

Direct detection of dark matter with MadDM v.2.0



Mihailo Backović^{a,*}, Antony Martini^a, Olivier Mattelaer^b, Kyoungchul Kong^c,
Gopolang Mohlabeng^c

^a Centre for Cosmology, Particle Physics and Phenomenology (CP3), Université Catholique de Louvain, Chemin du Cyclotron 2, B-1348 Louvain-la-Neuve, Belgium

^b Institute for Particle Physics Phenomenology (IPPP), Durham University, Durham, DH1 3LF, United Kingdom

^c Department of Physics and Astronomy, University of Kansas, Lawrence, KS, 66045, USA

ARTICLE INFO

Article history:

Received 26 May 2015

Received in revised form

25 September 2015

Accepted 28 September 2015

Keywords:

Dark matter

Automation

MadDM

Direct detection

Directional detection

ABSTRACT

We present MadDM v.2.0, a numerical tool for dark matter physics in a generic model. This version is the next step towards the development of a fully automated framework for dark matter searches at the interface of collider physics, astro-physics and cosmology. It extends the capabilities of v.1.0 to perform calculations relevant to the direct detection of dark matter. These include calculations of spin-independent/spin-dependent nucleon scattering cross sections and nuclear recoil rates (as a function of both energy and angle), as well as a simplified functionality to compare the model points with existing constraints. The functionality of MadDM v.2.0 incorporates a large selection of dark matter detector materials and sizes, and simulates detector effects on the nuclear recoil signals. We validate the code in a wide range of dark matter models by comparing results from MadDM v.2.0 to the existing tools and literature.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

While there are many intriguing, indirect hints of the existence of dark matter (DM), the evidence of irrefutable signals and the true nature of DM remain elusive (see Ref. [1] and references therein for recent progress in DM searches). Various direct detection experiments are currently ongoing, and many are under development. The results from direction detection experiments have so far produced stringent constraints on the DM scattering cross section off atomic nuclei, especially for the DM in the mass range of the electroweak (EW) scale. More recently, the LHC analyses of jet(s) + missing energy and $W/Z/t$ + missing energy channels, and their variants, provided complementary information to underground fixed target experiments. The complementarity was particularly evident in the improved sensitivity for a relatively light DM in simplified model scenarios [2]. Despite a significant effort, none of the individual experiments have provided a strong indication on the mass scale of dark matter particles. It is therefore essential to search for evidence of dark matter in many different

ways: by directly producing DM candidate at particle colliders, as well as by detecting it in our galaxy and beyond [3].

Searching for DM at the interface of collider physics, astroparticle physics and cosmology is perhaps the most promising path to discovery and discrimination of different DM models, which requires substantial amount of computational power and relevant tools. While many collider event generators are publicly available [4], the number of numerical tools which are able to calculate signals of dark matter at the interface of collider physics and astroparticle physics remains very limited [5,6]. MadDM [7] emerged as an attempt to build such a tool. The goal of the MadDM project is to form a simple, user friendly and all-in-one DM phenomenology framework. Our ambition is to allow both experimentalists and theorists to calculate accurate signatures of generic DM models at colliders, in our galaxy and in the early universe with only a few clicks of a button. We also aim to provide flexibility such that users easily replace the existing modules with their own implementation or link MadDM functionality with other existing tools.

The user friendly architecture of the MadGraph5_aMC@NLO [8, 9] package provides an ideal framework for the development of MadDM. Numerous MadGraph additions and extensions offer many interesting directions that development of MadDM can go into. Version 1.0 of MadDM focused on computation of DM relic density in a generic model [7]. Here we discuss the next step in the development of MadDM: direct detection of galactic DM. In addition to the ability to calculate DM–nucleon cross sections,

* Corresponding author.

E-mail addresses: mihailo.backovic@uclouvain.be (M. Backović), antony.martini@uclouvain.be (A. Martini), o.p.c.mattelaer@durham.ac.uk (O. Mattelaer), kckong@ku.edu (K. Kong), gopolang.mohlabeng@ku.edu (G. Mohlabeng).

MadDM v.2.0 goes a step further and provides the directional information of nuclear recoil, as well as the ability to simulate the effects of detector systematics on recoil distributions. To the best of our knowledge, MadDM is the first publicly available code with such a capability. The code is able to calculate double differential recoil rates, as a function of recoil angle and energy, as well as the energy distribution and angular distribution of nuclear recoil.

Directional information of DM scattering can be of great importance for the next generation dark matter direct detection experiments. In case a DM signal is observed, directional information about nuclear recoil rates could lead to more precise measurements of intrinsic particle properties of DM [10] as well as astrophysical information such as the DM velocity distribution [11,12]. Conversely, if next generation direct detection experiments lead only to more stringent limits on DM mass and scattering cross section, directional detection could become crucial in overcoming the irreducible neutrino background [13].

Finally, MadDM v.2.0 incorporates a simplified functionality for testing model points against experimental constraints, whereby the user can choose the range of DM relic densities and upper bounds on DM–nucleon scattering cross sections for which the model is consistent with the experiments. The code will automatically compare the output of the calculation to the user specified constraints and determine whether the model point is allowed or not.

New version of MadDM follows the already established structure of MadDM v.1.0. A Python module generates relevant amplitudes for relic density and direct detection (with ALOHA [14]), while a FORTRAN module handles the heavy numerical calculations. As before, MadDM is compatible with UFO (Universal Feyn-Rules Output [15]) model which contains a dark matter candidate and can be easily linked to any width or mass spectrum generator which can produce a Les Houches formatted parameter card [16,17].

The following sections describe the new functionality of the MadDM code for the direct detection of DM, while we refer the reader to Ref. [7] for more information on calculation of DM relic density. Section 2 describes the theoretical background of dark matter direct detection, including directionality. A description of the new routines in the MadDM code is presented in Section 3. Finally, we show several validations and example calculations using the MadDM v.2.0 code in Section 4. We reserve the appendix for a brief description of the MadDM code structure.

2. Review of direct detection and its implementation in MadDM

2.1. DM–nucleon elastic cross section

The possibility that DM could weakly interact with ordinary matter has led to a development of a myriad of experiments which search for signatures of galactic DM scattering off nuclei [18–22]. Being sensitive to the nuclear recoil of the scattering DM–nucleon event in underground detectors, experiments can constrain model parameters involved in the DM–nucleon (nucleus) elastic scattering cross section, while in the case of a discovery, the same information may be used to determine properties of DM particles. The abundance of independent experiments has so far led to important limits on the DM–nucleus cross-section, among which the most recent (and most stringent) one comes from the LUX collaboration [18,23]. In this section, we give an overview of the theoretical formalism of DM–nucleus elastic scattering in order to allow the reader to understand the procedures MadDM performs in computing DM direct detection rates [24–26].

The MadDM v.2.0 code incorporates the following definition of the DM–nucleus:

spin-independent (SI) scattering cross section:

$$\sigma_{SI} = \frac{4}{\pi} \mu_A^2 \cdot [Z \cdot f_p + (A - Z) \cdot f_n]^2, \quad (1)$$

where $\mu_A = \frac{m_\chi m_A}{m_\chi + m_A}$ is the DM–nucleus reduced mass and f_p and f_n are proton/neutron spin-independent form factors respectively [11,13,27]. We use Z to denote the number of protons in a nucleus and A to denote the number of protons and neutrons. Similarly, for **spin-dependent** (SD) interactions, we use the definition:

$$\sigma_{SD} = \frac{16}{\pi} \mu_A^2 \cdot \frac{J_A + 1}{J_A} (f'_p + f'_n)^2, \quad (2)$$

where f'_p and f'_n are the proton/neutron spin-dependent form factors respectively, and J_A is the spin of the nucleus. Particle physics enters the calculation of $\sigma_{SD/SI}$ via the form factors f_N and f'_N where $N = p, n$.

The spin-independent quantity for a scalar current, f_N^S is defined as

$$f_N^S = m_N \sum_{q=u,d,s} \frac{\alpha_q}{m_q} f_N^{Sq} + \frac{2}{27} m_N f_N^{SG} \sum_{q=c,b,t} \frac{\alpha_q}{m_q}, \quad (3)$$

where α_q are the coefficients of the low energy matrix elements defined by

$$\langle \mathcal{M} \rangle (\chi q \rightarrow \chi q) = \alpha_q \langle \bar{\psi}_q \psi_q \rangle. \quad (4)$$

The quantities $f_N^{Sq/G}$ are nucleon form factors related to the low energy elastic scattering matrix elements as follows:

$$\langle N | \bar{\psi}_q \psi_q | N \rangle = \frac{m_N}{m_q} \cdot f_N^{Sq} \quad (\text{for } u, d, s), \quad (5)$$

$$\langle N | \bar{\psi}_q \psi_q | N \rangle = \frac{2}{27} \frac{m_N}{m_q} \cdot f_N^{SG} \quad (\text{for } c, b, t). \quad (6)$$

The values for nucleon form factors f_N^S are typically extracted from data or from chiral perturbation theory [28] and depend on the type of current which defines the DM–nucleon interaction. The default values for **scalar** spin-independent form factors of light quarks used in MadDM are:

$$\begin{aligned} f_p^{Su} &= 0.0153, & f_n^{Su} &= 0.0110, \\ f_p^{Sd} &= 0.0191, & f_n^{Sd} &= 0.0273, \\ f_p^{Ss} &= 0.0447, & f_n^{Ss} &= 0.0447, \end{aligned} \quad (7)$$

while the gluon scalar form factor is

$$f_N^{SG} = 1 - \sum_q f_N^{Sq}. \quad (8)$$

The **vector current** interaction is characterized by a simpler set of form factors. Since the vector current is conserved, the DM–nucleon form factor can be obtained from the sum of the currents of the valence quarks, i.e.:

$$f_N^V = \sum_{q=u,d} f_N^{Vq} \cdot \alpha_q, \quad (9)$$

where $f_p^{Vu} = f_n^{Vd} = 2$ and $f_p^{Vd} = f_n^{Vu} = 1$.

For spin-dependent interactions, the following equation defines the proton/neutron form factors f'_N in Eq. (2):

$$f'_N = \sum_{q=u,d,s} \alpha_q \Delta_N^q, \quad (10)$$

where the quantities Δ_N^q are nucleon form factors, applies for both **axial–vector** and **tensor currents**. In MadDM, we use the following default values for the axial–vector interaction:

$$\begin{aligned}\Delta_p^{Avu} &= \Delta_n^{Avd} = 0.842, \\ \Delta_p^{Avd} &= \Delta_n^{Avu} = -0.427, \\ \Delta_p^{AVs} &= \Delta_n^{AVs} = -0.085,\end{aligned}\quad (11)$$

while we use another set of coefficients for the tensor interaction:

$$\begin{aligned}\Delta_p^{Tu} &= \Delta_n^{Td} = 0.84, \\ \Delta_p^{Td} &= \Delta_n^{Tu} = -0.23, \\ \Delta_p^{Ts} &= \Delta_n^{Ts} = -0.046.\end{aligned}\quad (12)$$

The nucleon form factors for the axial–vector current, Δ_p^{Avu} and Δ_p^{Avd} , have been provided by HERMES and COMPASS experiments [29,30] while tensor coefficients come from lattice calculations [31].

The coefficients α_q can be extracted from the full matrix elements for χq scattering via Fiertz transformations. While this is relatively straightforward to do analytically, numerical implementations of Fiertz transformations are non-trivial. In the following section, we discuss the numerical method MadDM uses to compute the low energy coefficients α_q from the full matrix elements in the χq scattering.

2.2. Projection operator method for extraction of low energy coefficients

The procedure for computing the low energy coefficients for DM–nucleon scattering in MadDM is similar to the procedure implemented in MicrOMEGAs [26].¹ At the quark level, the input quark–DM interacting Lagrangian at low energy can be re-written as a set of effective operators as follows:

$$\mathcal{L}_{Q^2=0} = \sum_i \alpha_i \mathcal{O}_q^i, \quad (13)$$

where the operators $\mathcal{O}_q$ standing for χq scattering are defined in Table 1, and the index i runs over all the relevant contributions. The list of operators can be separated in odd and even operators under the interchange of quarks and anti-quarks as follows:

$$\begin{aligned}\mathcal{L}_{Q^2=0} &= (\mathcal{L}_{SI}^e + \mathcal{L}_{SI}^o) + (\mathcal{L}_{SD}^e + \mathcal{L}_{SD}^o) \\ &= \sum_{q,s} \alpha_{q,s}^{SI} \mathcal{O}_{q,s}^{SI} + \sum_{q,s} \alpha_{q,s}^{SD} \mathcal{O}_{q,s}^{SD},\end{aligned}\quad (14)$$

where $q = u, d, s, c, b, t$ and s stands for even (e) or odd (o) operator. The Fiertz transformation which gives $\alpha_{q,s}$ coefficients projects the matrix element for $\chi q \rightarrow \chi q$ onto the set of effective operators. Numerically, this is equivalent to taking the interference term of the matrix element for $\chi q \rightarrow \chi q$ scattering with the even effective operator matrix element for the same process. The projection over the effective operators will select either SI or SD contributions since Eq. (13) is written in an orthogonal basis.

The even and odd coefficients can be further separated by considering scattering off of both quarks and anti-quarks. The sum and difference of the even and odd coefficients can be written as:

$$\alpha_{q,e} + \alpha_{q,o} = \frac{|\mathcal{M}^{q*} \cdot \mathcal{M}_{\text{eff}}^{q,e}|}{|\mathcal{M}_{\text{eff}}^{q,e}|^2}, \quad (15)$$

$$\alpha_{q,e} - \alpha_{q,o} = \frac{|\mathcal{M}^{\bar{q}*} \cdot \mathcal{M}_{\text{eff}}^{\bar{q},e}|}{|\mathcal{M}_{\text{eff}}^{\bar{q},e}|^2}, \quad (16)$$

where the $|\mathcal{M}_{\text{eff}}^{q,e}|^2$ in the denominator takes into account the fact that the effective operators are not properly normalized. In Eq. (16) we used the property that the odd coefficient changes sign under the interchange of $q \rightarrow \bar{q}$. Note that the procedure of projection operators for extracting $\alpha_{q,s}$ is completely generic although the list of effective operators is non-exhaustive. Indeed, operators such as $\bar{\psi} \gamma_5 \psi$ are neglected since they are suppressed in the zero momentum limit.

2.3. Differential nucleus recoil rate

The direct detection module of MadDM allows the user to calculate quantities relevant for DM–nucleus scattering beyond the spin-independent and spin-dependent cross section. One is, for instance, often interested in studies of nucleus recoil rates, where traditionally the quantity of interest has been dR/dE , the differential nuclear recoil rate as a function of recoil energy. However, valuable information is contained in the angular distribution of the recoil rate $dR/d\Omega$. Despite the fact that no currently operating experiment has the ability to efficiently measure directionality of the nuclear recoil with high statistics, there are few ongoing efforts in this direction as well as concrete proposals for future experiments [10,13,34–37]. Hence, we deemed it important to include the ability to calculate the angular rates into the MadDM code.

One reason why angular information could be important is that low energy recoil events recorded by direct detection experiments lie well within the range dominated by cosmogenic backgrounds and those resulting from radioactivity. A strong signature of dark matter may come from directional detection, which exhibits a very large forward–backward asymmetry (A_{FB}) in the angular distribution of nuclear recoil events. A large A_{FB} would be difficult to mimic with any known backgrounds as one expects the angular distribution of such processes to be roughly uniform.

The fact that background processes are (typically) spatially isotropic could be used to overcome another impeding problem. As the direct detection bounds on the DM–nucleon cross-sections get lower, neutrino backgrounds will present an obstacle [38] and directional detection could provide a way to circumvent the so called “neutrino floor” [39].

In case a signal is ever observed, the study of directionality in DM–nucleus recoils will be of foremost interest. The angular distribution of the recoil rates could provide useful information about the astrophysical aspects of the DM galactic halo profile [11,40], as well as information on the anisotropy in the DM halo velocity distribution and prospects for more accurate measurements of the DM mass and interaction cross-section [10].

In this section we present a detailed overview of the theoretical background of differential recoil rates for DM detection. We begin by considering our solar system moving through the Galactic “DM-wind” in the direction of the Cygnus X – 2 constellation as illustrated in Fig. 1. A DM particle of mass m_χ , incident at velocity $\mathbf{v} = v(\sin \alpha \cos \beta \hat{\mathbf{x}} + \sin \alpha \sin \beta \hat{\mathbf{y}} + \cos \alpha \hat{\mathbf{z}})$ in the detector, elastically scatters off a target nucleus of mass m_N , stationary in the detector, and causes it to recoil with velocity $\mathbf{u} = u(\sin \phi \cos \theta \hat{\mathbf{x}} + \sin \phi \sin \theta \hat{\mathbf{y}} + \cos \phi \hat{\mathbf{z}})$ and momentum $\mathbf{q}$ in the direction (θ, ϕ) . The event rate per unit recoil energy and unit recoil solid angle is

$$\frac{d^2 R}{dE_R d\Omega_{(\theta, \phi)}} = \frac{2 \rho_0}{m_\chi} \int \frac{d\sigma}{dq^2 d\Omega_{(\theta, \phi)}} v f(\mathbf{v}) d^3 v, \quad (17)$$

where ρ_0 is the dark matter density in our local galaxy and $f(\mathbf{v})$ is the velocity distribution of the DM in the galactic halo. The differential cross-section

$$\frac{d\sigma}{dq^2 d\Omega_{(\theta, \phi)}} = \frac{\sigma_{\chi N}}{8\pi \mu^2 v} F^2(q) \delta\left(v \cos \theta - \frac{q}{2\mu}\right), \quad (18)$$

¹ For improved treatments on QCD effect in dark matter scattering off nucleon, see Ref. [32] and Ref. [33] for recent development.

Table 1

List of effective operators taken from Ref. [26], implemented into MadDM. S , ψ_χ and A_χ correspond to scalar, fermion and vector DM fields, respectively.

	DM spin	Even	Odd
SI	0	scalar current $2M_\chi S S^* \bar{\psi}_q \psi_q$	vector current $i(\partial_\mu S S^* - S \partial_\mu S^*) \bar{\psi}_q \gamma^\mu \psi_q$
	1/2	$\bar{\psi}_\chi \psi_\chi \bar{\psi}_q \psi_q$	$\bar{\psi}_\chi \gamma_\mu \psi_\chi \bar{\psi}_q \gamma^\mu \psi_q$
	1	$2M_\chi A_\chi^* A_\chi \bar{\psi}_q \psi_q$	$i(A_\chi^{\alpha\mu} \partial_\mu A_{\chi\alpha} - A_\chi^{\alpha\mu} \partial_\mu A_{\chi\alpha}^*) \bar{\psi}_q \gamma_\mu \psi_q$
SD	1/2	axial-vector current $\bar{\psi}_\chi \gamma^\mu \gamma^5 \psi_\chi \bar{\psi}_q \gamma_\mu \gamma^5 \psi_q$	tensor current $-\frac{1}{2} \bar{\psi}_\chi \sigma_{\mu\nu} \psi_\chi \bar{\psi}_q \sigma^{\mu\nu} \psi_q$
	1	$\sqrt{6}(\partial_\alpha A_{\chi\beta}^* A_{\chi\nu} - A_{\chi\beta}^* \partial_\alpha A_{\chi\nu}) \epsilon^{\alpha\beta\nu\mu} \bar{\psi}_q \gamma_5 \gamma_\mu \psi_q$	$i\frac{\sqrt{3}}{2}(A_{\chi\mu} A_{\chi\nu}^* - A_{\chi\nu}^* A_{\chi\mu}) \bar{\psi}_q \sigma^{\mu\nu} \psi_q$

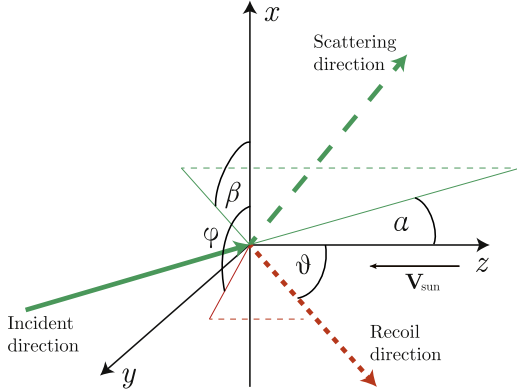


Fig. 1. Geometry of DM scattering off a target nucleus in the detector. The DM is incident at an angle (α, β) relative to the z axis. The nucleus recoils in the (θ, ϕ) direction. Note that β and ϕ angles lie in the x, y plane. The thin and dashed thin lines serve to illustrate the projection of the incoming and recoil vectors onto the z axis and the xy plane.

is obtained from the two body scattering phase space distribution [12]. Here, $\mu = m_\chi m_N / (m_\chi + m_N)$ is the reduced mass of the DM–nucleus system, $q = \sqrt{2 m_N E_R}$ is the recoil momentum with E_R the recoil energy, angle θ is the recoil angle which determines the direction between the recoiling nucleus and the initial DM trajectory, while $\sigma_{\chi N}$ is the DM–nucleus scattering cross-section, weighted by the nuclear elastic scattering form factor $F(q)$. Assuming the nucleus is a sphere with uniform density, the form-factor is the Fourier transform of the nuclear density and includes all the relevant nuclear effects. For a detailed discussion on $F(q)$ see Section 2.4. The DM–nucleus cross-section is summed over contributions from the spin-independent (σ_{SI}) and spin-dependent (σ_{SD}) cross-sections respectively, as defined in Eq. (1) and Eq. (2) respectively. Eq. (17) becomes

$$\frac{d^2 R}{dE d\Omega_{(\theta, \phi)}} = \frac{N_0 \rho_0 \sigma_{\chi N}}{\pi A r m_\chi^2} F^2(q) \int \delta\left(v \cos \theta - \frac{q}{2\mu}\right) f(\mathbf{v}) d^3 v, \quad (19)$$

where $r = 4m_N m_\chi / (m_N + m_\chi)^2$, $N_0 = 6.022 \times 10^{26} \text{ kg}^{-1}$ is Avogadro's number, $m_N = 0.932 A$ is the target mass and A the atomic mass number (the factor of 0.932 is the value of atomic mass units (AMU) in GeV). This form of the double differential recoil rate is particularly useful as the quantity

$$\hat{f}(v_q, \hat{\mathbf{q}}) = \int \delta(\mathbf{v} \cdot \hat{\mathbf{q}} - v_q) f(\mathbf{v}) d^3 v, \quad (20)$$

is easily identified as the three dimensional Radon transform,² where $\hat{\mathbf{q}}$ is the recoil momentum direction, $\mathbf{v}$ is the velocity of the DM particle and $v_q = \frac{q}{2\mu}$ is the minimum velocity required for

the DM to impart a recoil momentum $\mathbf{q}$ on the nucleus. $\hat{f}(v_q, \hat{\mathbf{q}})$ represents the velocity distribution for a detector stationary in the galactic frame. We consider an observer moving through the Galaxy with velocity $\mathbf{V}_{\text{lab}}$ in the galactic frame. $\mathbf{V}_{\text{lab}}$ is related to the DM velocity in the lab frame ($\mathbf{v}_{\text{lab}}$) and the DM velocity in the galactic frame ($\mathbf{v}_{\text{gal}}$) by the Galilean transformation $\mathbf{v}_{\text{lab}} = \mathbf{v}_{\text{gal}} - \mathbf{V}_{\text{lab}}$. The transformation takes into account the motion of a detector relative to the DM particles as seen in the lab frame. By the properties of the Radon transform for a pure translation $\mathbf{V}_{\text{lab}}$, we obtain

$$\hat{f}_{\text{lab}}(v_q, \hat{\mathbf{q}}) = \hat{f}_{\text{gal}}(v_q + \mathbf{V}_{\text{lab}} \cdot \hat{\mathbf{q}}, \hat{\mathbf{q}}). \quad (21)$$

We assume a Maxwell-Boltzmann velocity distribution which is truncated at the DM escape velocity v_{esc} such that

$$f_{\text{MB}}(v) = \frac{1}{k_{\text{esc}} \pi^{3/2} v_0^3} e^{-v^2/v_0^2}, \quad (22)$$

for $v < v_{\text{esc}}$ and $f_{\text{MB}}(v) = 0$ otherwise. In Eq. (22), v_0 is the most probable speed of the DM in the halo with typical values of 220 ~ 250 km/s and k_{esc} is a normalization factor which is obtained by integrating the velocity distribution in the galactic frame from 0 to v_{esc} . For $\mathbf{V}_{\text{lab}} = \mathbf{v}_E$, where $\mathbf{v}_E$ is the velocity of the Earth with respect to the galactic frame, Eq. (21) becomes

$$\hat{f}_{\text{MB}}(v_q + \mathbf{v}_E \cdot \hat{\mathbf{q}}, \hat{\mathbf{q}}) = \frac{1}{k_{\text{esc}} \sqrt{\pi} v_0} \left[\exp\left(-\frac{(v_q + \mathbf{v}_E \cdot \hat{\mathbf{q}})^2}{v_0^2}\right) - \exp\left(-\frac{v_{\text{esc}}^2}{v_0^2}\right) \right]. \quad (23)$$

Combining Eqs. (19) and (23) we obtain an expression for a double differential recoil rate in the laboratory frame:

$$\frac{d^2 R}{dE_R d\cos \theta} = \frac{2 N_0 \rho_0 \sigma_{\chi N}}{\pi^{1/2} A r v_0 m_\chi^2} \frac{F^2(E_R)}{k_{\text{esc}}} \times \left[\exp\left(-\frac{(v_E \cos \theta - v_{\text{min}})^2}{v_0^2}\right) - \exp\left(-\frac{v_{\text{esc}}^2}{v_0^2}\right) \right], \quad (24)$$

where $\mathbf{v}_E \cdot \hat{\mathbf{q}} = -v_E \cos \theta$ and $v_q = v_{\text{min}} = v_0 \sqrt{E_R/E_0 r}$, with $E_0 = 1/2 m_\chi v_0^2$ being the most probable kinetic energy of the DM. The velocity of the Earth v_E , with respect to the galactic frame is calculated in the appendix of Ref. [41] and includes the motion of the Earth with respect to the Sun, the proper motion of the Sun and the motion of the solar system with respect to the galactic center. Eq. (24) assumes azimuthal symmetry of the DM velocity profile as illustrated in Fig. 1.

2.4. Nuclear form factors and detector resolution

In order to compute the recoil rates, we take into account both spin-dependent (axial-vector) and spin-independent (scalar) DM–nucleon interactions. The contributions of different types of interactions show up in the DM–nucleus cross-section and in the

² For more information about the Radon transforms as applied to directional detection, we refer the reader to Ref. [12].

Table 2

List of stable isotopes and their abundances for the different target materials used in the calculation of the spin-independent quantities.

Target material		List of stable isotopes & abundances
Xenon	A	123.9, 125.9, 127.9, 128.9, 129.9, 130.9, 131.9, 133.9, 135.9
Z = 54	Abundance (%)	0.09, 0.09, 1.92, 25.44, 4.08, 21.18, 26.89, 10.44, 8.87
Germanium	A	69.92, 71.92, 72.92, 73.92, 75.92
Z = 32	Abundance (%)	20.84, 27.54, 7.73, 36.28, 7.61
Silicon	A	27.98, 28.98, 29.97
Z = 14	Abundance (%)	92.23, 4.68, 3.087
Neon	A	19.99, 20.99, 21.99
Z = 10	Abundance (%)	90.48, 0.27, 9.25
Argon	A	37.962, 39.962
Z = 18	Abundance (%)	0.0632, 99.6
Sodium	A	22.9
Z = 11	Abundance (%)	100
Iodine	A	126.9
Z = 53	Abundance (%)	100
Carbon	A	12.0, 13.0
Z = 6	Abundance (%)	98.89, 1.11
Fluorine	A	18.998
Z = 9	Abundance (%)	100
Sulfur	A	31.9, 32.97, 33.96, 35.96
Z = 16	Abundance (%)	94.9, 0.76, 4.29, 0.02

form-factor. For the spin-independent case, we consider the Helm form factor as described in Ref. [41]:

$$F_{SI}(q) = \frac{3[\sin(qr_n) - qr_n \cos(qr_n)]}{(qr_n)^3} e^{-(qs)^2/2}, \quad (25)$$

with $q = 6.92 \times 10^{-3} A^{1/2} E_R^{1/2} \text{ fm}^{-1}$ the recoil momentum, $r_n = \sqrt{c^2 + 7/3\pi^2 a^2 - 5s^2} \text{ fm}$ the effective radius of the nucleus, $c = (1.23A^{1/3} - 0.60) \text{ fm}$, $a = 0.52 \text{ fm}$ and $s = 0.9 \text{ fm}$. Eq. (25) is in fact the Wood–Saxon form factor, with $r_n = c^2 - 5s^2$ and $c = 1.2A^{1/3} \text{ fm}$. Furthermore for the spin-dependent case, we consider

$$F_{SD}^2(q) = \frac{S(q)}{S(0)}, \quad (26)$$

where the recoil momentum dependent structure function is given by $S(q) = a_0^2 S_{00}(q) + a_0 a_1 S_{01}(q) + a_1^2 S_{11}(q)$ and $S(0)$ is the structure function at zero momentum transfer. For the coefficients

$$a_0 = \frac{11}{18} (\Delta_u + \Delta_d) + \frac{5}{18} \Delta_s, \quad (27)$$

$$a_1 = \frac{11}{18} (\Delta_u + \Delta_d) + \frac{5}{18} \Delta_s, \quad (28)$$

we use the nucleon form factors, Δ_i ($i = u, d, s$), as defined in Eq. (11). The nuclear spin structure functions S_{ij} for different nuclei are approximated using fitting functions. For Xenon, Iodine and Sodium we use the functions from the Nijmegen II fitting as calculated in Ref. [42]. For Germanium we use those calculated in Ref. [43], while for Fluorine and Silicon we use the fitting functions found in Ref. [44]. All the information we use for both spin-independent and spin-dependent interactions are summarized in Tables 2 and 3³ respectively. For the nuclear structure functions that are currently not very well calculated (such as Carbon, Sulfur, Neon and Argon), we use the Gaussian distribution $S_{ij}(q) = S_{ij}(0) \exp\left[-\frac{q^2 R_A^2}{4}\right]$ as described in Ref. [26] with $R_A = 1.7A^{1/3} -$

Table 3

List of structure functions, moments and angular momenta for different materials. The fitting polynomials for the structure functions can be found in the references cited. We were not able to find the magnetic moments for Argon and Sulfur thus we set them to be zero, thus MadDM does not currently compute the SD rates for them. We were also not able to find the orbital spin values for Argon, Neon and Sulfur, we set these to $\frac{1}{2}$ in the code at the moment.

Target material	S_{ij}	$\langle S_n \rangle$	$\langle S_p \rangle$	J
Xenon	Ref. [42]	−0.272	−0.009	$\frac{3}{2}$
Germanium	Ref. [43]	0.378	0.030	$\frac{9}{2}$
Silicon	Ref. [44]	0.13	−0.002	$\frac{1}{2}$
Neon	Refs. [26,44]	0.294	0.02	$\frac{1}{2}^*$
Argon	Ref. [26]	0*	0*	$\frac{1}{2}^*$
Sodium	Ref. [42]	0.0199	0.2477	$\frac{3}{2}$
Iodine	Ref. [42]	0.075	0.309	$\frac{5}{2}$
Carbon	Ref. [26]	−0.172	−0.009	$\frac{1}{2}$
Fluorine	Refs. [44,45]	−0.0087	0.4751	$\frac{1}{2}$
Sulfur	Ref. [26]	0*	0*	$\frac{1}{2}^*$

$0.28 - 0.78(A^{1/3} - 3.8 + \sqrt{(A^{1/3} - 3.8)^2 + 0.2})$ for $A < 100$ and $R_A = 1.5A^{1/3}$ for $A > 100$. We note that the Gaussian approximation breaks down for large recoil energies [26].

Finally we use Eq. (24) to simulate the scattering of dark matter with the nucleus in a detector. We take into account the energy and angular resolutions of a generic detector as applied to the distribution of the scattering events. We assume a Gaussian resolution function to smear the distribution of recoil events in both energy and angle with standard deviations $\sigma_E = \lambda\sqrt{E}$ (the energy resolution) and σ_θ (constant angular resolution) as in Ref. [10]. Default resolutions are given by $\lambda = 1$ and $\sigma_\theta = 30^\circ$ with an option for users to implement their own detector resolutions.

3. How to use MadDM v.2.0

3.1. Running MadDM

Addition of the direct detection code to the existing MadDM package follows the syntax and philosophy of the previous MadDM version. The code is split up into the Python and the FORTRAN modules, whereby diagram generation and the structure of folders

³ We were also not able to find the values in asterisk. For Neon, Argon and Sulfur, we set the orbital spin values to $\frac{1}{2}$ to avoid any infinities in the calculation of the spin-dependent cross-sections as shown in Eq. (2), while we set the default values of the magnetic moments for Argon and Sulfur to 0.

and files is handled by Python, and the numerical calculations are performed by the FORTRAN module. As in the previous version, the code does not require any pre-compilation. Simply placing the code within a MadGraph5_aMC@NLO folder suffices. Note however, that Numpy is now a pre-requisite for running the MadDMv2.0 code. In order to use the plotting functionality, Matplotlib is also required.

In the following we briefly describe how to use the MadDM code relevant for direct detection of dark matter, while we give detailed descriptions of the main functions/features of the MadDM v2.0 package in [Appendix 4](#).

As in version 1.0, the main MadDM program can be executed via the maddm.py script. The interface will prompt the user to enter the name of the UFO dark matter model (TAB auto-complete included), a name of the projects, as well as offer a choice of available computations (relic density, DM–nucleon cross sections and recoil rates). The Python module of MadDM will then proceed to generate all the relevant scattering/annihilation diagrams and set up the directory structure of the user defined project in Projects/<projectname> folder.

The user can choose to enter the DM candidate manually, or to allow MadDM to determine the dark matter candidate automatically (the code assumes that the lightest particle with the PDG code greater than 25 and zero width is the DM candidate). The code also offers the user to edit the default parameter card, and pre-set the model parameters before it searches for the DM candidate.

Upon determining the DM candidate, MadDM will attempt to find any co-annihilation candidates. The user is prompted to enter the mass splitting within which the co-annihilation partners are deemed to contribute significantly to the calculation of relic density (the default is 10%).

Finally, the code will prompt the user to edit the maddm_card.inc file, which contains all relevant parameters for the numerical calculations. These include, but are not limited to precision parameters for the relic density calculation, nucleon form factors for direct detection, choices of target materials and detector parameters. Upon execution and output of the results, the code will offer the user the option to perform a parameter scan. Alternative to the user interface in maddm.py the MadDM code can also be run within any user defined Python script. We provide such an example in the example.py file.

3.2. UFO model conventions

Now we proceed to discuss some important conventions on UFO files required by the MadDMv2.0 code. We implemented the projection operator formalism in MadDM v2.0 via a recent feature of MadGraph5_aMC@NLO which allows the user to merge two UFO model files. We provide UFO models with vertices of effective operators as a part of the MadDM code, so that an interference term between the matrix elements of the user defined model, and the effective operators can be extracted. The vertices are then automatically added to the DM model provided by the user during the execution of the code. The merging procedure of MadGraph5_aMC@NLO is sensitive to the UFO convention and the user should take special care to make sure that the DM model used in MadDM is compatible with the effective operator models embedded in MadDM v2.0.⁵ We have generated our set of effective operators with FeynRules 2.0. We advise the user to test the merging procedure in MadGraph5_aMC@NLO before running MadDM in order to avoid UFO conflicts. Using this UFO convention, the DM particle should be defined as

```
DM = Particle(pdg_code = pdg code,
...
spin = 1,
mass = Param.DMMASS,
width = Param.WDM,
...)
```

in case of a real scalar DM (for example). In this convention, **it is compulsory to assign a PDG code to the DM particle as well as a mass and a width**. Next, the mass and width block parameters have to match the following template:

```
DMMASS = Parameter(name = 'DMMASS',
...
lhablock = 'MASS',
lhacode = [ pdg code ])

WDM = Parameter(name = 'WDM',
...
lhablock = 'DECAY',
lhacode = [ pdg code ])
```

where the lhablock and lhacode parameters should be exactly like the template when replacing the PDG code by the PDG code of the DM particle. Note also that we cannot use mdm for the DM mass name due to conflict with internal MadDM parameters. For the same reason, if the user wants to define new coupling orders, the following ones should not be used: SIEFFS, SIEFFF, SIEFFV, SDEFFF, SDEFFV.

Finally, the user should **make sure that direct detection diagrams have a QED order lower or equal to 2** since MadDM generates diagrams with the QED=2 flag.

4. Validations

We performed validations of the MadDM code in a diverse class of benchmark models. In each case, we verified that the code reproduces accurately the results from existing literature or public codes such as MicrOMEGAs.

4.1. Validations of the DM–nucleon cross section calculation

4.1.1. Simplified DM models

The simplest benchmark model we consider for validation is a scalar DM which communicates to the SM via the Higgs boson. The model contains only one diagram contributing to direct detection i.e. a t -channel Higgs exchange. The Higgs–DM coupling is included in the $\frac{g}{2}H^\dagger HS$ operator where S is the DM scalar field. It implies that the model leads only to spin-independent cross section.

For the purpose of validating calculations of the spin-dependent cross sections, we consider a similar simplified model, with an axial–vector mediator which couples through fermionic DM and SM via the following Lagrangian:

$$\mathcal{L}_A = g_\chi^A \bar{\chi} \gamma^\mu \gamma_5 \chi V_\mu + g_q^A \bar{q} \gamma^\mu \gamma_5 q V_\mu, \quad (29)$$

where χ and V_μ are the DM and mediator fields, respectively. [Fig. 2](#) shows our results for proton/neutron DM cross section for the two simplified models against results obtained with MicrOMEGAs. In both cases, we find an excellent agreement between MadDM and MicrOMEGAs with the differences at $\lesssim 1\%$ level. We have also checked that the results from MadDM are consistent results from MicrOMEGAs in case of simplified models with different mediators (scalar, fermion and vector) and different DM candidate (scalar, fermion and vector), which are shown in [Table 1](#).

⁴ For the reference on the relic density calculation see Ref. [7].

⁵ The statement is relevant only for direct/directional detection. Relic density will proceed correctly without regard for the UFO version.

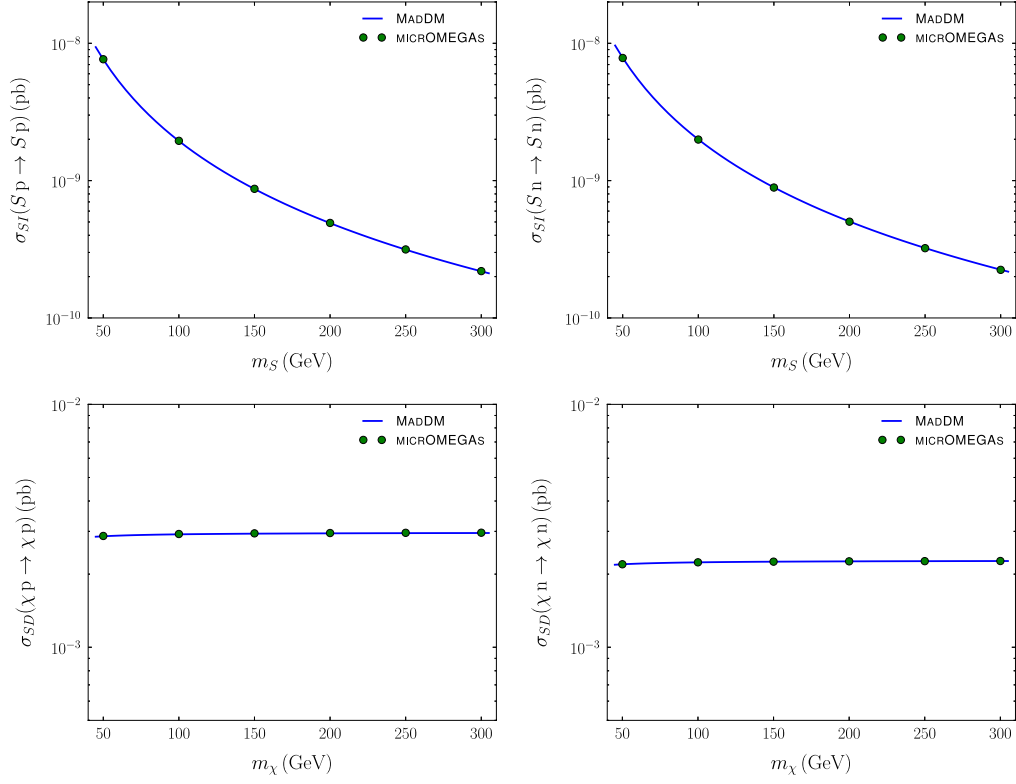


Fig. 2. Spin-independent elastic scattering cross section of scalar DM in the simplified SM model scenario with proton and neutron for $\delta = 0.1$ (top). Spin-dependent elastic scattering cross section of Dirac fermion DM with proton and neutron for $g_\chi^A = 0.1$ (bottom).

4.1.2. Minimal universal extra dimensions

After validating MadDM for a class of simplified models, we turn to more complex models. For this purpose, we chose to examine the results of MadDM for the Minimal Universal Extra Dimension model (MUED) against the results in existing literature.

MUED is the simplest model containing a Kaluza–Klein (KK) dark matter candidate among extra dimensions theories. The lightest KK particle, the KK-photon (γ_1) appears as a dark matter candidate in vanilla UED scenarios. At tree level, the inverse radius R^{-1} of the extra-dimension corresponds roughly to the mass of the massive fields at level one. The mass splitting between KK-quark (q_1) and KK-photon:

$$\Delta = \frac{m_{q_1} - m_{\gamma_1}}{m_{\gamma_1}}, \quad (30)$$

plays an important role as a free parameter in direct detection. The details and phenomenology of MUED will not be discussed here since it has been studied extensively in the past [46–49] and we will proceed directly to the comparison of MadDM results on direct detection to the results in literature.

The diagrams contributing to DM–nucleon cross section are displayed in Fig. 3. The Higgs-exchange diagram contributes to SI DM–nucleon cross section while the other two diagrams involving KK quarks contribute both for SI and SD DM–nucleon cross sections. We compared the MadDM results against those using the private code used in Refs. [46,49]. The results, shown in Fig. 4, show perfect agreement between MadDM and the private code over a wide range of dark matter masses and for several values of Δ . The DM–neutron cross sections are also in excellent agreement.

4.1.3. Minimal supersymmetric standard model

Finally, we also validated MadDM for SPS1a benchmark point in Minimal Supersymmetric Standard Model (MSSM) [50]. MSSM neutralino dark matter is similar to KK photon dark matter in that

the processes relevant for direct detection typically proceed via squark exchange in the s - and u -channels and Higgs exchange in the t -channel (see Fig. 3). In Table 4, we compare MadDM with MicrOMEGAS for DM–nucleon cross-section for several SPS points. The ratio of the two tools cross-sections indicates that MadDM agrees with MicrOMEGAS to $\sim 1\%$ level, with the exception of the SPS5 point. We have checked that the statement is also true if we vary the neutralino mass around the SPS1a point ($m_{\tilde{\chi}_1^0} = 96.68$ GeV).

We note that it is challenging to compare other MSSM parameter points between MicrOMEGAS and MadDM. The default implementation of the MSSM model in MicrOMEGAS contains numerous model specific optimizations and improvements which we are not included in MadDM. Hence, the only way we could compare the two codes in an “apples to apples” fashion in the context of MSSM was to use the MSSM implementation provided on the FeynRules website. Despite numerous technical difficulties, we were able to generate MSSM models for several specific SPS points, by first pre-processing the SLHA cards from SUSYHIT to fit the format required by FeynRules and then using those parameter sets to output both UFO and CalcHEP models. For the purpose of comparison, we manually turned off running of the QCD coupling in MicrOMEGAS by setting the variable `qcdNLO=1` in `directDet.c` and turned off the contributions from higher twist operators by setting `Twist20n=0`, as these features are not built into MadDM.

4.2. Recoil rates

Upon validating the DM–nucleon scattering cross section results from MadDM, we proceed to the recoil rates for DM scattering off a target nucleus. We begin with a simple, model independent validation of the recoil rate calculation, where we simply assume that the DM–nucleon cross section $\sigma_{\chi n} = 10^9$ pb, chosen for the

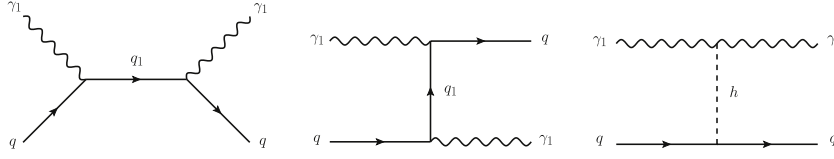


Fig. 3. Feynman diagrams for elastic scattering of KK photon with a quark.

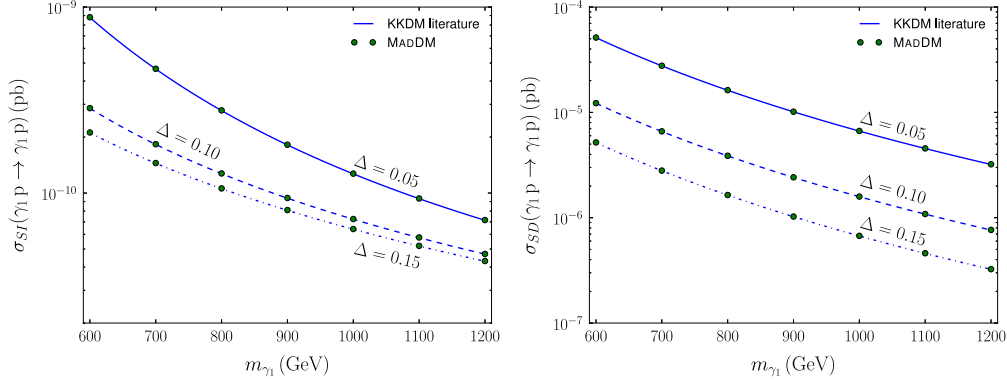


Fig. 4. Spin-independent (left panel) and spin-dependent (right panel) DM-proton cross section for MUED. The blue curves correspond to theoretical values coming from Kaluza–Klein dark matter literature [48,49]. The green points are the MadDM data. We show the results for three values of the Δ parameter.

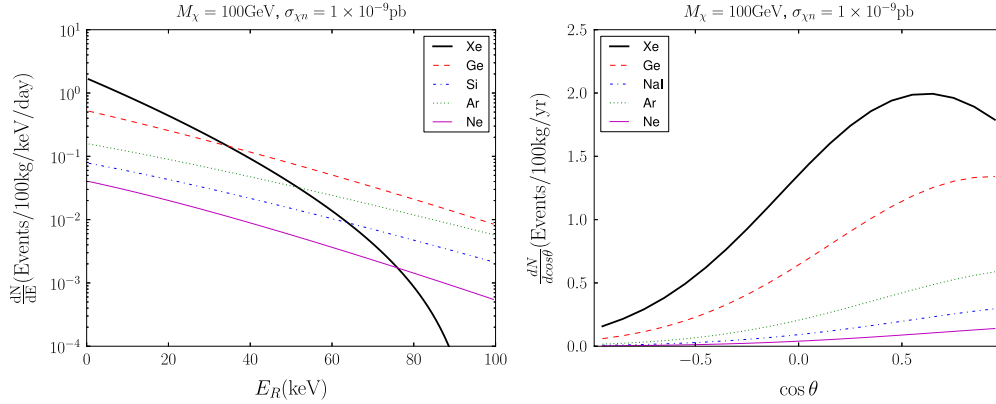


Fig. 5. Nuclear recoil energy (left panel) and angular (right panel) distributions for spin-independent interactions for different materials, assuming a 100 kg detector measuring events over one year for a DM mass of 100 GeV and DM-nucleon cross-section of 1×10^{-9} pb.

purpose of comparison with the results from Ref. [20]. To reproduce the SI recoil rates as a function of energy/angle as in Ref. [20], we employ the differential recoil spectrum of Eq. (24), integrated over time and angle/energy. Fig. 5 shows the spin-independent recoil rates as a function of recoil energy (left) and recoil angle (right). We find that both distributions are in a very good agreement with the results found in Ref. [20], over a wide range of target materials.

As a next validation, we check the recoil rates in UED model following the procedure described in Ref. [48], which shows sum of SI and SD recoil rates. The spin-dependent recoil rates are sensitive to numerical values of various quantities such as magnetic moments and parametrization of form factors. We use those values quoted in the references that are cited in Ref. [48]. Fig. 6 shows nuclear recoil energy distributions as a function of recoil energy for Xenon, Germanium and NaI. KK photon mass is chosen to be 1000 GeV with the DM-nucleon scattering cross-sections for both spin-dependent and spin-independent for $\Delta = 15\%$ as illustrated in Fig. 4. Despite the minor differences in the spin-dependent recoil rates, we find that Fig. 6 shows a good agreement between MadDM and the calculations in Ref. [48].

Furthermore, we compare the differential energy rates obtained from MicrOMEGAS with those obtained from MadDM for a Higgs portal scalar dark matter model for 4 different materials:

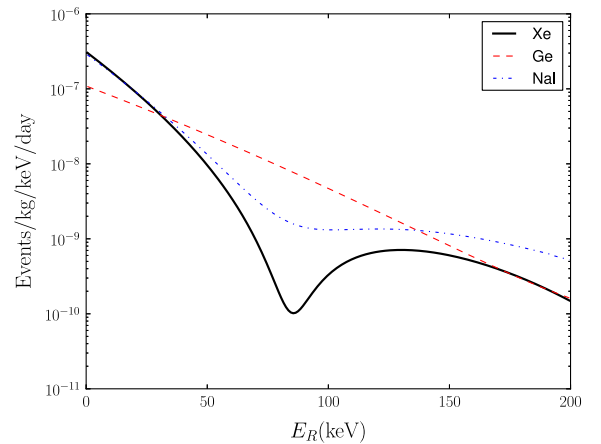


Fig. 6. Nuclear recoil energy distributions as a function of recoil energy for Xenon, Germanium and NaI. These are the rates for the UED model assuming a mass of 1000 GeV for the DM (γ_1) with the DM-nucleon scattering cross-sections for both spin-dependent and spin-independent for $\Delta = 15\%$ as illustrated in Fig. 4.

Xenon, Iodine, Germanium and Sodium. We consider only the most abundant isotopes of these materials as is assumed in

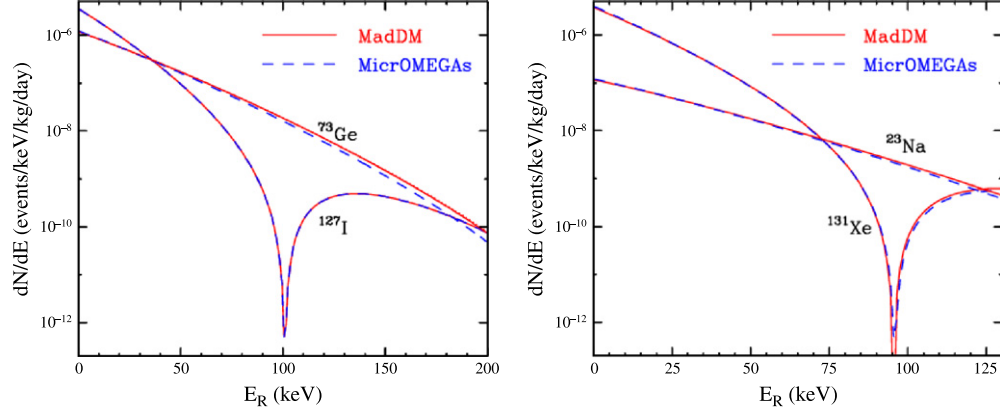


Fig. 7. Comparison of recoil energy distribution for I and Ge (left panel) and for Xe and Na (right panel) for a Higgs portal scalar dark matter model.

Table 4
MadDM and MicrOMEGAs comparison for DM–nucleon cross sections in MSSM.

σ type	MadDM [pb]	micrOMEGAs [pb]	% difference
SPS1a			
SI (proton)	2.16E–10	2.18E–10	0.9
SI (neutron)	2.15E–10	2.17E–10	0.5
SD (proton)	6.53E–6	6.56E–6	0.5
SD (neutron)	8.79E–6	8.79E–6	0.0
SPS1b			
SI (proton)	3.87E–11	3.89E–11	0.5
SI (neutron)	3.85E–11	3.87E–11	0.4
SD (proton)	1.20E–6	1.21E–6	0.5
SD (neutron)	1.47E–6	1.47E–6	0.0
SPS2			
SI (proton)	7.35E–11	7.39E–11	0.5
SI (neutron)	7.35E–11	7.39E–11	0.5
SD (proton)	9.86E–6	9.89E–6	0.3
SD (neutron)	8.05E–6	8.05E–6	0.0
SPS3			
SI (proton)	5.55E–11	6.02E–11	0.4
SI (neutron)	5.53E–11	5.98E–11	0.1
SD (proton)	1.63E–6	1.64E–6	0.6
SD (neutron)	2.08E–6	2.08E–6	0.0
SPS4			
SI (proton)	2.04E–10	2.07E–10	1.3
SI (neutron)	2.03E–10	2.05E–10	1.0
SD (proton)	7.56E–6	7.58E–6	0.3
SD (neutron)	7.82E–6	7.82E–6	0.0
SPS5			
SI (proton)	6.97E–11	4.33E–11	4.6
SI (neutron)	6.95E–11	4.31E–11	4.0
SD (proton)	4.05E–8	4.09E–8	1.0
SD (neutron)	4.77E–7	4.77E–7	0.0
SPS9			
SI (proton)	5.01E–11	5.03E–11	0.3
SI (neutron)	5.00E–11	4.98E–11	0.3
SD (proton)	1.11E–6	1.12E–6	0.6
SD (neutron)	1.72E–6	1.72E–6	0.0

MicrOMEGAs. SD cross section vanishes in this case since DM is a scalar. Fig. 7 illustrates the comparison between MicrOMEGAs and MadDM for the differential energy rates as a function of recoil energy in keVnr, while Table 5 shows several numerical comparisons of the total expected number of events for a 1 kg-day normalization. We find an excellent agreement between the two codes over a wide range of recoil energies and target materials.

4.3. LUX exclusion bound

As a final validation of the MadDM code, we attempt to reproduce the exclusion limit on the DM–nucleon cross-section as a function of DM mass similar to the LUX 2013 experimental

Table 5

Total number of events for a Higgs portal dark matter model with 100 GeV mass. We compare the number of events as obtained from MicrOMEGAs and MadDM for a 1 kg detector measuring events over a day.

	MicrOMEGAs	MadDM
^{131}Xe	5.41×10^{-5}	5.42×10^{-5}
^{73}Ge	2.99×10^{-5}	3.07×10^{-5}
^{127}I	5.28×10^{-5}	5.30×10^{-5}
^{23}Na	3.09×10^{-6}	3.12×10^{-6}

results [18]. For this purpose, we assume the efficiency function of nuclear recoils displayed in the black curve of Fig. 1 in Ref. [23].

Fig. 8 shows the results from MadDM for different energy threshold cuts as compared to the data reported by the LUX experiment. We present the contours assuming 2.3 events, coinciding with the number of events at 90% confidence as required by the Feldman–Cousins confidence intervals. We find a good match between the LUX data and limits from MadDM. As we could not obtain information on what value of the energy threshold cut is used in the LUX limit, we considered different values of threshold cuts. As illustrated in Fig. 8, the threshold cuts only impact the lower mass side since a higher threshold cut reduces the statistics for a lower DM mass.

Note that, as described in the previous sections, the exclusion curves in Fig. 8 can be obtained in MadDM by using LUX_Exclusion routine found in the test routines part in maddm.f. The routine multiplies $\frac{dR}{dE}$ by the efficiency obtained from Ref. [23], which is then weighted by a 50% acceptance rate for nuclear recoils as stated in the LUX analysis. From the recoil spectrum weighted by the efficiency and the acceptance rate, the function then calculates the total number of expected events. The default value for the detector efficiency is 100%, and can be easily replaced by a user defined function.

5. Conclusions

The identity of dark matter is one of the most profound mysteries in particle physics, astrophysics, and cosmology. Recent data from gamma rays, supernovae luminosities, cosmic microwave anisotropies, and galactic rotation curves all point consistently to the existence of dark matter with ~ 5 times more abundance compared to ordinary matter. At the same time, all known particles are excluded as possible dark matter candidates, making the dark matter problem perhaps the most pressing motivation for physics beyond the Standard Model. Little is currently known about the mass scale of dark matter, suggesting that discovery and characterization of DM will likely require a synergistic approach

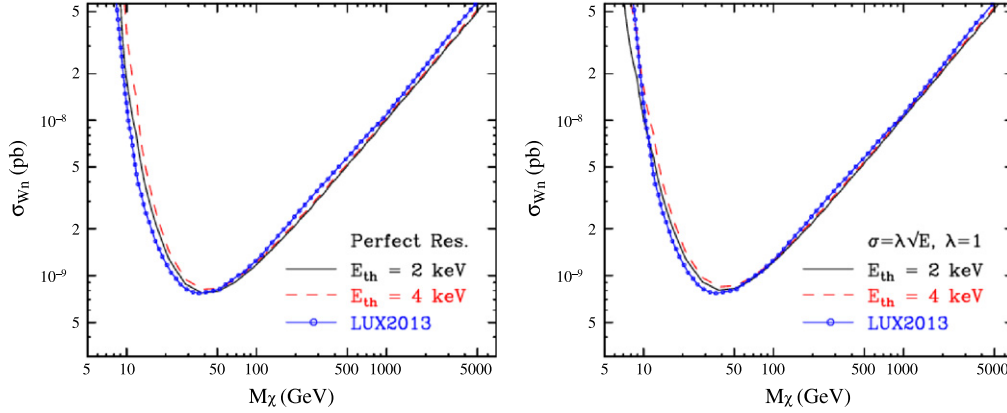


Fig. 8. 90% confidence limits on the spin-independent DM–nucleon scattering cross-section (in picobarns) for an unsmeared energy distribution (left panel) and the smeared distribution with $\lambda = 1$ (right panel). Limits are obtained from MadDM for 2 keV (black solid) and 4 keV (red dashed) and LUX limits are shown in blue curve with circular data points. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

including well-balanced programs in direct detection, indirect detection, particle colliders and astrophysical probes.

In order to efficiently combine results from various dark matter searches sparked a demand for a new generation of numerical tools. MadDM is an on-going effort to bridge DM collider phenomenology with astro-physics and cosmology of DM, with the ultimate goal to provide an “all in one” dark matter phenomenology package which can be easily incorporated into the future dark matter searches at the LHC.

In our current work, we presented MadDM 2.0, which includes direct detection of dark matter in a generic UFO model. The code computes the total DM–nucleus scattering rate, recoil energy spectrum and angular distributions of dark matter elastic scattering off nuclei for various target materials, including user-provided detector resolution and detector efficiency. MadDM v.2.0 follows the “easy to use, easy to install” philosophy of the previous version, featuring great flexibility so that users can replace the existing modules with their own implementation or link MadDM functionality with other existing tools. We have performed detailed valuation of MadDM against MicrOMEGAs and private codes for various benchmark dark matter models and found excellent agreements with existing tools and literature.

In addition to relic abundance and direct detection, the future renditions of the MadDM code will also provide ability to calculate cosmic-ray fluxes from DM annihilation, including gamma rays, cosmic electrons/positrons and protons/anti-protons. Furthermore, recent improvements in aMC@NLO framework allow for computations of amplitudes for loop induced diagrams [9], giving us an opportunity to create the first automated DM tool which can calculate relic density, direct detection and indirect-detection signals originating in loop-induced processes.

Acknowledgments

We are grateful to Valentin Hirschi for helpful discussions and advice on the MadGraph5_aMC@NLO, and Adam Para and Jonghee Yoo for discussion on directional dark matter detection, and Fabio Maltoni for comments on our manuscript and encouragement on development of MadDM. We also thank Benjamin Fuks with questions regarding MSSM implementation in FeynRules. We also thank organizers and participants of MC4BSM workshops for their questions and comments during meetings. This work is supported in part by the Belgian Federal Science Policy Office through the Interuniversity Attraction Pole P7/37, by the National Research Foundation of South Africa under Grant No. 88614, and by the U.S. DOE under Grant No. DE-FG02-12ER41809, and by Durham International Junior Research Fellowships.

Appendix A. Structure of MadDM v.2.0

MadDM v.2.0 follows the directory and file structure of MadDM v.1.0. The Python module is arranged into the following simple structure:

- **Eff0Operators**: A folder containing UFO files for effective operators, which are used to extract the low energy coefficients of the $\chi q \rightarrow \chi q$ matrix elements. The UFO files were generated using the FeynRules 2.0 convention (see Section 3.2 for more details).
- **ExpData**: all experimental data used by MadDM is stored in this folder. Currently, only the LUX exclusion bound for the spin-independent DM–nucleon cross section is included. Future versions of MadDM will include a more elaborate database of experimental bounds.
- All user defined projects are automatically output into the **Projects** folder. All numerical calculations and result output is written into the **Projects** folder.
- **Templates** contains necessary FORTRAN routines and template files for relic density and direct detection. The structure of the **Template** folder serves as a skeleton for a user defined Project.
- **MGoutput.py** contains a simplified version of the MadGraph output routines, which write the matrix elements into FORTRAN files. The routines within **MGoutput.py** will write the **matrix.f** files, into the **Projects** folder, but will omit writing the additional files which are not necessary for MadDM to run, but are ordinarily written by MadGraph output routines.
- **darkmatter.py** contains the essential part of MadDM Python module. The routines within this library contain class definitions for the **darkmatter** objects, as well as numerous functions which find dark matter candidates and generate the MadDM FORTRAN module.
- A self contained example of a MadDM calculation without the use of the user interface can be found in **example.py**. The example calculation goes through the process of importing a user defined UFO model, finding the DM candidate, generating the FORTRAN module and calculating the DM–nucleon cross section and relic density.
- **init.py** contains the definitions and functions used by the MadDM user interface.
- **maddm.py** is the main Python executable. It will initiate the user interface defined in **init.py** and lead the user through the steps of DM relic density and calculations relevant for DM–nucleon scattering.

- `param_scan_default.py` is a pre-defined parameter scanning Python script. The file contains a skeleton of the Python code which will scan over DM model parameters and calculate relevant physical quantities. In order to run the parameter scanning script, a minimum of user intervention is required. All places where user input is necessary are marked, and limited to the choice of DM model parameters, number of parameter to scan over and definitions of parameter ranges.

The FORTRAN module in each `Projects` folder contains the following 7 subdirectories.

- `Cards`: A directory which contains `param_card.dat`. The structure and placement of the parameter card follows the standard formats of the MadGraph5_aMC@NLO framework.
- `Source` directory contains FORTRAN code for the DM model, as well as necessary HELAS libraries.
- `include` directory contains various include files for the FORTRAN code. The `maddm_card.inc` which contains user defined parameters of the DM relic density and direct detection calculations is located in the `include` folder. Note that any changes in the `maddm_card.inc` file will only be reflected upon the re-compilation of the MadDM FORTRAN module.
- Amplitudes for relic abundance and direct detection are saved under `matrix_elements`.
- All results (relic density, cross sections and distributions) are found in `output` directory.
- `src` directory contains the MadDM FORTRAN code. The main program is located in the `maddm.f` file.

All the other files in the main directory are executable and perform the following tasks.

- `maddm.x` is the main FORTRAN executable. Note that all output of the `maddm.x` will be written into the `output/maddm.out` file by default.
- `make_plots` is a Python script containing calls to plotting MadDM plotting routines. All calls are commented out by default. Note that it is necessary to install both Numpy and Matplotlib in order for plotting functionality to work in MadDM.
- `plotting.py` is a library of user friendly plotting functions. See `make_plots` for examples of how to use plotting functionality of MadDM. Function definitions in `plotting.py` use libraries from both Numpy and Matplotlib.

Appendix B. Description of MadDM v.2.0 routines

B.1. Python module

We upgraded the Python module of MadDM to allow users to select which calculation they would like to perform. For this purpose we introduced three switches, which are members of `darkmatter` class:

- `<darkmatter> _do_relic_density`: True if the user wants to calculate dark matter relic density, False otherwise.
- `<darkmatter> _do_direct_detection`: True if the user wants to calculate spin-dependent and spin-independent cross sections for dark matter scattering off protons and neutrons, False otherwise.
- `<darkmatter> _do_directional_detection`: True if the user wants to calculate recoil rates (as a function of energy and angle) assuming detectors made of various materials. Please note that this flag cannot be set to True if `_do_direct_detection = False`.

In order to reflect the changes in the code, we have changed the name of the function which executes the numerical module of MadDM from `CalculateRelicDensity()` to `Calculate()`. The function now takes the output file name, `output_file` as a parameter, specifying the path of the file where all the calculation results will be written. The default value for the output file is `output/maddm.out`. Changing the output file path is particularly useful if the user wishes to parallelize a parameter scan, as it avoids the problem of many jobs attempting to read/write the same file.

The only two new relevant Python functions in the `darkmatter` class are:

- `darkmatter: [excluded, excluded_omegah2, excluded_dd]`: `is_excluded(omega_min, omega_max, si_limit_x, si_limit_y)`: This function compares the results of a DM calculation obtained with the `Calculate()` function with the existing bounds on relic density and spin-independent DM–nucleon scattering cross sections. The parameters `omega_min` and `omega_max` represent the minimum and maximum Ωh^2 for which the parameter point is not ruled out while the `si_limit_x` and `si_limit_y` are one dimensional arrays of numbers representing the x and y values of the spin-independent cross section exclusion limit. The default limit implemented in MadDM is the LUX bound, which can be found in a text file inside the `ExpData` subdirectory of MadDM. **The function assumes that the DM–nucleon cross section values are given in pb.** The function returns an array of three boolean variables. `excluded` signals whether the model point is excluded overall, while `excluded_omegah2` and `excluded_dd` signal whether the model point is excluded by the individual constraint. This function should be called only after `Calculate()` has been executed.
- `darkmatter: None: GenerateDiagramsDirDetect()`: The routine generates all $\chi q \rightarrow \chi q$ diagrams, where $q = u, d, s, c, b, t$ quarks, as well as $\chi \bar{q} \rightarrow \chi \bar{q}$ diagrams and extracts the interference terms relevant for computing the low energy effective operators for χ -nucleon scattering. The function takes no values and returns no values.

Simply calling `GenerateDiagramsDirDetect()` will automatically set `_do_direct_detection` to True and result in a calculation which will go as far as χ -nucleon spin-independent and spin-dependent cross sections. The user can proceed to calculate the differential recoil rates and simulate dark matter events scattering off nuclei by also setting the flag `_do_directional_detection = True`.

B.2. FORTRAN module

In the FORTRAN module, we added a library of function definitions in `direct_detection.f` and

`directional_detection.f` files inside the `src` directory, which calculate the direct detection nucleon cross sections, differential recoil rates and other relevant quantities. The relevant function definitions are:

- `sigma_nucleon(nucleon)`: The function calculates the numerical value of a cross section for dark matter particle scattering off a nucleon at 0 momentum transfer. `nucleon=1, 0` refers to a proton/neutron respectively. The function reads in the nucleon form factors as defined in `maddm_card.inc` (see below for more details).
- `FHelm(Er, A)`: This function calculates the Helm form factor as given in Eq. (25). The function reads the recoil energy in `keVnr(Er)`, and the mass number of the target atom (A) as input parameters.

- **FWS(Er, A)**: This function calculates the Wood–Saxon form factor. The function reads the recoil energy in keVn (Er), and the mass number of the target atom (A) as input parameters.
- **FWSmass(m_DM, A)**: This function calculates the Wood–Saxon form factor. The function reads the WIMP mass in GeV (m_DM), and the mass number of the target atom (A) as input parameters.
- **coefficients(mater, coeff_s00, coeff_s01, coeff_s11)**: This routine outputs the coefficients for the polynomial fits to the nuclear structure functions S_{ij} for the spin-dependent form factor. These are referenced in Table 3. The input is the material mater which is chosen in maddm_card.inc.
- **Spin_matrix(mater, J, avg_sp, avg_sn)**: Provides the spin J and nuclear magnetic moments avg_sp and avg_sn for the different target nuclei. mater is the material chosen in maddm_card.inc.
- **Structure_Func(Er, A, mater, s00, s01, s11)**: Uses the coefficients routine to calculate the structure functions (s00, s01 and s11) that are used in the spin-dependent nuclear form factor. The input is the recoil energy Er in keVn, the atomic mass A and the target material mater.
- **FormSD(Er, A, mater)**: Calculates the spin-dependent form factor as a function of recoil energy in keVn from the structure functions in Structure_Func according to Eq. (26).
- **FormSDmass(m_DM, A, mater)**: Calculates the spin-dependent form factor as a function of WIMP mass in GeV from the structure functions in Structure_Func according to Eq. (26).
- **VE(days)**: This function calculates the velocity of the earth (in km/s) relative to the galactic center as a function of time in days according to appendix B in [41]. The function takes into account the proper motion of the solar system with respect to the surrounding stars, the motion of the Earth around the Sun and provides galactic coordinates of the Earth as it moves each day on its orbit. The input parameter days represents the day of the year relative to December 31st of the calendar year.
- **d2RdEdcostheta**
(Er, costheta, sigmawn0SI, sigmawp0SI, sigmawn0SD, sigmawp0SD, M_dm, ve):
This routine calculates the recoil rates as a function of energy and angle on a particular day of measurement, based on Eq. (24). The maddm_card.inc file contains numerical values of necessary quantities such as the target material, size of the detector (in kg), the velocity of the earth (calculated inside VE), the most probable DM velocity v_0 , the escape velocity of the DM from the galactic Halo v_{esc} , the minimum DM velocity required for recoil v_{min} all in units of km/s, and the local halo density ρ_0 , in $\text{GeV c}^{-2} \text{cm}^{-3}$. The variables sigmawp0SI and sigmawn0SI represent the spin-independent cross sections, while sigmawp0SD and sigmawn0SD are the spin-dependent cross-sections for DM scattering off protons and neutrons respectively in units of pb. M_dm and ve are the dark matter mass in GeV and the Earth's velocity relative to the Galactic center respectively.
- **target_material(material, A, Ab, Z, n)**: This subroutine provides the double differential module d2RdEdcostheta with the correct target material the user has chosen inside maddm_card.inc. The input is an integer material provided through a common block from the maddm_card.inc file, while the output is an array of mass numbers (A) for the target atoms, their respective abundances (Ab), the atomic number (Z) as well as the length of the mass number arrays (n). These are listed in Table 2. For instance, consider that the target material has been set to 1 (Xenon, which is default) in the maddm_card.inc file. Then target_material will output an array of 9 elements for the mass number of the 9 most stable Xenon isotopes and their corresponding abundance information. It will also provide $Z = 54$.
- **GaussmearAngle(uaevents, smaevents)**: This function smears a histogram of theoretical distributions (uaevents) using a Gaussian distribution. The angular resolution is set inside maddm_card.inc by changing the sig_theta parameter [10]. The default angular resolution is 30° .
- **GaussmearEnergy(uevents, smevents)**: This module applies energy smearing to the distribution of angle smeared events (uevents) using a Gaussian resolution function with uncertainty $\sigma_E = \lambda \sqrt{E_R}$, where E_R is the recoil energy in keV and λ is a constant which determines the level of smearing. The default value is $\lambda = 1$ inside maddm_card.inc. The smearing function can convert low energy recoil events into negative energies. An energy threshold cut alleviates this problem.
- **Theory_Simulation**
(mass, sigma_wnSI, sigma_wpSI, sigma_wnSD, sigma_wpSD, Theory): This subroutine uses the dark matter mass in GeV (mass) and DM–Nucleon cross-sections for the spin-independent (sigma_wpSI for proton, sigma_wnSI for neutron) and spin-dependent (sigma_wpSD for proton, sigma_wnSD for neutron) contributions to simulate the theoretical double differential distribution (with respect to recoil energy and recoil angle) and outputs the result as an array named Theory representing the distribution of unsmeared recoil events. In this routine, the double differential rate is approximated at the center of each energy–angle–day bin.
- **eff(x)**: Function that interpolates the data points from LUX for the nuclear recoil efficiency. The input x is the recoil energy (in keV) and the output is the detector efficiency at that energy.
- **efficiency(x, flag)**: Provides the DM detection efficiency as a function of recoil energy. The variable x represents the recoil energy in keV and flag is an integer variable which, when chosen as 0, gives a constant 100% detection efficiency. If flag is 1, the efficiency is 50% and if flag is 2, the efficiency is that obtained from the function eff, which is the nuclear recoil detection efficiency from the LUX experiment in Fig. 1 of Ref. [23]. For a conservative estimate we extend and interpolate the efficiency to 100 keV. New detector efficiency may be implemented by user in place of this function definition.
- **directional_detection**
(mass, sigma_wnSI, sigma_wpSI, sigma_wnSD, sigma_wpSD, flag, N_events): This routine calculates the differential distributions of dark matter recoil rates given the spin-independent (sigma_wpSI for proton, sigma_wnSI for neutron) and spin-dependent (sigma_wpSD for proton, sigma_wnSD for neutron) cross sections in pb as well as the DM mass in GeV (mass). The subroutine calls the theory_simulation routine which performs the event distribution simulation. If the user specified smearing = .true. in the maddm_card.inc file the function will incorporate detector smearing into the resulting distributions. The function calculates the double differential distribution (as a function of energy and angle), and integrates over angle and time to obtain the differential energy distribution ($\frac{dN}{dE}$). The energy distribution is weighted with the detector efficiency function efficiency. Furthermore, the function also integrates over energy and day to obtain the differential angular distribution ($\frac{dN}{d\cos\theta}$) as well as the total rate ($\frac{dN}{dt} = R$) after integrating over energy and angle. All the distributions are written to separate files in the user defined project output directory. The total number of events for a 1 ton detector over the course of 1 year is stored in the variable N_events. The integer variable flag is used to choose the type of efficiency to be used.

In addition to functions which calculate quantities relevant for DM–nucleon scattering, we added a number of important variables to the FORTRAN module, in the `maddm_card.inc` file inside the include directory of the project folder⁶:

- `do_relic_density`: A logical flag which determines whether a relic density calculation should be performed. It can be either `.true.` or `.false.`. If users have not generated the FORTRAN module with the relic density calculation (i.e. there is no call to `GenerateDiagrams()` in the Python module), setting this flag to `.true.` will result in an error.
- `do_direct_detection`: A logical flag which determines whether a direct detection calculation should be performed. It can be either `.true.` or `.false.`. If users have not generated the FORTRAN module with the direct detection calculation (i.e. there is no call to `GenerateDiagramsDirDetect()` in the Python module), setting this flag to `.true.` will result in an error.
- `do_directional_detection`: A logical flag which determines whether to calculate differential distributions for recoil rates (both energy and angle dependence). It can be either `.true.` or `.false.`. If users have not generated the FORTRAN module with the direct detection calculation (i.e. there is no call to `GenerateDiagramsDirDetect()` in the Python module), setting this flag to `.true.` will result in an error. In addition, one cannot set to
`do_directional_detection = .true.` if
`do_direct_detection = .false.`
- `SPu...` `SPg`, `SNu...` `SNg`: Values for the **scalar** nucleon form factors for the proton (`SPx`) and neutron (`SNx`). The nucleon constituent ($u, d, \dots, g$) is labeled by the last letter in the variable name.
- `VPu`, `VPd`, `VNu`, `VNd`: Values for the **vector** nucleon form factors for the proton (`VPx`) and neutron (`VNx`). The nucleon constituent ($u, d, \dots, g$) is labeled by the last letter in the variable name.
- `AVPu...` `AVPs`, `AVNu...` `AVNs`: Values for the **axial-vector** nucleon form factors for the proton (`AVPx`) and neutron (`AVNx`). The nucleon constituent ($u, d, \dots, g$) is labeled by the last letter in the variable name.
- `SigPu...` `SigPs`, `SigNu...` `SigNs`: Values for the **tensor** nucleon form factors for the proton (`SigPx`) and neutron (`SigNx`). The nucleon constituent ($u, d, \dots, g$) is labeled by the last letter in the variable name.
- `material`: Variable which selects the detector material. It can take values 1–13 for the following implemented materials: 1–Xe, 2–Ge, 3–Si, 4–Ar, 5–Ne, 6–Na, 7–I, 8–C, 9–F, 10–S. For compound materials set `mater_comp` as `.true.` in `maddm_card.inc`, 11–NaI, 12–CF₄ and 13–CS₂.
- `vMP`: Most probable speed for dark matter particles in the galactic halo in km/s, assuming the Maxwell–Boltzmann velocity distribution. The default value of `vMP` is $v_0 = 220$ km/s.
- `vescape`: Escape velocity for dark matter in the galactic halo in km/s, the default value is $v_{\text{esc}} = 650$ km/s.
- `rhoDM`: Mass density of dark matter in the halo in $\text{GeV c}^{-2} \text{cm}^{-3}$, default value is $0.3 \text{ GeV c}^{-2} \text{cm}^{-3}$.
- `detector_size`: Mass of the dark matter detector target in kg, the default value is 1000 kg.
- `En_threshold`: The recoil energy threshold cut (in keVnr).
- `lambda`: Energy resolution parameter defined in $\sigma_E = \lambda \sqrt{E}$ with $\lambda = 1$ as default.

- `sig_theta`: Angular resolution of the detector (in degrees) with default value of $\sigma_\theta = 30^\circ$.
- `smearing`: A logical flag which determines whether the results should be smeared by detector effects. If set to `.true.` it will smear both the recoil energy and angle.

For the purpose of dark matter scattering event simulations, we also incorporated a number of parameters which are used as inputs for the nuclear recoil scattering simulation:

- `En_min`, `En_max`: Minimum and maximum recoil energy in keVnr.
- `cos_min`, `cos_max`: Minimum and maximum cosine of the recoil angle.
- `day_min`, `day_max`: Starting and ending day of the experiment relative to December 31st.
- `Energy_bins`, `cos_theta_bins`, `day_bins`: The number of bins in histograms for scattering rate as a function of recoil energy, cosine angle and day of experiment respectively.

B.3. Test routines

For debugging purposes we also provide a set of test routines and functions which can be used in `maddm.f` and are commented out by default. Simply un-commenting the calls and re-compiling the FORTRAN module will make the test functions accessible. The list of test function includes the following:

- `d2RdEdcos_test` (`Er`, `costheta`, `sigmawn0SI`, `sigmawp0SI`, `sigmawn0SD`, `sigmawp0SD`, `M_dm`, `day`): This subroutine uses the `d2RdEdcostheta` routine described in the previous section to print out all the ingredients that go into the double differential function, for consistency checking. The input variables are defined in `d2RdEdcostheta` above. The output is printed on the screen and includes: DM mass, the spin-dependent and spin-independent cross-sections, the target material used as well as its mass numbers and the isotope abundances, the detector size and other variables that factor into Eq. (24).
- `V_E_test` (`day`, `num_array`): This routine provides the velocity of the Earth through the galactic halo on a specific day relative to December 31st. `num_array` is an integer flag. When set to 0, `V_E_test` prints one number representing the velocity on the day specified by `day`. However when `num_array` is set to 1, the routine outputs an array of days and velocities (in the galactic frame) with the minimum and the maximum days set in the `maddm_card.inc` file. The output file is written in `output/VE_test.dat`.
- `Form_test` (`Er`, `num_array`): Outputs the spin-dependent and spin-independent form factors for the most abundant elements. It uses the subroutine `target_MA`, which is exactly the same as the routine `target_material` above, but only provides the most abundant elements. When `num_array` is 0, the output is the form factor at a specific input recoil energy. When `num_array` is 1, the output is an array of the form factor squared in the file `output/formfac_test.dat`. The first column is the recoil energy, the second is the spin-independent form factor squared while the third is the spin-dependent form factor squared.
- `Form_testmass` (`m_DM`, `num_array`): Outputs the spin-dependent and spin-independent form factors for the most abundant elements. It uses the subroutine `target_MA`. When `num_array` is 0, the output is the form factor at a specific input WIMP mass in GeV. When `num_array` is 1, the output is an array of the form factor squared in the file `output/formfacmass_test.dat`. The first column is the WIMP mass, the second is the spin-independent form factor squared while the third is the spin-dependent form factor squared.

⁶ Please note that one must recompile the numerical code by typing `make` in the project folder in order for any changes to the `maddm_card.inc` file to take effect properly.

- `LUX_Exclusion(xmin, xmax, ymin, ymax, step)`: Scans the DM–nucleon vs Mass parameter space to provide the LUX exclusion limit. This is done by using the `flag = 2` in the `directional_detection` routine, which ensures that we use the efficiency function obtained from the LUX 2013 results [23]. `xmin` represents the minimum mass, `xmax`, the maximum mass, `ymin` is the minimum cross-section and `ymax` is the maximum cross-section in the scanning parameter range. `nstep` is the number of scanning points for each direction. The scanning is done in log space, the linear scanning is commented out, for linear scanning, one has just to uncomment the `Lux_exclusion` line in the test routine part of `maddm.f`

References

- [1] L. Bergstrom, Ann. Phys. 524 (2012) 479. [arXiv:1205.4882](#) [astro-ph.HE].
- [2] J. Abdallah, A. Ashkenazi, A. Boveia, G. Busoni, A. De Simone, et al. 2014. [arXiv:1409.2893](#) [hep-ph].
- [3] S. Arrenberg, H. Baer, V. Barger, L. Baudis, D. Bauer, et al. 2013. [arXiv:1310.8621](#) [hep-ph].
- [4] S. Ask, N.D. Christensen, C. Duhr, C. Grojean, S. Hoeche, et al. 2012. [arXiv:1209.0297](#) [hep-ph].
- [5] P. Gondolo, J. Edsjo, P. Ullio, L. Bergstrom, M. Schelke, et al., J. Cosmol. Astropart. Phys. 0407 (2004) 008. [arXiv:astro-ph/0406204](#) [astro-ph].
- [6] G. Belanger, F. Boudjema, A. Pukhov, 2014. [arXiv:1402.0787](#) [hep-ph].
- [7] M. Backovic, K. Kong, M. McCaskey, Phys. Dark Universe 5–6 (2014) 18. [arXiv:1308.4955](#) [hep-ph].
- [8] J. Alwall, M. Herquet, F. Maltoni, O. Mattelaer, T. Stelzer, J. High Energy Phys. 1106 (2011) 128. [arXiv:1106.0522](#) [hep-ph].
- [9] J. Alwall, R. Frederix, S. Frionone, V. Hirschi, F. Maltoni, et al., J. High Energy Phys. 1407 (2014) 079. [arXiv:1405.0301](#) [hep-ph].
- [10] G. Mohlabeng, K. Kong, J. Li, A. Para, J. Yoo, 2015. [arXiv:1503.03937](#) [hep-ph].
- [11] C.J. Copi, J. Heo, L.M. Krauss, Phys. Lett. B461 (1999) 43. [arXiv:hep-ph/9904499](#) [hep-ph].
- [12] P. Gondolo, Phys. Rev. D66 (2002) 103513. [arXiv:hep-ph/0209110](#) [hep-ph].
- [13] P. Grothaus, M. Fairbairn, J. Monroe, Phys. Rev. D90 (2014) 055018. [arXiv:1406.5047](#) [hep-ph].
- [14] P. de Aquino, W. Link, F. Maltoni, O. Mattelaer, T. Stelzer, Comput. Phys. Comm. 183 (2012) 2254. [arXiv:1108.2041](#) [hep-ph].
- [15] C. Degrande, C. Duhr, B. Fuks, D. Grellscheid, O. Mattelaer, et al., Comput. Phys. Comm. 183 (2012) 1201. [arXiv:1108.2040](#) [hep-ph].
- [16] P.Z. Skands, B. Allanach, H. Baer, C. Balazs, G. Belanger, et al., J. High Energy Phys. 0407 (2004) 036. [arXiv:hep-ph/0311123](#) [hep-ph].
- [17] B. Allanach, C. Balazs, G. Belanger, M. Bernhardt, F. Boudjema, et al., Comput. Phys. Comm. 180 (2009) 8. [arXiv:0801.0045](#) [hep-ph].
- [18] C. Faham, LUX Collaboration, (2014). [arXiv:1405.5906](#) [hep-ex].
- [19] A. Brown, XENON, AIP Conf. Proc. 1604 (2014) 313.
- [20] T. Saab, CDMS, AIP Conf. Proc. 1604 (2014) 436.
- [21] C.E. Aalseth, P.S. Barbeau, J. Colaresi, J.J. Collar, J. Diaz Leon, J.E. Fast, N. Fields, T.W. Hossbach, M.E. Keillor, J.D. Kephart, A. Knecht, M.G. Marino, H.S. Miley, M.L. Miller, J.L. Orrell, D.C. Radford, J.F. Wilkerson, K.M. Yocum, Phys. Rev. Lett. 107 (2011) 141301.
- [22] R. Bernabei, et al., DAMA Collaboration, LIBRA Collaboration, Eur. Phys. J. C67 (2010) 39. [arXiv:1002.1028](#) [astro-ph.GA].
- [23] D. Akerib, et al., LUX, Phys. Rev. Lett. 112 (2014) 091303. [arXiv:1310.8214](#) [astro-ph.CO].
- [24] B.A. Dobrescu, D. Hooper, K. Kong, R. Mahbubani, J. Cosmol. Astropart. Phys. 0710 (2007) 012. [arXiv:0706.3409](#) [hep-ph].
- [25] M. Freytsis, Z. Ligeti, Phys. Rev. D83 (2011) 115009. [arXiv:1012.5317](#) [hep-ph].
- [26] G. Belanger, F. Boudjema, A. Pukhov, A. Semenov, Comput. Phys. Comm. 180 (2009) 747. [arXiv:0803.2360](#) [hep-ph].
- [27] D.N. Spergel, Phys. Rev. D37 (1988) 1353.
- [28] J. Beringer, et al., Particle Data Group, Phys. Rev. D86 (2012) 010001.
- [29] A. Airapetian, et al., HERMES, Phys. Rev. D75 (2007) 012007. [arXiv:hep-ex/0609039](#) [hep-ex].
- [30] E. Ageev, et al., Compass, Phys. Lett. B647 (2007) 330. [arXiv:hep-ex/0701014](#) [hep-ex].
- [31] S. Aoki, M. Doui, T. Hatsuda, Y. Kuramashi, Phys. Rev. D56 (1997) 433. [arXiv:hep-lat/9608115](#) [hep-lat].
- [32] M. Drees, M. Nojiri, Phys. Rev. D48 (1993) 3483. [arXiv:hep-ph/9307208](#) [hep-ph].
- [33] J. Hisano, R. Nagai, N. Nagata, J. High Energy Phys. 1505 (2015) 037. [arXiv:1502.02244](#) [hep-ph].
- [34] D. Santos, J. Billard, G. Bosson, J. Bouly, O. Bourrion, et al., J. Phys. Conf. Ser. 460 (2013) 012007. [arXiv:1304.2255](#) [physics.ins-det].
- [35] E. Daw, J. Fox, J.-L. Gauvreau, M. Gold, L. Harmon, et al., J. Instrum. 9 (2014) P07021. [arXiv:1307.5525](#) [physics.ins-det].
- [36] G. Sciolla, et al. DM-TPC collaboration, (2008). [arXiv:0806.2673](#) [astro-ph].
- [37] D. Nygren, J. Phys. Conf. Ser. 460 (2013) 012006.
- [38] J. Billard, L. Strigari, E. Figueroa-Feliciano, Phys. Rev. D89 (2014) 023524. [arXiv:1307.5458](#) [hep-ph].
- [39] J.H. Davis, J. Cosmol. Astropart. Phys. 1503 (2015) 012. [arXiv:1412.1475](#) [hep-ph].
- [40] C.J. Copi, L.M. Krauss, Phys. Rev. D63 (2001) 043507. [arXiv:astro-ph/0009467](#) [astro-ph].
- [41] J. Lewin, P. Smith, Astropart. Phys. 6 (1996) 87.
- [42] M. Ressel, D. Dean, Phys. Rev. C56 (1997) 535. [arXiv:hep-ph/9702290](#) [hep-ph].
- [43] V. Dimitrov, J. Engel, S. Pittel, Phys. Rev. D51 (1995) 291. [arXiv:hep-ph/9408246](#) [hep-ph].
- [44] V. Bednyakov, Simkovic F, Phys. Part. Nucl. 37 (2006) S106. [arXiv:hep-ph/0608097](#) [hep-ph].
- [45] P. Divari, T. Kosmas, J. Vergados, L. Skouras, Phys. Rev. C61 (2000) 054612.
- [46] S. Arrenberg, L. Baudis, K. Kong, K.T. Matchev, J. Yoo, Phys. Rev. D78 (2008) 056002. [arXiv:0805.4210](#) [hep-ph].
- [47] S. Arrenberg, L. Baudis, K. Kong, K.T. Matchev, J. Yoo, 2013. [arXiv:1307.6581](#) [hep-ph].
- [48] G. Servant, T.M. Tait, New J. Phys. 4 (2002) 99. [arXiv:hep-ph/0209262](#) [hep-ph].
- [49] H.-C. Cheng, J.L. Feng, K.T. Matchev, Phys. Rev. Lett. 89 (2002) 211301. [arXiv:hep-ph/0207125](#) [hep-ph].
- [50] B. Allanach, M. Battaglia, G. Blair, M.S. Carena, A. De Roeck, et al., Eur. Phys. J. C25 (2002) 113. [arXiv:hep-ph/0202233](#) [hep-ph].