User Manual for DMFortFactor

A Fast Fortran Code for WIMP-Nucleus Form Factors

Oliver C. Gorton Changfeng Jiao Calvin W. Johnson

Contents

1 Introduction

We present here a fast modern Fortran code, DMFortFactor, for computing WIMP-nucleus scattering event rates using a previously studied theoretical framework Fitzpatrick et al. (2013), now with advanced algorithmic and numerical implementation, including the ability to take advantage of multi-core CPUs. Furthermore, we enhance accessibility by including Python wrappers with example scripts.

This program is principally concerned with computing the dark matter-nucleus differential event rate as a function of the nuclear recoil energy E_R:

$$\frac{dR_D}{dE_R} = N_T \frac{\rho_\chi}{m_\chi} \int_{v_{min}}^{v_{escape}} \frac{2m_T}{4\pi v^2} \frac{1}{2j_\chi + 1} \frac{1}{2j_T + 1} \sum_{spins} |\mathcal{M}(v,q)|^2 \tilde{f}(\vec{v}) v d^3 v$$

This quantity has units of events/GeV and is implicitly multiplied by an effective exposure of 1 Kilogram-Day of target nuclei. This is done by taking $N_t = 1\ kilogram \cdot day/m_T$, where m_T is the mass of the target nucleus in GeV. Recoil energies E_R are given in keV.

2 Python interface

We provide a Python interface (a wrapper) for the Fortran code and a number of example scripts demonstrating its use. The wrapper comes with two Python functions `EventrateSpectra`, and `NucFormFactor` which can be imported from `dmfortfactor.py` in the Python directory.

To include this module in your own code, add the following lines of code:

```
import sys
sys.path.append("../python")
import dmfortfactor as dm
```

Replace "../python" with the path to the `dmfortfactor/python` diretory on your system.

Each function has three required arguments:

1. Z the number of protons in the target nucleus,
2. N the number of neutrons in the target nucleus, and
3. The filename for the one-body density matrix file describing the nuclear structure. If no other arguments are provided, default values will be used for all of the remaining necessary parameters, including zero interaction strength.

2.1 Event rate spectra

To calculate an event rate with a nonzero interaction, the user should also provide one or more of the optional EFT coupling coefficient arrays: `cpvec, cnvec, csvec, cvvec`. These set the couplings to protons, neutrons, isoscalar, and isovector, respectively. The 0^{th} index sets the first operator coefficient: `cpvec[0]`$= c_1^p$, etc. Finally, the user can also pass a dictionary of valid control keywords and values to the function in order to set any of the control words defined in the manual.

To compute the event-rate spectra for ^{131}Xe with a WIMP mass of 50 GeV and a $c_3^v = 0.0048$ coupling, one might call:

```
import dmfortfactor as dm
control_dict = {"wimpmass" : 50.0}
cvvec = np.zeros(15)
cvvec[2] = 0.0048
```

```
Erkev, ER = dm.EventrateSpectra(
            Z = 54,
            N = 77,
            dres = "../dres/xe131gcn",
            controlwords = control_dict,
            cvvec = cvvec,
            exec_path = "../src/dmfortfactor")
```

This will return the differential event rate spectra for recoil energies from 1 keV to 1 MeV in 1 keV steps.

The file `xe131gcn.dres` must be accessible at the relative or absolute path name specified (in this case `../dres/`), and contain a valid one-body density matrix for ^{131}Xe. Similarly, the DMFortFactor executable (`dmfortfactor`) should be accessible from the user's default path - or else the path to the executable should be specified, as in the above example (`exec_path = "../build/dmfortfactor"`).

2.2 Nuclear response functions for WimPyDD

There are published codes which compute the WIMP-nucleus event rate spectra, etc., but which rely on nuclear form factors from an external source. Once such code is WimPyDD (Jeong et al. 2021). We have provided an additional option in DMFortFactor which computes these nuclear form factors from the target density-matrix and exports the results to a data file.

The Python wrapper-function `NucFormFactor` runs the DMFortFactor option to export the nuclear response functions to file, and additionally creates and returns an interpolation function $W(q)$ which can be called. In the following code listing, the nuclear response function for ^{131}Xe is generated for transfer momentum from 0.001 to 10.0 GeV/c.

```
import dmfortfactor as dm
cwords = {
        "usemomentum": 1} # epmin/max/step sets momentum instead of energy
Wfunc = dm.NucFormFactor(
        Z = 54,
        N = 77,
        dres = "../dres/xe131gcn",
        controlwords = cwords,
        epmin = 0.001,
        epmax = 10.0,
        epstep = 0.001)
Wfunc(0.001)
```

The final line returns an (8,2,2)-shaped array with the evaluate nuclear response functions at $q = 0.001$ GeV/c. Note that, had we not set the keyword `usemomentum` to 1, the function input values would have been specified in terms of recoil energy (the default) instead of transfer momentum.

3 Usage guide

DMFortFactor can be used interactively from the command line, where the user is guided by prompts for a small number of datafiles and parameters. Naturally, this interactive process can be expedited by piping a pre-written input file into the command line interface (CLI).

The command line interface (CLI) to the code prompts the user for the type of calculation they wish to perform from a menu of options, then the target nucleus, given by the number of protons Z and neutrons N, two input files, and finally other CLI inputs depending on the compute-option chosen. These final inputs are typically three numbers specifying the range of recoil energies or momentum for which to compute the output. After running, the code prints the results to a plain text file in tabulated format.

The two files are:

1. Control file
2. Nuclear structure input

The control file specifies the EFT interaction and any optional settings desired.

The nuclear structure inputs needed are one-body density matrices, defined below. We supply a library of density matrices for many of the common expected targets, as listed here. The density matrix files are written in plain ASCII, using the format output by the nuclear configuration-interaction code BIGSTICK (Johnson, Ormand, and Krastev 2013; Johnson et al. 2018). The only assumption is that the single-particle basis states are harmonic oscillator states; the user must supply the harmonic oscillator single particle basis frequency Ω, typically given in MeV as $\hbar\Omega$, or the related length parameter $b = \sqrt{\hbar/M\Omega}$, where M is the nucleon mass. [1]

While the DMFortFactor executable can be run by itself, we provide Python APIs for integrating the Fortran program into Python work flows; see section .

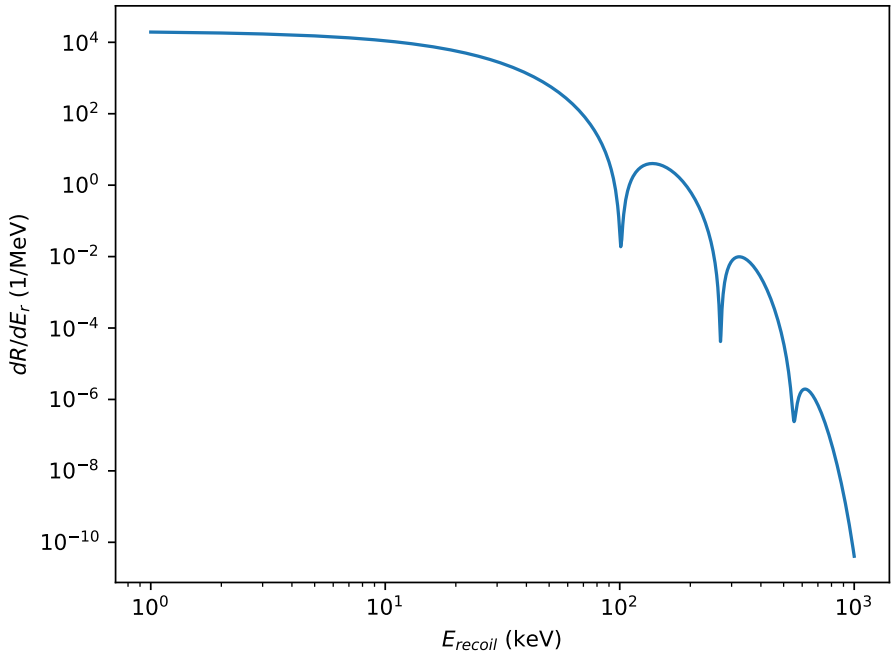
We also provide a generic application programming interface (API) for the Python language. This API essentially offers a prescribed and easy-to-use way to run DMFortFactor in a programmatic way. Any sufficiently experienced linux user could probably write a script to produce any possible set of inputs to our code. But our API removes the need by making it easy for anyone who can use a Python function to write their own advanced scripts allowing them to perform parameter studies and comparisons of different inputs to the theory.

3.1 SuperQuickstart guide

- Navigate to the `src/` directory from wherever you have stored `dmfortfactor/` (e.g. `cd src/`, or `cd ~/Downloads/dmfortfactor/src/`)
- Run the command: `make openmp`
- Navigate to the directory `runs/` (e.g. `cd ../runs/`)
- Run the command: `python3 ../examples/exampleXe.py`

This should generate the following figure:

[1] If density matrices are generated in some other single-particle basis, such as those from a Woods-Saxon potential or a Hartree-Fock calculation, that basis must be expanded into harmonic oscillator states. By using harmonic oscillator basis states one can efficiently compute the matrix elements. One can use either phenomenological or *ab initio* model spaces and interactions; as an example, we provide density matrices for ^{12}C, both from the phenomenological Cohen-Kurath shell model interaction (Cohen and Kurath 1965), and from an interaction derived from chiral effective field theory (Entem and Machleidt 2003).



3.2 Compiling with make

There are multiple directories in the project. All of the Fortran code which needs to be compiled is found in the `src/` directory.

The easiest way to get started is simply to navigate to the `src` directory and run

```
make dmfortfactor
```

This will compile `DMFortFactor` using `gfortran`. If you want to use a different compiler, you must edit the following line in the Makefile:

```
#COMP = <compiler>
COMP = gfortran
```

changing gfortran to your compiler of choice.

If you want a OpenMP parallelized version of the code, you can compile with:

```
make openmp
```

Both of these options will compile the source code and leave the executable, called `dmfortfactor` in the `src` directory. Note that if you change from a serial executable to a parallel executable (or vice versa) you should run:

```
make clean
```

3.3 Required files

There are two files required for any calculation:

1. Control file (.control)
2. Nuclear density matrix file (.dres)

Additionally, if the user enables the option `usenergyfile`, then a file containing the input energies or momentum will also be required.

3.3.1 Control file

Each EFT parameter is written on its own line in [mycontrolfile].control, with four values: the keyword "coefnonrel", the operator number (integer 1..16), the coupling type ("p"=proton, "n"=neutron, "s"=scalar, "v"=vecctor), and the coefficient value. For example,

```
coefnonrel    1    s     3.1
```

would set $c_1^{\tau=0} = 3.1$. We take the isospin convention:

$$c^0 = c^p + c^n$$

$$c^1 = c^p - c^n$$

Thus, the previous example is equivalent to:

```
coefnonrel    1    p     1.55
coefnonrel    1    n     1.55
```

The control file also serves a more general but optional function: to set any parameter in the program to a custom value.
Simply add an entry to the control file with two values: the first should be the keyword for the parameter and the second should be the value to set that parameter to. For example, to set the velocity of the earth in the galactic frame to 240 km/s, you should add the line:

```
vearth  240.0
```

As an example, here is the complete control file used to calculate the event rate for the c_1^n coupling to ^{131}Xe:

```
# Coefficient matrix (non-relativistic)
# Ommitted values are assumed to be 0.0.
# c_i^t
# i = 1,...,16
# t: p=proton n=neutron s=scaler v=vector
coefnonrel  1  n  0.00048
wimpmass 150.0
vearth 232.0
maxwellv0 220.0
dmdens 0.3
usemomentum 0
useenergyfile 0
ntscale  2500.0
printdensities 0
#vescape 550.
```

Uncommenting the last line would set the escape velocity to 550 km/s. A complete list of keywords is given in section **??**.

3.3.2 Nuclear density matrix file (.dres)

We adopt the output format from the { BIGSTICK} shell-model code. The output one-body densities are written to a file with extension { .dres}. We provide a full specification of this plain-text-file format in the { docs} directory. Here, we show the form of the file and explain its contents.

```
 State      E        Ex         J       T
   1   -330.17116   0.00000     1.500  11.500
 Single particle state quantum numbers
```

```
ORBIT      N     L   2 x J
     1     0     2     3
     2     0     2     5
     3     1     0     1
 Initial state #    1 E = -330.17117 2xJ, 2xT =    3  23
 Final state   #    1 E = -330.17117 2xJ, 2xT =    3  23
 Jt =   0, proton      neutron
    1    1   1.55844   5.40558
```

The file is comprised of three sections:

1. many-body state information
2. single-particle state quantum numbers
3. density matrix element blocks

Only the ground state is needed for inelastic WIMP-nucleus scattering calculations. The single-particle state quantum numbers specify the quantum numbers for the simple-harmonic oscillator states involved in the one-body operators.

Finally, the one-body density matrix elements are listed in nested blocks with three layers: i. the initial and final state specification (corresponding to the many-body states listed in section (1) of the file), ii. the angular momentum carried by the one-body density matrix operator, labeled { Jt} here, and iii. the single-particle state labels { a}, { b} in columns 1 and 2 (corresponding to the single-particle state labels listed in section (2) of the file) and the proton and neutron (isospin-0 and isospin-1) density matrix elements in columns 3 and 4.

Both (i) and (ii) must be specified along with columns 1 and 2 of (iii) in order to fully determine a matrix element $\rho_K^{f,i}(a,b)$, where $K = J_t$. Note that the values of K are restricted by conservation of angular momentum; both between the many-body states labeled i and f, and the single-particle states labeled a and b.

3.4 Command-line interface

The program will prompt the user for the minimum necessary inputs to run a calculation with default parameter values, including the name of a control file which contains the EFT coefficients, and optionally, additional changes to the calculation parameters.

After selecting the option {[er]} to compute an event rate spectra, there are six further lines of input. These will be explained by an example:

```
 Enter the target proton number
54
 Enter the target neutron number
77
 Enter name of control file (.control):
xe131
...
  Enter name of one-body density file (.dres)
xe131gcn
...
 What is the range of recoil energies in kev?
 Enter starting energy, stoping energy, step size:
0.0001 250. 1.0
```

The first two entries are self-evident: we specifiy the number of protons and neutrons in the target nucleus. In this case, 54 and 74, respectively, for ^{131}Xe.

Third is the name of the control file containing the EFT coefficients and other, optional, settings. The '.control' file extension should be omitted. This contents of this file will be explained in more detail later.

Fourth is the file containing the nuclear wave functions in the form of one-body density matrices. Only the ground-state need be included. The '.dres' file extension is omitted.

Fifth and finally are three numbers specifying the range of recoil energies E_R that the differential scattering rate should be computed for.

The event rate spectra will be written to a file, and as a side effect of the calculation, the total event rate for the energy range requested will be estimated by numerical integration. Note that the accuracy of this result will depend on the choice of the step size.

3.5 Compute options explained

There are a handful of options available from the main menu of the code:

- [er] Differential event rate, for a range of recoil energies or transfer momenta
- [cs] Differential cross section at a fixed recoil energy over a range of speeds
- [tp] Transition/scattering probability at a fixed recoil energy over a range of speeds
- [te] Total integrated events (without producing spectra file)
- [wd] Nuclear response functions at a given value of $y = (qb/2)^2$
- [ws] Nuclear response function spectra (for a range of q or E)

The string in square brackets [x] is the compute-option. Once a compute-option is chosen, subsequent CLI inputs are the same up until the density matrix file (.dres) has been read-in. Then, the inputs depend on the compute-option chosen.

- [cs] Differential cross section per recoil energy. Four additional inputs:
  1. E-recoil (keV)
  2. v-start (km/s)
  3. v-stop (km/s)
  4. v-step (km/s)
- [tp] Scattering probability. Same as [2].
- [te] Total scattering events per detector (does not produce spectra data). This option uses adaptive quadrature to perform the integral of the event rate spectra with the fewest number of evaluations to reach the desired relative error. This will be much faster than the result from options [1]. Three additional inputs:
  1. E-start (keV)
  2. E-stop (keV)
  3. Desired relative error (decimal value)
- [wd] Nuclear response function test. This compute-option allows the user to evaluate the nuclear response functions $W_i^{x,x'}(y)$ for a provided value of y. All combinations of x and x' will be printed for both isospin and proton-neutron couplings. Two additional inputs are required:
  1. Function number (1 - 8)
  2. Value of $y = (qb/2)^2$ (dimensionless)
- [ws] Nuclear response function spectra. Enter a range of recoil energy or momentum values to evaluate the nuclear response funcions on. Tabulates the data to a file - one momentum per line or energy per line. Momentum/energy is written to the first column. The inputs are:
  1. E-start (keV)
  2. E-stop (keV)
  3. E-step (keV)

The following 32 columns store the (8,2,2)-dimensional response functions $W_i^{x,x'}$:

```
q_1  W_1^00 W_2^00 ... W_8^00 W_1^10 ... W_8^11
q_2  W_1^00 W_2^00 ... W_8^00 W_1^10 ... W_8^11
...
q_m  W_1^00 W_2^00 ... W_8^00 W_1^10 ... W_8^11
```

For the event-rate spectra [er] and for the nuclear response function spectra [ws], the range of values is either over recoil energy E_r (kev) (the default) or over the transfer momentum q (Gev/c). To use q instead of E_r, use the control word { usemomentum} set to 1.

4 Nuclear structure input

Users must provide nuclear one-body density matrix elements of the form:

$$\rho^{\Psi}_{K,T}(a,b) = \langle \Psi | [\hat{c}^{\dagger}_a \hat{c}_b]_{K,T} | \Psi \rangle,$$

where Ψ is the nuclear-target wave function and $\hat{c}^{\dagger}$, $\hat{c}$ are the one-body creation, destruction operators. The matrix elements must be stored in a file in a standard format produced by shell-model codes like BIGSTICK.

Table of nuclear data we include with the program. Each corresponds to a (.dres) density matrix file. The source indicates the nuclear Hamiltonian that was used to generate the wave function data:

Nuclei	Isotopes	Source
Si	28, 29	(Brown and Richter 2006)
Xe	128, 129, 130, 131, 132, 134, 136	
Ar	40	
C	12	(Cohen and Kurath 1965)
He	4	

4.1 Filling core orbitals for phenomenological interactions

Since standard one-body density matrices in phenomenological model spaces contain only matrix elements for orbitals in the valence space, it is necessary to infer the matrix elements for the core orbitals. Our code does this by default, but the user can disable this option using the `fillnuclearcore` control word.

For phenomenological interactions one typically has a 'frozen' core of nucleons which do not participate in the two-body forces of the Hamiltonian. In such cases the single-particle space listed in the .dres file consists only of the valence orbitals and the one-body density matrices are only specified for the valence orbitals.

DMFormFactor reads the valence space orbitals from the .dres file and infers the number of core nucleons by subtracting the number of valence protons and neutrons from the number of nucleons in the target nucleus. The core orbitals are assumed to be one of the standard shell model orbital sets associated with possible cores: He-4, O-16, Ca-40, Ni-56, Sn-100.

The one-body density matrix elements for the core orbitals are then determined from the (full) occupation of the core orbitals. In proton-neutron format:

$$\rho^{\Psi}_{J,x=p,n}(a,b)_{(core)} = \delta_{a,b}[j_a][J],$$

where $[y] \equiv \sqrt{2y+1}$ and j_a is the angular momentum of a-orbit. J is the total spin of the nuclear target state Ψ. And in isospin format for a target state with total isospin T:

$$\rho^{\Psi}_{J,\tau=0}(a,b)_{(core)} = \delta_{a,b}[1/2][j_a][J][T],$$
$$\rho^{\Psi}_{J,\tau=1}(a,b)_{(core)} = 0.0.$$

5 Details of computation

We present the equations necessary to reproduce the code. For a more complete description of the theory, see Phys. Rev. C 89.065501.

5.1 Differential event rate

$$\frac{dR}{dE_r}(E_r) = N_T n_\chi \int_{v_{min}}^{v_{escape}} \frac{d\sigma}{dE_r}(v, E_r)\tilde{f}(\vec{v})vd^3v,$$

where E_r is the recoil energy of the WIMP-nucleus scattering event, N_T is the number of target nuclei, $n_\chi = \rho_\chi/m_\chi$ is the local dark matter number density, σ is the WIMP-nucleus cross section. The dark matter velocity distribution in the lab frame, $\tilde{f}(\vec{v})$, is obtained by boosting the Galactic-frame distribution $f(\vec{v})$: $\tilde{f}(\vec{v}) = f(\vec{v} + \vec{v}_{earth})$, where $\vec{v}_{earth}$ is the velocity of the earth in the galactic rest frame. The simplest model is a three-dimensional Maxwell distribution:

$$f(\vec{v}) \propto e^{-\vec{v}^2/v_0^2},$$

where v_0 is some scaling factor (typically taken to be around 220 km/s).

In order to evaluate the integral over the dark matter distribution, we make the conversion to spherical coordinates. We need to evaluate an integral of the form:

$$I = \int_{v_{min}}^{v_{max}} d^3v \frac{f(\vec{v} + \vec{v}_{earth})}{v} = \int_{v_{min}}^{v_{max}} d^3v \frac{1}{v} e^{-(\vec{v}+\vec{v}_{earth})^2/v_0^2}$$

Noting that $(\vec{v} + \vec{v}_{earth})^2 = \vec{v}^2 + \vec{v}_{earth}^2 + 2vv_{earth}\cos(\theta)$, with $\|\vec{v}\| \equiv v$ and θ defining the angle between the two vectors, it's convenient to make the substitution $d^3v = d\phi d(\cos\theta)v^2dv$:

$$I = \int_0^{2\pi} d\phi \int_{v_{min}}^{v_{max}} dv \int_{-1}^{1} d(\cos\theta) e^{-2vv_{earth}\cos\theta/v_0^2} v^2 \frac{1}{v} e^{-(\vec{v}^2+\vec{v}_{earth}^2)/v_0^2} \tag{1}$$

$$= 2\pi \int_{v_{min}}^{v_{max}} dv v e^{-(\vec{v}^2+\vec{v}_{earth}^2)/v_0^2} \left(-\frac{v_0^2}{2vv_{earth}} e^{-2vv_{earth}\cos\theta/v_0^2} \right)_{-1}^{1} \tag{2}$$

$$= \frac{\pi v_0^2}{v_{earth}} \int_{v_{min}}^{v_{max}} dv e^{-(\vec{v}^2+\vec{v}_{earth}^2)/v_0^2} \left(-e^{-2vv_{earth}/v_0^2} + e^{+2vv_{earth}/v_0^2} \right) \tag{3}$$

$$= \frac{\pi v_0^2}{v_{earth}} \int_{v_{min}}^{v_{max}} dv \left(-e^{(v+v_{earth})^2/v_0^2} + e^{(v-v_{earth})^2/v_0^2} \right) \tag{4}$$

$$= \frac{\pi v_0^2}{v_{earth}} \int_{v_{min}}^{v_{max}} dv \left(g(v - v_{earth}) - g(v + v_{earth}) \right) \tag{5}$$

where in the last equality, we have defined a one-dimensional Gaussian form

$$g(v) \propto e^{-v^2/v_0^2}.$$

The final expression for I can be trivially generalized to other spherically symmetric velocity-dependent forms of the differential cross section. What's important is the reduction of the velocity-boosted d^3v integral to a radial integral which can be carried out with one-dimensional quadrature:

$$\int_{v_{min}}^{v_{max}} d^3v \sigma(v) e^{-(\vec{v}+\vec{v}_{earth})^2/v_0^2} \tag{6}$$

$$= \frac{\pi v_0^2}{v_{earth}} \int_{v_{min}}^{v_{max}} dv \sigma(v) v^2 \left(g(v - v_{earth}) - g(v + v_{earth}) \right). \tag{7}$$

The Fortran code uses this equation to evaluate the event rate integral with quadrature. Analytic solutions exist in the form of error functions; we use quadrature since it makes easy to later modify the velocity distribution (as long as it remains spherically symmetric). For example, adding a velocity cut-off is as easy as changing the limit on the quadrature, with no need to write a whole new subroutine.

5.2 Differential cross section

The differential cross section for the target nucleus can be expressed in terms of either the nuclear recoil energy E_R, or the momentum transfer q:

$$\frac{d\sigma(v, E_R)}{dE_R} = 2m_T \frac{d\sigma(v)(v, \vec{q}^2)}{d\vec{q}^2} = 2m_T \frac{1}{4\pi v^2} T(v, q),$$

Where v is the velocity of the dark matter particles in the lab-frame, q is the momentum transfer of the scattering event, m_T is the mass of the target nucleus, and $T(v, q)$ is the transition or scattering probability. We can see here that the differential cross section has an explicit $1/v^2$ dependence, independent of any velocity dependence of $T(v, q)$.

5.3 Transition probability

The scattering probability is

$$T(v, q) = \frac{1}{2j_\chi + 1} \frac{1}{2j_T + 1} \sum_{spins} |\mathcal{M}(v, q)|^2$$

where j_χ is the spin of the WIMP, j_T is the spin angular momentum of the target nucleus, and $\mathcal{M}$ Galilean invariant amplitude, which is defined by

$$T(v, q) = \frac{4\pi}{2j_T + 1} \frac{1}{(4m_\chi)^2} \sum_{x=p,n} \sum_{x'=p,n}^{1} \sum_{i=1}^{8} R_i^{xx'}(v^2, q^2) W_i^{xx'}(q) \tag{8}$$

where m_χ is the mass of the dark matter particle and x is an index used to sum over isospin couplings. The coefficients $R_i^{x,x'}$ are dark matter particle response functions, to be define in another section. The operators $W_i^{xx'}(q)$ are nuclear response functions, which are sums over matrix elements of nuclear operators constructed from Bessel spherical harmonics and vector spherical harmonics.

5.4 Dark matter response functions

There are 8 dark matter response functions which group 15 operator coefficients c_i^x according the pair of nuclear response functions which they multiply.

As a shorthand, $cl(j) \equiv 4j(j+1)/3$, and $v^{\perp 2} \equiv v^2 - (q/2\mu_t)^2$.

$$R_M^{xx'}(v,q) = \frac{1}{4} cl(j_\chi)[v^{\perp 2}(c_5^x c_5^{x'} q^2 + c_8^x c_8^{x'}) + c_{11}^x c_{11}^{x'} q^2] \tag{9}$$

$$+ (c_1^x + c_2^x v^{\perp 2})(c_1^{x'} + c_2^{x'} v^{\perp 2}) \tag{10}$$

$$R_{\Sigma''}^{xx'}(v,q) = \frac{1}{16} cl(j_\chi)[c_6^x c_6^{x'} q^4 + (c_{13}^x c_{13}^{x'} q^2 + c_{12}^x c_{12}^{x'}) v^{\perp 2} + 2 c_4^x c_6^{x'} q^2 + c_4^x c_4^{x'}] + \frac{1}{4} c_{10}^x c_{10}^{x'} q^2 \tag{11}$$

$$R_{\Sigma'}^{xx'}(v,q) = \frac{1}{32} cl(j_\chi)[2 c_9^x c_9^{x'} q^2 + (c_{15}^x c_{15}^{x'} q^4 + c_{14}^x c_{14}^{x'} q^2 \tag{12}$$

$$- 2 c_{12}^x c_{15}^{x'} q^2 + c_{12}^x c_{12}^{x'}) v^{\perp 2} + 2 c_4^x c_4^{x'}] + \frac{1}{8} (c_3^x c_3^{x'} q^2 + c_7^x c_7^{x'}) v^{\perp 2} \tag{13}$$

$$R_{\Phi''}^{xx'}(v,q) = \frac{q^2}{16 m_N^2} cl(j_\chi)(c_{12}^x - c_{15}^x q^2)(c_{12}^{x'} - c_{15}^{x'} q^2) + \frac{q^4}{4 m_N^2} c_3^x c_3^{x'} \tag{14}$$

$$R_{\tilde{\Phi}'}^{xx'}(v,q) = \frac{q^2}{16 m_N^2} cl(j_\chi)(c_{13}^x c_{13}^{x'} q^2 + c_{12}^x c_{12}^{x'}) \tag{15}$$

$$R_{\Delta}^{xx'}(v,q) = \frac{q^2}{4 m_N^2} cl(j_\chi)(c_5^x c_5^{x'} q^2 + c_8^x c_8^{x'}) + 2 \frac{q^2}{m_N^2} c_2^x c_2^{x'} v^{\perp 2} \tag{16}$$

$$R_{\Delta\Sigma'}^{xx'}(v,q) = \frac{q^2}{4 m_N} cl(j_\chi)(c_4^x c_5^{x'} - c_8^x c_9^{x'}) - \frac{q^2}{m_N} c_2^x c_3^{x'} v^{\perp 2} \tag{17}$$

$$R_{\Phi'' M}^{xx'}(v,q) = \frac{q^2}{4 m_N} cl(j_\chi) c_{11}^x (c_{12}^{x'} - c_{15}^{x'} q^2) + \frac{q^2}{m_N} c_3^{x'} (c_1^x + c_2^x v^{\perp 2}) \tag{18}$$

$$\tag{19}$$

5.5 Cross terms

Previous work has focused on setting limits on a single operator coupling at a time. But of course, multiple couplings may exist simultaneously, and in fact, some nuclear response functions are only activated with specific pairs of EFT coefficients.

To create a minimal list of inputs to validate all possible nonzero couplings, we need to test each coefficient on its own ($i = 1, ..., 15$), and also test the following 9 unique combinations: (1,2), (1,3), (2,3), (4, 5), (5,6), (8,9), (11,12), (11,15), (12,15).

Table of EFT coefficient interactions. Shows which coefficients multiply each coefficient in addition to itself.

Coefficient	Couples to
1	2, 3
2	1, 3
3	1, 2
4	5, 6
5	4
6	4
7	
8	9
9	8
10	
11	12, 15
12	11, 15
13	
14	
15	11, 12

5.6 Operators

The code uses the EFT coefficients in explicit proton-neutron couplings, i.e. the interaction is defined by:

$$\mathcal{H} = \sum_{x=p,n} \sum_{i=1,15} c_i^x \mathcal{O}_i$$

and the 15 momentum-dependent operators are:

$$\mathcal{O}_1 = 1_\chi 1_N \tag{20}$$
$$\mathcal{O}_2 = (v^\perp)^2 \tag{21}$$
$$\mathcal{O}_3 = i\vec{S}_N \cdot \left(\frac{\vec{q}}{m_N} \times \vec{v}^\perp \right) \tag{22}$$
$$\mathcal{O}_4 = \vec{S}_\chi \cdot \vec{S}_N \tag{23}$$
$$\mathcal{O}_5 = i\vec{S}_\chi \cdot \left(\frac{\vec{q}}{m_N} \times \vec{v}^\perp \right) \tag{24}$$
$$\mathcal{O}_6 = \left(\vec{S}_\chi \cdot \frac{\vec{q}}{m_N} \right) \left(\vec{S}_N \cdot \frac{\vec{q}}{m_N} \right) \tag{25}$$
$$\mathcal{O}_7 = \vec{S}_N \cdot \vec{v}^\perp \tag{26}$$
$$\mathcal{O}_8 = \vec{S}_\chi \cdot \vec{v}^\perp \tag{27}$$
$$\mathcal{O}_9 = i\vec{S}_\chi \cdot \left(\vec{S}_N \times \frac{\vec{q}}{m_N} \right) \tag{28}$$
$$\mathcal{O}_{10} = i\vec{S}_N \cdot \frac{\vec{q}}{m_N} \tag{29}$$
$$\mathcal{O}_{11} = i\vec{S}_\chi \cdot \frac{\vec{q}}{m_N} \tag{30}$$
$$\mathcal{O}_{12} = \vec{S}_\chi \cdot \left(\vec{S}_N \times \vec{v}^\perp \right) \tag{31}$$
$$\mathcal{O}_{13} = i \left(\vec{S}_\chi \cdot \vec{v}^\perp \right) \left(\vec{S}_N \cdot \frac{\vec{q}}{m_N} \right) \tag{32}$$
$$\mathcal{O}_{14} = i \left(\vec{S}_\chi \cdot \frac{\vec{q}}{m_N} \right) \left(\vec{S}_N \cdot \vec{v}^\perp \right) \tag{33}$$
$$\mathcal{O}_{15} = - \left(\vec{S}_\chi \cdot \frac{\vec{q}}{m_N} \right) \left(\left(\vec{S}_N \times \vec{v}^\perp \right) \cdot \frac{\vec{q}}{m_N} \right) \tag{34}$$

5.7 Nuclear response functions

The EFT physics has been grouped into eight WIMP response functions $R_i^{x,x'}$, and eight nuclear response functions $W_i^{x,x'}$. The first six nuclear response functions have the following form:

$$W_X^{x,x'} = \sum_J \langle \Psi | \, X_J^x \, | \Psi \rangle \, \langle \Psi | \, X_J^{x'} \, | \Psi \rangle \,,$$

with X selecting one of the six electroweak operators,

$$X_J = M_J, \Delta_J, \Sigma'_J, \Sigma''_J, \tilde{\Phi}'_J, \Phi''_J,$$

and Ψ being the nuclear wave function for the ground state of the target nucleus. The sum over operators spins J is restricted to even or odd values of J, depending on restrictions from conservation of parity and charge conjugation parity (CP) symmetry.

Two additional response functions add interference-terms:

$$W_{M\Phi''}^{x,x'} = \sum_J \langle\Psi|\, M_J^x \,|\Psi\rangle \langle\Psi|\, \Phi_J^{''x'} \,|\Psi\rangle\,,$$

$$W_{\Delta\Sigma'}^{x,x'} = \sum_J \langle\Psi|\, \Sigma_J^{'x} \,|\Psi\rangle \langle\Psi|\, \Delta_J^{x'} \,|\Psi\rangle\,.$$

The indices i in equation (**??**) correspond to these operators as: $i \to X$ for $i = 1, .., 6$, and $i = 7 \to M\Phi''$, $i = 8 \to \Delta\Sigma'$.

DMFortFactor can print the nuclear form factors to a file over a range of either transfer momenta or recoil energy.

5.8 Nuclear (electroweak) operators

There are six parity-and-CP-conserving nuclear operators, $M_J, \Delta_J, \Sigma'_J, \Sigma''_J, \tilde{\Phi}'_J, \Phi''_J$, describing the electro-weak coupling of the WIMPs to the nucleon degrees of freedom. These are constructed from Bessel spherical and vector harmonics (Donnelly and Haxton 1979):

$$M_{JM}(q\vec{x}) \equiv j_J(qx) Y_{JM}(\Omega_x) \tag{35}$$

$$\vec{M}_{JML}(q\vec{x}) \equiv j_L(qx) \vec{Y}_{JLM}(\Omega_x), \tag{36}$$

where, using unit vectors $\vec{e}_{\lambda=-1,0,+1}$,

$$Y_{JLM}(\Omega_x) = \sum_{m\lambda} \langle Lm1\lambda|(L1)JM_J\rangle\, Y_{Lm}(\Omega_x)\vec{e}_\lambda. \tag{37}$$

The six multipole operators are defined as:

$$M_{JM} \tag{38}$$

$$\Delta_{JM} \equiv \vec{M}_{JJM} \cdot \frac{1}{q}\vec{\nabla} \tag{39}$$

$$\Sigma'_{JM} \equiv -\,i \left\{ \frac{1}{q}\vec{\nabla} \times \vec{M}_{JJM} \right\} \cdot \vec{\sigma} \tag{40}$$

$$\Sigma''_{JM} \equiv \left\{ \frac{1}{q}\vec{\nabla} M_{JM} \right\} \cdot \vec{\sigma} \tag{41}$$

$$\tilde{\Phi}'_{JM} \equiv \left(\frac{1}{q}\vec{\nabla} \times \vec{M}_{JJM} \right) \cdot \left(\vec{\sigma} \times \frac{1}{q}\vec{\nabla} \right) + \frac{1}{2}\vec{M}_{JJM} \cdot \vec{\sigma} \tag{42}$$

$$\Phi''_{JM} \equiv i \left(\frac{1}{q}\vec{\nabla} M_{JM} \right) \cdot \left(\vec{\sigma} \times \frac{1}{q}\vec{\nabla} \right) \tag{43}$$

The matrix elements of these operators can be calculated for standard wave functions from second-quantized shell model calculations:

$$\begin{aligned} \langle\Psi_f|\, X_J \,|\Psi_i\rangle &= \mathrm{Tr}\Big(X_J \rho_J^{f,i} \Big) \\ &= \sum_{a,b} \langle a|\, |X_J|\, |b\rangle\, \rho_J^{fi}(ab), \end{aligned}$$

where single-particle orbital labels a imply shell model quantum number n_a, l_a, j_a, and the double-bar $||$ indicates reduced matrix elements (Edmonds 1996). For elastic collisions, only the ground state is involved, i.e. $\Psi_f = \Psi_i = \Psi_{g.s.}$.

We assume a harmonic oscillator single-particle basis, with the important convention that the radial nodal quantum number n_a starts at 0, that is, we label the orbitals as $0s, 0p, 1s0d$, etc.., and *not* starting with

$1s, 1p$, etc. Then, the one-body matrix elements for operators $\langle a|\,|X_J^{(f)}|\,|b\rangle$, built from spherical Bessel functions and vector spherical harmonics, have closed-form expressions in terms of confluent hypergeometric functions (Donnelly and Haxton 1979).

The nuclear structure input is in the form of one-body density matrices between many-body eigenstates,

$$\rho_J^{fi}(ab) = \frac{1}{\sqrt{2J+1}}\langle\Psi_f||[\hat{c}_a^\dagger \otimes \tilde{c}_b]_J||\Psi_i\rangle,$$

where $\hat{c}_a^\dagger$ is the fermion creation operator (with good angular momentum quantum numbers), $\tilde{c}_b$ is the time-reversed (Edmonds 1996) fermion destruction operator. Here the matrix element is reduced in angular momentum but not isospin, and so are in proton-neutron format. These density matrices are the product of a many-body code, in our case BIGSTICK Johnson et al. (2018), although one could use one-body density matrices, appropriately formatted, from any many-body code.

5.9 Electroweak matrix elements

To compute the matrix elements of the electroweak operators in a harmonic oscillator basis, we use the derivations from (Donnelly and Haxton 1979). Namely, equations (1a) - (1f) and (3a) - (3d), which express the necessary geometric matrix elements in terms of matrix elements of the spherical Bessel functions. Here, we write out the remaining explicit formulas for obtaining matrix elements of the Bessel functions $j_L(y)$ in a harmonic oscillator basis in terms of the confluent hypergeometric function:

$${}_1F_1(a,b,z) = \sum_{n=0}^{\infty}\frac{a^{(n)}z^n}{b^{(n)}n!},$$

which makes use of the rising factorial function:

$$m^{(n)} = \frac{(m+n-1)!}{(m-1)!}.$$

The first additional relation is:

$$\begin{aligned}\langle n'l'j'|\,j_L(y)\,|nlj\rangle = \frac{2^L}{(2L+1)!!}y^{L/2}e^{-y}\sqrt{(n'-1)!(n-1)!}\sqrt{\Gamma(n'+l'+1/2)\Gamma(n+l+1/2)}\\ \times\sum_{m=0}^{n-1}\sum_{m'=0}^{n'-1}\frac{(-1)^{m+m'}}{m!m'!(n-m-1)!(n'-m'-1)!}\\ \times\frac{\Gamma[(l+l'+L+2m+2m'+3)/2]}{\Gamma(l+m+3/2)\Gamma(l'+m'+3/2)}\;{}_1F_1[(L-l'-l-2m'-2m)/2;L+3/2;y],\end{aligned}$$

which is computed in DMFortFactor by the function `BesselElement`.

The two additional relations are needed. As computed by `BesselElementMinus`:

$$\begin{aligned}\langle n'l'j'|\,j_L(y)(\frac{d}{dy}-\frac{l}{y})\,|nlj\rangle = \frac{2^(L-1)}{(2L+1)!!}y^{(L-1)/2}e^{-y}\sqrt{(n'-1)!(n-1)!}\sqrt{\Gamma(n'+l'+1/2)\Gamma(n+l+1/2)}\\ \times\sum_{m=0}^{n-1}\sum_{m'=0}^{n'-1}\frac{(-1)^{m+m'}}{m!m'!(n-m-1)!(n'-m'-1)!}\frac{\Gamma[(l+l'+L+2m+2m'+2)/2]}{\Gamma(l+m+3/2)\Gamma(l'+m'+3/2)}\\ \times\Big\{-\frac{1}{2}(l+l'+L+2m+2m'+2)\;{}_1F_1[(L-l'-l-2m'-2m-1)/2;L+3/2;y]\\ +2m\;{}_1F_1[(L-l'-l-2m'-2m+1)/2;L+3/2;y]\Big\}.\end{aligned}$$

As computed by `BesselElementPlus`:

$$\langle n'l'j'|\, j_L(y)(\frac{d}{dy}+\frac{l}{y})\,|nlj\rangle = \frac{2^(L-1)}{(2L+1)!!}y^{(L-1)/2}e^{-y}\sqrt{(n'-1)!(n-1)!}\sqrt{\Gamma(n'+l'+1/2)\Gamma(n+l+1/2)}$$
$$\times\sum_{m=0}^{n-1}\sum_{m'=0}^{n'-1}\frac{(-1)^{m+m'}}{m!m'!(n-m-1)!(n'-m'-1)!}\frac{\Gamma[(l+l'+L+2m+2m'+2)/2]}{\Gamma(l+m+3/2)\Gamma(l'+m'+3/2)}$$
$$\times\Big\{-\frac{1}{2}(l+l'+L+2m+2m'+2)\ {}_1F_1[(L-l'-l-2m'-2m-1)/2;L+3/2;y]$$
$$+(2l+2m+1)\ {}_1F_1[(L-l'-l-2m'-2m+1)/2;L+3/2;y]\Big\}.$$

5.10 Wigner vector coupling functions

We implement a standard set of functions and subroutines for computing the vector-coupling 3-j, 6-j, and 9-j symbols using the Racah alebraic expressions (Edmonds 1996).

One method we use to improve compute time is to cache Wigner 3-j and 6-j symbols~[**?**] (used to evaluate electro-weak matrix elements) in memory at the start of run-time. As a side effect, our tests show that this adds a constant compute time to any given calculation of roughly 0.3 seconds in serial execution and uses roughly 39 MB of memory (for the default table size). As a point of comparison, the ^{131}Xe example with all-nonzero EFT coefficients has a run-time of 30 seconds in parallel execution. If we disable the table caching, the run-time is roughly 150 seconds, 5 times longer. The size of the table stored in memory can be controlled via the control file with the keywords `sj2tablemin` and `sj2tablemax`.

For the 3-j symbol, we use the relation to the Clebsh-Gordon vector-coupling coefficients:

$$\begin{pmatrix} j_1 & j_2 & J \\ m_1 & m_1 & M \end{pmatrix} = (-1)^{j_1-j_2-M}(2J+1)^{-1/2}$$
$$(j_1 j_2 m_1 m_2|j_1 j_2; J, -M).$$

The vector coupling coefficients are computed as:

$$(j_1 j_2 m_1 m_2|j_1 j_2; J, M) = \delta(m_1+m_1, m)(2J+1)^{1/2}\Delta(j_1 j_2 J)$$
$$\times[(j_1+m_1)(j_1-m_1)(j_2+m_2)(j_2-m_2)(J+M)(J-M)]^{1/2}\sum_z(-1)^z\frac{1}{f(z)},$$

where

$$f(z) = z!(j_1+j_2-J-z)!(j_1-m_2-z)!$$
$$\times(j_2+m_2-z)!(J-j_2+m_1+z)!(J-m_1-m_2+z)!,$$

and

$$\Delta(abc) = \left[\frac{(a+b-c)!(a-b+c)!(-a+b+c)!}{(a+b+c+1)!}\right]^{1/2}.$$

The sum over z is over all integers such that the factorials are well-defined (non-negative-integer arguments).

Similarly, for the 6-j symbols:

$$\begin{Bmatrix} j_1 & j_2 & j_3 \\ m_1 & m_1 & m_3 \end{Bmatrix} = \Delta(j_1 j_2 j_3)\Delta(j_1 m_2 m_3)\Delta(m_1 j_2 m_3)$$
$$\times\Delta(m_1 m_2 j_3)\sum_z(-1)^z\frac{(z+1)!}{g(z)},$$

with

$$g(z) = (\alpha - z)!(\beta - z)!(\gamma - z)! \times (z - \delta)!(z - \epsilon)!(z - \zeta)!(z - \eta)!$$

$$\alpha = j_1 + j_1 + m_1 + m_2 \qquad \beta = j_2 + j_3 + m_2 + m_3$$
$$\gamma = j_3 + j_1 + m_3 + m_1$$
$$\delta = j_1 + j_2 + j_3 \qquad \epsilon = j_1 + m_2 + m_3$$
$$\zeta = m_1 + j_2 + m_3 \qquad \eta = m_1 + m_2 + j_3.$$

For the 9-j symbol, we use the relation to the 6-j symbol:

$$\begin{Bmatrix} j_1 & j_2 & j_3 \\ j_4 & j_5 & j_6 \\ j_7 & j_8 & j_9 \end{Bmatrix} = \sum_k (-1)^{2k}(2k+1) \times \begin{Bmatrix} j_1 & j_4 & j_7 \\ j_8 & j_9 & z \end{Bmatrix} \begin{Bmatrix} j_2 & j_5 & j_8 \\ j_4 & z & j_6 \end{Bmatrix} \begin{Bmatrix} j_3 & j_6 & j_9 \\ z & j_1 & j_2 \end{Bmatrix}.$$

The 6-j symbols used to calculate the 9-j symbol are first taken from any tabulated values. Otherwise, they are computed as previously described.

6 Control file keywords

Keyword	Symbol	Meaning	Units	Default
dmdens	ρ_χ	Local dark matter density.	GeV/cm^3	0.3
dmspin	j_χ	Instrinsic spin of WIMP particles.	$\hbar$	$\frac{1}{2}$
fillnuclearcore		Logical flag (enter 0 for False, 1 for True) to fill the inert-core single-particle orbitals in the nuclear level densities. Phenomenological shell model calculations typically provide only the density matrices for the active valence-space orbitals, leaving it to the user to infer the core-orbital densities. This option automatically assigns these empty matrix elements assuming a totally filled core.		1 (true)
gaussorder		Order of the Gauss-Legendre quadrature to use when using Type 2 quadrature. (See quadtype.) An n-th order routine will perform n function evaluations. Naturally, a higher order will result in higher precision, but longer compute time.		12
hofrequency	$\hbar\omega$	Set the harmonic oscillator length by specifying the harmonic oscillator frequency. (b = 6.43/sqrt($\hbar\omega$)). If using an *ab initio* interaction, $\hbar\omega$ should be set to match the value used in the interaction.	MeV	See hoparameter.
hoparameter	b	Harmonic oscillator length. Determines the scale of the nuclear wavefunction interaction.	fm	See eqn. (**??**).
maxwellv0	v_0	Maxwell-Boltzman velocity distribution scaling factor.	km/s	220.0
mnucleon	m_N	Mass of a nucleon. It's assumed that $m_p \approx m_n$.		
GeV	0.938272			
ntscale	N_t	Effective number of target nuclei scaling factor. The differential event rate is multiplied by this constant in units of kilogram-days. For example, if the detector had a total effective exposure of 2500 kg days, one might enter 2500 for this value.	kg days	1.0

Keyword	Symbol	Meaning	Units	Default
quadrelerr		Desired relative error for the adaptive numerical quadrature routine (quadtype 1).		10^{-6}
quadtype		Option for type of numerical quadrature. (Type 1 = adaptive 8th order Gauss-Legendre quadrature. Type 2 = static n-th order Gauss-Legendre quadrature.)		1 (type 1)
sj2tablemax		Maximum value of $2 \times J$ used when caching Wigner 3-J and 6-J functions into memory.		12
sj2tablemin		Minimum value of $2 \times J$ used when caching Wigner 3-J and 6-J functions into memory.		-2
useenergyfile		Logical flag (enter 0 for False, 1 for True) to read energy grid used for calculation from a user-provided file intead of specifying a range.		0 (false)
usemomentum		Logical flag (enter 0 for False, 1 for True) to use momentum transfer intead of recoil energy as the independent variable.		0 (false)
vearth	v_{earth}	Speed of the earth in the galactic frame.	km/s	232.0
vescape	v_{escape}	Galactic escape velocity. Particles moving faster than this speed will escape the galaxy, thus setting an upper limit on the WIMP velocity distribution.	km/s	12 $\times v_{scale}$
weakmscale	m_v	Weak interaction mass scale. User defined EFT coefficients are divided by m_v^2.	GeV	246.2
wimpmass	m_χ	WIMP particle mass.	GeV	50.0

7 References

Anand, Nikhil, A. Liam Fitzpatrick, and W. C. Haxton. 2014. "Weakly Interacting Massive Particle-Nucleus Elastic Scattering Response." *Phys. Rev. C* 89 (June): 065501. https://doi.org/10.1103/PhysRevC.89.065501.

Brown, B. Alex, and W. A. Richter. 2006. "New 'USD' Hamiltonians for the *sd* Shell." *Phys. Rev. C* 74 (September): 034315. https://doi.org/10.1103/PhysRevC.74.034315.

Cohen, S, and D Kurath. 1965. "Effective Interactions for the 1p Shell." *Nuclear Physics* 73 (1): 1–24.

Donnelly, T. W., and W. C. Haxton. 1979. "Multipole Operators in Semileptonic Weak and Electromagnetic Interactions with Nuclei: Harmonic Oscillator Single-Particle Matrix Elements." *Atomic Data and Nuclear Data Tables* 23 (2): 103–76. https://doi.org/https://doi.org/10.1016/0092-640X(79)90003-2.

Edmonds, Alan Robert. 1996. *Angular Momentum in Quantum Mechanics*. Princeton University Press.

Entem, D. R., and R. Machleidt. 2003. "Accurate Charge-Dependent Nucleon-Nucleon Potential at Fourth Order of Chiral Perturbation Theory." *Phys. Rev. C* 68 (October): 041001. https://doi.org/10.1103/PhysRevC.68.041001.

Fitzpatrick, A. Liam, Wick Haxton, Emanuel Katz, Nicholas Lubbers, and Yiming Xu. 2013. "The Effective Field Theory of Dark Matter Direct Detection." *Journal of Cosmology and Astroparticle Physics* 2013 (02): 004–4. https://doi.org/10.1088/1475-7516/2013/02/004.

Jeong, Injun, Sunghyun Kang, Stefano Scopel, and Gaurav Tomar. 2021. "WimPyDD: An Object-Oriented Python Code for the Calculation of WIMP Direct Detection Signals." *arXiv Preprint arXiv:2106.06207*.

Johnson, Calvin W., W. Erich Ormand, and Plamen G. Krastev. 2013. "Factorization in Large-Scale Many-Body Calculations." *Computer Physics Communications* 184: 2761–74.

Johnson, Calvin W., W. Erich Ormand, Kenneth S. McElvain, and Hongzhang Shan. 2018. "BIGSTICK: A Flexible Configuration-Interaction Shell-Model Code." https://arxiv.org/abs/1801.08432v1.

8 Update log

Oliver Gorton, Changfeng Jiao, and Calvin Johnson

8.0.1 Version 1.10 update (Jan. 12, 2021)

- Validation test scripts
- Improved organization of code
- Comprehensive documentation

8.0.2 Version 1.7 update (May 5, 2021)

- Improved speed by re-writing Gaussian-Legendre routine in modern fortran with fixed (non-adaptive) number of evaluation points. The new implementation also supports parallelization, which the previously referenced library did not.

8.0.3 Version 1.6 update (Apr. 19, 2021)

- Improved speed by caching Wigner coefficients in memory
- Added options to compute transition probabilities and differential cross sections. (For fixed recoil energy, for a range of velocities.)
- More options in the Makefile
- Previous versions claimed compatibility with isospin-formalism density matrices. This turns out not to be the case. An appropriate error trap has been added.

8.0.4 Version 1.5 update (Feb. 22, 2021)

- Renamed executable to dmf90factor.x (previously darkmatter.x)
- A ".sps" file is no longer required; the code now deduces this information using data provided by the (still required) ".dres" file.
- Manual now has quick-start guides for the Fortran and Python interfaces

8.0.5 Version 1.4 update (Jan. 14, 2021)

New features: * Compute integrated event rate spectra (total events) using adaptive integration routine * Computing an event rate spectra will also report the total integrated event rate * EFT coefficients can now be provided as either proton/neutron couplings, or as scaler/vector isospin couplings

Bugfixes: * Updated definition of proton-neutron to isospin transformation to be consistent with Mathematica script definition (script, not paper) ### Version 1.3 update (Jan. 13, 2021) New features: * ~~Now supports nuclear density matrix files in either isospin or proton-neutron formalism~~ *see Version 1.6 notes.* * Inputs and outputs now carry specified units

Bugfixes: * Fixed bug involving illegal sqrt() evaluations * Updated numerical quadrature routine to library (instead of 'homebrew') * Fixed error in denisty-matrix core-filler

8.0.6 Version 1.2 update (Nov. 24, 2020)

- Data reorganized to support future extension to multiple target species.
- Added python script which easily compares event rate spectra for different dark matter masses. To run the example, cd to sample/ and run:

  ```
  python ../python/masscompare.py
  ```

  Script is general and can be used to compare runs for any variable which can be modified in the control file.

8.0.7 Version 1.1 update (Nov. 20, 2020)

- Now supports computing event rate spectra (event rate versus recoil energy in kev)
- Energy range is entered either (a) as a linear grid by specifying Emin, Emax, Estep, or (b) from a file specifying energies

- Now takes advantage of multi-core systems using openMP

8.0.8 Developer contacts

- Calvin Johnson (PI) cjohnson@sdsu.edu
- Oliver Gorton (Grad. student) ogorton@sdsu.edu