10}$$

$$\mathcal{M}_{-\lambda'-\lambda}^J = \tilde{\eta} \tilde{\eta}' \mathcal{M}_{\lambda'\lambda}^J \tag{11}$$

$$\mathcal{M}_{\lambda'\lambda}^{J\pm} = \langle J, \lambda'; \pm | \mathcal{M} | J, \lambda; \pm \rangle = \mathcal{M}_{\lambda'\lambda}^J \pm \tilde{\eta} \mathcal{M}_{\lambda'-\lambda}^J = \mathcal{M}_{\lambda'\lambda}^J \pm \tilde{\eta}' \mathcal{M}_{-\lambda'\lambda}^J, \quad (12)$$

$$\mathcal{M}_{\lambda'-\lambda}^{J\pm} = \pm \tilde{\eta} \mathcal{M}_{\lambda'\lambda}^{J\pm} \equiv \eta \mathcal{M}_{\lambda'\lambda}^{J\pm}, \quad \mathcal{M}_{-\lambda'\lambda}^{J\pm} = \pm \tilde{\eta}' \mathcal{M}_{\lambda'\lambda}^{J\pm} \equiv \eta' \mathcal{M}_{\lambda'\lambda}^{J\pm} \quad (13)$$

with $\eta = PP_1P_2(-1)^{J_1+J_2-J} = P(-1)^{1/2+J}$.

The potential $\mathcal{V}_{\lambda'\lambda}^{J^P}$ has analogous relations.

$$\begin{aligned} i\mathcal{M}_{\lambda\lambda'}^J &= i\mathcal{V}_{\lambda\lambda'}^J + \sum_{\lambda''} i\mathcal{V}_{\lambda\lambda''}^J G_0 i\mathcal{M}_{\lambda''\lambda'}^J, \quad \eta' i\mathcal{M}_{\lambda-\lambda'}^J = \eta' i\mathcal{V}_{\lambda-\lambda'}^J + \sum_{\lambda''} i\mathcal{V}_{\lambda\lambda''}^J G_0 \eta' i\mathcal{M}_{\lambda''-\lambda'}^J \\ \Rightarrow i\mathcal{M}_{\lambda\lambda'}^{J^P} &= i\mathcal{V}_{\lambda\lambda'}^{J^P} + \sum_{\lambda''} i\mathcal{V}_{\lambda\lambda''}^J G_0 i\mathcal{M}_{\lambda''\lambda'}^{J^P}, \\ i\mathcal{M}_{\lambda\lambda'}^{J^P} &= i\mathcal{V}_{\lambda\lambda'}^{J^P} + \sum_{\lambda''} i\mathcal{V}_{\lambda-\lambda''}^J G_0 i\mathcal{M}_{-\lambda''\lambda'}^{J^P} = i\mathcal{V}_{\lambda\lambda'}^{J^P} + \sum_{\lambda''} i\mathcal{V}_{\lambda-\lambda''}^J G_0 \eta'' i\mathcal{M}_{-\lambda''\lambda'}^{J^P} \\ \Rightarrow i\mathcal{M}_{\lambda\lambda'}^{J^P} &= i\mathcal{V}_{\lambda\lambda'}^{J^P} + \frac{1}{2} \sum_{\lambda''} i\mathcal{V}_{\lambda\lambda''}^{J^P} G_0 i\mathcal{M}_{\lambda''\lambda'}^{J^P}, \end{aligned} \quad (14)$$

As shown in Eq. (14), the amplitudes are not independent. If we only keep the independent amplitudes, the equation for definite parity can be written as

$$\begin{aligned} i\mathcal{M}_{ij}^{J^P} &= i\mathcal{V}_{ij}^{J^P} + \frac{1}{2} \sum_{\lambda''} i\mathcal{V}_{i\lambda''}^{J^P} G_0 i\mathcal{M}_{\lambda''j}^{J^P} = i\mathcal{V}_{ij}^{J^P} + \frac{1}{2} i\mathcal{V}_{i0}^{J^P} G_0 i\mathcal{M}_{0j}^{J^P} + \sum_{k \neq 0} i\mathcal{V}_{ik}^{J^P} G_0 i\mathcal{M}_{kj}^{J^P}, \\ \Rightarrow i\mathcal{M}_{ij}^{J^P} &= i\mathcal{V}_{ij}^{J^P} + \sum_k i\mathcal{V}_{ik}^{J^P} G_0 i\mathcal{M}_{kj}^{J^P}. \end{aligned} \quad (15)$$

where i, j, k are the indices for the independent amplitudes. And we redefine

$$f_i f_j \mathcal{M}_{\lambda'\lambda}^{J^P} \rightarrow \mathcal{M}_{ij}^{J^P}$$

with $f_0 = \frac{1}{\sqrt{2}}$ and $f_{i \neq 0} = 1$ with 0 for the amplitudes with $\lambda_1 = \lambda_2 = 0$.

The Bethe-Salpeter equation for partial-wave amplitude with fixed spin-parity J^P reads ,

$$i\mathcal{M}_{\lambda'\lambda}^{J^P}(\mathbf{k}', \mathbf{k}) = i\mathcal{V}_{\lambda'\lambda}^{J^P}(\mathbf{k}', \mathbf{k}) + \sum_{\lambda''} \int \frac{k''^2 dk''}{(2\pi)^3} i\mathcal{V}_{\lambda'\lambda''}^{J^P}(\mathbf{k}', \mathbf{k}'') G_0(k'') i\mathcal{M}_{\lambda''\lambda}^{J^P}(\mathbf{k}'', \mathbf{k}). \quad (16)$$

Note that here the sum extends only over indices for the independent amplitudes. The partial wave potential is defined as

$$i\mathcal{V}_{\lambda'\lambda}^{J^P}(\mathbf{k}', \mathbf{k}) = 2\pi \int d\cos\theta [d_{\lambda\lambda'}^J(\theta) i\mathcal{V}_{\lambda'\lambda}^J(\mathbf{k}', \mathbf{k}) + \eta d_{-\lambda\lambda'}^J(\theta) i\mathcal{V}_{\lambda'-\lambda}^J(\mathbf{k}', \mathbf{k})], \quad (17)$$

Or, with the independent amplitudes as

$$i\mathcal{M}_{ij}^{J^P}(\mathbf{k}', \mathbf{k}) = i\mathcal{V}_{ij}^{J^P}(\mathbf{k}', \mathbf{k}) + \sum_k \int \frac{k''^2 dk''}{(2\pi)^3} i\mathcal{V}_{ik}^{J^P}(\mathbf{k}', \mathbf{k}'') G_0(k'') i\mathcal{M}_{kj}^{J^P}(\mathbf{k}'', \mathbf{k}). \quad (18)$$

$$i\mathcal{V}_{ij}^{J^P}(\mathbf{k}', \mathbf{k}'') = f_i f_j 2\pi \int d\cos\theta [d_{\lambda\lambda'}^J(\theta) i\mathcal{V}_{\lambda'\lambda}^J(\mathbf{k}', \mathbf{k}) + \eta d_{-\lambda\lambda'}^J(\theta) i\mathcal{V}_{\lambda'-\lambda}^J(\mathbf{k}', \mathbf{k})]. \quad (19)$$

Note here the $f_i f_j$ is also incorporated. Additionally, the form factors for the interacting particles are also included in the potential, modifying it as $\mathcal{V} \rightarrow f(k') \mathcal{V} f(k)$. Consequently, the resulting amplitude \mathcal{M} also includes these form factors.

The potential $\mathcal{V}_{\lambda'-\lambda}(\mathbf{k}', \mathbf{k})$ is introduced by fv function in main file as

fV(k, l, SYS, IA0, CHf, CHi)

where k and l are for the momenta and helicities of final and initial particles. SYS is for the system information. $IA0$ is for the interaction information., CHf and CHi are for the information of final and initial channels.

Transition of $\mathcal{V}_{\lambda'-\lambda}(\mathbf{k}', \mathbf{k})$ to $\mathcal{V}_{ij}^{JP}(\mathbf{k}', \mathbf{k}'')$ performed in `qBSE.fKernel` which is an internal function.

Transformation to a matrix equation

Now We have a integral equation with singularity in $G_0 = \frac{1}{2E((s-E)^2 - \omega^2)}$ at $W = E_1 + E_2$. This singularity can be isolated as,

$$\begin{aligned} i\mathcal{M}^{JP}(\mathbf{k}', \mathbf{k}) &= i\mathcal{V}^{JP}(\mathbf{k}, \mathbf{k}') + \int \frac{k''^2 dk''}{(2\pi)^3} i\mathcal{V}^{JP}(\mathbf{k}, \mathbf{k}'') G_0(k'') i\mathcal{M}^{JP}(\mathbf{k}'', \mathbf{k}') \\ &= i\mathcal{V}^{JP} + \mathcal{P} \int \frac{k''^2 dk''}{(2\pi)^3} i\mathcal{V}^{JP} G_0 \mathcal{M}^{JP} - \pi i \int \frac{k''^2 dk''}{(2\pi)^3} \mathcal{V}_o^{JP} \delta G_0 \mathcal{M}_o^{JP} \\ &= i\mathcal{V}^{JP} + \int \frac{dk''}{(2\pi)^3} \left[k''^2 G_0 i\mathcal{V}^{JP} \mathcal{M}^{JP} - \frac{M \mathcal{V}_o^{JP} \mathcal{M}_o^{JP}}{k''^2 - \bar{k}''^2} \right] - i \frac{\bar{k}''^2 \delta \bar{G}_0}{8\pi^2} \mathcal{V}_o^{JP} \mathcal{M}_o^{JP} \\ &= i\mathcal{V}^{JP} + \int \frac{dk''}{(2\pi)^3} k''^2 G_0 i\mathcal{V}^{JP} \mathcal{M}^{JP} - \left[\int \frac{dk''}{(2\pi)^3} \frac{M}{k''^2 - \bar{k}''^2} + i \frac{\bar{k}''^2 \delta \bar{G}_0}{8\pi^2} \right] \mathcal{V}_o^{JP} \mathcal{M}_o^{JP} \end{aligned}$$

with $\delta G_0 = \frac{\delta(s-E-\omega)}{2E(s-E+\omega)} \theta(s - m_1 - m_2) = \delta \bar{G}_0 \delta(k'' - \bar{k}'') = \frac{1}{4Wk''} \delta(k'' - \bar{k}'') \theta(s - m_1 - m_2)$, $M = [k''^2(k''^2 - \bar{k}''^2) G_0]_{k'' \rightarrow \bar{k}''} \theta(s - m_1 - m_2) = -\frac{\bar{k}''^2}{2W} \theta(s - m_1 - m_2)$.

We have

$$Im G = -\rho/2 = -\frac{\bar{k}''}{32\pi^2 W}. \quad (20)$$

It suggests the unitary is satisfied if the potential $i\mathcal{V}$ is real.

$$-T^\dagger \rho T = 2T^\dagger Im G T = 2T^\dagger (-Im T^{-1}) T = 2T^\dagger \frac{1}{2i} (T^{\dagger-1} - T^{-1}) T = i(T - T^\dagger) \quad (21)$$

where $T = i\mathcal{M}$.

With the Gauss discretization, the one-dimensional equation can be transformed as a matrix equation as

$$i\mathcal{M}_{ik}^{JP} = i\mathcal{V}_{ik}^{JP} + \sum_{j=0}^N i\mathcal{V}_{ij}^{JP} G_j i\mathcal{M}_{jk}^{JP} \Rightarrow M^{JP} = V^{JP} + V^{JP} G M^{JP} \quad (22)$$

$$G_j = \begin{cases} -\frac{i\bar{q}}{32\pi^2 W} + \sum_j \left[\frac{w(q_j)}{(2\pi)^3} \frac{\bar{q}^2}{2W(q_j^2 - \bar{q}^2)} \right] & \text{for } j = 0, \text{ if } Re(W) > m_1 + m_2, \\ \frac{w(q_j)}{(2\pi)^3} \frac{q_j^2}{2E(q_j)[(W-E(q_j))^2 - \omega^2(q_j)]} & \text{for } j \neq 0 \end{cases}$$

where $\bar{q} = \frac{1}{2W} \sqrt{[W^2 - (m_1 + m_2)^2][W^2 - (m_1 - m_2)^2]}$. The indices i, j, k is for discrete momentum values, independent helicities, and coupled channels.

The propagator is calculated in `qBSE.fProp` which is an internal function.

The default dimension is $[G] = 1$. Recalling that a factor of $2m$ should be included if a constituent particle is a fermion, we have $[G] = \text{GeV}^{n_f} \rightarrow [V] = [M] = \text{GeV}^{-n_f}$, with n_f being the number of fermions. Therefore, under our convention where $\bar{u}u = 1$, the dimension of the potential must satisfy the above requirements. This criterion can be employed to verify the consistency of the Lagrangian and the derived potential.

Hence, for the channels above its thresholds, the matrix have an extra dimension.

We take two channel as example to explain the coupled-channel equation. The region of W is divided as

$W < m_1$, $m_1 < W < m_2$ and $W > m_2$.

$$V = \begin{pmatrix} V_{11}^{NN} & V_{12}^{NN} \\ V_{21}^{NN} & V_{22}^{NN} \end{pmatrix}, \quad G = \begin{pmatrix} G_1^N & 0 \\ 0 & G_2^N \end{pmatrix}, \quad W < m_1, \quad (23)$$

$$V = \begin{pmatrix} V_{11}^{N+1N+1} & V_{12}^{N+1N} \\ V_{21}^{NN+1} & V_{22}^{NN} \end{pmatrix}, \quad G = \begin{pmatrix} G_1^{N+1} & 0 \\ 0 & G_2^N \end{pmatrix}, \quad m_1 < W < m_2 \quad (24)$$

$$V = \begin{pmatrix} V_{11}^{N+1N+1} & V_{12}^{N+1N+1} \\ V_{21}^{N+1N+1} & V_{22}^{N+1N+1} \end{pmatrix}, \quad G = \begin{pmatrix} G_1^{N+1} & 0 \\ 0 & G_2^{N+1} \end{pmatrix}, \quad W > m_2 \quad (25)$$

The informations about the dimensions are calculated in `qBSE.WORKSPACE`. The matrix V and G are calculated in `qBSE.srAB`. Note that such function is internal function and not used by the user.

Code

Attention: The following details are specific to the old version of the code, which includes Fortran code and Julia code versions prior to v0.2.4. In the new version, the treatment described below is obsolete.

In old code, we choose $\hat{V}^{J^P} = f(k')V^{J^P}f(k)/4\pi$, $\hat{G} = 4\pi G$, and $\hat{M}^{J^P} = f(k')M^{J^P}f(k)/4\pi$. The form factors are also included in to the amplitudes and the potential kernel. Hence, the qBSE becomes

$$\hat{M}^{J^P} = \hat{V}^{J^P} + \hat{V}^{J^P} \hat{G} \hat{M}^{J^P}. \quad (26)$$

Such convention is consistent with that in the chiral unitary approach.

$$\begin{aligned} \hat{V}^{J^P} &= V^{J^P}/4\pi = i\mathcal{V}_{\lambda'\lambda''}^{J^P}(\mathbf{p}', \mathbf{p}'')/4\pi = f_i f_j i\mathcal{V}_{\lambda'\lambda}^{J^P}(\mathbf{p}', \mathbf{p})/4\pi \\ &= \frac{1}{2} f_i f_j \int d\cos\theta [d_{\lambda\lambda'}^J(\theta) i\mathcal{V}_{\lambda'\lambda}(\mathbf{p}', \mathbf{p}) + \eta d_{-\lambda\lambda'}^J(\theta) i\mathcal{V}_{\lambda'-\lambda}(\mathbf{p}', \mathbf{p})], \end{aligned} \quad (27)$$

$$\hat{G}_j = \begin{cases} -\frac{i\bar{q}}{8\pi W} + \sum_j \left[\frac{w(q_j)}{2\pi^2} \frac{\bar{q}^2}{2W(q_j^2 - \bar{q}^2)} \right] & \text{for } j = 0, \text{ if } \text{Re}(W) > m_1 + m_2, \\ \frac{w(q_j)}{2\pi^2} \frac{q_j^2}{2E(q_j)[(W - E(q_j))^2 - \omega^2(q_j)]} & \text{for } j \neq 0 \end{cases}$$

Physical observable

After extend the energy in the center of mass frame W into complex energy plane as z , the pole can be found by variation of z to satisfy

$$|1 - V(z)G(z)| = 0 \quad (28)$$

with $z = E_R + i\Gamma_R/2$.

The $|1 - V(z)G(z)|$ in complex enery plane is calculated in `qBSE.res` and the pole can be found with function `qBSE.showPoleInfo`.

With the obtained amplitude M^{J^P} , we can also calculate the physical observable. Note that all physical observable are at real axis, we choose the onshell momentum as

$$M_{ij}(z) = \{[(1 - VG)^{-1}]V\}_{ij} \quad (29)$$

with i and k chosen as the onshell momentum, that is, 0 dimension for G , and extra dimension for V .

The $|M|^2$ for each channel is calculated in `qBSE.res`.

The cross section for the channel considered

For the open channel, the cross section can be obtained as

$$\frac{d\sigma}{d\Omega} = \frac{1}{\tilde{j}_1 \tilde{j}_2} \frac{1}{64\pi^2 s} \frac{k'}{k} \sum_{\lambda\lambda'} |i\mathcal{M}_{\lambda\lambda'}(\mathbf{k}', \mathbf{k})|^2. \quad (30)$$

where j_1 and j_2 is the spin of the initial particles, and we define $\tilde{j} = 2j + 1$.

The total cross section can be written as

$$\sigma = \frac{1}{\tilde{j}_1 \tilde{j}_2} \frac{1}{64\pi^2 s} \frac{k'}{k} \sum_{J,\lambda} N_J^2 |i\mathcal{M}_{\lambda\lambda'}^J(\mathbf{k}', \mathbf{k})|^2 = \frac{1}{\tilde{j}_1 \tilde{j}_2} \frac{1}{64\pi^2 s} \frac{k'}{k} \sum_{J^P, ij} N_J^2 |M_{ij}^{J^P}|^2. \quad (31)$$

Here, \mathbf{k}' and \mathbf{k} are onshell momenta, so we only choose ij for the onshell momenta. Since we adopt $\mathcal{V} \rightarrow f(k')\mathcal{V}f(k)$, the amplitudes $M^{J^P} \rightarrow f(k')M^{J^P}f(k)$. The form factors vanish due to onshellness for initial and final states of a scattering.

NOTE: $4MM'$ should be multiplied due to convention $\bar{u}u = 1$ adopted in code.

$$\begin{aligned} \sigma &\propto \sum_{J,\lambda'\lambda} |M_{\lambda'\lambda}^J|^2 = \sum_{J,\lambda'j=0} |M_{\lambda'j}^J|^2 + \sum_{J,\lambda'j>0} [|M_{\lambda'j}^J|^2 + |M_{\lambda'-j}^J|^2] \\ &= \sum_{J^P, \lambda'j=0} \delta_{\eta,1} |\frac{1}{2}M_{\lambda'j}^{J^P}|^2 + \sum_{J,\lambda'j>0} \left[\frac{1}{4}|M_{\lambda'j}^{J^+} + M_{\lambda'j}^{J^-}|^2 + \frac{1}{4}|M_{\lambda'j}^{J^+} - M_{\lambda'j}^{J^-}|^2 \right] \\ &= \sum_{J^P, i=0j=0} \delta_{\eta,1} \delta_{\eta',1} |\frac{1}{2}M_{ij}^{J^P}|^2 + \sum_{J^P, i>0j=0} 2\delta_{\eta,1} |\frac{1}{2}M_{ij}^{J^P}|^2 + \sum_{J^P, \lambda'j>0} \frac{1}{2}|M_{\lambda'j}^{J^P}|^2 \\ &= \sum_{J^P, i=0j=0} \delta_{\eta,1} \delta_{\eta',1} |\frac{1}{2}M_{ij}^{J^P}|^2 + \sum_{J^P, i>0j=0} \delta_{\eta,1} |\frac{1}{\sqrt{2}}M_{ij}^{J^P}|^2 + \sum_{J^P, i=0j>0} \delta_{\eta',1} |\frac{1}{\sqrt{2}}M_{ij}^{J^P}|^2 + \sum_{J^P, i>0j>0} |M_{\lambda'j}^{J^P}|^2 \\ &= \sum_{J^P, i=0j=0} \delta_{\eta,1} \delta_{\eta',1} |\frac{1}{2}M_{ij}^{J^P}|^2 + \sum_{J^P, i>0j=0} \delta_{\eta,1} |\frac{1}{\sqrt{2}}M_{ij}^{J^P}|^2 + \sum_{J^P, i=0j>0} \delta_{\eta',1} |\frac{1}{\sqrt{2}}M_{ij}^{J^P}|^2 + \sum_{J^P, i>0j>0} |M_{ij}^{J^P}|^2 \\ &= \sum_{J^P, i\geq 0j\geq 0} |f_i f_j M_{ij}^{J^P}|^2 = \sum_{J^P, i\geq 0j\geq 0} |M_{ij}^{J^P}|^2 \end{aligned} \quad (32)$$

The cross section for certain channel is calculated with `qBSE.simpleXsection`

Argand plot

The amplitudes can be written as

$$i\mathcal{M}(\mathbf{k}', \mathbf{k}) = -8\pi\sqrt{s}f(\mathbf{k}', \mathbf{k}) = -\frac{8\pi\sqrt{s}}{|\mathbf{k}|} \sum_{JM} N_J^2 D_{\lambda_R, \lambda}^{J*}(\phi', \theta', 0) a_{\lambda\lambda'}^J(|\mathbf{k}'|, |\mathbf{k}|) D_{M, \lambda'}^J(\phi, \theta, 0). \quad (33)$$

where $a^J = -\frac{|\mathbf{k}|}{8\pi\sqrt{s}}\mathcal{M}(|\mathbf{k}|)$, which can be displayed as a trajectory in an Argand plot.

Three body decay

kinematics

Lorentz boost

Here, we consider an process $Y \rightarrow m_1 X \rightarrow m_1 m_2 m_3$.

To study a $1 \rightarrow 3$ decay with the qBSE, we need consider the center of mass frame (CMS) of Y (which is also the laboratory frame in this issue) and the $m_1 m_2$ where the qBSE is applied. The momenta of initial and final particles in the CMS of Y , remarked as *lab*, are

$$P^{lab} = (W, 0, 0, 0); \quad p_1^{lab} = (E_1^{lab}, \mathbf{p}_1^{lab}); \quad p_2^{lab} = (E_2^{lab}, \mathbf{p}_2^{lab}); \quad p_3^{lab} = (E_3^{lab}, \mathbf{p}_3^{lab}) \quad (34)$$

The Lorentz boost from $(m, \mathbf{0})$ to (E, \mathbf{k}) ,

$$\Lambda_\nu^\mu = \frac{1}{m} \begin{pmatrix} E(\mathbf{k}) & k_x & k_y & k_z \\ k_x & m + \frac{k_x k_x}{E+m} & \frac{k_x k_y}{E+m} & \frac{k_x k_z}{E+m} \\ k_y & \frac{k_y k_x}{E+m} & m + \frac{k_y k_y}{E+m} & \frac{k_y k_z}{E+m} \\ k_z & \frac{k_z k_x}{E+m} & \frac{k_z k_y}{E+m} & m + \frac{k_z k_z}{E+m} \end{pmatrix}.$$

With the Lorentz boost the momenta for particle 23 in the laboratory frame $(E_{23}^{lab}, -\mathbf{p}_1^{lab})$ can be written with the momenta in the CMS of particles 23 $(M_{23}, \mathbf{0})$ as $p^{lab} = \Lambda(E_{23}^{lab}, -\mathbf{p}_1^{lab})p^{cm}$,

$$\begin{aligned} \mathbf{p}^{lab} &= \mathbf{p}^{cm} - \frac{\mathbf{p}_1^{lab}}{M_{23}} \left[\frac{-\mathbf{p}_1^{lab} \cdot \mathbf{p}^{cm}}{W - E_1^{lab}(\mathbf{p}_1^{lab}) + M_{23}} + p^{0cm} \right], \\ p^{0lab} &= \frac{1}{M_{23}} [(W - E_1^{lab}(\mathbf{p}_1^{lab}))p^{0cm} - \mathbf{p}_1^{lab} \cdot \mathbf{p}^{cm}], \end{aligned} \quad (35)$$

where the $p_{23} + p_1 = P$ is applied, and $M_{23} = \sqrt{(p_2^{lab} + p_3^{lab})^2} = \sqrt{(p_2^{cm} + p_3^{cm})^2} = \sqrt{(P - p_1)^2}$.

The momenta in CMS of 23 can also be written with the momentum in laboratory frame as $p = \Lambda(E_{23}^{lab}, \mathbf{p}_1^{lab})p^{lab}$

$$\begin{aligned} \mathbf{p}^{cm} &= \mathbf{p}^{lab} + \frac{\mathbf{p}_1^{lab}}{M_{23}} \left[\frac{\mathbf{p}_1^{lab} \cdot \mathbf{p}^{lab}}{M - E_1^{lab}(\mathbf{p}_1^{lab}) + M_{23}} + p^{0lab} \right], \\ p^{0cm} &= \frac{1}{M_{23}} [(M - E_1^{lab}(\mathbf{p}_1^{lab}))p^{0lab} + \mathbf{p}_1^{lab} \cdot \mathbf{p}^{lab}]. \end{aligned} \quad (36)$$

For the outgoing particles 2 and 3, the \mathbf{p}^{cm} should be set along the z axis, hence, an additional rotation is adopted as $p = R(-\theta^{cm}, -\phi^{cm})p^{cm}$

$$R(-\theta, -\phi) = \frac{1}{m} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta \cos \phi & \cos \theta \sin \phi & -\sin \theta \\ 0 & -\sin \phi & \cos \phi & 0 \\ 0 & \sin \theta \cos \phi & \sin \theta \sin \phi & \cos \theta \end{pmatrix}.$$

As described above, we perform both a Lorentz boost and a rotation. Through the Lorentz boost, the momentum p^{lab} in the lab frame transforms into p^{cm} in the center-of-mass system (CMS) of particles 2 and 3. The value of p_1^{cm} depends on p_1^{lab} and M_{23} , and the lab-frame energy E_1^{lab} can also be derived from M_{23} . Thus, p_1^{cm} is ultimately dependent only on Ω_1^{lab} and M_{23} . The momenta $p_{2,3}^{cm}$ depend on $p_{2,3}^{lab}$ and Ω_1^{lab} and can be expressed in terms of M_{23} and the spherical angle Ω_3^{cm} .

After rotation, p^{cm} becomes p in the new CMS frame of particles 2 and 3. The momenta $p_{2,3}$ of the final particles are aligned along the z-axis and depend solely on M_{23} . However, for intermediate particles, they are functions of p'_3 and Ω'_3 . Finally, p_1 depends on Ω_1^{lab} , M_{23} , and Ω_3^{cm} .

The Lorentz boost is performed by `Xs.LorentzBoost`, and rotation by `Xs.Rotation`.

A function `qBSE.LorentzBoostRotation` is also provided to make both a Lorentz boost and a rotation.

Amplitude

Because the $|\mathcal{M}|^2$ is invariant in different reference frame, the amplitude for the direct decay can be written with the momenta in cm frame of partilces 1 and 2 obtained with Lorentz boost and rotation, as

$$\begin{aligned}
i\mathcal{M}_{\lambda_1;\lambda_2,\lambda_3;\lambda}^d(p_1, p_2, p_3) &= i\mathcal{A}_{\lambda_1;\lambda_2,\lambda_3;\lambda}(\Lambda p_1, p_2, p_3) = i\mathcal{A}_{\lambda_1;\lambda_2,\lambda_3;\lambda}(\Omega_1^{lab}, M_{23}, \Omega_3^{cm}), \quad \text{for onshell} \\
i\mathcal{M}_{\lambda_1;\lambda_2,\lambda_3;\lambda}^d(p_1, p'_2, p'_3) &= i\mathcal{A}_{\lambda_1;\lambda_2,\lambda_3;\lambda}(\Omega_1^{lab}, \Omega'_3, p'_3, M_{23}, \Omega_3^{cm}) \\
&= \sum_{JM} N_J D_{M\lambda_{32}}^{J*}(\Omega'_3) i\mathcal{A}_{\lambda_1;\lambda_2,\lambda_3;\lambda}^{JM}(\Omega_1^{lab}, p'_3, M_{23}, \Omega_3^{cm}), \quad \text{for offshell} \\
i\mathcal{A}_{\lambda_1;\lambda'_2,\lambda'_3;\lambda}^{JM}(\Omega_1^{lab}, p'_2, M_{23}, \Omega_3^{cm}) &= N_J \int d\Omega'_3 D_{M,\lambda'_{32}}^J(\Omega'_3) i\mathcal{A}_{\lambda_1;\lambda'_2,\lambda'_3;\lambda}(\Omega_1^{lab}, \Omega'_3, p'_3, M_{23}, \Omega_3^{cm}) \\
i\mathcal{M}_{\lambda_1;\lambda_2,\lambda_3;\lambda}^Z(p_1, p_2, p_3) &= i \int \frac{d^4 p'_3}{(2\pi)^4} \mathcal{T}_{\lambda_2,\lambda_3}(p_2, p_3; p'_2, p'_3) G(p'_3) \mathcal{A}_{\lambda_1;\lambda}(p_1, p'_2, p'_3) \\
&= \sum_{\lambda'_2\lambda'_3} \int \frac{d^3 p'_3}{(2\pi)^3} i\mathcal{T}_{\lambda_2,\lambda_3;\lambda'_2,\lambda'_3}(p_2, p_3; p'_2, p'_3) G_0(p'_3) i\mathcal{A}_{\lambda_1;\lambda'_2,\lambda'_3;\lambda}(p_1, p'_2, p'_3) \\
&= \sum_{\lambda'_2\lambda'_3} \int \frac{d^3 p'_3}{(2\pi)^3} i\mathcal{T}_{\lambda_2,\lambda_3;\lambda'_2,\lambda'_3}(\Omega'_3, p'_3, M_{23}) G_0(p'_3) i\mathcal{A}_{\lambda_1;\lambda'_2,\lambda'_3;\lambda}(\Omega_1^{lab}, \Omega'_3, p'_3, M_{23}, \Omega_3^{cm}) \\
&= \sum_{\lambda'_2\lambda'_3} \int \frac{p_3'^2 dp'_3 d\Omega'_3}{(2\pi)^3} \sum_{JM} N_J^2 \delta_{M\lambda_{32}} i\mathcal{T}_{\lambda_2,\lambda_3;\lambda'_2,\lambda'_3}^{JM}(p'_3, M_{23}) D_{M\lambda'_{32}}^J(\Omega'_3) \\
&\quad \cdot G_0(p'_3) \sum_{J'M'} N_{J'} D_{M'\lambda'_{32}}^{J'*}(\Omega'_3) i\mathcal{A}_{\lambda_1;\lambda'_2,\lambda'_3;\lambda}^{J'M'}(\Omega_1^{lab}, p'_3, M_{23}, \Omega_3^{cm}) \\
&= \sum_{\lambda'_2\lambda'_3} \int \frac{p_3'^2 dp'_3 d\Omega'_3}{(2\pi)^3} \sum_J N_J i\mathcal{T}_{\lambda_2,\lambda_3;\lambda'_2,\lambda'_3}^J(p'_3, M_{23}) G_0(p'_3) i\mathcal{A}_{\lambda_1;\lambda'_2,\lambda'_3;\lambda}^{J\lambda_{32}}(\Omega_1^{lab}, p'_3, M_{23}, \Omega_3^{cm}) \\
&= \sum_J N_J \sum_{\lambda'_2\lambda'_3} \int \frac{p_3'^2 dp'_3}{(2\pi)^3} i\mathcal{T}_{\lambda_2,\lambda_3;\lambda'_2,\lambda'_3}^J(p'_3, M_{23}) G_0(p'_3) i\mathcal{A}_{\lambda_1;\lambda'_2,\lambda'_3;\lambda}^{J\lambda_{32}}(\Omega_1^{lab}, p'_3, M_{23}, \Omega_3^{cm}) \\
&\equiv i\mathcal{M}_{\lambda_2,\lambda_3;\lambda}^Z(\Omega_1^{lab}, M_{23}, \Omega_3^{cm}), \\
i\mathcal{M}_{\lambda_2;\lambda_1,\lambda_3;\lambda}^Z(p_1, p_2, p_3) &= \sum_J N_J \sum_{\lambda'_1\lambda'_3} \int \frac{\tilde{p}_3'^2 d\tilde{p}_3'}{(2\pi)^3} i\mathcal{T}_{\lambda_1,\lambda_3;\lambda'_1,\lambda'_3}^J(\tilde{p}_3', M_{13}) G_0(\tilde{p}_3') i\mathcal{A}_{\lambda_2;\lambda'_1,\lambda'_3;\lambda}^{J\lambda_{31}}(\tilde{\Omega}_3^{lab}, \tilde{p}_3', M_{13}, \tilde{\Omega}_3^{cm}) \\
&\equiv i\mathcal{M}_{\lambda_2;\lambda_1,\lambda_3;\lambda}^Z(\tilde{\Omega}_3^{lab}, M_{13}, \tilde{\Omega}_3^{cm}). \tag{37}
\end{aligned}$$

Note: when consider the rescattering of different particles, the different cm frames should be adopted.

Decay width

The case with one rescattering

Phase space

The phase space is given by

$$d\Phi = (2\pi)^4 \delta^4(P - p_1 - p_2 - p_3) \frac{d^3 p_1}{2E_1(2\pi)^3} \frac{d^3 p_2}{2E_2(2\pi)^3} \frac{d^3 p_3}{2E_3(2\pi)^3} \quad (38)$$

To study the invariant mass spectrum of the particles 2 and 3, it is convenient to rewrite the Lorentz-invariant phase space $d\Phi$ by taking as integration variables the direction of the momentum of particle 2 \mathbf{p}_2^{cm} in the center-of-mass (cm) frame of particles 2 and 3. Thus, we first rewrite the phase factor as

$$d\Phi = (2\pi)^4 \delta(E_2^{cm} + E_3^{cm} - W_{23}) \delta^3(\mathbf{p}_2^{cm} + \mathbf{p}_3^{cm}) \frac{d^3 p_1^{lab}}{2E_1^{lab}(2\pi)^3} \frac{d^3 p_2^{cm}}{2E_2^{cm}(2\pi)^3} \frac{d^3 p_3^{cm}}{2E_3^{cm}(2\pi)^3} \quad (39)$$

where $W_{23}^2 = (M - E_1^{lab})^2 - |\mathbf{p}_1^{lab}|^2$. Here the Lorentz invariance of the $\frac{d^3 p}{2E(2\pi)^3}$ and $\delta^4(P - p_1 - p_2 - p_3)$ is used.

Here, we use the momentum of the particle 3 in the center of mass system of two rescattering particles.

The momentum of the particle 3 has a relation $\mathbf{p}_3^{cm} = \frac{1}{2M_{23}} \sqrt{\lambda(M_{23}^2, m_3^2, m_2^2)}$.

Owing to the three-momentum δ function, the integral over \mathbf{p}_2^{cm} can be eliminated. Next, the quantity $d^3 p_3^{cm}$ is converted to dM_{23} by the relation,

$$d^3 p_3^{cm} = \frac{E_2^{cm} E_3^{cm} \mathbf{p}_3^{cm}}{M_{23}} dM_{23} d\Omega_3^{cm}, \quad (40)$$

where $M_{23} (= E_2^{cm} + E_3^{cm})$ is the invariant mass of the 23 system. We would like to integrate over the magnitude of the neutron momentum p_1 , which is related to W_{23} . Hence, the energy-conserving δ function is substituted as,

$$\delta(M_{23} - W_{23}) = \frac{W_{23}}{|M \mathbf{p}_1^{lab} / E_1^{lab}|} \delta(\check{p}_1 - p_1^{lab}) \quad (41)$$

where the \check{p}_1 satisfies $M_{23}^2 = (M - \check{E}_1)^2 - \check{p}_1^2$.

Performing the integral over \mathbf{p}_1^{lab} , we obtain the final expression of the decay width,

$$d\Phi = \frac{1}{(2\pi)^5} \frac{\check{p}_1 \mathbf{p}_3^{cm}}{8M} d\Omega_1^{lab} d\Omega_3^{cm} dM_{23} = \frac{1}{(2\pi)^3} \frac{\check{p}_1 \mathbf{p}_3^{cm}}{8M} d \cos \theta_1^{lab} d \cos \theta_3^{cm} dM_{23}, \quad (42)$$

Here, independence of the ϕ_1^{lab} and ϕ_2^{cm} on the integrand is applied.

Differential decay width

Here, we consider the rescattering of particles 2 and 3. The distribution can be obtained as

$$\frac{d\Gamma}{dM_{23}} = \int \frac{1}{2M} \sum_{\lambda_2, \lambda_3; \lambda} |i\mathcal{M}_{\lambda_2, \lambda_3; \lambda}|^2 \frac{1}{(2\pi)^5} \frac{\check{p}_1 \mathbf{p}_3}{8M} d\Omega_1^{lab} d\Omega_3^{cm} \quad (43)$$

If the Y is not scalar, there should be an additional factor $1/\tilde{j}_Y$. Here we

use the distribution above in the frame with Lorentz boost but without rotation. However,

we calculate the $|i\mathcal{M}_{\lambda_2, \lambda_3; \lambda}|^2$ in the frame with both Lorentz boost and rotation due to it is independent on the frames.

If the reflection effect is absent, for example, the πDD^* final state. The results can be simplified further.

The amplitude can be written as

$$\begin{aligned}
i\mathcal{M}_{\lambda_1\lambda_2,\lambda_3;\lambda}(p_1, p_2, p_3) &= \sum_{JM} N_J^3 D_{\lambda_R\lambda_1}^J D_{\lambda_R\lambda_{32}}^J (\Omega_3^{cm}) \left[i\mathcal{A}_{\lambda_1,\lambda_2,\lambda_3;\lambda}^{d,J}(M_{23}) \right. \\
&\quad \left. + \sum_{\lambda'_2\lambda'_3} \int \frac{p_3'^2}{(2\pi)^3} i\mathcal{T}_{\lambda_2,\lambda_3;\lambda'_2,\lambda'_3}^J(p'_3, M_{23}) G_0(p'_3) i\mathcal{A}_{\lambda'_2,\lambda'_3;\lambda}^J(p'_3, M_{23}) \right] \\
&\equiv \sum_{JM} N_J^3 D_{\lambda_R\lambda_1}^J (\Omega_1^{lab}) D_{\lambda_R\lambda_{32}}^J (\Omega_3^{cm}) i\mathcal{M}_{\lambda_2,\lambda_3;\lambda}^J(M_{23})
\end{aligned} \tag{44}$$

Inserting the above amplitude to the definition of the invariant mass spectrum, we have

$$\begin{aligned}
\frac{d\Gamma}{dM_{23}} &= \frac{1}{2M} \frac{1}{(2\pi)^5} \frac{\check{p}_1 p_3}{8M} \sum_{\lambda_2,\lambda_3;\lambda} \int \left| \sum_{JM} N_J^3 D_{\lambda_R\lambda_1}^J (\Omega_1^{lab}) D_{\lambda_R\lambda_{32}}^J (\Omega_3^{cm}) i\mathcal{M}_{\lambda_2,\lambda_3;\lambda}^J(M_{23}) \right|^2 d\Omega_1^{lab} d\Omega_3 \\
&= \frac{1}{2M} \frac{1}{(2\pi)^5} \frac{\check{p}_1 p_3^{cm}}{8M} \sum_{\lambda_2,\lambda_3;\lambda;J} |N_J i\mathcal{M}_{\lambda_2,\lambda_3;\lambda}^J(M_{23})|^2
\end{aligned}$$

Now we consider the amplitude with fixed parity,

$$\begin{aligned}
i\mathcal{M}_{\lambda_{23};\lambda}^J &= i\mathcal{A}_{\lambda_{23};\lambda}^{d,J} + i\mathcal{T}_{\lambda_{23},\lambda'_{23}}^J G_0 i\mathcal{A}_{\lambda'_{23};\lambda}^J, \quad \eta i\mathcal{M}_{-\lambda_{23};\lambda}^J = \eta i\mathcal{A}_{-\lambda_{23};\lambda}^{d,J} + \eta i\mathcal{T}_{-\lambda_{23},\lambda'_{23}}^J G_0 i\mathcal{A}_{\lambda'_{23};\lambda}^J \\
\Rightarrow i\mathcal{M}_{\lambda_{23};\lambda}^{J^P} &= i\mathcal{A}_{\lambda_{23};\lambda}^{d,J^P} + i\mathcal{T}_{\lambda_{23},\lambda'_{23}}^{J^P} G_0 i\mathcal{A}_{\lambda'_{23};\lambda}^J = i\mathcal{A}_{\lambda_{23};\lambda}^{d,J^P} + \eta' i\mathcal{T}_{\lambda_{23},\lambda'_{23}}^{J^P} G_0 i\mathcal{A}_{-\lambda'_{23};\lambda}^J = i\mathcal{A}_{\lambda_{23};\lambda}^{d,J^P} + \frac{1}{2} i\mathcal{T}_{\lambda_{23},\lambda'_{23}}^{J^P} G_0 i\mathcal{A}_{\lambda'_{23};\lambda}^{J^P} \tag{45}
\end{aligned}$$

Here $\mathcal{M}_{\lambda_{23};\lambda}^{J^P} = \mathcal{M}_{\lambda_{23};\lambda}^J + \eta \mathcal{M}_{\lambda_{23};\lambda}^J$, $\mathcal{A}_{\lambda_{23};\lambda}^{d,J^P} = \mathcal{A}_{\lambda_{23};\lambda}^{d,J} + \eta \mathcal{A}_{\lambda_{23};\lambda}^{d,J}$, $\mathcal{T}_{\lambda_{23},\lambda'_{23}}^{J^P} = \mathcal{T}_{\lambda_{23},\lambda'_{23}}^J + \eta \mathcal{T}_{-\lambda_{23},\lambda'_{23}}^J = \mathcal{T}_{\lambda_{23},\lambda'_{23}}^J + \eta' \mathcal{T}_{\lambda_{23},-\lambda'_{23}}^J$.

NOTE: For parity conserving interaction, we have $\mathcal{T}_{\lambda'\lambda}^J = \eta(\eta')^{-1} \mathcal{T}_{-\lambda'\lambda}$, which can be checked in the code by different definitions of \mathcal{T}^{J^P} .

We summarize the results as following,

$$\begin{aligned}
\frac{d\Gamma}{dM_{23}} &= \frac{1}{2M} \frac{1}{(2\pi)^5} \frac{\check{p}_1 p_3}{8M} \sum_{ij;J^P} |N_J \mathcal{M}_{ij}^{J^P}(M_{23})|^2, \\
\hat{\mathcal{M}}_{i;i}^{J^P}(M_{23}) &= \mathcal{A}_{j;i}^{d,J^P}(M_{23}) + \sum_k \int \frac{dp'_3 p_3'^2}{(2\pi)^3} \mathcal{T}_{j;k}^{J^P}(p'_3, M_{23}) G_0(p'_3) \mathcal{A}_{k;i}^{J^P}(p'_3, M_{23}),
\end{aligned} \tag{46}$$

where i and j denote the independent $\lambda_{2,3}$ and λ , and the factors $f_{i=0} = 1/\sqrt{2}$ and $f_{i \neq 0} = 1$ are inserted. The above equation can be abbreviated as $M = A^d + TGA$, where T is solved by the Bethe-Salpeter equation $T = V + VGT$. NOTE: The \hat{T} should be multiplied by 4π to be used as \mathcal{T} , and \hat{G}_0 should be divided by 4π to be used as G_0 , which is cancelled by each other.

The case with more than one rescattering

In the case with more than one rescattering, we should consider Monte-Carlo method to generate the event.

$$d\Gamma = \frac{1}{2E} \sum |\mathcal{M}|^2 d\Phi = \frac{1}{2E} \sum |\mathcal{M}|^2 (2\pi)^{4-3n} dR \tag{47}$$

The distribution can be calculated with `qBSE.Xsection`.

Here we consider a process with direct, 23 rescattering, 13 rescattering.

$$i\mathcal{M}_{\lambda_1;\lambda_2,\lambda_3;\lambda}^d(p_1, p_2, p_3) = i\mathcal{A}_{\lambda_1;\lambda_2,\lambda_3;\lambda}(p_1, p_2, p_3),$$

$$i\mathcal{M}_{\lambda_k;\lambda_i,\lambda_j;\lambda}^Z(p_k, p_i, p_j) = \sum_J N_J \sum_{\lambda'_i \lambda'_j} \int \frac{p_j'^2 dp_j'}{(2\pi)^3} i\mathcal{T}_{\lambda_i,\lambda_j;\lambda'_i,\lambda'_j}^J(p'_j, M_{ij}) G_0(p'_j) i\mathcal{A}_{\lambda_k;\lambda'_i,\lambda'_j;\lambda}^{J\lambda_{ji}}(\Omega_k^{lab}, p'_j, M_{ij}, \Omega_j^{cm}). \quad (48)$$

$$\begin{aligned} \sum_{\lambda'} T_{\lambda\lambda'}^J A_{\lambda'}^J &= T_{\lambda 0}^J A_0^J + \sum_{j>0} [T_{\lambda j}^J A_j^J + T_{\lambda -j}^J A_{-j}^J] \\ &= \frac{1}{4} T_{\lambda 0}^{J^P} A_0^{J^P} + \sum_{j>0} \left[\frac{1}{4} (T_{\lambda j}^{J^+} + T_{\lambda j}^{J^-}) (A_j^{J^+} + A_j^{J^-}) + \frac{1}{4} (T_{\lambda j}^{J^+} - T_{\lambda j}^{J^-}) (A_j^{J^+} - A_j^{J^-}) \right] \\ &= \frac{1}{4} T_{\lambda 0}^{J^P} A_0^{J^P} + \sum_{j>0} \left[\frac{1}{2} T_{\lambda j}^{J^+} A_j^{J^+} + \frac{1}{2} T_{\lambda j}^{J^-} A_j^{J^-} \right] \\ &= \frac{1}{2} \sum_P T_{\lambda}^{J^P} A^{J^P}. \end{aligned}$$

$$\mathcal{A}_{\lambda_k;\lambda'_i,\lambda'_j;\lambda}^{J^P\lambda_{ji}}(...) = \int d\Omega'_j \left[D_{\lambda_{ji},\lambda'_{ji}}^J(\phi'_j, \theta'_j, 0) \mathcal{A}_{\lambda_k;\lambda'_i,\lambda'_j;\lambda}(..., \Omega'_j, ...) + \eta' D_{\lambda_{ji},-\lambda'_{ji}}^J(\phi'_j, \theta'_j, 0) \mathcal{A}_{\lambda_k;-\lambda'_i,-\lambda'_j;\lambda}(..., \Omega'_j, ...) \right] \quad (49)$$

$$i\mathcal{M}_{\lambda_k;\lambda_i,\lambda_j;\lambda}^Z(p_k, p_i, p_j) = \sum_{J^P} \frac{N_J^2}{2} \int \frac{p_j'^2 dp_j'}{(2\pi)^3} \sum_{i'j'} i\mathcal{T}_{\lambda_i,\lambda_j;\lambda'_i,\lambda'_j}^{J^P}(p'_j, M_{ij}) G_0(p'_j) i\mathcal{A}_{\lambda_k;\lambda'_i,\lambda'_j;\lambda}^{J^P\lambda_{ji}}(\Omega_k^{lab}, p'_j, M_{ij}, \Omega_j^{cm}). \quad (50)$$

Here we move the N_J in \mathcal{A}^{J^P} to the definition of \mathcal{M} . The amplitude are calculated in qBSE.TGA .

$$\sum_{\lambda} \left| \frac{1}{2} T_{\lambda}^{J^P} A^{J^P} \right|^2 = \left| \frac{1}{2} T_0^{J^P} A^{J^P} \right|^2 + 2 \sum_{j>0} \left| \frac{1}{2} T_j^{J^P} A^{J^P} \right|^2 = \frac{1}{2} \sum_j |T^{J^P} A^{J^P}|^2. \quad (51)$$

$$\sum_{\lambda_k;\lambda_i,\lambda_j;\lambda} |i\mathcal{M}_{\lambda_k;\lambda_i,\lambda_j;\lambda}^Z(p_k, p_i, p_j)|^2 = \frac{1}{2} \sum_{J^P, \lambda_k, \lambda} N_J^2 \sum_j |T^{J^P} A^{J^P}|^2. \quad (52)$$

- The GEN package can generate the events with $(p_1^{lab}, p_2^{lab}, p_3^{lab})$ in the frame of Y .
- The M_{ij} can be obtained as $(p_i^{lab} + p_j^{lab})^2$, and used to solve the $\mathcal{T}_{\lambda_i,\lambda_j;\lambda'_i,\lambda'_j}^{J^P}(p'_j, M_{ij})$ with qBSE. Here, the p'_j is generated by the Gauss discretization.
- With Lorentz boost and rotation, the momentum of p_i in the center of mass frame of particles ij can be calculated. However, for the intermediate particles, the $D_{\lambda_{ji},\lambda'_{ji}}^J(\Omega'_j)$ is used to calculate $\mathcal{A}_{\lambda_k;\lambda'_i,\lambda'_j;\lambda}^{J\lambda_{ji}}(\Omega_k^{lab}, p'_j, M_{ij})$.
- Because only $|\mathcal{M}|^2$ has the same value in different frames, the \mathcal{M} for rescatterings of particles 23 and 13 can not be summed up directly.

qBSE package

The qBSE package is used to solve the Bethe-Salpeter equation with some auxiliary functions.

Data Structures for the Dimensions

In the qBSE approach, matrix dimensions for a coupled-channel system are organized hierarchically: the outermost level corresponds to different channels, the next to independent helicities, and the innermost to discretized momentum points. The data structures are designed to clearly separate and manage these levels.

structChannel

The `struct structChannel` structure encapsulates the properties of a physical channel in the qBSE package, often used as `CH` in the code. Its fields include:

- `p::Vector{Float64}` , `m::Vector{Float64}` , `J::Vector{Int64}` , `Jh::Vector{Int64}` , `P::Vector{Int64}` : Vectors specifying the momentum p , mass m , total angular momentum $J = J/J_h$, and parity (P) for each particle in the channel. Here, $J_h = 1$ for integer spin and $J_h = 2$ for half-integer spin.
- `cutoff::Float64` specifying the cutoff parameter for the channel.
- `name::String` , `name0::String` for the channel name, with and without charge. `name0` is used to define channels with definite isospin.
- `IHb::Int64` , `IHe::Int64` , `IHn::Int64` indicating the starting and ending indices of independent helicities, and the total number of independent helicity states for the channel.

These definitions ensure that each channel's quantum numbers and relevant parameters are explicitly tracked, supporting efficient matrix construction and manipulation in qBSE calculations.

structIndependentHelicity

The mutable `struct structIndependentHelicity` structure defines the independent helicity states in the qBSE package, commonly referenced as `IH` in the code. Its fields are:

- `iCH::Int64` : Index of the channel to which this independent helicity belongs.
- `hel::Vector{Int64}` , `helh::Vector{Int64}` : Vector of helicity values for the independent helicity states, with `helh` indicating whether each helicity state corresponds to a fermion or boson.
- `Dimb::Int64` , `Dime::Int64` , `Dimn::Int64` for the starting index, ending index, and total count of independent helicities (i.e., the dimension range).
- `Dimo::Int64` : Indicates whether an extra on-shell dimension is present ($+1$ if $W > \sum m$, otherwise 0). The total number of momentum discretization points for an independent helicity is $N_p + Dimo$.

These fields help manage the indexing and dimensionality of helicities, especially when handling on-shell and off-shell contributions in the qBSE framework.

structDimension

The mutable `struct structDimension` structure stores information about each dimension in total matrix. Its fields are:

- `iIH::Int64` : Index indicating which independent helicity this dimension belongs to.
- `k::ComplexF64` , `w::Float64` : The discretized momentum value and its corresponding integration weight for this dimension (used in numerical integration).
- `Dimo::Int64` : Indicates whether an extra on-shell dimension is present ($+1$ if $W > \sum m$, otherwise 0). The total number of discretization points for an independent helicity is $N_p + Dimo$.

This structure is auxiliary one to above structures and ensures that each discretized momentum point is properly associated with its helicity and integration weight, supporting accurate matrix construction and numerical calculations in the qBSE framework.

Data Structures for the Interactions

In the qBSE package, interaction and system information are encapsulated in dedicated data structures to facilitate efficient computation and clear organization.

structSys

The `struct structSys` (often referenced as `SYS` in the code) stores information about the system and the generally used discretization and angular integration data. Its fields include:

- `Sys::String` : A label identifying the system.
- `kv::Vector{Float64}` , `wv::Vector{Float64}` : Discretized momentum points and weights for the momentum discretization (used in numerical integration).
- `xv::Vector{Float64}` , `wxv::Vector{Float64}` : Discretized values and weight of $\cos \theta$.
- `d::Vector{Matrix{Float64}}` : Precomputed Wigner d -matrices of θ .
- `pv::Vector{Float64}` , `wpv::Vector{Float64}` : Discretized azimuthal angles and weight of ϕ discretization.
- `sp::Vector{Float64}` , `cp::Vector{Float64}` : Sine values and Cosine values of the discretized ϕ angles.

These fields provide all necessary information for numerical integration over momentum and angular variables in the qBSE framework.

structInterAction

The `struct structInterAction` structure (typically used as `IA` in the code) stores the properties of each interaction or exchange process. Its fields are:

- `Nex::Int64` : Total number of exchange particles or processes.
- `name_ex::Vector{String}` , `key_ex::Vector{Int64}` : Names and integer labels of the exchanged particles or interactions.
- `J_ex::Vector{Int64}` , `Jh_ex::Vector{Int64}` , `P_ex::Vector{Int64}` , `m_ex::Vector{Float64}` : Spin, parity, and mass of exchanged particles.
- `dc::Vector{Int64}` : Indicates whether the exchange is direct or crossed.
- `Ff::Vector{Float64}` : Flavor factors associated with each exchange.

These definitions ensure that all relevant quantum numbers and parameters for each interaction are explicitly tracked, supporting flexible and accurate construction of the interaction kernel in qBSE calculations.

Additional Data Structures

structMomentum

The mutable `struct structMomentum` structure stores the four-momenta and related kinematic quantities for a $2 \rightarrow 2$ scattering process. Its fields are:

- `i1::SVector{5,ComplexF64}` , `f1::SVector{5,ComplexF64}` : Initial and final momentum of particle 1.
- `i2::SVector{5,ComplexF64}` , `f2::SVector{5,ComplexF64}` : Initial and final momentum of particle 2.
- `q::SVector{5,ComplexF64}` : Momentum of the exchanged particle.
- `q2::Complex{Float64}` : Invariant mass squared (q^2) of the exchange.
- `qt::Complex{Float64}` : Additional kinematic variable (e.g., t -channel momentum transfer).

structHelicity

The mutable `struct structHelicity` structure stores the helicity information for the particles in a $2 \rightarrow 2$ interaction. Its fields are:

- `i1::Int64` , `i1h::Int64` : Helicity and denominator for initial particle 1.
- `f1::Int64` , `f1h::Int64` : Helicity and denominator for final particle 1.

- `i2::Int64` , `i2h::Int64` : Helicity and denominator for initial particle 2.
- `f2::Int64` , `f2h::Int64` : Helicity and denominator for final particle 2.

structParticle

The `struct structParticle` structure defines the properties of a single particle. Its fields are:

- `name::String` , `name0::String` , `nameL::String` : Particle name (with charge and without charge), LaTeX representation of the particle name.
- `m::Float64` : Mass of the particle.
- `J::Int64` , `Jh::Int64` , `P::Int64` : Spin and Parity.

A function is also provided to read particle information from a file and populate the particle list and key dictionary:

```
function particles!(particles::Vector{structParticle}, pkey::Dict{String,Int64}, filename::String)
```

- `particles` : A global vector (`const p = structParticle[]`) storing all particle information.
- `pkey` : A global dictionary (`const pkey = Dict{String,Int64}()`) mapping particle names to their indices.

This setup allows efficient lookup and management of particle properties for use in qBSE calculations.

Functions for the qBSE

function preprocessing(Sys, channels, Ff, qn; Np=10, Nx=5, Nphi=5)

This function prepares the system and channel data structures for qBSE calculations by setting up the necessary discretization and quantum number information.

Arguments:

- `Sys::String` : Label identifying the system, stored in `SYS.Sys` .
- `channels` : List of channels to be included in the calculation, stored in `IA[]` .
- `Ff` : Flavor factors, stored in `IA[]` .
- `qn` : Quantum numbers for the process.
- `Np` , `Nx` , `Nphi` : Number of momentum discretization points (default: 10), $\cos \theta$ discretization points (default: 5), azimuthal angle discretization points (default: 5).

Example for arguments:

```

Sys == "KNcp"
qn = (I=1, Ih=1, J=1, Jh=2, P=-1, C=-1, lRm=1)

channels = (
  ("K_b0", "N_p", qBSE.p[qBSE.pkey["K"]].m + 0.22 * 1.63),
  ("pi_0", "Sigma_p", qBSE.p[qBSE.pkey["pi"]].m + 0.22 * 1.63),
  ("pi_p", "Sigma_0", qBSE.p[qBSE.pkey["pi"]].m + 0.22 * 1.63),
  ("pi_p", "Lambda", qBSE.p[qBSE.pkey["pi"]].m + 0.22 * 1.63),
  ("eta", "Sigma_p", qBSE.p[qBSE.pkey["eta"]].m + 0.22 * 1.63)
)

Ff = Dict(
  "K_b0:N_p-->K_b0:N_p" => ([["V", 1], 1.0],),
  "K_b0:N_p-->pi_0:Sigma_p" => ([["V", 1], -sqrt(0.5)],),
  "K_b0:N_p-->pi_p:Sigma_0" => ([["V", 1], sqrt(0.5)],),
  "K_b0:N_p-->pi_p:Lambda" => ([["V", 1], -sqrt(1.5)],),
  "K_b0:N_p-->eta:Sigma_p" => ([["V", 1], sqrt(1.5)],),
  "pi_0:Sigma_p-->pi_p:Sigma_0" => ([["V", 1], -2.0],)
)

```

Returns:

- **SYS::structSys** : The processed system object containing discretization and system information.
- **IA::Matrix{structInterAction}** : The processed interaction objects, organized as a matrix.
- **CH::Vector{structChannel}** : The processed channel objects, each describing a physical channel.
- **IH::Vector{structIndependentHelicity}** : The processed independent helicity objects, ready for use in qBSE calculations.

function res(Range, iER, qn, SYS, IA, CH, IH, fV)

This function calculates the rescatering process by qBSE for a given range of energies.

Arguments:

- **Range** : The range of energies to be considered. For example
Range = (ERmin=1.2, ERmax=2.0, NER=200, EIt=0.200, NEI=20, Ep="cm")
- **iER** : The index of the energy range, energy can be obtained as
 $ER = Range.ERmax - iER * (Range.ERmax - Range.ERmin) / (Range.NER - 1)$.
- **qn** : The quantum numbers for the process.
- **SYS, IA, CH, IH** : Obtained by preprocessing.
- **fV** : The potential kernel for qBSE.

Returns:

- **Ect::Vector{ComplexF64}** : The complex energy values for the system.
- **reslogt::Vector{Float64}** : The values of $\log |1 - VG|$.
- **resM2::Matrix{Float64}** : The matrix $|M|^2$ for the channels.
- **IH::Vector{structIndependentHelicity}** : The independent helicity objects, recalculated from the input **IH**.
- **Dim::Vector{structDimension}** : The dimension objects, recalculated from the input **Dim**.
- **TG::Matrix{ComplexF64}** : The TG matrix used for decays.

function fV(k, l, SYS, IA0, CHf, CHi)

This function defines the potential kernel for the qBSE calculation. It is used to compute the interaction potential between particles in a scattering process.

Arguments:

- `k::structMomentum` , `l::structHelicity` : The momenta and helicities of initial and final particles.
- `SYS::structSys` : The same as above.
- `IA0::structInterAction` : The interaction object containing information about the interactions and their properties, one of element of `IA` above.
- `CHf::structChannel` , `CHi::structChannel` : The final and initial channel object containing information about the final state particles, one of element of `CH` above.

Returns:

A value representing the potential kernel, suitable for use in the qBSE package's calculations.

Note:

This is an internal function intended for use within the qBSE package. Users should not call this function directly.

function propFF(k, ex, L, LLi, LLf; lregu=1, lFFex=0)

In the `fV` function, form factors can be included via the auxiliary function `propFF` , which is used to regulate the interaction kernel.

Arguments:

- `k::structMomentum` : The kinematic information for the process.
- `ex::Int64` : Index of the exchanged particle, usually set as `ex = IA0.key_ex[le]` , where `le` runs over the exchanged mesons in `1:IA0.Nex` .
- `L::Float64` : The cutoff parameter for the exchanged meson, often chosen as the average of the initial and final particle cutoffs: $L = (LLi + LLf) / 2$.
- `LLi::Float64` , `LLf::Float64` : Cutoff parameters for the initial and final particles, typically set as `LLi = CHi.cutoff` , `LLf = CHf.cutoff` .
- `lregu::Int64` : Regularization flag for constituent particles (default: 1).
- `lFFex::Int64` : Type of form factor for the exchanged meson:
 - 0 : No form factor.
 - 1 : $\frac{L^2 - m^2}{L^2 - q^2}$
 - 2 : $\frac{L^4}{(m^2 - q^2)^2 + L^4}$
 - 3 : $\exp\left(-\frac{(m^2 - q^2)^2}{L^4}\right)$
 - 4 : $\frac{L^4 + (q_t - m^2)^2/4}{(q^2 - (q_t + m^2)/2)^2 + L^4}$

Description:

These options allow flexible control over the inclusion and type of form factors in the potential kernel, supporting different regularization schemes as needed for the physical system under study.

function simpleXsection(ER, resM2, CH, qn; Ep="cm")

This function computes the total cross section for a $2 \rightarrow 2$ scattering process using the squared amplitude matrix and channel information.

Arguments:

- `ER::Vector{Float64}` : Array of energy values (typically from `Ect`).
- `resM2::Matrix{Float64}` : Matrix of squared amplitudes $|M|^2$ for the process, as obtained from the qBSE calculation.
- `CH` : Vector of channel structures, as described above.
- `qn` : Quantum numbers for the process.
- `Ep` : Energy frame to use ("cm" for center-of-mass, "L" for laboratory; default: "cm").

Returns:

A `Vector{Matrix{Float64}}` value representing the total cross section at each energy point.

Description:

The function calculates the cross section by summing or integrating over the relevant momentum discretization points and applying the appropriate kinematic and symmetry factors, including spin averaging and flux normalization. It uses the channel and quantum number information to determine the initial and final state properties.

function lambda(m1, m2, m3)

This function calculates the Kallen function $\lambda(m_1, m_2, m_3)$, which is used in the phase space integration.

Decay

function setTGA(par, sij, k, tecm, i, j)

set the frame and other things for calculating TGA, which should be used before `TGA` function.

Arguments:

- `par` : The parameters for the TGA calculation.
- `sij` : The invariant mass squared of the two particles i and j .
- `k` : The momentum of the particle in the center-of-mass (CM) frame.
- `tecm` : The total energy in the center-of-mass frame.
- `i` : The index of the first particle.
- `j` : The index of the second particle.

Returns:

`para=(E=sqrt(sij), par=par, IH=IH, Dim=Dim, k=kn, P=Pn)`

TGA(cfina1, cinter, Vert, para)

This function calculates the transition amplitude for a given final state `cfina1` and interaction `cinter`.

Arguments:

- `cfina1` : The index of final state configuration (e.g., channel and particle information for the outgoing particles).
- `cinter` : The index of intermediate interaction.

- Vert : The vertex function and associated data, including the vertex structure, precomputed spinors or polarization vectors, and helicity assignments (e.g., Vert=Vertex14, ULc=GA1 * FR.U(para14.P, lLc), l2=l2, l3=l3).
- para : The calculation parameters, as returned by setTGA .

function Vertex14(p1, p2, l1, l2, Vert, k, P)

This function computes the initial decay vertex for a process involving two rescattering particles.

Arguments:

- p1 , p2 : Four-momenta of the two rescattering particles.
- l1 , l2 : Helicity indices of the two rescattering particles.
- Vert : Vertex structure or function, as defined in the context of the calculation.
- k : Momenta of all final state particles.
- P : Four-momentum of the parent (initial) particle.

Returns:

A value representing the decay vertex amplitude, suitable for use in the qBSE package's transition amplitude calculations.

Note:

This is an internal function intended for use within the qBSE package. Users should not call this function directly.

auxiliary function

```
qBSE.showSYSInfo(Range, qn, IA, CH, IH)
```

This function is used to display the system information, including the range of energies, quantum numbers, interaction information, and channel information.

```
qBSE.showPoleInfo(qn, Ec, reslog, "data/output.txt")
```

This function is used to display the pole information, including the energy, width, and other relevant parameters. The results are saved in a file named "output.txt" in the "data" directory.

Additional functions

function ch(pf, pin, amps)

This function constructs a channel tuple for use in decay width or cross section calculations, such as with Xs.Xsection .

Arguments:

- pf : List of final state particle names (e.g., ["pi_m", "pi_p", "pi_p", "Lambda"]).
- pin : List of initial state particle names (can be empty if not needed).
- amps : Amplitude information or function.

Returns:

A tuple of the form (pf=pf, namef=namef, mf=mf, pin=pin, namei=namei, mi=mi, amps=amps) , where:

- pf , pin : Final and initial state particle names.
- namef , namei : Channel names for final and initial states.

- `mf` , `mi` : Masses of final and initial state particles.
- `amps` : Amplitude information.

Example:

To compute the decay width for $\Lambda_c \rightarrow \pi^- \pi^- \pi^+ \Lambda$:

```
ch = qBSE.ch(["pi_m", "pi_p", "pi_p", "Lambda"], [], amps)
```

The resulting `ch` can be passed directly to `Xs.Xsection`. See the `Xs.Xsection` documentation for further usage details.

function LorentzBoostRotation(k, tecm, p1, p2)

This function performs a Lorentz boost and rotation to transform a particle's momentum from the lab (static) frame of the total system to the center-of-mass system (CMS) of two particles.

Arguments:

- `k` : The momentum of the particle in the CMS frame (after transformation).
- `tecm` : The total energy in the CMS frame.
- `p1` : The momentum of the first particle in the lab frame.
- `p2` : The momentum of the second particle in the lab frame.

Returns:

- `knew` : Array of momentum vectors after Lorentz boost and rotation.
- `Pnew` : Total four-momentum vector after the same transformations.

This function is useful for kinematic calculations where momenta need to be expressed in different reference frames, ensuring proper treatment of boosts and rotations.