DarkSE User's Guide

Andrzej Hryczuk
SISSA
via Bonomea 265, 34136 Trieste, Italy
e-mail:hryczuk@sissa.it

January 13, 2022

Abstract

DarkSE is a package for DarkSUSY supplying routines for including the Sommerfeld enhancement computation and its impact on the neutralino relic density in a general MSSM setup. This guide gives a short introduction how to install and use the package.

Disclaimer: The version 1.02 is still a beta release. It has not been extensively tested for every possible problems. Moreover, there are some known issues which we discuss at the end of this guide. Also this manual is still work in progress. Its future development relies strongly on the development of the code itself. In case of any questions, comments or suggestions please do not hesitate to write to: hryczuk@sissa.it.

1 General information

DarkSE is a package for DarkSUSY [1] written in FORTRAN, consisting of routines for including the Sommerfeld enhancement (SE) computation and its impact on the neutralino relic density in a general MSSM setup. It is publicly available for download from `http://people.sissa.it/~hryczuk`. The physics of DarkSE was presented in Refs. [2, 3]. Numerical algorithms used to solve the differential equations are based on "Numerical Recipies" [5].

If you use DarkSE please consider the original physics work behind and give credit to [3]. Please remember also to give credit to original DarkSUSY code [1] and possibly other works specified in the DarkSUSY manual.

2 Installation

Prerequisite for installing DarkSE is an installed DarkSUSY program[1]. Installation is very easy and consist only in copying several files and doing small changes in makefiles. For convenience we provide a simple bash script for doing this job (on Linux systems). To install:

1. Unpack the downloaded "darkse-v1.02.tar.gz" to the main DarkSUSY folder.

2. Go to the folder "DarkSE-v1.02"

3. Run[2]: *sh install.sh* (or for Mac OS: *sh install-macos.sh*)

The script asks you three questions about modifying the original DarkSUSY files: several very minor changes are needed to compile DarkSE. They consist of defining few variables in the common blocks and introducing a mass radiative correction to M_2 parameter in MSSM-7. Making these modifications do not alter the DarkSUSY functionality. There is also third optional change, which introduces a running of *g2weak*, which is absent in original version of DarkSUSY-5.0.5 (note: maybe this will change in next

[1] It is publicly available for download from `http://www.physto.se/~edsjo/darksusy`

[2] Change of file privileges by *chmod* might be needed.

releases!). All modified files are previously saved in the same place with ".backup" extension. Those files are also essential for later automatic uninstallation, which uses them to go back to original version. To uninstall just type in the DarkSE_v-1.1 folder *sh uninstall.sh.*

Troubleshooting: If after running the install script the code won't compile, try do the following. In the *src/se* directory copy the *makefile.in* to *makefile* and change the FF and FOPT variables to the ones used in any other of makefile's in your DarkSUSY installation.

In case the provided scripts fails to install and the solution described above won't work (for whatever reason), below we give a description how to install manually.

- the folder *se* copy to *./src/*
- the file *dssecom.h* copy to *./include/*
- files in *others*: *dsasdwdcossfsf.f* and *dschasct.f, dsmodelsetup.f* copy to folders *./src/as* and *./src/su* respectively (replace the original ones)
- in the *./src/se* folder make a new "makefile" (use makefile from other *src* sub-folder as a template)
- in the *./src/makefile.in* add to the list of folders the *se* one and below to the OBJWDIR variable add *se/dsse.o*

Done. Now try to compile DarkSUSY. In case of compilation problems with *libdarksusy.a* try to delete it and run *make* once again.

In Mac OS there may be problems with compiling DarkSUSY itself, caused by the FeynHiggs 2.6.5.1 package. To solve this it is enough to link to the older version:

```
cd contrib
rm feynhiggs
ln -s FeynHiggs-2.6.5 feynhiggs
```

3 Usage

Together with the package in the downloaded archive the User will find a file called "dsmain_template.f". It is an example main program showing how to use DarkSE. It is based on the example programs of DarkSUSY code. It does not explain DarkSUSY structure and routines, but only concentrates on the DarkSE part. For a general DarkSUSY description the reader is referred to its own manual.

Every subroutine or function has its header which gives some short description of what it does. Some of the files contain routines which are adopted version of the DarkSUSY ones, with typically only modified callings or added new variables to common blocks. Description of those can be found in the DarkSUSY manual.

After successful installation the code should compile with the "dsmain_template.f" file as a main program.

Apart from definitions of variables needed, the usage of DarkSE boils down to one call of the main subroutine *sommerfeld.* It then calls other routines, for given model: determining which channels can be Sommerfeld enhanced, write their particle IDs on screen (commented by default), computes and tabulates the Sommerfeld factors, computes the present day cross sections of neutralino to W^+W^-, $Z\gamma$ and $\gamma\gamma$ and writes them on screen (test version!). Next it calls the computation of relic densities with and without the SE (the case without is only to see the effect, can be safely commented). By default only relic density with co-annihilations is considered, but this can be safely modified by the User. Finally it computes the effective annihilation rate $W_{\rm eff}$ and thermally averaged cross section $\langle\sigma_{\rm eff}v\rangle$ and saves them to files.

If one wants to inspect the Sommerfeld factors, the subroutine *sewrite* can be used, which saves to files tabulated factors together with its interpolations for a given value of p and T (for details see the description in the routine). It is enough to uncomment it in *sommerfeld.f.*

4 Description of parameters

Standard usage consist only of choosing the physical model and values of its parameters[3]. The DarkSUSY together with the help of DarkSE will do the rest and give you the results. However, there may be cases in which one might want to change some internal parameters used in the package, for instance to get better accuracy (sacrificing some speed) or enlarge the regions in which Sommerfeld effect is computed and/or taken into account for the relic density calculations. In this section we will give a description of several internal parameters that can be easily tuned to modify the work of the code.

MPHIMAX In *sechannel.f.* Maximal mass of the interaction boson to be considered in the Sommerfeld effect computation. By default set to 200 GeV. This value takes into account all gauge bosons and typically lighter Higgs.

EPS In *sechannel.f* and *sommerfeld.f.* Algorithm determining if a given particle pair can have enhanced/suppressed cross section, is based on comparing their coupling strengths and mass (see next two parameters). The EPS parameter determines the minimal value of this coupling, i.e. for a given particle pair and interaction boson the code checks both couplings and both have to be larger than EPS to accept this channel as a candidate. The default value is 10^{-3}.

DMMAX In *sechannel.f.* In the case of multi-channel Sommerfeld effect computations, we set a maximal value of the mass splitting between the intermediate particle and the neutralino. This value cannot be large, i.e. it has to be much smaller than the neutralino mass, since the computation method is not valid if this is not the case. The default value for DMMAX is 5 GeV.

COANN In *sommerfeld.f.* It may happen that some heavy particles in the spectrum also satisfy conditions for the Sommerfeld effect computation. This however has no impact on the neutralino relic density if their masses are too high for co-annihilation to be relevant. The COANN parameter sets the maximal value of $\Delta m/m_\chi$ where Δm is the mass difference of the given particle and the neutralino. By default it is set to be 0.25.

IMAX In *sesigmaeff.f.* Number of points for which the $W_{\rm eff}$ and $\langle \sigma_{\rm eff} v \rangle$ are saved.

VMIN In *dssecom.h.* Cut-off on the velocity, i.e. the Sommerfeld enhancement is not computed if the velocity is smaller than VMIN. This is mainly set for speed (since solving Schrödinger equations becomes very slow for very small v), but it has also two theoretical reasons. Firstly, this region does not have an impact on the relic density. Secondly, in this regime the bound state physics may have to be considered for reliable predictions, so the Sommerfeld effect is most probably not reliable. By default set to 10^{-3}.

NSEmax In *dssecom.h.* Maximal number of different particles that may be included in the SE computation. By default set to 8, can be freely changed (it affects the size of tables).

NTmax and NPmax In *dssecom.h.* How many points for tabulation of SE in temperature (NTmax) and momentum (NPmax). This should be adjusted for a specific case. The more points the slower the computation, but typically more accurate interpolation. As a general guideline one can look on how many channels are present for a given model, and in particular if there are cases with $fchn > 1$ (i.e. more than one Schrödinger equation). If not, the default values of 20 and 30, respectively, should be sufficient. Otherwise, values of at least 30 and 100 are recommended.

Others If one wants to change the range on which the Boltzmann equation is solved, changes have to be applied in two places: in *setab.f* the Tmin parameter (minimal temprature for the SE) has to be adjusted, and in *sedsrdeqn.f* x1 has to be changed from xfinal to the Users choice (consistent with Tmin!).

[3] Like in DarkSUSY also here all parameters are or dimensionless or are expressed in GeV.

5 Known issues

The current version of DarkSE is able to compute the Sommerfeld corrections if up to three different two-particle states can enter into the ladder diagram (again, for explanations see [2]). Although it is easily extendable to other cases (though, it needs some coding), it is not of a great interest for possible physical applications and the stability of the solution drops down quite fastly. The point is that every new two-particle state adds another Schrödinger type equation, which boils down to adding four more real 1st order differential equations. This makes the solution of the Sommerfeld factors much less reliable.

In fact some problems can be observed already at the level of currently implemented states. Especially in the case of 3 coupled Schrödinger equations, for some parameters of the model the solution may become unstable. This is typically seen from the result, and can be inspected by looking onto the tabulated values of effective annihilation rate $W_{\rm eff}$ and maybe the Sommerfeld factors themselves.[4]

However, the current version of the code won't give any warning for most of these kind of situations. This is planned to be improved with possible future releases.

6 Change log

v1.02

- chenged the Sommerfeld effect computation in the Wino case from the cross-section to the amplitude level as described in [4]
- corrected the thermal corrections issue, where "dmrad" parameter was not restored to 0.17 GeV after *thermcor.f*
- added saving the full solution of the Boltzmann equation in *sommerfeld.f*
- corrected the compilation issue on Mac OS X
- moved all saved files to "data" folder
- other minor corrections

7 Acknowledgments

The heart of the code are the routines determining the Sommerfeld factors. Theoretical method for doing this computation was developed by Roberto Iengo in Ref. [6]. Subsequently, it was extended in our work with Piero Ullio and Roberto Iengo.

I would like also to thank Piero Ullio for help with DarkSUSY, discussions and encouragements in writing the code.

References

[1] P. Gondolo, J. Edsjo, P. Ullio, L. Bergstrom, M. Schelke and E. A. Baltz, JCAP **0407** (2004) 008 [arXiv:astro-ph/0406204].

[2] A. Hryczuk, R. Iengo and P. Ullio, JHEP **1103** (2011) 069 [arXiv:1010.2172 [hep-ph]].

[3] A. Hryczuk, Phys. Lett. B **699** (2011) 271 [arXiv:1102.4295 [hep-ph]].

[4] A. Hryczuk and R. Iengo, JHEP **1201** (2012) 163 [arXiv:1111.2916 [hep-ph]].

[5] W.H. Press, B. Flannery, A. Teukolsky, W. Vetterling, *Numerical recipes in FORTRAN 77: The Art of Scientific Computing*, Cambridge University Press 1992

[6] R. Iengo, "Sommerfeld enhancement: general results from field theory diagrams," JHEP **0905** (2009) 024 [arXiv:0902.0688 [hep-ph]].

[4] There is also a possibility of checking the Schrödinger equations solutions. An example is present in *seschreqns.f* at the very end of the *seschreqns* subroutine, though commented by default.