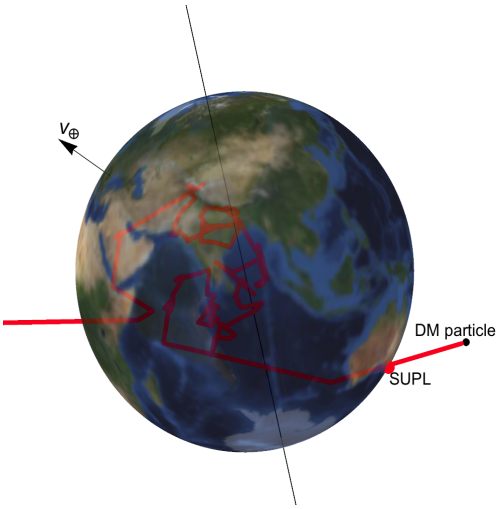
DaMaSCUS Documentation

Release 1.0.2

Timon Emken

Mar 25, 2020

Contents:



Authors: Timon Emken & Chris Kouvaris

- DaMaSCUS is a MC simulator of dark matter particles as they move through the Earth and scatter on terrestrial nuclei.
- It allows to compute the local distortions of the DM density and velocity distribution caused by collisions with nuclei.
- The distorted distribution functions and redistributed densities are used to give precise estimates of time-dependent signal rates for direct detection experiments and diurnal modulations.
- A full, realistic model of the Earth is implemented as well as the Earth's time-dependent velocity and orientation in the galactic frame.
- DaMaSCUS is written in C++ and fully parallelized (MPI).

For the underlying physics check out the paper . For the code visit the repository .

CHAPTER 1

Getting started

1.1 Requirements

1.1.1 Dependencies

These are the dependencies of DaMaSCUS:

- **libconfig**: To handle the input configuration files we use the textit{libconfig} library.

  http://www.hyperrealm.com/libconfig/

- **Eigen**: DaMaSCUS relies heavily on this linear algebra C++ library.

  http://eigen.tuxfamily.org/

- **openMPI**: For the parallelization we implemented our code using the open *Message Passing Interface*.

  https://www.open-mpi.org

1.2 Download

The DaMaSCUS code is available at

https://github.com/temken/DaMaSCUS/ .

To download it via git simply run

```
$ git clone https://github.com/temken/DaMaSCUS/
```

in your console or terminal.

1.2.1 Folder Structure

You will now find the following folders in your destination directory:

- **/bin/**: After successful compilation this folder contains two executables as well as the configuration file.
- **/build/**: This folder contains all object files. Both the object files and the executables in **/bin/** are deleted via

```
$ make clean
```

- **/data/**: Once a simulation run is performed, the generated data will be stored here
- **/include/**: The DaMaSCUS header files are stored here. Necessary 3rd party libraries can also be placed here.
- **/plots/**: To visualize the results, created by the analysis module we include the small Mathematica package *DaMaSCUStoolbox* and an example notebook creating and saving plots.
- **/results/**: The analysis module saves its results and histograms here.
- **/src/**: All the source code files of the two DaMaSCUS modules can be found here.

1.3 Installation

DaMaSCUS consists of two more ore less independent modules:

1. **DaMaSCUS-Simulator**: Simulates the dark matter trajectories and genererates the raw data.
2. **DaMaSCUS-Analyzer**: Analyzes the raw data and calculates e.g. velocity histograms or detection rates.

The code is compiled using the Makefile. You might have to adjust the first lines

```
#Compiler and compiler flags
CXX := mpic++
CXXFLAGS := -Wall -std=c++11
LIB := -lconfig++
INC := -I include
(...)
```

to your local settings. Next to install DaMaSCUS and compile the code simply run

```
$ make
```

from the root directory in your terminal. Alternatively you can also run

```
$ make simulator
```

or

```
$ make analyzer
```

to just compile one of the modules.

Finally, running

```
$ make clean
```

deletes all object files and executables.

CHAPTER 2

Using DaMaSCUS

2.1 Work Flow

To simulate dark matter trajectories and analyze the generated data DaMaSCUS has a clear work-flow.

1. Adjust your input parameter (such as the dark matter mass) inside the configuration file and asign a simulation ID to identify this simulation run.
2. Run the simulation module to generate the raw data.
3. Run the analysis module to process the data.
4. The results can be plotted e.g. with the included Mathematica notebook.

2.1.1 1. Configuration File

You can find an example configuration file at

```
/bin/DaMaSCUS.cfg
```

Here you adjust all the input parameter for the next DaMaSCUS run. We go through it block by block.

```
//DaMaSCUS Configuration File

//Simulation input parameter
                simID           =       "exampleID";    //MC Simulation ID
                initialruns     =       100000000L;     //Number of particles in the␣
↪initial MC run
                samplesize      =       10000;          //velocity sample size per␣
↪isodetection ring
                vcutoff         =       1.0;            //velocity cutoff in cm/sec
                rings           =       36;                             //number of␣
↪isodetection rings
```

First you assign the simulation run a unique identifying ID. You also decide the number of particles you want to simulate in the initial run without scatterings (initialruns) and how many data points you need in each isodetection ring (samplesize). The velocity cut-off, below which a trajectory simulation is aborted.

New in version 1.1: The number of isodetection rings is now flexible and can be set in the configuration file (rings).

Warning: The "L" behind the value for initialruns is necessary to denote that it is a **long int**.

```
//Simulation Time:
                date            =       [15,02,2016];   //Date [dd,mm,yyyy]
                time            =       [0,0,0];        //Universal time [h,m,s]
```

Next you fix the simulation date and time, which is mostly used to determine the Earth's velocity in the galactic frame.

```
//Dark Matter Data
        //Particle data
                mass            =       500.0;          //in MeV
                sigma           =       1.0;            //in pb
                formfactor      =       "None";         //Options: "None",
↪"HelmApproximation"
        //DM Halo
                halomodel       =       "SHM";          //Options: Standard Halo␣
↪Model "SHM",...
                rho                     =       0.3;    //DM halo energy density in␣
↪GeV/cm^3
```

These are the essential input parameter. Here you fix the DM mass, its cross-section with nucleons and whether or not to use a Helm form factor. For light DM it is recommended to deactivate it, since this speeds up the simulation considerably.

So far only the standard halo model is implemented, for which we can set the DM energy density.

```
//Detector depth:
                depth           =       1400.0;         //in meter
```

The depth of the isodetection rings, i.e. the underground depth of the detector of interest is determined here.

```
//Analysis parameter
                experiment      =       "CRESST-II";    //Options: "LUX" for heavy DM,
↪"CRESST-II" for light DM and "None"
```

And finally we decide the type of data analysis: 1. Set "LUX" for a LUX-type detector. Use this option for heavy dark matter. 2. Use "CRESST-II" for a CRESST-type detector, which is sensitive to DM masses down to 500 MeV. 3. Set "None", if you are e.g. just interested in the resulting speed distributions across the globe.

Warning: Note that the configuration file can be sensitive to the input parameter type. For example it might complain if an input parameter of type **double** is given as "1" instead of "1.0".

2.1.2 2. Running the simulation

After setting the input parameter and assigning a unique simulation ID we can start the MC simulation from the **/bin/** directory. To start run

```
$ mpirun -n N DaMaSCUS-Simulator input.cfg
```

where *N* is the number of MPI processes and *input.cfg* is your configuration file.

After a successful run your terminal should show something like

```
$ mpirun -n 4 DaMaSCUS-Simulator DaMaSCUS.cfg

##############################
DaMaSCUSv1.0 - Simulation

Starting Time: Wed Aug 16 11:34:08 2017
Simulation ID: exampleID

Creating logfile.
Creating folder for velocity data.
Start initial MC simulation run without DM scatterings.
        Initial run finished    (1 s).

Start main MC simulation run with scatterings.
        Main MC run finished    (4 s).

Processing Time:        5.89347s (0:0:5:893).
##############################
```

A copy of the used configuration file is stored in the **/data/** directory together with the raw data. In addition a logfile, which documents important input and output parameter is created in the **/results/** folder.

2.1.3 3. Analyze the data

Next we can analyze the generated data by running

```
$ mpirun -n N DaMaSCUS-Analyzer SimulationID
```

in your terminal from the **/bin/** directory. Again *N* is the number of MPI processes. The analysis type is set inside the config file **/data/SimulationID.cfg** and can be adjusted after the simulation has finished. The terminal output of a successful analysis looks like

```
$ mpirun -n 4 DaMaSCUS-Analyzer exampleID

##############################
DaMaSCUSv1.0 - Data Analysis

Starting Time:  Wed Aug 16 11:49:05 2017
Simulation ID:  exampleID
Experiment:     None

Creating folder for histograms.
Done.

Creating temporary files.
Reading in local DM densities.
Broadcast local DM densities to all MPI processes.
Start data analysis...

MPI rank        Isodetection ring       Local Progress  Computing time[s]
↪Residual time estimate[s]
```

(continues on next page)

(continued from previous page)

```
(*some infos about the progress*)

Data analysis complete.
Creating ASCII output.
Delete temporary files and finish.

Processing Time:        1.02264s (0:0:1:22).
##############################
```

2.1.4 4. Plot the results

After the both modules have finished their computations you can use the included Mathematica notebook **/plots/plots.nb** to create and save plots of e.g. the speed histograms or the event rate modulation.

CHAPTER 3

Citing DaMaSCUS

If you decide to use the textsc{DaMaSCUS} code, please cite

> Emken, T., Kouvaris, C.: DaMaSCUS,(2017), Astrophysics Source Code Library, record [ascl:1706.003] .

as well as the original paper,

> Emken, T., Kouvaris, C.: DaMaSCUS: The Impact of Underground Scatterings on Direct Detection of Light Dark Matter, [JCAP 1710 (2017) no.10, 031] , [arXiv:1706.02249] .

CHAPTER 4

Release History

The code is under continuous development and will be extended and updated over time.

- 25.03.2020: Release of version 1.1.
  - Number of isodetection rings is now a user input.
- 16.08.2017: Release of version 1.0.2.
  - Complete restructuring of the DaMaSCUS directories to a more clear and standard format.
  - New documentation page to replace the pdf manual.
- 15.06.2017: Release of v1.0.1.
  - major bug fix concerning mostly very high cross-sections.
- 06.06.2017: Release of version 1.0.

CHAPTER 5

License

CHAPTER 6

Contact & Support

For questions, bug reports or other suggestions please contact

Timon Emken (emken@chalmers.se)

CHAPTER 7

Visualization

7.1 Simulation of a single DM particle

7.2 Simulation of a multiple particles

7.3 Evolution of Isodetection Rings

CHAPTER 8

Contact & Support

For questions, bug reports or other suggestions please contact

Timon Emken (emken@chalmers.se)