1 The Mathematica Script: Documentation

The formalism presented in this paper, with its factorization cross sections into products of WIMP and nuclear responses, is the basis for the Mathematica script presented here. The script was constructed so that experimental groups would be able to conduct model independent analyses of their experiments using the EFT framework. We have integrated the particle and nuclear physics in ways that should make the code useful to nuclear structure and particle theorists as well, as described in previous sections.

In this section, which also serves as a `readme` file for the program, we discuss the usage of the program itself.

1.1 Initialization

Our Mathematica package, along with all of the associated documentation, can be found at http://www.ocf.berkeley.edu/~nanand/software/dmformfactor/. To initialize the package, either put `dmformfactor.m` in your directory for Mathematica packages and run

```
<<‘dmformfactor
```

or initialize the package file itself from its source directory. For example

```
<<"/Users/me/myfiles/dmformfactor.m"
```

1.2 Summary of Functions

In order to compute the WIMP response functions $R_i^{\tau\tau'}(\vec{v}_T^{\perp 2}, \frac{\vec{q}^2}{m_N^2})$, the user must first call functions setting the dark matter mass and spin as well as the coefficients of the effective Lagrangian. In order to compute the nuclear response functions $W_i((qb/2)^2)$, the user must specify the Z and A of the isotope. The density matrices and the oscillator parameter b needed in the calculation of the W_i are set internally in the script, though there are options to override the internal values. The nuclear ground state spin and isospin (the script assumes exact isospin, consistent with an input density matrix that is doubly reduced - see text) are also set internally, once Z and A are input.

- `SetJChi` and `SetMChi`: These set the dark matter spin and mass, respectively. Simply call:

  ```
  SetJChi[j]
  ```

  and

  ```
  SetMChi[m]
  ```

  to set the dark matter spin to `j` and the dark matter mass to `m`. The unit `GeV` is recognized by the script; for example, calling `SetMChi[10 GeV]` sets the dark matter mass to 10 GeV.

- `SetIsotope[Z,A,bFM, filename]`

  This sets the nuclear physics input, including the charge `Z` and atomic number `A` of the isotope, the file for the density matrices that the user wants to use, and the oscillator parameter b[fm] (that is, b in femtometers). If the users elects to use the default density matrices (which are available for ^{19}F, ^{23}Na, ^{70}Ge, ^{72}Ge, ^{73}Ge, ^{74}Ge, ^{76}Ge, ^{127}I, ^{128}Xe, ^{129}Xe, ^{130}Xe, ^{131}Xe, ^{132}Xe, ^{134}Xe, and ^{136}Xe), then simple take `filename` to be “default” (note that one must still specify the correct `Z`

and `A` for the isotope of interest. Otherwise, users must provide their own density matrix file, to be read in by the program. Similarly, entering "default" for b will employ the approximate formula $b[\text{fm}] = \sqrt{41.467/(45A^{-1/3} - 25A^{-2/3})}$. To use another value of b[fm], enter a numerical value. The nuclear mass is set to Am_N.

- `SetCoeffsNonrel[i,value,isospin]`

  This sets the coefficients c_i of the EFT operators $\mathcal{O}_i$. The script allows the user to set values for $\{c_1, c_3, c_4, ..., c_{15}\}$; note that c_2 is excluded, for reasons discussed in the text. We have chosen a normalization such that the coefficients c_i all have dimensions (Energy)$^{-2}$;[1] to compensate for this, the dimensionless user input for `value` is multiplied by m_V^{-2}, with $m_V \equiv 246.2$ GeV.

  The coefficients carry an isospin index α that can be specified in one of two ways, as a coupling to protons and neutrons, $\{c_i^p, c_i^n\}$, in which case the associated operator is

$$\left[c_i^p \frac{1+\tau_3}{2} + c_i^n \frac{1-\tau_3}{2}\right] \mathcal{O}_i \tag{1}$$

  or as a coupling to isospin, $\{c_i^0, c_i^1\}$, where the associated operator is

$$\left[c_i^0 + c_i^1 \tau_3\right] \mathcal{O}_i. \tag{2}$$

  For the former, the input should be "n" for neutrons and "p" for protons. For example:

```
SetCoeffsNonrel[4,12.3, "p"]
```

  whereas for the latter it should be 0 for isoscalar and 1 for isovector. All coefficients are set to 0 by default when the package is initialized. `SetCoeffsNonrel` will change only the coefficient specified, and will leave all other coefficients unchanged. So, for example, if one initializes the package and calls `SetCoeffsNonrel[4,12.3, 0]`, then c_4^p and c_4^n will both be 6.15, with all other coefficients vanishing. If one then calls `SetCoeffsNonrel[4,3.3,''p'']`, then c_4^p will be set to 3.3, but c_4^n will not change and will still be 6.15. Thus by making two calls, an arbitrary combination of $\{c_4^p, c_4^n\}$ or equivalently $\{c_4^0, c_4^1\}$ can be set.

- `SetCoeffsRel[i,value,isospin]`

  These functions are similar to `SetCoeffsNonrel`, except that they set the coefficients d_j of the 20 covariant interactions $\mathcal{L}_{\text{int}}^j$ defined in Table 1. The coefficients d_j are dimensionless, by inserting appropriate powers of the user-defined scale m_M, set by the user function `SetMM`. This scale is set by default to be $m_M = m_V \equiv 246.2$ GeV. We adopt a convention where the spinors in $\mathcal{L}_{\text{int}}^j$ are defined as normalized to unity: with this convention a nonrelativistic reduction of the $\mathcal{L}_{\text{int}}^j$ in the second column of Table 1 would give the results in the fourth column. [As noted in the paper, we use a spinor normalization of $2m$ in our derivations, but extract the factor of $4m_\chi m_N$ in order to maintain the definition above.]

  `SetCoeffsNonrel` and `SetCoeffsRel` cannot be used together. By default, the package assumes you will use `SetCoeffsNonrel`. The first time the user calls `SetCoeffsRel`, the package will first reset all coefficients back to zero before calling `SetCoeffsRel`, after which point it will act normally. A subsequent call to `SetCoeffsNonrel` will similarly first reset all coefficients back to zero and then revert to non-relativistic mode.

  Since the relativistic operators implicitly assume spin-1/2 WIMPs, any call to `SetCoeffsRel` automatically sets $j_\chi = 1/2$.

- `SetMM[mM]`

  Set the fiducial scale m_M for the relativistic coefficients d_i.

[1] Note that this convention for the c_i's differs from that in [1].

j	$\mathcal{L}^j_{\rm int}$	Nonrelativistic Reduction	$\sum_i c_i\mathcal{O}_i$	P/T
1	$\bar\chi\chi\bar N N$	$1_\chi 1_N$	$\mathcal{O}_1$	E/E
2	$i\bar\chi\chi\bar N\gamma^5 N$	$i\frac{\vec q}{m_N}\cdot\vec S_N$	$\mathcal{O}_{10}$	O/O
3	$i\bar\chi\gamma^5\chi\bar N N$	$-i\frac{\vec q}{m_\chi}\cdot\vec S_\chi$	$-\frac{m_N}{m_\chi}\mathcal{O}_{11}$	O/O
4	$\bar\chi\gamma^5\chi\bar N\gamma^5 N$	$-\frac{\vec q}{m_\chi}\cdot\vec S_\chi\frac{\vec q}{m_N}\cdot\vec S_N$	$-\frac{m_N}{m_\chi}\mathcal{O}_6$	E/E
5	$\frac{P^\mu}{m_{\rm M}}\bar\chi\chi\frac{K_\mu}{m_{\rm M}}\bar N N$	$4\frac{m_\chi m_N}{m_{\rm M}^2}1_\chi 1_N$	$4\frac{m_\chi m_N}{m_{\rm M}^2}\mathcal{O}_1$	E/E
6	$\frac{P^\mu}{m_{\rm M}}\bar\chi\chi\bar N i\sigma_{\mu\alpha}\frac{q^\alpha}{m_{\rm M}}N$	$-\frac{m_\chi}{m_N}\frac{\vec q^{\,2}}{m_{\rm M}^2}1_\chi 1_N-4i\frac{m_\chi}{m_{\rm M}}\vec v^\perp\cdot\left(\frac{\vec q}{m_{\rm M}}\times\vec S_N\right)$	$-\frac{m_\chi}{m_N}\frac{\vec q^{\,2}}{m_{\rm M}^2}\mathcal{O}_1+4\frac{m_\chi m_N}{m_{\rm M}^2}\mathcal{O}_3$	E/E
7	$\frac{P^\mu}{m_{\rm M}}\bar\chi\chi\bar N\gamma_\mu\gamma^5 N$	$-4\frac{m_\chi}{m_{\rm M}}\vec v^\perp\cdot\vec S_N$	$-4\frac{m_\chi}{m_{\rm M}}\mathcal{O}_7$	O/E
8	$i\frac{P^\mu}{m_{\rm M}}\bar\chi\chi\frac{K_\mu}{m_{\rm M}}\bar N\gamma^5 N$	$4i\frac{m_\chi}{m_{\rm M}}\frac{\vec q}{m_{\rm M}}\cdot\vec S_N$	$4\frac{m_\chi m_N}{m_{\rm M}^2}\mathcal{O}_{10}$	O/O
9	$\bar\chi i\sigma^{\mu\nu}\frac{q_\nu}{m_{\rm M}}\chi\frac{K_\mu}{m_{\rm M}}\bar N N$	$\frac{m_N}{m_\chi}\frac{\vec q^{\,2}}{m_{\rm M}^2}1_\chi 1_N+4i\frac{m_N}{m_{\rm M}}\vec v^\perp\cdot\left(\frac{\vec q}{m_{\rm M}}\times\vec S_\chi\right)$	$\frac{m_N}{m_\chi}\frac{\vec q^{\,2}}{m_{\rm M}^2}\mathcal{O}_1-4\frac{m_N^2}{m_{\rm M}^2}\mathcal{O}_5$	E/E
10	$\bar\chi i\sigma^{\mu\nu}\frac{q_\nu}{m_{\rm M}}\chi\bar N i\sigma_{\mu\alpha}\frac{q^\alpha}{m_{\rm M}}N$	$4\left(\frac{\vec q}{m_{\rm M}}\times\vec S_\chi\right)\cdot\left(\frac{\vec q}{m_{\rm M}}\times\vec S_N\right)$	$4\left(\frac{\vec q^{\,2}}{m_{\rm M}^2}\mathcal{O}_4-\frac{m_N^2}{m_{\rm M}^2}\mathcal{O}_6\right)$	E/E
11	$\bar\chi i\sigma^{\mu\nu}\frac{q_\nu}{m_{\rm M}}\chi\bar N\gamma^\mu\gamma^5 N$	$-4i\left(\frac{\vec q}{m_{\rm M}}\times\vec S_\chi\right)\cdot\vec S_N$	$-4\frac{m_N}{m_{\rm M}}\mathcal{O}_9$	O/E
12	$i\bar\chi i\sigma^{\mu\nu}\frac{q_\nu}{m_{\rm M}}\chi\frac{K_\mu}{m_{\rm M}}\bar N\gamma^5 N$	$\left[i\frac{\vec q^{\,2}}{m_\chi m_{\rm M}}-4\vec v^\perp\cdot\left(\frac{\vec q}{m_{\rm M}}\times\vec S_\chi\right)\right]\frac{\vec q}{m_{\rm M}}\cdot\vec S_N$	$\frac{m_N}{m_\chi}\frac{\vec q^{\,2}}{m_{\rm M}^2}\mathcal{O}_{10}+4\frac{\vec q^{\,2}}{m_{\rm M}^2}\mathcal{O}_{12}+4\frac{m_N^2}{m_{\rm M}^2}\mathcal{O}_{15}$	O/O
13	$\bar\chi\gamma^\mu\gamma^5\chi\frac{K_\mu}{m_{\rm M}}\bar N N$	$4\frac{m_N}{m_{\rm M}}\vec v^\perp\cdot\vec S_\chi$	$4\frac{m_N}{m_{\rm M}}\mathcal{O}_8$	O/E
14	$\bar\chi\gamma^\mu\gamma^5\chi\bar N i\sigma_{\mu\alpha}\frac{q^\alpha}{m_{\rm M}}N$	$-4i\vec S_\chi\cdot\left(\frac{\vec q}{m_{\rm M}}\times\vec S_N\right)$	$4\frac{m_N}{m_{\rm M}}\mathcal{O}_9$	O/E
15	$\bar\chi\gamma^\mu\gamma^5\chi\bar N\gamma^\mu\gamma^5 N$	$-4\vec S_\chi\cdot\vec S_N$	$-4\mathcal{O}_4$	E/E
16	$i\bar\chi\gamma^\mu\gamma^5\chi\frac{K^\mu}{m_{\rm M}}\bar N\gamma^5 N$	$4i\vec v^\perp\cdot\vec S_\chi\frac{\vec q}{m_{\rm M}}\cdot\vec S_N$	$4\frac{m_N}{m_{\rm M}}\mathcal{O}_{13}$	E/O
17	$i\frac{P^\mu}{m_{\rm M}}\bar\chi\gamma^5\chi\frac{K_\mu}{m_{\rm M}}\bar N N$	$-4i\frac{m_N}{m_{\rm M}}\frac{\vec q}{m_{\rm M}}\cdot\vec S_\chi$	$-4\frac{m_N^2}{m_{\rm M}^2}\mathcal{O}_{11}$	O/O
18	$i\frac{P^\mu}{m_{\rm M}}\bar\chi\gamma^5\chi\bar N i\sigma_{\mu\alpha}\frac{q^\alpha}{m_{\rm M}}N$	$\frac{\vec q}{m_{\rm M}}\cdot\vec S_\chi\left[i\frac{\vec q^{\,2}}{m_N m_{\rm M}}-4\vec v^\perp\cdot\left(\frac{\vec q}{m_{\rm M}}\times\vec S_N\right)\right]$	$\frac{\vec q^{\,2}}{m_{\rm M}^2}\mathcal{O}_{11}+4\frac{m_N^2}{m_{\rm M}^2}\mathcal{O}_{15}$	O/O
19	$i\frac{P^\mu}{m_{\rm M}}\bar\chi\gamma^5\chi\bar N\gamma_\mu\gamma^5 N$	$4i\frac{\vec q}{m_{\rm M}}\cdot\vec S_\chi\vec v_\perp\cdot\vec S_N$	$4\frac{m_N}{m_{\rm M}}\mathcal{O}_{14}$	E/O
20	$\frac{P^\mu}{m_{\rm M}}\bar\chi\gamma^5\chi\frac{K_\mu}{m_{\rm M}}\bar N\gamma^5 N$	$-4\frac{\vec q}{m_{\rm M}}\cdot\vec S_\chi\frac{\vec q}{m_{\rm M}}\cdot\vec S_N$	$-4\frac{m_N^2}{m_{\rm M}^2}\mathcal{O}_6$	E/E

Table 1: The Lagrangian densities $\mathcal{L}^j_{\rm int}$, the operators obtained after nonrelativistic reduction that would be used between Pauli spinors to generate the invariant amplitude, the corresponding effective interactions in terms of the EFT operators, and the transformation properties of the interactions (even E or odd O) under parity and time reversal. Bjorken and Drell spinor and gamma matrix conventions are used. The scale $m_{\rm M}$, which usually would be known from the context of the theory, can be put into the Mathematica script, or set to its default value, m_V (see text for further discussion).

- `ZeroCoeffs[]`

  Calling `ZeroCoeffs[]` simply resets all operators coefficients to zero.

- `ResponseNuclear[y,i,tau,tau2]`

  This function prints out any of the eight nuclear response functions $W_i^{\tau\tau_2}(y)$:

$$
\begin{aligned}
W_M^{\tau\tau'}(y) &= \sum_{J=0,2,\ldots}^{\infty} \langle j_N||\ M_{J;\tau}(q)\ ||j_N\rangle\langle j_N||\ M_{J;\tau'}(q)\ ||j_N\rangle \\
W_{\Sigma''}^{\tau\tau'}(y) &= \sum_{J=1,3,\ldots}^{\infty} \langle j_N||\ \Sigma''_{J;\tau}(q)\ ||j_N\rangle\langle j_N||\ \Sigma''_{J;\tau'}(q)\ ||j_N\rangle \\
W_{\Sigma'}^{\tau\tau'}(y) &= \sum_{J=1,3,\ldots}^{\infty} \langle j_N||\ \Sigma'_{J;\tau}(q)\ ||j_N\rangle\langle j_N||\ \Sigma'_{J;\tau'}(q)\ ||j_N\rangle \\
W_{\Phi''}^{\tau\tau'}(y) &= \sum_{J=0,2,\ldots}^{\infty} \langle j_N||\ \Phi''_{J;\tau}(q)\ ||j_N\rangle\langle j_N||\ \Phi''_{J;\tau'}(q)\ ||j_N\rangle \\
W_{\Phi''M}^{\tau\tau'}(y) &= \sum_{J=0,2,\ldots}^{\infty} \langle j_N||\ \Phi''_{J;\tau}(q)\ ||j_N\rangle\langle j_N||\ M_{J;\tau'}(q)\ ||j_N\rangle \\
W_{\tilde{\Phi}'}^{\tau\tau'}(y) &= \sum_{J=2,4,\ldots}^{\infty} \langle j_N||\ \tilde{\Phi}'_{J;\tau}(q)\ ||j_N\rangle\langle j_N||\ \tilde{\Phi}'_{J;\tau'}(q)\ ||j_N\rangle \\
W_{\Delta}^{\tau\tau'}(y) &= \sum_{J=1,3,\ldots}^{\infty} \langle j_N||\ \Delta_{J;\tau}(q)\ ||j_N\rangle\langle j_N||\ \Delta_{J;\tau'}(q)\ ||j_N\rangle \\
W_{\Delta\Sigma'}^{\tau\tau'}(y) &= \sum_{J=1,3,\ldots}^{\infty} \langle j_N||\ \Delta_{J;\tau}(q)\ ||j_N\rangle\langle j_N||\ \Sigma'_{J;\tau'}(q)\ ||j_N\rangle. \qquad (3)
\end{aligned}
$$

  This involves a folding of the single-particle matrix elements with the density matrices. The results are printed as analytic functions in the dimensionless variable $y = (qb/2)^2$. The i run from 1 to 8, according to 1) W_M, 2) $W_{\Sigma''}$, 3) $W_{\Sigma'}$, 4) $W_{\Phi''}$, 5) $W_{\tilde{\Phi}'}$, 6) W_{Δ}, 7) $W_{M\Phi''}$, and 8) $W_{\Sigma'\Delta}$.

- `TransitionProbability[v,q(,IfRel)]`

  This is the main user function. It first prints out the Lagrangian that is being used.

  Second, it folds the $W_i^{\tau\tau'}(y)$ and $R_i^{\tau\tau'}(\vec{v}_T^{\perp 2}, \frac{\vec{q}^{\,2}}{m_N^2})$ to form

$$
P_{\rm tot} = \frac{1}{2j_\chi+1}\frac{1}{2j_N+1}\sum_{\rm spins}|\mathcal{M}|^2_{\rm nucleus-HO/EFT}, \qquad (4)
$$

  given by

$$
\begin{aligned}
&\frac{1}{2j_\chi+1}\frac{1}{2j_N+1}\sum_{\rm spins}|\mathcal{M}|^2_{\rm nucleus-HO/EFT} = \frac{4\pi}{2j_N+1}\sum_{\tau=0,1}\sum_{\tau'=0,1} \\
&\Bigg\{\ \left[R_M^{\tau\tau'}(\vec{v}_T^{\perp 2}, \frac{\vec{q}^{\,2}}{m_N^2})\ W_M^{\tau\tau'}(y) + R_{\Sigma''}^{\tau\tau'}(\vec{v}_T^{\perp 2}, \frac{\vec{q}^{\,2}}{m_N^2})\ W_{\Sigma''}^{\tau\tau'}(y) + R_{\Sigma'}^{\tau\tau'}(\vec{v}_T^{\perp 2}, \frac{\vec{q}^{\,2}}{m_N^2})\ W_{\Sigma'}^{\tau\tau'}(y)\right] \\
&+\frac{\vec{q}^{\,2}}{m_N^2}\ \left[R_{\Phi''}^{\tau\tau'}(\vec{v}_T^{\perp 2}, \frac{\vec{q}^{\,2}}{m_N^2})\ W_{\Phi''}^{\tau\tau'}(y) + R_{\Phi''M}^{\tau\tau'}(\vec{v}_T^{\perp 2}, \frac{\vec{q}^{\,2}}{m_N^2})\ W_{\Phi''M}^{\tau\tau'}(y) + R_{\tilde{\Phi}'}^{\tau\tau'}(\vec{v}_T^{\perp 2}, \frac{\vec{q}^{\,2}}{m_N^2})W_{\tilde{\Phi}'}^{\tau\tau'}(y)\right. \\
&\quad\left. +\ R_{\Delta}^{\tau\tau'}(\vec{v}_T^{\perp 2}, \frac{\vec{q}^{\,2}}{m_N^2})\ W_{\Delta}^{\tau\tau'}(y) + R_{\Delta\Sigma'}^{\tau\tau'}(\vec{v}_T^{\perp 2}, \frac{\vec{q}^{\,2}}{m_N^2})\ W_{\Delta\Sigma'}^{\tau\tau'}(y)\right]\Bigg\}. \qquad (5)
\end{aligned}
$$

It then evaluates the transition probability for the numerical values of b and m_N. As b is in fm, the substitution is $y = (qb/(2\hbar c))^2 \sim (qb/2(0.197\text{Gev fm}))^2$. As m_N is input in GeV, this evaluates Eq. (5) as a function `TransitionProbability[vsq,q]` where `q` is in GeV. This function can be printed out or plotted numerically.

The conventional relativistic normalization of the amplitude differs from the non-relativistic normalization by a factor of $1/(4m_\chi m_T)$. Since the conventional relativistic normalization is commonly used and produces a dimensionless value for $|\mathcal{M}|^2$, we also provide an optional argument `IfRel`, which if set to `True` will output (4) with the relativistic normalization convention (that is, it will multiply by $(4m_\chi m_T)^2$ to produce a dimensionless transition probability). By default, it is set to `False`.

- `DiffCrossSection[ERkeV,v]`

  From the transition probability $P_{\rm tot}$, one can immediately obtain the differential cross section per recoil energy:

$$\frac{d\sigma}{dE_R} = \frac{m_T}{2\pi v^2} P_{\rm tot}. \tag{6}$$

  The function `DiffCrossSection[ERkeV,v]` takes as arguments the recoil energy in units of keV and the velocity of the incoming DM particle in the lab frame. It first prints out the Lagrangian being used, and then outputs the differential cross-section $\frac{d\sigma}{dE_R}$.

- `ApproxTotalCrossSection[v]`

  From the differential cross-section $\frac{d\sigma}{dE_R}$, one can also obtain the total cross-section as a function of v by integrating over recoil energies. In general, this depends on energy thresholds and, written in closed form, is a complicated analytic function due to the exponential damping factor e^{-2y} in the response functions, so for precise values it is simplest to do the energy integration numerically. However, for approximate results we can consider the limit of small nuclear harmonic oscillator parameter b, in which case the exponential factor e^{-2y} can be neglected. For fixed v, the integration over E_R from zero up to the kinematic threshold $E_{R,\rm max} = 2\frac{\mu_T^2 v^2}{m_T}$ can be performed analytically. The function `ApproxTotalCrossSection[v]` takes as argument the velocity v of the incoming DM particle in the lab frame and, after printing out the Lagrangian being used, outputs this approximate total cross-section $\sigma(v)$.

- `EventRate[`N_T`,`ρ_χ`,`q`,`v_e`,`v_0`(,`$v_{\rm esc}$`)]`

  One can determine the total detector event rate (per unit time per unit detector mass per unit recoil energy) in terms of the transition probability $P_{\rm tot}$. One simply multiplies $P_{\rm tot}$ by the appropriate prefactor and integrates over the halo velocity distribution, as follows:

$$\frac{dR_D}{dE_R} = N_T \frac{\rho_\chi m_T}{2\pi m_\chi} \left\langle \frac{1}{v} P_{\rm tot}(v^2, q^2) \right\rangle \tag{7}$$

  Here, $\langle \ldots \rangle$ indicates averaging over the halo velocity distribution. N_T is the number of target nuclei per detector mass, ρ_χ is the local dark matter density, m_χ is the dark matter mass, and m_N is the nucleon mass. In general, the halo average integral should include a lower-bound on the magnitude of the velocity at $v_{\rm min}$, which is $v_{\rm min} = \frac{q}{2\mu_T}$ for elastic scattering:

$$\langle h(q,\vec{v})\rangle \equiv \int_{v_{\rm min}(q)}^{\infty} v^2 dv \int d^2\Omega f_v(\vec{v}+\vec{v}_e) h(q,\vec{v}). \tag{8}$$

  The vector $\vec{v}_e$ is the Earth's velocity in the galactic rest frame. While there has been much work recently on understanding theoretical constraints on the halo distribution from N-body simulations and from general considerations of dynamics, little is known by direct observation and there are still large uncertainties. A very simple approximation that suffices for general considerations is to take a Maxwell-Boltzmann distribution:

$$f_v(\vec{v}) = \frac{1}{\pi^{3/2} v_0^3} e^{-v^2/v_0^2}, \tag{9}$$

where v_0 is roughly 220 km/s, about the rms velocity of the visible matter distribution (though N-body simulations suggest that the dark matter distribution may be shallower, and a larger v_0 may be more appropriate). The function EventRate[q,b,v$_e$,v$_0$] evaluates the event rate $\frac{dR_D}{dE_R}$ assuming this Maxwell-Boltzmann distribution as default. A cut-off Maxwell-Boltzmann distribution is also implemented as an option, in which case

$$f_v(\vec{v}) \quad \propto \quad \left(e^{-v^2/v_0^2} - e^{v_{\rm esc}^2/v_0^2}\right) \Theta(v_{\rm esc}^2 - \vec{v}^2) \tag{10}$$

where $v_{\rm esc}$ is the escape velocity, and the subtraction above is included to make the distribution shut down smoothly. In this case, $v_{\rm esc}$ should be included as an optional argument to EventRate; if it is not included, it is set to a default value of $12v_0$ (which is essentially $v_{\rm esc} = \infty$).

- `SetHALO[halo]`

  This sets the halo distribution used. The variable *halo* can be set either to "MB", in which case the Maxwell-Boltzmann distribution is used, or "MBcutoff", in which case the cut-off Maxwell-Boltzmann distribution is used. It is set to "MB" by default.

- `SetHelm[UseHelm]`

  Calling `SetHelm[True]` sets the structure function for the density operator M_J to be given by the Helm form factor, rather than by the structure function obtained from the density matrix. `SetHelm[False]` implements the structure function based on the density matrix, which is the default setting.

1.3 Examples

A full example for the transition probability would look like the following:

```
<< "/Users/me/mypackages/dmformfactor.m";
SetJChi[1/2]
SetMChi[50 GeV]
F19filename="default";
bFM="default";
SetIsotope[9, 19, bFM, F19filename]
SetCoeffsNonrel[3, 3.1, "p"]
TransitionProbability[v,qGeV]
TransitionProbability[v,qGeV,True]
```

To additionally calculate the event rate $\frac{dR_D}{dE_R}$ in a Maxwell-Boltzmann halo velocity distribution, one can call

```
mNucleon=0.938 GeV;
NT=1/(19 mNucleon);
Centimeter=(10^13 Femtometer);
rhoDM=0.3 GeV/Centimeter^3;
ve=232 KilometerPerSecond;
v0=220 KilometerPerSecond;
EventRate[NT,rhoDM,qGeV,ve,v0]
```

For a cut-off Maxwell-Boltzmann halo, an escape velocity must also be specified:

```
mNucleon=0.938 GeV;
NT=1/(19 mNucleon);
Centimeter=(10^13 Femtometer);
```

```
rhoDM=0.3 GeV/Centimeter^3;
ve=232 KilometerPerSecond;
v0=220 KilometerPerSecond;
vesc=550 KilometerPerSecond;
SetHalo["MBcutoff"];
EventRate[NT,rhoDM,qGeV,ve,v0,vesc]
```

Finally, to get a quick estimate of the experimental bound from the 225 live day run of XENON100, one can use the standard spin-independent isoscalar interaction for a generic isotope of xenon, taking xenon-131 for instance. Taking into account efficiencies, the total effective exposure is approximately 2500 kg days. A relativistic operator coefficient of $2f_p/\mathrm{GeV}^2$ with $f_p = 4 \cdot 10^{-9}$ predicts only a couple of events, and so should be close to the upper limit of their allowed cross-section:

```
mNucleon=0.938 GeV;
NT=1/(131 mNucleon);
Centimeter=(10^13 Femtometer);
rhoDM=0.3 GeV/Centimeter^3;
SetMChi[150 GeV]
ve=232 KilometerPerSecond;
v0=220 KilometerPerSecond;
vesc=550 KilometerPerSecond;
SetHALO["MBcutoff"];
Xe131filename="default";
bFM="default";
SetIsotope[54, 131, bFM, Xe131filename]
SetCoeffsRel[1,2fp,0]
myrate[qGeV_]=(2500 KilogramDay) EventRate[NT,rhoDM,qGeV,ve,v0,vesc];
fp=2.4*10^(-4);
NIntegrate[myrate[qGeV] GeV*(qGeV GeV/(131 mNucleon)),{qGeV,0,10}]
```

The final line of output should be 2.06 for the value of the integral, which gives the predicted number of events. The factor $\frac{q}{131 m_N} = \frac{q}{m_T}$ inside the integral is from the change of variables from dE_R to dq, since $E_R = q^2/2m_T$. In this example, the WIMP is sufficiently heavy that the exact low-energy threshold changes the prediction by less than a factor of two, so to get a rough estimate we have just integrated down to zero energy. Finally, we can look what nucleon scattering cross-section corresponds to $f_p = 2.4 \cdot 10^{-4}$:

$$\sigma_p \quad = \quad \frac{(4 m_N m_T f_p / m_V^2)^2}{16\pi (m_N + m_T)^2} = 1.7 \cdot 10^{-45} \mathrm{cm}^2 \tag{11}$$

which agrees to within a factor of a few with the published upper bound on σ_p from the XENON100 collaboration [2]. A more accurate calculation of the bound would include, among other corrections, the exact energy thresholds in the integral over momentum transfer, an average over the year as the earth's velocity changes, a sum over different isotopes according to their natural abundance, and a more precise treatment of energy-dependent efficiencies.

1.4 Density Matrix Syntax

If one calls `SetIsotope[Z,A, filename]` with a custom density matrix, the input density matrix file must contain the reduced density matrix elements $\Psi^{J,T}(|\alpha|, |\beta|)$ to be used. The in and out states $|\alpha|$ and $|\beta|$ should be specified by their principle quantum number N and their total angular momentum j. See [3] for more details. The format of the file for each projection onto operators of spin J and isospin J should be as follows:

ONE-BODY DENSITY MATRIX $\cdots$ 2J0= $2J$, $\cdots$ $2T$

$$\begin{array}{llll} \cdots\ N^1_{\rm in} & 2j^1_{\rm in} & N^1_{\rm out} & 2j^1_{\rm out} \quad \Psi^{J,T}\left(\{N^1_{\rm in}, j^1_{\rm in}\}; \{N^1_{\rm out}, j^1_{\rm out}\}\right) \\ & \vdots & \vdots & \vdots \\ \cdots\ N^n_{\rm in} & 2j^n_{\rm in} & N^n_{\rm out} & 2j^n_{\rm out} \quad \Psi^{J,T}\left(\{N^n_{\rm in}, j^n_{\rm in}\}; \{N^n_{\rm out}, j^n_{\rm out}\}\right) \end{array}$$

Dots "$\cdots$" indicate places where the code will simply ignore what appears there - the routines reading in the input are searching for regular expressions that match the above syntax. Consequently, additional lines in the file that are not of the above form will also be ignored. This is probably clearest to follow by seeing an explicit example. For instance, the density matrix for ^{19}F is:

```
         INITIAL STATE CHARGE CONJ SYM =  0  TIME REVERSAL SYM =  0

         FINAL STATE CHARGE CONJ SYM =  0  TIME REVERSAL SYM =  0
               -23.88003     -23.88003

 ONE-BODY DENSITY MATRIX FOR 2JF =  1 2TF =  1 2JI =  1 2TI =  1 2JO =  0  TO = , 0
              NBRA   2*JBRA   NKET   2*JKET     VALUE
               0       1       0       1        4.00000000
               1       1       1       1        4.00000000
               1       3       1       3        5.65685425
               2       1       2       1        1.22525930
               2       3       2       3        0.20366116
               2       5       2       5        0.85835832

 ONE-BODY DENSITY MATRIX FOR 2JF =  1 2TF =  1 2JI =  1 2TI =  1 2JO =  0  TO = , 2
              NBRA   2*JBRA   NKET   2*JKET     VALUE
               2       1       2       1        0.36984837
               2       3       2       3        0.04794379
               2       5       2       5        0.32467225

 ONE-BODY DENSITY MATRIX FOR 2JF =  1 2TF =  1 2JI =  1 2TI =  1 2JO =  2  TO = , 0
              NBRA   2*JBRA   NKET   2*JKET     VALUE
               2       1       2       1        0.44514263
               2       3       2       1       -0.01197751
               2       1       2       3        0.01197751
               2       3       2       3       -0.05428837
               2       5       2       3       -0.12172578
               2       3       2       5        0.12172578
               2       5       2       5        0.12280637

 ONE-BODY DENSITY MATRIX FOR 2JF =  1 2TF =  1 2JI =  1 2TI =  1 2JO =  2  TO = , 2
              NBRA   2*JBRA   NKET   2*JKET     VALUE
               2       1       2       1       -0.40780345
               2       3       2       1       -0.01278520
               2       1       2       3        0.01278520
               2       3       2       3        0.01209672
               2       5       2       3        0.10547489
               2       3       2       5       -0.10547489
               2       5       2       5       -0.24110544
```

Example density matrix file shown for ^{19}F. The density matrices for ^{19}F, ^{23}Na, ^{70}Ge, ^{72}Ge,^{73}Ge,^{74}Ge,^{76}Ge,

^{127}I, ^{128}Xe,^{129}Xe,^{130}Xe,^{131}Xe,^{132}Xe,^{134}Xe, and ^{136}Xe are already built into the program and no external file is needed.

References

[1] A. L. Fitzpatrick, W. Haxton, E. Katz, N. Lubbers, and Y. Xu, ArXiv e-prints (2012), 1203.3542.

[2] XENON100 Collaboration, E. Aprile *et al.*, (2012), 1207.5988.

[3] W. Haxton and C. Lunardini, (2007), 0706.2210.