Documentation for `rate_NR` and `rate_genNR`

The `rate_NR` and `rate_genNR` submodules of `dmdd` allow calculation of differential and total rates for arbitrary nonrelativistic operators. **(We assume throughout that the dark matter is spin-1/2.)**

`rate_NR` allows calculation of rates for isolated nuclear responses with arbitrary momentum and velocity dependence up to second order. The rate that is calculated is proportional to

$$\frac{dR}{dE_R}(E_R, v) \propto \sum_{N=p,n} \sum_{N'=p,n} \left[\sum_{X=\text{std}} R_X \widetilde{W}_X^{(N,N')}(y) + \frac{\vec{q}^2}{m_N^2} \sum_{X=\text{novel}} R_X \widetilde{W}_X^{(N,N')}(y) \right] \quad (1)$$

$$= \sum_{X=\text{all}} \left[R_X^{(0)} + R_X^{(v^2)} + R_X^{(q^2)} + R_X^{(v^2 q^2)} + R_X^{(q^4)} \right] \widetilde{W}_X^{(N,N')}(y) \quad (2)$$

Each response is turned on or off with `sigma_[[response name]]`. The ratio of proton to neutron couplings can be set independently for each response with `fnfp_[[response name]]`. Dimensionless momentum- and velocity-dependent coefficients multiplying *each response* may be set by `v2co_[[response name]]`, `q2co_[[response name]]`, `v2q2co_[[response name]]`, and `q4co_[[response name]]`. The three standard responses additionally have "standard coefficients" with no velocity- or momentum-dependence that are invoked by `stdco_[[response name]]`.

By default, the leading coefficient for each response is set to be nonzero. We use the seven responses compatible with the symmetries of the effective field theory that can be UV completed (including two interference terms) [1]. The M, Σ', and Σ'' are the "standard" responses:

- the M nuclear response, controlled by `sigma_M`, `fnfp_M`, etc. The leading coefficient corresponding to $R_M^{(0)}$ is `stdco_M`.
- the Σ' nuclear response, controlled by `sigma_SigP`, `fnfp_SigP`, etc. The leading coefficient corresponding to $R_{\Sigma'}^{(0)}$ is `stdco_SigP`.
- the Σ'' nuclear response, controlled by `sigma_SigPP`, `fnfp_SigPP`, etc. The leading coefficient corresponding to $R_{\Sigma''}^{(0)}$ is `stdco_SigPP`.
- the Φ'' nuclear response, controlled by `sigma_PhiPP`, `fnfp_PhiPP`, etc. Because `stdco_PhiPP` does not exist, the leading coefficient corresponding to $R_{\Phi''}^{(q^2)}$ is `q2co_PhiPP`.
- the Δ nuclear response, controlled by `sigma_Delta`, `fnfp_Delta`, etc. Because `stdco_Delta` does not exist, the leading coefficient corresponding to $R_{\Delta}^{(q^2)}$ is `q2co_Delta`.
- the $M - \Phi''$ interference term, controlled by `sigma_MPhiPP`, `fnfp_MPhiPP`, etc. Because `stdco_MPhiPP` does not exist, the leading coefficient corresponding to $R_{M-\Phi''}^{(q^2)}$ is `q2co_MPhiPP`.
- the $\Sigma' - \Delta$ interference term, controlled by `sigma_SigPDelta`, `fnfp_SigPDelta`, etc. Because `stdco_SigPDelta` does not exist, the leading coefficient corresponding to $R_{\Delta-\Sigma'}^{(q^2)}$ is `q2co_SigPDelta`.

It is possible that a rate calculated with `rate_NR` is negative: this is a signal that the coefficients chosen are unphysical. Because the coefficients that *enter the rate* are chosen by the user, arbitrary choices of the parameters in `rate_NR` may provide meaningless rates.

`rate_genNR` allows calculations of rates for a nonrelativistic Lagrangian with arbitrary coefficients. `dmdd` automatically calculates the correct momentum and velocity dependence *in the rate* up to second order in v^2 and $\vec{q}^2$. We use the naming convention of [2] for the dimensionful coefficients, so there are twenty-eight free parameters characterizing these coefficients. These are entered by setting `cXN`, where `X` $\in \{1, 3, 4, 5, \ldots 15\}$ and `N` $= p, n$. Each coefficient has mass dimension negative two, but an overall mass scale `c_scale_global` has been factored out and set to 500 GeV. This may be changed for a given rate by setting the keyword argument `c_scale`. There is *only one scale per rate.* To allow for novel momentum dependence *in the coefficients*, each coefficient must be entered as a `numpy` array. The (Pythonically numbered) n^{th} element of this array multiplies $(\vec{q}^2/m_{\text{DM}}^2)^n$, so the array elements carry no momentum-dependence, multiply $\vec{q}^2/m_{\text{DM}}^2$, and multiply $\vec{q}^4/m_{\text{DM}}^4$, respectively. Thus, for the nonrelativistic Lagrangian

$$\mathcal{L}_{\text{NR}} = \sum_{X\in\{1,3,4,5,\ldots 15\}} \sum_{N=p,n} c_{XN}\mathcal{O}_{XN} \tag{3}$$

$$= \sum_{X\in\{1,3,4,5,\ldots 15\}} \sum_{N=p,n} \left[c_{XN}^{(0)} + \frac{\vec{q}^2}{m_{\text{DM}}^2} c_{XN}^{(2)} + \frac{\vec{q}^4}{m_{\text{DM}}^4} c_{XN}^{(4)}\right] \mathcal{O}_{XN} \tag{4}$$

the corresponding free parameters are

- `c_scale`, which sets the order of magnitude for the cross section
- `cXN=np.array([c0XN,c2XN,c4XN])`, where `X` $\in \{1, 3, 4, 5, \ldots 15\}$ and `N` $= p, n$, and first, second, and third elements, respectively, multiply 1, $\vec{q}^2/m_{\text{DM}}^2$, and $\vec{q}^4/m_{\text{DM}}^4$, corresponding to $c_{XN}^{(0)}, c_{XN}^{(2)}$, and $c_{XN}^{(4)}$ in Eq. 4.

Because these coefficients are combined inside the rate calculator, it should be impossible to get a negative rate. From the UV perspective, completely generic values for the couplings may be attainable only by precise fine-tuning, but this is not nonphysical like a negative rate.

Example We calculate the rate given by the $\vec{L}\cdot\vec{S}$ operator [3] scattering off of an iodine target at 0.1, 1, and 10 keVNR, for which the coupling to neutrons is ten times as great as the coupling to protons. This can be accomplished in three ways:

- ```
  dmdd.rate_UV.dRdQ(np.array([.1,1.,10.]), sigma_LS=1.,
  fnfp_LS=10., element='iodine')
  ```
- ```
  dmdd.rate_NR.dRdQ(np.array([.1,1.,10.]), sigma_M=1.,
  fnfp_M=10., stdco_M=0., q4co_M=(0.1/0.938272)**4.,
  sigma_MPhiPP=1., fnfp_MPhiPP=10., q2co_MPhiPP=0.,
  q4co_MPhiPP=4.*(0.1/0.938272)**4., sigma_PhiPP=1.,
  fnfp_PhiPP=10., q2co_PhiPP=0., q4co_PhiPP=4.*(0.1/0.938272)**4.,
  sigma_SigP=1., fnfp_SigP=10., stdco_SigP=0.,
  v2q2co_SigP=2.*0.1**2/0.938272**2., q4co_SigP=0.1**4./(50.**2.*0.938272**2.)
  - 0.5*0.1**4./0.938272**2.* (dmdd.constants.ELEMENT_INFO['iodine']['weight']
  *0.938272 +50.)**2./ (dmdd.constants.ELEMENT_INFO['iodine']['weight']*0.938272
  *50.)**2., element='iodine')*8.8036e4
  ```

- `dmdd.rate_genNR.dRdQ(np.array([.1,1.,10.]),`
  `c1p=np.array([0.,-(50./0.938272)**2./4.,0.]),`
  `c1n=np.array([0.,-10.*(50./0.938272)**2./4.,0.]),`
  `c3p=np.array([-1.,0.,0.]), c3n=np.array([-10.,0.,0.]),`
  `c4p=np.array([0.,50./0.938272,0.]), c4n=np.array([0.,10.*50./0.938272,0.]),`
  `c6p=np.array([-0.938272/50.,0.,0.]), c6n=np.array([-10.*0.938272/50.,0.,0.]),`
  `c_scale=92.933, element='iodine')`

using the calculators in `rate_UV`, `rate_NR`, and `rate_genNR`, respectively.

References

[1] A. L. Fitzpatrick, W. Haxton, E. Katz, N. Lubbers and Y. Xu, JCAP **1302**, 004 (2013) [arXiv:1203.3542 [hep-ph]].

[2] N. Anand, A. L. Fitzpatrick and W. C. Haxton, Phys. Rev. C **89**, no. 6, 065501 (2014) [arXiv:1308.6288 [hep-ph]].

[3] M. I. Gresham and K. M. Zurek, Phys. Rev. D **89**, no. 12, 123521 (2014) [arXiv:1401.3739 [hep-ph]].