



DarkSUSY DarkSUSY_main

Manual and description of routines

Documentation harvested from documentation and source directories with harvestdoc.pl

Fri Dec 9 22:04:57 2022

<http://www.darksusy.org>

**Joakim Edsjö^{a*}, Torsten Bringmann^{b†}, Paolo Gondolo^{c‡},
Piero Ullio^{d§}, Lars Bergström^{a¶}, Mia Schelke^{||},
Edward A. Baltz^{**} and Gintaras Duda^{††}**

^a *The Oskar Klein Centre for Cosmoparticle Physics, Department of Physics, Stockholm University, AlbaNova, SE-106 91 Stockholm, Sweden.*

^b *Department of Physics, University of Oslo, Box 1048, NO-0371 Oslo, Norway*

^c *Department of Physics, University of Utah, 115 South 1400 East, Suite 201, Salt Lake City, UT 84112-0830, USA.*

^d *SISSA and INFN, Sezione di Trieste, via Bonomea 265, 34136 Trieste, Italy.*

*E-mail address: edsjo@fysik.su.se

†E-mail address: torsten.bringmann@fys.uio.no

‡E-mail address: paolo.gondolo@utah.edu

§E-mail address: ullio@he.sissa.it

¶E-mail address: lbe@fysik.su.se

||E-mail address: schelke@...

**E-mail address: eabaltz@physics.columbia.edu

††E-mail address: gkduda@creighton.edu

Abstract

DarkSUSY is a package for numerical dark matter calculations including, but not limited to, supersymmetric dark matter. This manual describes the theoretical background as well as details about the actual routines. Everything is not covered, but it should hopefully prove useful if you need more information than in our published articles.

Disclaimer. This manual is work in progress.

We try to keep it clear and up-to-date with the code, but there will be cases where changes/improvements in the code, for one reason or another, will not have propagated into the manual. Hence, check the actual code if you want to be certain about how a given process/feature is implemented.

Contents

I	Prelude	8
1	Introduction	9
2	Quick start guide	10
2.1	Installation	10
2.1.1	Options for install	11
2.1.2	System requirements	11
2.2	Example programs	11
2.2.1	Auxiliary example programs	12
2.2.2	Making your own example programs	13
2.3	Modifying individual subroutines or functions	13
3	Guiding principles of the code	14
3.1	The main DarkSUSY library <code>ds_core</code>	14
3.2	Particle physics modules	15
3.2.1	Using a particle physics module	17
3.2.2	Adding a new particle physics module	17
3.3	Halo models	18
3.4	Interface functions	18
3.5	Commonly used functions	19
3.6	Replaceable functions	19
4	Comparison to previous DarkSUSY versions	22
5	Original articles	24
II	Main DarkSUSY routines in <code>src/</code>	25
6	<code>an_yield</code>: Annihilation yields in the halo – yields from simulations	26
6.1	Annihilation in the halo, yields – theory	26
6.1.1	Monte Carlo simulations	26
7	<code>aux</code>: General routines	28
7.1	General routines	28
8	<code>aux_xdiag</code>: Diagonalization routines	29
8.1	XDIAG	29
9	<code>cr_aux</code>: Cosmic rays – general	30
9.1	Cosmic Rays – auxiliary routines	30

10 cr_axi: Cosmic rays – diffusion routines for axisymmetric distributions	31
10.1 Cosmic ray propagation in axially symmetric halos	31
11 cr_gamma: Cosmic rays – Gamma fluxes	34
11.1 Gamma rays from the halo – theory	34
11.2 Continuous gamma yields and line signals	35
11.3 Fluxes	35
11.4 Gamma rays from the halo – routines	36
12 cr_nu: Cosmic rays – Neutrino fluxes	38
12.1 Neutrino fluxes from the halo – theory	38
13 cr_ps: Cosmic rays – point sources	39
13.1 Cosmic Ray propagation for point sources	39
14 dd: Direct detection	40
14.1 Direct detection – theory	40
14.2 Cosmic-ray upscattered dark matter	40
15 dd_crdm: Direct detection – cosmic ray upscattered DM	41
15.1 Upscattering of dark matter by cosmic rays	41
15.2 CRDM– routines	42
16 dmd_astro: Astrophysical source functions	44
17 dmd_mod: Dark matter distributions	45
17.1 Dark matter distributions – theory	45
17.1.1 Rescaling of the WIMP density	45
17.2 Implementation in DarkSUSY	46
17.2.1 Dark matter distributions – routines	47
18 dmd_vel: Dark matter velocity distributions	48
18.1 Dark matter phase-space distributions – theory	48
19 fi: Freeze-In	49
19.1 Freeze-in – theory	49
19.2 Freeze-in – routines	50
20 ini: Initialization routines	52
20.1 Initialisation routines	52
21 kd: Kinetic decoupling	53
21.1 Kinetic decoupling and microhalos (kd) – theory	53
21.1.1 Kinetic decoupling	53
21.1.2 The smallest protohalos	54
21.2 Kinetic decoupling – routines	55
22 rd: Relic density	56
22.1 Relic density – theoretical background	56
22.1.1 The Boltzmann equation and thermal averaging	56
22.1.2 Review of the Boltzmann equation with coannihilations	56
22.1.3 Thermal averaging	58
22.1.4 Reformulation of the Boltzmann equation	60
22.2 Relic density – numerical integration of the density equation	63

22.3	Relic density – asymmetric dark matter	64
22.4	Relic density – coupled Boltzmann equations	66
22.4.1	Boltzmann equation at phase-space level	66
22.4.2	Boltzmann equations for number density and velocity dispersion	66
22.5	Relic density – tabulation of W_{eff}	67
22.5.1	Method A: pre-tabulation of W_{eff}	68
22.5.2	Method B: Breit-Wigner fit to resonances and tabulation on the fly	68
22.5.3	Method C: full expression of W_{eff}	69
22.5.4	Method D: quick and dirty	69
22.5.5	Comparison of methods	70
22.5.6	Time constraints	70
22.6	Relic density – routines	70
22.6.1	Global parameters	71
22.6.2	Brief description of internal routines for standard Boltzmann approach	72
22.6.3	Implementation of coupled Boltzmann equations	74
23	se_aux: Auxiliary routines for WIMP annihilation in the Sun/Earth	75
23.1	Sun and Earth models – auxiliary routines	75
24	se_mod: Sun and Earth models	76
24.1	Sun and Earth models – theory	76
24.2	Sun and Earth models – routines	76
25	se_nu: Capture and annihilation in the Sun/Earth	78
25.1	Neutrinos from the Sun and Earth – theory	78
25.1.1	Neutrino yield from annihilations	78
25.1.2	Evolution of the number density in the Earth/Sun	79
25.1.3	Approximate capture rate expressions	80
25.1.4	Earth and Sun composition	81
25.1.5	More accurate capture rate expressions	82
25.1.6	Accurate capture rates in the Earth for general velocity distributions	82
25.1.7	Accurate capture rates for the Earth for a Maxwell-Boltzmann velocity distribution	84
25.1.8	Effects of WIMP diffusion in the solar system	85
25.2	Neutrinos from Sun and Earth – routines	86
26	se_yield: Yields from annihilation in the Sun/Earth	87
26.1	Muon yields from annihilation in the Earth/Sun – theory	87
26.1.1	Monte Carlo simulations with WimpSim	87
27	si: Self-interactions	92
27.1	Dark matter self-interactions (si) – theory	92
27.2	Self-interactions – routines	94
III	Particle physics modules in src_models	95
28	Basic principles and common routines	96
28.1	common/aux: auxiliary routines	96
28.2	common/sm: standard model	96
29	The empty model	98

30	Generic decaying dark matter	99
31	Generic FIMPs	100
31.1	generic_fimp/fi: Freeze-in routines	101
32	Generic WIMPs	102
32.1	Generic WIMP and simple extensions	102
32.2	Asymmetric dark matter	103
33	The Minimal Supersymmetric Standard Model	104
33.1	mssm/ac: Accelerator constraints	105
33.1.1	Accelerator bounds	105
33.2	mssm/an: (Co-)annihilation cross sections	106
33.2.1	Annihilation cross sections – theory	106
33.2.1.1	Annihilation cross sections	106
33.2.1.2	Coannihilation diagrams	106
33.2.1.3	Neutralino and chargino annihilation	106
33.2.1.4	Squark-squark annihilation	107
33.2.1.5	Squark-neutralino annihilation	111
33.2.1.6	Squark-chargino annihilation	111
33.2.1.7	Degrees of freedom	112
33.2.2	Annihilation routines - general remarks	112
33.2.2.1	General routines	113
33.2.2.2	Neutralino and chargino (co)annihilation cross sections	113
33.3	mssm/an_1l: Annihilation cross sections (1-loop)	113
33.3.1	Annihilation cross sections at 1-loop – general	113
33.4	mssm/an_ib: Internal bremsstrahlung	113
33.4.1	Internal Bremsstrahlung (IB) – theory	113
33.4.1.1	General considerations	114
33.4.1.2	IB from neutralino annihilations	114
33.4.1.3	The implementation in DarkSUSY	115
33.5	mssm/an_sf: Annihilation cross sections (with sfermions)	116
33.5.1	Annihilation cross sections with sfermions – general	116
33.6	mssm/an_stu: t , u and s diagrams for ff -annihilation	116
33.6.1	Annihilation amplitudes for fermion-fermion annihilation	116
33.7	mssm/dd: Direct detection	117
33.7.1	Direct detection – theory	117
33.8	mssm/ge: General SUSY model setup: masses, vertices etc	119
33.8.1	Supersymmetric model	119
33.8.1.1	Parameters	119
33.8.1.2	Mass spectrum	119
33.8.1.3	Three-particle vertices	122
33.8.2	General supersymmetry – routines	124
33.9	mssm/ge_cmssm: cMSSM interface (Isasugra)	125
33.9.1	mSUGRA (ISASUGRA) interface to DarkSUSY	125
33.10	mssm/ge_slha: SUSY Les Houches Accord interface	125
33.10.1	SUSY Les Houches Accord	125
33.11	mssm/ini: Initialization routines	125
33.11.1	Initialization routines	125
33.12	mssm/rd: Relic density	126
33.12.1	Relic density of neutralinos	126
33.12.2	Internal degrees of freedom	126

33.12.2.1 Neutralino-chargino annihilation	126
33.12.2.2 Chargino-chargino annihilation	127
33.12.2.3 Neutralino-sfermion annihilation	128
33.12.2.4 Chargino-sfermion annihilation	129
33.12.2.5 Sfermion-sfermion annihilation	129
33.12.2.6 Squark-squark annihilation	130
33.12.2.7 Sfermion-squark annihilation	131
33.12.2.8 Summary of degrees of freedom	131
33.13 Relic density – more details on routines	131
33.13.1 Global parameters	131
34 Silveira-Zee (Scalar Singlet)	133
34.1 silveira_ze/fi: Freeze-in routines	133
35 Self-Interacting Dark Matter	134
35.1 vdSIDM/an: Annihilation routines	134
35.2 vdSIDM/cr: Cosmic rays	135
35.3 vdSIDM/ini: Model setup	135
35.4 vdSIDM/kd: Kinetic decoupling	135
35.5 vdSIDM/rd: Relic density	136
35.6 vdSIDM/si: Dark matter self-interactions	136
Bibliography	136

Part I
Prelude

Chapter 1

Introduction

DarkSUSY is a set of Fortran routines to allow advanced numerical calculations connected to dark matter physics. In part I of this Manual we explain the basic structure of the code, and provide an introduction on how to get quickly started and use it. Part II is devoted to a description of the DarkSUSY *core library*, which contains all routines that are completely independent of the underlying DM particle physics. In part III, we describe the additional libraries, or *particle modules*, for specific DM models that are currently implemented (including supersymmetric dark matter in the Minimal Supersymmetric Standard Model, the MSSM). The modular structure of the code allows to easily link the core library to any of those particle modules, as well as to add user-defined new modules in a straight-forward way.

In this manual we will mainly cover the more technical aspects of DarkSUSY, i.e. how to call different subroutines, both particle-physics dependent and not, and how to change various switches and options in more advanced routines. We will only briefly review the necessary physics involved when needed and refer the reader to [1, 2] and the original papers behind DarkSUSY (see Section 5) for more details. If you use DarkSUSY please consider the original physics work behind and give proper credit to [2] and the relevant references in Section 5. If you use non-standard options, e.g. a different propagation model for antiprotons, please remember to give proper credit to that model.

Chapter 2

Quick start guide

2.1 Installation

To get started, first download DarkSUSY from www.darksusy.org and unpack the tar file. To compile, run the following in the folder where you unpacked it

```
./configure  
make
```

You will then have compiled the main DarkSUSY library, as well as all supplied particle physics modules. To test whether the installation was successful, type

```
cd examples/test  
./dstest
```

The program will take up to a minute to run and reports if there are any problems.*
After about a minute's runtime, you should get an output of the form

```
=====
```

Summary of performed DarkSUSY tests

```
=====
```

Number of errors reported by dstest_mssm:	0
Number of errors reported by dstest_silveira_zee:	0
Number of errors reported by dstest_genWIMP:	0

```
=====
```

If you get something else than 0 errors, you should check the output more carefully. The way the test program runs is that for each observable it compares the result with a pre-calculated value.

*Strictly speaking, this is only a test of the default particle physics module, `mssm`. To see what happens behind the scenes, you can run in verbose mode by replacing `'testlevel/2/'` with `'testlevel/1/'` at the beginning of `dstest.f`. Then type `make` and run `dstest` again. The program is also extensively commented, and can be used to learn about which DarkSUSY routines to call.

If the difference is larger than 0.3% an error is issued. You normally would not expect larger differences than this due to just numerical errors.

Even if you now have DarkSUSY running, it is more fun to start doing some calculations on your own. Possible next steps are explained in the next Sections.

Happy running!

2.1.1 Options for install

Even if the above install usually works, sometimes you want to use special options, like a specific compiler. Most options are specified at the time of configure and then propagated through to the DarkSUSY makefiles. Examples of options are

- To specify that you want to compile with `gfortran`, you can e.g. run `configure` with the following options (on one line)

```
./configure CC=gcc CFLAGS=-g CXX=g++ CXXFLAGS=-g FC=gfortran
FCFLAGS="-O -ffixed-line-length-none -fopenmp"
```

For your convenience, this particular choice is available as a script `conf.gfortran` that can be invoked instead of the string above.

2.1.2 System requirements

DarkSUSY should run on most Linux/Unix systems including Mac OS X. You need to have a Fortran, C and C++ compiler available and the typical developer tools (like `make` and `ranlib`). You also need to have `perl` installed. If you are creating new particle physics modules and want to use the tools available to automatically create makefiles for you, you also need to have `autoconf` installed.

External, or ‘contributed’, code residing in `/contrib` typically needs further packages installed, and relies on more specific system requirements. As of DarkSUSY version 6.2, the additional requirements include `curl` (+`libcurl!`), `autoconf`, `aclocal` and `cmake`. Note that these requirements may change, *please consult the webpage for the most up-to date list!* Experience has shown that some of the contributed packages are notoriously challenging to build on some systems, such that the standard `make` command does not even proceed to the actual DarkSUSY code. In case you cannot resolve these errors, the make system offers a lightweight version of DarkSUSY that does not depend on contributed packages at all. For this, just do `make distclean` followed by `configure` and then

```
make darksusy_light
```

instead of the usual `make`. No contributed code will be built in this case (and particle modules heavily relying on these external packages, like the `mssm` module, will be disabled).

2.2 Example programs

DarkSUSY is primarily a library that is intended to be used with your own main programs. However, to get you started, we supply a few sample programs in the `/examples` directory. These can be used as they are, but they are also extensively commented to help you understand which routines you are supposed to call for the most typical calculations. The most instructive general-purpose programs in `/examples`, apart from the already mentioned `/test/dstest`, are

dsmain_wimp.F An example main program to calculate various DM observables. It can be the starting point if you want to write your own programs (see below). Note that you can run *the same code* with different particle modules that provide a WIMP DM candidate (the default is the `mssm` module); simply do

```
./> make -B dsmain_wimp DS_MODULE=<MY_MODULE>
```

and then `./dsmain_wimp` in the `/examples` directory. Choosing `<MY_MODULE>=generic_wimp`, e.g., will produce results for a generic WIMP DM candidate (see Chapter 32 for details on the implementation).

dsmain_decay.f A main program that calculates the same observables, where relevant, as `dsmain_wimp.F` – but for a generic decaying DM candidate (see Chapter 30 for details on the implementation) rather than a WIMP.

2.2.1 Auxiliary example programs

In the folder `examples/aux/` we list a set of instructive auxiliary example programs that illustrate more specific usage of DarkSUSY. These example programs are typically set up for specific particle physics modules (e.g. `generic_wimp`) but, as explained below in more detail, it is straightforward to use them for other particle physics modules as well. Below we describe some of the currently available auxiliary example programs – but *note that the actual list of example programs in `examples/aux/` is growing with almost every, even minor, release*. We thus strongly advise to check the actual list of these programs, and their headers for a quick summary.

flxconv.f This program can be used to convert between different fluxes from the Sun and the Earth. It can be used to convert a limit on a muon flux to a limit on the scattering cross section, or to a limit on the annihilation rate (or vice versa).

DMhalo_predef.f This program illustrates how to use the default version of `dsdmsdriver` (provided with the DarkSUSY release) to load additional profiles into the currently active halo database by using its pre-defined halo parameterizations.

DMhalo_table.f This program demonstrates how to load a halo profile from a table in an accompanying data file.

DMhalo_new.f This program demonstrates how to correctly extend `dsdmsdriver` when adding a new profile parameterization in order to consistently make it available to all DarkSUSY routines that rely on the DM density (in this concrete example, we add the spherical Zhao profile [3], aka $\alpha\beta\gamma$ profile).

DMhalo_bypass.f Here, we demonstrate a work-around of completely bypassing the default `dsdmsdriver` setup when switching to a user-provided new DM density profile. While easier to implement than `DMhalo_new.f`, such an approach has the significant drawback that the advanced DarkSUSY system of automatic tabulation of quantities related to DM rates cannot easily be exploited.

generic_wimp_oh2.f This program calculates the annihilation cross section needed to produce the correct relic density (as measured by Planck within errors). It does this for a range of WIMP masses so that you can plot e.g. the required annihilation cross section versus mass. It also lets you incorporate threshold effects (either as a hard cut, default), or with a more sophisticated sub-threshold treatment, that also illustrates the use of replaceable functions.

ScalarSinglet_RD.f This program calculates the couplings required to get the correct relic density in the Silveira-Zee (scalar singlet) model.

ucmh_test.f This program is an example of how the ultra-compact mini halo routines can be used.

wimpyields.f This is a simple program that sets up a generic WIMP with a given mass that annihilates to a given channel and then calculates the yields of different particles from the hadronization/decay of the annihilation products.

caprates.f This is a simple program that scans a range of WIMP masses and calculates the capture rates in the Sun via spin-independent and spin-dependent scattering.

caprates_ff.f This program is similar to `caprates.f`, but a bit more advanced as it performs the capture rate calculation with the most complete numerical setup and also calculates the capture rate on individual elements in the Sun.

2.2.2 Making your own example programs

The simplest way to create your own example program is to copy one of the Fortran example files in `examples/aux/` to a directory of your choice, then change the name of that file and modify it. You also need to copy the makefile from `examples/aux/` to the same directory, and make sure that you update the name of the file that you just changed.[†] Running ‘make’ then compiles your own new example program without touching the release version of the DarkSUSY code. Always keeping your own code, or your modifications of DarkSUSY, separate like this is the recommended way to proceed because it facilitates debugging and minimizes the risk of introducing errors. Don’t modify any of the DarkSUSY makefiles directly as they are overwritten every time you run `configure`.

If you want to *compile with a different particle physics module*, you will in general need to do go through three simple steps:

1. change the corresponding block in the makefile, i.e. simply set the variable `DS_MODULE` to a different value (you may also have to add additional libraries, e.g. `lisajet` for the `mssm` module)
2. update the model setup routines to the new particle module (see the various example programs; for more details, have a look at the respective Section of this manual and the header of setup routines starting with `dsgivemodel`).
3. make sure that your main program only calls routines that are supported for the new particle module. (If it doesn’t, it will not compile, stating the functions that are not supported)

An explicit demonstration of how all these three steps are taken care of in one single example is provided in `dsmain_wimp.F`. Note that the pre-compiler directives in that example are only necessary if you want to be able to compile the *same* code with two (or more) different particle modules.

2.3 Modifying individual subroutines or functions

If you want to modify some existing DarkSUSY function or subroutine, *don’t do it*, instead add your own routine as a replaceable function (see also Section 3.6), by running the script `scr/make_replaceable.pl` on the routine you wish to have a user-replaceable version of. You will then find that version in the corresponding `user_replaceables` folder, where you can edit it to your liking. Following these steps, it is guaranteed that the newly created user-replaceable function is properly included in the library where the original DS function used to be, with all makefiles being automagically updated.

An alternative – and often even simpler – way of using user-replaceable functions is to leave the DS libraries untouched, and to instead link to the user-supplied function only when making the main program (this option is indicated in the left-most part of Fig. 3.1). For an explicit example, see (the makefile for) `generic_wimp_oh2.f`.

[†]If you want, you can delete all the blocks for the other (not needed) example programs, in order to have the makefile look clearer and easier to understand.

Chapter 3

Guiding principles of the code

DarkSUSY (as of version 6) has a new structure compared to earlier versions of the code. The most striking difference is that we have split the particle physics model dependent parts from the rest of the code. This means in practice that we have separated DarkSUSY into one set of routines, `ds_core`, which contains no reference to any specific particle model, as well as distinct sets of routines for each implemented model of particle physics. For supersymmetry, for example, all routines that require model-dependent information now reside in the `mssm` module. The advantage with this setup is that `ds_core` and the particle physics modules may be put in separate FORTRAN libraries, which implies that DarkSUSY can now have several particle physics modules side by side. The user then simply decides at the linking stage, i.e. when making the main program, which particle physics module to include.

For this to work, the main library communicates with the particle physics modules via so-called *interface* functions (or subroutines), with pre-defined signatures and functionalities. Note that a particle physics module does not have to provide all of these predefined functions: which of them are required is ultimately determined only when the user links the main program to the (`ds_core` and particle module) libraries. Assume for example that the main program wants to calculate the gamma-ray flux from DM (a functionality provided by `ds_core`). This is only possible if the particle module provides an interface function for the local cosmic ray source function; if it does not, the main program will not compile and a warning is issued that points to the missing interface function. If, on the other hand, interface functions required by direct detection routines would be missing in this example, this would not create any problems at either runtime or the compile stage.

In addition, we have added the concept of *replaceable functions*, which allows users to replace essentially any function in DarkSUSY with a user-supplied version. DarkSUSY ships with dedicated tools to assist you setting up both replaceable functions and new particle physics modules. Fig. 3.1 illustrates these concepts by showing how a typical program would use DarkSUSY; below, we describe each of them in more detail.

3.1 The main DarkSUSY library `ds_core`

As introduced above, the main library is in some sense the heart of the new DarkSUSY, offering all the functionality that a user typically would be interested in without having explicitly to refer to specific characteristics of a given particle physics model (after initialization of such a model). The main library thus contains routines for, e.g., cosmic ray propagation, solar models, capture rates for the Sun/Earth, a Boltzmann solver for the relic density calculation, yield tables from annihilation/decay etc. None of the routines in the main library contains any information about the particle physics module. Instead any information needed is obtained by calling a required function that resides in the particle physics module which is linked to (see below).

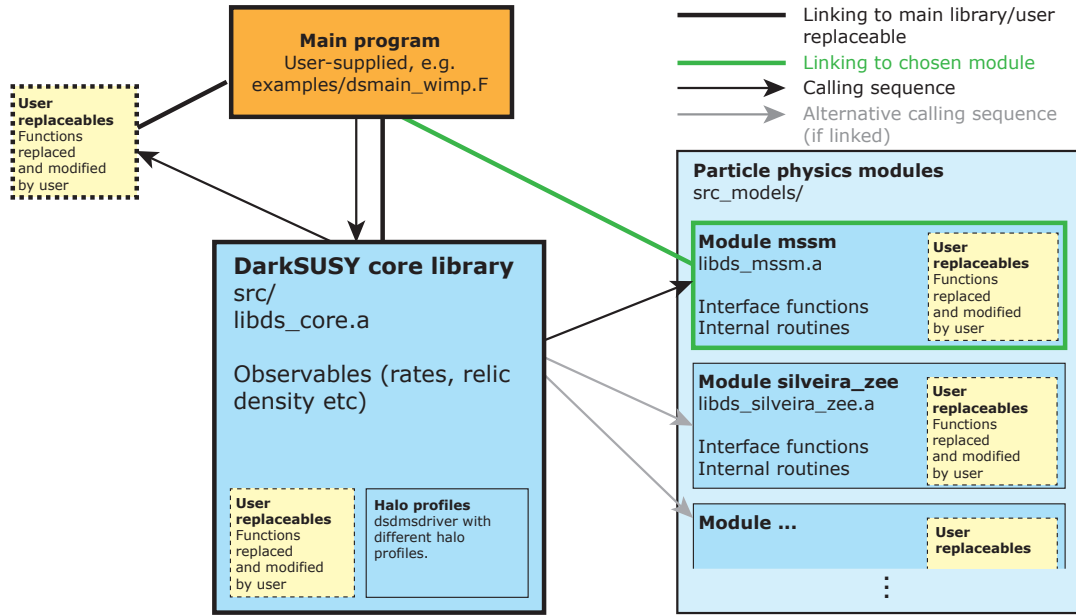


Figure 3.1: Conceptual illustration of how to use DarkSUSY. The main program links to both the main library, `ds_core`, and to *one* of the available particle physics modules. User-replaceable functions are optional and may be linked to directly from the main program, or indirectly by including them in the various libraries. See `examples/dsmain_wimp.F` for an example of a main program that demonstrates typical usage of DarkSUSY for different particle physics modules.

The source code for all functions and subroutines in the main library can be found in the `src/` directory of the DarkSUSY installation folder, with subdirectory names indicating subject areas as summarized in Table 3.1.

3.2 Particle physics modules

The particle physics modules contain the parts of the code that depend on the respective particle physics model. Examples are cross section calculations, yield calculations etc. The routines in the particle physics module have access to all routines in `ds_core`, whereas the reverse is in general not true (with the exception of a very limited set of interface functions that each particle module provides).

DarkSUSY 5 and earlier was primarily used for supersymmetric, and neutralino DM, and those parts of the code now reside in the MSSM module `mssm`. However, many people used DarkSUSY even before for e.g. a generic WIMP setup, which was doable for parts of the code, but not all of it. We now provide a generic WIMP module `generic_wimp` that can be used for these kinds of calculations in a much more general way. In a similar spirit, we also provide a module `generic_decayingDM` for phenomenological studies of decaying DM scenarios. As an example for an actual particle physics model other than supersymmetry, DarkSUSY furthermore includes a module `silveira_zee` which implements the DM candidate originally proposed by Silveira and Zee [4] and which now is often referred to as Scalar Singlet DM [5, 6, 7]. We also include an empty model, `empty`, which is of course not doing any real calculations, but contains (empty versions of) all interface functions that the core library is aware of – which is very useful for debugging and testing purposes. Designing new particle modules is a straight-forward exercise, see below, and it is generally a good idea to start with the most similar module that is already available.

Subdirectory name	Description
an_yield	Annihilation yields in the halo – yields from simulations
aux	General routines
aux_xcernlib	CERN routines needed by DarkSUSY
aux_xdiag	Diagonalization routines
aux_xnswclib	
aux_xquadpack	CMLIB routines needed by DarkSUSY
cr_aux	Cosmic rays – general
cr_axi	Cosmic rays – diffusion routines for axisymmetric distributions
cr_gamma	Cosmic rays – Gamma fluxes
cr_nu	Cosmic rays – Neutrino fluxes
cr_ps	Cosmic rays – point sources
dd	Direct detection
dd_cr dm	Direct detection – cosmic ray upscattered DM
dmd_astro	Astrophysical source functions
dmd_aux	Auxiliary functions for dark matter distribution routines
dmd_mod	Dark matter distributions
dmd_vel	Dark matter velocity distributions
fi	Freeze-In
ini	Initialization routines
kd	Kinetic decoupling
rd	Relic density
se_aux	Auxiliary routines for WIMP annihilation in the Sun/Earth
se_mod	Sun and Earth models
se_nu	Capture and annihilation in the Sun/Earth
se_yield	Yields from annihilation in the Sun/Earth
si	Self-interactions
ucmh	Ultra-compact mini-halos

Table 3.1: Organization of the main library `ds_core`: all functions and subroutines reside in the `src/` folder of the DarkSUSY installation, with the names of the subdirectories indicating the subject area. (Note: this table is automatically generated from the actual directory structure in `src/.`)

A list of the currently available particle physics modules is given in table 3.2, and the source code for the particle modules can be found in the `src_models/` directory of the DarkSUSY installation folder, each of the subdirectories (e.g. `src_models/mssm/`, `src_models/generic_wimp/`) typically reflecting a (sub)subdirectory structure analogous to what is shown in Table 3.1 for the core library. Many dark matter models, furthermore, constitute only relatively simple extensions to the standard model, inheriting most of its structure. For convenience, we therefore also provide various auxiliary routines, in `src_models/common/sm`, that each particle module automatically has access to, and which return basic standard model quantities like, e.g., the masses of standard model particles and their running (so additional BSM effects have to be implemented in the respective particle module).

Particle physics modules		
Module No.	Short name	Description
1	common	Basic principles and common routines
2	empty	The empty model
3	generic_decayingDM	Generic decaying dark matter
4	generic_fimp	Generic FIMPs
5	generic_wimp	Generic WIMPs
6	mssm	The Minimal Supersymmetric Standard Model
7	silveira_zee	Silveira-Zee (Scalar Singlet)
8	vdSIDM	Self-Interacting Dark Matter

Table 3.2: List of particle physics modules currently available in `src_models`. (Note: this table is automatically generated from the actual content of `src_models`.)

3.2.1 Using a particle physics module

How to use a particle physics module obviously depends on how it is implemented, and in principle there are no formal requirements on how this should be done – as long as the provided interface functions are correctly set up, see Section 3.4 further down. The one subroutine that *is* required to exist, however, is

```
dsinit_module
```

This is called directly from `dsinit`, and must be the first call to the particle module as it initializes all general settings and relevant common block variables.

While there is no required structure otherwise, there is a typical workflow associated to using a particle physics module in a main program. First, one needs to initialize a given model by specifying its model parameters. This is done by routines like

```
dsgivemodel...
```

A call to `dsgivemodel_decayingDM`, e.g., allows to enter the defining parameters for a DM model in the `generic_decayingDM` module, while `dsgivemodel_25` allows to enter the parameters for pMSSM model with 25 parameters in the `mssm` module. The next step is then typically a call to

```
dsmodelsetup
```

in order to transfer the model parameters to common blocks and calculate basic quantities like masses and couplings. Once a model is set up like this, a main program can use the full functionality of DarkSUSY supported by the respective module.

3.2.2 Adding a new particle physics module

To create a new particle physics module, the easiest way is to start from an existing one as a template and create a new one from that one. To help you in this process we provide a script `scr/make_module.pl` that takes two arguments, the module you want to start from and the new one you want to create (for further instructions, just call the script without arguments). It will then copy the module to a new one, change its name throughout the module and make sure that it is compiled by the makefiles and included properly when requested by the main programs. If you specify the option `-i` only interface functions will be copied (which creates a cleaner starting point, but also will most likely not compile without modifications). When creating a new module this way, the best is to copy from a module that is as similar as possible to your new model. If

you want a clean setup, you can always copy from the `empty` module. If you want something more phenomenological that has a basic framework for calculating observables, starting from the `generic_wimp` or `generic_decayingDM` might be a good idea. A general advice is to view the modules we provide as a starting point as inspiration for your new modules.

Even though a particle physics module does not need to include all interface functions (which ones are needed only depends on the observables you try to calculate in your main programs), it needs to provide an initialization routine `dsinit_module.f`. This routine should set a global variable `moduletag` to the name of the module so that routines that need to check if the correct module is loaded can do so. When using the script `scr/make_module.pl` this routine is always created and `moduletag` set as it should.

Note that the script `scr/make_module.pl` only provides the framework in the configure script and makefiles (or rather `makefile.in`'s) to make sure your module is compiled. It does *not* create a main program that uses your module, that is up to you. A good starting point for that can either be e.g. `dsmain_wimp.F` in examples that already contain different blocks for different modules (controlled by pre-compiler directives, see the code for more details). Another option is to use any of the example programs in `examples/aux` as a starting point.

To make upgrading to new DarkSUSY versions as smooth as possible, we advice to create your own folder with your main programs, using e.g. the makefile in `examples/aux` as a starting point. Don't modify any of the DarkSUSY makefiles directly as they are overwritten every time you run `configure`, see Section 2.2.2 for more details.

Note that for the `scr/make_module.pl` script to work you need to have `autoconf` installed as the script adds your new module to `configure.ac` and runs `autoconf` to create a new `configure` script.

3.3 Halo models

Several routines in the core library need to know which DM halo should be adopted for the calculations. With this DarkSUSY version, we introduce a new and flexible scheme that avoids pre-defined hardcoded functions to describe the DM density profiles, and allows to consistently define different DM targets at the same time. For convenience, we still provide several pre-defined options for such halo parameterizations, and the user can choose between, e.g., the Einasto [8] and the Navarro-Frenk-White profile [9], or read in any tabulated axi-symmetric (or spherically symmetric) profile. On a *technical* level, the halo models are handled by the `dsdmsdriver` routine which contains a database of which halo profiles the user has set up, and consistently passes this information to all parts in the code where it is needed.*

We provide detailed hands-on examples on how to use this system by a set of example programs `DMhalo_*.f`, see Section 2.2.1. For further details, we refer to Section 17.1 of this manual.

3.4 Interface functions

Interface functions are functions that routines in `ds_core` might need to call, and which therefore every particle physics module should contain if that particular observable is requested. Examples of interface functions include `dsddsigma` that returns the equivalent DM nucleus cross section, `dsresource` that returns the source term for DM-induced cosmic rays (relevant for indirect detection), `dsanxw` that returns the invariant annihilation rate (in the case of WIMPs), etc. All interface functions are found by looking in the empty module, and always contain the keyword 'interface' in the function or subroutine header.

*From a technical point of view, it is actually not `dsdmsdriver` itself which acts as an interface to the rest of the code, but the set of wrapper routines collected in `src/dmd_astro` (which all call `dsdmsdriver` in a specific way). Only those routines are called by cosmic-ray flux routines and other functions in `src`, never `dsdmsdriver` directly. The routines in `src/dmd_astro` therefore provide examples of functions that *cannot be replaced* individually in a consistent way, but only as a whole set (along with `dsdmsdriver`, in case the user wants to change the structure of the driver itself).

If a particle physics module contains the full set of interface functions, all routines in the main DarkSUSY code should work. However, this is not needed to set up a particle physics module. E.g. a generic WIMP model does not need to have decay rates set up. If a routine in the main DarkSUSY routines is called that rely on this interface being present, an error will be thrown when trying to compile your code. This of course applies to all interface functions, not just unphysical ones. E.g. if you are only interested in the relic density for a new particle physics module, you do not need to set up scattering rates, cosmic ray source functions, etc.

In Tab. 3.3, we provide the complete list of interface functions currently implemented in DarkSUSY, along with a brief description. For more details, consult the headers of these files.

3.5 Commonly used functions

Routines that we believe are particularly useful for most users to call, we denote as *commonly used functions*. A list of the commonly used functions in the main DarkSUSY library in `src` is shown in Table 3.4. The routines are described in more detail in their corresponding chapters in part II of this manual, as indicated in the table 3.4.

3.6 Replaceable functions

The concept of replaceable functions introduces the possibility to replace a DarkSUSY routine with one of your own. The way it works is that the user-provided function will be linked when you make your main program instead of the DarkSUSY default one. If, for example, you want to replace the yields from a typical final state of DM annihilation or decay (like $\bar{b}b$) with a new function of your own – e.g. because you are interested in comparing the tabulated Pythia [10] yields with those provided by PPPC [11] – you create your own version of the routine `dsanyield_sim` and let DarkSUSY use this one instead. To help you with this setup, we provide tools that can create (or delete) a replaceable function from any DarkSUSY function and set up the makefiles to use this user-supplied function instead. We also provide a simple way of managing large ‘libraries’ of user-supplied functions, via a list imported by the makefiles, where the user can determine on the fly which user-replaceable functions should be included and which ones should not. Note that both routines in `ds_core` and in any of the particle physics modules can be replaced in this way, including interface functions. There is also a possibility for the particle physics modules to replace a function in the core library. This is not used often, but is used e.g. for the function `dsrdxi.f` (which gives a possibility to have different dark matter and background temperatures in the early Universe).

There is also a simple way to bypass this system – indicated in the left-most part of Fig. 3.1 – by directly linking to the user-supplied function only when making the main program. For an explicit example, see (the makefile for) `generic_wimp_oh2.f`.

Routine	Appears in modules	Used by	Description
dsacbnd	2, 6		Check collider bounds
dsanwx_finiteT	2, 4, 7	fi	Self-annihilation invariant rate at finite temperature
dsanwx	2, 4, 5, 6, 7, 8	fi, rd	Self-annihilation invariant rate
dscrsource_line	2, 3, 5, 6, 7, 8	cr_axi, cr_gamma, cr_nu	Source term for monochromatic contributions from dark matter
dscrsource	2, 3, 5, 6, 7, 8	cr_axi, cr_gamma, cr_nu, ucmh	Source term for dark matter induced cosmic rays
dsddgpgn	2, 5, 6, 7, 8	se_nu	Four fermion couplings.
dsddsigma	1, 2	dd, dd_crdm, se_nu	UNpolarized *equivalent* WIMP nucleus cross section including form factors
dsdmspin	2, 3, 5, 6, 7, 8	se_nu	dark matter spin
dsidnumberset	1		
dsidnumber	1	dd_crdm	
dsinit_module	2, 3, 4, 5, 6, 7, 8	ini	Intialzation of module
dskdm2simp	2, 5, 6, 7, 8	kd	Scattering amplitude squared, expanded in powers of energy
dskdm2	2, 5, 6, 7, 8	kd	Full scattering amplitude squared, averaged over momentum transfer
dskdparticles	2, 5, 6, 7, 8	kd, rd	Intialization of kinetic decoupling for module
dsmodelsetup	2, 3, 4, 5, 6, 7, 8		Sets up a new particle physics model
dsmwimp	2, 3, 4, 5, 6, 7, 8	cr_aux, cr_axi, dd, dd_crdm, fi, kd, rd, se_nu	Dark matter mass
dsrdparticles	2, 4, 5, 6, 7, 8	fi, rd	Particles included in relic density calculation (coannihilations, resonances and thresholds)
dsseyield	2, 5, 6, 7, 8	ini, se_nu, se_yield	Yields from annihilation in the Sun/Earth
dssigmav0tot	2, 5, 6, 7, 8	cr_axi, se_nu	Total annihilation cross section at $v = 0$
dssisigtm	2, 8	si	Self-interaction momentum-transfer cross section

Table 3.3: Table of interface functions, which modules that contain them (with numbering from table 3.2, where they are used and a short description of them. (Note: this table is automatically generated from scanning through the particle physics modules in `src_models`. The description is taken from the empty module routine headers.)

Routine	Description	Chapter
dsanyield_sim	Simulated particle yields	6
dsdbdphidtaxi	Local galactic differential antideuteron flux from dark matter	10
dsepdphidpaxi	Local galactic differential positron flux from dark matter	10
dspbdphidtaxi	Local galactic differential antiproton flux from dark matter	10
dscrgaflux_line_v0ann	Flux of monoenergetic gamma-rays from annihilation in the halo, in the limit of zero relative velocity	11
dscrgaflux_v0ann	Flux of gamma-rays from annihilation in the halo, in the limit of zero relative velocity	11
dsgafluxsph	gamma-rays from decay/annihilation from halo specified by halo label	11
dscrmuflux_v0ann	Neutrino-induced muon flux from WIMP annihilation in the halo	12
dsddrde	Differential WIMP-nucleus recoil rates	14
dsddhelp	help with options for scattering cross sections	14
dsddDMCRcountrate	experimental sensitivity to cosmic-ray upscattered dark matter	15
dsdfactor	D-factor (l.o.s. integral for decaying DM)	??
dsjfactor	J-factor (l.o.s. integral for annihilating DM)	??
dsdmdselect_halomodel	select between halo models in the halo repository	17
dsdmdset_halomodel	add an existing halo model to halo repository	17
dsfi2to2oh2	Calculate the relic density Ωh^2 from freeze-in	19
dsinit	Initialize DarkSUSY (should always be called)	20
dskdmcut	Cutoff mass in linear power spectrum	21
dskdtkd	Kinetic decoupling temperature	21
dsrdomega	Calculate the relic density Ωh^2 of a dark matter particle	22
dsrdomega_aDM	Calculate the relic density Ωh^2 of an asymmetric	22
dsrdomega_cBE	Calculate the relic density Ωh^2 of a dark matter particle, using coupled Boltzmann equations	22
dssenu_rates	Rates of neutrinos, neutrino-induced leptons and hadronic showers from DM annihilations in the Sun/Earth	25
dssisigtmav	velocity-averaged momentum-transfer cross section for dark matter self-interactions	27

Table 3.4: List of commonly used functions and subroutines. (Note: this table is automatically generated from the routines in src that have the commonly used tag in their header.)

Chapter 4

Comparison to previous DarkSUSY versions

For those DarkSUSY users who are familiar with previous version of the code, the main structural difference introduced with DarkSUSY 6.0 is a highly modular setup that allows to fully disentangle the astrophysics-related parts from those that rely on a specific particle physics model (see Chapter 3 for a detailed description). There are also many new or significantly improved physics capabilities introduced after version 5 and 4 [1] – including electroweak and strong corrections to DM annihilation, improved routines to solve the Boltzmann equation for chemical and kinetic decoupling, and a new framework to handle the propagation of cosmic rays. For a detailed list of those new features, we refer to the publication describing this release of the code [2].

One technical aspect that has changed are the particle codes (k -variables), which are now treated as (module-)internal codes. They can be used by the particle physics module if the module so wishes, but the interface functions and routines in `ds_core` instead use PDG codes when referring to particles.

In the course of re-organizing the code, it also became necessary to re-name some of the basic functions and subroutines that existed in earlier versions and which users may have become familiar with. For convenience, we therefore list below, in Table 4.1, the most commonly used functionalities in version 4 and 5 that have changed in name or usage with the present version of the code.

routine name up to DarkSUSY 5	new routine name	comments
<code>dshmset</code>	<code>dsdmdset_halomodel</code>	Setting up and referring to DM halos has fundamentally changed. See Section 3.3 for an overview and all Chapters with names containing <code>src_dmd</code> in Part II of the manual for more details.
<code>dshmj</code>	<code>dsjfactor</code>	Line-of-sight integrals now take a halo label as input, and can be computed for various objects at the same time.
<code>dssusy</code>	<code>[dsmodelsetup]</code>	Setting up a model (calculating the mass spectrum, relevant 3-particle vertices etc.) now depends on the particle module implementation.

dsmhtkd dsmhmcut	dskdtkd dskdmcut	The 'microhalo' routines are now more properly referred to as 'kinetic decoupling' routines.
dshmrescale_rho	—	A mismatch between local halo density and DM abundance for a given DarkSUSY module is no longer hidden as a rescaling factor in a common block. Instead, such factors typically enter as explicit parameters in direct and indirect detection routines.
dshaloyield	[dsanyield]	total cosmic ray yield from <i>neutralino</i> annihilation
dshayield	dsanyield_sim	simulated cosmic ray yields from individual annihilation/decay channels to SM particles. Note that internal channel codes are now replaced with PDG codes of the final state particles as input.
dshrgacontdiff	dsgafluxsph	Gamma-ray flux routines now work seamlessly together with both the new halo setup and the modular particle physics structure.
dshrgaline	dscrgaflux_line_v0ann dscrgaflux_dec	Gamma-ray line routines now return both number, energies and widths of all such signals that are present in the current particle setup.
dshrbardiff dshrdbardiff dsepdiff	dspbdphidtaxi dsdbdphidtaxi dsepdphidpaxi	Cosmic-ray propagation routines have been re-written from scratch. They are now much more flexible and can be used for any axisymmetric halo/diffusion model.
dsntrates	dssenu_rates	Neutrino rates from inside the sun or earth
dshrmuhalo	dscrmuflux_v0ann dscrmuflux_v0ann	Neutrino-induced muon flux from the halo (for annihilating and decaying DM, respectively).

Table 4.1: ‘Translation table’ for how the most commonly used functionalities in DarkSUSY version 5 and earlier have changed with the present version of the code. Routines in parentheses, e.g. [dsanyield], are no longer part of the DarkSUSY main library and only provided by specific particle physics modules. For more detailed descriptions of the new routines and functions, see the headers of the respective files.

Chapter 5

Original articles

A large part of the routines in the present DarkSUSY version have been implemented in the context of original research work. Therefore, when using those routines, please give proper credit not only to the main DarkSUSY paper [1, 2] but also to the relevant articles in the following list:

- Relic density – freeze-out** P. Gondolo and G. Gelmini, Nucl. Phys. **B360** (1991) 145 [12]; J. Edsjö and P. Gondolo, Phys. Rev. **D56** (1997) 1879 [13]; J. Edsjö, M. Schelke, P. Ullio and P. Gondolo, JCAP **04** (2003) 001 [14]. For coupled Boltzmann equations: T. Binder, T. Bringmann, M. Gustafsson and A. Hryczuk, Eur. Phys. J. **C 81** (2021) 577 [15].
- Relic density – freeze-in** T. Bringmann, S. Heeba, F. Kahlhoefer and K. Vangsnes, arXiv: 2111.14871 (2021) [16].
- Kinetic decoupling and microhalos** T. Bringmann, NJP **11** (2009) 10527 [17].
- Continuous gamma-rays** L. Bergström, J. Edsjö and P. Ullio, Phys. Rev. **D58** (1998) 083507 [18].
- Neutrino telescopes** L. Bergström, J. Edsjö and P. Gondolo, Phys. Rev. **D58** (1998) 103519 [19].
- Positrons** E.A. Baltz and J. Edsjö, Phys. Rev. **D59** (1999) 023511 [20].
- Antiprotons** L. Bergström, J. Edsjö and P. Ullio, ApJ **526** (1999) 215 [21].
- QCD corrections to DM annihilation** T. Bringmann, A. J. Galea, P. Walia, Phys.Rev. **D93** (2016) 043529 [22].
- Cosmic-ray upscattered dark matter** T. Bringmann and M. Pospelov, Phys. Rev. Lett. **122** (2019) 17, 171801 [23]; K. Bondarenko *et al.*, JHEP **03** (2020) 118 [24].
- General MSSM, direct detection** L. Bergström and P. Gondolo, Astrop. Phys. **5** (1996) 263 [25].
- Gamma lines from supersymmetric DM** L. Bergström and P. Ullio, Nucl. Phys. **B504** (1997) 27 [26]; P. Ullio and L. Bergström, Phys. Rev. **D57** (1998) 1962 [27].
- Internal bremsstrahlung in the MSSM (γ and e^+)** T. Bringmann, L. Bergström and J. Edsjö, JHEP **0801** (2008) 049 [28]; L. Bergström, T. Bringmann and J. Edsjö, Phys.Rev. **D78** (2008) 103520 [29].
- MSSM Electroweak corrections to indirect detection yields** T. Bringmann and F. Calore, Phys.Rev.Lett. **112** (2014) 071301 [30]; T. Bringmann, F. Calore, A. J. Galea and M. Garny, JHEP **1709** (2016) 041 [31].

Part II

Main DarkSUSY routines in src/

Chapter 6

an_yield: Annihilation yields in the halo – yields from simulations

6.1 Annihilation in the halo, yields – theory

Here we calculate yields of different particles from annihilation of dark matter particles in the halo.

6.1.1 Monte Carlo simulations

We need to evaluate the yield of different particles per WIMP annihilation. The hadronization and/or decay of the annihilation products are simulated with PYTHIA [10] 6.426. The simulations are done for a set of 18 WIMP masses, $m_\chi = 10, 25, 50, 80.3, 91.2, 100, 150, 176, 200, 250, 350, 500, 750, 1000, 1500, 2000, 3000$ and 5000 GeV. We tabulate the yields and then interpolate these tables in DarkSUSY.

The simulations are here simpler than those for annihilation in the Sun/Earth since we don't have a surrounding medium that can stop the annihilation products. We here simulate for 8 'fundamental' annihilation channels $c\bar{c}$, $b\bar{b}$, $t\bar{t}$, $\tau^+\tau^-$, W^+W^- , Z^0Z^0 , gg and $\mu^+\mu^-$. Compared to the simulations in the Earth and the Sun, we now let pions and kaons decay and we also let antineutrons decay to antiprotons. For each mass we simulate 2.5×10^6 annihilations and tabulate the yield of antiprotons, positrons, gamma rays (not the gamma lines), muon neutrinos and neutrino-to-muon conversion rates and the neutrino-induced muon yield, where in the last two cases the neutrino-nucleon interactions has been simulated with PYTHIA as outlined in section 26.1.1

With these simulations, we can calculate the yield for any of these particles for a given particle physics model. In `src`, we only include the channels that are actually simulated. In the different particle physics modules in `src_model`, the summation over all possible final states and possibly more complex channels, like scalars decaying to other particles is done.

Note that simulations are typically done without specifying a particular polarization state of the final state particles. This is however not entirely correct as the possible polarization states will depend on the particle physics model we have.

Even if the simulations are not performed in this more general way yet, we have set up the structure here to eventually provide the yields in this form. In particular the most general routine to calculate the yields from any of the simulated channels is `dsanyield_sim_ls`. Apart from the mass, energy and yield type, this routine also takes as arguments the PDG codes of the final state particles and the polarization state of the final state. In particular, you are required to provide the

quantum numbers

- j = the quantum number for the total angular momentum
- P = the parity of the final state
- l = the quantum number for the orbital angular momentum in the final state
- s = the quantum number for the total spin in the final state

We want to move in a direction where this routine is the one that should be used. While getting there, we also provide a simpler routine which just gives the polarization state in terms of helicity and polarization, **dsanyield_sim**. This routine takes the PDG code of the final state and the polarization as

- Left-handed or right-handed polarization for fermions.
- Longitudinal or transverse polarization for vector bosons.

This routine for simplicity assumes that the final state particles have the same polarization state.

Chapter 7

aux:

General routines

7.1 General routines

In **aux/**, we collect routines that are of general interest to many other routines in DarkSUSY. E.g., we have routines to find elements in arrays (used for interpolation), Bessel functions, error functions, spline routines, etc.

Chapter 8

aux_xdiag: Diagonalization routines

8.1 XDIAG

This folder contains a set of routines to perform numerical diagonalization of complex matrices. Even though analytical routines can be used for up to 5×5 matrices, they tend to be numerically unstable for matrices occurring in some particle physics models (for example supersymmetry). Hence, we use numerical routines to perform the diagonalization instead.

Chapter 9

cr_ aux: Cosmic rays – general

9.1 Cosmic Rays – auxiliary routines

This folder contains auxiliary functions needed by the cosmic ray routines. For example, it contains different versions of simple integration routines and a set of routines to handle the correct setting and interpreting of labels for the tabulation of confinement times and Green’s functions needed for the calculation of cosmic ray fluxes.

Physics-wise, we include here also the interstellar intensity of the dominant cosmic ray species, dI/dR , where R is the particle’s rigidity, and the corresponding fluxes, $d\Phi/dT = 4\pi(dR/dT)(dI/dR)$. These are needed by the routines for cosmic-ray upscattering – see Section 14.2 – and returned by the functions `dscrSRintensity` and `dscrSRflux`.

We finally provide for convenience two rescaling factors that can be applied to all (charged and neutral) cosmic ray fluxes that result from the annihilation of DM particles in the halo, in situations where the DM candidate in question (as initialized with the standard sequence of `dsgive_model_XXX` and `dsmodelsetup`) only provides some fraction of the total DM density in the halo. For symmetric DM, this simply amounts to a rescaling by the DM density squared, as returned by the function `dscrrescale_sym`. If the DM density has both a symmetric and an asymmetric component, on the other hand, one should rescale by the factor returned by `dscrrescale_asym`.

Chapter 10

cr_axi:

Cosmic rays – diffusion routines for axisymmetric distributions

10.1 Cosmic ray propagation in axially symmetric halos

There is a clean asymmetry between particles and antiparticles in the standard cosmic ray picture: The bulk of cosmic rays – protons, nuclei and electrons – are mainly “primary” species, i.e. particles accelerated in astrophysical sources and then copiously injected in the interstellar medium; “secondary” components, including antimatter, are instead produced in the interaction of primaries with the interstellar medium during the propagation. It follows that there is a pronounced deficit of antimatter compared to matter in the locally measured cosmic ray flux (about 1 antiproton in 10^4 protons). When considering instead a source term due to DM annihilations or decays, a significant particle-antiparticle asymmetry is in general not expected, and antiprotons, positrons and antideuterons turn out to be competitive indirect DM probes.

Charged particles propagate diffusively through the regular and turbulent components of Galactic magnetic fields. This makes it more involved for local measurements to track spectral and morphological imprints of DM sources than, e.g., for the gamma-ray and neutrino channels (though searches for spectral features in CR positron fluxes still lead to very competitive limits [32]). In fact the transport of cosmic rays in the Galaxy is still a debated subject: Most often one refers to the quasi-linear theory picture (with magnetic inhomogeneities as a perturbation compared to regular field lines) in which propagation can be described in terms of a (set of) equation(s) linear in the density of a given species, containing terms describing diffusion in real space, diffusion in momentum space (the so-called reacceleration), convective effects due to Galactic winds, energy or fragmentation losses and primary and secondary sources (see, e.g., Ref. [33] for a review). Dedicated codes have been developed to solve numerically this transport equation, including GALPROP [34], DRAGON [35] and PICARD [36].

Here we follow instead a semi-analytical approach, analogous to that developed for the USINE code [37]. In particular, we model the propagation of antiprotons and antideuterons by considering the steady state equation [21]

$$\frac{\partial N}{\partial t} = 0 = \nabla \cdot (D \nabla N) - \nabla \cdot (\vec{u} N) - \frac{N}{\tau_N} + Q. \quad (10.1)$$

We solve it for situations where *i*) the diffusion coefficient D can have an arbitrary dependence on the particle rigidity but can at most take two different values in the Galactic disc and in the diffusive halo, *ii*) the convective velocity \vec{u} has a given fixed modulus and is oriented outwards

and perpendicular to the disc, *iii*) the loss term due to inelastic collisions has an interaction time τ_N which is energy dependent but spatially constant in the disc and going to infinity in the halo (corresponding to a constant target gas density in the disc and no gas in the halo), *iv*) the DM source Q is spatially axisymmetric and has a generic energy dependence. Under these approximations and assuming, as is usually done, that the propagation volume is a cylinder centred at the disc and that particles can freely escape at the boundaries of the diffusion region, Eq. (10.1) can be solved analytically by expanding N in a Fourier-Bessel series; the computation of the flux involves, at each energy, a sum over the series of zeros of a Bessel function of first kind and order zero, and a volume integral of the spatially dependent part in the axisymmetric source term Q (basically the DM density ρ_χ for decaying DM and its square for pair annihilating DM) times a weight function depending on the given zero in the series (see [21] for further details).

Since the path lengths for antiprotons and antideuterons are rather large, of the order of a few kpc, taking average values for parameters in the transport equation rather than the more realistic modelling that can be implemented in full numerical solutions, has no large impact in case of extended and rather smooth sources such as for DM. Eq. (10.1) neglects reacceleration effects, which may in general be relevant at low energies (rigidities below a few GV); however even this does not have a large impact in case of the species at hand, see, e.g., [38] for a comparison of results with numerical and semi-analytical solutions for cosmic-ray antiprotons. The power of our semi-analytic approach is that one can store values of the solution of the transport equation obtained by assuming a given mono-energetic source – provided by the functions `dspbtdaxi` and `dsdbtdaxi` for, respectively, antiprotons and antideuterons – and then apply these as weights to any particle source term $\mathcal{S}_n(E_f)$ as introduced above. For antiprotons and antideuterons, this latter step is done in the functions that compute the local galactic differential fluxes from DM annihilation and decay, `dspbdbphidtaxi` and `dsdbdbphidtaxi`, respectively. Note that while the outputs of `dspbtdaxi` and `dsdbtdaxi` are labelled “confinement time” in the code, since they do have a dimension of time and scale the dependence between source and flux, one cannot trade them for what is usually intended as confinement time for standard cosmic ray components, given that the morphology of the DM source is totally different from supernova remnant distributions usually implemented for describing ordinary primary components.

The structure we implemented gives a particularly clear advantage when the code is used to scan over many particle physics DM models, but only over a limited number of propagation parameters and DM density profiles. For such an application, it is useful to tabulate the ‘confinement times’ (returned by `dspbtdaxi` and `dsdbtdaxi`) over a predefined range of energy; this is done in the functions `dspbtdaxitab` and `dsdbtdaxitab` when calling the flux routines with an appropriate option (and only in case the DM halo profile currently active is within the halo profile database). Such tabulations can be saved and re-loaded for later use; here the proper table association is ensured by a propagation parameter label setting system in analogy to the one implemented for the halo profile database. Computing such a table on the first call is rather CPU consuming, especially for DM profiles that are singular towards the Galactic center, so in case only a few flux computations are needed it may be better to switch off the tabulation option; this is true also in case the flux is needed at a small number of energy values, since the latest 100 (non-equivalent) calls to `dspbtdaxi` and `dsdbtdaxi` are stored in memory (with the corresponding propagation parameters and halo model correctly associated).

The case for positrons is treated analogously, except that energy losses and spatial diffusion are the most important effects for propagating cosmic ray leptons. The transport equation we solve semi-analytically therefore has the form [20]

$$\frac{\partial N}{\partial t} = 0 = \nabla \cdot (D \nabla N) + \frac{\partial}{\partial p} \left(\frac{dp}{dt} N \right) + Q, \quad (10.2)$$

where the functional form of D and Q can be chosen as for antiprotons and antideuterons, and the energy loss rate dp/dt can have a generic momentum dependence but needs again to be spatially constant. Assuming the same topology for the propagation volume and free escape conditions at

the vertical boundaries (to compute the local positron flux the radial boundary turns out to be irrelevant) , the solution of the propagation equation is given in terms of a Green’s function in energy (the function `dsepgreenaxi` in the code) to be convoluted over the source energy spectrum at emission for a given particle DM candidate. This last step is performed by the function `dsepdndpaxi` returning the local positron number density, while `dsepdphidpaxi` converts it to a flux and is the function which should be called from the main file. The method to implement this solution is a slight generalization of the one described [20] and generalizes the one introduced in [39] for a spherically symmetric configuration to an axisymmetric system.

The computation of the Green’s function involves a volume integral over the spatially dependent part of the DM source function Q (again basically the DM density ρ_χ for decaying DM and its square for pair annihilating DM) and the implementation via the so-called method of image charges (again a sum over a series) of the free escape boundary condition. It can again be CPU expensive for singular halo profiles, but its tabulation is always needed since the Green function appears in a convolution. The main limiting factor with respect to full numerical solutions is that one is forced to assume an (spatially) average value for dp/dt . However this has not a severe impact on our results for the local DM-induced positron flux since, especially for energies above 10 GeV, the bulk of the DM contribution to the local flux stems from a rather close-by emission volume; one thus just has to make sure to normalize dp/dt to the mean value for *local* energy losses, as opposed to the mean value in the Galaxy, which are mainly due to the synchrotron and inverse Compton processes.

While the transport equations (10.1) and (10.2) are essentially the same as considered in previous releases of the code, their implementation in the present release is completely new and appears to be numerically more stable. In particular cases with very singular DM profiles still give numerically accurate results and converge faster (in case of antiprotons and antideuterons implementing a procedure which applies to point sources). Note however that the case of very singular DM profiles is also the one in which our models or *any* propagation model is subject to a significant uncertainty related to the underlying physics, since propagation in the Galactic center region is difficult to model and probably rather different from what can be tested in the local neighbourhood by measurements of primary and secondary cosmic rays. Finally, the implementation starting from DarkSUSY 6 is more flexible regarding parameter choices, such as for the rigidity scaling of the diffusion coefficient and energy scaling of energy losses, in a framework which is now fully consistent for antiprotons, antideuterons and positrons.

Chapter 11

`cr_gamma:`

Cosmic rays – Gamma fluxes

11.1 Gamma rays from the halo – theory

Among the yields of decay or pair annihilations of halo dark matter particles, the role played by gamma-rays could be a major one. Unlike the cases involving charged particles, for gamma-rays it is straightforward to relate the distribution of sources and the expected flux at the earth. Most flux estimated can be obtained just by summing over the contributions along lines of sight (or better, geodesics): gamma-rays have a low enough cross section on gas and dust and therefore the Galaxy is essentially transparent to them (except perhaps in the innermost part, very close to the region where a massive black hole is inferred); absorption by starlight and infrared background becomes efficient only for very far away sources (redshift larger than about 1).

It follows that in case the gamma-ray signal is detectable, this might be the only chance for mapping the fine structure of a dark halo, with a much better resolution for inhomogeneities (clumps) with respect to what is achievable through dynamical measurements or lensing effects. This is especially true for annihilating dark matter. Turning the latter argument around, if the fine structure of the Galactic halo is clumpy, or if a large density enhancement is present towards the Galactic center, as seen in N-body simulations of dark matter halos, this dark matter detection technique may be more promising than indicated by the estimates in which smooth non-singular halo scenarios are considered (recall that the fluxes per unit volume are proportional to the square of the dark matter density locally in space).

Several targets have been discussed as sources of gamma-rays from the annihilation of dark matter particles. An obvious source is the dark halo of our own galaxy [40, 41, 42, 43] and in particular the Galactic center, as the dark matter density profile is expected, in most models, to be peaked towards it, possibly with huge enhancements close to the central black hole. The Galactic center is an ideal target for both ground- and space-based gamma-ray telescopes. As satellite experiments have much wider field of views and will provide a full sky coverage, they can in principle test the hypothesis of gamma-rays emitted in clumps of dark matter which may be present in the halo [44, 45, 46, 18, 47, 48]. Another possibility which has been considered is the case of gamma-ray fluxes from external nearby galaxies [49]. Furthermore, it has been proposed to search for an extragalactic flux originated by all cosmological annihilations of dark matter particles [50, 51, 52]. DarkSUSY is suitable to compute the gamma-ray flux from all these (and possibly other) sources.

11.2 Continuous gamma yields and line signals

The bulk of the gamma-ray yield from DM annihilation or decay arises from quark jets that when they hadronize/decay give rise to neutral pions which decay to gamma rays. At loop level, it could also be possible to produce monochromatic gamma rays, which could be a smoking gun for dark matter searches. The advantage with the gamma-ray lines is the distinctive spectral signature, which has no plausible astrophysical counterpart.

Compared to the monochromatic flux, the gamma-ray flux produced in π^0 decays is much larger but has less distinctive features. The photon spectrum in the process of a pion decaying into 2γ is, independent of the pion energy, peaked at half of the π^0 mass, about 70 MeV, and symmetric with respect to this peak if plotted in logarithmic variables. Of course, this is true both for pions produced by DM and, e.g., for those generated by cosmic ray protons interacting with the interstellar medium.

When considered together with to the cosmic-ray induced Galactic gamma-ray background, the DM induced signal looks like a component analogous to the secondary flux due to nucleon nucleon interactions: it is drowned into the Bremsstrahlung component at low energy, while it may be the dominant contribution at energies above 1 GeV or so. In fact, if the exotic component is indeed significant an option to disentangle it would be to search for a break in the energy spectrum at about the WIMP mass, where the line feature might be present as well: while the maximal energy for a photon emitted in WIMP pair annihilations is equal to the WIMP mass, the component from cosmic ray protons extends to much higher energies, essentially with the same spectral index as for the proton spectrum (the role played by the third main background component, inverse Compton emission, has still to be settled at the time being and may worsen the problem of discrimination against background).

Besides this (weak) spectral feature, another way to disentangle the dark matter signal may be to exploit a directional signature: data with a wide angular coverage should be analyzed to search for a gamma-ray flux component following the shape and density profile of the dark halo, including eventual contributions from clumps.

11.3 Fluxes

Given a density distribution of DM particles, we can define a source function that tells us how many particles that are produced per volume element per time unit (and per energy interval for differential yields) from either annihilation or decay. The DM signal ultimately only depends on the local injection rate of some stable (cosmic ray) particle f , per volume and energy,

$$\frac{dQ(E_f, \mathbf{x})}{dE_f} = \sum_n \rho_\chi^n(\mathbf{x}) \langle \mathcal{S}_n(E_f) \rangle . \quad (11.1)$$

Here, ρ_χ is the DM density (of the respective component, in case of multi-component DM) and the ensemble average $\langle \dots \rangle$ is taken over the DM velocities; in principle, it depends on the spatial location \mathbf{x} , but in many applications of interest this can be neglected.

The particle source terms $\mathcal{S}_n(E_f)$ encode the full information about the DM particle physics model. For a typical WIMP DM candidate, e.g., only the annihilation part ($n = 2$) contributes,

$$\mathcal{S}_2(E_f) = \frac{1}{N_\chi m_\chi^2} \sum_i \sigma_i v \frac{dN_i}{dE_f} , \quad (11.2)$$

where σ_i is the annihilation cross section of two DM particles into final state i and dN_i/dE_f is the resulting number of stable particles of type f per such annihilation and unit energy. N_χ is a symmetry factor that depends on the nature of DM; if DM is (not) its own antiparticle we have $N_\chi = 2$ ($N_\chi = 4$). More generally, taking into account that there may be an asymmetry between

DM and anti-DM, $n_+ - n_- > 0$ and that hence only the symmetric fraction $r = 2n_-/(n_+ + n_-)$ can annihilate, this factor is given by

$$N_\chi = 2 \frac{n_+ n_-}{(n_+ + n_-)^2} = \frac{4}{r(2-r)}. \quad (11.3)$$

The right-hand side of Eq. (11.2) must be further integrated over $f(v)$, the velocity distribution of the *relative* velocity of the two dark matter particles, but in practice it is typically sufficient to evaluate it for $v = 0$.

For decaying DM, on the other hand, we have

$$\mathcal{S}_1(E_f) = \frac{1}{m_\chi} \sum_i \Gamma_i \frac{dN_i}{dE_f}, \quad (11.4)$$

where Γ_i denotes the partial decay widths. Let us stress, however, that Eq. (11.1) is much more general in that it encapsulates also DM that is *both* annihilating and decaying, multi-component SM, as well as DM models with an internal Z^n symmetry [53, 54, 55, 56].

For a telescope pointing in the direction ψ , the expected DM-induced differential flux in gamma rays or neutrinos – i.e. the expected number of particles per unit area, time and energy – from a sky-region $\Delta\psi$ is thus given by a line-of-sight integral

$$\frac{d\Phi}{dE} = \frac{1}{4\pi} \int_{\Delta\psi} d\Omega \int_{\text{l.o.s.}} d\ell \frac{d\mathcal{Q}}{dE}. \quad (11.5)$$

For an effectively point-like source at distance d , this line-of-sight integral simplifies to

$$\int_{\Delta\psi} d\Omega \int_{\text{l.o.s.}} d\ell \frac{d\mathcal{Q}}{dE} \longrightarrow \frac{1}{d^2} \int dV \frac{d\mathcal{Q}}{dE}, \quad (11.6)$$

where the integral is over the extension of the source (much smaller than d).

For decaying DM, the above line-of-sight integral always factorizes into the particle source term S_1 given in Eq. (11.4) and a term that only depends on the DM distribution,

$$\frac{d\Phi^{\text{dec}}}{dE d\Omega} = \frac{1}{4\pi} J^{\text{dec}} S_1, \quad J^{\text{dec}} \equiv \int_{\text{l.o.s.}} d\ell \rho \quad (11.7)$$

For annihilating DM, the corresponding factorization strictly speaking *only* holds *if* the annihilation rate is independent of velocity:

$$\frac{d\Phi^{\text{ann}}}{dE d\Omega} = \frac{1}{4\pi} J^{\text{ann}} S_2, \quad J^{\text{ann}} \equiv \int_{\text{l.o.s.}} d\ell \rho^2. \quad (11.8)$$

While notable exceptions exist (in particular for resonances [57], p -wave annihilation [58] and Sommerfeld-enhanced annihilation [59]), this is a commonly encountered situation and hence of general interest.

11.4 Gamma rays from the halo – routines

DarkSUSY provides the functions `dscrgaflux_dec` and `dscrgaflux_v0ann` that take J^{dec} (or J^{ann}) as input and return the fluxes given in Eqs. (11.8) and (11.7). Here, the subscript `_v0ann` refers to the fact that, for the purpose of those routines, S_2 is evaluated in the limit of vanishing relative velocity of the annihilating DM pair. While this is the only situation of practical interest in many DM models, future DarkSUSY versions will offer support for a full velocity dependence of this quantity. In analogy with `dscrsource_line`, the core library furthermore provides routines `dscrgaflux_line_dec`

and `dscrgaflux_line_v0ann` (and correspondingly for neutrinos) that return strength, width and location of *monochromatic* ('line') signals. The latter is a convenient change with respect to previous versions of the code, as one cannot generally know how many lines there are (this depends on the particle physics module).

If a halo label is available, one can also more conveniently call `dsgafluxsph` instead. This function directly returns the gamma-ray flux, for that halo, automatically calculating the required line-of-sight integrals and adding decaying and annihilating DM components depending on which particle model is initialized.

Chapter 12

cr_nu:

Cosmic rays – Neutrino fluxes

12.1 Neutrino fluxes from the halo – theory

Usually, the flux of neutrinos from annihilation of WIMPs in the Milky Way halo is too small to be detectable, but for some clumpy or cuspy models, it might be detectable. The calculation of the neutrino-flux follows closely the calculation of the continuous gamma ray flux, with the main addition that neutrino interactions close to the detector are also included. Hence, both the neutrino flux and the neutrino-induced muon flux can be obtained. The neutrino to muon conversion rate in the Earth can also be obtained.

Chapter 13

`cr_ps:`

Cosmic rays – point sources

13.1 Cosmic Ray propagation for point sources

In this directory, we collect the relevant cosmic ray routines for DM point sources like ordinary DM clumps or mini-spikes around intermediate-mass black holes.

Chapter 14

dd: Direct detection

14.1 Direct detection – theory

Specific choices of nuclear structure functions can be selected by calling `dsdd_set('sf', label)`, where the character variable `label` indicates the set of structure functions. For the default option ('best'), e.g., the code automatically picks the best currently available structure function (depending on the nucleus). This mean, in order Fourier-Bessel, Sum-of-Gaussians, Fermi, Lewin-Smith. The function returning the value of $\tilde{\sigma}_{\chi T}$ is to be provided by an interface function `dsddsigma(v,Er,A,Z,sigij, ierr)` residing in the particle physics module, where on input $v=v$, $Er=E_R$, $A=A$, $Z=Z$ and on output the 27×27 array `sigij` contains the (partial) equivalent cross sections $\tilde{\sigma}_{ij}$ in cm^2 and the integer `ierr` contains a possible error code. The order of the entries in `sigij` corresponds to that of the independent nonrelativistic operators \mathcal{O}_i ; for the first 11 entries, we use the same operators and convention as in Ref. [60], while for the last 16 entries we add the additional operators discussed in Ref. [61]. In particular, `sigij(1,1)` is the usual spin-independent cross section and `sigij(4,4)` is the usual spin-dependent cross section. In addition, the direct detection module in `DarkSUSY` provides utility functions that can be used in the computation of the cross section. For example, the subroutine `dsddgg2sigma(v, er,A,Z,gg,sigij,ierr)` computes the (partial) equivalent scattering cross sections $\tilde{\sigma}_{ij}$ for nucleus (A, Z) at relative velocity v and recoil energy E_R starting from values of the G_i^N constants in `gg`, with nuclear structure functions set by the previous call to `dsdd_set`. The actual nuclear recoil event rate as given in

$$\frac{dR}{dE_R} = \sum_T c_T \frac{\rho_\chi^0}{m_T m_\chi} \int_{v>v_{\min}} \frac{d\sigma_{\chi T}}{dE_R} \frac{f(\mathbf{v}, t)}{v} d^3v, \quad (14.1)$$

finally, is computed by the function `dsddrde`. The latter two functions are independent of the specific particle physics implementation and hence are contained in the core library. In the above expression, the sum runs over nuclear species in the detector, c_T being the detector mass fraction in nuclear species i . m_T is the nuclear (target) mass, and $\mu_{\chi T} = m_\chi m_T / (m_\chi + m_T)$ is the reduced DM–nucleus mass. Furthermore, ρ_χ^0 is the local DM density, \mathbf{v} the DM velocity relative to the detector, $v = |\mathbf{v}|$, and $f(\mathbf{v}, t)$ is the (3D) DM velocity distribution. In order to impart a recoil energy E_R to the nucleus, the DM particle needs a minimal speed of $v_{\min} = \sqrt{M_T E_R / 2\mu_{\chi T}^2}$.

14.2 Cosmic-ray upscattered dark matter

Chapter 15

dd_crdm: Direct detection – cosmic ray upscattered DM

15.1 Upscattering of dark matter by cosmic rays

DM particles move at non-relativistic velocities in the Galaxy, implying that the kinetic energy of DM particles with sub-GeV masses is not sufficient to trigger nuclear recoil energies in the keV range, as needed by classical direct detection facilities. However, as long as the elastic scattering cross section between DM and nuclei does not vanish, there must also be an irreducible local flux component of (semi-)relativistic DM particles, resulting from high-energy cosmic-ray (CR) particles scattering on DM particles initially (almost) at rest [23]. As a result, even very light DM is accessible to direct detection (and neutrino) experiments.

Here we briefly review how CRDM is produced, attenuated in the overburden and produces recoil rates in direct detection or neutrino experiments. The implementation of the corresponding routines in DarkSUSY is based on Refs. [23, 24, 62] and explained in Sec. 15.2.

Production

We consider CR nuclei N , with a flux of $d\Phi_N/dT_N$, that elastically scatter on non-relativistic DM particles χ in the Galaxy. For DM with a mass m_χ and density profile $\rho_\chi(\mathbf{r})$, at Galactic position \mathbf{r} , this induces a relativistic CRDM flux incident on Earth (at the top of the atmosphere) of

$$\frac{d\Phi_\chi}{dT_\chi} = \int \frac{d\Omega}{4\pi} \int_{\text{l.o.s.}} dl \frac{\rho_\chi}{m_\chi} \sum_N \int_{T_N^{\min}}^{\infty} dT_N \frac{d\sigma_{\chi N}}{dT_\chi} \frac{d\Phi_N}{dT_N} \quad (15.1)$$

$$\equiv D_{\text{eff}} \frac{\rho_\chi^{\text{local}}}{m_\chi} \sum_N \int_{T_N^{\min}}^{\infty} dT_N \frac{d\sigma_{\chi N}}{dT_\chi} \frac{d\Phi_N^{\text{LIS}}}{dT_N}. \quad (15.2)$$

Here $d\sigma_{\chi N}/dT_\chi$ is the differential elastic scattering cross section for CR nuclei scattering on DM at rest, and T_χ is the resulting kinetic recoil energy. The minimal CR energy T_N^{\min} for this to happen is given by

$$T_N^{\min} = \begin{cases} \left(\frac{T_\chi}{2} - m_N \right) \left[1 - \sqrt{1 + \frac{2T_\chi}{m_\chi} \frac{(m_N + m_\chi)^2}{(2m_N - T_\chi)^2}} \right] & \text{for } T_\chi < 2m_N \\ \sqrt{\frac{m_N}{m_\chi}} (m_N + m_\chi) & \text{for } T_\chi = 2m_N \\ \left(\frac{T_\chi}{2} - m_N \right) \left[1 + \sqrt{1 + \frac{2T_\chi}{m_\chi} \frac{(m_N + m_\chi)^2}{(2m_N - T_\chi)^2}} \right] & \text{for } T_\chi > 2m_N \end{cases} . \quad (15.3)$$

Finally, introducing the effective distance D_{eff} as above allows us to express the CRDM flux in the solar system in terms of the relatively well measured *local* interstellar CR flux, $d\Phi_N^{\text{LIS}}/dT_N$, and the *local* DM density, ρ_χ^{local} .

Attenuation

Once entering the atmosphere and/or soil, the CRDM flux given by Eq. (15.2) will be attenuated due to scattering on the corresponding nuclei. This can be modelled by the energy loss equation

$$\frac{dT_\chi^z}{dz} = - \sum_N n_N \int_0^{\omega_\chi^{\text{max}}} d\omega_\chi \frac{d\sigma_{\chi N}}{d\omega_\chi} \omega_\chi. \quad (15.4)$$

This equation relates the average kinetic energy at depth z , T_χ^z , to an initial energy T_χ at the top of the atmosphere ($z = 0$). Note that the sum here runs over the nuclei N in the overburden, i.e. no longer over the CR species, and ω_χ is the *energy loss* of a CRDM particle in a single collision. For elastic scattering, ω_χ equals by definition the nuclear recoil energy T_N and the maximal energy loss of a DM particle with initial kinetic energy T_χ^z is given by

$$\omega_\chi^{\text{max}} = T_N^{\text{max}} = \frac{2m_N}{s} \left[(T_\chi^z)^2 + 2m_\chi T_\chi^z \right], \quad (15.5)$$

where

$$s = (m_N + m_\chi)^2 + 2m_N T_\chi^z \quad (15.6)$$

is the (squared) CMS energy of the process. For inelastic scattering, on the other hand, the energy loss can be as high as $\omega_\chi^{\text{max}} = T_\chi^z$ (see Ref. [62] for a detailed discussion).

Recoil rate in detector

The final recoil rate of CRDM particles, from their elastic scattering with nuclei in underground detectors like XENON, is given by

$$\frac{d\Gamma_N}{dT_N} = \int_{T_\chi^{\text{min}}}^{\infty} dT_\chi \frac{d\sigma_{\chi N}}{dT_N} \frac{d\Phi_\chi}{dT_\chi}. \quad (15.7)$$

In this expression, the integration is performed over the energy of the DM particles *before* entering the atmosphere, while the elastic scattering cross section $d\sigma_{\chi N}/dT_N$ must be evaluated at the *actual* DM energy, T_χ^z , at the detector location; similarly, $T_\chi^{\text{min}} = T_\chi(T_\chi^{z,\text{min}})$ represents the minimal *initial* CRDM energy that is needed to induce a nuclear recoil of energy T_N at depth z . This requires a numerical solution of Eq. (15.4) to obtain $T_\chi^z(T_\chi)$, as well as its inverse.

In general, $d\sigma_{\chi N}/dT_N$ is a function of both s and the (spatial) momentum transfer,

$$Q^2 = 2m_N T_N. \quad (15.8)$$

If the (dominant) dependence on Q^2 factorizes – as, for example, in the case of standard form factors – then the rate in Eq. (15.7) features the *same* Q^2 -dependence as that for the standard population of non-relativistic halo DM, allowing to directly re-interpret published limits on the latter into limits on the former [23].

15.2 CRDM– routines

The CRDM routines are initialized with a call to `dsddcrdm_init`; this is typically done directly from `dsinit` and sets default values for, e.g., ρ_χ^{local} and D_{eff} , as well as common block variables that steer accuracy settings and which channels (including inelastic scattering) to include. For typical applications, the functions of greatest interest are `dsddDMCRflux` and `dsddDMCRcount`, with detailed usage illustrated in the example programs `examples/aux/DDCR_flux.f` and `examples/aux/DDCR_limits.f`, respectively.

Central functions

The function `dsddDMCRflux` returns the CRDM flux before attenuation, as given by Eq. (15.2). This function takes the product $\rho_{\chi}^{\text{local}} \cdot D_{\text{eff}}$ directly as input, thus overriding the default values set in the initialization process described above. The sum over CR nuclei N is performed over the dominant CR species contributing to the CRDM flux, as returned by `dscrISRflux` (see section 9.1).

The other commonly used function, `dsddDMCRcountrate`, only takes the name of a specific experiment as input (for example Xenon1T [63], but also neutrino experiments such as Borexino [64]) and returns the recoil rate in Eq. (15.7) in dimensionless units – namely in units of the recoil rate that would correspond to the reported 95 *C.L.* limit of that experiment. Internally, this is essentially a driver routine for the more general function `dsddDMCRdgammaadt` that returns the full recoil rate for a specific element.

In principle, the only interface function that a particle module needs to provide in order to access the CRDM routines is `dsddsigma`, returning the nuclear scattering cross section in the non-relativistic limit.* An important *replaceable* function, however, is `dsddsigmarel` which by default (i.e. the instance one can find in `src/dd_crmdm`) simply returns unity – but which in general is assumed to return the ratio of the full (relativistic) elastic scattering cross section to the cross section in the non-relativistic limit. The example program `examples/aux/DDCR_limits.f` (along with makefile target `DDCR_limits_q2`) illustrates how to replace this function in practice, e.g. for interactions mediated by light scalar or vector particles.

Further functions of potential interest

Per default, the CRDM routines fully implement the effect of inelastic scattering – as modelled in Ref. [62] – on the attenuation of the CRDM flux in the overburden of an experiment. For that purpose, the subroutine `dsddTDMattenuation` handles the conversion between T_{χ} and T_{χ}^z ; for code-speed-up, the solution to Eq. (15.4) is tabulated only once and then interpolated (with the full differential cross section $d\sigma_{\chi N}/d\omega_{\chi}$ provided by the function `dsddDMCRsigsoil`).

For neutrino telescopes based on liquid scintillators, furthermore, one needs to relate the proton recoil energy in the scintillator to an apparent electron T_e equivalent before one can compare Eq. (15.7) to reported limits (see Refs. [65, 23] for details). This conversion is handled by the function `dsddDMCRquenching`.

*This function also exists as an auxiliary function in `src_models/common/aux`, implying that a particle module in practice only needs to provide `dsddgpgn` for the nucleon couplings.

Chapter 16

dmd_astro: Astrophysical source functions

In the folder `src/dmd_astro` we provide a set of wrapper routines that all call the halo driver function `dsdmsdriver` in a specific way. The rest of the code communicates with `dsdmsdriver`, and hence the halo model(s), only via these functions and never by calling `dsdmsdriver` directly. The routines in `src/dmd_astro` therefore provide examples of functions that *cannot be replaced* individually in a consistent way, but only as a whole set (along with `dsdmsdriver`, in case the user wants to change the structure of the driver itself).

Chapter 17

dmd_mod: Dark matter distributions

17.1 Dark matter distributions – theory

All the dark matter detection rates depend in one way or another on the properties of the Milky Way dark matter halo. We will here outline the halo model that by default is included with DarkSUSY.

Observationally, the distribution of DM on scales relevant for DM searches is only poorly constrained. The situation is somewhat improved when instead referring to the results of large N -body simulations of gravitational clustering, which consistently find that DM halos *on average* are well described by Einasto [8] or Navarro-Frenk-White profiles [9], with the halo mass being essentially the only free parameter (after taking into account that the halo concentration strongly correlates with the halo mass [66]). On the other hand, there is a considerable halo-to-halo scatter associated to these findings, so that it remains challenging to make concrete predictions for individual objects – in particular if they are located in cosmologically somewhat ‘special’ environments like in the case of the Milky Way and its embedding in the Local Group. Even worse, baryonic physics can have a large impact on the DM profiles, especially on their inner parts most relevant for indirect detection, and even though hydrodynamic simulations taking into account such effect have made tremendous progress in recent years [67, 68, 69, 70], there is still a significant uncertainty related to the modelling of the underlying processes. In light of this situation, there is a considerable degree of freedom concerning halo models and the DM density profiles, and a code computing observables related to DM should be able to fully explore this freedom.

17.1.1 Rescaling of the WIMP density

While it is natural to assume that the DM particles described by a given particle module implemented in DarkSUSY make up most of the DM in our galaxy, they may also just constitute a sub-dominant part of a multi-component realization of DM. What is more, there might be both a thermal contribution to the cosmological DM abundance – as computed by the relic density routines in DarkSUSY – and a non-thermal contribution, e.g. via out-of-equilibrium production or via the decay of heavier particles. In this context, it is important to remember that DM detection rates only depend on the *local* DM density *of that particular DM candidate*. For the case of direct detection, e.g., the rate scales linearly with the local density *at earth*, while for indirect detection it scales linearly or quadratically with the local DM density *at the point of decay or or annihilation*, respectively.

In previous versions of the code, the ratio of thermal relic abundance returned by `dsrdomega` and observed cosmological DM abundance was used to internally rescale the results from rate

calculations. For the reasons given above, this is not fully satisfactory and in any case obscures the origin of this rescaling. Starting from DarkSUSY 6, this is therefore no longer the case. All rate routines now assume that the local DM density equals the local density in the particular DM candidate realized in the particle module – unless they explicitly take the local DM density as an input parameter (which in that case refers to the local density in the particles described by the respective DarkSUSY module). An example for such an exception are the neutrino telescope routines, because the combined effect of DM capture and annihilation makes the dependence on the local DM density more complicated. In all other cases, e.g. if DM rate routines just take a halo label as input, the user has to make sure to rescale the rates, as described above, *by hand* to reflect possible sub-dominant DM populations.

17.2 Implementation in DarkSUSY

The implementation of dark matter halo models and related quantities in the library `ds_core` follows a new and highly flexible scheme, compared to earlier versions of the code, avoiding pre-defined hardcoded functions. For convenience a few pre-defined options are provided, however these can be either complemented by other profiles eventually needed, or the entire sample configuration can be simply replaced linking to a user-defined setup - in both cases without editing routines provided in this release of the code. A further improvement compared to previous versions of DarkSUSY is that different dark matter density profiles, possibly referring to different dark matter detection targets, can be defined at the same time: E.g., one can easily switch back and forth from a computation of the local positron flux induced by dark matter annihilations or decays in the Milky Way halo to the computation of the gamma-ray flux from an external halo or a Milky Way satellite within the same particle physics scenario. Finally the present implementation simplifies the task of keeping track of consistent definitions for related quantities, such as, e.g., a proper connection between the dark matter profile and the source function for a given dark matter yield (see Chapter 6), or calling within an axisymmetric coordinate system a spherically symmetric function (and preventing the opposite).

At the basis of the implementation, there is the subroutine `dsdmsdriver` routine which acts as an interface to quantities related to the dark matter density profiles. This routine must contain a complete set of instructions on how to retrieve the different observables: E.g. it checks the scaling of the various DM source functions (Chapter 6 with the DM density ρ_χ – namely ρ for decaying dark matter and ρ^2 for dark matter pair annihilations (in case the effect of substructures is neglected) – and passes this information to the routines for propagations of charged cosmic rays in the Galaxy (Chapter 10), assuming that such source function is axisymmetric; line of sight integration routines (needed, e.g. for the computation of gamma-ray fluxes, see Chapter 11) call this same subroutine, but may assume instead that the corresponding source function is spherically symmetric. The routine `dsdmsdriver` must contain the specification on whether the dark matter density profile is spherically symmetric or axisymmetric, and in the first case provide a consistent numerical output to both calls, in the latter return an error to the second call (since a spherically symmetric profile was expected). It may also be useful to use the `dsdmsdriver` routine for initialization calls, for instance to set parameters for a given parametric density profile, and test calls, for instance to print which dark matter profile is currently active within a set of available profiles. The input/output structure of the routine is rather general, with the first entry being however fixed to an integer flag specifying the action of the routine; currently available values and relative action are implicitly defined (through integer variables such as ‘`idmddensph`’ referring to the spherical dark matter density profile) in an include file and are global parameter. Such set of definitions should not be changed, but can be enlarged in case further profile-related quantities would be needed.

While a specific `dsdmsdriver` routine should match the user needs in the problem at hand, the present release provides a sample version, illustrating the flexibility of the setup. In particular the version included in the library `ds_core` assumes that the dark matter density profile is spherically

symmetric and does not include dark matter substructure; it allows to choose as dark matter profile one among three parametric profiles, namely the Einasto [8], the Navarro-Frenk-White [9] and the Burkert [71] profiles, or a profile interpolated from a table of values of the dark matter density at a given radius. Besides providing parameters as needed in case of parametric profiles, to complete the initialization of a profile one should also specify: *i*) an inner truncation radius, namely some r_{ic} fixing $\rho(r) = \rho(r_{ic})$ for any $r < r_{ic}$ (the choice has an impact on predictions for dark matter rates only for very singular dark matter profiles; choosing a value which is not too small allows for a faster numerical convergence of some rate computations); *ii*) an outer truncation radius, namely some r_{oc} beyond which the profile is assumed to be zero; *iii*) the distance from the observer of the center of the profile, corresponding to the Sun galactocentric distance only in case the profile refers to the Milky Way; *iv*) whether it is a profile that refers to the Milky Way and hence for which rates that are Milky Way specific, such as the local contribution to antimatter fluxes, can be computed; *v*) whether it is a profile to be saved in a halo profile database for later use.

Regarding this last point, the code implements a procedure of associating the set of entries fully specifying a halo profile (namely the choice of the parametric profile, the corresponding parameters and the entries *i*–*v*) above) to a given *input label*, and this can be reloaded at any time when needed; in particular all indirect detection flux routines have the label among their input parameters, so that in case of several dark matter detection targets or several profiles for the same targets it becomes unambiguous which profile is being considered. On the other hand it may be the case that the user needs to loop over many different profiles without keeping track of all of them for later reuse (e.g., in a scan over parameter space in the estimate of line of sight integrals towards a dwarf satellite); in such case the profile can be defined as “temporary”, with only the latest set temporary profile available at any given time. For halos that are stored in the halo profile database, one can save and/or read from disc tabulated quantities, such as, e.g., the Green function needed for the computation of the local positron flux, for temporary profiles tables can (or in some cases need) to be computed running the code but are overwritten any time the temporary profile is changed. While in the previous releases of DarkSUSY, halo parameters were typically set via common blocks to be included in the main file, the default `dsdmsdriver.f` implements a procedure in which, when initializing a given halo profile, parameters are given as an input in association with a corresponding parameter label, and profile settings are specified as character strings appearing within the profile label.

Along with the default `dsdmsdriver.f` routine, which is unfortunately rather involved since it allows for several different options, in the present release we provide example main files which illustrate a few of the possible user needs. Those are described in the ‘quick start’ part of this manual, see Section 2.2.1, and cover examples of how to use pre-defined halo profiles, read in tabulated profiles as well as how to create completely new ones.

17.2.1 Dark matter distributions – routines

The main routines in this directory are

dsdmsdriver Main driver function for setting the halo profile and retrieving observables. All calls to the halo models go through this routine.

dsdmssethm Wrapper routine for `dsdmsdriver` to set a halo model.

dsdmsselecthm Wrapper routine for `dsdmsdriver` to select an already defined halo model.

Chapter 18

dmd_vel: Dark matter velocity distributions

18.1 Dark matter phase-space distributions – theory

The DM velocity and density profiles cannot be chosen independently, in principle, but have to satisfy consistency relations. For a spherically symmetric and isotropic system, e.g., the two profiles are related by the Eddington equation [72, 73]. A fully self-consistent implementation of phase-space distributions will be available with a later DarkSUSY version, at which point the directory `dmd_vel` will become obsolete.

Until then, the user can freely choose a DM velocity distribution among those provided in `src/dmd_vel` – but should keep in mind this consistency requirement when comparing direct detection rates (which require the local DM velocity profile) to, e.g., the gamma-ray flux from the galactic center (which requires choosing a density profile). Concretely, it is the function `dshmuDF` that returns the 3D distribution function $f(\mathbf{v})$ needed by the direct detection routines. It allows to switch between various pre-implemented functional forms, including tabulated velocity profiles, but can of course also be replaced by an arbitrary function supplied by the user (c.f. Section 3.6).

One of these options is the often adopted truncated gaussian, which in the detector frame moving at speed v_O relative to the galactic halo reads

$$f(v) = \frac{1}{\mathcal{N}_{\text{cut}}} \frac{v^2}{uv_O\sigma} \left\{ \exp\left[-\frac{(u-v_O)^2}{2\sigma^2}\right] - \exp\left[-\frac{\min(u+v_O, v_{\text{cut}})^2}{2\sigma^2}\right] \right\} \quad (18.1)$$

for $v_{\text{esc}} < v < \sqrt{v_{\text{esc}}^2 + (v_O + v_{\text{cut}})^2}$ and zero otherwise, with $u = \sqrt{v^2 + v_{\text{esc}}^2}$ and

$$\mathcal{N}_{\text{cut}} = \frac{v_{\text{cut}}}{\sigma} \exp\left(-\frac{v_{\text{cut}}^2}{2\sigma^2}\right) - \sqrt{\frac{\pi}{2}} \operatorname{erf}\left(\frac{v_{\text{cut}}}{\sqrt{2}\sigma}\right). \quad (18.2)$$

As default, we have taken the halo line-of-sight (one-dimensional) velocity dispersion $\sigma = 120$ km/s,* the galactic escape speed $v_{\text{cut}} = 600$ km/s, the relative Earth-halo speed $v_O = 264$ km/s (a yearly average) and the Earth escape speed $v_{\text{esc}} = 11.9$ km/s. These parameters can be changed by the user.

*Other authors write $\exp(-3v^2/2\bar{v}^2)$, in which case $\bar{v} = \sqrt{3}\sigma$.

Chapter 19

fi: Freeze-In

19.1 Freeze-in – theory

The freeze-in mechanism to produce DM in the early Universe describes scenarios where the interaction with the primordial heat bath is so weak that equilibrium was never obtained [74, 75, 76]. This is in contrast to the situation of typical WIMPs that acquire their relic density through the *freeze-out* mechanism, as described in Section 22.1. The implementation of freeze-in routines in DarkSUSY follows a formulation [16] of the freeze-in process that maximizes the analogy with production by freeze-out.

Concretely, the abundance of feably interacting particles (FIMPs) increases as

$$\frac{dY_\chi}{dx} = \frac{(n_\chi^{\text{MB}})^2}{xs\tilde{H}} \langle \sigma v \rangle \quad (19.1)$$

as long as it stays far below the equilibrium abundance (which we *define* here as the freeze-in regime). Here, the DM abundance is $Y_\chi \equiv n_\chi/s$, with n_χ the DM density and s the total entropy (assumed to be conserved). Further, $x \equiv m_\chi/T$ and $\tilde{H} \equiv H/[1 + (1/3)d(\log g_{\text{eff}}^s)/d(\log T)]$, where g_{eff}^s denote the effective entropy degrees of freedom.

Following slightly different conventions, this is essentially Eq. (22.54) for the freeze-out case, with three important differences:

- By assumption, there is no ‘backreaction’ term that would describe the annihilation of FIMPs into heat bath particles.
- The factor n_χ^{MB} denotes the number density corresponding to a *would-be* Maxwell-Boltzmann distribution of DM particles. Unlike in the case of WIMPs – cf. the assumption $f_\chi(E, T) = A(T)f_\chi^{\text{eq}}(E/T)$ entering in the derivation of Eq. (22.54) – the above formulation does not assume that the *actual* DM distribution is related to a thermal one.
- In analogy with the freeze-out case, Eq. (19.1) is formulated in terms the DM *annihilation* cross section σ . Different from WIMPs, however, DM is relativistic during a significant part of the production period, such that in particular effects due to quantum statistics can not be neglected, viz. Bose enhancement/Fermi suppression due to large occupancies of the SM particles in the final states.

The thermal average appearing in Eq. (19.1) is most conveniently expressed in terms of a model-

independent thermal kernel and a model-dependent invariant rate W_{eff} :

$$\langle \sigma v \rangle = \int_1^\infty d\tilde{s} \frac{x\sqrt{\tilde{s}-1} K_1(2\sqrt{\tilde{s}x})}{2m_\chi^2 K_2^2(x)} W_{\text{eff}}(s, T). \quad (19.2)$$

The main conceptual difference to the corresponding expression for WIMPs is that the invariant rate is itself T -dependent, as a direct consequence of the third point in the bullet list above. Concretely, the invariant rate is given by

$$W_{\text{eff}}(s, T) \equiv 16m_\chi^2 \frac{x\tilde{s}\sqrt{\tilde{s}-1}}{K_1(2\sqrt{\tilde{s}x})} \int_1^\infty d\gamma \sqrt{\gamma^2-1} e^{-2\sqrt{\tilde{s}x}\gamma} \sum_{\psi_1\psi_2} \sigma_{\chi\chi \rightarrow \psi_1\psi_2}(s, \gamma), \quad (19.3)$$

where the integration is over Lorentz boosts γ from the CMS to the cosmic rest frame, and the in-medium cross section can be written as

$$\sigma_{\chi\chi \rightarrow \psi_1\psi_2}(s, \gamma) = \frac{N_\psi^{-1}}{8\pi s} \frac{|\mathbf{k}_{\text{CM}}|}{\sqrt{s-4m_\chi^2}} \int_{-1}^1 \frac{d\cos\theta}{2} |\overline{\mathcal{M}}|_{\chi\chi \rightarrow \psi_1\psi_2}^2(s, \cos\theta) G_{\psi_1\psi_2}(\gamma, s, \cos\theta), \quad (19.4)$$

where $N_\psi = 2$ for identical SM particles ($\psi_1 = \psi_2$) and $N_\psi = 1$ otherwise. The effect of quantum statistics in the final state, leading to an enhancement or decrease of the corresponding cross section in vacuum, is encoded in the quantity

$$G_{\psi_1\psi_2}(\gamma, s, \cos\theta) = 1 + \epsilon_\psi^2 e^{-2\sqrt{\tilde{s}x}\gamma} - \epsilon_\psi \left\{ e^{-\frac{1}{T}(E_{\psi_1}\gamma + |\mathbf{k}_{\text{CM}}| \cos\theta \sqrt{\gamma^2-1})} + e^{-\frac{1}{T}(E_{\psi_2}\gamma - |\mathbf{k}_{\text{CM}}| \cos\theta \sqrt{\gamma^2-1})} \right\}, \quad (19.5)$$

with $E_{\psi_i} = \sqrt{\mathbf{k}_{\text{CM}}^2 + m_{\psi_i}^2}$ and $\epsilon_\psi = +1$ ($\epsilon_\psi = -1$) for fermions (bosons).

It is worth noting that for $G_{\psi_1\psi_2} = 1$ – which formally corresponds to setting ‘ $\epsilon_\psi = 0$ ’ in the last expression – one recovers the standard expression for the cross section in the CMS frame which, in particular, is independent of γ . The integral in Eq. (19.4) can then be solved analytically [16] resulting as expected in the T -independent expression (22.18) for W_{eff} in the non-relativistic case.

The other relevant observation is that Eq. (19.2) allows to include other T -dependent effects in the model-dependent part as well. Notable examples include thermal masses as well as phase transitions, which generally affect both interaction rates and the spectrum of relevant SM states. While this can in principle also affect relic density calculations for WIMPs – if freeze-out happens, e.g., very close to a phase transition – it is generically much more important for freeze-in because of the large range of temperatures during which freeze-in production is efficient.

19.2 Freeze-in – routines

The routines provided in `src/fi` are accessible after an initial call to `dsfi_init` (as done directly from `dsinit`), which sets some global performance flags related to the handling of SM phase-transitions and finite-temperature effects (see documentation in the `.f` file for further details). Typically, the function of greatest interest will be `dsfi2to2oh2`, and there should be no need to call any of the other routines directly.

`dsfi2to2oh2` numerically solves the Boltzmann equation (19.1) and returns the resulting DM relic density in terms of Ωh^2 as a function of the reheating temperature (defined as the starting point of the integration). As explained in more detail in the function header, and in analogy to the corresponding routine `dsrdomega` for freeze-out calculations, `dsfi2to2oh2` further takes two performance flags as input – ‘`option`’ steering particle-module specific settings, and ‘`fast`’ allowing

to adjust global settings (such as whether to take into account the effect of quantum statistics) to be different from the default ones.

Technically, the integration of Eq. (19.1) – with the r.h.s. provided by `dsfi2to2rhs` – is performed in `dsfi2to2ab`, taking special care of not missing the QCD and EW phase transitions. Another function worth mentioning is `dsfithav` which returns the thermally averaged cross section as given in Eq. (19.2). Internally, this function rescales the invariant rate to a normalization comparable to WIMPs, before using the same thermal average routine `dsrdthav` as in that case (which requires a call to the subroutine `dsrd_set` in order to notify the routines in `/rd` that the invariant rate depends on p and T rather than only on p).

There are two interface functions that a particle physics module must provide for the freeze-in routines in `src/` to work: *i*) subroutine `dsrdparticles` provides kinematic information about (potentially T -dependent) thresholds and resonances. The same interface function is also required by the freeze-out routines in `src/rd`. *ii*) a function `dsanwx_finiteT` returning the temperature-dependent rate entering in Eq. (19.2).

Chapter 20

ini: Initialization routines

20.1 Initialisation routines

This directory contains general initialisation routines that need to be called to prepare calculations with DarkSUSY, but which are independent of the particle physics. Most importantly, it contains the subroutine `dsinit`, which should be called at the beginning of any program using DarkSUSY. In particular, this routine calls various more specific initialisation routines that are relevant only for specific applications (such as the relic density calculation) and therefore reside in the respective directory in `src/`. It also calls `dsinit_module`, which initializes the specific particle module that the user has chosen to link to when compiling the main program. Another subroutine called by `dsinit` is `ini/dsreadnuclides`. It reads in names and properties of nuclei and stores them in common blocks, to which both direct detection and solar/earth capture routines need access.

Lastly, the directory `/ini` also contains two files `dsdirname.c.in` and `dsvername.c.in`. These are needed to provide a system-wide reference to the installation directory and version of DarkSUSY, respectively. These refer to include files that are set at configure time, `dsdir.h` and `dsver.h`. The user should never have to modify these manually as they are determined at the configure stage.

Chapter 21

kd: Kinetic decoupling

21.1 Kinetic decoupling and microhalos (kd) – theory

Even after *chemical decoupling*, which sets the DM relic density (see Section 22.1.1), DM frequently scatters with the very abundant standard model particles and thereby stays in local thermal equilibrium with the heat bath until the temperature has dropped by another factor of between 10 and a few 1000 (for typical WIMPs like, e.g. neutralino DM in the MSSM); after *kinetic decoupling*, even these scattering events cease and DM no longer interacts with standard model particles. Inhomogeneities in the DM density can only develop after this has happened, and the DM particles have sufficiently cooled down so that free streaming becomes negligible. The scale of kinetic decoupling can therefore directly be translated into a cutoff in the power spectrum of (dark) matter fluctuations and thus the size of the smallest (at least when not taking into account primordial black holes) gravitationally bound objects in the universe (for a review, see Ref. [17]).

21.1.1 Kinetic decoupling

The process of kinetic decoupling is governed by the full Boltzmann equation for the WIMP distribution function $f(\mathbf{x}, \mathbf{p})$, which in a flat Friedmann-Robertson Walker spacetime reads

$$E(\partial_t - H \mathbf{p} \cdot \nabla_{\mathbf{p}}) f = C[f], \quad (21.1)$$

where $C[f]$ is known as the collision term. The Boltzmann equation quoted in Eq. (22.1) for the description of the (chemical) DM freeze-out process is actually just the first moment of this expression, i.e. obtained by integrating it over $\int d^3p$ (after dividing it by E). As was realized in [77, 17], kinetic decoupling can be described to a very high precision by considering, instead, the *second* moment of Eq. (21.1). For this purpose, one introduces the WIMP "temperature" T_χ ,

$$\int \frac{d^3p}{(2\pi)^3} \frac{\mathbf{p}^2}{E} f(\mathbf{p}) \equiv 3 T_\chi n_\chi, \quad (21.2)$$

as a parameter that characterizes the deviation from thermal equilibrium (for which $T_\chi = T$ holds). After kinetic decoupling, the DM 'temperature' will simply decrease due to the expansion of the universe and, for non-relativistic DM, scale like $T_\chi \propto a^{-2}$. It is thus natural to define the moment of decoupling as the transition between these two regimes [17]. Allowing for the scattering partners to have a different temperature ($T_{\tilde{\gamma}}$) than that of the photons (T), this implies

$$T_\chi(T) = \begin{cases} T_{\tilde{\gamma}}(T) & \text{for } T \gtrsim T_{\text{kd}} \\ T_{\tilde{\gamma}}(T_{\text{kd}}) \left(\frac{a(T_{\text{kd}})}{a(T)} \right)^2 & \text{for } T \lesssim T_{\text{kd}} \end{cases} \quad (21.3)$$

For practical purposes, one may now further introduce

$$x \equiv m_\chi/T, \quad (21.4)$$

$$y \equiv m_\chi T_\chi s^{-2/3}. \quad (21.5)$$

Multiplying Eq.(21.1) by \mathbf{p}^2/E , integrating it over \mathbf{p} and keeping only the leading order terms* in \mathbf{p}^2/m_χ^2 then results in [17, 79]

$$\frac{d \log y}{d \log x} = \left(1 - \frac{1}{3} \frac{d \log g_{*S}}{d \log x}\right) \frac{\gamma(T_\chi)}{H(T)} \left(\frac{y_{\text{eq}}}{y} - 1\right). \quad (21.6)$$

Here, g_{*S} is the number of effective entropy degrees of freedom, y_{eq} is the value of y in thermal equilibrium and γ denotes the momentum transfer rate,

$$\gamma(T_\chi) = \frac{1}{48\pi^3 g_\chi T_\chi m_\chi^3} \sum_i \int d\omega k^4 (1 \mp g_i^\pm) g_i^\pm(\omega) |\mathcal{M}|_{t=0}^2_{s=m_\chi^2+2m_\chi\omega+m_i^2}, \quad (21.7)$$

where the sum runs over all DM scattering partners (counting, e.g., electrons and positrons separately), $k \equiv |\mathbf{k}|$ is their momentum and ω their energy. The g_i denote the SM distribution functions, which are assumed to be thermal (note, however, that no assumption has been made about the form of the WIMP distribution function f). The scattering amplitude squared in this expression, $|\mathcal{M}|^2$, is understood to be *summed* over all internal (spin or color) degrees of freedom, including initial ones. If it is not Taylor expandable around $t = 0$, one has to make the replacement [80, 81]

$$|\mathcal{M}|_{t=0}^2_{s=m_\chi^2+2m_\chi\omega+m_i^2} \longrightarrow \langle |\mathcal{M}|^2 \rangle_t \equiv \frac{1}{8k^4} \int_{-4k_{CM}^2}^0 dt(-t) |\mathcal{M}|^2 \quad (21.8)$$

in the above expression. We may easily check that the asymptotic behaviour for T_χ described by Eq. (21.6) is consistent with the expectation outlined above: At large T , we have $\gamma \gg H$, thus enforcing $T_\chi = T$; when T becomes small, the WIMPs completely decouple from the thermal bath and y stays constant, i.e. $T_\chi \propto s^{2/3} \propto a^{-2}$. A practical way to determine the *kinetic decoupling* temperature T_{kd} as defined in Eq. (21.3) is by solving the above differential equation until y stays constant (indicated by the ' $x \rightarrow \infty$ ')

$$x_{\text{kd}} = \frac{m_\chi}{T_{\text{kd}}} \equiv g_{\text{eff}}(T_{\text{kd}}) y|_{x \rightarrow \infty}. \quad (21.9)$$

21.1.2 The smallest protohalos

Before kinetic decoupling, WIMPs are tightly coupled to the heat bath, so any small-scale perturbations in the DM fluid would immediately be washed out. For temperatures $T \lesssim T_{\text{kd}}$, however, this is no longer the case and perturbations in the DM density start to develop under the influence of gravity; however, the remaining viscous coupling between the two fluids and, subsequently, the free-streaming of the WIMPs generate an exponential cutoff in the power spectrum [82], with a characteristic comoving damping scale k_{fs} . The WIMP mass contained in a sphere of the corresponding size is thus given by [17]

$$M_{\text{fs}} \approx \frac{4\pi}{3} \rho_\chi \left(\frac{\pi}{k_{\text{fs}}}\right)^3 = 4.0 \times 10^{-6} \left(\frac{1 + \ln(g_{\text{eff}}^{1/4} T_{\text{kd}}/30 \text{ MeV})/18.6}{(m_\chi/100 \text{ GeV})^{1/2} g_{\text{eff}}^{1/4} (T_{\text{kd}}/30 \text{ MeV})^{1/2}}\right)^3 M_\odot. \quad (21.10)$$

*For very early decoupling, next-order correction terms may become relevant [78].

Acoustic oscillations also have to be taken into account as a damping mechanism and lead to a similar exponential cutoff in the power spectrum [83, 84]. In this case, the characteristic damping mass is given by the total amount of DM inside the horizon at the time of kinetic decoupling:

$$M_{\text{ao}} \approx \frac{4\pi}{3} \frac{\rho_\chi}{H^3} \Big|_{T=T_{\text{kd}}} = 3.4 \times 10^{-6} \left(\frac{T_{\text{kd}} g_{\text{eff}}^{1/4}}{50 \text{ MeV}} \right)^{-3} M_\odot. \quad (21.11)$$

Note that T_{kd} in the above expressions only holds when using the definition given in Eq. (21.3); for an alternative definition, the expected magnitude of the cutoff mass has to be correspondingly re-scaled.

In general, the actual cutoff in the power spectrum is given by $M_{\text{cut}} = \max[M_{\text{fs}}, M_{\text{ao}}]$; which of the two physically independent damping mechanisms is more efficient depends on the particle nature of the WIMP. The expected mass for the smallest gravitationally bound objects in the universe is then also simply given by M_{cut} . Numerically, the formation of protohalos with masses down to the cutoff scale has been confirmed and their evolution could be followed until a redshift of $z \sim 26$ [85]; the further survival probabilities of these objects, however, as well as the resulting consequences for the indirect detection of DM, are subject to a presently still ongoing discussion.

21.2 Kinetic decoupling – routines

Before using any of the routines provided in `src/kd` for the first time, one has to call `dskd_set` in order to make some necessary initializations; in particular, a call to this routine ensures that the relevant tables for the relativistic degrees of freedom in the early universe are correctly read in. Typically, the subroutines of greatest interest will be `dskdtkd` and `dskdmcut`, and there should be no need to call any of the other routines directly.

`dskdtkd` numerically solves the Boltzmann equation (21.6) and determines T_{kd} as given in Eq. (21.9). Here, special care is taken to accurately handle potential resonances in the scattering amplitude; to this end, `dskdboltz_init` identifies the location of all relevant resonances and passes this information to `dskdgammarate` where the integral of Eq. (21.7) is performed.

Finally, `dskdmcut` returns the mass cutoff in the power spectrum, with an input parameter determining whether it is M_{fs} or M_{ao} ; the default call results in $M_{\text{cut}} = \max[M_{\text{fs}}, M_{\text{ao}}]$, i.e. the mass of the smallest protohalos.

There are three interface functions that a particle physics module must provide for the kinetic decoupling routines in `src/` to work: `dskdm2` returns the full scattering matrix element squared, evaluated at $t = 0$ or averaged over t , while `dskdm2simp` returns only the leading contribution for small ω (expressed as a simple power-law in ω , in which case there exists an analytic solution for T_{kd} [77]). Lastly, the particle physics module must provide a routine `dskdparticles` which, in analogy to the routine `dsrdparticles` for the case of chemical decoupling discussed above, sets the location of resonances in $|\mathcal{M}|^2$.

Chapter 22

rd: Relic density

22.1 Relic density – theoretical background

22.1.1 The Boltzmann equation and thermal averaging

Griest and Seckel [86] have worked out the Boltzmann equation when coannihilations are included. We start by reviewing their expressions and then continue by rewriting them into a more convenient form that resembles the familiar case without coannihilations. This allows us to use similar expressions for calculating thermal averages and solving the Boltzmann equation whether coannihilations are included or not. The implementation in DarkSUSY is based upon the work done in [13].

22.1.2 Review of the Boltzmann equation with coannihilations

Consider annihilation of N DM particles χ_i ($i = 1, \dots, N$) with masses m_i and internal degrees of freedom (statistical weights) g_i . Also assume that $m_1 \leq m_2 \leq \dots \leq m_{N-1} \leq m_N$ and that R -parity is conserved. Note that for the mass of the lightest stable of the particles we will use the notation m_χ and m_1 interchangeably.

The evolution of the number density n_i of particle i is

$$\begin{aligned} \frac{dn_i}{dt} = & -3Hn_i - \sum_{j=1}^N \langle \sigma_{ij} v_{ij} \rangle (n_i n_j - n_i^{\text{eq}} n_j^{\text{eq}}) \\ & - \sum_{j \neq i} [\langle \sigma'_{Xij} v_{ij} \rangle (n_i n_X - n_i^{\text{eq}} n_X^{\text{eq}}) - \langle \sigma'_{Xji} v_{ij} \rangle (n_j n_X - n_j^{\text{eq}} n_X^{\text{eq}})] \\ & - \sum_{j \neq i} [\Gamma_{ij} (n_i - n_i^{\text{eq}}) - \Gamma_{ji} (n_j - n_j^{\text{eq}})]. \end{aligned} \quad (22.1)$$

The first term on the right-hand side is the dilution due to the expansion of the Universe. H is the Hubble parameter. The second term describes $\chi_i \chi_j$ annihilations, whose total annihilation cross section is

$$\sigma_{ij} = \sum_X \sigma(\chi_i \chi_j \rightarrow X). \quad (22.2)$$

The third term describes $\chi_i \rightarrow \chi_j$ conversions by scattering off the cosmic thermal background,

$$\sigma'_{Xij} = \sum_Y \sigma(\chi_i X \rightarrow \chi_j Y) \quad (22.3)$$

being the inclusive scattering cross section. The last term accounts for χ_i decays, with inclusive decay rates

$$\Gamma_{ij} = \sum_X \Gamma(\chi_i \rightarrow \chi_j X). \quad (22.4)$$

In the previous expressions, X and Y are (sets of) standard model particles involved in the interactions, v_{ij} is the ‘relative velocity’ defined by

$$v_{ij} = \frac{\sqrt{(p_i \cdot p_j)^2 - m_i^2 m_j^2}}{E_i E_j} \quad (22.5)$$

with p_i and E_i being the four-momentum and energy of particle i , and finally n_i^{eq} is the equilibrium number density of particle χ_i ,

$$n_i^{\text{eq}} = \frac{g_i}{(2\pi)^3} \int d^3 \mathbf{p}_i f_i \quad (22.6)$$

where \mathbf{p}_i is the three-momentum of particle i , and f_i is its equilibrium distribution function. In the Maxwell-Boltzmann approximation it is given by

$$f_i = e^{-E_i/T}. \quad (22.7)$$

The thermal average $\langle \sigma_{ij} v_{ij} \rangle$ is defined with equilibrium distributions and is given by

$$\langle \sigma_{ij} v_{ij} \rangle = \frac{\int d^3 \mathbf{p}_i d^3 \mathbf{p}_j f_i f_j \sigma_{ij} v_{ij}}{\int d^3 \mathbf{p}_i d^3 \mathbf{p}_j f_i f_j} \quad (22.8)$$

Normally, the decay rate of particles χ_i other than the lightest which is stable is much faster than the age of the universe. Since we have assumed R -parity conservation, all of these particles decay into the lightest one. So its final abundance is simply described by the sum

$$n = \sum_{i=1}^N n_i. \quad (22.9)$$

For n we get the following evolution equation

$$\frac{dn}{dt} = -3Hn - \sum_{i,j=1}^N \langle \sigma_{ij} v_{ij} \rangle (n_i n_j - n_i^{\text{eq}} n_j^{\text{eq}}) \quad (22.10)$$

where the terms on the second and third lines in Eq. (22.1) cancel in the sum.

The scattering rate of DM particles off particles in the thermal background is much faster than their annihilation rate, because the scattering cross sections σ'_{Xij} are of the same order of magnitude as the annihilation cross sections σ_{ij} but the background particle density n_X is much larger than each of the DM particle densities n_i when the former are relativistic and the latter are non-relativistic, and so suppressed by a Boltzmann factor. In this case, the χ_i distributions remain in thermal equilibrium, and in particular their ratios are equal to the equilibrium values,

$$\frac{n_i}{n} \simeq \frac{n_i^{\text{eq}}}{n^{\text{eq}}}. \quad (22.11)$$

We then get

$$\frac{dn}{dt} = -3Hn - \langle \sigma_{\text{eff}} v \rangle (n^2 - n_{\text{eq}}^2) \quad (22.12)$$

where

$$\langle \sigma_{\text{eff}} v \rangle = \sum_{ij} \langle \sigma_{ij} v_{ij} \rangle \frac{n_i^{\text{eq}} n_j^{\text{eq}}}{n^{\text{eq}} n^{\text{eq}}}. \quad (22.13)$$

22.1.3 Thermal averaging

So far the reviewing. Now let's continue by reformulating the thermal averages into more convenient expressions.

We rewrite Eq. (22.13) as

$$\langle \sigma_{\text{eff}} v \rangle = \frac{\sum_{ij} \langle \sigma_{ij} v_{ij} \rangle n_i^{\text{eq}} n_j^{\text{eq}}}{n_{\text{eq}}^2} = \frac{A}{n_{\text{eq}}^2}. \quad (22.14)$$

For the denominator we obtain, using Boltzmann statistics for f_i ,

$$n^{\text{eq}} = \sum_i n_i^{\text{eq}} = \sum_i \frac{g_i}{(2\pi)^3} \int d^3 p_i e^{-E_i/T} = \frac{T}{2\pi^2} \sum_i g_i m_i^2 K_2 \left(\frac{m_i}{T} \right) \quad (22.15)$$

where K_2 is the modified Bessel function of the second kind of order 2.

The numerator is the total annihilation rate per unit volume at temperature T ,

$$A = \sum_{ij} \langle \sigma_{ij} v_{ij} \rangle n_i^{\text{eq}} n_j^{\text{eq}} = \sum_{ij} \frac{g_i g_j}{(2\pi)^6} \int d^3 \mathbf{p}_i d^3 \mathbf{p}_j f_i f_j \sigma_{ij} v_{ij} \quad (22.16)$$

It is convenient to cast it in a covariant form,

$$A = \sum_{ij} \int W_{ij} \frac{g_i f_i d^3 \mathbf{p}_i}{(2\pi)^3 2E_i} \frac{g_j f_j d^3 \mathbf{p}_j}{(2\pi)^3 2E_j}. \quad (22.17)$$

W_{ij} is the (unpolarized) annihilation rate per unit volume corresponding to the covariant normalization of $2E$ colliding particles per unit volume. W_{ij} is a dimensionless Lorentz invariant, related to the (unpolarized) cross section through*

$$W_{ij} = 4p_{ij} \sqrt{s} \sigma_{ij} = 4\sigma_{ij} \sqrt{(p_i \cdot p_j)^2 - m_i^2 m_j^2} = 4E_i E_j \sigma_{ij} v_{ij}. \quad (22.18)$$

Here

$$p_{ij} = \frac{[s - (m_i + m_j)^2]^{1/2} [s - (m_i - m_j)^2]^{1/2}}{2\sqrt{s}} \quad (22.19)$$

is the momentum of particle χ_i (or χ_j) in the center-of-mass frame of the pair $\chi_i \chi_j$.

Averaging over initial and summing over final internal states, the contribution to W_{ij} of a general n -body final state is

$$W_{ij}^{n\text{-body}} = \frac{1}{g_i g_j S_f} \sum_{\text{internal d.o.f.}} \int |\mathcal{M}|^2 (2\pi)^4 \delta^4(p_i + p_j - \sum_f p_f) \prod_f \frac{d^3 \mathbf{p}_f}{(2\pi)^3 2E_f}, \quad (22.20)$$

where S_f is a symmetry factor accounting for identical final state particles (if there are K sets of N_k identical particles, $k = 1, \dots, K$, then $S_f = \prod_{k=1}^K N_k!$). In particular, the contribution of a two-body final state can be written as

$$W_{ij \rightarrow kl}^{2\text{-body}} = \frac{p_{kl}}{16\pi^2 g_i g_j S_{kl} \sqrt{s}} \sum_{\text{internal d.o.f.}} \int |\mathcal{M}(ij \rightarrow kl)|^2 d\Omega, \quad (22.21)$$

where p_{kl} is the final center-of-mass momentum, S_{kl} is a symmetry factor equal to 2 for identical final particles and to 1 otherwise, and the integration is over the outgoing directions of one of the final particles. As usual, an average over initial internal degrees of freedom is performed.

*The quantity w_{ij} in Ref. [87] is $W_{ij}/4$.

We now reduce the integral in the covariant expression for A , Eq. (22.17), from 6 dimensions to 1. Using Boltzmann statistics for f_i (a good approximation for $T \lesssim m$)

$$A = \sum_{ij} \int g_i g_j W_{ij} e^{-E_i/T} e^{-E_j/T} \frac{d^3 \mathbf{p}_i}{(2\pi)^3 2E_i} \frac{d^3 \mathbf{p}_j}{(2\pi)^3 2E_j}, \quad (22.22)$$

where \mathbf{p}_i and \mathbf{p}_j are the three-momenta and E_i and E_j are the energies of the colliding particles. Following the procedure in Ref. [12] we then rewrite the momentum volume element as

$$d^3 \mathbf{p}_i d^3 \mathbf{p}_j = 4\pi |\mathbf{p}_i| E_i dE_i 4\pi |\mathbf{p}_j| E_j dE_j \frac{1}{2} d\cos\theta \quad (22.23)$$

where θ is the angle between \mathbf{p}_i and \mathbf{p}_j . Then we change integration variables from E_i, E_j, θ to E_+, E_- and s , given by

$$\begin{cases} E_+ &= E_i + E_j \\ E_- &= E_i - E_j \\ s &= m_i^2 + m_j^2 + 2E_i E_j - 2|\mathbf{p}_i||\mathbf{p}_j| \cos\theta, \end{cases} \quad (22.24)$$

whence the volume element becomes

$$\frac{d^3 \mathbf{p}_i}{(2\pi)^3 2E_i} \frac{d^3 \mathbf{p}_j}{(2\pi)^3 2E_j} = \frac{1}{(2\pi)^4} \frac{dE_+ dE_- ds}{8}, \quad (22.25)$$

and the integration region $\{E_i \geq m_i, E_j \geq m_j, |\cos\theta| \leq 1\}$ transforms into

$$s \geq (m_i + m_j)^2, \quad (22.26)$$

$$E_+ \geq \sqrt{s}, \quad (22.27)$$

$$\left| E_- - E_+ \frac{m_j^2 - m_i^2}{s} \right| \leq 2p_{ij} \sqrt{\frac{E_+^2 - s}{s}}. \quad (22.28)$$

Notice now that the product of the equilibrium distribution functions depends only on E_+ and not E_- due to the Maxwell-Boltzmann approximation, and that the invariant rate W_{ij} depends only on s due to the neglect of final state statistical factors. Hence we can immediately integrate over E_- ,

$$\int dE_- = 4p_{ij} \sqrt{\frac{E_+^2 - s}{s}}. \quad (22.29)$$

The volume element is now

$$\frac{d^3 \mathbf{p}_i}{(2\pi)^3 2E_i} \frac{d^3 \mathbf{p}_j}{(2\pi)^3 2E_j} = \frac{1}{(2\pi)^4} \frac{p_{ij}}{2} \sqrt{\frac{E_+^2 - s}{s}} dE_+ ds \quad (22.30)$$

We now perform the E_+ integration. We obtain

$$A = \frac{T}{32\pi^4} \sum_{ij} \int_{(m_i+m_j)^2}^{\infty} ds g_i g_j p_{ij} W_{ij} K_1 \left(\frac{\sqrt{s}}{T} \right) \quad (22.31)$$

where K_1 is the modified Bessel function of the second kind of order 1.

We can take the sum inside the integral and define an effective annihilation rate W_{eff} through

$$\sum_{ij} g_i g_j p_{ij} W_{ij} = g_1^2 p_{\text{eff}} W_{\text{eff}} \quad (22.32)$$

with

$$p_{\text{eff}} = p_{11} = \frac{1}{2} \sqrt{s - 4m_1^2}. \quad (22.33)$$

In other words

$$W_{\text{eff}} = \sum_{ij} \frac{p_{ij}}{p_{11}} \frac{g_i g_j}{g_1^2} W_{ij} = \sum_{ij} \sqrt{\frac{[s - (m_i - m_j)^2][s - (m_i + m_j)^2]}{s(s - 4m_1^2)}} \frac{g_i g_j}{g_1^2} W_{ij}. \quad (22.34)$$

Because $W_{ij}(s) = 0$ for $s \leq (m_i + m_j)^2$, the radicand is never negative.

In terms of cross sections, this is equivalent to the definition

$$\sigma_{\text{eff}} = \sum_{ij} \frac{p_{ij}^2}{p_{11}^2} \frac{g_i g_j}{g_1^2} \sigma_{ij}. \quad (22.35)$$

Eq. (22.31) then reads

$$A = \frac{g_1^2 T}{32\pi^4} \int_{4m_1^2}^{\infty} ds p_{\text{eff}} W_{\text{eff}} K_1 \left(\frac{\sqrt{s}}{T} \right) \quad (22.36)$$

This can be written in a form more suitable for numerical integration by using p_{eff} instead of s as integration variable. From Eq. (22.33), we have $ds = 8p_{\text{eff}} dp_{\text{eff}}$, and

$$A = \frac{g_1^2 T}{4\pi^4} \int_0^{\infty} dp_{\text{eff}} p_{\text{eff}}^2 W_{\text{eff}} K_1 \left(\frac{\sqrt{s}}{T} \right) \quad (22.37)$$

with

$$s = 4p_{\text{eff}}^2 + 4m_1^2 \quad (22.38)$$

So we have succeeded in rewriting A as a 1-dimensional integral.

From Eqs. (22.37) and (22.15), the thermal average of the effective cross section results

$$\langle \sigma_{\text{eff}} v \rangle = \frac{\int_0^{\infty} dp_{\text{eff}} p_{\text{eff}}^2 W_{\text{eff}} K_1 \left(\frac{\sqrt{s}}{T} \right)}{m_1^4 T \left[\sum_i \frac{g_i}{g_1} \frac{m_i^2}{m_1^2} K_2 \left(\frac{m_i}{T} \right) \right]^2}. \quad (22.39)$$

This expression is very similar to the case without coannihilations, the differences being the denominator and the replacement of the annihilation rate with the effective annihilation rate. In the absence of coannihilations, this expression correctly reduces to the formula in Gondolo and Gelmini [12].

The definition of an effective annihilation rate independent of temperature is a remarkable calculational advantage. As in the case without coannihilations, the effective annihilation rate can in fact be tabulated in advance (see Section 22.5 for details), before taking the thermal average and solving the Boltzmann equation.

In the effective annihilation rate, coannihilations appear as thresholds at \sqrt{s} equal to the sum of the masses of the coannihilating particles. We show an example for neutralino DM (in the `mssm` module) in Fig. 22.1 where it is clearly seen that the coannihilation thresholds appear in the effective invariant rate just as final state thresholds do. For the same example, Fig. 22.2 shows the differential annihilation rate per unit volume dA/dp_{eff} , the integrand in Eq. (22.37), as a function of p_{eff} . We have chosen a temperature $T = m_\chi/20$, a typical freeze-out temperature. The Boltzmann suppression contained in the exponential decay of K_1 at high p_{eff} is clearly visible. At higher temperatures the peak shifts to the right and at lower temperatures to the left. For the particular model shown in Figs. 22.1–22.2, the relic density results $\Omega_\chi h^2 = 0.030$ when coannihilations are included and $\Omega_\chi h^2 = 0.18$ when they are not. Coannihilations have lowered $\Omega_\chi h^2$ by a factor of 6.

22.1.4 Reformulation of the Boltzmann equation

We now follow Gondolo and Gelmini [12] to put Eq. (22.12) in a more convenient form by considering the ratio of the number density to the entropy density,

$$Y = \frac{n}{s}. \quad (22.40)$$

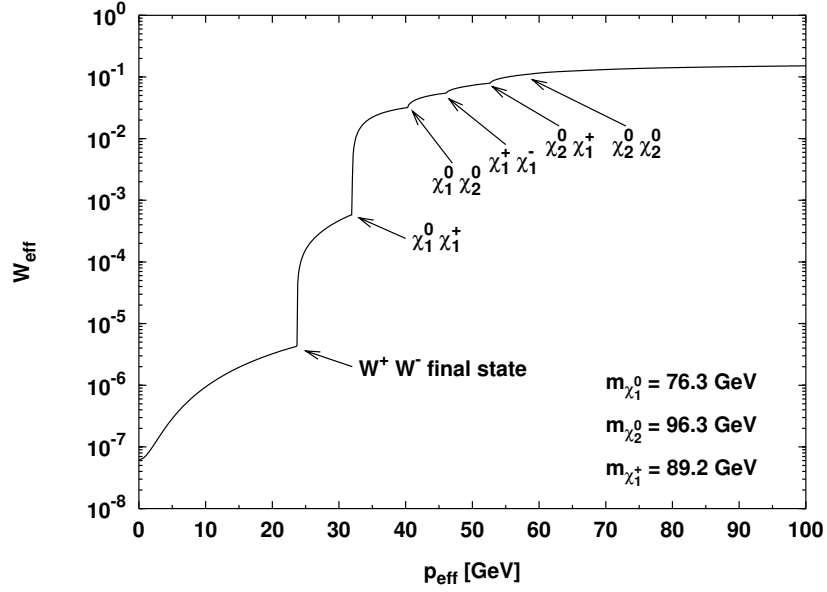


Figure 22.1: The effective invariant annihilation rate W_{eff} as a function of p_{eff} for an example m_{SSM} model. The final state threshold for annihilation into W^+W^- and the coannihilation thresholds, as given by Eq. (22.34), are indicated. The $\chi_2^0\chi_2^0$ coannihilation threshold is too small to be seen.

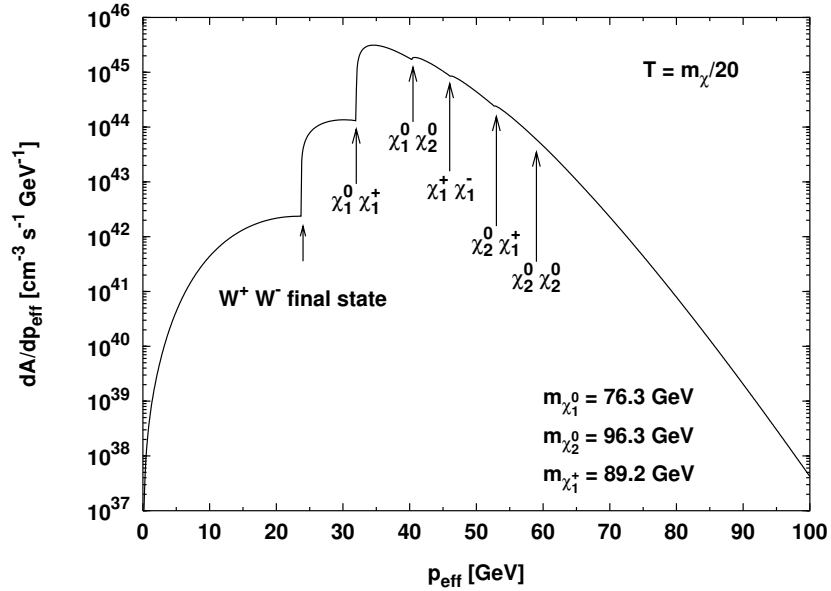


Figure 22.2: Total differential annihilation rate per unit volume dA/dp_{eff} for the same model as in Fig. 22.1, evaluated at a temperature $T = m_\chi/20$, typical of freeze-out. Notice the Boltzmann suppression at high p_{eff} .

Consider

$$\frac{dY}{dt} = \frac{d}{dt} \left(\frac{n}{s} \right) = \frac{\dot{n}}{s} - \frac{n}{s^2} \dot{s} \quad (22.41)$$

where dot means time derivative. In absence of entropy production, $S = R^3 s$ is constant (R is the scale factor). Differentiating with respect to time we see that

$$\dot{s} = -3 \frac{\dot{R}}{R} s = -3Hs \quad (22.42)$$

which yields

$$\dot{Y} = \frac{\dot{n}}{s} + 3H \frac{n}{s}. \quad (22.43)$$

Hence we can rewrite Eq. (22.12) as

$$\dot{Y} = -s \langle \sigma_{\text{eff}} v \rangle (Y^2 - Y_{\text{eq}}^2). \quad (22.44)$$

The right-hand side depends only on temperature, and it is therefore convenient to use temperature T instead of time t as independent variable. Defining $x = m_1/T$ we have

$$\frac{dY}{dx} = -\frac{m_1}{x^2} \frac{1}{3H} \frac{ds}{dT} \langle \sigma_{\text{eff}} v \rangle (Y^2 - Y_{\text{eq}}^2). \quad (22.45)$$

where we have used

$$\frac{1}{\dot{T}} = \frac{1}{\dot{s}} \frac{ds}{dT} = -\frac{1}{3Hs} \frac{ds}{dT} \quad (22.46)$$

which follows from Eq. (22.42). With the Friedmann equation in a radiation dominated universe

$$H^2 = \frac{8\pi G \rho}{3}, \quad (22.47)$$

where G is the gravitational constant, and the usual parameterization of the energy and entropy densities in terms of the effective degrees of freedom g_{eff} and h_{eff} ,

$$\rho = g_{\text{eff}}(T) \frac{\pi^2}{30} T^4, \quad s = h_{\text{eff}}(T) \frac{2\pi^2}{45} T^3, \quad (22.48)$$

we can cast Eq. (22.45) into the form [12]

$$\frac{dY}{dx} = -\sqrt{\frac{\pi}{45G}} \frac{g_*^{1/2} m_1}{x^2} \langle \sigma_{\text{eff}} v \rangle (Y^2 - Y_{\text{eq}}^2) \quad (22.49)$$

where Y_{eq} can be written as

$$Y_{\text{eq}} = \frac{n_{\text{eq}}}{s} = \frac{45x^2}{4\pi^4 h_{\text{eff}}(T)} \sum_i g_i \left(\frac{m_i}{m_1} \right)^2 K_2 \left(x \frac{m_i}{m_1} \right), \quad (22.50)$$

using Eqs. (22.15), (22.40) and (22.48).

The parameter $g_*^{1/2}$ is defined as

$$g_*^{1/2} = \frac{h_{\text{eff}}}{\sqrt{g_{\text{eff}}}} \left(1 + \frac{T}{3h_{\text{eff}}} \frac{dh_{\text{eff}}}{dT} \right) \quad (22.51)$$

For g_{eff} , h_{eff} and $g_*^{1/2}$, the user can choose different implementations by a call to the routine `dsrd_set` (see that routine for further details). The default for those effective relativistic degrees of freedom is to use the results from Drees et al [88].

To obtain the relic density we integrate Eq. (22.54) from $x = 0$ to $x_0 = m_\chi/T_0$ where T_0 is the photon temperature of the Universe today. The relic density today in units of the critical density is then given by

$$\Omega_\chi = \rho_\chi^0/\rho_{\text{crit}} = m_\chi s_0 Y_0/\rho_{\text{crit}} \quad (22.52)$$

where $\rho_{\text{crit}} = 3H^2/8\pi G$ is the critical density, s_0 is the entropy density today and Y_0 is the result of the integration of Eq. (22.54). With a background radiation temperature of $T_0 = 2.726$ K we finally obtain

$$\Omega_\chi h^2 = 2.755 \times 10^8 \frac{m_\chi}{\text{GeV}} Y_0. \quad (22.53)$$

It is worth stressing that the entire formalism described above is readily adapted to the case of freeze-out happening in a secluded *dark sector*, with only relatively modest modifications [89]. This option has been fully implemented in DarkSUSY since version 6.3 [90].

22.2 Relic density – numerical integration of the density equation

Let us write the evolution equation for the density,

$$\frac{dY}{dx} = -\sqrt{\frac{\pi}{45G}} \frac{g_*^{1/2} m_1}{x^2} \langle \sigma_{\text{eff}} v \rangle (Y^2 - Y_{\text{eq}}^2) \quad (22.54)$$

as

$$\frac{dY}{dx} = \lambda(Y^2 - q^2), \quad (22.55)$$

where λ contains the annihilation rate and q represents the thermal-equilibrium density.

This equation is stiff and an explicit method, like Euler or Runge-Kutta, fails to converge. To obtain a numerical solution, we use an adaptive implicit trapezoidal method which we explain in the following. Basically we discretize the equation first with a trapezoidal then with an Euler method, and adapt the step size according to the difference in the updated function values.

For simplicity we denote the right hand side of eq. (22.55) as $f(x)$. We further write $f_i = f(x_i)$ and similarly for the other functions $\lambda(x)$ and $q(x)$. Given $Y_i = Y(x_i)$ we find $Y_{i+1} = Y(x_{i+1})$ with $x_{i+1} = x_i + h$ as follows.

First we discretize the evolution equation as

$$Y_{i+1} - Y_i = h \frac{f_i + f_{i+1}}{2}. \quad (22.56)$$

We insert

$$f_i = \lambda_i (Y_i^2 - q_i^2), \quad (22.57)$$

$$f_{i+1} = \lambda_{i+1} (Y_{i+1}^2 - q_{i+1}^2), \quad (22.58)$$

and solve the resulting quadratic equation for Y_{i+1} to obtain

$$Y_{i+1} = \frac{c}{1 + \sqrt{1 + uc}}, \quad (22.59)$$

where

$$c = 2Y_i + u [(q_{i+1}^2 + \rho q_i^2) - \rho Y_i^2], \quad (22.60)$$

$$u = h\lambda_{i+1}, \quad (22.61)$$

$$\rho = \lambda_i/\lambda_{i+1}. \quad (22.62)$$

In the expression for c we have explicitly indicated the order of evaluation which we found avoids round-off errors. If in eq. (22.59) $1 + uc$ is negative, we simply reduce the step h to $h/2$ and try again.

Secondly we discretize the evolution equation as

$$Y_{i+1} - Y_i = hf_{i+1}. \quad (22.63)$$

We insert the expression for f_{i+1} and solve the quadratic equation for Y_{i+1} to obtain

$$Y'_{i+1} = \frac{1}{2} \frac{c'}{1 + \sqrt{1 + uc'}}, \quad (22.64)$$

where

$$c' = 4(Y_i + uq_{i+1}^2). \quad (22.65)$$

Again if in eq. (22.64) $1 + uc' < 0$, we reduce the step h to $h/2$ and try again.

We then adapt the step size according to the relative difference of Y_{i+1} and Y'_{i+1} ,

$$d = \left| \frac{Y_{i+1} - Y'_{i+1}}{Y_{i+1}} \right|. \quad (22.66)$$

If the difference is larger than a prefixed ϵ , set at 0.01, we reduce the step size h to $hs/\sqrt{\epsilon}$ but never to less than $h/10$. s is a safety factor set to 0.9. If $d < \epsilon$, we increase the step size by a factor $s/\sqrt{\epsilon}$ but never by more than a factor of 5. We do not allow the step size to become smaller than $h_{\min} = 10^{-9}$. Error code 5 is reported if this happens. Error code 4 occurs when x_{i+1} is numerically equal to x_i because of round-off. Error code 6 occurs when the number of steps exceeds a maximum of 100000. Finally the initial step size is taken to be 0.01.

22.3 Relic density – asymmetric dark matter

The discussion above is readily generalized to the case where DM is not self-conjugate, and where there may be an asymmetry between the number density of DM particles, n_+ , and the number density of anti-DM particles, n_- . Neglecting oscillations and CP violations, the Boltzmann equations for these number densities are then given by

$$\dot{n}_{\pm} 3Hn_{\pm} = -\langle \sigma v \rangle (n_+ n_- - n_{\text{eq}}^2). \quad (22.67)$$

or, equivalently,

$$\frac{dY_{\pm}}{dx} = -\lambda (Y_+ Y_- - Y_{\text{eq}}^2). \quad (22.68)$$

We can then introduce an asymmetry parameter

$$\eta \equiv Y_+ - Y_- > 0, \quad (22.69)$$

which we assume to be fixed at a primordial stage, and to stay constant during the subsequent cosmological evolution. In particular, this requires *i*) the absence of DM-number violating interactions (such as through Majorana mass terms allowing for oscillations) and *ii*) the absence of entropy production, as in the standard formulation of freeze-out described above. The Boltzmann equation for $Y \equiv Y_-$ thus becomes

$$\frac{dY}{dx} = -\lambda (Y^2 + Y\eta - Y_{\text{eq}}^2). \quad (22.70)$$

Clearly, for $\eta \rightarrow 0$, we recover the Boltzmann equation in the symmetric case.

The *total* DM relic abundance in such a scenario is obtained as

$$\Omega_{\text{DM}} h^2 = c \times (Y_-^{\infty} + Y_+^{\infty}) = c \times (2Y^{\infty} + \eta), \quad (22.71)$$

with

$$c = 2.755 \times 10^{10} \left(\frac{m_{\text{DM}}}{100 \text{ GeV}} \right), \quad (22.72)$$

where Y^∞ is the abundance of the anti-DM component obtained when integrating Eq. (22.70) until today, i.e. formally in the limit of $x \rightarrow \infty$. The symmetric component – of DM (and anti-DM) – thus makes up a fraction

$$r \equiv \frac{2Y^\infty}{2Y^\infty + \eta} = \frac{1}{1 + \eta/(2Y^\infty)} \quad (22.73)$$

$$= 1 - \eta \frac{c}{\Omega_{\text{DM}} h^2} \quad (22.74)$$

of the total DM density today. It is also worth noting that for $m_{\text{DM}} > 4.07 \text{ GeV}/(10^{10} \eta)$ already the asymmetric component would contribute more than the observed DM density, $\Omega_{\text{obs}} h^2 = 0.112$.

The numerical implementation of Eq. (22.70) follows the same idea as in the symmetric ($\eta \rightarrow 0$), i.e. by an implicit trapezoidal method with adaptive stepsize h . Concretely, when going from x_i to $x_{i+1} = x_i + h$, we estimate the abundance $Y_{i+1} \equiv Y(x_{i+1})$ to be

$$Y_{i+1} = Y_i + \frac{h}{2} (Y'_i + Y'_{i+1}) \quad (22.75)$$

$$= \frac{\tilde{C}_i}{2} - \frac{1}{2} \tilde{\lambda}_{i+1} Y_{i+1}^2 - \frac{1}{2} \tilde{\eta}_{i+1} Y_{i+1}, \quad (22.76)$$

where $\tilde{\lambda}_i \equiv h\lambda(x_i)$, $\tilde{\eta}_i \equiv \eta\tilde{\lambda}_i$ and

$$\tilde{C}_i \equiv 2Y_i(1 - \frac{\tilde{\eta}_i}{2}) - \tilde{\lambda}_i (Y_i^2 - Y_{\text{eq},i}^2) + \tilde{\lambda}_{i+1} Y_{\text{eq},i+1}^2. \quad (22.77)$$

This constitutes a quadratic equation for Y_{i+1} , solved by

$$Y_{i+1} = \frac{\tilde{C}_i / [1 + \tilde{\eta}_{i+1}/2]}{1 + \sqrt{1 + \tilde{\lambda}_{i+1} \tilde{C}_i / [1 + \tilde{\eta}_{i+1}/2]^2}}. \quad (22.78)$$

In order to estimate the local relative error we also calculate the abundance at x_{i+1} with a modified Euler method. This alternative estimate we denote by a small y_{i+1} :

$$y_{i+1} \equiv Y_i + hY'_{i+1} \quad (22.79)$$

$$= Y_i + \tilde{\lambda}_{i+1} Y_{\text{eq},i+1}^2 - \tilde{\lambda}_{i+1} y_{i+1}^2 - \tilde{\eta}_{i+1} y_{i+1}, \quad (22.80)$$

which we solve for

$$y_{i+1} = \frac{\tilde{c}_i / (2[1 + \tilde{\eta}_{i+1}])}{1 + \sqrt{1 + \tilde{\lambda}_{i+1} \tilde{c}_i / [1 + \tilde{\eta}_{i+1}]^2}}, \quad (22.81)$$

with

$$\tilde{c}_i \equiv 4Y_i + 4\tilde{\lambda}_{i+1} Y_{\text{eq},i+1}^2. \quad (22.82)$$

When solving the Boltzmann equation (22.70), as implemented in Eq. (22.78), finally, we adaptively decrease the stepsize h if

$$(Y_{i+1} - y_{i+1}) / Y_{i+1} \quad (22.83)$$

exceeds a given tolerance (stored in the common block variable `compeps`). The initial stepsize for h , finally, is stored in the common block variable `hstep`.

22.4 Relic density – coupled Boltzmann equations

22.4.1 Boltzmann equation at phase-space level

Everything discussed so far rests on the assumption that the DM particles are in full *kinetic* equilibrium during the freeze-out process. If that is not the case, one must in principle directly solve the full Boltzmann equation at the phase-space level. Not taking into account coannihilations, it reads [91, 92]

$$(\partial_t - Hp\partial_p) f_\chi = \frac{1}{E} (C_{\text{ann}}[f_\chi] + C_{\text{el}}[f_\chi]) , \quad (22.84)$$

where

$$\begin{aligned} C_{\text{ann}} &= \frac{1}{2g_\chi} \int \frac{d^3\tilde{p}}{(2\pi)^3 2\tilde{E}} \int \frac{d^3k}{(2\pi)^3 2\omega} \int \frac{d^3\tilde{k}}{(2\pi)^3 2\tilde{\omega}} \\ &\times (2\pi)^4 \delta^{(4)}(\tilde{p} + p - \tilde{k} - k) \\ &\times \left[|\mathcal{M}|_{\tilde{\chi}\chi \leftrightarrow \tilde{f}f}^2 g(\omega)g(\tilde{\omega}) - |\mathcal{M}|_{\tilde{\chi}\chi \rightarrow \tilde{f}f}^2 f_\chi(E)f_\chi(\tilde{E}) \right] \end{aligned} \quad (22.85)$$

describes the effect of two-body annihilations, and the collision term for elastic scattering processes is given by

$$\begin{aligned} C_{\text{el}} &= \frac{1}{2g_\chi} \int \frac{d^3k}{(2\pi)^3 2\omega} \int \frac{d^3\tilde{k}}{(2\pi)^3 2\tilde{\omega}} \int \frac{d^3\tilde{p}}{(2\pi)^3 2\tilde{E}} \\ &\times (2\pi)^4 \delta^{(4)}(\tilde{p} + \tilde{k} - p - k) |\mathcal{M}|_{\chi f \leftrightarrow \chi f}^2 \\ &\times \left[(1 \mp g^\pm(\omega)) g^\pm(\tilde{\omega}) f_\chi(\tilde{E}) - (\omega \leftrightarrow \tilde{\omega}, E \leftrightarrow \tilde{E}) \right] . \end{aligned} \quad (22.86)$$

Here, $f_\chi(t, p)$ is the DM phase-space density, and both collision terms and the squared amplitudes $|\mathcal{M}|^2$ for the respective process are implicitly summed over all heat bath particles f , and final *and* initial state internal degrees of freedom, respectively. For the heat bath particles, the phase-space distribution is given by $g^\pm(\omega) = 1/[\exp(\omega/T) \pm 1]$.

DRAKE [15] is at the time of this writing (pre-release of DarkSUSY 6.4) the only numerical tool that directly solves Eq. (22.84), which comes with its own challenges. In DarkSUSY we follow a different approach, namely considering (momentum) *moments* of Eq. (22.84). In particular, if f_χ follows a Maxwell-Boltzmann distribution, integration over $\int d^3p$ directly results in the Boltzmann equation for the number density discussed previously, Eq. (22.12). Otherwise, one needs to at least consider the second moment of the DM distribution as well, leading to a set of Boltzmann equations that couple the evolution of the DM number density and velocity dispersion, respectively. This method was first introduced in Ref. [93], and later refined in Ref. [78].[†]

22.4.2 Boltzmann equations for number density and velocity dispersion

Concretely, we obtain two coupled equations by (i) directly integrating Eq. (22.84) and (ii) first multiplying by p^2 before doing so. One can then introduce the (non-thermal) DM velocity dispersion as a ‘DM temperature parameter’,

$$T_\chi \equiv \frac{g_\chi}{3n_\chi} \int \frac{d^3p}{(2\pi)^3} \frac{p^2}{E} f_\chi , \quad (22.87)$$

[†]For the case of an interacting dark sector that is in local thermal equilibrium with itself (but with a temperature T_χ different from that of the SM heat bath), this results in an *exact* description of the DM density evolution. However, this method still provides a very reasonable approximation to the full result (at the phase-space density level) even in many other case, like e.g. in the case of Scalar Singlet DM near the Higgs resonance [78].

as well as

$$y \equiv m_\chi T_\chi s^{-2/3}. \quad (22.88)$$

The two coupled Boltzmann equations then take the form

$$\frac{x}{Y} \frac{dY}{dx} = \frac{sY}{\tilde{H}} \left[\frac{Y_{\text{eq}}^2}{Y^2} \langle \sigma v \rangle_T - \langle \sigma v \rangle_{T_\chi} \right], \quad (22.89)$$

$$\frac{x}{y} \frac{dy}{dx} = \frac{\gamma w}{\tilde{H}} \left[\frac{y_{\text{eq}}}{y} - 1 \right] + \frac{sY}{\tilde{H}} \left[\langle \sigma v \rangle_{T_\chi} - \langle \sigma v \rangle_{2,T_\chi} \right] + \frac{sY}{\tilde{H}} \frac{Y_{\text{eq}}^2}{Y^2} \left[\frac{y_{\text{eq}}}{y} \langle \sigma v \rangle_{2,T} - \langle \sigma v \rangle_T \right] + 2(1-w) \frac{H}{\tilde{H}}. \quad (22.90)$$

Here, a subscript T or T_χ indicates the temperature at which to take thermal averages. Introducing $s \equiv E_{\text{CM}}^2/(4m_\chi^2)$, these can be expressed as

$$\langle \sigma v \rangle_T = \int_1^\infty d\tilde{s} \sigma_{\tilde{\chi}\chi \rightarrow \tilde{f}f} v_{\text{lab}} \frac{2x\sqrt{\tilde{s}-1}(2\tilde{s}-1)K_1(2\sqrt{\tilde{s}}x)}{K_2^2(x)}, \quad (22.91)$$

which is the same thermal average as for standard freeze-out, and

$$\langle \sigma v \rangle_{2,T} = \frac{g_\chi^2}{T n_{\chi,\text{eq}}^2} \int \frac{d^3p d^3\tilde{p}}{(2\pi)^6} \frac{p^2}{3E} \sigma_{\tilde{\chi}\chi \rightarrow \tilde{f}f} f_{\chi,\text{eq}}(\mathbf{p}) f_{\chi,\text{eq}}(\tilde{\mathbf{p}}) \quad (22.92)$$

$$= \int_1^\infty d\tilde{s} \sigma_{\tilde{\chi}\chi \rightarrow \tilde{f}f} v_{\text{lab}} \frac{2x\sqrt{\tilde{s}-1}(2\tilde{s}-1)K_1(2\sqrt{\tilde{s}}x)}{K_2^2(x)} \times F_2(\tilde{s}, x) \quad (22.93)$$

with

$$F_2(\tilde{s}, x) \equiv \frac{2\tilde{s}x^2 e^{-2\sqrt{\tilde{s}}x}}{3\sqrt{\tilde{s}-1}K_1(2\sqrt{\tilde{s}}x)} \quad (22.94)$$

$$\times \int_1^\infty d\epsilon_+ e^{-2\sqrt{\tilde{s}}x(\epsilon_+-1)} \left[\epsilon_+ \sqrt{(\tilde{s}-1)(\epsilon_+^2-1)} + \frac{1}{2\sqrt{\tilde{s}}} \log \left(\frac{\sqrt{\tilde{s}}\epsilon_+ - \sqrt{(\tilde{s}-1)(\epsilon_+^2-1)}}{\sqrt{\tilde{s}}\epsilon_+ + \sqrt{(\tilde{s}-1)(\epsilon_+^2-1)}} \right) \right],$$

which is a p^2 -weighted version of it.

In Eqs. (22.89,22.90), we also introduced $\tilde{H} \equiv H/[1 + \tilde{g}(x)]$ with $\tilde{g} \equiv \frac{1}{3} \frac{T}{g_{\text{eff}}} \frac{dg_{\text{eff}}}{dT}$ and

$$w \equiv 1 - \frac{\langle p^4/E^3 \rangle}{6T_\chi} = 1 - \frac{g_\chi}{6T_\chi n_{\chi,\text{eq}}} \int \frac{d^3p}{(2\pi^3)} \frac{p^4}{E^3} f_{\chi,\text{eq}}. \quad (22.95)$$

Finally, γ is the momentum transfer rate – see section 21.1 and in particular Eq. (21.7).

Numerically integrating the coupled system of Eqs. (22.89,22.90) one can now obtain the DM abundance today, Y_0 , and from this as usual the density $\Omega_\chi h^2$ from Eq. (22.53). The example program `examples/aux/oh2_cBE_ScalarSinglet.f` demonstrates how to do this with DarkSUSY for the case of ScalarSinglet DM (as opposed to `examples/aux/oh2_ScalarSinglet.f`, which demonstrates the same when following the standard approach based on Eq. (22.44)). More details about the numerical implementation of the underlying routines are given further down.

22.5 Relic density – tabulation of W_{eff}

One of the most CPU-intensive tasks when calculating the relic density is the evaluation of the annihilation cross section, i.e. the effective annihilation rate W_{eff} in Eq. (22.34). W_{eff} enters in the

calculation of the thermally averaged cross section, Eq. (22.39), which then in turn enters when solving the Boltzmann equation. As we see in Eqs. (22.34) and (22.39), W_{eff} does not in itself depend on the temperature, just on the effective momentum p_{eff} as defined in Eq. (22.33). The temperature dependence enters in Eq. (22.39).

We can use the effective annihilation rate as provided by the particle module to calculate the thermal average in each step of the Boltzmann equation solving, but this will be very inefficient (unless the effective annihilation rate is a very simple expression, which it typically is not). Instead we tabulate W_{eff} before solving the Boltzmann equation and interpolate in this table during the Boltzmann equation solving. A crucial step in this tabulation is to use as few points as possible, while still having enough points to make the interpolated table accurate enough. For this purpose, DarkSUSY has two main ways of tabulating W_{eff} . We will first go through the old standard way of tabulating, and then describe a new improved method. The new method is as of DarkSUSY 6.4 the default.

The different methods are chosen with the flag `fast` when calling `dsrdomega`.

22.5.1 Method A: pre-tabulation of W_{eff}

In Fig. 22.1 we see an example of an effective annihilation rate. However, we do not need all parts of this function to be well tabulated, as can be seen in Fig. 22.2, where the Boltzmann suppression is included. The function in Fig. 22.2 is essentially the function that is integrated to get the thermally averaged cross section $\langle\sigma_{\text{eff}}v\rangle$. As is clear from these figures is that it is most important to have a good tabulation for lower p_{eff} than for really high ones. With this in mind, the pre-tabulation method tabulates W_{eff} in the following way:

1. A set of points are added between $p_{\text{eff}} = 0$ and typically $p_{\text{eff}} = 2m_{\text{WIMP}}$.
2. A set of points is added (sparser) from $p_{\text{eff}} = 2m_{\text{WIMP}}$ up to p_{max} , where p_{max} is typically of the order of $10m_{\text{WIMP}}$.
3. A set of points are added at each resonance
4. A set of points are added at each threshold, both final state thresholds and coannihilation thresholds.
5. Optionally, a test is performed on a normalized version of the $W_{\text{eff}} - p_{\text{eff}}$ plane of tabulated points to see if the angle between lines connection the adjacent tabulated points is large. If it is, new points are added.
6. A spline interpolation is set up. If the difference between the spline interpolation and a linear interpolation is too large (checked midway in between tabulated points), a point is added.

Finally, a spline interpolation is used to interpolate in the table. When the actual thermal average is calculated, the integration is split at thresholds and around resonances to make sure that these are well integrated.

`fast=0,1,2,3,9` all use method A (without the angle test 5 above), with decreasing accuracy (and increasingly faster calculation).

Up until DarkSUSY 6.3, the default was `fast=1`.

22.5.2 Method B: Breit-Wigner fit to resonances and tabulation on the fly

In this method, a completely different approach is used to tabulate W_{eff} . First, W_{eff} is calculated on and near resonances to see if they can be well-fit with a Breit-Wigner form on a linear curve. We thus check if the following function is a good fit to W_{eff} on and near the resonance:

$$W_{\text{BW}} = a + f_{\text{BW}}(p_{\text{eff}}) \quad (22.96)$$

$$f_{\text{BW}} = \frac{k}{(E^2 - m_{\text{res}}^2)^2 + m_{\text{res}}^2 \Gamma_{\text{res}}^2} p_{\text{eff}}^\alpha \quad ; \quad E = 2\sqrt{m_{\text{WIMP}}^2 + p_{\text{eff}}^2} \quad (22.97)$$

In this expression, a , k and α are constants to be fit, whereas m_{res} and Γ_{res} are given from the known properties of the resonance, i.e. they are not fit.

If the fit is good, the analytical form above is used instead of the actual W_{eff} expression on and around the resonance (how far away it is used depends on how far away the fit is good). Note that to get good fits we include both a constant term, a linear term and a correction factor p_{eff}^α . On this form, the Breit-Wigner fit is usually quite good many Γ_{res} away from the peak.

Technically, the fit is done by calculating W_{eff} on peak and `ngastart` times the resonance width away from the peak. If the difference in W_{eff} on and off peak is larger than `wresratio`, the resonance is deemed important and we fit a , k and α . We then check that the fit is better than `wdev`. If it is, we go 50% further away from the peak and check if the fit is still better than `wdev` compared to the full W_{eff} calculation. If it is, we continue like this until we have found out how far away from the peak the fit is good.

Secondly, a few points are optionally added near thresholds. Finally, the Boltzmann solver and thermal average integration call a wrapper routine to the full W_{eff} expression that

1. Check if W_{eff} is requested for a p_{eff} where we have a good Breit-Wigner fit. If we do, that fit is used to calculate W_{eff} .
2. Check if we already have calculated W_{eff} close to the requested p_{eff} . If we have, we use either a linear or quadratic interpolation to interpolate between previously calculated W_{eff} points. If we do not have a calculation for a nearby p_{eff} , W_{eff} will be evaluated and added to our table of calculated values. The criterion for when to add new points is different if we are close to resonances or thresholds.

Hence, this way, we add points on the fly where the thermal average integration routine is actually asking for W_{eff} values. Compare this to method A where we try to foresee where values are needed. Also with this method, the integration of the thermal average is split around resonances and at thresholds to make sure these are well-handled.

`fast=20,21` use this method with decreasing accuracy (and increased speed). As of DarkSUSY 6.4, the default is to use `fast=20`. Compared to the previous default `fast=1`, the new method is typically both faster and more accurate, especially for complex W_{eff} .

22.5.3 Method C: full expression of W_{eff}

This is not really a tabulation method, but we include it here for completeness. There is also an option to skip tabulation completely and instead use the full W_{eff} expression when solving the Boltzmann equation and calculating the thermally averaged annihilation cross section. This option is included mostly for checks and debugging, as it is typically extremely slow. It can however be used when W_{eff} is fast to calculate and tabulations are not needed to speed things up.

22.5.4 Method D: quick and dirty

This method uses an old way of calculating the relic density by approximating the annihilation cross section as

$$\sigma v = a + bv^2$$

where a and b are constants and v is the relative velocity. The method finds a and b from evaluating W_{eff} at $p_{\text{eff}} = 0$ and $p_{\text{eff}} = 0.25m_{\text{WIMP}}$ and extracts a and b from these values. It then calculates the relic density by an approximate analytic solution to the Boltzmann equation, following Ref. [92]. The method should not really be used, and is included only for comparison. The errors from this way of calculating the relic density can be large, several orders of magnitude if there are important thresholds, coannihilations or resonances.

22.5.5 Comparison of methods

With a pre-release of DarkSUSY 6.4, we calculated the relic density for the long set of 19 MSSM models in `dstest_mssm`. As reference results, we used calculations with `fast=99`. Compared to this reference, the errors for the different methods are shown in Table 22.1. As can be seen, `fast=20` is both faster and more accurate than both `fast=0` and `fast=1`.

Method	fast	Average error	Max error	Approximate CPU time
C	99	–	–	$> 10^6$ s
A	0	0.024%	–0.150%	135 s
A	1	0.21%	1.47%	131 s
B	20	0.010%	0.025%	61 s
B	21	0.037%	0.135%	58 s

Table 22.1: Table of comparison of relic density calculations with different `fast` settings. These were run on a MacBook Pro with an M1 Pro processor, using a pre-release of DarkSUSY 6.4. The CPU-times are approximate.

22.5.6 Time constraints

When scanning large parameter spaces for complex particle physics models, some models can be significantly slower to calculate the relic density for than others. This happens when e.g. there are many coannihilating particles and lots of resonances and thresholds. In these cases, it is sometimes preferable to put a cap on how long a certain calculation can take. For both method A and B above, one can set an approximate maximum number of CPU seconds one is willing to spend on the relic density calculation. This is done by setting the global variable `rdt_max` to the maximum number of seconds for the calculation. This is then used by the code as an approximate upper limit, in practice it spends a little bit longer to wrap things up. For method A, if the upper limit is reached during tabulation of W_{eff} , the tabulation is terminated the the relic density routine returns $\Omega h^2 = 0$. For method B, if the upper limit is reached, it will just stop adding more points to the W_{eff} table, but it will proceed with interpolating the table it has. Hence, in this case, one would still get a result, but maybe not as one expects. A warning flag is issued in this case to indicate that a time cut-off has been applied. The default is to not use any time constraint at all.

22.6 Relic density – routines

In `src/rd`, the general relic density routines are found. These routines can be used for any dark matter candidate and e.g. the interface to neutralino dark matter is in `src_models/mssm/rd`. The initialization of these routines happens via a call to `dsrd_init` (this is typically done automatically, from `dsinit`); `dsrd_set` can be used to steer general global behaviour, in particular which parameterization of the relativistic plasma (standard model) degrees of freedom that should be used.

The main routine for relic density calculations is `dsrdomega`. It calculates and returns the relic density as well as the approximate temperature of freeze-out. It takes a few arguments: i) one `option` argument that typically (depending on the particle physics module) determines how co-annihilations are included, ii) one `fast` argument that determines which tabulation method that should be used and how careful the Boltzmann solver should be in calculating the relic density. See Section 22.5 and the header of `dsrdomega` for more details.

In full analogy to `dsrdomega` for the symmetric case, `dsrdomega_aDM` calculates the relic density in the presence of a primordial asymmetry. Compared to `dsrdomega`, this function takes one additional input parameter η . It also has one additional output parameter, r , that describes the

late-time fraction of the symmetric DM component as introduced in Eq. (22.73). Note that η is a free input parameter here that can in principle be chosen differently from the asymmetry parameter η_{model} used to set up a model with DM particles that is not self-conjugate (see, e.g., the generic WIMP example in chapter 32) – though this is discouraged. The value of r returned by `dsrdomega_aDM` is always consistent with the input value of η , i.e. the result of a relic density calculation as given by Eq. (22.73).

(Indirect detection yields, on the other hand, will per default use the expression in Eq. (22.74) to define r , under the assumption that Ω_{DM} corresponds to all of the observed DM. This behaviour can be changed by using the rescaling function `dscrrescale_asym` described in section 9.1.)

22.6.1 Global parameters

All internal settings of the relic density routines are set in common blocks in `dsrdcom.h`. The most important parameters that can be changed are given below. If changing these, be aware though that many of these are set in `dsrdomega` when it is called with a given `fast` setting (see `dsrdomega` for details).

Most important general parameters in `dsrdcom.h`

Purpose: Provide a set of parameters, with which the internal behaviour of the relic density routines can be changed.

Parameters

<code>tharsi</code>	<code>i</code>	Size of the coannihilation, resonance and threshold arrays (default=50). Increase this size if you have more than 50 coannihilating particles, more than 50 resonances or more than 50 thresholds.
<code>rduerr</code>	<code>i</code>	Logical unit number where error messages are printed.
<code>rdtag</code>	<code>c*12</code>	Idtag that is printed in case of errors.
<code>hstep</code>	<code>r8</code>	Initial step size in h for Boltzmann solver
<code>hmin</code>	<code>r8</code>	Minimum allowed step size in h in Boltzmann solver
<code>compeps</code>	<code>r8</code>	Relative accuracy of Boltzmann solver
<code>xinit</code>	<code>r8</code>	Initial value of x for Boltzmann solver
<code>xfinal</code>	<code>r8</code>	Final value of x for Boltzmann solver
<code>umax</code>	<code>r8</code>	Maximum value of u for $\langle\sigma v\rangle$ integration.
<code>theps</code>	<code>r8</code>	Relative accuracy in thermal average integration in <code>dsrdthav</code> .
<code>thepshi</code>	<code>r8</code>	Relative accuracy in thermal average integration for high x in <code>dsrdthav</code> .
<code>ptopr</code>	<code>r8</code>	Cut-off upper limit in p_{eff} for thermal average integration. Unit: m_{WIMP}
<code>pmxt</code>	<code>r8</code>	Maximum momentum, only used for high temperature region (freeze-in).
<code>rdt_max</code>	<code>r8</code>	Maximum number of seconds to spend on relic density calculation. If the time limit is exceeded, <code>dsrdomega</code> returns with an error flag and the result 0 for method A, <code>fast=0,1,2,3,9</code> and an approximate result for method B, <code>fast=20,21</code> . The time is the total CPU time (i.e. summed up over all cores/threads) and the limit is approximate as it is only checked before a new point is added to the W_{eff} tabulation.
<code>rd_excl_th_int</code>	<code>b</code>	exclude (or not) integration over threshold (just a tiny region, approximated with linear W_{eff})

Most important parameters for method A in `dsrdcom.h`

Purpose: Provide a set of parameters, for method A, `fast=0,1,2,3,9`.

Parameters

<code>waccd</code>	<code>r8</code>	Allowed difference between linear and spline interpolation. If difference is larger than this, a new point is inserted.
<code>dpminr</code>	<code>r8</code>	Minimum allowed difference in p_{eff} between to points in $W_{\text{eff}} - p_{\text{eff}}$ -table. Points are not inserted closer to each other than this. Unit: m_{WIMP} .
<code>dpthr</code>	<code>r8</code>	Scale for point insertion around thresholds. Unit: m_{WIMP} .

wdiffr	r8	Relative difference in W_{eff} on and close to a resonance to deem it important.
wdiff	r8	Relative difference in W_{eff} below and above threshold to deem it important.
pdivr	r8	Division between low and high p_{eff} region. Unit: m_{WIMP} .
dpres	r8	Scale for point insertion around resonances. Unit: Γ_{res} .
cosmin	r8	maximum $\cos\theta$ -difference between adjacent lines in normalized $W_{\text{eff}}-p_{\text{eff}}$ -table. If difference is larger than this, a point is added, if <code>cthstest=1</code> .
cthstest	i	switch to determine if $\cos\theta$ -check should be performed (1) or not (0).
spltest	i	Switch to determine if spline test should be performed (1) or not (0).
nlow	i	Number of points to add in low p_{eff} region
nhigh	i	Number of points to add in high p_{eff} region
npres	i	Number of points to add on each side of a resonance (if important).
nthup	i	Number of points to add just above a threshold.

Most important parameters for method B in `dsrdcom.h`

<i>Purpose:</i>	Provide a set of parameters, for method B, <code>fast=20,21</code> .	
<i>Parameters</i>		
dpminr	r8	Minimum allowed difference in p_{eff} between to points in $W_{\text{eff}}-p_{\text{eff}}$ -table. Points are not inserted closer to each other than this. Unit: m_{WIMP} .
dpthr	r8	Scale for point insertion around thresholds. Unit: m_{WIMP} .
rdquadrerr	r8	Parameter that determines if quadratic interpolation is expected to be good enough so that we do not need to add a new point.
rdlinerr	r8	Parameter that determines if linear interpolation is expected to be good enough so that we do not need to add a new point.
wresratio	r8	Ratio of on/off peak to treat resonance as important (off-peak means <code>ngastart</code> number of widths from peak).
wdev	r8	Maximum deviation from Breit-Wigner fit to calculation to use fit
ngammamax	r8	Maximum number of widths of resonance to include in fit.
ngastart	r8	Number of widths from resonance to start checking Breit-Wigner fit.
almin	r8	Minimum α allowed in Breit-Wigner fit.
almax	r8	Maximum α allowed in Breit-Wigner fit.
ngafac	r8	Factor to increase <code>nga</code> (number of widths) with in each step when checking a resonance.

22.6.2 Brief description of internal routines for standard Boltzmann approach

Below, the remaining routines related to relic density calculation in the standard approach (based on the Boltzmann equation for the number density) are briefly mentioned. For more details, we refer to the routines themselves.

Routine	Purpose
dsrdomega	Main routine to calculate the relic density.
dsrdens	Routine to solve the Boltzmann equation and return the relic density. Called by <code>dsrdomega</code> .
dsrdaddpt	To add one point in the $W_{\text{eff}}-p_{\text{eff}}$ table, method A.
dsrdbreit_wigner	Returns the Breit-Wigner function for given input parameters.
dsrdbw_get	Get W_{eff} from Breit-Wigner fit if we are on a well-fit resonance, method B.
dsrdbw_setup	Setup Breit-Wigner fits, method B.
dsrddeltaneff	ΔN_{eff} (additional number of effective neutrino degrees of freedom)
dsrddof	Returns the SM degrees of freedom as a function of the temperature in the early Universe.

dsrddofDS	Returns potential dark sector degrees of freedom as a function of the temperature in the early Universe.
dsrddpmin	To return the allowed minimal distance in p_{eff} between two points in the $W_{\text{eff}}-p_{\text{eff}}$ plane. The returned value depends on if there is a resonance present or not at the given p_{eff} .
dsrdeqn	To solve the relic density equation by means of an implicit trapezoidal method with adaptive stepsize and termination.
dsrdequil	returns entropy and equilibrium abundance of DM particles
dsrdfunc	To return the invariant annihilation rate times the thermal distribution.
dsrdfuncs	To provide dsrdfunc in a form suitable for numerical integration.
dsrdHubble	Hubble rate (changing this allows to study the effect of modified expansion rates <i>during freeze-out</i> on the relic density)
dsrdlny	To return $\ln W_{\text{eff}}$ for a given p_{eff} .
dsrdnormlz	To return a unit vector in a given direction.
dsrdqad	To calculate the relic density with a quick-and-dirty method. It uses the approximative expressions in Kolb & Turner with the cross section expanded in v^2 .
dsrdqrkck	To numerically integrate a function with a Runge-Kutta method
dsrdquad	Auxiliary function to perform quadratic interpolation
dsrdreaddof	Reads relevant tables for relativistic d.o.f. (as controlled by a previous call to ds_set).
dsdrhs	To calculate terms on the right-hand side in the Boltzmann equation.
dsrdsetdefaults	(re-)sets default values of some common block variables in dsrdcom.h that might get changed by other routines in /rd or /fi
dsrdsingledof	Effective energy d.o.f. for a semi-relativistic particle
dsrdspline	To set up the table $W_{\text{eff}}-p_{\text{eff}}$ for spline interpolation, method A.
dsrdstart	To sort and store information about coannihilations, resonances and thresholds in common blocks.
dsrdstate	saves or resets the common blocks in dsrdcom.h to previously stored values
dsrdtab	To set up the table $W_{\text{eff}}-p_{\text{eff}}$, method A.
dsrdthav	To calculate the thermally averaged annihilation cross section $\langle\sigma v\rangle$ at a given temperature.
dsrdthclose	returns the location of the threshold closest to a given momentum p ...
dsrdthlim	To determine the end-points for the thermal average integration.
dsrdthtest	To check if a given entry in the $W_{\text{eff}}-p_{\text{eff}}$ table is at a threshold.
dsrdwfunc	To write out dsrdfunc for a given $x = m_\chi/T$.
dsrdwintp	To return the invariant rate W_{eff} for any given p_{eff} by performing a spline interpolation in the $W_{\text{eff}}-p_{\text{eff}}$ table, method A.
dsrdwintpch	To check the spline interpolation in the $W_{\text{eff}}-p_{\text{eff}}$ table and compare with a linear interpolation.
dsrdwintprint	Print out a the table of invariant rate with the points that are tabulated.
dsrdwintrp	To write out a table of the invariant rate W_{eff} and some internal integration variables and expressions.
dsrdwprint	Print out a the table of invariant rate within a given interval
dsrdwres	To write out the table $W_{\text{eff}}-p_{\text{eff}}$.
dsrdtabth	Add a few points at thresholds, method B.

dsrdwx	To return the invariant rate W_{eff} for any given p_{eff} by either using Breit-Wigner fits, making an interpolation, or adding points on the fly to the $W_{\text{eff}} - p_{\text{eff}}$ -table. Method B.
dsrdxi	temperature ratio $\xi = T_\chi/T$

22.6.3 Implementation of coupled Boltzmann equations

In analogy to `dsrdomega` for the standard approach of tracking only the DM number density, DarkSUSY also provides a function `dsrdomega_cBE` that calculates the relic density based on the coupled system of Boltzmann equations (22.89,22.90). This function takes the same input parameters as `dsrdomega`, but has additional output parameters containing the value of x up to which the Boltzmann equations were adaptively integrated (because Y would remain constant afterwards) as well as the value of y at this maximal x (from which the time of kinetic decoupling can be inferred, cf. section 21.1).

The actual integration of the Boltzmann equations is performed by `dsrdens_cBE` in two steps, or phases, using the SFODE algorithm from the NSWC mathematical library (optimized for n -dimensional, stiff ODEs). During the first phase, where the momentum exchange rate is much higher than the Hubble rate (i.e. long before kinetic decoupling), the problem is simplified to the standard 1D case; the right-hand side of the Boltzmann equations – as returned by `dsdrrhs_cBE` – is thus only given by the first of the two equations, Eq. (22.89), with $T_\chi = T$. Only in a second phase, once γ is no longer much smaller than H , `dsdrrhs_cBE` returns the full r.h.s. of Eqs. (22.89,22.90).

Below, the remaining routines exclusively related to relic density calculations based on coupled Boltzmann equations are briefly mentioned. For more details, we refer to the routines themselves.

Routine	Purpose
dsrdp4E3av	Computes $\langle p^4/E^3 \rangle$.
dsrdthav_2	To calculate the thermally averaged annihilation cross section $\langle \sigma v \rangle_2$ at a given temperature.
dsrdthav_select	Returns all thermal averages appearing in coupled Boltzmann equations; based on the interpolation of pre-tabulated values.
dsrdthav_tab	Tabulates the various thermal averages appearing in coupled Boltzmann equations

Chapter 23

`se_aux`:

Auxiliary routines for WIMP annihilation in the Sun/Earth

23.1 Sun and Earth models – auxiliary routines

This directory contains parameterizations and semi-analytical calculations of auxiliary fluxes, like Earth atmospheric fluxes. They are not up to date with the latest calculations and are provided as is for convenience.

Chapter 24

se_mod: Sun and Earth models

24.1 Sun and Earth models – theory

We need to have accurate models for the Sun and the Earth, both in terms of mass distributions, but also in terms chemical compositions. We need these to be able to calculate the capture rate of WIMPs in the Sun and the Earth accurately. These very same models are also used by external codes like **WimpSim** which simulates annihilations in the Earth and Sun (with the help of event generators) and subsequent neutrino interactions and oscillations on the way out of the Sun/Earth. Those simulation results are included as tables in `se_yield`.

In DarkSUSY we include several solar models, some of which are very detailed regarding chemical composition. Note that these details enable us to perform very detailed capture rate calculations on all relevant elements in the Sun. The solar models included are

- The BS05(OP) model with heavier elements from Grevesse and Sauval.
- The BS05(AGS,OP) model with updated heavier element fractions.
- The AGSS09 model (default), heavier element estimates from meteorites.
- The AGSS09 model with heavier elements from photospheric measurements.
- The AGS05 model.
- The GS98 model.

See `dssem_sunset.f` for details and references. The reason we have several different options for the heavier elements is that the capture rate in the Sun quite depends on these abundances, even if they are small. The reason being that the capture rate approximately goes as A^4 where A is the atomic number.

For the Earth, the model used is a standard reference model.

24.2 Sun and Earth models – routines

The main routines that could be of interest to most users are

Routine	Purpose
<code>dssem_sundens</code>	Density in the Sun.
<code>dssem_sundenscomp</code>	Composition in the Sun.

dssem_sunset	Routine to set which solar model to use.
dssem_sunmass	Mass of Sun within a given radius.
dssem_sunfrac	Mass fraction of a given element at a given radius in the Sun.
dssem_sunpot	Potential in Sun at given radius.
dssem_earthdens	Density in the Earth.
dssem_earthmass	Mass of Earth within a given radius.
dssem_earthpot	Potential in Earth at given radius.

There are more routines available, see `src/se_mod` for details.

Chapter 25

se_nu:

Capture and annihilation in the Sun/Earth

25.1 Neutrinos from the Sun and Earth – theory

Note. This chapter is not up to date with the latest additions to the code, but provided as is as a reference right now.

There are several indirect methods for detection of WIMPs. One of the most promising [94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 19] is to make use of the fact that scattering of halo neutralinos by the Sun and the planets, in particular the Earth, during the several billion years that the Solar system has existed, will have trapped these neutralinos within these astrophysical bodies. Being trapped within the Solar or terrestrial material, they will sink towards the center, where a considerable enrichment and corresponding increase of annihilation rate will occur.

Searches for neutralino annihilation into neutrinos will be subject to extensive experimental investigations in view of the new neutrino telescopes (AMANDA, IceCube, Baikal, NESTOR, ANTARES) planned or under construction [109]. A high-energy neutrino signal in the direction of the centre of the Sun or Earth is an excellent experimental signature which may stand up against the background of neutrinos generated by cosmic-ray interactions in the Earth's atmosphere.

There are several different approximations one could do, or processes to include when calculating the capture rates in the Earth/Sun and many of these are coded into DarkSUSY. The default in DarkSUSY is always to use the best calculations available, but more approximate (older) routines are also available, as well as more speculative signals, like the Damour-Krauss signal (not included by default). If you want to use something else than the defaults, or want to call more internal routines (more internal than dsnrates or dsntdiffrates), you should read the following sections carefully.

25.1.1 Neutrino yield from annihilations

The differential neutrino flux from neutralino annihilation is

$$\frac{dN_\nu}{dE_\nu} = \frac{\Gamma_A}{4\pi D^2} \sum_f B_\chi^f \frac{dN_\nu^f}{dE_\nu} \quad (25.1)$$

where Γ_A is the annihilation rate, D is the distance of the detector from the source (the central region of the Earth or the Sun), f is the neutralino pair annihilation final states, and B_χ^f are the

branching ratios into the final state f . dN_ν^f/dE_ν are the energy distributions of neutrinos generated by the final state f and are obtained from the PYTHIA simulations described in section ??.

In comparison with calculations using the results of [110] (e.g. [111, 104, 112, 107, 106, 113, 114]), this Monte Carlo treatment of the neutrino propagation through the Sun does not need the simplifying assumptions previously made, namely neutral currents are no more assumed to be much weaker than charged currents and energy loss is no more considered continuous.

The neutrino-induced muon flux may be detected in a neutrino telescope by measuring the muons that come from the direction of the centre of the Sun or Earth. For a shallow detector, this usually has to be done in the case of the Sun by looking (as always the case for the Earth) at upward-going muons, since there is a huge background of downward-going muons created by cosmic-ray interactions in the atmosphere. There is always in addition a more isotropic background coming from muon neutrinos created on the other side of the Earth in such cosmic-ray events (and also from cosmic-ray interactions in the outer regions of the Sun). The flux of muons at the detector is given by

$$\frac{dN_\mu}{dE_\mu} = N_A \int_{E_\mu^{\text{th}}}^\infty dE_\nu \int_0^\infty d\lambda \int_{E_\mu}^{E_\nu} dE'_\mu P(E_\mu, E'_\mu; \lambda) \frac{d\sigma_\nu(E_\nu, E'_\mu)}{dE'_\mu} \frac{dN_\nu}{dE_\nu}, \quad (25.2)$$

where λ is the muon range in the medium (ice or water for the large detectors in the ocean or at the South Pole, or rock which surrounds the smaller underground detectors), $d\sigma_\nu(E_\nu, E'_\mu)/dE'_\mu$ is the weak interaction cross section for production of a muon of energy E'_μ from a parent neutrino of energy E_ν , and $P(E_\mu, E'_\mu; \lambda)$ is the probability for a muon of initial energy E'_μ to have a final energy E_μ after passing a path-length λ inside the detector medium. E_μ^{th} is the detector threshold energy, which for “small” neutrino telescopes like Baksan, MACRO and Super-Kamiokande is around 1 GeV. Large area neutrino telescopes in the ocean or in Antarctic ice typically have thresholds of the order of tens of GeV, which makes them sensitive mainly to heavy neutralinos (above 100 GeV) [19].

The integrand in Eq. (25.2) is weighted towards high neutrino energies, both because the cross section σ_ν rises approximately linearly with energy and because the average muon energy, and therefore the range λ , also grow approximately linearly with E_ν . Therefore, final states which give a hard neutrino spectrum (such as heavy quarks, τ leptons and W or Z bosons) are usually more important than the soft spectrum arising from light quarks and gluons.

25.1.2 Evolution of the number density in the Earth/Sun

Neutralinos are steadily being trapped in the Sun or Earth by scattering, whereas annihilations take them away. Let $N(t)$ be the total number of neutralinos trapped, at time t , in the core of, for example, the Earth. The annihilation rate of neutralino pairs can be written as

$$\Gamma_a(t) = \frac{1}{2} C_a N^2(t). \quad (25.3)$$

The evolution of $N(t)$ is the result of the competition between capture and annihilation:

$$\frac{dN}{dt} = C_c(t) - C_a N^2 \quad (25.4)$$

The constant C_c is the capture rate, and C_a entering equations (25.3) and (25.4) is linked to the annihilation cross-section σ_a , and to some effective volumes V_j , $j = 1, 2$, taking into account the quasi-thermal distribution of neutralinos in the Earth core:

$$C_a = \langle \sigma_a v \rangle \frac{V_2}{V_1^2}, \quad (25.5)$$

$$V_j \simeq 2.3 \times 10^{25} \left(\frac{j m_X}{10 \text{ GeV}} \right)^{-3/2} \text{ cm}^3. \quad (25.6)$$

This has the solution for the annihilation rate implemented in DarkSUSY

$$\Gamma_A = \frac{C_c}{2} \tanh^2 \left(\frac{t}{\tau} \right), \quad (25.7)$$

where the equilibration time scale $\tau = 1/\sqrt{C_c C_a}$. In most cases for the Sun, and in the cases of observable fluxes for the Earth, τ is much smaller than a few billion years, and therefore equilibrium is often a good approximation ($\dot{N}(t) = 0$). This means that it is the capture rate which is the important quantity that determines the neutrino flux. However, in the program we keep the exact formula (25.7), with some modifications discussed in Sec. ??).

25.1.3 Approximate capture rate expressions

The capture rate induced by scalar (spin-independent) interactions between the neutralinos and the nuclei in the interior of the Earth or Sun is the most difficult one to compute, since it depends sensitively on the Higgs mass, form factors, and other poorly known quantities. However, this spin-independent capture rate calculation is the same as for direct detection treated in Chapter 14. Therefore, there is a strong correlation between the neutrino flux expected from the Earth (which is mainly composed of spin-less nuclei) and the signal predicted in direct detection experiments [19, 115]. It seems that even the large (kilometer-scale) neutrino telescopes planned, when searching for neutralino annihilation in the Earth, will not be competitive with the next generation of direct detection experiments when it comes to detecting neutralino dark matter. However, the situation concerning the Sun is more favourable. Due to the low counting rates for the spin-dependent interactions in terrestrial detectors, high-energy neutrinos from the Sun constitute a competitive and complementary neutralino dark matter search. Of course, even if a neutralino is found through direct detection, it will be extremely important to confirm its identity and investigate its properties through indirect detection. In particular, the mass can be determined with reasonable accuracy by looking at the angular distribution of the detected muons [116, 117].

For the Sun, dominated by hydrogen, the axial (spin-dependent) cross section is important and relatively easy to compute. A reasonably good approximation is given by [118]

$$\frac{C_{\odot}^{\text{sd}}}{(1.3 \cdot 10^{23} \text{ s}^{-1}) (270 \text{ km s}^{-1}/\bar{v})} = \left(\frac{\rho_{\chi}}{0.3 \text{ GeV cm}^{-3}} \right) \left(\frac{100 \text{ GeV}}{m_{\chi}} \right) \left(\frac{\sigma_{p\chi}^{\text{sd}}}{10^{-40} \text{ cm}^2} \right) \quad (25.8)$$

where $\sigma_{p\chi}^{\text{sd}}$ is the cross section for neutralino-proton elastic scattering via the axial-vector interaction, \bar{v} is the dark-matter velocity dispersion, and ρ_{χ} is the local dark matter mass. The capture rate in the Earth is dominated by scalar interactions, where there may be kinematic and other enhancements, in particular if the mass of the neutralino almost matches one of the heavy elements in the Earth. For this case, a more detailed analysis is called for, which is available in [119] with convenient approximations in [118]. In fact, also for the Sun the spin-independent contribution can be important, in particular iron may contribute non-negligibly. For the Sun, the approximation in [118] is also available,

$$\frac{C_{\odot}^{\text{si}}}{(4.8 \cdot 10^{22} \text{ s}^{-1}) (270 \text{ km s}^{-1}/\bar{v})} = \left(\frac{\rho_{\chi}}{0.3 \text{ GeV cm}^{-3}} \right) \left(\frac{100 \text{ GeV}}{m_{\chi}} \right) \times \sum_A \left(\frac{\sigma_A^{\text{si}}}{10^{-40} \text{ cm}^2} \right) F_A(m_{\chi}) f_A \phi_A S(m_{\chi}/m_A) / m_A, \quad (25.9)$$

where f_A is the mass fraction of element A and ϕ_A is the typical gravitational potential (relative to the surface) for that element. I.e. an element that is concentrated in the core will have a higher ϕ_A

Element	Mass number (A)	Average parameters	
		f_i	ϕ_i
Hydrogen, H	1	0.670	3.15
Helium-4, ^4He	4	0.311	3.40
Carbon, C	12	0.00237	2.85
Nitrogen, N	14	0.00188	3.83
Oxygen, O	16	0.00878	3.25
Neon, Ne	20	0.00193	3.22
Magnesium, Mg	24	0.000733	3.22
Silicon, Si	28	0.000798	3.22
Sulphur, S	32	0.000550	3.22
Iron, Fe	56	0.00142	3.22

Table 25.1: The composition of the Sun with average parameters to be used in the approximative relations given in [118]. These values are updated with the solar model of [120] and differs slightly from the values used in [118].

Element	Mass number (A)	Mass fraction		Average parameters	
		Core	Mantle	f_i	ϕ_i
Oxygen, O	16	0.0	0.440	0.298	1.20
Silicon, Si	28	0.06	0.210	0.162	1.24
Magnesium, Mg	24	0.0	0.228	0.154	1.20
Iron, Fe	56	0.855	0.0626	0.319	1.546
Calcium, Ca	40	0.0	0.0253	0.0171	1.20
Phosphor, P	30	0.002	0.00009	0.00071	1.56
Sodium, Na	23	0.0	0.0027	0.00183	1.20
Sulphur, S	32	0.019	0.00025	0.0063	1.59
Nickel, Ni	59	0.052	0.00196	0.0181	1.57
Aluminum, Al	27	0.0	0.0235	0.0159	1.20
Chromium, Cr	52	0.009	0.0026	0.0047	1.44

Table 25.2: The composition of the Earth's core and mantle. The core mass fractions are from [121][Table 4] and the mantle mass fractions are from [121][Table 2]. The average mass fractions and potentials in the last two columns are weighted averages assuming a core mass of $1.93 \cdot 10^{24}$ kg and a mantle mass of $4.04 \cdot 10^{24}$ kg with average potentials (relative to the surface) of 1.6 in the core and 1.2 in the mantle [119].

than an element at the surface. A is the atomic number of the element and M_A is its mass. The factor S is a kinematical suppression factor [118, 103]. In the next subsection we will go through the compositions of the Earth/Sun that we use.

The approximate capture rate expressions above are coded into the routines `dsntcapsun` and `dsntcapearth`. More accurate expressions will follow in the coming subsections.

25.1.4 Earth and Sun composition

When the capture rates are calculated, we need to know the composition and density of the Earth/Sun as a function of depth.

In [118] they used average mass fractions and potentials for the location of the various elements in the Sun. We have updated these to the BP2000 [120] values instead, as given in Table 25.1

For the Earth, we have also implemented more accurate density profiles and more up-to date chemical distributions within the Earth. We use the estimates for the Earth composition given in [121][Table 2 for the mantle and Table 4 for the core]. In Table 25.2 we list these values together with

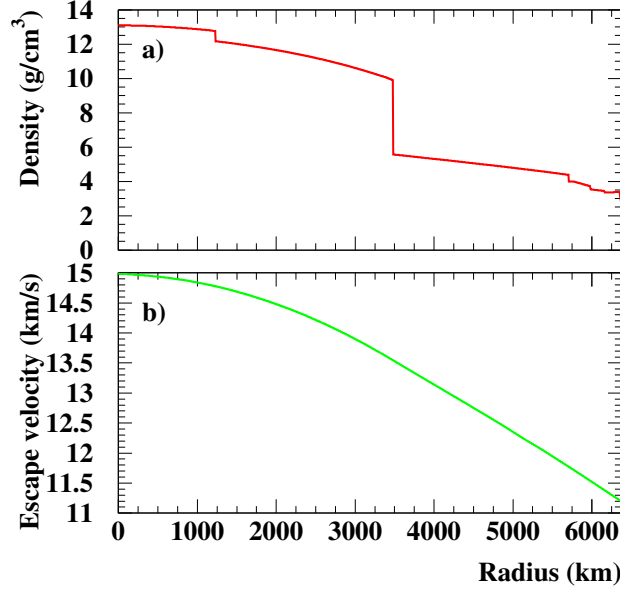


Figure 25.1: In a) the density profile and in b) the escape velocity in the Earth is shown.

the average parameters f_i and ϕ_i that should be used in the expressions for the approximate capture rates in the previous section. Note that using these average parameters instead of integrating over the full radius is equivalent to putting all the elements of the give type at the gravitational potential ϕ_i .

We also need the density profile of the Earth, and for this we use the values in [122]. Using this density profile, we can calculate the gravitational potential, $\phi(r)$ inside the Earth and from this one the escape velocity v inside the Earth,

$$v = 11.2 \sqrt{\frac{\phi(r)}{\phi(R_\oplus)}} \text{ km/s.} \quad (25.10)$$

In Fig. 25.1 we show the density profile and escape velocity inside the Earth.

25.1.5 More accurate capture rate expressions

Another complicating factor when calculating the capture rates is the integration over the velocity distribution. In [119], parts of the integrations are performed analytically for a Gaussian velocity distributions. These expressions are also coded in `DarkSUSY` for the Earth and give a more accurate calculation of the capture rate in the Earth than the approximations given above. The routine `dsntcapearth2` performs these calculations for the Earth.

25.1.6 Accurate capture rates in the Earth for general velocity distributions

If one wants even more accurate and general expressions for the capture rates in the Sun/Earth, we have also implemented the full expressions in [119], but without assuming that the velocity distribution is a Gaussian (or Maxwell-Boltzmann). These routines are now the default in `DarkSUSY`.

We will here outline how these expressions look like for the Earth and how they can be used both for a Maxwell-Boltzmann distribution and for a general velocity distribution. The expressions will of course look analogously for the Sun. We start with the general case and study the special case of a Maxwell-Boltzmann distribution in the next section.

We will divide the Earth into shells and calculate the capture from element i in each shell individually. At the end we will integrate over all the shells and sum over all the elements in the Earth. The capture rate from element i per unit shell volume is given by [119][Eq. (2.8)]

$$\frac{dC_i}{dV} = \int_0^{u_{max}} du \frac{f(u)}{u} w \Omega_{v,i}^-(w) \quad (25.11)$$

where $f(u)$ is the velocity distribution (normalized such that $\int_0^\infty f(u) = n_\chi$ where n_χ is the number density of WIMPs). The expression $\Omega_{v,i}^-(w)$ is related to the probability that we scatter to orbits below the escape velocity. w is the velocity at the given shell and it is related to the velocity at infinity u and the escape velocity v by

$$w = \sqrt{u^2 + v^2}. \quad (25.12)$$

The upper limit of integration is a priori set to $u_{max} = \infty$, but we will see below that due to kinematical reasons we can set it to a lower value (Eq. (25.17) below). If we allow for a form factor suppression of the form [119][Eq. (A3)]

$$|F(q^2)|^2 = \exp\left(-\frac{\Delta E}{E_0}\right) \quad (25.13)$$

with [119][Eq. (A4)]

$$E_0 = \frac{3\hbar^2}{2m_\chi R^2} \quad (25.14)$$

we can evaluate $w\Omega_{v,i}^-(w)$ and arrive at the expression [119][Eq. (A6)]

$$w\Omega_{v,i}^-(w) = \sigma_i n_i \frac{\mu_+^2}{\mu} 2E_0 \left[e^{-\frac{m_\chi u^2}{2E_0}} - e^{-\frac{\mu}{\mu_+} m_\chi \frac{u^2 + v^2}{2E_0}} \right] \Theta\left(\frac{\mu}{\mu_+} - \frac{u^2}{u^2 + v^2}\right) \quad (25.15)$$

where we have introduced

$$\mu = \frac{m_\chi}{m_i} \quad ; \quad \mu_\pm = \frac{\mu \pm 1}{2} \quad (25.16)$$

with m_i the mass of element i . The Heaviside step function Θ plays the role of only including WIMPs that can scatter to a velocity lower than the escape velocity v . To simplify our calculations we can drop this step function in Eq. (25.15) and instead set the upper limit of integration in Eq. (25.11) to

$$u_{max} = \sqrt{\frac{\mu}{\mu_-}} v \quad (25.17)$$

We also need the scattering cross section on element i , which can be written as [118][Eq. (9-25)]

$$\sigma_i = \sigma_p A_i^2 \frac{(m_\chi m_i)^2}{(m_\chi + m_i)^2} \frac{(m_\chi + m_p)^2}{(m_\chi m_p)^2} \quad (25.18)$$

where A_i is the atomic number of the element, m_p is the proton mass and σ_p is the scattering cross section on protons.

We now have what we need to calculate the capture rate. In Eq. (25.11) we integrate over the velocity for our chosen velocity distribution. We then integrate this equation over the radius of the Earth and sum over all the different elements in the Earth,

$$C = \int_0^{R_\oplus} dr \sum_i \frac{dC_i}{dV} 4\pi r^2 \quad (25.19)$$

Note that we have not assumed anything special about our velocity distribution, it doesn't even have to be isotropic since the distribution of elements evenly in the shells will make an anisotropic distribution on average to behave as an isotropic one.

The routines that calculate the capture rates with these general (and accurate) expressions are `dsntcapearthnum` and `dsntcapsunnum`. As these calculations are somewhat time-consuming, we have also added a possibility to tabulate the result and interpolate in these tables. To use (or create, if the table files are missing) instead call `dsntcapearthtab` and `dsntcapsuntab`. These last two routines are the default in `DarkSUSY`. The velocity distribution used is determined by a switch when the halo model is set (i.e. when `dshm_set` is called).

25.1.7 Accurate capture rates for the Earth for a Maxwell-Boltzmann velocity distribution

We will here give some more information on how the approximations introduced in the beginning of this chapter are derived from the general expressions in the preceding section.

If the velocity distribution is of Maxwell-Boltzmann type we can greatly simplify our expressions above as we can perform the integration over velocity analytically. The integration over radius can also be further simplified by using the average mass fractions f_i and potentials ϕ_i in Tables 25.1–25.2.

If the velocity distribution in the halo is Maxwell-Boltzmann, it looks like

$$f_h(u)du = n_\chi \frac{4}{\sqrt{\pi}} \left(\frac{3}{2}\right)^{\frac{3}{2}} \frac{u^2}{\bar{v}^3} e^{-\frac{3}{2}\frac{u^2}{\bar{v}^2}} du \quad (25.20)$$

where \bar{v} is the three-dimensional velocity dispersion and n_χ is the number density of WIMPs in the halo. However, the solar system moves through the halo with a velocity v_* and the distribution on observer with this velocity through the halo will see is

$$f_*(u) = f_h(u) e^{-\frac{3}{2}\frac{v_*^2}{\bar{v}^2}} \frac{\sinh\left(\frac{3uv_*}{\bar{v}^2}\right)}{\frac{3uv_*}{\bar{v}^2}} = n_\chi \sqrt{\frac{3}{2\pi}} \frac{u}{\bar{v}v_*} \left[e^{-\frac{3}{2}\frac{(u-v_*)^2}{\bar{v}^2}} - e^{-\frac{3}{2}\frac{(u+v_*)^2}{\bar{v}^2}} \right] \quad (25.21)$$

Now one would naively believe that this is not the distribution that an observer at the Earth will see. First of all, the Earth is moving with respect to the Sun and secondly, the WIMPs have gained speed by the gravitational attraction of the Sun when they reach the Earth. Both of these arguments are true and the distribution of WIMPs in the halo will not look like Eq. (25.21) to an observer on the Earth. However, Gould [123] showed that WIMPs from the halo can diffuse into the solar system due to gravitational interactions with the planets and this distribution of WIMPs will roughly look like as if the Earth was in free space moving through the halo with the velocity of the solar system, i.e. Eq. (25.21). We will later scrutinize this statement, as it turns out that it does not quite hold, but as a first guess it is a reasonable approximation. For the Sun, though, the velocity distribution give above is the correct one for a Maxwell-Boltzmann distribution.

With the distribution Eq. (25.21) we can analytically perform the integration over velocity in Eq. (25.11). After some algebra we arrive at [119][Eq. (A10)]

$$\begin{aligned} \frac{dC_i}{dV} = & \left(\frac{8}{3\pi}\right)^{\frac{1}{2}} \frac{\sigma_i n_i n_\chi \bar{v}}{2b\eta} \\ & \left[\frac{e^{-a\tilde{\eta}^2}}{\sqrt{1+a}} \left[2\widetilde{\text{erf}}(\tilde{\eta}) - \widetilde{\text{erf}}(\hat{A}_+) + \widetilde{\text{erf}}(\hat{A}_-) \right] \right. \\ & \left. - \frac{e^{-b\tilde{\eta}^2}}{\sqrt{1+b}} e^{-(a-b)A^2} \left[2\widetilde{\text{erf}}(\tilde{\eta}) - \widetilde{\text{erf}}(\check{A}_+) + \widetilde{\text{erf}}(\check{A}_-) \right] \right] \quad (25.22) \end{aligned}$$

where $\widetilde{\text{erf}}$ is the modified error function,

$$\widetilde{\text{erf}}(x) = \frac{\sqrt{\pi}}{2} \text{erf}(x) \quad ; \quad \text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-y^2} dy. \quad (25.23)$$

Following Gould [119], we have in Eq. (25.22) introduced the following shorthand notation:

$$\begin{aligned} \eta &= \sqrt{\frac{3}{2}} \frac{v_+^2}{\bar{v}^2} \quad ; \quad a = \frac{m_{\text{chi}} \bar{v}^2}{3E_0} \quad ; \quad b = \frac{\mu}{\mu_+^2} a \\ \hat{\eta} &= \frac{\eta}{\sqrt{1+a}} \quad ; \quad \check{\eta} = \frac{\eta}{\sqrt{1+b}} \\ A^2 &= \frac{3}{2} \frac{v_+^2}{\bar{v}^2} \frac{\mu}{\mu_-^2} \quad ; \quad \hat{A} = A\sqrt{1+a} \quad ; \quad \check{A} = A\sqrt{1+b} \\ \hat{A}_{\pm} &= \hat{A} \pm \hat{\eta} \quad ; \quad \check{A}_{\pm} = \check{A} \pm \check{\eta} \end{aligned} \quad (25.24)$$

If we wish, we can now integrate Eq. (25.22) over radius just like in the previous section, but we can without losing too much accuracy, replace this integration with a sum over the elements in the Earth with their respective typical location. I.e. we can write

$$C = \sum_i \frac{dC_i}{dV} \frac{1}{n_i} \frac{f_i M_{\oplus}}{m_i} \quad (25.25)$$

where we instead of the number density n_i use the total number of atoms of the given type $f_i M_{\oplus}/m_i$. Note that for each element in the sum we should evaluate this expression at the given typical gravitational potential ϕ_i of the element, i.e. with the escape velocity given by Eq. (25.10). The mass fractions f_i and typical potentials ϕ_i are listed in Table 25.2 (and analogously in Table 25.1 for the Sun). This approximation introduces an error of no more than about 1–2% for a Maxwell-Boltzmann distribution*

The capture rate evaluated with the expressions shown here are encoded into the routine `dsnt-capturearth2`. Note that we have not coded the corresponding approximate expressions for the Sun. Instead, as given in the preceding section, we now have more accurate expressions for both the Sun and the Earth.

25.1.8 Effects of WIMP diffusion in the solar system

As the Earth has a rather low escape velocity, the Earth will only be able to capture WIMPs that have a rather low velocity with respect to the Earth. However, WIMPs from the halo have gained speed in the gravitational potential from the Sun and will essentially be impossible to capture by the Earth. Hence, the Earth will only capture WIMPs that have diffused around in the solar system (by gravitational interactions with the other planets). Gould showed [123] that effectively this diffusion will lead to the same phase space distribution at the Earth as if the Earth was in free space (i.e. neglecting the solar potential). However, numerical simulations of asteroids showed that they are thrown into the Sun due to perturbations of the orbits by other planets, see e.g. [124]. These analyses led to worried that maybe the population of WIMPs diffusing around in the solar system is not as big as thought [125]. In [126], Lundberg and Edsjö investigated this issue with detailed numerical simulations of WIMP orbits in the solar system, showing that the annihilation rate in the Earth is typically reduced by up to two orders of magnitude. In `DarkSUSY`, we include these results for the neutrino rates from the Earth by using the velocity distribution at the Earth (as obtained in [126]). This velocity distribution is then used as input for our numerical capture rate routines instead of the usual approximation of using the halo velocity distribution directly. Using these new velocity distributions for the Earth is the default in `DarkSUSY`.

*Note that it is not advisable to use this approximation for general velocity distributions. If one e.g. has a lower limit on possible velocities, u_{min} , for heavy WIMPs capture will then only be possible very close to the central core. Replacing the actual distribution of potentials $\phi(r)$ with the typical value ϕ_i may then introduce larger errors. We will encounter these kind of distributions shortly.

25.2 Neutrinos from Sun and Earth – routines

This folder contains routines to calculate the neutrino-induced muon flux from the Earth and the Sun in various models.

There are several different methods of calculation available (determined by `secalcmet` in `dssecom.h`). Method 1 uses the approximate formulae for the capture rates in the Earth/Sun from the Jungman, Kamionkowski and Griest review [118]. Method 2, uses the same expression for the Sun, but the full expression from Gould [119] for capture in the Earth. Method 4 uses a full numerical integration over the velocity distribution (instead of assuming that it is Gaussian) and method 5, finally, also performs a full numerical integration over the momentum transfer in the form factors (instead of assuming exponential form factors). The default is to use method 5, but instead of using the numerical routines directly, tables are used. One can always force using the numerical routines instead of the tables if one so wishes. There are several options regarding how many elements to include in the Sun summation over elements ('lo', 'med' and 'hi'). The easiest way to select method is by calling `dssenu_set`. For the Earth, method 5 is not expected to make a large difference due to the smaller momentum transfers in the Earth and method 5 reverts back to method 4 for the Earth.

A call to `dssenu_set('default')` is made in `dsinit`, but can be changed by the user by calling `dssenu_set` after `dsinit`.

To calculate the neutrino-induced muon flux from the Earth, you call `dssenu_rates`.

Chapter 26

se_yield: Yields from annihilation in the Sun/Earth

(JE: I am responsible for updating this part)

26.1 Muon yields from annihilation in the Earth/Sun – theory

We need to take into account all processes that yield muon neutrinos from annihilation in the Earth/Sun. To do this, we use a Monte Carlo, WimpSim [127], to simulate annihilations in the center of the Sun/Earth, neutrino oscillations and neutrino interactions on the way out of the Sun/Earth and to the detector.

26.1.1 Monte Carlo simulations with WimpSim

We need to evaluate the yield of different particles per neutralino annihilation. The hadronization and/or decay of the annihilation products are simulated with PYTHIA [10] 6.426 and we here describe how the simulations are done. For annihilation in the Sun/Earth the simulations are done for a set of 22 neutralino masses, $m_\chi = 3, 6, 10, 25, 50, 80.3, 91.2, 100, 150, 176, 200, 250, 350, 500, 750, 1000, 1500, 2000, 3000, 5000, 7500$ and 10000 GeV. We tabulate the yields and then interpolate these tables in DarkSUSY.

We are mainly interested in the flux of high energy muon neutrinos and neutrino-induced muons at a neutrino telescope. We simulate 13 ‘fundamental’ annihilation channels, for each mass (where kinematically allowed) above. In Table 26.1 we list the ‘fundamental’ channels for which simulations are run and the full set of more complex channels. Pions and kaons get stopped before they decay and are thus made stable in the PYTHIA simulations so that they don’t produce any neutrinos. For annihilation channels containing Higgs bosons, we can calculate the yield from these fundamental channels by letting the Higgs bosons decaying in flight (see below). We also take into account the energy losses of B -mesons in the Sun and the Earth by following the approximate treatment of [110] but with updated B -meson interaction cross sections as given in [128]. We also take neutrino-interactions on the way out of the Sun into account by considering the charged-current interaction as a neutrino-loss and the neutral current interactions are simulated with nusigma [129]. The neutrino-nucleon charged current interactions close to the detector are also simulated with nusigma and finally the multiple Coulomb scattering of the muon on its way to the detector is calculated using distributions from [130]. We have used the CTEQ6 structure functions in these simulations. We also take into account neutrino oscillations with a full three-neutrino Monte Carlo. All of these

processes are put into the simulation package `WimpSim` [127] that can be downloaded separately. Results from simulation runs with this package are included with `DarkSUSY`. For more details on these simulations, see [131].

For each annihilation channel and mass we simulate 10^7 annihilations and tabulate the final results as a neutrino-yield, neutrino-to-lepton conversion rate, a muon yield and hadronic shower yields differential in energy and angle from the center of the Sun/Earth. We also tabulate the integrated yield above a given threshold and below an opening angle θ . We assumed throughout that the surrounding medium is water with a density of 1.0 g/cm^3 . Hence, the neutrino-to-muon conversion rates have to be multiplied by the density of the medium. In the muon fluxes, the density cancels out (to within a few percent). All results are summarized as yield tables that can be loaded and interpolated in with `DarkSUSY`. This is done with the function `dswayieldf`. There are three kinds of yields (two-dimensional in opening angle θ and energy), `kind=1` gives integrated yields, `kind=2` gives differential yields and `kind=3` gives yields integrated in angle, but differential in θ . For each kind, there are 26 different types of yield available according to Table 26.4. As a default, only type 3–4, 9–10 and 13–14 are included in the `DarkSUSY` download, as these are the most commonly used types. If you need any other types, download the auxiliary data files from <http://www.darksusy.org>, unpack them in `share/DarkSUSY` in the `DarkSUSY` root directory, and then do a usual `configure` and `make` to install them. Also note that the `kind=3` yields are not tabulated directly, but are instead calculated and tabulated when the simulation tables are read in during `DarkSUSY` initialization.

With these simulations, we can calculate the yield for any of these particles for a given MSSM model. For the Higgs bosons, which decay in flight, an integration over the angle of the decay products with respect to the direction of the Higgs boson is performed. Given the branching ratios for different annihilation channels it is then straightforward to compute the yield above any given energy threshold and within any angular region around the Sun or the center of the Earth. The routine `dswayieldone` calculates the yield for one channel, i.e. even these complex channels containing Higgs bosons, whereas the main routine `dswayield` calculates the total yield for a given model. Note that the WIMP annihilation yield routines do not know about SUSY at all, so before they are called, a routine `dswasetup` is called to set up the annihilation branching ratios for the WIMP and decay channels for the Higgs bosons. In Tables 26.2 and 26.3, the Higgs decay width channels are given. If these routines are used with other particle physics models, replace `dswasetup` with a routine appropriate for your particle physics model and then call `dswayield` as usual.

Annihilation channel ch	Particles	Internal channel chi	Internal channel array index chii
1	$S_1^0 S_1^0$	-	-
2	$S_1^0 S_2^0$	-	-
3	$S_2^0 S_2^0$	-	-
4	$S_3^0 S_3^0$	-	-
5	$S_1^0 S_3^0$	-	-
6	$S_2^0 S_3^0$	-	-
7	$S^- S^+$	-	-
8	$Z^0 S_1^0$	-	-
9	$Z^0 S_2^0$	-	-
10	$Z^0 S_3^0$	-	-
11	$W^- S^+ / W^+ W^-$	-	-
12	$Z^0 Z^0$	9	9
13	$W^+ W^-$	8	8
14	$\nu_e \bar{\nu}_e$	12	11
15	$e^+ e^-$	-	-
16	$\nu_\mu \bar{\nu}_\mu$	13	12
17	$\mu^+ \mu^-$	10	-
18	$\nu_\tau \bar{\nu}_\tau$	14	13
19	$\tau^+ \tau^-$	11	10
20	$u \bar{u}$	2	2
21	$d \bar{d}$	1	1
22	$c \bar{c}$	4	4
23	$s \bar{s}$	3	3
24	$t \bar{t}$	6	6
25	$b \bar{b}$	5	5
26	gg	7	7
27	qqq	-	-
28	$\gamma\gamma$	-	-
29	$Z^0 \gamma$	-	-

Table 26.1: The annihilation channels ch used in dswayieldone. Also shown are the internal channel numbers chi used for the fundamental channels used in the simulations (used by routine dswayieldf). To save some additional space with the data files in memory, there are also array index channel numbers chii that are only used internally to access the right elements of the yield arrays. S denotes scalars (Higgs bosons).

Decay width channel dch	Particles
1–29	Same as the annihilation channels in Table 26.1.
30	Sfermions
31	Neutralinos
32	Charginos

Table 26.2: The neutral scalar (Higgs) decay width channels used. In DarkSUSY these are stored in the array hdwidth(i,j) where i is the decay channel index above and j is the Higgs number (1–3 for H_1^0 , H_2^0 and H_3^0 respectively). For the wa routines, these decay branching ratios (partial width divided by total width) are stored in dswas0br(i,j).

Decay width channel dch	Particles
1	$u\bar{d}$
2	$u\bar{s}$
3	$u\bar{b}$
4	$c\bar{d}$
5	$c\bar{s}$
6	$c\bar{b}$
7	$t\bar{d}$
8	$t\bar{s}$
9	$t\bar{b}$
10	$\nu_e e^+$
11	$\nu_\mu \mu^+$
12	$\nu_\tau \tau^+$
13	$W^+ S_1^0$
14	$W^+ S_2^0$
15	$W^+ S_3^0$
20	Sfermions
21	Neutralinos and charginos

Table 26.3: The (positively) charged scalar (Higgs) decay width channels used. In DarkSUSY these are stored in the array `hdwidth(i,4)` where `i` is the decay channel index above. For the `wa` routines, these decay branching ratios (partial width divided by total width) are stored in `dswaschr(i)`.

Yield type type	Yield	Unit
1	ν_e	$10^{-30} \text{ m}^{-2} \text{ annihilation}^{-1}$
2	$\bar{\nu}_e$	$10^{-30} \text{ m}^{-2} \text{ annihilation}^{-1}$
3	ν_μ	$10^{-30} \text{ m}^{-2} \text{ annihilation}^{-1}$
4	$\bar{\nu}_\mu$	$10^{-30} \text{ m}^{-2} \text{ annihilation}^{-1}$
5	ν_τ	$10^{-30} \text{ m}^{-2} \text{ annihilation}^{-1}$
6	$\bar{\nu}_\tau$	$10^{-30} \text{ m}^{-2} \text{ annihilation}^{-1}$
7	e^- at neutrino-nucleon vertex	$10^{-30} \text{ m}^{-3} \text{ annihilation}^{-1}$
8	e^+ at neutrino-nucleon vertex	$10^{-30} \text{ m}^{-3} \text{ annihilation}^{-1}$
9	μ^- at neutrino-nucleon vertex	$10^{-30} \text{ m}^{-3} \text{ annihilation}^{-1}$
10	μ^+ at neutrino-nucleon vertex	$10^{-30} \text{ m}^{-3} \text{ annihilation}^{-1}$
11	τ^- at neutrino-nucleon vertex	$10^{-30} \text{ m}^{-3} \text{ annihilation}^{-1}$
12	τ^+ at neutrino-nucleon vertex	$10^{-30} \text{ m}^{-3} \text{ annihilation}^{-1}$
13	μ^- at an imaginary plane at detector	$10^{-30} \text{ m}^{-2} \text{ annihilation}^{-1}$
14	μ^+ at an imaginary plane at detector	$10^{-30} \text{ m}^{-2} \text{ annihilation}^{-1}$
15	hadronic shower from ν_e CC int. at neutrino-nucleon vertex	$10^{-30} \text{ m}^{-3} \text{ annihilation}^{-1}$
16	hadronic shower from $\bar{\nu}_e$ CC int. at neutrino-nucleon vertex	$10^{-30} \text{ m}^{-3} \text{ annihilation}^{-1}$
17	hadronic shower from ν_μ CC int. at neutrino-nucleon vertex	$10^{-30} \text{ m}^{-3} \text{ annihilation}^{-1}$
18	hadronic shower from $\bar{\nu}_\mu$ CC int. at neutrino-nucleon vertex	$10^{-30} \text{ m}^{-3} \text{ annihilation}^{-1}$
19	hadronic shower from ν_τ CC int. at neutrino-nucleon vertex	$10^{-30} \text{ m}^{-3} \text{ annihilation}^{-1}$
20	hadronic shower from $\bar{\nu}_\tau$ CC int. at neutrino-nucleon vertex	$10^{-30} \text{ m}^{-3} \text{ annihilation}^{-1}$
21	hadronic shower from ν_e NC int. at neutrino-nucleon vertex	$10^{-30} \text{ m}^{-3} \text{ annihilation}^{-1}$
22	hadronic shower from $\bar{\nu}_e$ NC int. at neutrino-nucleon vertex	$10^{-30} \text{ m}^{-3} \text{ annihilation}^{-1}$
23	hadronic shower from ν_μ NC int. at neutrino-nucleon vertex	$10^{-30} \text{ m}^{-3} \text{ annihilation}^{-1}$
24	hadronic shower from $\bar{\nu}_\mu$ NC int. at neutrino-nucleon vertex	$10^{-30} \text{ m}^{-3} \text{ annihilation}^{-1}$
25	hadronic shower from ν_τ NC int. at neutrino-nucleon vertex	$10^{-30} \text{ m}^{-3} \text{ annihilation}^{-1}$
26	hadronic shower from $\bar{\nu}_\tau$ NC int. at neutrino-nucleon vertex	$10^{-30} \text{ m}^{-3} \text{ annihilation}^{-1}$

Table 26.4: The yield types available from the `wa` routines. All of these yields are at the detector (currently IceCube). Note that the units are for integrated yields (`kind=1`), for differential yields (`kind=2`), the units should be multiplied by $\text{GeV}^{-1} \text{ degree}^{-1}$. CC int. = charged current interactions. NC int. = neutral current interactions.

Chapter 27

si: Self-interactions

27.1 Dark matter self-interactions (si) – theory

The CDM paradigm, resting on cold and collisionless dark matter, describes cosmological structure formation with remarkable accuracy at scales larger than about one Mpc. At smaller cosmological scales, on the other hand, this paradigm is less well tested, and currently still allows for DM self-interactions as strong as the interaction between nucleons – and thus much stronger than current limits on DM interacting with SM particles. The possibility that DM could be relatively strongly interacting with itself [132] and thereby leave imprints on cosmological observables related to structure formation has therefore seen significant interest in recent years, both from an astrophysical and a model-building perspective. Self-interacting DM (SIDM) could even address [133, 134, 135, 136, 137, 138, 139, 140] the most pressing potential small-scale problems of Λ CDM cosmology [141], referred to as ‘core-vs.-cusp’ [142, 143], ‘too-big-to-fail’ [144, 145], ‘diversity’ [146, 147] and ‘missing satellites’ problems [148, 149] (the latter problem is addressed for SIDM with late kinetic decoupling, which is a natural combination in many models [79]). Those specific discrepancies with the Λ CDM paradigm may of course well turn out to be either due to poorly modelled baryonic effects or observational uncertainties – but these examples serve as evidence that SIDM *can* leave observable imprints even in the absence of interactions with the SM. An unambiguous detection of such signatures, which are not expected in for example traditional WIMP models of DM, would significantly narrow down the range of possible particle explanations for the nature of DM.

The traditional reference quantity for the impact of DM self-interactions on the halo structure is the momentum transfer cross section,

$$\sigma_T \equiv \int d\Omega (1 - \cos\theta) \frac{d\sigma}{d\Omega}, \quad (27.1)$$

where σ is the standard cross section for DM-DM scattering. This has the advantage of regulating large forward-scattering amplitudes, which should not affect the DM distribution (the same goes for backward scattering which, however, is treated symmetrically in this prescription. See Ref. [150, 151] for a more detailed discussion). The size of this quantity that is of cosmological relevance is very roughly given by

$$\sigma_T/m_\chi \sim 1 \text{ cm}^2/\text{g}. \quad (27.2)$$

Cross sections in this ballpark, in other words, may leave observable imprints and possibly address the various Λ CD; small-scale structure problems mentioned above, while much smaller cross sections have no impact on the structure of DM halos. Much larger values of σ_T/m_χ are ruled out, on

the other hand, in particular from observations of galaxy clusters. For a detailed review that summarizes both various constraints on DM self-interactions and models that have been discussed in the literature, we refer to Ref. [152].

The details of the DM self-interactions depend on the underlying particle theory. A simple contact interaction term in the Lagrangian, for example, would lead to a constant (velocity-independent) σ_T . Another well-motivated option is that the interaction is mediated by a massive particle with mass m_{med} , which leads to a Yukawa potential between the two DM particles:

$$V(r) = \pm \frac{\alpha_\chi}{r} e^{-m_{\text{med}} r}. \quad (27.3)$$

Here, $\alpha_\chi = g_\chi^2/(4\pi)$ describes the coupling strength between mediator and DM particles, and the different signs refer to attractive (-) and repulsive (+) potentials. Scalar mediators only generate attractive potentials, while for vector mediators this depends on the particles involved in the scattering: for (e.g. Dirac) DM scattering with anti-DM, the force is attractive, otherwise it is repulsive. An interesting phenomenological aspect for the scattering of nonrelativistic particles in such a potential is the resulting strong velocity-dependence of σ_T ; this allows to achieve large scattering cross sections for the relatively small velocities of ~ 30 km/s typically encountered in dwarf galaxies (where one observes potential discrepancies with the Λ CDM expectations) while evading the strong bounds at cluster scales for velocities of ~ 1000 km/s. The transfer cross section resulting from scattering in a Yukawa potential has been extensively studied, and depends on the scattering regime (v_{rel} is the relative velocity between the DM particles):

- In the **Born regime** ($\alpha_\chi m_\chi \lesssim m_{\text{med}}$), σ_T can be calculated perturbatively, leading (for both attractive and repulsive potentials) to [153]

$$\sigma_T^{\text{Born}} = \frac{8\pi\alpha_\chi^2}{m_\chi^2 v_{\text{rel}}^4} \left(\ln \left[1 + \frac{m_\chi^2 v_{\text{rel}}^2}{m_{\text{med}}^2} \right] - \frac{m_\chi^2 v_{\text{rel}}^2}{m_{\text{med}}^2 + m_\chi^2 v_{\text{rel}}^2} \right). \quad (27.4)$$

- In the **classical regime** ($m_\chi v_{\text{rel}} \gtrsim m_{\text{med}}$), a large numbers of partial waves contributes such that non-perturbative effects are important. In principle, this can be computed by numerically solving the Schrödinger equation for each case and then summing all contributions [150]. A computationally much more efficient method is to use parameterizations [154] of numerical results from the Plasma literature that have been obtained for screened Coulomb scattering [155, 156]. For an attractive potential, these are given by

$$\sigma_T^- = \frac{\pi}{m_{\text{med}}^2} \times \begin{cases} 2\beta^2 \ln[1 + \beta^{-2}] & \text{for } \beta \lesssim 10^{-2} \\ \frac{7\beta^{1.8} + 1960(\beta/10)^{10.3}}{1 + 1.4\beta + 0.006\beta^4 + 160(\beta/10)^{10}} & \text{for } 10^{-2} \lesssim \beta \lesssim 10^2 \\ 0.81(1 + \ln\beta - (2\ln\beta)^{-1})^2 & \text{for } \beta \gtrsim 10^2 \end{cases} \quad (27.5)$$

where $\beta \equiv 2\alpha_\chi m_{\text{med}}/(m_\chi v_{\text{rel}}^2)$, while for a repulsive potential we have

$$\sigma_T^+ = \frac{\pi}{m_{\text{med}}^2} \times \begin{cases} 2\beta^2 \ln[1 + \beta^{-2}] & \text{for } \beta \lesssim 10^{-2} \\ \frac{8\beta^{1.8}}{1 + 5\beta^{0.9} + 0.85\beta^{1.6}} & \text{for } 10^{-2} \lesssim \beta \lesssim 10^4 \\ (\ln 2\beta - \ln \ln 2\beta)^2 & \text{for } \beta \gtrsim 10^4 \end{cases} \quad (27.6)$$

- Finally, there is the **resonant regime** (for $m_\chi v_{\text{rel}} \lesssim m_{\text{med}}$ or $\alpha_\chi m_\chi \gtrsim m_{\text{med}}$). Full numerical solutions can be obtained by solving the Schrödinger equation explicitly in this regime. These are well described by the following analytic expressions that result from approximating the Yukawa potential with a Hulthén potential [150]

$$\sigma_T^{\text{Hulthn}} = \frac{16\pi}{m_\chi v_{\text{rel}}^2} \sin^2 \left(\text{Arg} \left[\frac{i\Gamma[i\Theta v_{\text{rel}}]}{\Gamma[\lambda_+] \Gamma[\lambda_-]} \right] \right), \quad (27.7)$$

where $\Theta \equiv \frac{m_\chi}{\sqrt{2\zeta(3)m_{\text{med}}}}$ and $\lambda_\pm \equiv 1 + i\Theta v_{\text{rel}}/2 \pm \sqrt{\alpha_\chi \Theta - \Theta^2 v_{\text{rel}}^2/4}$ for an attractive potential and $\lambda_\pm \equiv 1 + i\Theta v_{\text{rel}}/2 \pm i\sqrt{\alpha_\chi \Theta + \Theta^2 v_{\text{rel}}^2/4}$ for a repulsive potential.

27.2 Self-interactions – routines

In DarkSUSY, $\sigma_T(v_{\text{rel}})/m_\chi$ is provided by an interface function `dssigtm` returned by the particle module which, from the perspective of the core library, can take any functional form. The function `dssigmatmav` then computes the velocity average $\langle \sigma_T \rangle / m_\chi$, assuming a Maxwellian velocity distribution of the DM particles; this gives a better estimate of the effect of DM self-interactions than just evaluating `dssigtm` directly for a typical halo velocity. The core library furthermore provides several auxiliary routines, to be used by any particle module, for the commonly encountered specific transfer cross sections in the presence of a Yukawa potential as discussed above. In particular, `dssigmatborn` returns the expression given in Eq. (27.4), `dssigmatclassical` those given in Eqs. (27.5, 27.6), and `dssigmatres` those given in Eq. (27.7). For the latter, we adopt the complex gamma function as implemented in the CERN library (based on Ref. [157]), and use analytic expansions where a naive usage of these routines is problematic due to limited numerical precision (which is relevant for a significant fraction of the physically interesting parameter range).

Part III

Particle physics modules in src_models

Chapter 28

Basic principles and common routines

The general concept of a particle physics module, and how it communicates with the `core` library via interface functions, was already introduced in Chapter 3 – see in particular Section 3.2, and Table 3.3 for an automatically updated list of all interface functions that the `core` library is aware of. Note that there is no *principal* restriction on which interface functions a particle module must provide: the main program will determine at compilation time whether it needs a functionality of the `core` library that requires certain interface functions to exist.

Every particle physics module must provide an independent representation of the full particle content of the respective particle theory. How this is done is fully flexible, and completely up to the module. In practice, however, there are common frameworks, like the Standard Model, that appear repeatedly. For convenience, we therefore store auxiliary routines and setups that *may* be used by more than just one particle module in `src_models/common`. Common blocks and header files included by more than one particle module are found in `src_model/include`. In order to keep up the modularity of the code, routines in `src_models/common` thus *only* include files in `src_model/include`.

28.1 `common/aux`: auxiliary routines

Here we collect various routines that belong to particle physics, and hence do not reside in `ds_core`, but are not only useful for one specific particle module. Currently, the most important are

- A set of functions to handle *unique model IDs* for each particle model, which is essential in order not to repeat identical calculations when changing the model and thus to optimize numerical performance. A new such model ID should be assigned with `dsnewidnumber` whenever a new particle model is initialized (for the modules provided with the `DarkSUSY` release, this is automatically done in `dsmodelsetup`). Any routine in `src_models` can test with `dsidnumberset` whether this ID number is indeed set, and retrieve its value with `dsidnumber` (which then can be compared to the locally stored value of the ID number that was valid at the time when the respective routine was called the previous time).
- The function `dsanthreshold` returns the correction factor to a 2-body rate close to a kinematical threshold, resulting from the fact that one or both of the particles may be slightly off-shell. This implements the simplified treatment presented in Ref. [31], c.f. their Fig. 5, and hence assumes that the decay products of the virtual particle are (effectively) massless.

28.2 `common/sm`: standard model

The routines collected in this folder provide a convenient shortcut to include the most basic properties of the standard model in a BSM particle module. Currently, the most important functionalities

collected here are given by

- An initialization routine `dsinit_sm`, which can be called directly from the corresponding `dsinit_module`. After a call to this routine, the functions `dsmass(kPDG)` and `dswidth(kPDG)` correctly return masses and widths of the standard model particles – `kPDG` being the PDG code [130] – as stored centrally in `src_models/include/dssmparam.h`.
- Running quark masses are provided up to the 4-loop level, with SM contributions only, both for pole and \overline{MS} masses [158].
- The SM contributions to the strong coupling constant are also provided up to the 4-loop level.
- Another convenience function is `dsgf2s2thw`, which calculates $\sin^2(\theta_W)$ at the m_Z scale: in praxis, its most important application is to ensure that a particle module can adopt a consistent relation between m_Z and m_W , which for example can be crucial in order to keep large numerical cancellations under control.

Let us stress that standard model physics in `DarkSUSY` is by far not restricted to the routines collected in `common/sm`. Much is presently still contained in the `mssm` module. It will be (further) disentangled from the SUSY-specific parts as new modules are added to the code that require this functionality.

Chapter 29

The empty model

The empty module cannot be used to perform any real calculations. Its purpose is for debugging and testing only, with the main advantage being that it contains (empty versions of) all interface functions that the core library is aware of (for a full list, see Table 3.3). In some cases, this makes it a good starting point to create new particle physics modules.

Chapter 30

Generic decaying dark matter

This module provides a simple phenomenological template for a generic decaying dark matter model. It is fully specified by the DM mass m_χ and the total decay rate Γ , along with a list of decay channels. The latter are specified by their branching ratios and the PDG codes of the final state standard model particles. Non-standard model final states can also be included, and are treated as invisible decays.

The phenomenology of this module is restricted to indirect DM searches, with full access to the corresponding routines in `ds_core`. DM scattering with nuclei is assumed to be negligible, as well thermal production of DM in the early universe.

Chapter 31

Generic FIMPs

Similar to the case of `generic_wimp`, the module `generic_fimp` provides a simple phenomenological template for a large class of DM candidates. Rather than being based on an actual particle physics model, it mostly serves to provide an illustration of how the functionalities of DarkSUSY can be used in phenomenological freeze-in studies of ‘vanilla’ FIMP DM, when only providing the absolute minimum of input parameters.

A generic FIMP model in DarkSUSY is set up by a call to `dsgivemodel_generic_fimp`, followed as usual by a call to `dsmodelsetup`. The model is thus fully defined by the input parameters of `dsgivemodel_generic_fimp`: the mass m_χ of the DM particle and a flag stating whether the DM particle is its own anti-particle or not; a *single* SM species that the DM particle is assumed to couple to (defined by its PDG code, `PDGSM`) and three parameters (c, Λ, n) that specify strength and type of the interaction. Concretely, the squared amplitude for $2 \rightarrow 2$ production in this model is assumed to take the form

$$|\mathcal{M}|^2 \equiv c \left(\frac{s}{\Lambda^2} \right)^n, \quad (31.1)$$

where s is the CMS energy squared.* Very roughly, this would, *e.g.*, correspond to the amplitude expected for contact interactions in an effective field theory that results from integrating out heavy d.o.f. (where ‘heavy’ in this context means higher energies than the temperature where freeze-in production dominates).

Presently, the `generic_fimp` module only allows the user to calculate the relic density from freeze-in production (as well as, in principle, the would-be density from freeze-out), since rates are expected to be too small to be relevant for indirect and direct detection signals. For phenomenological studies of the latter, the `generic_wimp` module is thus better suited.

***Technical note.** Allowed values for `PDGSM` are all elementary SM particles in the EW broken phase. Strictly speaking, these choices are ill-defined above the EW phase transition and, for gluons and quarks, below the hadronization scale; results of freeze-in calculations must hence be interpreted with care if the dominant freeze-in production happens outside this range.

For reference, we treat situations outside this range as follows:

- For quark and gluon final states, we use Eq. (31.1) only for $T > T_{\text{qcd}}$ and for $\sqrt{s} > E_{\text{hadron}}$ (global parameters set in `dsinit` and `dsinit_sm`, respectively); outside this range, we set $|\mathcal{M}|^2 = 0$.
- For $T > T_{\text{EW}}$, fermions are chiral; we model their thermal masses as the *average* of the contributions from left- and right handed states – see `dsmass_finiteT`. Photons we continue to describe as a $U(1)$ particle with electromagnetic couplings, while we describe both Z and W^\pm as unbroken $SU(2)$ bosons (with temperature-dependent transverse masses).

We stress that this implementation is just a heuristic way of gauging the impact of phase transitions on the result of freeze-in calculations when insisting on the possibility of specifying the interacting SM species with a single (integer) parameter `PDGSM` (in analogy to the `generic_wimp` module). A more realistic implementation is necessarily model-dependent (see, e.g., the `silveira_ze` module described in Chapter 34).

31.1 `generic_fimp/fi`: Freeze-in routines

A call to `dsfiset_generic_FIMP` prior to `dsfi2to2rhs` allows to perform freeze-in calculations with modified default settings. For example, call `dsfiset_generic_FIMP('quantum_statistics','off')` will lead to a freeze-in calculation without taking into account the effect of quantum statistics for the heat bath particles, and call `dsfiset_generic_FIMP('finiteT','off')` will switch off other finite-temperature effects (thermal masses and further temperature-dependence due to phase transitions). Default settings are restored with corresponding calls to the same routine, replacing 'off' with 'default'.

Chapter 32

Generic WIMPs

Similar to the case of `generic_decayingDM`, the module `generic_wimp` provides a simple phenomenological template for a large class of DM candidates. Rather than being based on an actual particle physics model, it mostly serves to provide an illustration of how the functionalities of DarkSUSY can be used in phenomenological studies of 'vanilla' WIMP DM, when only providing the absolute minimum of input parameters.

32.1 Generic WIMP and simple extensions

A generic WIMP model in DarkSUSY is set up by a call to `dsgivemodel_generic_wimp`, and hence fully defined by the input parameters of that routine:

- The mass of the DM particle, m_{WIMP} .
- The annihilation cross section $\langle\sigma v\rangle$.
- The PDG code of the particle (and antiparticle) that the WIMP annihilates to. It is assumed to annihilate to a two-body final state with this set of final state particles.
- The spin-independent WIMP-nucleon scattering cross section, σ_{SI} . This is assumed to apply for scattering off protons and neutrons.

In addition the DM particle is in this simple setup assumed to be self-conjugate (its own antiparticle), with spin 0, with no spin-dependent scattering cross sections for WIMP-nucleon scatterings and with one internal degree of freedom. In most cases, this is what is usually called a generic WIMP.

However, at times it can be useful to extend this model to also include other properties of the DM particle. The routine `dsgivemodel_generic_wimp_opt` allows to set additional properties. It takes two arguments, a key to determine what property to set and a value, according to the following

Optional parameters in `dsgivemodel_generic_wimp_opt`.

<code>spin</code>	r8	Allows to set the spin of the DM particle. The internal degrees of freedom is updated accordingly.
<code>sigsip</code>	r8	Spin-independent WIMP-proton scattering cross section.
<code>sigsin</code>	r8	Spin-independent WIMP-neutron scattering cross section.
<code>sigsdp</code>	r8	Spin-dependent WIMP-proton scattering cross section.
<code>sigsdn</code>	r8	Spin-dependent WIMP-neutron scattering cross section.

This simple setup allows to use essentially the full functionality of the core library, from relic density calculation to direct detection rates and cosmic ray propagation and indirect detection signals in general. If needed, it is relatively simple to extend the generic WIMP model if you wish

to explore other similar models. Please use the script `scr/make_module.pl` to create a copy of the generic WIMP module and make your changes there.

32.2 Asymmetric dark matter

While a call to `dsgivemodel_generic_wimp` (followed by `dsmodelsetup`) always initializes a DM particle that is treated as its own antiparticle, the `generic_wimp` model also provides the possibility of initializing a DM particle that is *not* self-conjugate. This is done by a call to `dsgivemodel_generic_wimp` instead, which takes the primordial DM asymmetry, $(n_+ - n_-)/s > 0$, as additional input parameter η_{model} .

The main effect of η_{model} appears in the source term entering in indirect detection yields, provided by the interface function `dscrsource`. Concretely, the source term will be reduced by a factor of

$$\xi = \frac{1}{2}r(2 - r) \quad (32.1)$$

compared to the case of self-conjugate DM, see also Eq. (11.3). Here, r describes the ratio of the symmetric to the total DM component as given by Eq. (22.74):

$$r = \frac{2n_-}{n_+ + n_-} = 1 - \eta_{\text{model}} \frac{c}{\Omega_{\text{DM,obs}} h^2}, \quad (32.2)$$

with $c = 2.755 \times 10^{10} (m_{\text{DM}}/100 \text{ GeV})$ and $\Omega_{\text{DM,obs}} h^2 \approx 0.112$ being the total cosmological DM density (stored in the common block variable `omegacdmh2`).

The value of r given in Eq. (32.2) assumes that all of the observed DM is made up of particles with the properties given in this specific model. If, instead, this only provides a subdominant DM fraction, or if the value of r following from the relic density calculation, cf. section 22.3, is inconsistent with the value obtained by Eq. (32.2), the default rescaling given by Eq. (32.1) may not be appropriate. For that purpose, we provide the auxiliary function `dscrrescale_asym`, see section 9.1, to help users rescale the default output of indirect detection routines as intended.

Chapter 33

The Minimal Supersymmetric Standard Model

The implementation of the minimal supersymmetric standard model (MSSM) in the module `mssm` mainly follows that of `DarkSUSY` versions 4 [1] (see also Section 33.8 in this manual). In particular, the conventions for the superpotential and soft supersymmetry-breaking potential are the same as implemented in [25], and thus similar to [159, 160]. The full set of input parameters to be provided at the weak scale thus consists of the pseudoscalar mass (m_A), the ratio of Higgs vacuum expectation values ($\tan\beta$), the Higgsino (μ) and gaugino (M_1, M_2, M_3) mass parameters, trilinear couplings ($A_{Eaa}, A_{Uaa}, A_{Daa}$, with $a = 1, 2, 3$) as well as soft sfermion masses ($M_{Qaa}^2, M_{Laa}^2, M_{Uaa}^2, M_{Daa}^2, M_{Eaa}^2$, with $a = 1, 2, 3$).^{*} Internally, those values are stored in `mssm` common blocks. The user may either provide them directly or by setting up pre-defined phenomenological MSSM models with a reduced number of parameters through a call to a routine like `dsgive_model` or `dsgive_model25` (followed by a call to `dsmodelsetup`). The former sets up the simplest of those models, defined by the input parameters $\mu, M_2, m_A, \tan\beta$, a common scalar mass m_0 , and trilinear parameters A_t and A_b ; M_1 and M_3 are then calculated by assuming the GUT condition, and the remaining MSSM parameters are given by $\mathbf{M}_Q = \mathbf{M}_U = \mathbf{M}_D = \mathbf{M}_E = \mathbf{M}_L = m_0\mathbf{1}$, $\mathbf{A}_U = \text{diag}(0, 0, A_t)$, $\mathbf{A}_D = \text{diag}(0, 0, A_b)$, $\mathbf{A}_E = \mathbf{0}$. Similarly, `dsgive_model25` sets up a pMSSM model with 25 free parameters (see the header of that file for details). Alternatively, all those values can be set by reading in an SLHA file, or providing GUT scale parameters in the case of cMSSM models (via an interface to the ISASUGRA code, as included in ISAJET [161, 162]).

Compared to previous versions of the code, `DarkSUSY 6` has a new interface to read and write SUSY Les Houches Accord (SLHA) files [163, 164].

All particle and sparticle masses are stored in a common block array `mass()`. For neutralino masses, we include the leading loop corrections [165, 166, 167] but neglect the relatively small corrections for charginos [165] (in both cases, masses cannot be negative in our convention).[†] Likewise, all mixing matrices and decay widths are available as common block arrays. The latter are currently only computed for the Higgs particles (via an interface to the `FeynHiggs` [168, 169, 170, 171, 172] package), while the other sparticles have fictitious widths of 0.5% of the sparticle mass (for the sole purpose of regularizing annihilation amplitudes close to poles). Again, the conventions for masses and mixings follow exactly those of Ref. [1], to which we refer for further details.

^{*}Note that currently only diagonal matrices are allowed. While not being the most general ansatz possible, this implies the absence of flavour changing neutral currents at tree-level in all sectors of the model.

[†]Unless, of course, those values are provided by an SLHA file. This comment also applies to the following simplifications concerning both sparticle masses and widths.

33.1 mssm/ac: Accelerator constraints

33.1.1 Accelerator bounds

DarkSUSY contains a set of routines for a rough check whether a given model is excluded by accelerator constraints. These routines are called **dsacbnd[*number*]**. The policy is that when we update DarkSUSY with new accelerator constraints, we keep the old routine, and add a new routine with the last number incremented by one. Which routine that is called is determined by calling **dsacset** with a tag determining which routine to call. To check the accelerator constraints, then call **dsacbnd** which calls the right routine for you. Upon return, **dsacbnd** returns an exclusion flag, **warning**. If zero, the model is OK, if non-zero, the model is likely excluded. The cause for the exclusion is coded in the bits of **excl** according to table 33.1.

We stress that these bounds are in most cases only *approximate* limits: DarkSUSY generally focusses on theoretical predictions for such observables, given a DM model realization, rather than on the implementation of experimental likelihoods and the possibility to derive statistically well-defined limits from those. For the latter, we instead refer to packages like DarkBit [173] (or ColliderBit [174] for accelerator-based constraints)

excl				
Bit set	Octal value	Decimal value		Reason for exclusion
0	1	1		Chargino mass
1	2	2		Gluino mass
2	4	4		Squark mass
3	10	8		Slepton mass
4	20	16		Invisible Z width
5	40	32		Higgs mass in excluded region
6	100	64		Neutralino mass
7	200	128		$b \rightarrow s\gamma$
8	400	256		ρ parameter
9	1000	512		$(g-2)_\mu$
10	2000	1024		$B_s \rightarrow \mu^+\mu^-$
11	4000	2048		squark-gluino
12	10000	4096		Higgs mass does not fit observed Higgs

Table 33.1: The bits of **excl** are set to indicate by which process this particular model is excluded. Check if a bit is set with **btest(excl,bit)**.

Compared to previous DarkSUSY versions, we use in particular updated limits from HiggsBounds [175] on the mass of the MSSM Higgs bosons, as well as approximate bounds on squark and gluino masses from LHC 8 TeV data [176]. For $b \rightarrow s\gamma$, we keep our genuine routines in **mssm/ac_bsg** for this rare decay (see Ref. [1] for a more detailed description) but now use as a default the result from SuperIso [177]; we compare this to the current limit of $\mathcal{B}(B \rightarrow X_s\gamma) = (3.27 \pm 0.14) \times 10^{-4}$ as adopted in FlavBit [178], based on data from BarBar and Belle [179, 180, 181]. SuperIso also computes the rate for the rare leptonic decay $B_s^0 \rightarrow \mu^+\mu^-$, which we compare to the LHCb measurement of $\mathcal{B}(B_s^0 \rightarrow \mu^+\mu^-) = (3.0 \pm 0.6^{+0.3}_{-0.2}) \times 10^{-9}$ [182]. Finally, $a_\mu \equiv (g-2)_\mu/2$ is calculated by **dsgm2muon**, based on [183]; in **dsacbnd**, this is compared to the observed value of $a_{\mu,\text{obs}} = (11659208.9 \pm 6.3) \times 10^{-10}$ [184] after subtracting the SM expectation as specified in PrecisionBit [185].

33.2 mssm/an: (Co-)annihilation cross sections

33.2.1 Annihilation cross sections – theory

For the relic density calculations, we need all possible (co)annihilation cross sections between neutralinos, charginos and sfermions.

33.2.1.1 Annihilation cross sections

We have calculated all two-body final state cross sections at tree level involving neutralinos, charginos, sneutrinos, sleptons and squarks in the initial state. A complete list is given below.

Since we have so many different diagrams contributing, we have to use some method where the diagrams can be calculated efficiently. To achieve this, we calculate the diagrams with general expressions for vertices, masses etc so that they can be reused for other processes. How we do this in practice differs a bit between different sets of annihilation diagrams.

For neutralino-neutralino, neutralino-chargino and chargino-chargino annihilation, we classify the diagrams according to their topology (s -, t - or u -channel) and to the spin of the particles involved. We then compute the helicity amplitudes for each type of diagram analytically with REDUCE [186] using general expressions for the vertex couplings.

The strength of the helicity amplitude method is that the analytical calculation of a given type of diagram has to be performed only once and the sum of the contributing diagrams for each set of initial and final states can be done numerically afterwards.

For the diagrams involving sfermions, FORM is used to analytically calculate the amplitudes. This output is then converted into Fortran with a PERL script, **form2f** [187].

33.2.1.2 Coannihilation diagrams

All Feynman diagrams for which we calculate the annihilation cross section are listed in the coming sections. $s(x)$, $t(x)$ and $u(x)$ denote a tree-level Feynman diagram in which particle x is exchanged in the s -, t - and u -channel respectively.

The convention used in this list of included coannihilation diagrams is that if a sfermion is denoted \tilde{f} , then its antiparticle is denoted \tilde{f}^* .

33.2.1.3 Neutralino and chargino annihilation

Indices i, j, k run from 1 to 4, and indices c, d, e from 1 to 2. $u, \tilde{u}, d, \tilde{d}, \nu, \tilde{\nu}, \ell, \tilde{\ell}, f$ and \tilde{f} are generic notations for up-type quarks, up-type squarks, down-type quarks, down-type squarks, neutrinos, sneutrinos, leptons, sleptons, fermions and sfermions. A sum of diagrams over (s)fermion generation indices and over the neutralino and chargino indices k and e is understood (no sum over indices i, j, c, d).

Neutralino-neutralino annihilation

Initial state	Final state	Feynman diagrams
$\chi_i^0 \chi_j^0$	$H_1 H_1, H_1 H_2, H_2 H_2, H_3 H_3$	$t(\chi_k^0), u(\chi_k^0), s(H_{1,2})$
	$H_1 H_3, H_2 H_3$	$t(\chi_k^0), u(\chi_k^0), s(H_3), s(Z^0)$
	$H^- H^+$	$t(\chi_e^+), u(\chi_e^+), s(H_{1,2}), s(Z^0)$
	$Z^0 H_1, Z^0 H_2$	$t(\chi_k^0), u(\chi_k^0), s(H_3), s(Z^0)$
	$Z^0 H_3$	$t(\chi_k^0), u(\chi_k^0), s(H_{1,2})$
	$W^- H^+, W^+ H^-$	$t(\chi_e^+), u(\chi_e^+), s(H_{1,2,3})$
	$Z^0 Z^0$	$t(\chi_k^0), u(\chi_k^0), s(H_{1,2})$
	$W^- W^+$	$t(\chi_e^+), u(\chi_e^+), s(H_{1,2}), s(Z^0)$
	$f \bar{f}$	$t(f_{L,R}), u(f_{L,R}), s(H_{1,2,3}), s(Z^0)$

Neutralino-chargino annihilation

Initial state	Final state	Feynman diagrams
$\chi_c^+ \chi_i^0$	$H^+ H_1, H^+ H_2$	$t(\chi_k^0), u(\chi_e^+), s(H^+), s(W^+)$
	$H^+ H_3$	$t(\chi_k^0), u(\chi_e^+), s(W^+)$
	$W^+ H_1, W^+ H_2$	$t(\chi_k^0), u(\chi_e^+), s(H^+), s(W^+)$
	$W^+ H_3$	$t(\chi_k^0), u(\chi_e^+), s(H^+)$
	$H^+ Z^0$	$t(\chi_k^0), u(\chi_e^+), s(H^+)$
	γH^+	$t(\chi_e^+), s(H^+)$
	$W^+ Z^0$	$t(\chi_k^0), u(\chi_e^+), s(W^+)$
	γW^+	$t(\chi_e^+), s(W^+)$
	$u \bar{d}$	$t(\tilde{d}_{L,R}), u(\tilde{u}_{L,R}), s(H^+), s(W^+)$
	$\nu \bar{\ell}$	$t(\tilde{\ell}_{L,R}), u(\tilde{\nu}_L), s(H^+), s(W^+)$

Chargino-chargino annihilation

Initial state	Final state	Feynman diagrams
$\chi_c^+ \chi_d^-$	$H_1 H_1, H_1 H_2, H_2 H_2, H_3 H_3$	$t(\chi_e^+), u(\chi_e^+), s(H_{1,2})$
	$H_1 H_3, H_2 H_3$	$t(\chi_e^+), u(\chi_e^+), s(H_3), s(Z^0)$
	$H^+ H^-$	$t(\chi_k^0), s(H_{1,2}), s(Z^0, \gamma)$
	$Z^0 H_1, Z^0 H_2$	$t(\chi_e^+), u(\chi_e^+), s(H_3), s(Z^0)$
	$Z^0 H_3$	$t(\chi_e^+), u(\chi_e^+), s(H_{1,2})$
	$H^+ W^-, W^+ H^-$	$t(\chi_k^0), s(H_{1,2,3})$
	$Z^0 Z^0$	$t(\chi_e^+), u(\chi_e^+), s(H_{1,2})$
	$W^+ W^-$	$t(\chi_k^0), s(H_{1,2}), s(Z^0, \gamma)$
	$\gamma \gamma$ (only for $c = d$)	$t(\chi_c^+), u(\chi_c^+)$
	$Z^0 \gamma$	$t(\chi_d^+), u(\chi_c^+)$
	$u \bar{u}$	$t(\tilde{d}_{L,R}), s(H_{1,2,3}), s(Z^0, \gamma)$
	$\nu \bar{\nu}$	$t(\tilde{\ell}_{L,R}), s(Z^0)$
	$\bar{d} \bar{d}$	$t(\tilde{u}_{L,R}), s(H_{1,2,3}), s(Z^0, \gamma)$
	$\bar{\ell} \bar{\ell}$	$t(\tilde{\nu}_L), s(H_{1,2,3}), s(Z^0, \gamma)$
$\chi_c^+ \chi_d^+$	$H^+ H^+$	$t(\chi_k^0), u(\chi_k^0)$
	$H^+ W^+$	$t(\chi_k^0), u(\chi_k^0)$
	$W^+ W^+$	$t(\chi_k^0), u(\chi_k^0)$

33.2.1.4 Squark-squark annihilation

Note: The tables below are not entirely up to date, more processes are included than shown in the tables.

We will here denote squarks as \tilde{q}_a^i and \tilde{q}_b^j where i and j are the family indices and a and b are the mass eigenstate indices (running from 1 to 2). k and l will also be used as family indices for

processes including more squarks. Colour indices are suppressed. \tilde{u}^i is used as a generic notation for any up-type squark where i denotes the family index. Down-type squarks are denoted analogously.

Note that we will not (except in rare occasions) show processes for $\tilde{\nu}$ and $\tilde{\ell}$ separately since they can easily be obtained from the squark processes by replacing \tilde{u} with $\tilde{\nu}$ and \tilde{d} with $\tilde{\ell}$ (and noting that we only have one mass eigenstate for the $\tilde{\nu}$). Also note that the $\tilde{\nu} - \tilde{\ell}$ -sector is assumed not to be flavour-changing.

$\tilde{d}_a^i \tilde{d}_b^{i*}$ annihilation

Initial state	Final state	Diagrams	Note
$\tilde{d}_a^i \tilde{d}_b^{i*}$	$\gamma\gamma, Z\gamma$	$t(\tilde{d}_{1,2}^i), u(\tilde{d}_{1,2}^i), p$	
$\tilde{d}_a^i \tilde{d}_b^{i*}$	ZZ	$t(\tilde{d}_{1,2}^i), u(\tilde{d}_{1,2}^i), p, s(H_1, H_2)$	
$\tilde{d}_a^i \tilde{d}_b^{i*}$	$W^- W^+$	$p, s(H_1, H_2, Z, \gamma), t(\tilde{u}_{1,2}^k)$	$k = 1, 2, 3$
$\tilde{d}_a^i \tilde{d}_b^{i*}$	ZH_2, ZH_1	$t(\tilde{d}_{1,2}^i), u(\tilde{d}_{1,2}^i), s(Z, H_3)$	
$\tilde{d}_a^i \tilde{d}_b^{i*}$	ZH_3	$t(\tilde{d}_{1,2}^i), u(\tilde{d}_{1,2}^i), s(H_1, H_2)$	
$\tilde{d}_a^i \tilde{d}_b^{i*}$	$\gamma H_2, \gamma H_1, \gamma H_3$	$t(\tilde{d}_{1,2}^i), u(\tilde{d}_{1,2}^i)$	
$\tilde{d}_a^i \tilde{d}_b^{i*}$	$H_2 H_2, H_1 H_1, H_1 H_2$	$t(\tilde{d}_{1,2}^i), u(\tilde{d}_{1,2}^i), p, s(H_1, H_2)$	
$\tilde{d}_a^i \tilde{d}_b^{i*}$	$H_2 H_3, H_1 H_3$	$s(Z, H_3), t(\tilde{d}_{1,2}^i), u(\tilde{d}_{1,2}^i)$	
$\tilde{d}_a^i \tilde{d}_b^{i*}$	$H_3 H_3$	$s(H_1, H_2), p, t(\tilde{d}_{1,2}^i), u(\tilde{d}_{1,2}^i)$	
$\tilde{d}_a^i \tilde{d}_b^{i*}$	$W^- H^+$	$s(H_1, H_2, H_3), t(\tilde{u}_{1,2}^k)$	$k = 1, 2, 3$
$\tilde{d}_a^i \tilde{d}_b^{i*}$	$H^- H^+$	$s(H_1, H_2, Z, \gamma), p, t(\tilde{u}_{1,2}^k)$	$k = 1, 2, 3$
$\tilde{d}_a^i \tilde{d}_b^{i*}$	$f\bar{f} (f \neq d^i)$	$s(H_1^*, H_2^*, H_3^*, Z, \gamma^*, g^\dagger), t(\chi_c^+)^{\dagger}$	†) $f = u^k (k = 1, 2, 3), \star$ Not for $f = \nu, \ddagger$) Only for squarks/quarks
$\tilde{d}_a^i \tilde{d}_b^{i*}$	$d^i \bar{d}^i$	$s(H_1, H_2, H_3, Z, \gamma, g^\dagger), t(\tilde{\chi}_k^0, \tilde{g}^\dagger)$	†) Only for squarks
$\tilde{d}_a^i \tilde{d}_b^{i*}$	Zg	$t(\tilde{d}_{1,2}^i), u(\tilde{d}_{1,2}^i), p$	Only for squarks
$\tilde{d}_a^i \tilde{d}_b^{i*}$	gg	$t(\tilde{d}_{1,2}^i), u(\tilde{d}_{1,2}^i), s(g), p$	Only for squarks
$\tilde{d}_a^i \tilde{d}_b^{i*}$	$g\gamma$	$t(\tilde{d}_{1,2}^i), u(\tilde{d}_{1,2}^i), p$	Only for squarks
$\tilde{d}_a^i \tilde{d}_b^{i*}$	gH_1, gH_2, gH_3	$t(\tilde{d}_{1,2}^i), u(\tilde{d}_{1,2}^i)$	Only for squarks

$\tilde{d}_a^i \tilde{d}_b^{j*}$ annihilation ($i \neq j$)

Initial state	Final state	Diagrams	Note
$\tilde{d}_a^i \tilde{d}_b^{j*}$	$W^+ W^-$	$t(\tilde{u}_{1,2}^k)^{\dagger}$	Not included at present
$\tilde{d}_a^i \tilde{d}_b^{j*}$	$W^+ H^-$	$t(\tilde{u}_{1,2}^k)^{\dagger}$	Not included at present
$\tilde{d}_a^i \tilde{d}_b^{j*}$	$H^+ H^-$	$t(\tilde{u}_{1,2}^k)^{\dagger}$	Not included at present
$\tilde{d}_a^i \tilde{d}_b^{*j}$	$d^i \bar{d}^j$	$t(\tilde{\chi}_k^0, \tilde{g}^\dagger)$	†) Only for squarks
$\tilde{d}_a^i \tilde{d}_b^{*j}$	$u^k \bar{u}^l$	$t(\tilde{\chi}_c^+)$	Only $k = i, l = j$ at present

$\tilde{d}_a^i \tilde{d}_b^i$ annihilation

Initial state	Final state	Diagrams	Note
$\tilde{d}_a^i \tilde{d}_b^i$	$d^i d^i$	$t(\tilde{\chi}_k^0, \tilde{g}^\dagger), u(\tilde{\chi}_k^0, \tilde{g}^\dagger)$	†) Only for squarks

$\tilde{d}_a^i \tilde{d}_b^j$ annihilation ($i \neq j$)

Initial state	Final state	Diagrams	Note
$\tilde{d}_a^i \tilde{d}_b^j$	$d^i d^j$	$t(\tilde{\chi}_k^0, \tilde{g}^\dagger)$	†) Only for squarks

$\tilde{u}_a^i \tilde{u}_b^{i*}$ annihilation

Initial state	Final state	Diagrams	Note
$\tilde{u}_a^i \tilde{u}_b^{i*}$	$\gamma\gamma^\dagger, Z\gamma^\dagger$	$t(\tilde{u}_{1,2}^i), u(\tilde{u}_{1,2}^i), p$	†) Not for $\tilde{\nu}$
$\tilde{u}_a^i \tilde{u}_b^{i*}$	ZZ	$t(\tilde{u}_{1,2}^i), u(\tilde{u}_{1,2}^i), p, s(H_1, H_2)$	
$\tilde{u}_a^i \tilde{u}_b^{i*}$	W^-W^+	$p, s(H_1, H_2, Z, \gamma^\dagger), u(\tilde{d}_{1,2}^k)$	$k = 1, 2, 3, \dagger$) Not for $\tilde{\nu}$
$\tilde{u}_a^i \tilde{u}_b^{i*}$	ZH_2, ZH_1	$t(\tilde{u}_{1,2}^i), u(\tilde{u}_{1,2}^i), s(Z, H_3^\dagger)$	†) Not for $\tilde{\nu}$
$\tilde{u}_a^i \tilde{u}_b^{i*}$	ZH_3	$t(\tilde{u}_{1,2}^i)^\dagger, u(\tilde{u}_{1,2}^i)^\dagger, s(H_1, H_2)$	†) Not for $\tilde{\nu}$
$\tilde{u}_a^i \tilde{u}_b^{i*}$	$\gamma H_2^\dagger, \gamma H_1^\dagger, \gamma H_3^\dagger$	$t(\tilde{u}_{1,2}^i), u(\tilde{u}_{1,2}^i)$	†) Not for $\tilde{\nu}$
$\tilde{u}_a^i \tilde{u}_b^{i*}$	H_2H_2, H_1H_1, H_1H_2	$t(\tilde{u}_{1,2}^i), u(\tilde{u}_{1,2}^i), p, s(H_1, H_2)$	
$\tilde{u}_a^i \tilde{u}_b^{i*}$	H_2H_3, H_1H_3	$s(Z, H_3^\dagger), t(\tilde{u}_{1,2}^i)^\dagger, u(\tilde{u}_{1,2}^i)^\dagger$	†) Not for $\tilde{\nu}$
$\tilde{u}_a^i \tilde{u}_b^{i*}$	H_3H_3	$s(H_1, H_2), p, t(\tilde{u}_{1,2}^i)^\dagger, u(\tilde{u}_{1,2}^i)^\dagger$	†) Not for $\tilde{\nu}$
$\tilde{u}_a^i \tilde{u}_b^{i*}$	W^-H^+	$s(H_1, H_2, H_3^\dagger), u(\tilde{d}_{1,2}^k)$	$k = 1, 2, 3, \dagger$) Not for $\tilde{\nu}$
$\tilde{u}_a^i \tilde{u}_b^{i*}$	H^+H^-	$s(H_1, H_2, Z, \gamma^\dagger), p, t(\tilde{d}_{1,2}^k)$	$k = 1, 2, 3, \dagger$) Not for $\tilde{\nu}$
$\tilde{u}_a^i \tilde{u}_b^{i*}$	$f\bar{f} (f \neq u^i)$	$s(H_1^\times, H_2^\times, H_3^\times, Z, \gamma^{\dagger\times}, g^\dagger), t(\chi_c^+)^\star$	†) Not for $\tilde{\nu}$, \star) If $f = d^k$ ($k = 1, 2, 3$), ‡) Only for squarks/quarks, \times) Not for ν
$\tilde{u}_a^i \tilde{u}_b^{i*}$	$u^i\bar{u}^i$	$s(H_1^\times, H_2^\times, H_3^\times, Z, \gamma^\times, g^\dagger), t(\tilde{\chi}_k^0, \tilde{g}^\dagger)$	\times) Not for ν , ‡) Only for squarks
$\tilde{u}_a^i \tilde{u}_b^{i*}$	Zg	$t(\tilde{u}_{1,2}^i), u(\tilde{u}_{1,2}^i), p$	Only for squarks
$\tilde{u}_a^i \tilde{u}_b^{i*}$	gg	$t(\tilde{u}_{1,2}^i), u(\tilde{u}_{1,2}^i), s(g), p$	Only for squarks
$\tilde{u}_a^i \tilde{u}_b^{i*}$	$g\gamma$	$t(\tilde{u}_{1,2}^i), u(\tilde{u}_{1,2}^i), p$	Only for squarks
$\tilde{u}_a^i \tilde{u}_b^{i*}$	gH_1, gH_2, gH_3	$t(\tilde{u}_{1,2}^i), u(\tilde{u}_{1,2}^i)$	Only for squarks

 $\tilde{u}_a^i \tilde{u}_b^{j*}$ annihilation ($i \neq j$)

Initial state	Final state	Diagrams	Note
$\tilde{u}_a^i \tilde{u}_b^{j*}$	W^+W^-	$t(\tilde{d}_{1,2}^k)^\dagger$	Not included at present, †) Not for $\tilde{\ell}$
$\tilde{u}_a^i \tilde{u}_b^{j*}$	W^+H^-	$t(\tilde{d}_{1,2}^k)^\dagger$	Not included at present, †) Not for $\tilde{\ell}$
$\tilde{u}_a^i \tilde{u}_b^{j*}$	H^+H^-	$t(\tilde{d}_{1,2}^k)^\dagger$	Not included at present, †) Not for $\tilde{\ell}$
$\tilde{u}_a^i \tilde{u}_b^{j*}$	$u^i\bar{u}^j$	$t(\tilde{\chi}_k^0, g^\dagger)$	†) Only for squarks
$\tilde{u}_a^i \tilde{u}_b^{j*}$	$d^k\bar{d}^l$	$t(\tilde{\chi}_c^+)$	Only $k = i, l = j$ at present

 $\tilde{u}_a^i \tilde{u}_b^i$ annihilation

Initial state	Final state	Diagrams	Note
$\tilde{u}_a^i \tilde{u}_b^i$	$u^i\bar{u}^i$	$t(\tilde{\chi}_k^0, \tilde{g}^\dagger), u(\tilde{\chi}_k^0, \tilde{g}^\dagger)$	†) Only for squarks

 $\tilde{u}_a^i \tilde{u}_b^j$ annihilation ($i \neq j$)

Initial state	Final state	Diagrams	Note
$\tilde{u}_a^i \tilde{u}_b^j$	$u^i\bar{u}^j$	$t(\tilde{\chi}_k^0, \tilde{g}^\dagger)$	†) Only for squarks

 $\tilde{u}_a^i \tilde{d}_b^{i*}$ annihilation

Initial state	Final state	Diagrams	Note
$\tilde{u}_a^i \tilde{d}_b^{i*}$	$H^+ H_1, H^+ H_2$	$t(\tilde{d}_{1,2}^i), u(\tilde{u}_{1,2}^i), p, s(W^+, H^+)$	
$\tilde{u}_a^i \tilde{d}_b^{i*}$	$H^+ H_3$	$t(\tilde{d}_{1,2}^i), u(\tilde{u}_{1,2}^i)^\dagger, p, s(W^+)$	†) Not for $\tilde{\ell}$
$\tilde{u}_a^i \tilde{d}_b^{i*}$	γH^+	$t(\tilde{u}_{1,2}^i)^\dagger, u(\tilde{d}_{1,2}^i), s(H^+)$	†) Not for $\tilde{\ell}$
$\tilde{u}_a^i \tilde{d}_b^{i*}$	$Z H^+$	$t(\tilde{u}_{1,2}^i), u(\tilde{d}_{1,2}^i), s(H^+)$	
$\tilde{u}_a^i \tilde{d}_b^{i*}$	$W^+ H_1, W^+ H_2$	$t(\tilde{d}_{1,2}^i), u(\tilde{u}_{1,2}^i), s(W^+, H^+)$	
$\tilde{u}_a^i \tilde{d}_b^{i*}$	$W^+ H_3$	$t(\tilde{d}_{1,2}^i), u(\tilde{u}_{1,2}^i)^\dagger, s(H^+)$	†) Not for $\tilde{\ell}$
$\tilde{u}_a^i \tilde{d}_b^{i*}$	$W^+ \gamma$	$t(\tilde{d}_{1,2}^i), u(\tilde{u}_{1,2}^i)^\dagger, s(W^+), p$	†) Not for $\tilde{\ell}$
$\tilde{u}_a^i \tilde{d}_b^{i*}$	$W^+ Z$	$t(\tilde{d}_{1,2}^i), u(\tilde{u}_{1,2}^i), s(W^+), p$	
$\tilde{u}_a^i \tilde{d}_b^{i*}$	$u^k \tilde{d}^l$	$s(H^+, W^+)^\star, t(\tilde{\chi}_m^0, \tilde{g}^\dagger) \delta^{ik} \delta^{il}$	†) Not for $\tilde{\ell}, \star$) Only $k = l$ at present
$\tilde{u}_a^i \tilde{d}_b^{i*}$	$W^+ g$	$t(\tilde{d}_{1,2}^i), u(\tilde{u}_{1,2}^i), p$	Only for squarks
$\tilde{u}_a^i \tilde{d}_b^{i*}$	$g H^+$	$t(\tilde{u}_{1,2}^i), u(\tilde{d}_{1,2}^i)$	Only for squarks

$\tilde{u}_a^i \tilde{d}_b^{j*}$ **annihilation** ($i \neq j$) For squarks we can have the following processes

Initial state	Final state	Diagrams	Note
$\tilde{u}_a^i \tilde{d}_b^{j*}$	$H^+ H_1, H^+ H_2$	$t(\tilde{d}_{1,2}^j), u(\tilde{u}_{1,2}^i), p, s(W^+, H^+)$	Not included at present
$\tilde{u}_a^i \tilde{d}_b^{j*}$	$H^+ H_3$	$t(\tilde{d}_{1,2}^j), u(\tilde{u}_{1,2}^i), p, s(W^+)$	Not included at present
$\tilde{u}_a^i \tilde{d}_b^{j*}$	$H^+ \gamma$	$t(\tilde{d}_{1,2}^j), u(\tilde{u}_{1,2}^i), s(H^+)$	Not included at present
$\tilde{u}_a^i \tilde{d}_b^{j*}$	$H^+ Z$	$t(\tilde{d}_{1,2}^j), u(\tilde{u}_{1,2}^i), s(H^+)$	Not included at present
$\tilde{u}_a^i \tilde{d}_b^{j*}$	$W^+ H_1, W^+ H_2$	$t(\tilde{d}_{1,2}^j), u(\tilde{u}_{1,2}^i), s(W^+, H^+)$	Not included at present
$\tilde{u}_a^i \tilde{d}_b^{j*}$	$W^+ H_3$	$t(\tilde{d}_{1,2}^j), u(\tilde{u}_{1,2}^i), s(H^+)$	Not included at present
$\tilde{u}_a^i \tilde{d}_b^{j*}$	$W^+ \gamma$	$t(\tilde{d}_{1,2}^j), u(\tilde{u}_{1,2}^i), s(W^+), p$	Not included at present
$\tilde{u}_a^i \tilde{d}_b^{j*}$	$W^+ Z$	$t(\tilde{d}_{1,2}^j), u(\tilde{u}_{1,2}^i), s(W^+), p$	Not included at present
$\tilde{u}_a^i \tilde{d}_b^{j*}$	$u^k \tilde{d}^l$	$s(H^+, W^+)^\dagger, t(\tilde{\chi}_m^0, \tilde{g}) \delta^{ik} \delta^{jl}$	†) Not included at present

whereas for sneutrinos and sleptons, we can only have the process

Initial state	Final state	Diagrams	Note
$\tilde{\nu}^i \tilde{\ell}_b^{j*}$	$\nu^i \bar{\ell}^j$	$t(\tilde{\chi}_k^0)$	

$\tilde{u}_a^i \tilde{d}_b^i$ **annihilation**

Initial state	Final state	Diagrams	Note
$\tilde{u}_a^i \tilde{d}_b^i$	$u^k d^l$	$t(\tilde{\chi}_m^0, \tilde{g}^\dagger) \delta^{ik} \delta^{il}, u(\tilde{\chi}_c^+)^\star$	†) Only for squarks, \star) Only $i = k = l$ at present

$\tilde{u}_a^i \tilde{d}_b^j$ **annihilation** ($i \neq j$)

Initial state	Final state	Diagrams	Note
$\tilde{u}_a^i \tilde{d}_b^j$	$u^k d^l$	$t(\tilde{\chi}_m^0, \tilde{g}^\dagger) \delta^{ik} \delta^{jl}, u(\tilde{\chi}_c^+)^\star$	†) Only for squarks, \times) For $\tilde{\nu} \bar{\ell}$ only when $i = l, j = k, \star$) Only included when $i = l, j = k$ at present

33.2.1.5 Squark-neutralino annihilation

Note: The tables below are not entirely up to date, more processes are included than shown in the tables.

We will here denote squarks as \tilde{u}_a^i and \tilde{d}_a^i where i is the family index and a is the mass eigenstate index (running from 1 to 2).

$\tilde{u}_a^i \tilde{\chi}_j^0$ annihilation

Initial state	Final state	Diagrams	Note
$\tilde{u}_a^i \tilde{\chi}_j^0$	γu^i	$s(u^i), t(\tilde{u}_{1,2}^i)$	Only for squarks
$\tilde{u}_a^i \tilde{\chi}_j^0$	$Z u^i$	$s(u^i), t(\tilde{u}_{1,2}^i), u(\tilde{\chi}_k^0)$	
$\tilde{u}_a^i \tilde{\chi}_j^0$	$H_1 u^i, H_2 u^i$	$s(u^i)^\dagger, t(\tilde{u}_{1,2}^i), u(\tilde{\chi}_k^0)$	†) Only for squarks
$\tilde{u}_a^i \tilde{\chi}_j^0$	$H_3 u^i$	$s(u^i)^\dagger, t(\tilde{u}_{1,2}^i)^\dagger, u(\tilde{\chi}_k^0)$	†) Only for squarks
$\tilde{u}_a^i \tilde{\chi}_j^0$	$W^+ d^k$	$s(u^i), t(\tilde{d}_{1,2}^k), u(\tilde{\chi}_c^+)$	$k = 1, 2, 3, \tilde{u}_a^{i*} \tilde{\chi}_j^0 \rightarrow$ $W^- \bar{d}^k$ in the code
$\tilde{u}_a^i \tilde{\chi}_j^0$	$H^+ d^k$	$s(u^i), t(\tilde{d}_{1,2}^k), u(\tilde{\chi}_c^+)$	$k = 1, 2, 3, \tilde{u}_a^{i*} \tilde{\chi}_j^0 \rightarrow H^- \bar{d}^k$ in the code
$\tilde{u}_a^i \tilde{\chi}_j^0$	$g u^i$	$s(u^i), t(\tilde{u}_{1,2}^i)$	

$\tilde{d}_a^i \tilde{\chi}_j^0$ annihilation

Initial state	Final state	Diagrams	Note
$\tilde{d}_a^i \tilde{\chi}_j^0$	γd^i	$s(d^i), t(\tilde{d}_{1,2}^i)$	
$\tilde{d}_a^i \tilde{\chi}_j^0$	$Z d^i$	$s(d^i), t(\tilde{d}_{1,2}^i), u(\tilde{\chi}_k^0)$	
$\tilde{d}_a^i \tilde{\chi}_j^0$	$H_1 d^i, H_2 d^i$	$s(d^i), t(\tilde{d}_{1,2}^i), u(\tilde{\chi}_k^0)$	
$\tilde{d}_a^i \tilde{\chi}_j^0$	$H_3 d^i$	$s(d^i), t(\tilde{d}_{1,2}^i), u(\tilde{\chi}_k^0)$	
$\tilde{d}_a^i \tilde{\chi}_j^0$	$W^- u^k$	$s(d^i), t(\tilde{u}_{1,2}^k), u(\tilde{\chi}_c^+)$	$k = 1, 2, 3$
$\tilde{d}_a^i \tilde{\chi}_j^0$	$H^- u^k$	$s(d^i), t(\tilde{u}_{1,2}^k), u(\tilde{\chi}_c^+)$	$k = 1, 2, 3$
$\tilde{d}_a^i \tilde{\chi}_j^0$	$g d^i$	$s(d^i), t(\tilde{d}_{1,2}^i)$	

33.2.1.6 Squark-chargino annihilation

Note: The tables below are not entirely up to date, more processes are included than shown in the tables.

We will here denote squarks as \tilde{q}_a^i where i is the family index and a is the mass eigenstate index (running from 1 to 2).

$\tilde{u}_a^i \tilde{\chi}_c^+$ annihilation

Initial state	Final state	Diagrams	Note
$\tilde{u}_a^i \tilde{\chi}_c^+$	$W^+ u^k$	$t(\tilde{d}_{1,2}^i), u(\tilde{\chi}_c^0) \delta^{ik}$	Only $k = l = i$ at present
$\tilde{u}_a^i \tilde{\chi}_c^+$	$H^+ u^k$	$t(\tilde{d}_{1,2}^i), u(\tilde{\chi}_c^0) \delta^{ik}$	Only $k = l = i$ at present

$\tilde{u}_a^{i*} \tilde{\chi}_c^+$ annihilation

Initial state	Final state	Diagrams	Note
$\tilde{u}_a^{i*} \tilde{\chi}_c^+$	$Z \bar{d}^k$	$s(\bar{d}^k), t(\tilde{u}_{1,2}^i), u(\tilde{\chi}_c^+)$	$k = 1, 2, 3$
$\tilde{u}_a^{i*} \tilde{\chi}_c^+$	$\gamma \bar{d}^k$	$s(\bar{d}^k), t(\tilde{u}_{1,2}^i)^\dagger, u(\tilde{\chi}_c^+)$	$k = 1, 2, 3, \dagger$ Only for squarks
$\tilde{u}_a^{i*} \tilde{\chi}_c^+$	$H_1 \bar{d}^k, H_2 \bar{d}^k$	$s(\bar{d}^k), t(\tilde{u}_{1,2}^i), u(\tilde{\chi}_c^+)$	$k = 1, 2, 3$
$\tilde{u}_a^{i*} \tilde{\chi}_c^+$	$H_3 \bar{d}^k$	$s(\bar{d}^k), t(\tilde{u}_{1,2}^i)^\dagger, u(\tilde{\chi}_c^+)$	$k = 1, 2, 3, \dagger$ Only for squarks
$\tilde{u}_a^{i*} \tilde{\chi}_c^+$	$W^+ \bar{u}^k$	$s(\bar{d}^l), u(\tilde{\chi}_c^0) \delta^{ik}$	Only $k = l = i$ at present
$\tilde{u}_a^{i*} \tilde{\chi}_c^+$	$H^+ \bar{u}^k$	$s(\bar{d}^l), u(\tilde{\chi}_c^0) \delta^{ik}$	Only $k = l = i$ at present
$\tilde{u}_a^{i*} \tilde{\chi}_c^+$	$g \bar{d}^k$	$s(\bar{d}^k), t(\tilde{u}_a^i)$	$k = 1, 2, 3$, only for squarks

 $\tilde{d}_a^i \tilde{\chi}_c^+$ annihilation

Initial state	Final state	Diagrams	Note
$\tilde{d}_a^i \tilde{\chi}_c^+$	$Z u^k$	$s(u^k), t(\tilde{d}_{1,2}^i), u(\tilde{\chi}_c^+)$	Only $k = i$ at present
$\tilde{d}_a^i \tilde{\chi}_c^+$	γu^k	$s(u^k)^\dagger, t(\tilde{d}_{1,2}^i), u(\tilde{\chi}_c^+)$	Only $k = i$ at present, \dagger Only for squarks
$\tilde{d}_a^i \tilde{\chi}_c^+$	$H_1 u^k, H_2 u^k$	$s(u^k)^\dagger, t(\tilde{d}_{1,2}^i), u(\tilde{\chi}_c^+)$	Only $k = i$ at present, \dagger Only for squarks
$\tilde{d}_a^i \tilde{\chi}_c^+$	$H_3 u^k$	$s(u^k)^\dagger, t(\tilde{d}_{1,2}^i), u(\tilde{\chi}_c^+)$	Only $k = i$ at present, \dagger Only for squarks
$\tilde{d}_a^i \tilde{\chi}_c^+$	$W^+ d^k$	$s(u^l), u(\tilde{\chi}_c^0) \delta^{ik}$	Only $k = l = i$ at present
$\tilde{d}_a^i \tilde{\chi}_c^+$	$H^+ d^k$	$s(u^l), u(\tilde{\chi}_c^0) \delta^{ik}$	Only $k = l = i$ at present
$\tilde{d}_a^i \tilde{\chi}_c^+$	$g u^k$	$s(u^k), t(\tilde{d}_a^i)$	Only $k = i$ at present, only for squarks

 $\tilde{d}_a^{i*} \tilde{\chi}_c^+$ annihilation

Initial state	Final state	Diagrams	Note
$\tilde{d}_a^{i*} \tilde{\chi}_c^+$	$W^+ \bar{d}^k$	$t(\tilde{u}_{1,2}^l), u(\tilde{\chi}_c^0) \delta^{ik}$	Only $k = l = i$ at present
$\tilde{d}_a^{i*} \tilde{\chi}_c^+$	$H^+ \bar{d}^k$	$t(\tilde{u}_{1,2}^l), u(\tilde{\chi}_c^0) \delta^{ik}$	Only $k = l = i$ at present

33.2.1.7 Degrees of freedom

We have to be careful with the internal degrees of freedom, g , of the particles. We can either treat e.g. a χ_i^+ and a χ_i^- as two separate particles with two degrees of freedom each, or we can treat them as one particle χ_i^\pm with four degrees of freedom. The latter approach has an advantage that we simplify our expressions for the effective annihilation cross sections when coannihilations are needed. Hence, we use that approach here. For a more detailed discussion about this, see Section 33.12.2.

33.2.2 Annihilation routines - general remarks

The annihilation cross section routines is divided into several parts, mostly for historical reasons. The layout is roughly as follows:

`src/an` Here we keep the main routines for both neutralino- neutralino annihilation cross sections and the effective annihilation cross section in the relic density calculations. The steering routines for neutralino and chargino coannihilations are also kept here.

`src/anstu` Here keep the t -, u - and s - diagram expressions for fermion-fermion coannihilations (i.e. neutralino and chargino coannihilations).

`src/as` Here all the coannihilation cross sections including sfermions are kept.

We will here describe the `src/an`-routines.

33.2.2.1 General routines

The general routine to call for an effective annihilation cross section (to be used for relic density calculations) is `dsanwx`, which returns the invariant annihilation rate (integrated over $\cos\theta$). The actual cross section, differential in $\cos\theta$ is calculated by `dsandwdcos` which includes all the coannihilations needed. This is set up in `mssm/rd/dsrdparticles` (typically called from `dsrdomega` in the core library) which determines which coannihilating particles to include.

For other applications where the annihilation rate is needed, e.g. annihilation in the galactic halo, one can call the specific annihilation rate routine directly. The main one is `dsandwdcosnn` for neutralino-neutralino annihilation. To simplify this task, we supply a routine `dssigmav0` which calls `dsandwdcosnn` for neutralino-neutralino annihilation at zero relative velocity and returns the result, either as the total annihilation cross section, or the cross section for a specific channel. See the header of `dssigmav` for details.

33.2.2.2 Neutralino and chargino (co)annihilation cross sections

The routines `dsandwdcosnn`, `dsandwdcosc` and `dsandwdcoscc` calculate the annihilation cross sections (returning the invariant annihilation rate) for neutralino-neutralino, neutralino-chargino and chargino-chargino annihilations. Which particles the cross section is calculated for is given by particle indices as defined in `mssm/include/dsmssm.h`.

All the annihilation routines return the invariant rate instead of the cross section. The invariant annihilation rate between particle i and j is defined as

$$W_{ij} = 4p_{ij}\sqrt{s}\sigma_{ij} = 4\sigma_{ij}\sqrt{(p_i \cdot p_j)^2 - m_i^2 m_j^2} = 4E_i E_j \sigma_{ij} v_{ij}. \quad (33.1)$$

See Chapter 22 for more details.

33.3 mssm/an_1l: Annihilation cross sections (1-loop)

33.3.1 Annihilation cross sections at 1-loop – general

The annihilation cross sections at 1-loop that we have implemented in DarkSUSY are those to $\gamma\gamma$, $Z\gamma$ and gg . The derivation of these is described in the works [26, 27]. Let us mention that these one-loop expressions formally violate unitarity for diagrams with electroweak gauge-boson exchange because they do not take into account the nonperturbative effects related to multiple electroweak gauge boson exchange (“Sommerfeld effect”) as described in [188, 189]. In practice, one can still trust the result for sub-TeV neutralinos – but should keep in mind that the implemented cross sections become unphysical in the limit $m_W/m_\chi \rightarrow 0$.

To see how these routines are called, see the file `src_models/mssm/an/dsandwdcosnn.f` where the $\gamma\gamma$, $Z\gamma$ and gg contributions are added to the annihilation cross section at the end.

33.4 mssm/an_ib: Internal bremsstrahlung

33.4.1 Internal Bremsstrahlung (IB) – theory

Whenever WIMPs annihilate into pairs of charged particles $X\bar{X}$, this process will inevitably be accompanied by internal bremsstrahlung (IB), i.e. the emission of an additional photon in the final state (note that in contrast to ordinary, or external, bremsstrahlung no external electromagnetic field is required for the emission of the photon). In many cases, IB photons completely dominate

the annihilation spectrum at the highest accessible energies. The resulting, characteristic sharp step at an energy corresponding to the dark matter particle's mass, often accompanied by a bump-like feature at slightly smaller energies, is a spectral signature that is hard to mimic by astrophysical sources and in that respect similar to monochromatic photons. In fact, being an $\mathcal{O}(\alpha_{\text{em}})$ correction to the tree-level annihilation rate, one would generically expect IB photons to be more copiously produced than the loop-suppressed [i.e. $\mathcal{O}(\alpha_{\text{em}}^2)$] monochromatic photons; this has been confirmed in [28] for a large part of the supersymmetric parameter space, not the least due to the appearance of efficient enhancement mechanisms.

33.4.1.1 General considerations

For didactic purposes, one may follow [28] and distinguish between photons directly radiated off the external legs (*final state radiation*, FSR) and photons radiated from virtual charged particles (*virtual internal bremsstrahlung*, VIB). The IB photons are thus the total contribution from both FSR and VIB photons.

For relativistic charged final states, FSR diagrams are always dominated by photons emitted *collinearly* with X or \bar{X} . This is a purely kinematical effect and related to the fact that the propagator of the corresponding outgoing particle,

$$D(p) \propto ((k+p)^2 - m_X^2)^{-1}, \quad (33.2)$$

diverges in this situation. Here, k and p denote the momenta of the photon and the outgoing particle, respectively. The resulting photon spectrum turns out to be of a universal form, almost independent of the underlying particle physics model (see, e.g., [190]):

$$\frac{dN_{\gamma, \text{FSR}}^{X\bar{X}}}{dx} \approx \frac{\alpha Q_X^2}{\pi} \mathcal{F}_X(x) \log\left(\frac{s(1-x)}{m_X^2}\right). \quad (33.3)$$

Here, Q_X and m_X are the electric charge and mass of X ; the splitting function $\mathcal{F}(x)$ depends only on the spin of the final state particles and takes the form

$$\mathcal{F}_{\text{fermion}}(x) = \frac{1 + (1-x)^2}{x} \quad (33.4)$$

for fermions and

$$\mathcal{F}_{\text{boson}}(x) = \frac{1-x}{x} \quad (33.5)$$

for bosons. Due to the logarithmic enhancement that becomes apparent in Eq. (33.3), FSR photons are often the main source for IB. A prominent example where FSR in this universal form not only dominates IB but in fact the total gamma-ray spectrum from WIMP annihilations, is the case of Kaluza-Klein dark matter [191].

In general, one can single out two situations where photons emitted from virtual charged particles may give an even more important contribution to the total IB spectrum than FSR: i) the three-body final state $X\bar{X}\gamma$ satisfies a symmetry of the initial state that cannot be satisfied by the two-body final state $X\bar{X}$ or ii) X is a boson and the annihilation into $X\bar{X}$ is dominated by t -channel diagrams, with the t -channel particle almost degenerate in mass with the annihilating WIMP. In contrast to FSR, the contribution from VIB photons can not be given in a model-independent way but is very sensitive to the underlying short-distance physics. For more details, see [28].

33.4.1.2 IB from neutralino annihilations

For supersymmetric dark matter annihilations, the relevant final states for IB are W^+W^- , $W^\pm H^\mp$, H^+H^- and $f\bar{f}$; both of the situations just mentioned above can arise, and *VIB* contributions become important in considerable regions of the parameter space.

Let us first note that for neutralino annihilations, in contrast to the situation for, e.g., Kaluza-Klein dark matter, we cannot in general expect very large FSR contributions. This is because the lightest charged final states, for which the logarithmic enhancement shown in Eq. (33.3) would be most effective, are fermionic and therefore strongly helicity suppressed. Fermion final states containing an additional photon, however, are not subject to such a suppression [192]. In the limit of vanishing fermion mass, and assuming that both corresponding sfermions have the same mass, the photon multiplicity is given by [28]

$$\frac{dN^{f^+f^-}}{dx} = \alpha_{\text{em}} Q_f^2 \frac{|\tilde{g}_R|^4 + |\tilde{g}_L|^4}{64\pi^2} \left(m_\chi^2 \langle \sigma v \rangle_{\chi\chi \rightarrow f\bar{f}} \right)^{-1} \quad (33.6)$$

$$\times (1-x) \left\{ \frac{4x}{(1+\mu)(1+\mu-2x)} - \frac{2x}{(1+\mu-x)^2} - \frac{(1+\mu)(1+\mu-2x)}{(1+\mu-x)^3} \log \frac{1+\mu}{1+\mu-2x} \right\},$$

where $\mu \equiv m_{\tilde{f}_R}^2/m_\chi^2 = m_{\tilde{f}_L}^2/m_\chi^2$ and $\tilde{g}_R P_L$ ($\tilde{g}_L P_R$) denotes the coupling between neutralino, fermion and right-handed (left-handed) sfermion. In the above expression, a large factor m_χ^2/m_f^2 due to the lifted helicity suppression (from $\langle \sigma v \rangle_{\chi\chi \rightarrow f\bar{f}} \propto m_f^2 m_\chi^{-4}$) appears, and another enhancement at high photon energies for sfermions degenerate with the neutralino.

For large neutralino masses $m_\chi \gg m_W$ and charginos almost degenerate with the neutralino, W^+W^- and $W^\pm H^\mp$ final states are affected by the second of the mechanisms for VIB enhancement that were discussed in the previous subsection. Of these two channels, IB from W^+W^- nearly always dominates; for pure Higgsinos (or Winos), the resulting photon multiplicity in this limit is well approximated by [28, 193]:

$$\frac{dN^{W^+W^-}}{dx} \approx \frac{\alpha_{\text{em}}}{\pi} \frac{4(1-x+x^2)^2}{(1-x+\epsilon/2)x} \left[\log \left(2 \frac{1-x+\epsilon/2}{\epsilon} \right) - 1/2 + x - x^3 \right], \quad (33.7)$$

where $\epsilon \equiv m_W/m_\chi$.

Charged Higgs pairs H^+H^- , finally, provide yet another interesting example where the two-body final state is not allowed due to symmetry restrictions; in this case, in the limit $v \rightarrow 0$, enforced by CP conservation. The annihilation into $H^+H^-\gamma$, on the other hand, is possible. However, since charged Higgs bosons in most models have considerably larger masses than gauge bosons, they are expected to give negligible IB contributions compared to the latter.

33.4.1.3 The implementation in DarkSUSY

Let us now briefly describe how IB is implemented in DarkSUSY. The total gamma-ray spectrum from WIMP annihilations is given by

$$\frac{dN_{\gamma,\text{tot}}}{dx} = \sum_f B_f \left(\frac{dN_{\gamma,\text{sec}}^f}{dx} + \frac{dN_{\gamma,\text{IB}}^f}{dx} + \frac{dN_{\gamma,\text{line}}^f}{dx} \right), \quad (33.8)$$

where B_f denotes the branching ratio into the annihilation channel f . The first term encodes the contribution from secondary photons, mainly produced through the decay of neutral pions, as described in Chapter 11. We recall here that these contributions are included by using the Monte Carlo code Pythia [10] to simulate the decay of a hypothetical particle with mass $2m_\chi$ and user-specified branching ratios B_f . In this way, also FSR associated to this decay is automatically included in $dN_{\gamma,\text{sec}}^f/dx$ (the main contribution here comes from photons directly radiated off the external legs, but also photons radiated from other particles in the decay cascade are taken into account).

Of course, IB from the decay of such a hypothetical particle cannot in general be expected to show the same characteristics as IB from the actual annihilation of two WIMPs and for this reason an additional term $dN_{\gamma,\text{IB}}^f/dx$ is included that accounts for the difference between the full

IB contribution and the FSR part already taken into account for by Pythia. For details regarding the implemented procedure of separating these two contributions in a consistent way, see [28]. The contributions $dN_{\gamma, \text{IB}}^f/dx$, in contrast to $dN_{\gamma, \text{sec}}^f/dx$, are generically highly model-dependent. At the moment, they are fully implemented only for neutralino annihilations and included by default. However, one may easily switch to a user-defined contribution, choose to neglect IB completely or, for comparison, to only include the FSR part by replacing the call to `dsibyield` in `mssm/an_yield/dsanyield`. For example, a version of `dsibyield` containing only the FSR part of the yield is provided in `mssm/an_ib/dsanyield_fsr`.

For the supersymmetric case, the full expressions for all relevant three-body final states are implemented, i.e. not just the approximations given in the last subsection, which only apply to the limits described there. For performance reasons, IB is only included when virtual t -channel particles are sufficiently degenerate in mass that large IB contributions to the total spectrum can be expected. With a call to `dsIBset`, this default behaviour can be customized. Finally, radiative corrections to the annihilation into charged particles $X\bar{X}$ of course also change the number of $X\bar{X}$ pairs per annihilation and thus the corresponding yield of particles in the further decay of the annihilation products. At the moment, apart from the photon yield, only the IB positron yield is implemented in DarkSUSY. By default, only the annihilation into the dominating channel $e^+e^-\gamma$ is taken into account in this case; again, this behaviour can be modified by a call to `dsIBset`.

33.5 mssm/an_sf: Annihilation cross sections (with sfermions)

33.5.1 Annihilation cross sections with sfermions – general

In this directory, all the (co)annihilation cross sections involving one or more sfermions in the initial state are calculated. The code here is based upon the work described in [14]. All the cross sections are calculated with `Form` and converted to Fortran with a script `form2f` [187].

The main routines here are

Routine	Purpose
<code>dsasdwdcossf</code>	Calculates the invariant annihilation rate between two sfermions in the initial state.
<code>dsasdwdcossfchi</code>	Calculates the invariant annihilation rate between one sfermion and one fermion (neutralino or chargino) in the initial state.

33.6 mssm/an_stu: t , u and s diagrams for ff -annihilation

33.6.1 Annihilation amplitudes for fermion-fermion annihilation

In this directory, all the helicity amplitudes needed for neutralino-neutralino, neutralino-chargino and chargino-chargino annihilation are calculated. The helicity amplitudes have been calculated with general expressions for vertices, masses etc. in `Reduce` and converted to Fortran files. The calculation of these are described in more detail in [13].

Each routine here adds the contribution to the helicity amplitudes from one particular diagram and the sum over contributed diagrams is done in the routines `an/dsandwdcosnn`, `an/dsandwdcosc` and `an/dsandwdcoscc`. The naming convention for the routines here is the following: The first part of the routine name is `dsan` to indicate that they deal with annihilations in DarkSUSY. The next character tells which kind of process it is s -, t - or u -channel and the next two characters tell which initial state particles we have (**f** for fermion), the next character is the kind of propagating particle (**f** for fermion, **s** for scalar and **v** for vector boson), and finally, the last two characters tell the

kind of final state particles. So, to take an example, the routine `dsansffsv` calculates the helicity amplitudes for annihilation of two fermions to two vector bosons via s -channel exchange of a scalar. There are also a few special cases (routines ending in `ex` or `in`) for diagrams with clashing arrows.

33.7 mssm/dd: Direct detection

33.7.1 Direct detection – theory

NOTE: The description below is outdated. DarkSUSY now includes much better form factors.

If neutralinos are indeed the CDM needed on galaxy scales and larger, there should be a substantial flux of these particles in the Milky Way halo. Since the interaction strength is essentially given by the same weak couplings as, e.g., for neutrinos there is a non-negligible chance of detecting them in low-background counting experiments [194]. Due to the large parameter space of MSSM, even with the simplifying assumptions above, there is a rather wide span of predictions for the event rate in detectors of various types. It is interesting, however, that the models giving the largest rates are already starting to be probed by present direct detection experiments [25, 195].

The rate for direct detection of galactic neutralinos, integrated over deposited energy assuming no energy threshold, is

$$R = \sum_i N_i n_\chi \langle \sigma_{i\chi} v \rangle, \quad (33.9)$$

where N_i is the number of nuclei of species i in the detector, n_χ is the local galactic neutralino number density, $\sigma_{i\chi}$ is the neutralino-nucleus elastic cross section, and the angular brackets denote an average over v , the neutralino speed relative to the detector as described in Chapter 17.

In DarkSUSY, the basic quantities computed are the neutralino-nucleon cross sections, which are free of complications related to nuclear structure, and various experimental details like energy threshold, efficiencies etc. However, as a crude estimate of the expected rates in realistic detectors, the total neutralino-nucleus scattering rates can be obtained for ^{76}Ge , Al_2O_3 (sapphire) and NaI. Usually, it is the spin-independent interaction that gives the most important contribution in target materials such as Na, Cs, Ge, I, or Xe, due to the enhancement caused by the coherence of all nucleons in the target nucleus.

The neutralino-nucleus elastic cross section can be written as

$$\sigma_{i\chi} = \frac{1}{4\pi v^2} \int_0^{4m_{i\chi}^2 v^2} dq^2 G_{i\chi}^2(q^2), \quad (33.10)$$

where $m_{i\chi}$ is the neutralino-nucleus reduced mass, q is the momentum transfer and $G_{i\chi}(q^2)$ is the effective neutralino-nucleus vertex. We write

$$G_{i\chi}^2(q^2) = A_i^2 F_S^2(q^2) G_S^2 + 4\Lambda_i^2 F_A^2(q^2) G_A^2, \quad (33.11)$$

which shows the coherent enhancement factor A_i^2 for the spin-independent cross section (often Λ_i^2 is written as $\lambda^2 J(J+1)$). We assume gaussian nuclear form factors [119]

$$F_S(q^2) = F_A(q^2) = \exp(-q^2 R_i^2 / 6\hbar^2), \quad (33.12)$$

$$R_i = (0.3 + 0.89A_i^{1/3}) \text{fm}, \quad (33.13)$$

which should provide us with a good approximation of the integrated detection rate [196, 197], in which we are only interested. (To obtain the differential rate with reasonable accuracy, better approximations are needed [198].)

Using heavy-squark effective lagrangians [199, 200, 201, 103, 202], we get

$$G_S = \sum_{q=u,d,s,c,b,t} \langle \bar{q}q \rangle \left(\sum_{h=H_1, H_2} \frac{g_{h\chi\chi} g_{hqq}}{m_h^2} - \frac{1}{2} \sum_{k=1}^6 \frac{g_{L\bar{q}_k\chi q} g_{R\bar{q}_k\chi q}}{m_{\bar{q}_k}^2} \right) \quad (33.14)$$

and

$$G_A = \sum_{q=u,d,s} \Delta q \left(\frac{g_{Z\chi\chi} g_{Zqq}}{m_Z^2} + \frac{1}{8} \sum_{k=1}^6 \frac{g_{L\bar{q}_k\chi q}^2 + g_{R\bar{q}_k\chi q}^2}{m_{\bar{q}_k}^2} \right). \quad (33.15)$$

The g 's are elementary vertices involving the particles indicated by the indices, and they read

$$g_{h\chi\chi} = \begin{cases} (g_{Z\chi_2} - g_y Z_{\chi_1}) (-Z_{\chi_3} \cos \alpha + Z_{\chi_4} \sin \alpha), & \text{for } H_1, \\ (g_{Z\chi_2} - g_y Z_{\chi_1}) (Z_{\chi_3} \sin \alpha + Z_{\chi_4} \cos \alpha), & \text{for } H_2, \end{cases} \quad (33.16)$$

$$g_{hqq} = \begin{cases} -Y_q \cos \alpha / \sqrt{2}, & \text{for } H_1, \\ +Y_q \sin \alpha / \sqrt{2}, & \text{for } H_2, \end{cases} \quad (33.17)$$

$$g_{Z\chi\chi} = \frac{g}{2 \cos \theta_W} (Z_{\chi_3}^2 - Z_{\chi_4}^2) \quad (33.18)$$

$$g_{Zqq} = -\frac{g}{2 \cos \theta_W} T_{3q}, \quad (33.19)$$

$$g_{L\bar{q}_k\chi q} = g_{LL} \Gamma_{QL}^{kq} + g_{RL} \Gamma_{QR}^{kq}, \quad (33.20)$$

$$g_{R\bar{q}_k\chi q} = g_{LR} \Gamma_{QL}^{kq} + g_{RR} \Gamma_{QR}^{kq}, \quad (33.21)$$

with

$$g_{LL} = -\frac{1}{\sqrt{2}} \left(T_{3q} g_{Z\chi_2} + \frac{1}{3} g_y Z_{\chi_1} \right), \quad (33.22)$$

$$g_{RR} = \sqrt{2} e_q g_y Z_{\chi_1}, \quad (33.23)$$

$$g_{LR} = g_{RL} = \begin{cases} -Y_q Z_{\chi_3}, & \text{for } q = u, c, t, \\ -Y_q Z_{\chi_4}, & \text{for } q = d, s, b, \end{cases} \quad (33.24)$$

and

$$Y_q = \begin{cases} m_q / v_2, & \text{for } q = u, c, t, \\ m_q / v_1, & \text{for } q = d, s, b. \end{cases} \quad (33.25)$$

Defining ($N = n, p$)

$$f_{Tq}^N \equiv \frac{\langle N | m_q \bar{q}q | N \rangle}{m_N}, \quad (33.26)$$

we take in Darksusy the numerical values [203]

$$\begin{aligned} f_{Tu}^p &= 0.023, & f_{Td}^p &= 0.034, \\ f_{Tc}^p &= 0.0595, & f_{Ts}^p &= 0.14, \\ f_{Tt}^p &= 0.0595, & f_{Tb}^p &= 0.0595 \end{aligned} \quad (33.27)$$

and

$$\begin{aligned} f_{Tu}^n &= 0.019, & f_{Td}^n &= 0.041, \\ f_{Tc}^n &= 0.0592, & f_{Ts}^n &= 0.14, \\ f_{Tt}^n &= 0.0592, & f_{Tb}^n &= 0.0592. \end{aligned} \quad (33.28)$$

For the quark contributions to the nucleon spin we take [204]

$$\Delta u = 0.77, \quad \Delta d = -0.40, \quad \Delta s = -0.12. \quad (33.29)$$

However, the older set of data [205]

$$\Delta u = 0.77, \quad \Delta d = -0.49, \quad \Delta s = -0.15 \quad (33.30)$$

can optionally be used.

Moreover, we take for the Λ factors

$$\Lambda_{A1}^2 = 0.087, \quad \Lambda_{Na}^2 = 0.041 \quad \text{and} \quad \Lambda_I^2 = 0.007, \quad (33.31)$$

according to the odd-group model [206].

33.8 mssm/ge:

General SUSY model setup: masses, vertices etc

33.8.1 Supersymmetric model

We will here review the definition of the MSSM as given in [1].

33.8.1.1 Parameters

In our notation, the superpotential and the soft supersymmetry-breaking scalar potential minimal supersymmetric standard model (MSSM) with R-parity conservation [159, 160, 207] read respectively

$$\begin{aligned} W &= \epsilon_{ij} \left(-\hat{\mathbf{e}}_R^* \mathbf{h}_E \hat{\mathbf{l}}_L^i \hat{H}_1^j - \hat{\mathbf{d}}_R^* \mathbf{h}_D \hat{\mathbf{q}}_L^i \hat{H}_1^j + \hat{\mathbf{u}}_R^* \mathbf{h}_U \hat{\mathbf{q}}_L^i \hat{H}_2^j - \mu \hat{H}_1^i \hat{H}_2^j \right), \\ V_{\text{soft}} &= \epsilon_{ij} \left(-\tilde{\mathbf{e}}_R^* \mathbf{A}_E \mathbf{h}_E \tilde{\mathbf{l}}_L^i H_1^j - \tilde{\mathbf{d}}_R^* \mathbf{A}_D \mathbf{h}_D \tilde{\mathbf{q}}_L^i H_1^j + \tilde{\mathbf{u}}_R^* \mathbf{A}_U \mathbf{h}_U \tilde{\mathbf{q}}_L^i H_2^j - B \mu H_1^i H_2^j \right. \\ &\quad \left. + \text{h.c.} \right) \\ &\quad + H_1^{i*} m_1^2 H_1^i + H_2^{i*} m_2^2 H_2^i \\ &\quad + \tilde{\mathbf{q}}_L^{i*} \mathbf{M}_Q^2 \tilde{\mathbf{q}}_L^i + \tilde{\mathbf{l}}_L^{i*} \mathbf{M}_L^2 \tilde{\mathbf{l}}_L^i + \tilde{\mathbf{u}}_R^{i*} \mathbf{M}_U^2 \tilde{\mathbf{u}}_R^i + \tilde{\mathbf{d}}_R^{i*} \mathbf{M}_D^2 \tilde{\mathbf{d}}_R^i + \tilde{\mathbf{e}}_R^{i*} \mathbf{M}_E^2 \tilde{\mathbf{e}}_R^i. \end{aligned} \quad (33.32)$$

Here i and j are SU(2) indices ($\epsilon_{12} = +1$), \mathbf{h} 's, \mathbf{A} 's and \mathbf{M} 's are 3×3 matrices in generation space, and the other boldface letters are vectors in generation space.

The current version of DarkSUSY uses only a restricted set of parameters. Namely the number of free parameters (a grand total of 124 [208]) is reduced by setting the off-diagonal elements of the \mathbf{A} 's and \mathbf{M} 's to zero and imposing CP conservation (except in the CKM matrix).

33.8.1.2 Mass spectrum

For easy reference, we now give the particle mass matrices, together with our convention for the mixing matrices.

Concerning the Higgs sector, we choose as independent parameters $\tan \beta$ and the mass m_A of the CP-odd Higgs boson. The code provides five options for the calculation of the Higgs masses:

`higloop=0`: tree level formulas; `higloop=1`: the effective potential approach in [209, 210, 211] (correcting the sign of μ in eq. (4) of [211]); `higloop=2`: the effective potential approach in [212] with addition of D-terms and correction of some signs and numerical factors; `higloop=3`: the analytical approximations to the RGE-improved effective potential in [213]; `higloop=4`: the pole mass calculation in [214]; `higloop=5`: FeynHiggs (requires FeynHiggs to be installed, default) [215]

The masses of the Higgs bosons are obtained from

$$\mathcal{M}_H^2 = \begin{pmatrix} m_Z^2 \cos^2 \beta + m_A^2 \sin^2 \beta + \Delta_{11} & -\sin \beta \cos \beta (m_Z^2 + m_A^2) + \Delta_{12} \\ -\sin \beta \cos \beta (m_Z^2 + m_A^2) + \Delta_{21} & m_Z^2 \sin^2 \beta + m_A^2 \cos^2 \beta + \Delta_{22} \end{pmatrix} \quad (33.34)$$

$$m_{H^\pm}^2 = m_A^2 + m_W^2 + \Delta_\pm. \quad (33.35)$$

Channel i=	Higgs boson			
	H_1^0 j=1	H_2^0 j=2	H_3^0 j=3	H^+ j=4
1	$c\bar{c}$	$c\bar{c}$	$c\bar{c}$	$u\bar{d}$
2	$b\bar{b}$	$b\bar{b}$	$b\bar{b}$	$u\bar{s}$
3	$t\bar{t}$	$t\bar{t}$	$t\bar{t}$	$u\bar{b}$
4	$\tau^+\tau^-$	$\tau^+\tau^-$	$\tau^+\tau^-$	$c\bar{d}$
5	W^+W^-	W^+W^-	–	$c\bar{s}$
6	Z^0Z^0	Z^0Z^0	–	$c\bar{b}$
7	–	$H_1^0H_1^0$	–	$t\bar{d}$
8	$H_2^0H_2^0$	–	–	$t\bar{s}$
9	$H_3^0H_3^0$	$H_3^0H_3^0$	–	$t\bar{b}$
10	H^+H^-	H^+H^-	–	$\nu_e e^+$
11	–	–	ZH_1^0	$\nu_\mu\mu^+$
12	–	–	ZH_2^0	$\nu_\tau\tau^+$
13	ZH_3^0	ZH_3^0	–	$W^+H_1^0$
14	W^+H^-/W^-H^+	W^+H^-/W^-H^+	W^+H^-/W^-H^+	$W^+H_2^0$
15	$\mu^+\mu^-$	$\mu^+\mu^-$	$\mu^+\mu^-$	$W^+H_3^0$
16	$s\bar{s}$	$s\bar{s}$	$s\bar{s}$	–
17	gg	gg	gg	–
18	$\gamma\gamma$	$\gamma\gamma$	$\gamma\gamma$	–
19	$Z^0\gamma$	$Z^0\gamma$	$Z^0\gamma$	–
20	$\tilde{f}\tilde{f}'$	$\tilde{f}\tilde{f}'$	$\tilde{f}\tilde{f}'$	$\tilde{f}\tilde{f}'$

Table 33.2: Higgs partial widths $\text{hdwidth}(i,j)$. Index i refers to the decay channel and index j to the Higgs boson. All widths are given in GeV. Note that typically we have that $m_{H_2} < m_{H_3} < m_{H^+} < m_{H_1}$ so many of these decay channels are not kinematically allowed, but included for completeness. If the HDECAY interface is used, the channels where $m_{H_2} < m_{H_3} < m_{H^+} < m_{H_1}$ is not satisfied are not included. Channels 16–19 are only included if HDECAY is used.

The quantities Δ_{ij} and Δ_\pm are the one-loop radiative corrections, calculated according to the value of `higloop` as described above. Diagonalization of \mathcal{M}_H^2 gives the two CP-even Higgs boson masses, $m_{H_{1,2}}$, and their mixing angle α ($-\pi/2 < \alpha < 0$). For `higloop`=4, the pole masses are then obtained solving $m_{H_i}^{2\text{pole}} = m_{H_i}^2 + \Pi_{ii}(m_{H_i}^{2\text{pole}}) - \Pi_{ii}(0)$, where $\Pi_{ii}(p^2)$ is $H_i H_i$ the self-energy. In this case, m_{H_3} is the pole mass and m_A is the running mass.

The Higgs widths are calculated at tree level, but with QCD corrections. The decays to supersymmetric particles are also included in the total width, so the sum of the partial widths in Table 33.2 does not necessarily sum up to the total width given in `width(k)`. The loop corrections are also available via an interface to HDECAY.

The neutralinos $\tilde{\chi}_i^0$ are linear combinations of the neutral gauginos \tilde{B} , \tilde{W}_3 and of the neutral higgsinos \tilde{H}_1^0 , \tilde{H}_2^0 . In this basis, we write their mass matrix as

$$\mathcal{M}_{\tilde{\chi}_{1,2,3,4}^0} = \begin{pmatrix} M_1 & 0 & -m_Z s_W c_\beta & +m_Z s_W s_\beta \\ 0 & M_2 & +m_Z c_W c_\beta & -m_Z c_W s_\beta \\ -m_Z s_W c_\beta & +m_Z c_W c_\beta & \delta_{33} & -\mu \\ +m_Z s_W s_\beta & -m_Z c_W s_\beta & -\mu & \delta_{44} \end{pmatrix}, \quad (33.36)$$

with $c_W = \cos\theta_W$, $s_W = \sin\theta_W$, $c_\beta = \cos\beta$, and $s_\beta = \sin\beta$. Here δ_{33} and δ_{44} are radiative corrections important when two higgsinos are close in mass. Their explicit expressions are from ref. [165]. To neglect these radiative corrections set `neuloop`=0 instead of `neuloop`=1 (default). The neutralino mass eigenstates are written as

$$\tilde{\chi}_i^0 = N_{i1}\tilde{B} + N_{i2}\tilde{W}_3 + N_{i3}\tilde{H}_1^0 + N_{i4}\tilde{H}_2^0. \quad (33.37)$$

The phases of N_{ij} are chosen so that the neutralino masses $m_{\tilde{\chi}_i^0} \geq 0$.

The charginos are linear combinations of the charged gauge bosons \tilde{W}^\pm and of the charged higgsinos $\tilde{H}_1^-, \tilde{H}_2^+$. Their mass matrix,

$$\mathcal{M}_{\tilde{\chi}^\pm} = \begin{pmatrix} M_2 & \sqrt{2}m_W \sin \beta \\ \sqrt{2}m_W \cos \beta & \mu \end{pmatrix}, \quad (33.38)$$

is diagonalized by the following linear combinations

$$\tilde{\chi}_i^- = U_{i1}\tilde{W}^- + U_{i2}\tilde{H}_1^-, \quad (33.39)$$

$$\tilde{\chi}_i^+ = V_{i1}\tilde{W}^+ + V_{i2}\tilde{H}_1^+. \quad (33.40)$$

We choose $\det(U) = 1$ and $U^*\mathcal{M}_{\tilde{\chi}^\pm}V^\dagger = \text{diag}(m_{\tilde{\chi}_1^\pm}, m_{\tilde{\chi}_2^\pm})$ with non-negative chargino masses $m_{\tilde{\chi}_i^\pm} \geq 0$.

When discussing the squark mass matrix including mixing, it is convenient to choose a basis where the squarks are rotated in the same way as the corresponding quarks in the standard model. We follow the conventions of the particle data group [216] and put the mixing in the left-handed d -quark fields, so that the definition of the Cabibbo-Kobayashi-Maskawa matrix is $\mathbf{K} = \mathbf{V}_1\mathbf{V}_2^\dagger$, where \mathbf{V}_1 (\mathbf{V}_2) rotates the interaction left-handed u -quark (d -quark) fields to mass eigenstates. For sleptons we choose an analogous basis, but due to the masslessness of neutrinos no analog of the CKM matrix appears.

We then obtain the general 6×6 \tilde{u} - and \tilde{d} -squark mass matrices:

$$\mathcal{M}_{\tilde{u}}^2 = \begin{pmatrix} \mathbf{M}_Q^2 + \mathbf{m}_u^\dagger \mathbf{m}_u + D_{LL}^u \mathbf{1} & \mathbf{m}_u^\dagger (\mathbf{A}_U^\dagger - \mu^* \cot \beta) \\ (\mathbf{A}_U - \mu \cot \beta) \mathbf{m}_u & \mathbf{M}_U^2 + \mathbf{m}_u \mathbf{m}_u^\dagger + D_{RR}^u \mathbf{1} \end{pmatrix}, \quad (33.41)$$

$$\mathcal{M}_{\tilde{d}}^2 = \begin{pmatrix} \mathbf{K}^\dagger \mathbf{M}_Q^2 \mathbf{K} + \mathbf{m}_d \mathbf{m}_d^\dagger + D_{LL}^d \mathbf{1} & \mathbf{m}_d^\dagger (\mathbf{A}_D^\dagger - \mu^* \tan \beta) \\ (\mathbf{A}_D - \mu \tan \beta) \mathbf{m}_d & \mathbf{M}_D^2 + \mathbf{m}_d \mathbf{m}_d^\dagger + D_{RR}^d \mathbf{1} \end{pmatrix}, \quad (33.42)$$

and the general sneutrino and charged slepton mass matrices

$$\mathcal{M}_{\tilde{\nu}}^2 = \mathbf{M}_L^2 + D_{LL}^\nu \mathbf{1} \quad (33.43)$$

$$\mathcal{M}_{\tilde{e}}^2 = \begin{pmatrix} \mathbf{M}_L^2 + \mathbf{m}_e \mathbf{m}_e^\dagger + D_{LL}^e \mathbf{1} & \mathbf{m}_e^\dagger (\mathbf{A}_E^\dagger - \mu^* \tan \beta) \\ (\mathbf{A}_E - \mu \tan \beta) \mathbf{m}_e & \mathbf{M}_E^2 + \mathbf{m}_e \mathbf{m}_e^\dagger + D_{RR}^e \mathbf{1} \end{pmatrix}. \quad (33.44)$$

Here

$$D_{LL}^f = m_Z^2 \cos 2\beta (T_{3f} - e_f \sin^2 \theta_w), \quad (33.45)$$

$$D_{RR}^f = m_Z^2 \cos 2\beta e_f \sin^2 \theta_w. \quad (33.46)$$

In the chosen basis, $\mathbf{m}_u = \text{diag}(m_u, m_c, m_t)$, $\mathbf{m}_d = \text{diag}(m_d, m_s, m_b)$ and $\mathbf{m}_e = \text{diag}(m_e, m_\mu, m_\tau)$.

The slepton and squark mass eigenstates \tilde{f}_k ($\tilde{\nu}_k$ with $k = 1, 2, 3$ and \tilde{e}_k, \tilde{u}_k and \tilde{d}_k with $k = 1, \dots, 6$) diagonalize the previous mass matrices and are related to the current sfermion eigenstates $\tilde{\mathbf{f}}_L$ and $\tilde{\mathbf{f}}_R$ via ($a = 1, 2, 3$)

$$\tilde{f}_{La} = \sum_{k=1}^6 \tilde{f}_k \Gamma_{FL}^{*ka}, \quad (33.47)$$

$$\tilde{f}_{Ra} = \sum_{k=1}^6 \tilde{f}_k \Gamma_{FR}^{*ka}. \quad (33.48)$$

Table 33.3: Particle codes (synonyms are separated by commas).

ν_e	knue,knu(1)	γ	kgamma	$\tilde{\chi}_i^0$	kn(i) $i = 1 \dots 4$	\tilde{u}_1	ksu(1),ksqu(1)
e	ke,kl(1)	W^\pm	kw	$\tilde{\chi}_k^\pm$	kcha(k) $k = 1, 2$	\tilde{u}_2	ksu(2),ksqu(4)
ν_μ	knumu,knu(2)	Z^0	kz	\tilde{g}	kgluin	\tilde{d}_1	ksd(1),ksqd(1)
μ	kmu,kl(2)	g	kgluon	$\tilde{\nu}_e$	ksnue,ksnu(1)	\tilde{d}_2	ksd(2),ksqd(4)
ν_τ	kntau,knu(3)			\tilde{e}_1	kse(1),ksl(1)	\tilde{c}_1	ksc(1),ksqu(2)
τ	ktau,kl(3)			\tilde{e}_2	kse(2),ksl(4)	\tilde{c}_2	ksc(2),ksqu(5)
u	ku,kqu(1)	H^0	kh1	$\tilde{\nu}_\mu$	ksnumu,ksnu(2)	\tilde{s}_1	kss(1),ksqd(2)
d	kd,kqd(1)	h^0	kh2	$\tilde{\mu}_1$	ksmu(1),ksl(2)	\tilde{s}_2	kss(2),ksqd(5)
c	kc,kqu(2)	A^0	kh3	$\tilde{\mu}_2$	ksmu(2),ksl(5)	\tilde{b}_1	ksb(1),ksqd(3)
s	ks,kqd(2)	H^\pm	khc	$\tilde{\nu}_\tau$	ksnuta,ksnu(3)	\tilde{b}_2	ksb(2),ksqd(6)
b	kb,kqd(3)	G^0	kgold0	$\tilde{\tau}_1$	kstau(1),ksl(3)	\tilde{t}_1	kst(1),ksqu(3)
t	kt,kqu(3)	G^\pm	kgoldc	$\tilde{\tau}_2$	kstau(2),ksl(6)	\tilde{t}_2	kst(2),ksqu(6)

The squark and charged slepton mixing matrices $\mathbf{\Gamma}_{UL,R}$, $\mathbf{\Gamma}_{DL,R}$ and $\mathbf{\Gamma}_{EL,R}$ have dimension 6×3 , while the sneutrino mixing matrix $\mathbf{\Gamma}_{\nu L}$ has dimension 3×3 .

This version of DarkSUSY allows only for diagonal matrices \mathbf{A}_U , \mathbf{A}_D , \mathbf{A}_E , \mathbf{M}_Q , \mathbf{M}_U , \mathbf{M}_D , \mathbf{M}_E , and \mathbf{M}_L . This ansatz, while not being the most general one, implies the absence of tree-level flavor changing neutral currents in all sectors of the model. In this case, the squark mass matrices can be diagonalized analytically. For example, for the top squark one has, in terms of the top squark mixing angle $\theta_{\tilde{t}}$,

$$\Gamma_{UL}^{\tilde{t}_1\tilde{t}} = \Gamma_{UR}^{\tilde{t}_2\tilde{t}} = \cos\theta_{\tilde{t}}, \quad \Gamma_{UL}^{\tilde{t}_2\tilde{t}} = -\Gamma_{UR}^{\tilde{t}_1\tilde{t}} = \sin\theta_{\tilde{t}}. \quad (33.49)$$

Special values of the sfermion masses can be set with the parameters `msquarks`, and `msleptons`. If `msquarks=msleptons=0`, the sfermion masses are obtained with the diagonalization described above. If `msquarks>0` (or `msleptons>0`), all squark masses are set to `msquarks` (or all slepton masses to `msleptons`). Finally, if `msquarks<0` (or `msleptons<0`), the squark (or slepton) masses are set equal to the neutralino mass but never less than $|\text{msquarks}|$ (or $|\text{msleptons}|$). This is to provide the lightest possible sfermions compatible with a neutralino LSP. In all of these cases, there is no mixing between sfermions.

The particle masses are available in an array `mass(p)`, where p is the particle code from table ???. Similarly, particle decay width are available as `width(p)`, but currently only the width of the Higgs bosons are calculated, the other particles having fictitious widths of 1 or 5 GeV (for the sole purpose of regularizing annihilation amplitudes close to poles).

33.8.1.3 Three-particle vertices

We define three-particle vertices $\text{gl}(i,j,k) = g_{ij k}^L$ and $\text{gr}(i,j,k) = g_{ij k}^R$ as follows. We adopt the convention that the order of the particles in the indices is the order in which they appear in the corresponding lagrangian term, so the last particle is always entering. If there are charged particles in the vertex, they are both assumed positively charged, and the particle that exits the vertex is indexed before the particle that enters.

- Three scalar bosons:

$$\mathcal{L}_{\text{int}} = g_{\phi_i\phi_j\phi_k} m_W \phi_i \phi_j \phi_k \quad (33.50)$$

where ϕ_i is a Higgs or a Goldstone boson. In this case, $\text{gl}=\text{gr}=g$. Available vertices are $\phi_i\phi_j\phi_k = H_i^0 H_j^0 H_k^0$, $H_i^0 H^- H^+$, $H_i^0 A^0 A^0$, $H_i^0 G^0 G^0$, $H_i^0 G^- G^+$, $H_i^0 G^- H^+$, $H_i^0 G^- G^+$, $A^0 G^- H^+$, $A^0 G^0 H_i^0$, and permutations.

- Two scalar and one vector bosons:

$$\mathcal{L}_{\text{int}} = g_{V\phi_1\phi_2} V^\mu \phi_1 i \overleftrightarrow{\partial}_\mu \phi_2. \quad (33.51)$$

Available vertices are $V\phi_1\phi_2 = Z^0 H_i^0 A^0, Z^0 H^- H^+, \gamma H^- H^+, W^- H^+ A^0, W^- H^+ H_i^0$, and permutations.

- One scalar and two vector bosons:

$$\mathcal{L}_{\text{int}} = g_{\phi V_1 V_2} m_W g_{\mu\nu} \phi V_1^\mu V_2^\nu \quad (33.52)$$

Available vertices are $\phi V_1 V_2 = H_i^0 W^- W^+, H_i^0 Z^0 Z^0$.

- Three vector bosons:

$$i g_{V_1 V_2 V_3} [(k_1 - k_3)_\nu g_{\mu\lambda} + (k_3 - k_2)_\mu g_{\lambda\nu} + (k_2 - k_1) g_{\mu\nu}] \quad (33.53)$$

with all momenta incoming and assigned as $V_1^\mu(k_1), V_2^\nu(k_2)$ and $V_3^\lambda(k_3)$. Available vertices are $Z^0 W^- W^+$ and $\gamma W^- W^+$.

- One scalar boson and two Dirac fermions:

$$\mathcal{L}_{\text{int}} = \phi \bar{\psi}_1 (g_{\phi\psi_1\psi_2}^L P_L + g_{\phi\psi_1\psi_2}^R P_R) \psi_2 \quad (33.54)$$

Available vertices are $\phi\psi_1\psi_2 =$

- One vector boson and two Dirac fermions:

$$\mathcal{L}_{\text{int}} = V_\mu \bar{\psi}_1 \gamma^\mu (g_{V\psi_1\psi_2}^L P_L + g_{V\psi_1\psi_2}^R P_R) \psi_2 \quad (33.55)$$

Available vertices are $V\psi_1\psi_2 =$

- One scalar boson, one Dirac and one Majorana fermion:

$$\mathcal{L}_{\text{int}} = \phi \bar{\psi} (g_{\phi\psi\chi}^L P_L + g_{\phi\psi\chi}^R P_R) \chi \quad (33.56)$$

Available vertices are $\phi\psi\chi =$

- One vector boson, one Dirac and one Majorana fermion:

$$\mathcal{L}_{\text{int}} = V_\mu \bar{\psi} \gamma^\mu (g_{V\psi\chi}^L P_L + g_{V\psi\chi}^R P_R) \chi \quad (33.57)$$

Available vertices are $V\psi\chi =$

- One scalar boson and two Majorana fermions:

$$\mathcal{L}_{\text{int}} = \quad (33.58)$$

Available vertices are...

- One vector boson and two Majorana fermions:

$$\mathcal{L}_{\text{int}} = \quad (33.59)$$

Explicit expressions for the coupling constants g_{ijk} can be obtained in [159, 160, 207], with radiative corrections to trilinear scalar couplings in [217]. We have rederived from the superpotential all vertices we have implemented.

Implemented vertices: those listed above plus $Z^0 W^\pm W^\mp, Z^0 H_i^0 H_i^0, W^\pm H^\mp A^0, W^\pm H^\mp H_i^0, H_i^0 W^\pm W^\mp, H_i^0 Z^0 Z^0, Z^0 A^0 H, H_i^0 A^0 A^0, A^0 f f, H_i^0 f f, Z^0 f f, Z^0 \tilde{\chi}^0 \tilde{\chi}^0, H_i^0 \tilde{\chi}^0 \tilde{\chi}^0, Z^0 \tilde{\chi}^\pm \tilde{\chi}^\pm, W^\mp \tilde{\chi}^0 \tilde{\chi}^\pm, H^\mp \tilde{\chi}^0 \tilde{\chi}^\pm, \tilde{q} \tilde{q} q, \tilde{f} \tilde{\chi}^0 f, H_i^0 \tilde{\chi}^\pm \tilde{\chi}^\mp, A^0 \tilde{\chi}^\pm \tilde{\chi}^\mp, W^\pm f f', H^\pm f f', \gamma W^\pm W^\mp, \gamma H^\pm H^\mp, Z^0 \tilde{\chi}^\pm \tilde{\chi}^\mp, \gamma \tilde{\chi}^\pm \tilde{\chi}^\mp, \gamma f f, G H H, G G H, G^\mp \tilde{\chi}^0 \tilde{\chi}^\pm$.

33.8.2 General supersymmetry – routines

Input parameters, options, results, etc. are contained in common blocks in the file `dsmssm.h`, which the user has to include. The input parameters are ($a = 1, 2, 3$)

$ma = m_A$, $\tan\beta = \tan\beta$, $\mu = \mu$, $m1 = M_1$,
 $m2 = M_2$, $m3 = M_3$, $\text{asofte}(a) = A_{Eaa}$, $\text{asoftu}(a) = A_{Uaa}$,
 $\text{asoftd}(a) = A_{Daa}$, $\text{mass2q}(a) = M_{Qaa}^2$, $\text{mass2l}(a) = M_{Laa}^2$, $\text{mass2u}(a) = M_{Uaa}^2$,
 $\text{mass2d}(a) = M_{Daa}^2$, $\text{mass2e}(a) = M_{Eaa}^2$.

The options are (see previous subsections for a description)

`higloop` choice of tree-level or radiatively corrected Higgs boson masses;

`neuloop` choice of tree-level or radiatively corrected neutralino masses;

`msquarks,msleptons` choice of squark and slepton masses.

To initialize DarkSUSY for a new model, you should call

subroutine **dsmodelsetup**(unphys,warning)

Purpose: To calculate the particle spectrum, widths and couplings.

Output:

`unphys` *i* non-zero if the model is unphysical
`warning` *i* non-zero if (typically the Higgs) code has issued a warning.

which calculates couplings, masses and some basic cross sections.

The following subroutines specify the values of the model parameters, and read/write them to a file. The user should create his own versions by editing a copy of them. Please call them with a different name.

subroutine **dsgive_model**(mu,m2,ma,tanbe,msq,atm,abm)

Purpose: Set the MSSM parameters as specified by the arguments.

Inputs:

`mu` *r8* The μ parameter in GeV.
`m2` *r8* The M_2 parameter in GeV.
`ma` *r8* The mass of the CP-odd Higgs boson, m_A in GeV.
`tanbe` *r8* $\tan\beta$.
`msq` *r8* Sets \mathbf{M}^Q , etc. to a common mass scale m_0 in GeV.
`atm` *r8* Sets A_t in units of m_0 (range: -3 — 3).
`abm` *r8* Sets A_b in units of m_0 (range: -3 — 3).

There are also routines to setup more general MSSM models, the currently most general being `dsgive_model25`.

The following subroutines are useful in the analysis.

subroutine **wspctm**(unit)

Purpose: Write the particle mass spectrum and mixing matrices to unit `unit`.

Inputs:

`unit` *i* Unit number to write to.

subroutine **wvertx**(unit)

Purpose: Write all non-vanishing three-particle vertices to unit `unit`.

Inputs:

`unit` *i* Unit number to write to.

subroutine **wunph**(unit)

Purpose: Write the reason for which the model is not physically acceptable (tachyons, etc.) to unit `unit`.

Inputs:

`unit` *i* Unit number to write to.

subroutine `wexcl`(unit)*Purpose:* Write the reason(s) for which the model is experimentally excluded to unit `unit`.*Inputs:*`unit` `i` Unit number to write to.**subroutine `dswhwar`(unit)***Purpose:* Write the reason(s) for which the Higgs calculation issued warnings to unit `unit`.*Inputs:*`unit` `i` Unit number to write to.

33.9 `mssm/ge_cmssm`: cMSSM interface (Isasugra)

33.9.1 mSUGRA (ISASUGRA) interface to DarkSUSY

If `Isasugra` is available, DarkSUSY can use `Isasugra` to generate mSUGRA models. In `src_modles/mssm/ge_cmssm`, routines are available to transfer the mSUGRA parameters from DarkSUSY to `Isasugra`, call `Isasugra` and then transfer back the results to DarkSUSY. The philosophy of this interface is that whenever a user uses `Isasugra`, we should use all the results of `Isasugra` also in DarkSUSY. That means that instead of calculating the mass spectrum from the low-energy parameters obtained from `Isasugra`, we extract the masses and mixings from `Isasugra`.

33.10 `mssm/ge_slha`: SUSY Les Houches Accord interface

33.10.1 SUSY Les Houches Accord

DarkSUSY includes routines to read and write SUSY Les Houches Accord [163, 164] files (SLHA files). This is done with the help of SLHALIB by T. Hahn [218].

DarkSUSY will write SLHA2 files and expect to get SLHA2 as well. The implementation right now takes a middle path between dumping everything or just a minimal set of inputs to the SLHA2 file. This choice was made to make most other SLHA2-aware programs able to exchange SLHA2 files with DarkSUSY.

However, we have not made a careful testing with lots of other codes, so if you try this out, please let us know if there are some things that don't work or could work better.

33.11 `mssm/ini`: Initialization routines

33.11.1 Initialization routines

Before DarkSUSY is used for some calculations, it needs to be initialized. This is done with a call to `dsinit`, which in turn calls the particle-specific initialization routine `dsinit_module` that resides in each module. This means that the call to `dsinit` should be the first call in any program using DarkSUSY. Any calls the user makes to other routines, either to calculate things or select a different model (e.g. a different halo model) should come after the call to `dsinit`.

In the `mssm` module, the routine `dsinit_module` that all standard parameters are defined, such as standard model parameters and particle codes. It also sets more particle-specific defaults for DarkSUSY routines where this is available, via calls to routines such as `dsddset_mssm` and `dsanset_mssm`.

33.12 mssm/rd: Relic density

33.12.1 Relic density of neutralinos

In the folder `mssm/rd`, we provide one of the two interface functions that are necessary for the `core` library to solve the Boltzmann equation for any cold dark matter particle and hence calculate its relic density:[‡] `dsrdparticles` determines which particles can coannihilate (based on their mass differences) and puts these particles into a common block for the annihilation rate routines (`dsanwx`); it also checks where we have resonances and thresholds and adds these to an array, which is passed to the relic density routines. The relic density routines then use this knowledge to make sure the tabulation of the cross section and the integrations are performed correctly at these difficult points.

For convenience, we include one further routine in this folder: `dsrdwrate` writes the invariant rate to a specified unit. This is mostly useful for debugging purposes.

33.12.2 Internal degrees of freedom

In Section 22.1.2, we have reviewed the standard way of calculating the relic density. Here, we add some comments which are specific to the implementation of the effective invariant rate in the `mssm` module

If we look at Eqs. (22.34) and (22.39) we see that we have a freedom on how to treat particles degenerate in mass, e.g. a chargino can be treated either

- a) as two separate species χ_i^+ and χ_i^- , each with internal degrees of freedom $g_{\chi^+} = g_{\chi^-} = 2$, or,
- b) as a single species χ_i^\pm with $g_{\chi_i^\pm} = 4$ internal degrees of freedom.

Of course the two views are equivalent, we just have to be careful including the g_i 's consistently whichever view we take. In a), we have the advantage that all the W_{ij} that enter into Eq. (22.34) enter as they are, i.e. without any correction factors for the degrees of freedom. On the other hand we get many terms in the sum that are identical and we need some book-keeping machinery to avoid calculating identical terms more than once. On the other hand, with option b), the sum over W_{ij} in Eq. (22.34) is much simpler only containing terms that are not identical (except for the trivial identity $W_{ij} = W_{ji}$ which is easily taken care of). However, the individual W_{ij} will be some linear combinations of the more basic W_{ij} entering in option a), where the coefficients have to be calculated for each specific type of initial condition.

Below we will perform this calculation to show how the W_{ij} look like in option b) for different initial states. We will use a prime on the W_{ij} when they refer to these combined states to indicate the difference.

33.12.2.1 Neutralino-chargino annihilation

The starting point is Eq. (22.34) which we will use to define the W_{ij} in option b) such that W_{eff} is the same as in option a). Eq. (22.39) is then guaranteed to be the same in both cases since the sum in the denominator is linear in g_i .

Now consider annihilation between χ_i^0 and χ_c^+ or χ_c^- . The corresponding terms in Eq. (22.34)

[‡]The other, `dsanwx`, returns the effective invariant rate and resides in `mssm/an/`.

does for option a) read

$$\begin{aligned}
W_{\text{eff}} &= \sum_{ij} \frac{p_{ij}}{p_{11}} \frac{g_i g_j}{g_1^2} W_{ij} = \frac{p_{ic}}{p_{11}} \frac{2 \cdot 2}{2^2} \left[W_{\chi_c^0 \chi_c^+} + W_{\chi_c^0 \chi_c^-} + \underbrace{W_{\chi_c^+ \chi_c^0}}_{W_{\chi_c^0 \chi_c^+}} + \underbrace{W_{\chi_c^- \chi_c^0}}_{W_{\chi_c^0 \chi_c^-}} \right] \\
&= 2 \frac{p_{ic}}{p_{11}} \left[W_{\chi_c^0 \chi_c^+} + \underbrace{W_{\chi_c^0 \chi_c^-}}_{W_{\chi_c^0 \chi_c^+}} \right] = 4 \frac{p_{ic}}{p_{11}} W_{\chi_c^0 \chi_c^+}
\end{aligned} \tag{33.60}$$

For option b), we instead get

$$W_{\text{eff}} = \sum_{ij} \frac{p_{ij}}{p_{11}} \frac{g_i g_j}{g_1^2} W_{ij} = \frac{p_{ic}}{p_{11}} \frac{2 \cdot 4}{2^2} \left[W'_{\chi_c^0 \chi_c^\pm} + \underbrace{W'_{\chi_c^\pm \chi_c^0}}_{W'_{\chi_c^0 \chi_c^\pm}} \right] = 4 \frac{p_{ic}}{p_{11}} W'_{\chi_c^0 \chi_c^\pm} \tag{33.61}$$

Comparing Eq. (33.61) and Eq. (33.60) we see that they are identical if we make the identification

$$W'_{\chi_c^0 \chi_c^\pm} \equiv W_{\chi_c^0 \chi_c^+} \tag{33.62}$$

33.12.2.2 Chargino-chargino annihilation

First consider the case where we include the terms in the sum for which we have annihilation between χ_c^+ or χ_c^- and χ_d^+ or χ_d^- with $c \neq d$.

In option a), the corresponding terms in Eq. (22.34) reads

$$\begin{aligned}
W_{\text{eff}} &= \sum_{ij} \frac{p_{ij}}{p_{11}} \frac{g_i g_j}{g_1^2} W_{ij} \\
&= \frac{p_{cd}}{p_{11}} \frac{2 \cdot 2}{2^2} \left[W_{\chi_c^+ \chi_d^+} + W_{\chi_c^+ \chi_d^-} + W_{\chi_c^- \chi_d^+} + W_{\chi_c^- \chi_d^-} \right. \\
&\quad \left. + \underbrace{W_{\chi_d^+ \chi_c^+}}_{W_{\chi_c^+ \chi_d^+}} + \underbrace{W_{\chi_d^+ \chi_c^-}}_{W_{\chi_c^- \chi_d^+}} + \underbrace{W_{\chi_d^- \chi_c^+}}_{W_{\chi_c^+ \chi_d^-}} + \underbrace{W_{\chi_d^- \chi_c^-}}_{W_{\chi_c^- \chi_d^-}} \right] \\
&= 2 \frac{p_{cd}}{p_{11}} \left[W_{\chi_c^+ \chi_d^+} + W_{\chi_c^+ \chi_d^-} + \underbrace{W_{\chi_c^- \chi_d^+}}_{W_{\chi_c^+ \chi_d^-}} + \underbrace{W_{\chi_c^- \chi_d^-}}_{W_{\chi_c^+ \chi_d^+}} \right] \\
&= 4 \frac{p_{cd}}{p_{11}} \left[W_{\chi_c^+ \chi_d^+} + W_{\chi_c^+ \chi_d^-} \right]
\end{aligned} \tag{33.63}$$

In option b), the corresponding terms would instead read

$$W_{\text{eff}} = \sum_{ij} \frac{p_{ij}}{p_{11}} \frac{g_i g_j}{g_1^2} W'_{ij} = \frac{p_{cd}}{p_{11}} \frac{4 \cdot 4}{2^2} \left[W'_{\chi_c^\pm \chi_d^\pm} + \underbrace{W'_{\chi_d^\pm \chi_c^\pm}}_{W'_{\chi_c^\pm \chi_d^\pm}} \right] = 8 \frac{p_{cd}}{p_{11}} W'_{\chi_c^\pm \chi_d^\pm} \tag{33.64}$$

Comparing Eq. (33.63) and Eq. (33.64) we see that they are identical if we make the following identification

$$W'_{\chi_c^\pm \chi_d^\pm} \equiv \frac{1}{2} \left[W_{\chi_c^+ \chi_d^+} + W_{\chi_c^+ \chi_d^-} \right] \tag{33.65}$$

For clarity, let's also consider the case where $c = d$. In option a), the terms in W_{eff} are

$$\begin{aligned} W_{\text{eff}} &= \sum_{ij} \frac{p_{ij}}{p_{11}} \frac{g_i g_j}{g_1^2} W_{ij} = \frac{p_{cc}}{p_{11}} \frac{2 \cdot 2}{2^2} \left[W_{\chi_c^+ \chi_c^+} + W_{\chi_c^+ \chi_c^-} + \underbrace{W_{\chi_c^- \chi_c^+}}_{W_{\chi_c^+ \chi_c^-}} + \underbrace{W_{\chi_c^- \chi_c^-}}_{W_{\chi_c^+ \chi_c^+}} \right] \\ &= 2 \frac{p_{cc}}{p_{11}} \left[W_{\chi_c^+ \chi_c^+} + W_{\chi_c^+ \chi_c^-} \right] \end{aligned} \quad (33.66)$$

In option b), the corresponding term would instead read

$$W_{\text{eff}} = \sum_{ij} \frac{p_{ij}}{p_{11}} \frac{g_i g_j}{g_1^2} W'_{ij} = \frac{p_{cc}}{p_{11}} \frac{4 \cdot 4}{2^2} W'_{\chi_c^\pm \chi_c^\pm} = 4 \frac{p_{cc}}{p_{11}} W'_{\chi_c^\pm \chi_c^\pm} \quad (33.67)$$

Comparing Eq. (33.66) and Eq. (33.67) we see that they are identical if we make the following identification

$$W'_{\chi_c^\pm \chi_c^\pm} \equiv \frac{1}{2} \left[W_{\chi_c^+ \chi_c^+} + W_{\chi_c^+ \chi_c^-} \right] \quad (33.68)$$

i.e. the same identification as in the case $c \neq d$.

33.12.2.3 Neutralino-sfermion annihilation

For each sfermion we have in total four different states, \tilde{f}_1 , \tilde{f}_2 , \tilde{f}_1^* and \tilde{f}_2^* . Of these, the \tilde{f}_1 and \tilde{f}_2 in general have different masses and have to be treated separately. Considering only one mass eigenstate \tilde{f}_k , option a) then means that we treat \tilde{f}_k and \tilde{f}_k^* as two separate species with $g_i = 1$ degree of freedom each, whereas option b) means that we treat them as one species \tilde{f}'_k with $g_i = 2$ degrees of freedom. As before, the prime indicates that we mean both the particle and the antiparticle state.

Note, that for squarks we also have the number of colours $N_c = 3$ to take into account. In option a) we should choose to treat even colour state differently, i.e. $g_i = 1$, whereas $g_i = 6$ in case b). The expressions would be the same as above except that both the expression in a) and b) would be multiplied by the colour factor $N_c = 3$. The expression relating case a) and case b) is thus unaffected by this colour factor. Note however, that in option b) we take the average over the squark colours (or in this case calculate it only for one colour. See sections 33.12.2.6 and 33.12.2.7 below for more details.

For option a), Eq. (22.34) then reads

$$\begin{aligned} W_{\text{eff}} &= \sum_{ij} \frac{p_{ij}}{p_{11}} \frac{g_i g_j}{g_1^2} W_{ij} = \frac{p_{ik}}{p_{11}} \frac{2 \cdot 1}{2^2} \left[W_{\chi_i^0 \tilde{f}_k} + W_{\chi_i^0 \tilde{f}_k^*} + \underbrace{W_{\tilde{f}_k \chi_i^0}}_{W_{\chi_i^0 \tilde{f}_k}} + \underbrace{W_{\tilde{f}_k^* \chi_i^0}}_{W_{\chi_i^0 \tilde{f}_k^*}} \right] \\ &= \frac{p_{ik}}{p_{11}} \left[W_{\chi_i^0 \tilde{f}_k} + \underbrace{W_{\chi_i^0 \tilde{f}_k^*}}_{W_{\chi_i^0 \tilde{f}_k}} \right] = 2 \frac{p_{ik}}{p_{11}} W_{\chi_i^0 \tilde{f}_k} \end{aligned} \quad (33.69)$$

whereas for option b), Eq. (22.34) reads

$$W_{\text{eff}} = \sum_{ij} \frac{p_{ij}}{p_{11}} \frac{g_i g_j}{g_1^2} W'_{ij} = \frac{p_{ik}}{p_{11}} \frac{2 \cdot 2}{2^2} \left[W'_{\chi_i^0 \tilde{f}'_k} + \underbrace{W'_{\tilde{f}'_k \chi_i^0}}_{W'_{\chi_i^0 \tilde{f}'_k}} \right] = 2 \frac{p_{ik}}{p_{11}} W'_{\chi_i^0 \tilde{f}'_k} \quad (33.70)$$

Comparing Eq. (33.70) and Eq. (33.69) we see that they are identical if we make the identification

$$W'_{\chi_i^0 \tilde{f}'_k} \equiv W_{\chi_i^0 \tilde{f}_k} \quad (33.71)$$

For clarity, for squarks the corresponding expression would be

$$W'_{\chi_i^0 \tilde{q}'_k} \equiv \frac{1}{3} \sum_{a=1}^3 W_{\chi_i^0 \tilde{q}_k^a} \quad (33.72)$$

where a is a colour index.

33.12.2.4 Chargino-sfermion annihilation

In option a) the chargino has $g_i = 2$ and the sfermion has $g_i = 1$ degrees of freedom, whereas in option b), the chargino has $g_i = 4$ and the sfermion has $g_i = 2$ degrees of freedom

For option a), Eq. (22.34) then reads

$$\begin{aligned} W_{\text{eff}} &= \sum_{ij} \frac{p_{ij}}{p_{11}} \frac{g_i g_j}{g_1^2} W_{ij} \\ &= \frac{p_{ck}}{p_{11}} \frac{2 \cdot 1}{2^2} \left[W_{\chi_c^+ \tilde{f}_k} + W_{\chi_c^+ \tilde{f}_k^*} + W_{\chi_c^- \tilde{f}_k} + W_{\chi_c^- \tilde{f}_k^*} \right. \\ &\quad \left. + \underbrace{W_{\tilde{f}_k \chi_c^+}}_{W_{\chi_c^+ \tilde{f}_k}} + \underbrace{W_{\tilde{f}_k^* \chi_c^+}}_{W_{\chi_c^+ \tilde{f}_k^*}} + \underbrace{W_{\tilde{f}_k \chi_c^-}}_{W_{\chi_c^- \tilde{f}_k}} + \underbrace{W_{\tilde{f}_k^* \chi_c^-}}_{W_{\chi_c^- \tilde{f}_k^*}} \right] \\ &= \frac{p_{ck}}{p_{11}} \left[W_{\chi_c^+ \tilde{f}_k} + W_{\chi_c^+ \tilde{f}_k^*} + \underbrace{W_{\chi_c^- \tilde{f}_k}}_{W_{\chi_c^+ \tilde{f}_k^*}} + \underbrace{W_{\chi_c^- \tilde{f}_k^*}}_{W_{\chi_c^+ \tilde{f}_k}} \right] = 2 \frac{p_{ck}}{p_{11}} \left[W_{\chi_c^+ \tilde{f}_k} + W_{\chi_c^+ \tilde{f}_k^*} \right] \end{aligned} \quad (33.73)$$

In option b), Eq. (22.34) reads

$$W_{\text{eff}} = \sum_{ij} \frac{p_{ij}}{p_{11}} \frac{g_i g_j}{g_1^2} W'_{ij} = \frac{p_{ck}}{p_{11}} \frac{4 \cdot 2}{2^2} \left[W'_{\chi_c^\pm \tilde{f}'_k} + \underbrace{W'_{\tilde{f}'_k \chi_c^\pm}}_{W'_{\chi_c^\pm \tilde{f}'_k}} \right] = 4 \frac{p_{ck}}{p_{11}} W'_{\chi_c^\pm \tilde{f}'_k} \quad (33.74)$$

Comparing Eq. (33.74) and Eq. (33.73) we see that they are identical if we make the identification

$$W'_{\chi_c^\pm \tilde{f}'_k} \equiv \frac{1}{2} \left[W_{\chi_c^+ \tilde{f}_k} + W_{\chi_c^+ \tilde{f}_k^*} \right] \quad (33.75)$$

For clarity, for squarks the corresponding expression would be

$$W'_{\chi_c^\pm \tilde{q}'_k} \equiv \frac{1}{2} \frac{1}{3} \sum_{a=1}^3 \left[W_{\chi_c^+ \tilde{q}_k^a} + W_{\chi_c^+ \tilde{q}_k^{a*}} \right] \quad (33.76)$$

where a is a colour index.

33.12.2.5 Sfermion-sfermion annihilation

First consider the case where we have annihilation between sfermions of different types, i.e. annihilation between \tilde{f}_k or \tilde{f}_k^* and \tilde{f}_l or \tilde{f}_l^* .

For option a), Eq. (22.34) then reads

$$\begin{aligned}
W_{\text{eff}} &= \sum_{ij} \frac{p_{ij}}{p_{11}} \frac{g_i g_j}{g_1^2} W_{ij} \\
&= \frac{p_{kl}}{p_{11}} \frac{1 \cdot 1}{2^2} \left[W_{\tilde{f}_k \tilde{f}_l} + W_{\tilde{f}_k \tilde{f}_l^*} + W_{\tilde{f}_k^* \tilde{f}_l} + W_{\tilde{f}_k^* \tilde{f}_l^*} \right. \\
&\quad \left. + \underbrace{W_{\tilde{f}_l \tilde{f}_k}}_{W_{\tilde{f}_k \tilde{f}_l}} + \underbrace{W_{\tilde{f}_l \tilde{f}_k^*}}_{W_{\tilde{f}_k^* \tilde{f}_l}} + \underbrace{W_{\tilde{f}_l^* \tilde{f}_k}}_{W_{\tilde{f}_k \tilde{f}_l^*}} + \underbrace{W_{\tilde{f}_l^* \tilde{f}_k^*}}_{W_{\tilde{f}_k^* \tilde{f}_l^*}} \right] \\
&= \frac{1}{2} \frac{p_{kl}}{p_{11}} \left[W_{\tilde{f}_k \tilde{f}_l} + W_{\tilde{f}_k \tilde{f}_l^*} + \underbrace{W_{\tilde{f}_k^* \tilde{f}_l}}_{W_{\tilde{f}_k \tilde{f}_l^*}} + \underbrace{W_{\tilde{f}_k^* \tilde{f}_l^*}}_{W_{\tilde{f}_k \tilde{f}_l}} \right] = \frac{p_{kl}}{p_{11}} \left[W_{\tilde{f}_k \tilde{f}_l} + W_{\tilde{f}_k \tilde{f}_l^*} \right] \quad (33.77)
\end{aligned}$$

In option b) we would get

$$\begin{aligned}
W_{\text{eff}} &= \sum_{ij} \frac{p_{ij}}{p_{11}} \frac{g_i g_j}{g_1^2} W'_{ij} \\
&= \frac{p_{kl}}{p_{11}} \frac{2 \cdot 2}{2^2} \left[W'_{\tilde{f}'_k \tilde{f}'_l} + \underbrace{W'_{\tilde{f}'_l \tilde{f}'_k}}_{W'_{\tilde{f}'_k \tilde{f}'_l}} \right] = 2 \frac{p_{kl}}{p_{11}} \left[W'_{\tilde{f}'_k \tilde{f}'_l} \right] \quad (33.78)
\end{aligned}$$

Comparing Eq. (33.78) and Eq. (33.77) we see that they are identical if we make the identification

$$W'_{\tilde{f}'_k \tilde{f}'_l} \equiv \frac{1}{2} \left[W_{\tilde{f}_k \tilde{f}_l} + W_{\tilde{f}_k \tilde{f}_l^*} \right] \quad (33.79)$$

It is easy to show that this relation holds true even if $k = l$.

33.12.2.6 Squark-squark annihilation

Even though we treated sfermion-sfermion annihilation in the previous subsection, squarks have colour which can complicate things, so let's for clarity consider squarks separately.

Let's denote the squarks \tilde{q}_k^a where a is now a colour index. In option a) we will let each colour be a separate species, which means that $g_i = 1$ in this case. In option b) we will instead have $g_i = 6$. In option a) we would have

$$\begin{aligned}
W_{\text{eff}} &= \sum_{ij} \frac{p_{ij}}{p_{11}} \frac{g_i g_j}{g_1^2} W_{ij} \\
&= \frac{p_{kl}}{p_{11}} \frac{1 \cdot 1}{2^2} \sum_{a,b=1}^3 \left[W_{\tilde{q}_k^a \tilde{q}_l^b} + W_{\tilde{q}_k^a \tilde{q}_l^{b*}} + W_{\tilde{q}_k^{a*} \tilde{q}_l^b} + W_{\tilde{q}_k^{a*} \tilde{q}_l^{b*}} \right. \\
&\quad \left. + \underbrace{W_{\tilde{q}_l^a \tilde{q}_k^b}}_{W_{\tilde{q}_k^a \tilde{q}_l^b}} + \underbrace{W_{\tilde{q}_l^a \tilde{q}_k^{b*}}}_{W_{\tilde{q}_k^{a*} \tilde{q}_l^b}} + \underbrace{W_{\tilde{q}_l^{a*} \tilde{q}_k^b}}_{W_{\tilde{q}_k^a \tilde{q}_l^{b*}}} + \underbrace{W_{\tilde{q}_l^{a*} \tilde{q}_k^{b*}}}_{W_{\tilde{q}_k^{a*} \tilde{q}_l^{b*}}} \right] \\
&= \frac{1}{2} \frac{p_{kl}}{p_{11}} \sum_{a,b=1}^3 \left[W_{\tilde{q}_k^a \tilde{q}_l^b} + W_{\tilde{q}_k^a \tilde{q}_l^{b*}} + \underbrace{W_{\tilde{q}_k^{a*} \tilde{q}_l^b}}_{W_{\tilde{q}_k^a \tilde{q}_l^{b*}}} + \underbrace{W_{\tilde{q}_k^{a*} \tilde{q}_l^{b*}}}_{W_{\tilde{q}_k^a \tilde{q}_l^b}} \right] = \frac{p_{kl}}{p_{11}} \sum_{a,b=1}^3 \left[W_{\tilde{q}_k^a \tilde{q}_l^b} + W_{\tilde{q}_k^a \tilde{q}_l^{b*}} \right] \quad (33.80)
\end{aligned}$$

In option b) we would get

$$\begin{aligned} W_{\text{eff}} &= \sum_{ij} \frac{p_{ij}}{p_{11}} \frac{g_i g_j}{g_1^2} W'_{ij} \\ &= \frac{p_{kl}}{p_{11}} \frac{6 \cdot 6}{2^2} \left[W'_{\tilde{q}'_k \tilde{q}'_l} + \underbrace{W'_{\tilde{q}'_l \tilde{q}'_k}}_{W'_{\tilde{q}'_k \tilde{q}'_l}} \right] = 18 \frac{p_{kl}}{p_{11}} \left[W'_{\tilde{q}'_k \tilde{q}'_l} \right] \end{aligned} \quad (33.81)$$

Comparing Eq. (33.81) and Eq. (33.80) we see that they are identical if we make the identification

$$W'_{\tilde{q}'_k \tilde{q}'_l} \equiv \frac{1}{2} \frac{1}{9} \sum_{a,b=1}^3 \left[W_{\tilde{q}_k^a \tilde{q}_l^b} + W_{\tilde{q}_k^a \tilde{q}_l^{b*}} \right] \quad (33.82)$$

i.e. we get the same relation as for other sfermions, the only difference being that we in option b) should also take the average over the colour states.

33.12.2.7 Sfermion-squark annihilation

For clarity, if we have annihilation between a non-coloured sfermion and a squark, we would in the same way as in the previous subsection get

$$W'_{\tilde{f}'_k \tilde{q}'_l} \equiv \frac{1}{2} \frac{1}{3} \sum_{b=1}^3 \left[W_{\tilde{f}_k \tilde{q}_l^b} + W_{\tilde{f}_k \tilde{q}_l^{b*}} \right] \quad (33.83)$$

33.12.2.8 Summary of degrees of freedom

We have found above the following relations between option b) and option a),

$$\left\{ \begin{array}{l} W'_{\chi_i^0 \chi_j^\pm} \equiv W_{\chi_i^0 \chi_j^\pm} = W_{\chi_i^0 \chi_j^\mp} \quad , \quad \forall i = 1, \dots, 4, \quad j = 1, 2 \\ W'_{\chi_i^\pm \chi_j^\pm} \equiv \frac{1}{2} \left[W_{\chi_i^+ \chi_j^+} + W_{\chi_i^+ \chi_j^-} \right] = \frac{1}{2} \left[W_{\chi_i^- \chi_j^-} + W_{\chi_i^- \chi_j^+} \right] \quad , \quad \forall i = 1, 2, \quad j = 1, 2 \\ W'_{\chi_i^0 \tilde{f}'_k} \equiv W_{\chi_i^0 \tilde{f}_k} \quad , \quad \forall i = 1, \dots, 4, \quad k = 1, 2 \\ W'_{\chi_c^\pm \tilde{f}'_k} \equiv \frac{1}{2} \left[W_{\chi_c^+ \tilde{f}_k} + W_{\chi_c^+ \tilde{f}_k^*} \right] \quad , \quad \forall c = 1, 2, \quad k = 1, 2 \\ W'_{\tilde{f}'_k \tilde{f}'_l} \equiv \frac{1}{2} \left[W_{\tilde{f}_k \tilde{f}_l} + W_{\tilde{f}_k \tilde{f}_l^*} \right] \quad , \quad \forall k = 1, 2, \quad l = 1, 2 \\ W'_{\tilde{q}'_k \tilde{q}'_l} \equiv \frac{1}{2} \frac{1}{9} \sum_{a,b=1}^3 \left[W_{\tilde{q}_k^a \tilde{q}_l^b} + W_{\tilde{q}_k^a \tilde{q}_l^{b*}} \right] \quad , \quad \forall k = 1, 2, \quad l = 1, 2 \end{array} \right. \quad (33.84)$$

We don't list all the possible cases with squarks explicitly, the principle being that we in option b) should take the *average* over the squark colour states (see the squark-squark entry in the list above).

We will choose option b) and the code (dsandwdcoscn, dsandwdcoscn, dsasdwdcossfsf and dsasdwdcossfchi) should thus return W' as defined above. Note again that squarks are assumed to have $g_i = 6$ degrees of freedom in this convention and the summing over colours should also be taken into account in the code.

33.13 Relic density – more details on routines

33.13.1 Global parameters

When the relic density has been calculated, the integer variable `copart` in `dsandwcom.h` is set to indicate which coannihilating particles that have been included in the calculation. In Table 33.4, the meaning of this variable is shown for the case of supersymmetric particles.

Bit set	copart		PAW variables		Particle
	Octal value	Decimal value	cop1 bit	cop2 bit	
0	1	1	0	–	$\tilde{\chi}_1^0$
1	2	2	1	–	$\tilde{\chi}_2^0$
2	4	4	2	–	$\tilde{\chi}_3^0$
3	10	8	3	–	$\tilde{\chi}_4^0$
4	20	16	4	–	$\tilde{\chi}_{1\pm}$
5	40	32	5	–	$\tilde{\chi}_2\pm$
6	100	64	6	–	\tilde{e}_1
7	200	128	7	–	$\tilde{\mu}_1$
8	400	256	8	–	$\tilde{\tau}_1$
9	1 000	512	9	–	\tilde{e}_2
10	2 000	1 024	10	–	$\tilde{\mu}_2$
11	4 000	2 048	11	–	$\tilde{\tau}_2$
12	10 000	4 096	12	–	$\tilde{\nu}_e$
13	20 000	8 192	13	–	$\tilde{\nu}_\mu$
14	40 000	16 384	14	–	$\tilde{\nu}_\tau$
15	100 000	32 768	–	0	\tilde{u}_1
16	200 000	65 536	–	1	\tilde{c}_1
17	400 000	131 072	–	2	\tilde{t}_1
18	1 000 000	262 144	–	3	\tilde{u}_2
19	2 000 000	524 288	–	4	\tilde{c}_2
20	4 000 000	1 048 576	–	5	\tilde{t}_2
21	10 000 000	2 097 152	–	6	\tilde{d}_1
22	20 000 000	4 197 304	–	7	\tilde{s}_1
23	40 000 000	8 388 608	–	8	\tilde{b}_1
24	100 000 000	16 777 216	–	9	\tilde{d}_2
25	200 000 000	33 554 432	–	10	\tilde{s}_2
26	400 000 000	67 108 864	–	11	\tilde{b}_2

Table 33.4: The bits of `copart` are set to indicate which initial states that are included in the coannihilation calculation. In the output file `*.omegaco`, the value of `copart` is written in octal format. In PAW `cop1` and `cop2` are available. Check if a bit is set with `btest(cop1,bit)`.

Chapter 34

Silveira-Zee (Scalar Singlet)

The module `silveira_zee` implements the scalar singlet DM model of Silveira and Zee [4], also referred to as ‘scalar phantoms’ (which is the original name in [4]) and ‘singlet Higgs DM’. This model adds a gauge-singlet real scalar field S to the standard model, with imposed Z_2 symmetry $S \rightarrow -S$. Its Lagrangian is

$$\mathcal{L}_{SZ} = \mathcal{L}_{SM} + \frac{1}{2}\partial_\mu S \partial^\mu S - \frac{1}{2}\mu^2 S^2 - \frac{1}{2}\lambda S^2 H^\dagger H, \quad (34.1)$$

where H is the Standard Model Higgs doublet. After electroweak symmetry breaking, the S boson acquires a tree-level mass

$$m_S = \sqrt{\mu^2 + \frac{1}{2}\lambda v_0^2}, \quad (34.2)$$

where $v_0 = (\sqrt{2}G_F)^{-1/2} = 246.2$ GeV is the Higgs vacuum expectation value. The module uses the S mass m_S and the S -Higgs coupling constant λ as model parameters. These parameters are set with a call to `dsgivemodel_silveira_zee`, followed as usual by a call to `dsmodelsetup` to initialize the model for use with DarkSUSY.

34.1 `silveira_zee/fi`: Freeze-in routines

A call to `dsfiset_silveira_zee` prior to `dsfi2to2rhs` allows to perform freeze-in calculations with modified default settings. For example, call `dsfiset_silveira_zee('quantum_statistics','off')` will lead to a freeze-in calculation without taking into account the effect of quantum statistics for the heat bath particles, and call `dsfiset_silveira_zee('finiteT','off')` will switch off other finite-temperature effects (thermal masses and further temperature-dependence due to phase transitions). Default settings are restored with corresponding calls to the same routine, replacing ‘off’ with ‘default’.

Chapter 35

Self-Interacting Dark Matter

The vdSIDM module implements a simplified model setup where DM is confined to a dark sector without (direct) interactions with the standard model. Besides the DM particle, which is assumed to be non-relativistic, there is also a dark radiation (DR) particle which in general will contribute to the relativistic degrees of freedom in the universe – a quantity which is conventionally stated in terms of the contribution from a single neutrino species, ΔN_{eff} , and which is tightly constrained by both BBN and CMB. Finally, the module implements a mediator particle. In principle, the code structure is set up such that arbitrary spin combinations of these three particles can be implemented, but presently only scalar and vector mediators (for Dirac DM and DR) are available. If the mass of the mediator particle is much lighter than the DM mass, the resulting Yukawa potential between DM particles results in a self-scattering rate that is strongly velocity-dependent – as indicated in the name of the module.

35.1 vdSIDM/an: Annihilation routines

For the spin combinations that are currently implemented in the vdSIDM module, DM annihilates via the t - and u -channel to a pair of mediators. In the limit of small CMS energies $\sqrt{s} \rightarrow 2m_\chi$, this gives the following contributions to the invariant rate:

$$W_{\chi\chi \rightarrow VV} \xrightarrow{p_{\text{CM}} \ll m_\chi} 4\pi\alpha_\chi^2 \left(1 - \frac{m_V^2}{m_\chi^2}\right)^{3/2} \left(1 - \frac{m_V^2}{2m_\chi^2}\right)^{-2} \quad (35.1)$$

$$W_{\chi\chi \rightarrow \phi\phi} \xrightarrow{p_{\text{CM}} \ll m_\chi} 6\pi\alpha_\chi^2 p_{\text{CM}}^2 \left(1 - \frac{m_\phi^2}{m_\chi^2}\right)^{1/2} \left(1 - \frac{m_\phi^2}{2m_\chi^2}\right)^{-4} \left(1 - \frac{8m_\phi^2}{9m_\chi^2} + \frac{2m_\phi^4}{9m_\chi^4}\right), \quad (35.2)$$

where $\alpha_\chi \equiv g_\chi/(4\pi)$ and p_{CM} is the initial CMS momentum of (each of) the DM particles. Furthermore, DM can annihilate to a pair of DR particles, by a mediator exchange in the s -channel. This gives

$$W_{\chi\chi \rightarrow V^* \rightarrow \tilde{\gamma}\tilde{\gamma}} = 8\pi\alpha_\chi s^2 \left(1 + \frac{2m_\chi^2}{s}\right) \frac{\Gamma_{V \rightarrow \tilde{\gamma}\tilde{\gamma}}}{m_V} |D_V|^2 \quad (35.3)$$

$$W_{\chi\chi \rightarrow \phi^* \rightarrow \tilde{\gamma}\tilde{\gamma}} = 8\pi\alpha_\chi s^2 \left(1 - \frac{4m_\chi^2}{s}\right) \frac{\Gamma_{\phi \rightarrow \tilde{\gamma}\tilde{\gamma}}}{m_\phi} |D_\phi|^2, \quad (35.4)$$

where $D_{V,\phi}$ is defined as

$$|D_H(s)|^2 = \frac{1}{(s - m_H^2)^2 + m_H^2 \Gamma_H^2}, \quad (35.5)$$

i.e. as the inverse of the denominator of the respective propagator.

The interface function `dsanwx` adds the full tree-level expressions for these processes, and then multiplies the leading-order expressions in the $v \rightarrow 0$ limit by an enhancement factor $S(v)$ due to the Sommerfeld effect. The factor $S(v)$ itself is provided by the auxiliary function `dsansommerfeld` residing in `src_models/common/aux`, which implements the analytic expressions from Ref. [219, 150] (resulting from approximating the Yukawa potential by a Hulthén potential). The Sommerfeld factor depends on whether the process is s -wave dominated (as in the vector mediator case) or p -wave dominated (as in the scalar mediator case), which hence is one of the input parameters for `dsansommerfeld`.

35.2 vdSIDM/cr: Cosmic rays

The cosmic-ray source functions in vdSIDM simply return zero, as there are currently no couplings to SM particles implemented. In a future version of the code, the user will be able to add partial widths of the mediator particles to visible channels. The cosmic-ray source functions `dscrsource` and `dscrsource_line` will then be correspondingly updated.

35.3 vdSIDM/ini: Model setup

Currently, the module has two concrete example models fully implemented, where DM is a massive Dirac fermion ψ_χ , DR is a massless fermion ψ_γ , and the mediator is either a vector V or a scalar ϕ . The interaction parts of the respective Lagrangians are thus given by

$$\Delta\mathcal{L}_{\text{vector}} = g_\chi \bar{\psi}_\chi \not{V} \psi_\chi + g_\gamma \bar{\psi}_\gamma \not{V} \psi_\gamma \quad (35.6)$$

and

$$\Delta\mathcal{L}_{\text{scalar}} = g_\chi \bar{\psi}_\chi \psi_\chi \phi + g_\gamma \bar{\psi}_\gamma \psi_\gamma \phi. \quad (35.7)$$

Calling `dsgivemodel_vdSIDM_vector` or `dsgivemodel_vdSIDM_scalar` sets the DM mass, mediator mass and couplings for these models after which, as usual, the model is set up by a call to `dsmodelsetup`. This routine calculates for example the total decay width of the mediator, for which we add the contributions from decay to DR and (for heavy mediators, if kinematically allowed) decay into DM particles.

35.4 vdSIDM/kd: Kinetic decoupling

DM particles are kept in local thermal equilibrium via scattering with the DR particles. This is dominated by t -channel exchange of mediator particles, for which we implement the momentum-

averaged scattering amplitude returned from Ref. [79]:

$$\langle |\mathcal{M}|^2 \rangle_t^{\text{vector}} = \frac{256}{3} g_\chi^2 g_\gamma^2 \left(\frac{m_\chi}{m_V} \right)^4 \left(\frac{\omega}{m_\chi} \right)^2, \quad (35.8)$$

$$\langle |\mathcal{M}|^2 \rangle_t^{\text{scalar}} = \frac{512}{3} g_\chi^2 g_\gamma^2 \left(\frac{m_\chi}{m_\phi} \right)^4 \left(\frac{\omega}{m_\chi} \right)^2, \quad (35.9)$$

where ω is the energy of the DR particle. We also add the amplitudes for DM scattering directly off heat bath vector mediators, $\langle |\mathcal{M}|^2 \rangle_t = 64g_\chi^4/3$, and scalar mediators, $\langle |\mathcal{M}|^2 \rangle_t = 16g_\chi^4/3$ [79]. The interface function `dskdm2` returns the sum of these two contributions.

35.5 vdSIDM/rd: Relic density

In `src_models/vdSIDM/rd` we collect a set of routines to handle the temperature of the dark sector in this module. First of all, the interface function `dsrdxi` returns the temperature ratio $\xi = T_{\tilde{\gamma}}/T$, where T is the photon temperature. This is calculated by assuming that the two sectors were in thermal equilibrium ($\xi = 1$) at very early times when the DM particles were still relativistic, e.g. due to some portal that is effective only at very high energies and hence not relevant for the low-energy Lagrangian. After decoupling of the two sectors at a temperature $T_{\text{dc}} \gg m_\chi$, entropy would be separately conserved, which implies that the temperature evolves as

$$\xi(T) = \frac{[g_*^{\text{SM}}(T)/g_*^{\text{DS}}(T)]^{\frac{1}{3}}}{[g_*^{\text{SM}}(T_{\text{dc}})/g_*^{\text{DS}}(T_{\text{dc}})]^{\frac{1}{3}}}, \quad (35.10)$$

where $g_*^{\text{SM,DS}}$ are the entropy degrees of freedom in the visible and dark sector, respectively. Of course, `dsrdxi` could be replaced in the usual way by any user-supplied function in order to implement a different scenario for how the temperature of the dark sector evolves with time. Care must be taken, however, to ensure that such an implementation is consistent with the model assumptions. A constant value of ξ for example, as is often assumed for illustration, is *not* consistent with the assumption that there is no interaction between dark and visible sector.

Any additional relativistic energy density, as provided by the DR particles, is conventionally stated in terms of

$$\Delta N_{\text{eff}} \equiv \frac{\rho_{\text{DS}}}{\rho_{1\nu}} = \frac{4}{7} g_*^{\text{DS}} \left(\frac{T_{\tilde{\gamma}}}{T_\nu} \right)^4, \quad (35.11)$$

where $\rho_{1\nu}$ is the energy density contributed by one massless neutrino species, and T_ν is the neutrino temperature (which differs from the photon temperature after e^\pm annihilation). For convenience, DarkSUSY provides a function `dsrddeltaneff` to compute ΔN_{eff} as a function of (photon) temperature.

35.6 vdSIDM/si: Dark matter self-interactions

The interface function `dssisigtm` returns the transfer cross section σ_T for a Yukawa potential, by implementing the explicit expressions for the various scattering regimes discussed in detail in Section 27.1.

Bibliography

- [1] P. Gondolo, J. Edsjö, P. Ullio, L. Bergström, M. Schelke and E. A. Baltz, *DarkSUSY: Computing supersymmetric dark matter properties numerically*, *JCAP* **0407** (2004) 008, [[astro-ph/0406204](#)].
- [2] T. Bringmann, J. Edsjö, P. Gondolo, P. Ullio and L. Bergström, *DarkSUSY 6: An Advanced Tool to Compute Dark Matter Properties Numerically*, .
- [3] H. Zhao, *Analytical models for galactic nuclei*, *Mon. Not. Roy. Astron. Soc.* **278** (1996) 488–496, [[astro-ph/9509122](#)].
- [4] V. Silveira and A. Zee, *SCALAR PHANTOMS*, *Phys. Lett.* **B161** (1985) 136–140.
- [5] J. McDonald, *Gauge singlet scalars as cold dark matter*, *Phys. Rev.* **D50** (1994) 3637–3649, [[hep-ph/0702143](#)].
- [6] C. P. Burgess, M. Pospelov and T. ter Veldhuis, *The Minimal model of nonbaryonic dark matter: A Singlet scalar*, *Nucl. Phys.* **B619** (2001) 709–728, [[hep-ph/0011335](#)].
- [7] J. M. Cline, K. Kainulainen, P. Scott and C. Weniger, *Update on scalar singlet dark matter*, *Phys. Rev.* **D88** (2013) 055025, [[1306.4710](#)].
- [8] J. Einasto, *On the Construction of a Composite Model for the Galaxy and on the Determination of the System of Galactic Parameters*, *Trudy Astrofizicheskogo Instituta Alma-Ata* **5** (1965) 87–100.
- [9] J. F. Navarro, C. S. Frenk and S. D. M. White, *The Structure of cold dark matter halos*, *Astrophys. J.* **462** (1996) 563–575, [[astro-ph/9508025](#)].
- [10] T. Sjöstrand, S. Mrenna and P. Z. Skands, *PYTHIA 6.4 Physics and Manual*, *JHEP* **05** (2006) 026, [[hep-ph/0603175](#)].
- [11] M. Cirelli, G. Corcella, A. Hektor, G. Hutsi, M. Kadastik, P. Panci et al., *PPPC 4 DM ID: A Poor Particle Physicist Cookbook for Dark Matter Indirect Detection*, *JCAP* **1103** (2011) 051, [[1012.4515](#)].
- [12] P. Gondolo and G. Gelmini, *Cosmic abundances of stable particles: Improved analysis*, *Nucl. Phys.* **B360** (1991) 145–179.
- [13] J. Edsjö and P. Gondolo, *Neutralino relic density including coannihilations*, *Phys. Rev.* **D56** (1997) 1879–1894, [[hep-ph/9704361](#)].
- [14] J. Edsjö, M. Schelke, P. Ullio and P. Gondolo, *Accurate relic densities with neutralino, chargino and sfermion coannihilations in mSUGRA*, *JCAP* **0304** (2003) 001, [[hep-ph/0301106](#)].

- [15] T. Binder, T. Bringmann, M. Gustafsson and A. Hryczuk, *DRAKE: Dark matter Relic Abundance beyond Kinetic Equilibrium*, *Eur. Phys. J. C* **81** (2021) 577, [2103.01944].
- [16] T. Bringmann, S. Heeba, F. Kahlhoefer and K. Vangsnes, *Freezing-in a hot bath: resonances, medium effects and phase transitions*, 2111.14871.
- [17] T. Bringmann, *Particle Models and the Small-Scale Structure of Dark Matter*, *New J. Phys.* **11** (2009) 105027, [0903.0189].
- [18] L. Bergström, J. Edsjö and P. Ullio, *Possible indications of a clumpy dark matter halo*, *Phys. Rev.* **D58** (1998) 083507, [astro-ph/9804050].
- [19] L. Bergström, J. Edsjö and P. Gondolo, *Indirect detection of dark matter in km size neutrino telescopes*, *Phys. Rev.* **D58** (1998) 103519, [hep-ph/9806293].
- [20] E. A. Baltz and J. Edsjo, *Positron propagation and fluxes from neutralino annihilation in the halo*, *Phys. Rev.* **D59** (1998) 023511, [astro-ph/9808243].
- [21] L. Bergström, J. Edsjö and P. Ullio, *Cosmic anti-protons as a probe for supersymmetric dark matter?*, *Astrophys. J.* **526** (1999) 215–235, [astro-ph/9902012].
- [22] T. Bringmann, A. J. Galea and P. Walia, *Leading QCD Corrections for Indirect Dark Matter Searches: a Fresh Look*, *Phys. Rev.* **D93** (2016) 043529, [1510.02473].
- [23] T. Bringmann and M. Pospelov, *Novel direct detection constraints on light dark matter*, *Phys. Rev. Lett.* **122** (2019) 171801, [1810.10543].
- [24] K. Bondarenko, A. Boyarsky, T. Bringmann, M. Hufnagel, K. Schmidt-Hoberg and A. Sokolenko, *Direct detection and complementary constraints for sub-GeV dark matter*, *JHEP* **03** (2020) 118, [1909.08632].
- [25] L. Bergström and P. Gondolo, *Limits on direct detection of neutralino dark matter from $b \rightarrow s$ gamma decays*, *Astropart. Phys.* **5** (1996) 263–278, [hep-ph/9510252].
- [26] L. Bergström and P. Ullio, *Full one loop calculation of neutralino annihilation into two photons*, *Nucl. Phys.* **B504** (1997) 27–44, [hep-ph/9706232].
- [27] P. Ullio and L. Bergström, *Neutralino annihilation into a photon and a Z boson*, *Phys. Rev.* **D57** (1998) 1962–1971, [hep-ph/9707333].
- [28] T. Bringmann, L. Bergström and J. Edsjö, *New Gamma-Ray Contributions to Supersymmetric Dark Matter Annihilation*, *JHEP* **01** (2008) 049, [0710.3169].
- [29] L. Bergström, T. Bringmann and J. Edsjö, *New Positron Spectral Features from Supersymmetric Dark Matter - a Way to Explain the PAMELA Data?*, *Phys. Rev.* **D78** (2008) 103520, [0808.3725].
- [30] T. Bringmann and F. Calore, *Significant Enhancement of Neutralino Dark Matter Annihilation from Electroweak Bremsstrahlung*, *Phys. Rev. Lett.* **112** (2014) 071301, [1308.1089].
- [31] T. Bringmann, F. Calore, A. Galea and M. Garny, *Electroweak and Higgs Boson Internal Bremsstrahlung: General considerations for Majorana dark matter annihilation and application to MSSM neutralinos*, *JHEP* **09** (2017) 041, [1705.03466].
- [32] L. Bergström, T. Bringmann, I. Cholis, D. Hooper and C. Weniger, *New limits on dark matter annihilation from AMS cosmic ray positron data*, *Phys. Rev. Lett.* **111** (2013) 171101, [1306.3983].

- [33] A. W. Strong, I. V. Moskalenko and V. S. Ptuskin, *Cosmic-ray propagation and interactions in the Galaxy*, *Ann. Rev. Nucl. Part. Sci.* **57** (2007) 285–327, [astro-ph/0701517].
- [34] A. W. Strong and I. V. Moskalenko, *Propagation of cosmic-ray nucleons in the galaxy*, *Astrophys. J.* **509** (1998) 212–228, [astro-ph/9807150].
- [35] C. Evoli, D. Gaggero, A. Vittino, G. Di Bernardo, M. Di Mauro, A. Ligorini et al., *Cosmic-ray propagation with DRAGON2: I. numerical solver and astrophysical ingredients*, *JCAP* **1702** (2017) 015, [1607.07886].
- [36] R. Kissmann, *PICARD: A novel code for the Galactic Cosmic Ray propagation problem*, *Astropart. Phys.* **55** (2014) 37–50, [1401.4035].
- [37] D. Maurin, F. Donato, R. Taillet and P. Salati, *Cosmic rays below $z=30$ in a diffusion model: new constraints on propagation parameters*, *Astrophys. J.* **555** (2001) 585–596, [astro-ph/0101231].
- [38] C. Evoli, I. Cholis, D. Grasso, L. Maccione and P. Ullio, *Antiprotons from dark matter annihilation in the Galaxy: astrophysical uncertainties*, *Phys. Rev.* **D85** (2012) 123511, [1108.0664].
- [39] S. Colafrancesco, S. Profumo and P. Ullio, *Multi-frequency analysis of neutralino dark matter annihilations in the Coma cluster*, *Astron. Astrophys.* **455** (2006) 21, [astro-ph/0507575].
- [40] M. S. Turner, *Probing the Structure of the Galactic Halo with gamma Rays Produced by WIMP Annihilations*, *Phys. Rev.* **D34** (1986) 1921.
- [41] J. R. Ipser and P. Sikivie, *Estimates of the Density of Dark Matter Near the Center of the Galaxy*, *Phys. Rev.* **D35** (1987) 3695.
- [42] K. Freese and J. Silk, *Halo gamma-rays from cold dark matter annihilation*, *Phys. Rev.* **D40** (1989) 3828–3833.
- [43] V. Berezhinsky, A. Bottino and G. Mignola, *High-energy gamma radiation from the galactic center due to neutralino annihilation*, *Phys. Lett.* **B325** (1994) 136–142, [hep-ph/9402215].
- [44] G. Lake, *Detectability of gamma-rays from clumps of dark matter*, *Nature* **346** (1990) 39–40.
- [45] J. Silk and A. Stebbins, *Clumpy cold dark matter*, *Astrophys. J.* **411** (1993) 439–449.
- [46] C. Calcano-Roldan and B. Moore, *The Surface brightness of dark matter: Unique signatures of neutralino annihilation in the galactic halo*, *Phys. Rev.* **D62** (2000) 123005, [astro-ph/0010056].
- [47] L. Bergström, J. Edsjö, P. Gondolo and P. Ullio, *Clumpy neutralino dark matter*, *Phys. Rev.* **D59** (1999) 043506, [astro-ph/9806072].
- [48] L. Bergström, J. Edsjö and C. Gunnarsson, *Neutralino gamma-ray signals from accreting halo dark matter*, *Phys. Rev.* **D63** (2001) 083515, [astro-ph/0012346].
- [49] E. A. Baltz, C. Briot, P. Salati, R. Taillet and J. Silk, *Detection of neutralino annihilation photons from external galaxies*, *Phys. Rev.* **D61** (2000) 023514, [astro-ph/9909112].
- [50] D. B. Cline and Y.-T. Gao, *Low-energy cosmic γ -rays from cosmological photino annihilation*, *Astron. Astrophys.* **231** (1990) L23–L26.
- [51] Y.-T. Gao, F. W. Stecker and D. B. Cline, *The Lightest supersymmetric particle and the extragalactic gamma-ray background*, *Astron. Astrophys.* **249** (1991) 1–4.

- [52] L. Bergström, J. Edsjö and P. Ullio, *Spectral gamma-ray signatures of cosmological dark matter annihilation*, *Phys. Rev. Lett.* **87** (2001) 251301, [astro-ph/0105048].
- [53] G. Bélanger, K. Kannike, A. Pukhov and M. Raidal, *Z_3 Scalar Singlet Dark Matter*, *JCAP* **1301** (2013) 022, [1211.1014].
- [54] G. Bélanger, K. Kannike, A. Pukhov and M. Raidal, *Minimal semi-annihilating Z_N scalar dark matter*, *JCAP* **1406** (2014) 021, [1403.4960].
- [55] P. Ko and Y. Tang, *Self-interacting scalar dark matter with local Z_3 symmetry*, *JCAP* **1405** (2014) 047, [1402.6449].
- [56] S.-M. Choi and H. M. Lee, *SIMP dark matter with gauged Z_3 symmetry*, *JHEP* **09** (2015) 063, [1505.00960].
- [57] C. Arina, T. Bringmann, J. Silk and M. Vollmann, *Enhanced Line Signals from Annihilating Kaluza-Klein Dark Matter*, *Phys. Rev.* **D90** (2014) 083506, [1409.0007].
- [58] S. Campbell, B. Dutta and E. Komatsu, *Effects of Velocity-Dependent Dark Matter Annihilation on the Energy Spectrum of the Extragalactic Gamma-ray Background*, *Phys. Rev.* **D82** (2010) 095007, [1009.3530].
- [59] M. Lattanzi and J. I. Silk, *Can the WIMP annihilation boost factor be boosted by the Sommerfeld enhancement?*, *Phys. Rev.* **D79** (2009) 083523, [0812.0360].
- [60] A. L. Fitzpatrick, W. Haxton, E. Katz, N. Lubbers and Y. Xu, *The Effective Field Theory of Dark Matter Direct Detection*, *JCAP* **1302** (2013) 004, [1203.3542].
- [61] P. Gondolo and S. Scopel, , *in prep.* (2018) .
- [62] J. Alvey, T. Bringmann and H. Kolesova, *No room to hide: implications of cosmic-ray upscattering for GeV-scale dark matter*, 2209.03360.
- [63] XENON collaboration, E. Aprile et al., *Dark Matter Search Results from a One Ton-Year Exposure of XENON1T*, *Phys. Rev. Lett.* **121** (2018) 111302, [1805.12562].
- [64] BOREXINO collaboration, G. Alimonti et al., *Science and technology of BOREXINO: A Real time detector for low-energy solar neutrinos*, *Astropart. Phys.* **16** (2002) 205–234, [hep-ex/0012030].
- [65] B. Dasgupta and J. F. Beacom, *Reconstruction of supernova ν_μ , ν_τ , anti- ν_μ , and anti- ν_τ neutrino spectra at scintillator detectors*, *Phys. Rev.* **D83** (2011) 113006, [1103.2768].
- [66] A. V. Maccio', A. A. Dutton and F. C. v. d. Bosch, *Concentration, Spin and Shape of Dark Matter Haloes as a Function of the Cosmological Model: WMAP1, WMAP3 and WMAP5 results*, *Mon. Not. Roy. Astron. Soc.* **391** (2008) 1940–1954, [0805.1926].
- [67] J. Schaye et al., *The EAGLE project: Simulating the evolution and assembly of galaxies and their environments*, *Mon. Not. Roy. Astron. Soc.* **446** (2015) 521–554, [1407.7040].
- [68] M. Schaller, C. S. Frenk, R. G. Bower, T. Theuns, A. Jenkins, J. Schaye et al., *Baryon effects on the internal structure of Λ CDM haloes in the EAGLE simulations*, *Mon. Not. Roy. Astron. Soc.* **451** (2015) 1247–1267, [1409.8617].
- [69] L. Wang, A. A. Dutton, G. S. Stinson, A. V. Macciò, C. Penzo, X. Kang et al., *NIHAO project – I. Reproducing the inefficiency of galaxy formation across cosmic time with a large sample of cosmological hydrodynamical simulations*, *Mon. Not. Roy. Astron. Soc.* **454** (2015) 83–94, [1503.04818].

- [70] E. Tollet et al., *NIHAO – IV: core creation and destruction in dark matter density profiles across cosmic time*, *Mon. Not. Roy. Astron. Soc.* **456** (2016) 3542–3552, [1507.03590].
- [71] A. Burkert, *The Structure of dark matter halos in dwarf galaxies*, *IAU Symp.* **171** (1996) 175, [astro-ph/9504041].
- [72] A. S. Eddington, *The dynamics of a stellar system. Third paper: oblate and other distributions*, *MNRAS* **76** (Nov., 1915) 37.
- [73] R. Catena and P. Ullio, *The local dark matter phase-space density and impact on WIMP direct detection*, *JCAP* **05** (2012) 005, [1111.3556].
- [74] L. J. Hall, K. Jedamzik, J. March-Russell and S. M. West, *Freeze-In Production of FIMP Dark Matter*, *JHEP* **03** (2010) 080, [0911.1120].
- [75] X. Chu, T. Hambye and M. H. G. Tytgat, *The Four Basic Ways of Creating Dark Matter Through a Portal*, *JCAP* **05** (2012) 034, [1112.0493].
- [76] X. Chu, Y. Mambrini, J. Quevillon and B. Zaldivar, *Thermal and non-thermal production of dark matter via Z' -portal(s)*, *JCAP* **01** (2014) 034, [1306.4677].
- [77] T. Bringmann and S. Hofmann, *Thermal decoupling of WIMPs from first principles*, *JCAP* **0704** (2007) 016, [hep-ph/0612238].
- [78] T. Binder, T. Bringmann, M. Gustafsson and A. Hryczuk, *Early kinetic decoupling of dark matter: when the standard way of calculating the thermal relic density fails*, *Phys. Rev.* **D96** (2017) 115010, [1706.07433].
- [79] T. Bringmann, H. T. Ihle, J. Kersten and P. Walia, *Suppressing structure formation at dwarf galaxy scales and below: late kinetic decoupling as a compelling alternative to warm dark matter*, *Phys. Rev.* **D94** (2016) 103529, [1603.04884].
- [80] J. Kasahara, *Neutralino Dark Matter: The Mass of the smallest Halo and the golden Region*, Ph.D. thesis, University of Utah, 2009.
- [81] P. Gondolo, J. Hisano and K. Kadota, *The Effect of quark interactions on dark matter kinetic decoupling and the mass of the smallest dark halos*, *Phys. Rev.* **D86** (2012) 083523, [1205.1914].
- [82] A. M. Green, S. Hofmann and D. J. Schwarz, *The First wimpy halos*, *JCAP* **0508** (2005) 003, [astro-ph/0503387].
- [83] A. Loeb and M. Zaldarriaga, *The Small-scale power spectrum of cold dark matter*, *Phys. Rev.* **D71** (2005) 103520, [astro-ph/0504112].
- [84] E. Bertschinger, *The Effects of Cold Dark Matter Decoupling and Pair Annihilation on Cosmological Perturbations*, *Phys. Rev.* **D74** (2006) 063509, [astro-ph/0607319].
- [85] J. Diemand, B. Moore and J. Stadel, *Earth-mass dark-matter haloes as the first structures in the early Universe*, *Nature* **433** (2005) 389–391, [astro-ph/0501589].
- [86] K. Griest and D. Seckel, *Three exceptions in the calculation of relic abundances*, *Phys. Rev.* **D43** (1991) 3191–3203.
- [87] M. Srednicki, R. Watkins and K. A. Olive, *Calculations of Relic Densities in the Early Universe*, *Nucl. Phys.* **B310** (1988) 693.
- [88] M. Drees, F. Hajkarim and E. R. Schmitz, *The Effects of QCD Equation of State on the Relic Density of WIMP Dark Matter*, *JCAP* **1506** (2015) 025, [1503.03513].

- [89] T. Bringmann, P. F. Depta, M. Hufnagel and K. Schmidt-Hoberg, *Precise dark matter relic abundance in decoupled sectors*, *Phys. Lett. B* **817** (2021) 136341, [2007.03696].
- [90] T. Bringmann and J. Edsjö, *DarkSUSY 6.3 – Freeze-in, out-of-equilibrium freeze-out, cosmic-ray upscattering and further new features*, *PoS CompTools2021* (2022) 038, [2203.07439].
- [91] J. Bernstein, *KINETIC THEORY IN THE EXPANDING UNIVERSE*. Cambridge Monographs on Mathematical Physics. Cambridge University Press, Cambridge, U.K., 1988, 10.1017/CBO9780511564185.
- [92] E. W. Kolb and M. S. Turner, *The Early Universe*, *Front. Phys.* **69** (1990) 1–547.
- [93] L. G. van den Aarssen, T. Bringmann and Y. C. Goedecke, *Thermal decoupling and the smallest subhalo mass in dark matter models with Sommerfeld-enhanced annihilation rates*, *Phys. Rev.* **D85** (2012) 123512, [1202.5456].
- [94] L. M. Krauss.
- [95] W. H. Press and D. N. Spergel, *Capture by the sun of a galactic population of weakly interacting massive particles*, *Astrophys. J.* **296** (1985) 679–684.
- [96] J. Silk, K. A. Olive and M. Srednicki, *The Photino, the Sun and High-Energy Neutrinos*, *Phys. Rev. Lett.* **55** (1985) 257–259.
- [97] L. M. Krauss, K. Freese, W. Press and D. Spergel, *Cold dark matter candidates and the solar neutrino problem*, *Astrophys. J.* **299** (1985) 1001.
- [98] L. M. Krauss, M. Srednicki and F. Wilczek, *Solar System Constraints and Signatures for Dark Matter Candidates*, *Phys. Rev.* **D33** (1986) 2079–2083.
- [99] T. K. Gaisser, G. Steigman and S. Tilav, *Limits on Cold Dark Matter Candidates from Deep Underground Detectors*, *Phys. Rev.* **D34** (1986) 2206.
- [100] K. Griest and D. Seckel, *Cosmic Asymmetry, Neutrinos and the Sun*, *Nucl. Phys.* **B283** (1987) 681–705.
- [101] J. S. Hagelin, K. W. Ng and K. A. Olive, *A High-energy Neutrino Signature From Supersymmetric Relics*, *Phys. Lett.* **B180** (1986) 375–380.
- [102] K. Freese, *Can Scalar Neutrinos Or Massive Dirac Neutrinos Be the Missing Mass?*, *Phys. Lett.* **167B** (1986) 295–300.
- [103] M. Kamionkowski, *Energetic neutrinos from heavy neutralino annihilation in the sun*, *Phys. Rev.* **D44** (1991) 3021–3042.
- [104] F. Halzen, T. Stelzer and M. Kamionkowski, *Signatures of dark matter in underground detectors*, *Phys. Rev.* **D45** (1992) 4439–4442.
- [105] A. Bottino, V. de Alfaro, N. Fornengo, G. Mignola and M. Pignone, *Indirect search for neutralinos at neutrino telescopes*, *Phys. Lett.* **B265** (1991) 57–63.
- [106] A. Bottino, N. Fornengo, G. Mignola and L. Moscoso, *Signals of neutralino dark matter from earth and sun*, *Astropart. Phys.* **3** (1995) 65–76, [hep-ph/9408391].
- [107] R. Gandhi, J. L. Lopez, D. V. Nanopoulos, K.-j. Yuan and A. Zichichi, *Scrutinizing supergravity models through neutrino telescopes*, *Phys. Rev.* **D49** (1994) 3691–3703, [astro-ph/9309048].

- [108] L. Bergström, J. Edsjö and P. Gondolo, *Indirect neutralino detection rates in neutrino telescopes*, *Phys. Rev.* **D55** (1997) 1765–1770, [[hep-ph/9607237](#)].
- [109] F. Halzen, *Status of neutrino astronomy: The Quest for kilometer scale instruments*, *Comments Nucl. Part. Phys.* **22** (1997) 155–170, [[astro-ph/9701029](#)].
- [110] S. Ritz and D. Seckel, *Detailed Neutrino Spectra From Cold Dark Matter Annihilations in the Sun*, *Nucl. Phys.* **B304** (1988) 877–908.
- [111] G. F. Giudice and E. Roulet, *Energetic Neutrinos From Supersymmetric Dark Matter*, *Nucl. Phys.* **B316** (1989) 429–442.
- [112] M. Drees, G. Jungman, M. Kamionkowski and M. M. Nojiri, *Neutralino annihilation into gluons*, *Phys. Rev.* **D49** (1994) 636–647, [[hep-ph/9306325](#)].
- [113] G. Jungman and M. Kamionkowski, *Neutrinos from particle decay in the sun and earth*, *Phys. Rev.* **D51** (1995) 328–340, [[hep-ph/9407351](#)].
- [114] V. Berezhinsky, A. Bottino, J. R. Ellis, N. Fornengo, G. Mignola and S. Scopel, *Searching for relic neutralinos using neutrino telescopes*, *Astropart. Phys.* **5** (1996) 333–352, [[hep-ph/9603342](#)].
- [115] M. Kamionkowski, K. Griest, G. Jungman and B. Sadoulet, *Model independent comparison of direct versus indirect detection of supersymmetric dark matter*, *Phys. Rev. Lett.* **74** (1995) 5174–5177, [[hep-ph/9412213](#)].
- [116] J. Edsjö and P. Gondolo, *WIMP mass determination with neutrino telescopes*, *Phys. Lett.* **B357** (1995) 595–601, [[hep-ph/9504283](#)].
- [117] L. Bergström, J. Edsjö and M. Kamionkowski, *Astrophysical neutrino detection with angular and energy resolution*, *Astropart. Phys.* **7** (1997) 147–160, [[astro-ph/9702037](#)].
- [118] G. Jungman, M. Kamionkowski and K. Griest, *Supersymmetric dark matter*, *Phys. Rept.* **267** (1996) 195–373, [[hep-ph/9506380](#)].
- [119] A. Gould, *Resonant Enhancements in WIMP Capture by the Earth*, *Astrophys. J.* **321** (1987) 571.
- [120] J. N. Bahcall, M. H. Pinsonneault and S. Basu, *Solar models: Current epoch and time dependences, neutrinos, and helioseismological properties*, *Astrophys. J.* **555** (2001) 990–1012, [[astro-ph/0010346](#)].
- [121] W. F. McDonough, *Compositional Model for the Earth’s Core*, *Treatise on Geochemistry* **2** (Dec., 2003) 568.
- [122] *The Earth: its properties, composition, and structure*. Encyclopædia Britannica, Inc., 1999.
- [123] A. Gould, *Gravitational diffusion of solar system WIMPs*, *Astrophys. J.* **368** (Feb., 1991) 610–615.
- [124] P. Farinella, C. Froeschlé, C. Froeschlé, R. Gonczi, G. Hahn, A. Morbidelli et al., *Asteroids falling into the Sun*, *Nature* **371** (Sept., 1994) 314–317.
- [125] A. Gould and S. M. Khairul Alam, *Can heavy WIMPs be captured by the earth?*, *Astrophys. J.* **549** (2001) 72–75, [[astro-ph/9911288](#)].
- [126] J. Lundberg and J. Edsjö, *WIMP diffusion in the solar system including solar depletion and its effect on earth capture rates*, *Phys. Rev.* **D69** (2004) 123505, [[astro-ph/0401113](#)].

- [127] J. Edsjö, “WimpSim Neutrino Monte Carlo.” <http://www.fysik.su.se/~edsjo/wimpsim/>, 2007.
- [128] J. Edsjö, *Aspects of neutrino detection of neutralino dark matter*, Ph.D. thesis, Uppsala U., 1997. [hep-ph/9704384](https://arxiv.org/abs/hep-ph/9704384).
- [129] J. Edsjö, “NuSigma Neutrino Interaction Monte Carlo.” <http://www.fysik.su.se/~edsjo/wimpsim/>, 2007.
- [130] PARTICLE DATA GROUP collaboration, D. E. Groom et al., *Review of particle physics. Particle Data Group, Eur. Phys. J.* **C15** (2000) 1–878.
- [131] M. Blennow, J. Edsjö and T. Ohlsson, *Neutrinos from WIMP annihilations using a full three-flavor Monte Carlo*, *JCAP* **0801** (2008) 021, [[0709.3898](https://arxiv.org/abs/0709.3898)].
- [132] D. N. Spergel and P. J. Steinhardt, *Observational evidence for selfinteracting cold dark matter*, *Phys. Rev. Lett.* **84** (2000) 3760–3763, [[astro-ph/9909386](https://arxiv.org/abs/astro-ph/9909386)].
- [133] A. Loeb and N. Weiner, *Cores in Dwarf Galaxies from Dark Matter with a Yukawa Potential*, *Phys. Rev. Lett.* **106** (2011) 171302, [[1011.6374](https://arxiv.org/abs/1011.6374)].
- [134] M. Vogelsberger, J. Zavala and A. Loeb, *Subhaloes in Self-Interacting Galactic Dark Matter Haloes*, *Mon. Not. Roy. Astron. Soc.* **423** (2012) 3740, [[1201.5892](https://arxiv.org/abs/1201.5892)].
- [135] A. H. G. Peter, M. Rocha, J. S. Bullock and M. Kaplinghat, *Cosmological Simulations with Self-Interacting Dark Matter II: Halo Shapes vs. Observations*, *Mon. Not. Roy. Astron. Soc.* **430** (2013) 105, [[1208.3026](https://arxiv.org/abs/1208.3026)].
- [136] J. Zavala, M. Vogelsberger and M. G. Walker, *Constraining Self-Interacting Dark Matter with the Milky Way’s dwarf spheroidals*, *Mon. Not. Roy. Astron. Soc.* **431** (2013) L20–L24, [[1211.6426](https://arxiv.org/abs/1211.6426)].
- [137] L. G. van den Aarssen, T. Bringmann and C. Pfrommer, *Is dark matter with long-range interactions a solution to all small-scale problems of Λ CDM cosmology?*, *Phys. Rev. Lett.* **109** (2012) 231301, [[1205.5809](https://arxiv.org/abs/1205.5809)].
- [138] O. D. Elbert, J. S. Bullock, S. Garrison-Kimmel, M. Rocha, J. Onorbe and A. H. G. Peter, *Core formation in dwarf haloes with self-interacting dark matter: no fine-tuning necessary*, *Mon. Not. Roy. Astron. Soc.* **453** (2015) 29–37, [[1412.1477](https://arxiv.org/abs/1412.1477)].
- [139] A. Kamada, M. Kaplinghat, A. B. Pace and H.-B. Yu, *How the Self-Interacting Dark Matter Model Explains the Diverse Galactic Rotation Curves*, *Phys. Rev. Lett.* **119** (2017) 111102, [[1611.02716](https://arxiv.org/abs/1611.02716)].
- [140] A. Robertson et al., *The diverse density profiles of galaxy clusters with self-interacting dark matter plus baryons*, *Mon. Not. Roy. Astron. Soc.* **476** (2018) L20–L24, [[1711.09096](https://arxiv.org/abs/1711.09096)].
- [141] J. S. Bullock and M. Boylan-Kolchin, *Small-Scale Challenges to the Λ CDM Paradigm*, *Ann. Rev. Astron. Astrophys.* **55** (2017) 343–387, [[1707.04256](https://arxiv.org/abs/1707.04256)].
- [142] R. A. Flores and J. R. Primack, *Observational and theoretical constraints on singular dark matter halos*, *Astrophys. J.* **427** (1994) L1–4, [[astro-ph/9402004](https://arxiv.org/abs/astro-ph/9402004)].
- [143] B. Moore, *Evidence against dissipationless dark matter from observations of galaxy haloes*, *Nature* **370** (1994) 629.
- [144] M. Boylan-Kolchin, J. S. Bullock and M. Kaplinghat, *Too big to fail? The puzzling darkness of massive Milky Way subhaloes*, *Mon. Not. Roy. Astron. Soc.* **415** (2011) L40, [[1103.0007](https://arxiv.org/abs/1103.0007)].

- [145] M. Boylan-Kolchin, J. S. Bullock and M. Kaplinghat, *The Milky Way's bright satellites as an apparent failure of Λ CDM*, *Mon. Not. Roy. Astron. Soc.* **422** (2012) 1203–1218, [1111.2048].
- [146] K. A. Oman et al., *The unexpected diversity of dwarf galaxy rotation curves*, *Mon. Not. Roy. Astron. Soc.* **452** (2015) 3650–3665, [1504.01437].
- [147] K. A. Oman, J. F. Navarro, L. V. Sales, A. Fattahi, C. S. Frenk, T. Sawala et al., *Missing dark matter in dwarf galaxies?*, *Mon. Not. Roy. Astron. Soc.* **460** (2016) 3610–3623, [1601.01026].
- [148] B. Moore, S. Ghigna, F. Governato, G. Lake, T. R. Quinn, J. Stadel et al., *Dark matter substructure within galactic halos*, *Astrophys. J.* **524** (1999) L19–L22, [astro-ph/9907411].
- [149] A. A. Klypin, A. V. Kravtsov, O. Valenzuela and F. Prada, *Where are the missing Galactic satellites?*, *Astrophys. J.* **522** (1999) 82–92, [astro-ph/9901240].
- [150] S. Tulin, H.-B. Yu and K. M. Zurek, *Beyond Collisionless Dark Matter: Particle Physics Dynamics for Dark Matter Halo Structure*, *Phys. Rev.* **D87** (2013) 115007, [1302.3898].
- [151] F. Kahlhoefer, K. Schmidt-Hoberg and S. Wild, *Dark matter self-interactions from a general spin-0 mediator*, *JCAP* **1708** (2017) 003, [1704.02149].
- [152] S. Tulin and H.-B. Yu, *Dark Matter Self-interactions and Small Scale Structure*, *Phys. Rept.* **730** (2018) 1–57, [1705.02358].
- [153] J. L. Feng, M. Kaplinghat and H.-B. Yu, *Halo Shape and Relic Density Exclusions of Sommerfeld-Enhanced Dark Matter Explanations of Cosmic Ray Excesses*, *Phys. Rev. Lett.* **104** (2010) 151301, [0911.0422].
- [154] F.-Y. Cyr-Racine, K. Sigurdson, J. Zavala, T. Bringmann, M. Vogelsberger and C. Pfrommer, *ETHOS, An effective theory of structure formation: From dark particle physics to the matter distribution of the Universe*, *Phys. Rev.* **D93** (2016) 123527, [1512.05344].
- [155] S. A. Khrapak, A. V. Ivlev, G. E. Morfill and S. K. Zhdanov, *Scattering in the Attractive Yukawa Potential in the Limit of Strong Interaction*, *Phys. Rev. Lett.* **90** (2003) 225002.
- [156] S. A. Khrapak, A. V. Ivlev and G. E. Morfill, *Momentum transfer in complex plasmas*, *Phys. Rev. E* **70** (Nov, 2004) 056405.
- [157] Y. L. Luke, *The Special Functions and Their Approximations*, vol. II of *Mathematics in Science and Engineering, Volume 53-II*, Editor: Richard Bellman. 1969.
- [158] K. G. Chetyrkin, J. H. Kuhn and M. Steinhauser, *RunDec: A Mathematica package for running and decoupling of the strong coupling and quark masses*, *Comput. Phys. Commun.* **133** (2000) 43–65, [hep-ph/0004189].
- [159] H. E. Haber and G. L. Kane, *The Search for Supersymmetry: Probing Physics Beyond the Standard Model*, *Phys. Rept.* **117** (1985) 75–263.
- [160] J. F. Gunion and H. E. Haber, *Higgs Bosons in Supersymmetric Models. 1.*, *Nucl. Phys.* **B272** (1986) 1.
- [161] F. E. Paige, S. D. Protopopescu, H. Baer and X. Tata, *ISAJET 7.69: A Monte Carlo event generator for pp , anti- $p p$, and $e+e-$ reactions*, hep-ph/0312045.
- [162] <http://www.phy.bnl.gov/~isajet>.

- [163] P. Z. Skands et al., *SUSY Les Houches accord: Interfacing SUSY spectrum calculators, decay packages, and event generators*, *JHEP* **07** (2004) 036, [[hep-ph/0311123](#)].
- [164] B. C. Allanach et al., *SUSY Les Houches Accord 2*, *Comput. Phys. Commun.* **180** (2009) 8–25, [[0801.0045](#)].
- [165] M. Drees, M. M. Nojiri, D. P. Roy and Y. Yamada, *Light Higgsino dark matter*, *Phys. Rev.* **D56** (1997) 276–290, [[hep-ph/9701219](#)].
- [166] D. Pierce and A. Papadopoulos, *Radiative corrections to neutralino and chargino masses in the minimal supersymmetric model*, *Phys. Rev.* **D50** (1994) 565–570, [[hep-ph/9312248](#)].
- [167] A. B. Lahanas, K. Tamvakis and N. D. Tracas, *One loop corrections to the neutralino sector and radiative electroweak breaking in the MSSM*, *Phys. Lett.* **B324** (1994) 387–396, [[hep-ph/9312251](#)].
- [168] S. Heinemeyer, W. Hollik and G. Weiglein, *FeynHiggs: A Program for the calculation of the masses of the neutral CP even Higgs bosons in the MSSM*, *Comput. Phys. Commun.* **124** (2000) 76–89, [[hep-ph/9812320](#)].
- [169] <http://www.feynhiggs.de>.
- [170] S. Heinemeyer, W. Hollik and G. Weiglein, *QCD corrections to the masses of the neutral CP - even Higgs bosons in the MSSM*, *Phys. Rev.* **D58** (1998) 091701, [[hep-ph/9803277](#)].
- [171] S. Heinemeyer, W. Hollik and G. Weiglein, *The Masses of the neutral CP - even Higgs bosons in the MSSM: Accurate analysis at the two loop level*, *Eur. Phys. J.* **C9** (1999) 343–366, [[hep-ph/9812472](#)].
- [172] S. Heinemeyer, W. Hollik and G. Weiglein, *The Mass of the lightest MSSM Higgs boson: A Compact analytical expression at the two loop level*, *Phys. Lett.* **B455** (1999) 179–191, [[hep-ph/9903404](#)].
- [173] T. Bringmann et al., *DarkBit: A GAMBIT module for computing dark matter observables and likelihoods*, *Eur. Phys. J.* **C77** (2017) 831, [[1705.07920](#)].
- [174] GAMBIT collaboration, C. Balázs et al., *ColliderBit: a GAMBIT module for the calculation of high-energy collider observables and likelihoods*, *Eur. Phys. J.* **C77** (2017) 795, [[1705.07919](#)].
- [175] P. Bechtle, O. Brein, S. Heinemeyer, G. Weiglein and K. E. Williams, *HiggsBounds: Confronting Arbitrary Higgs Sectors with Exclusion Bounds from LEP and the Tevatron*, *Comput. Phys. Commun.* **181** (2010) 138–167, [[0811.4169](#)].
- [176] ATLAS collaboration, G. Aad et al., *Search for squarks and gluinos with the ATLAS detector in final states with jets and missing transverse momentum using $\sqrt{s} = 8$ TeV proton–proton collision data*, *JHEP* **09** (2014) 176, [[1405.7875](#)].
- [177] F. Mahmoudi, *SuperIso: A Program for calculating the isospin asymmetry of $B \rightarrow K^* \gamma$ in the MSSM*, *Comput. Phys. Commun.* **178** (2008) 745–754, [[0710.2067](#)].
- [178] THE GAMBIT FLAVOUR WORKGROUP collaboration, F. U. Bernlochner et al., *FlavBit: A GAMBIT module for computing flavour observables and likelihoods*, *Eur. Phys. J.* **C77** (2017) 786, [[1705.07933](#)].
- [179] BABAR collaboration, J. P. Lees et al., *Exclusive Measurements of $b \rightarrow s \gamma$ Transition Rate and Photon Energy Spectrum*, *Phys. Rev.* **D86** (2012) 052012, [[1207.2520](#)].

- [180] BABAR collaboration, J. P. Lees et al., *Precision Measurement of the $B \rightarrow X_s \gamma$ Photon Energy Spectrum, Branching Fraction, and Direct CP Asymmetry $A_{CP}(B \rightarrow X_{s+d} \gamma)$* , *Phys. Rev. Lett.* **109** (2012) 191801, [1207.2690].
- [181] BELLE collaboration, A. Abdesselam et al., *Measurement of the inclusive $B \rightarrow X_{s+d} \gamma$ branching fraction, photon energy spectrum and HQE parameters*, in *Proceedings, 38th International Conference on High Energy Physics (ICHEP 2016): Chicago, IL, USA, August 3-10, 2016*, 2016, 1608.02344, <https://inspirehep.net/record/1479946/files/arXiv:1608.02344.pdf>.
- [182] LHCb collaboration, R. Aaij et al., *Measurement of the $B_s^0 \rightarrow \mu^+ \mu^-$ branching fraction and effective lifetime and search for $B^0 \rightarrow \mu^+ \mu^-$ decays*, *Phys. Rev. Lett.* **118** (2017) 191801, [1703.05747].
- [183] T. Moroi, *The Muon anomalous magnetic dipole moment in the minimal supersymmetric standard model*, *Phys. Rev.* **D53** (1996) 6565–6575, [hep-ph/9512396].
- [184] MUON G-2 collaboration, G. W. Bennett et al., *Final Report of the Muon E821 Anomalous Magnetic Moment Measurement at BNL*, *Phys. Rev.* **D73** (2006) 072003, [hep-ex/0602035].
- [185] GAMBIT collaboration, P. Athron et al., *SpecBit, DecayBit and PrecisionBit: GAMBIT modules for computing mass spectra, particle decay rates and precision observables*, 1705.07936.
- [186] A. C. Hearn, *Reduce 3.5*, 1993.
- [187] J. Edsjö, “form2f, perl script to convert form output to fortran.”
- [188] J. Hisano, S. Matsumoto and M. M. Nojiri, *Explosive dark matter annihilation*, *Phys. Rev. Lett.* **92** (2004) 031303, [hep-ph/0307216].
- [189] J. Hisano, S. Matsumoto, M. M. Nojiri and O. Saito, *Non-perturbative effect on dark matter annihilation and gamma ray signature from galactic center*, *Phys. Rev.* **D71** (2005) 063528, [hep-ph/0412403].
- [190] A. Birkedal, K. T. Matchev, M. Perelstein and A. Spray, *Robust gamma ray signature of WIMP dark matter*, hep-ph/0507194.
- [191] L. Bergström, T. Bringmann, M. Eriksson and M. Gustafsson, *Gamma rays from Kaluza-Klein dark matter*, *Phys. Rev. Lett.* **94** (2005) 131301, [astro-ph/0410359].
- [192] L. Bergström, *Radiative Processes in Dark Matter Photino Annihilation*, *Phys. Lett.* **B225** (1989) 372–380.
- [193] L. Bergström, T. Bringmann, M. Eriksson and M. Gustafsson, *Gamma rays from heavy neutralino dark matter*, *Phys. Rev. Lett.* **95** (2005) 241301, [hep-ph/0507229].
- [194] M. W. Goodman and E. Witten, *Detectability of Certain Dark Matter Candidates*, *Phys. Rev.* **D31** (1985) 3059.
- [195] A. Bottino, F. Donato, G. Mignola, S. Scopel, P. Belli and A. Incicchitti, *Exploring the supersymmetric parameter space by direct search for WIMPs*, *Phys. Lett.* **B402** (1997) 113–121, [hep-ph/9612451].
- [196] J. R. Ellis and R. A. Flores, *Realistic Predictions for the Detection of Supersymmetric Dark Matter*, *Nucl. Phys.* **B307** (1988) 883–908.

- [197] J. R. Ellis and R. A. Flores, *Elastic supersymmetric relic - nucleus scattering revisited*, *Phys. Lett.* **B263** (1991) 259–266.
- [198] J. Engel, *Nuclear form-factors for the scattering of weakly interacting massive particles*, *Phys. Lett.* **B264** (1991) 114–119.
- [199] K. Griest, *Cross-Sections, Relic Abundance and Detection Rates for Neutralino Dark Matter*, *Phys. Rev.* **D38** (1988) 2357.
- [200] R. Barbieri, M. Frigeni and G. F. Giudice, *Dark Matter Neutralinos in Supergravity Theories*, *Nucl. Phys.* **B313** (1989) 725–735.
- [201] G. B. Gelmini, P. Gondolo and E. Roulet, *Neutralino dark matter searches*, *Nucl. Phys.* **B351** (1991) 623–644.
- [202] A. Bottino, V. de Alfaro, N. Fornengo, S. Mignola and S. Scopel, *On the neutralino as dark matter candidate. 2. Direct detection.*, *Astropart. Phys.* **2** (1994) 77–90, [[hep-ph/9309219](#)].
- [203] J. Gasser, H. Leutwyler and M. E. Sainio, *Sigma term update*, *Phys. Lett.* **B253** (1991) 252–259.
- [204] SPIN MUON collaboration, D. Adams et al., *A New measurement of the spin dependent structure function $g_1(x)$ of the deuteron*, *Phys. Lett.* **B357** (1995) 248–254.
- [205] R. L. Jaffe and A. Manohar, *The $G(1)$ Problem: Fact and Fantasy on the Spin of the Proton*, *Nucl. Phys.* **B337** (1990) 509–546.
- [206] J. Engel and P. Vogel, *Spin dependent cross-sections of weakly interacting massive particles on nuclei*, *Phys. Rev.* **D40** (1989) 3132–3135.
- [207] H. E. Haber and D. Wyler, *RADIATIVE NEUTRALINO DECAY*, *Nucl. Phys.* **B323** (1989) 267–310.
- [208] S. Dimopoulos and S. D. Thomas, *Dynamical relaxation of the supersymmetric CP violating phases*, *Nucl. Phys.* **B465** (1996) 23–33, [[hep-ph/9510220](#)].
- [209] J. R. Ellis, G. Ridolfi and F. Zwirner, *Radiative corrections to the masses of supersymmetric Higgs bosons*, *Phys. Lett.* **B257** (1991) 83–91.
- [210] J. R. Ellis, G. Ridolfi and F. Zwirner, *On radiative corrections to supersymmetric Higgs boson masses and their implications for LEP searches*, *Phys. Lett.* **B262** (1991) 477–484.
- [211] A. Brignole, J. R. Ellis, G. Ridolfi and F. Zwirner, *The Supersymmetric charged Higgs boson mass and LEP phenomenology*, *Phys. Lett.* **B271** (1991) 123–132.
- [212] M. Drees and M. M. Nojiri, *One loop corrections to the Higgs sector in minimal supergravity models*, *Phys. Rev.* **D45** (1992) 2482–2492.
- [213] M. Carena, J. R. Espinosa, M. Quiros and C. E. M. Wagner, *Analytical expressions for radiatively corrected Higgs masses and couplings in the MSSM*, *Phys. Lett.* **B355** (1995) 209–221, [[hep-ph/9504316](#)].
- [214] M. Carena, M. Quiros and C. E. M. Wagner, *Effective potential methods and the Higgs mass spectrum in the MSSM*, *Nucl. Phys.* **B461** (1996) 407–436, [[hep-ph/9508343](#)].
- [215] T. Hahn, S. Heinemeyer, W. Hollik, H. Rzehak and G. Weiglein, *FeynHiggs 2.7*, *Nucl. Phys. Proc. Suppl.* **205-206** (2010) 152–157, [[1007.0956](#)].

- [216] PARTICLE DATA GROUP collaboration, C. Patrignani et al., *Review of Particle Physics*, *Chin. Phys.* **C40** (2016) 100001.
- [217] H. E. Haber, *Perspectives in Higgs Physics II*. World Scientific, 1997.
- [218] T. Hahn, *SUSY Les Houches Accord 2 I/O made easy*, *Comput. Phys. Commun.* **180** (2009) 1681–1693, [[hep-ph/0605049](#)].
- [219] S. Cassel, *Sommerfeld factor for arbitrary partial wave processes*, *J. Phys.* **G37** (2010) 105009, [[0903.5307](#)].