

CDMS EFT code release guide

Introduction

This guide describes the code and data released in association with the recent SuperCDMS Effective Field Theory paper (arXiv:1503.03379). The code provided in this package is a MATLAB implementation of the effective field theory physics discussed in arXiv:1308.6288, 1211.2818, and 1203.3542. A python implementation of this code can also be found at https://bitbucket.org/kristi_schneck/darkmatter_eft/. The MATLAB implementation of the EFT code consists of numerous components:

- `drdq_EFT`
- `dsigma_dq_EFT_total`
- `getNuclearResponse`
- `getWIMPresponse`
- `getIsotopes`
- Helper functions: `getNucResponseLookup`, `getVelocityIntegrand`, `getWIMPresponse_integrand`

Note that you may have to change the path names in `getNucResponseLookup` and `getNuclearResponse` to your local copy of the code.

Code documentation

Dark Matter rate calculation – `drdq_EFT`

This is the main user function for calculating dark matter interaction rates. It takes seven arguments as input:

1. `E_recoil`: Recoil energy in keV at which to evaluate the function
2. `Target`: target element (currently ‘Ge’, ‘Xe’, ‘Si’, ‘F’, ‘I’, and ‘Na’ are supported)
3. `M`: dark matter mass in GeV

4. Jwimp: dark matter spin
5. c0: vector of 16 interaction coefficients for isoscalar interaction ($c_i^0 = \frac{1}{2}(c_i^p + c_i^n)$)
6. c1: vector of 16 interaction coefficients for isovector interaction ($c_i^1 = \frac{1}{2}(c_i^p - c_i^n)$)
7. vdf: dark matter velocity distribution function (currently ‘Maxwellian’ and ‘Mao_etal’ are supported)

The input coefficients c0 and c1 are dimensionless; units are restored within the function `getWIMPresponse`.

This function calculates the differential cross section for the specified dark matter parameters using the function `dsigma_dq_EFT_total` and integrates it over the specified velocity distribution to produce the differential event rate. The standard dark matter halo parameters are used for the ‘Maxwellian’ case, and for the ‘Mao_etal’ case, the parameters are set to $p = 2.7$ and $v_0/v_{esc} = .6875$ (these parameters are hard-coded into the function and can be modified only when using the option to compute the integral over velocity on the fly).

This function has two options for performing the integral over the velocity distribution: the integral can be performed numerically, or lookup tables can be used for speed. These options are controlled by a logic variable, `lgc_integral`. Note that if you wish to change the parameters of the alternate velocity distribution function, the slower numerical integral option must be used. The lookup table option uses two helper functions, `getVelocityIntegrand` and `getWIMPresponse_integrand`, to separate out the velocity dependence from the differential cross section.

Differential cross section – `dsigma_dq_EFT_total`

This function calculates the differential cross section for the specified dark matter parameters. `dsigma_dq_EFT` takes eight input arguments (the ninth is optional):

1. E_recoil: Recoil energy in keV at which to evaluate the function
2. vWIMP: dark matter velocity in km/s
3. Target: target element (currently ‘Ge’, ‘Xe’, ‘Si’, ‘F’, ‘I’, and ‘Na’ are supported)
4. M: dark matter mass in GeV
5. Jwimp: dark matter spin
6. Interaction: type of dark matter interaction (‘M’, ‘PhiPrimePrime’, ‘PhiTilde’, ‘Delta’, ‘MPhiPrimePrime’, ‘SigmaPrime’, ‘SigmaPrimePrime’, ‘SigmaPrimeDelta’)
7. c0: vector of 16 interaction coefficients for isoscalar interaction ($c_i^0 = \frac{1}{2}(c_i^p + c_i^n)$)
8. c1: vector of 16 interaction coefficients for isovector interaction ($c_i^1 = \frac{1}{2}(c_i^p - c_i^n)$)

9. `NucResponseLookup` (optional, MATLAB only): lookup cell with nuclear response parameters, generated by `getNucResponseLookup`. If the optional `NucResponseLookup` argument is not specified, the nuclear response lookup is generated at the start of the script using the function `getNucResponseLookup`.

Nuclear response function – `getNuclearResponse`

This function calculates the nuclear response for the given interaction and target nucleus using the provided lookup cell, `NucResponseLookup`. In the absence of this argument, it calculates the response from lookup files located in the directory `Nuclear_Response_LOOKUPS/`. The lookup tables are generated from the Mathematica package described in arXiv:1308.6288 using a script provided by Liam Fitzpatrick.

The function takes five input arguments (the sixth is optional) :

1. `E_recoil`: Recoil energy in keV at which to evaluate the function
2. `Target`: target element (currently ‘Ge’, ‘Xe’, ‘Si’, ‘F’, ‘I’, and ‘Na’ are supported)
3. `Interaction`: type of dark matter interaction (‘M’, ‘PhiPrimePrime’, ‘PhiTilde’, ‘Delta’, ‘MPhiPrimePrime’, ‘SigmaPrime’, ‘SigmaPrimePrime’, ‘SigmaPrimeDelta’)
4. `tau1`: parameter indicating whether first isospin parameter is isoscalar ($\tau1 = 0$) or isovector ($\tau1 = 1$)
5. `tau2`: parameter indicating whether second isospin parameter is isoscalar ($\tau2 = 0$) or isovector ($\tau2 = 1$)
6. `NucResponseLookup` (optional): lookup cell with nuclear response parameters, generated by `getNucResponseLookup`

If the optional `NucResponseLookup` argument is not included, the function loads the appropriate responses from `Nuclear_Response_LOOKUPS/`. Before using this function, you should change the path names to your local installation of the EFT code.

WIMP response function – `getWIMPresponse`

This function calculates the WIMP response function for the given interaction.

The function takes eight input arguments:

1. `E_recoil`: Recoil energy in keV at which to evaluate the function
2. `vWIMP`: dark matter velocity in km/s
3. `Target`: target element (currently ‘Ge’, ‘Xe’, ‘Si’, ‘F’, ‘I’, and ‘Na’ are supported)
4. `ci`: vector of 16 interaction coefficients corresponding to `tau1`

5. `ciPrime`: vector of 16 interaction coefficients corresponding to `tau2`
6. `M`: dark matter mass in GeV
7. `Jwimp`: dark matter spin
8. `Interaction`: type of dark matter interaction ('M', 'PhiPrimePrime', 'PhiTilde', 'Delta', 'MPhiPrimePrime', 'SigmaPrime', 'SigmaPrimePrime', 'SigmaPrimeDelta')

Isotope lookup table – `getIsotopes`

This function contains the relevant properties of the target elements supported by the EFT code. It takes a single argument, the name of the target element. Currently 'Ge', 'Xe', 'Si', 'F', 'I', and 'Na' are supported.

Nuclear response helper function – `getNucResponseLookup`

This function creates a lookup cell that can be passed as an optional argument to any of the EFT code functions in the MATLAB implementation. It was created to speed up the code by avoiding unnecessary calls to load the `.mat` lookup files. Before using the code, you will have to change the path name `ingetNucResponseLookup` to your local copy of the code.

Velocity integral helper functions

The functions `getVelocityIntegrand` and `getWIMPresponse_integrand` output the velocity-independent portions of the differential cross section. The WIMP responses are either independent of velocity or proportional to velocity squared, so these options are separated out. The integrals for the velocity-independent and velocity-squared terms are computed separately and stored in `.mat` files. When making a call to `drdq_EFT`, the code performs a linear interpolation between points in the lookup table and multiplies it by the output of `getVelocityIntegrand`.

There is typically no reason to use these two functions outside the lookup table algorithm implemented in the rate calculation.

Example plotting script

An example plotting script, `example_plot.m`, is also provided. Running this script produces the plot shown in Fig. 1. This script calculates the differential event rate for four isoscalar EFT operators and plots the results. Since the event rate can vary by many orders of magnitude depending on the operator by which the interaction proceeds, the four event rates are scaled so that 10 events are observed over the given energy range for each curve.

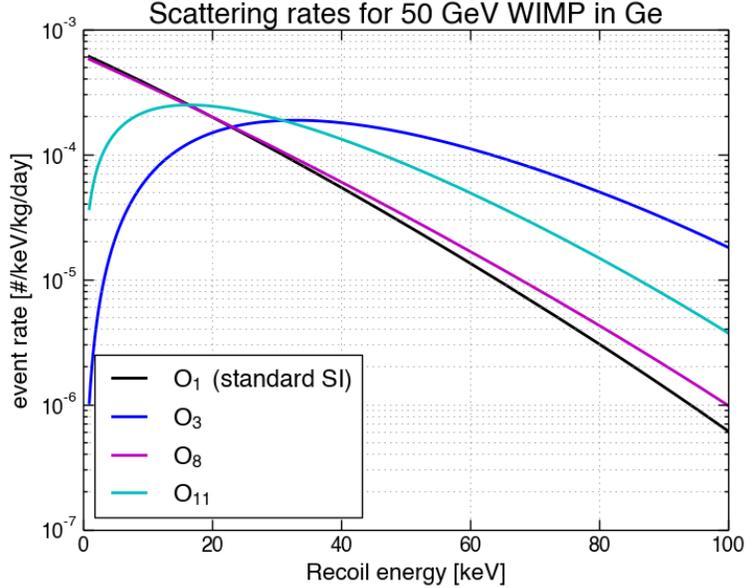


Figure 1: Example rate plot for 50 GeV WIMP in germanium, created by provided plotting script.

Principal component analysis scripts

This package also includes scripts which perform the principal component analysis discussed in section IV of arXiv:1503.03379. The primary function is `get_PCA_vectors.m`, which outputs a file containing the eigenvectors and eigenvalues for the specified WIMP interaction specified by the inputs. A helper function, `get_PCA_Amplitude.m`, provides the amplitudes for each entry in the principal component matrix (it is unlikely that this function will need to be used outside `get_PCA_vectors.m`). The inputs to `get_PCA_vectors.m` are:

1. `E_recoil`: Recoil energy in keV at which to evaluate the interference. If an array is specified, the function sums over the specified energies.
2. `vWIMP`: dark matter velocity in km/s
3. `Target`: target element ('Ge', 'Xe', 'Si', 'F', 'I', or 'Na')
4. `M`: dark matter mass in GeV
5. `Jwimp`: dark matter spin
6. `EFTindex1`, `EFTindex2`: pair of operators which undergo interference. The interfering pairs are discussed in arXiv:1503.03379 and the Fitzpatrick et al. source papers.
7. `filename`: name of the .mat file that saves the PCA eigenvalues and eigenvectors

`get_PCA_vectors.m` saves the output matrices of eigenvalues and eigenvectors for the principal component analysis to a `.mat` file specified by the ‘filename’ argument. Also saved in this file are the interaction parameters (recoil energy, WIMP mass, spin, and velocity, and EFT indices). The i^{th} column of the eigenvectors matrix gives the EFT coupling coefficients for the corresponding i^{th} eigenvalue in the eigenvalue matrix. The order of the coefficients in the eigenvector is $[c_i^0, c_i^1, c_j^0, c_j^1]$, where 0 and 1 correspond to isoscalar and isovector, and i and j correspond to EFTindex1 and EFTindex2 specified in the input to `get_PCA_vectors.m`.

Two example scripts are also included that demonstrate how to use the results. The first is `example_PCA.m`, which performs the PCA for $\mathcal{O}_1/\mathcal{O}_3$ interference for a 100 GeV WIMP scattering in germanium. This script demonstrates how to use the output of `get_PCA_vectors.m` and produces two plots: a plot of the differential event rate evaluated at each of the principal components, and a grid plot of all 2-dimensional projections of the 4-dimensional eigenvectors. Running this script for all interference cases and calculating the total event rate produces plots like Figure 9 of arXiv:1503.03379. In addition, this release includes the script `plot_0809_constructive_interference_rates.m`, which reproduces Figure 7 of arXiv:1503.03379, as shown in Fig. 2.

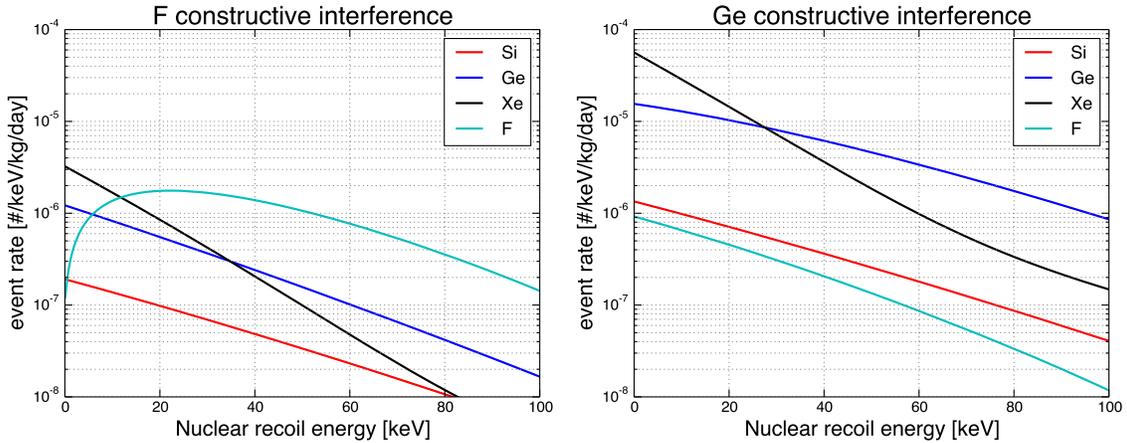


Figure 2: Differential event rate evaluated at the $\mathcal{O}_8/\mathcal{O}_9$ constructive interference vectors for fluorine (left) and germanium (right), as produced by the included script `plot_0809_constructive_interference_rates.m`.

Limits on EFT coupling coefficients

This release also contains the limits presented in arXiv:1503.03379. Plots in the style of Figure 1 of the CDMS EFT paper can be found in the subdirectory `limit_plots`, and text files containing the limits can be found in the subdirectory `limits`. The first column of the text file is the WIMP mass in GeV, and the subsequent columns are the limits on the

squared dimensionless isoscalar EFT coupling constants $((c_i^0)^2 * m_{weak}^4)$ for operators 1 and 3-15 in increasing order.