`dmdd` Rate Calculators

The `rate_UV`, `rate_NR`, and `rate_genNR` submodules of `dmdd` allow calculation of differential and total rates using, respectively, the functions `dRdQ` (with units counts/kg/sec/keV) and `R` (with units counts/kg/sec):

- `dmdd.rate_UV` describes the scattering controlled by a few Lorentz-invariant contact operators
- `dmdd.rate_NR` describes the scattering for arbitrary coefficients in front of the nuclear responses allowed by the appropriate effective field theory
- `dmdd.rate_genNR` describes the scattering for a completely generic coefficients of the full set of nonrelativistic contact operators

We describe the physics of these rate calculators here. **We assume throughout that the dark matter is spin-1/2.** We refer to the rest of the documentation for further details on the `dmdd` package.

`rate_UV` allows calculation of rates for a variety of Lorentz-invariant models of contact interactions between dark matter and the Standard Model. Each Lagrangian has an overall mass dimension negative two normalization as a free parameter; following conventional practice, we define this parameter to be the cross section for scattering off of a proton, σ^p. Some models have a second free parameter that sets the ratio of the dark matter couplings to neutrons and protons. Each model **M** only contributes to the rate when `sig_`[[**M**]] is set to be nonzero. If the ratio of the nucleon couplings is a free parameter, this is set by `fnfp_`[[**M**]]. Each model may have `_massless` appended to it to calculate the same rate assuming a massless mediator. The list of models currently incorporated are summarized in Table 1, along with the overall normalization that sets the scale of the cross section $\sigma^p_{\mathbf{M}}$. These UV-inspired models are discussed in depth in [1, 2].

Table 1: Models available to `dmdd.rate_UV`. The name of the model **M** recognized by `dmdd.rate_UV` is given in the first column, followed by its more conventional name in the second column. If f_n/f_p is a free parameter, we put a checkmark in the third column. The normalization of $\sigma^p_{\mathbf{M}}$ in the massive and massless mediator cases is given in the last two columns.

M	Colloquial Name	f_n/f_p free?	$\sigma_{\rm massive}$	$\sigma_{\rm massless}$
`si`	canonical spin-independent	✓	10^{-47} cm^2	10^{-48} cm^2
`sd`	canonical spin-dependent	✓	10^{-42} cm^2	10^{-43} cm^2
`anapole`	anapole moment	–	10^{-40} cm^2	10^{-45} cm^2
`magdip`	magnetic dipole moment	–	10^{-38} cm^2	10^{-39} cm^2
`elecdip`	electric dipole moment	–	10^{-44} cm^2	10^{-45} cm^2
`LS`	"$\vec{L}\cdot\vec{S}$" generating	✓	10^{-44} cm^2	10^{-42} cm^2
`f1`	pseudoscalar-scalar (DM-SM)	✓	10^{-47} cm^2	10^{-48} cm^2
`f2`	scalar-pseudoscalar (DM-SM)	✓	10^{-42} cm^2	10^{-43} cm^2
`f3`	pseudoscalar-pseudoscalar	✓	10^{-41} cm^2	10^{-42} cm^2

`rate_NR` allows calculation of rates for isolated nuclear responses with arbitrary momentum and velocity dependence up to second order in the effective field theory (EFT) expansion parameters v^2 and $\vec{q}^2/m_N^2$. The output of `dmdd.rate_NR.dRdQ` is proportional to [1, 3]

$$\frac{dR_{T|\mathbf{H}}}{dE_R} \propto \sigma_{\mathbf{H}} \sum_{N,N'} \left[\sum_{X=\text{std}} R_{X|\mathbf{H}} \widetilde{W}_{X|T}^{(N,N')} + \frac{\vec{q}^2}{m_N^2} \sum_{X=\text{novel}} R_{X|\mathbf{H}} \widetilde{W}_{X|T}^{(N,N')} \right] \tag{1}$$

$$\to \sigma_{\mathbf{H}}^p \sum_{N,N'} \sum_{X=\text{all}} \left[\frac{R_{X|\mathbf{H}}^{(0)} + v^2 R_{X|\mathbf{H}}^{(v^2)} + \frac{\vec{q}^2}{m_N^2} R_{X|\mathbf{H}}^{(q^2)} + \frac{v^2\vec{q}^2}{m_N^2} R_{X|\mathbf{H}}^{(v^2q^2)} + \frac{\vec{q}^4}{m_N^4} R_{X|\mathbf{H}}^{(q^4)}}{f_p^2} \right] \widetilde{W}_{X|T}^{(N,N')}, \tag{2}$$

where E_R is the nuclear recoil energy; T is the target element; $\mathbf{H}$ is the choice of response; N, N' represent the neutron or the proton; $\vec{q}^2$ implicitly depends on the nuclear recoil energy E_R; and we factor out the coupling of dark matter to the proton, f_p, from all of the response functions. The response functions $R_{X|\mathbf{H}}$ implicitly depend only on N, N' and $\mathbf{H}$. The functions $\widetilde{W}_{X|T}$ implicitly depend on N, N', T, and E_R. The overall normalization $\sigma_{\mathbf{H}}^p$ is free, just as $\sigma_{\mathbf{M}}^p$ is free in `rate_UV`.

Each response is turned on or off with `sigma_`[[**H**]], all of which have units of 10^{-47} cm^2. The ratio of proton to neutron couplings can be set independently for each response by specifying `fnfp_`[[**H**]]. Dimensionless momentum- and velocity-dependent coefficients multiplying each response may be set using `v2co_`[[**H**]], `q2co_`[[**H**]], `v2q2co_`[[**H**]], and `q4co_`[[**H**]]. The three standard responses additionally have "standard coefficients" with no velocity- or momentum-dependence that are invoked by `stdco_`[[**H**]]. By default, the leading coefficient for each response is set to be nonzero and the rest are set to vanish.

We use the seven responses compatible with the symmetries of the EFT that can be UV completed (including two interference terms) [3]. These are:

- the M nuclear response, controlled by `sigma_M`, `fnfp_M`, etc. The leading coefficient corresponding to $R_M^{(0)}$ is `stdco_M`. This response controls canonical spin-independent scattering.
- the Σ' nuclear response, controlled by `sigma_SigP`, `fnfp_SigP`, etc. The leading coefficient corresponding to $R_{\Sigma'}^{(0)}$ is `stdco_SigP`. This response partially contributes to the canonical spin-dependent scattering.
- the Σ'' nuclear response, controlled by `sigma_SigPP`, `fnfp_SigPP`, etc. The leading coefficient corresponding to $R_{\Sigma''}^{(0)}$ is `stdco_SigPP`. This response partially contributes to the canonical spin-dependent scattering.

Collectively, M, Σ', and Σ'' are the "standard" responses. The novel responses are

- 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 using `dmdd.rate_NR.dRdQ` will return negative values: 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.rate_genNR.dRdQ` and `dmdd.rate_genNR.R` *automatically* calculate 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 [4] for the dimensionful coefficients, so there are twenty-eight free parameters characterizing these coefficients. 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 $\left(\vec{q}^2/m_{\text{DM}}^2\right)^n$, so the first (second) [third] array elements correspond to terms in the calculation that carry no momentum-dependence (multiply $\vec{q}^2/m_{\text{DM}}^2$) [multiply $\vec{q}^4/m_{\text{DM}}^4$]. Thus, for the nonrelativistic Lagrangian

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

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

the corresponding free parameters are

- `c_scale` $\equiv 1/\sqrt{\bar{c}}$, which sets the order of magnitude for the cross section. Because $\mathcal{O}_{XN}$ is a dimension six contact operator, $\bar{c}$ is of mass dimension negative two. In `rate_genNR`, the corresponding free parameter is a **mass scale** that has units GeV and is set to 500 by default. There is *only one scale per rate.*

- `cXN=np.array([c0XN,c2XN,c4XN])`, where `X` $\in \{1,3,4,5,\ldots 15\}$ and `N` $= p, n$. The first, second, and third elements, respectively, are numbers that multiply 1, $\vec{q}^2/m_{\text{DM}}^2$, and $\vec{q}^4/m_{\text{DM}}^4$, corresponding to $\tilde{c}_{XN}^{(0)}, \tilde{c}_{XN}^{(2)}$, and $\tilde{c}_{XN}^{(4)}$ in Eq. 4. Thus, there are 28 independent `cXN` parameters that can contain up to 84 dimensionless coefficients. However, because some of the coefficients only enter the responses at higher order in the EFT expansions parameters, some of coefficients are not called: for example, c_{14} enters the responses accompanied by a factor $v^2\vec{q}^2/m_N^2$ [4], so only the first element of `c14N` is used by the rate calculator. Nonetheless, `c14n` and `c14p` must be entered as single-entry numpy arrays.

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 exquisite fine-tunings between Lorentz-invariant coefficients, but we do not consider this unphysical. The rate calculator accepts the lists of EFT coefficients and evaluates the rate numerically. To find a parametric form of the cross section for a given set of EFT coefficients, we refer to the discussion in [4].

Example We show three ways to calculate the rate given by the $\vec{L}\cdot\vec{S}$ operator [1] scattering off of an iodine target at 0.1, 1, and 10 keVNR, taking the coupling to neutrons to be ten times greater than the coupling to protons:

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

[2] V. Gluscevic, M. I. Gresham, S. D. McDermott, A. H. G. Peter and K. M. Zurek, to appear.

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

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