Helios-r2 - A Bayesian Nested-Sampling Retrieval Code

Code version 1.0
Software guide version 1.0

Contents

1 Introduction to Helios-r2 **4**
1.1 Overview . . . 4
1.2 Developers . . . 4
1.3 Licence . . . 4
1.4 About this guide . . . 5

I Helios-r2: General Guide **6**

2 Installation **7**
2.1 Prerequisites . . . 7
2.2 Compile MultiNest . . . 7
2.3 Configure Helios-r2 with CMake . . . 7

3 Running Helios-r2 **9**
3.1 Starting Helios-r2 . . . 9
3.2 Main configuration files . . . 9
3.2.1 Main retrieval parameter file . . . 10
3.2.2 Observation file . . . 12
3.2.3 Additional configuration and parameter files . . . 12

4 Observational data **13**
4.1 Supported observational types . . . 13
4.1.1 Spectroscopy . . . 13
4.1.2 Band-spectroscopy . . . 13
4.1.3 Photometry . . . 15
4.2 Input file structure . . . 15
4.2.1 Spectroscopy . . . 15
4.2.2 Band spectroscopy . . . 16
4.2.3 Photometry . . . 17

5 Opacity data **18**
5.1 Required format of the tabulated data . . . 18
5.2 Available pre-tabulated data . . . 20

6 Output files **21**

7 BrownDwarfSim example **22**

II Forward Models 23

8 General information 24
8.1 List of available forward models 24

9 The Brown Dwarf forward model 25
9.1 Overview 25
9.2 General parameter file 25
9.3 Prior configuration file 27
9.4 Postprocess output files 29

III Helios-r2: Detailed Code Descriptions 30

Bibliography 32

1 Introduction to Helios-r2

1.1 Overview

Helios-r2 is an open source model that can perform atmospheric retrieval of brown dwarf and exoplanet spectra. It has been introduced and described in Kitzmann et al. (2019) and is the successor to the original Helios-r code described by Lavie et al. (2017). This original version, however, has never been publicly released. The new version has been completely written from scratch in C++/CUDA and includes various improvements over the original version. The Helios-r2 model is part of the Exoclimes Simulation Platform (ESP) (https://github.com/exoclime) that provides a variety of different codes related to exoplanet science and general atmospheric modelling.

Helios-r2 uses a Bayesian statistics approach by employing a nested sampling method to generate posterior distributions and calculate the Bayesian evidence. The nested sampling itself is done by the MultiNest library (https://github.com/farhanferoz/MultiNest). The computationally most demanding parts of the model have been written in NVIDIA's CUDA language for an increase in computational speed. Helios-r2 can work on both, pure CPU as well as hybrid CPU/GPU setups. Running it purely on a CPU is not recommended, though, as the runtimes can be by a factor of 10 or 100 longer compared to running it on a GPU.

Successful applications include retrieval of brown dwarf emission spectra (Kitzmann et al., 2019) and secondary eclipse measurements of exoplanets (Bourrier et al., 2019).

1.2 Developers

The current version of Helios-r2 has been developed and written by

Daniel Kitzmann (daniel.kitzmann@csh.unibe.ch)

The opacity data is kindly provided by

Simon Grimm (simon.grimm@csh.unibe.ch)

using the HELIOS-k code of the ESP (Grimm & Heng (2015), https://github.com/exoclime/HELIOS-K).

1.3 Licence

Helios-r2 has been released under the GNU Public Licence (GPL) 3.0. That means, it can be freely copied, edited, and re-distributed. If the code is re-distributed it has to be released under at least a GPL 3.0 licence as well. The full licence of Helios-r2 can be found in the repository (LICENSE file) or under https://www.gnu.org/licenses/gpl-3.0.html.

The `Helios-r2` repository also contains two additional open source codes, the radiative transfer code `CDISORT` (Stamnes et al., 1988; Hamre et al., 2013) and the equilibrium chemistry code `FastChem` (version 1.0), written by Daniel Kitzmann and Joachim Stock (Stock et al., 2018) and available as part of the ESP: `https://github.com/exoclime/FastChem`. Both codes have also been licensed under the GPL 3.0 and are, under that licence, being re-distributed within the `Helios-r2` repository.

1.4 About this guide

This guide provides the basic information on the `Helios-r2` code and how to use it. It is structured into three different parts. For those who are interested in just running `Helios-r2` with an already existing forward model, reading of Part I & II should be sufficient.

- Part I contains the basic descriptions on how to compile and start `Helios-r2`. It also describes the structures of the basic input and output files, the different observation types, as well as the required format of the absorption cross-sections.
- Part II describes the forward models included in `Helios-r2`.
- Part III is a more in-depth description of the code itself. It provides information on how the code is structured internally and shows how to create new forward models and sub-modules to extend the capabilities of `Helios-r2`. Part III will also provide detailed descriptions of all the existing modules that can be used to construct new forward models. *This section has not been written yet and will be added in a future version of this guide.*

Part I

Helios-r2: General Guide

2 Installation

2.1 Prerequisites

Helios-r2 has been written in C++ and CUDA. It uses some of the features of the C++11 standard and, therefore, requires a compiler that implements this standard. It also needs additional compilers and libraries to compile and run. The complete list of prerequisites is:

- the G++ compiler, at least version 5.0
- the GCC compiler, at least version 5.0 (for CDISORT)
- CMake, at least version 3.10
- a compiled version of the MultiNest library
- the GFORTRAN compiler (required by MultiNest)
- the LAPACK library (required by MultiNest)
- the NVIDIA CUDA framework, including the NVCC compiler
- the Boost library headers

2.2 Compile MultiNest

Before you compile and install Helios-r2 itself, you fist need to create a static library of the MultiNest library. MultiNest is an open source code, available from https://github.com/farhanferoz/MultiNest. Download and follow the installation instructions from the MultiNest repository.

Helios-r2 needs two things from the MultiNest library: the compiled, static library libmultinest.a[1] and the MultiNest header file multinest.h[2]. Copy **both** files into the multinest folder located within the Helios-r2 source folder.

2.3 Configure Helios-r2 with CMake

Before Helios-r2 can be compiled, CMake is required to configure the compilation files, locate libraries, and write the makefiles that will perform the actual compilations. If required libraries are missing, CMake will report a corresponding error message. In this case, you need to install the missing libraries or compilers.

[1] By default, the file is located in the multinest_source_folder/lib folder.
[2] By default, the file can be found in multinest_source_folder/include.

To run the CMake configuration, first create the build folder inside the Helios-r2 source code folder, e.g. via terminal:

```
mkdir build
```

Switch to the folder and run CMake[3]:

```
cd build
cmake ..
```

After CMake successfully configured the compilation files, compile Helios-r2 by running:

```
make all
```

Upon successful compilation, the executable helios-r should be present in the main Helios-r2 folder. Congratulations, Helios-r2 is now ready to be run!

Note, if you want to move the Helios-r2 installation to a different computer it might be necessary to recompile both, MultiNest and Helios-r2. This is usually the case if the CUDA environment and system libraries have different versions than the ones the code has been originally compiled against. Sometimes an upgrade of the CUDA framework requires a recompilation of Helios-r2 as well. Otherwise, running Helios-r2 will produce a CUDA runtime error.

[3] don't forget the .. after the cmake command

3 Running Helios-r2

Helios-r2 expects to find all required configuration files in a separate retrieval folder. The same folder will also be used for the output files of MultiNest (posterior data, statistics of the nested sampling run) and Helios-r2 itself, which includes the posterior spectra or other postprocess data it produces.

The Helios-r2 repository contains an example of such a folder: BrownDwarfSim. This example contains all required observational data and configuration files to start Helios-r2. The example is further briefly discussed in Sect. 7.

3.1 Starting Helios-r2

Helios-r2 is started by calling the executable together with a command line argument that points to the folder containing the retrieval configuration files, e.g.:

```
./helios -r BrownDwarfSim/
```

Helios-r2 will quit with error messages if config files it expects are not found, the opacity data is not available, or if it finds errors in the configuration parameters.

After the nested sampling performed by MultiNest has been successfully completed, Helios-r2 will perform a postprocess procedure. Here, the posterior distributions from the MultiNest output will be used to calculate posterior spectra. Additionally, the specific forward model used by Helios-r2 might also have its own postprocessing methods. The Brown Dwarf forward model, for example, will calculate temperature profiles for all posterior distributions and determine effective temperatures.

It is also possible to start Helios-r2 just for the purpose of postprocessing an already converged model. This is done by adding post to the command line when starting Helios-r2, i.e.

```
./helios -r BrownDwarfSim/ post
```

would start the postprocess of the posterior data located in the folder BrownDwarfSim.

3.2 Main configuration files

Within the retrieval folder, Helios-r2 expects to find at least two config files: retrieval.config and observations.list. The former contains the general options for running Helios-r2 and the most important MultiNest options, the latter the list of observations that Helios-r2 will try to perform the retrieval for.

3.2.1 Main retrieval parameter file

The `retrieval.config` has the following structure:

```
#################
#General config#
#################
#Use GPU
Y

#OpenMP processor number (if 0, use maximum)
28

#Spectral resolution (cm-1)
1.0

#Opacity data folder
/media/data/opacity/helios/molecules

#Use error inflation prior
Y

######################
#Multinest parameter#
######################
#Importance nested sampling
Y

#Mode separation
N

#Number of live points
4000

#Efficiency
0.8

#Maximum number of iterations (0 for no limit)
0

#Resume
N

#Console feedback
Y

#Print parameter values and likelihoods
Y
```

The file consists of the following, general model options

- **Use GPU**
  Determines if the GPU is used. If set to `N`, the model will run just on the CPU.

- **OpenMP processor number**
  The number of CPU cores the code will use. The code will use several CPU cores to run

specific parts of the code in parallel. Use `0` if you want to use all available cores.

- **Spectral resolution**
  The constant step size in wavenumbers that `Helios-r2` will use for the high-resolution spectra it calculates. This value should be higher than the resolution of the observational data. Note that `Helios-r2` will always use the tabulated wavenumber points of the opacity data. Thus, this step size is not guaranteed to be constant over the entire spectral range. `Helios-r2` will always pick the closest available wavenumber point of the tabulated data to avoid interpolation in wavenumber space.

- **Opacity data folder**
  The root folder where the opacity data is stored.

- **Use error inflation prior**
  Determines whether the error inflation parameter is used during the calculation of the likelihood. See Kitzmann et al. (2019) for details.

Below the general options, the ones for `MultiNest` can be found. For more details regarding these parameters please also consult the `MultiNest` documentation and the corresponding published studies by Feroz & Hobson (2008) and Feroz et al. (2009).

- **Importance nested samling**
  Determines if `MultiNest` uses the important nested sampling approach. This requires a bit more memory but, on the other hand, also increases the convergence speed and the overall accuracy of the Bayesian evidence calculation. Unless memory is a real bottleneck, there should be no reason to deactivate important nested sampling.

- **Number of live points**
  Sets the number of live points uses by `MultiNest`.

- **Efficiency**
  Sets the efficiency that determines the way `MultiNest` draws new points from the parameter space. For more details on this parameter check the `MultiNest` documentation. The authors of `MultiNest` suggest to use an efficiency of 0.8 for parameter estimations and 0.3 when the Bayesian evidence is wanted at a high accuracy.

- **Maximum number of iterations**
  The maximum number of iterations `MultiNest` will use before the nested sampling is stopped. A value of 0 indicates that `MultiNest` will perform the nested sampling until its convergence criteria are met.

- **Resume**
  If this is set to `Y`, `MultiNest` will try to resume a previously started retrieval run. The

files `MultiNest` needs to restart the nested sampling must all be present in the retrieval folder. This option is usefull if `Helios-r2` is run on a cluster with a strict time limit. A restart can also be used if a previous `MultiNest` run was stopped at its maximum number of iterations.

- **Console feedback**
  `MultiNest` will regularly report the current total number of model evaluations and estimates for the Bayesian evidences when this option is turned on.

- **Print parameter values and likelihood**
  Determines whether the parameter values and the computed likelihood values for all models should be displayed. If `Helios-r2` is run on a cluster and the terminal output is redirected to a file, it is usually a good idea to deactivate this option. Otherwise, the output file could become quite large.

`MultiNest` has also additional parameters that are currently not accessible through the config file. They can be found in the file `retrieval/multinest_parameter.h` but require recompilation when changed. The parameter class already contains a place holder to read in an additional `MultiNest` parameters file that will be implemented in the future.

3.2.2 Observation file

The second file that `Helios-r2` expects to find in the retrieval data folder is the `observations.list` that contain the file paths to the different observations used in the retrieval. There has to be at least one observation present in the file. An example for a retrieval with multiple observations of the same object is shown below.

```
spectrum_wasp121b_tess.dat
spectrum_wasp121b_trappist.dat
spectrum_wasp121b_wfc3.dat
spectrum_wasp121b_2mass.dat
spectrum_wasp121b_spitzer.dat
```

The different observations don't have to be in any particular order and can also have overlapping wavelength ranges. Information on the format of the observational data files and the supported observation types can be found in Sect. 4.

3.2.3 Additional configuration and parameter files

Depending on the chosen forward model, additional configuration files might be required. The Brown Dwarf forward model, for example, also need its own model config, as well as a parameter list for the priors it uses. More detailed information on the forward model config files can be found in Part II for all forward models available in `Helios-r2`.

4 Observational data

4.1 Supported observational types

`Helios-r2` supports three different types of basic observational types: spectroscopy, band spectroscopy, and photometry. Bases on the type of observation, the required format of the data files differ slightly. In the following sections, the three basic types and their required input formats are described.

4.1.1 Spectroscopy

In spectroscopy mode, an observational spectrum is given at specific wavelengths λ_i, from λ_1 to λ_N. The upper panel of Fig. 4.1 shows this conceptually. `Helios-r2` itself operates in wavenumber space ν. It sets up a high-resolution wavenumber grid, with a step size determined by the corresponding option in the `retrieval.config` file. This grid is symbolised by the red lines in the lower part of Fig. 4.1.

To simulate the observed flux at the given wavelengths, `Helios-r2` creates a structure composed of spectral bands, one band for each observational wavelength. The boundaries of these bands in wavenumber space are halfway between adjacent wavelengths. For example, the boundaries $\nu_{i,1}$ and $\nu_{i,2}$ for the i-th band, corresponding to the wavelength λ_i, are determined by the adjacent wavelengths λ_{i-1} and λ_{i+1}.

`Helios-r2` calculates its model spectrum on the high-resolution wavenumber grid (the red lines). It then obtains the mean flux in each of the bands i via integration and identifies the result with the flux at the observational wavelengths λ_i. Optionally, before integrating the spectrum, the high-resolution spectrum can be convolved with a given instrument line profile to simulate the flux received by the detector (see Kitzmann et al. (2019)).

4.1.2 Band-spectroscopy

Band-spectroscopy is a degraded form of spectroscopy, where individual wavelengths have been summed up into bands to e.g. increase the signal-to-noise of a low-signal observation. This is, for example, commonly done for exoplanet observations with the WFC3 instrument on the Hubble Space Telescope. The band structure itself does not need to be regular.

As depicted in the middle panel of Fig. 4.1, the observational data is assumed to consist of $i = 1 \cdots N$ spectral bands, each with given wavelength boundaries $\lambda_{i,1}$ and $\lambda_{i,2}$. `Helios-r2` will create the same band structure in wavenumber space. Just like for spectroscopy calculations, the high-resolution spectrum of `Helios-r2` will be integrated over each band i to obtain the mean flux of the corresponding observation. Optionally, before integrating the spectrum, the high-resolution spectrum can be convolved with a given instrument line profile to simulate the flux received by the detector.

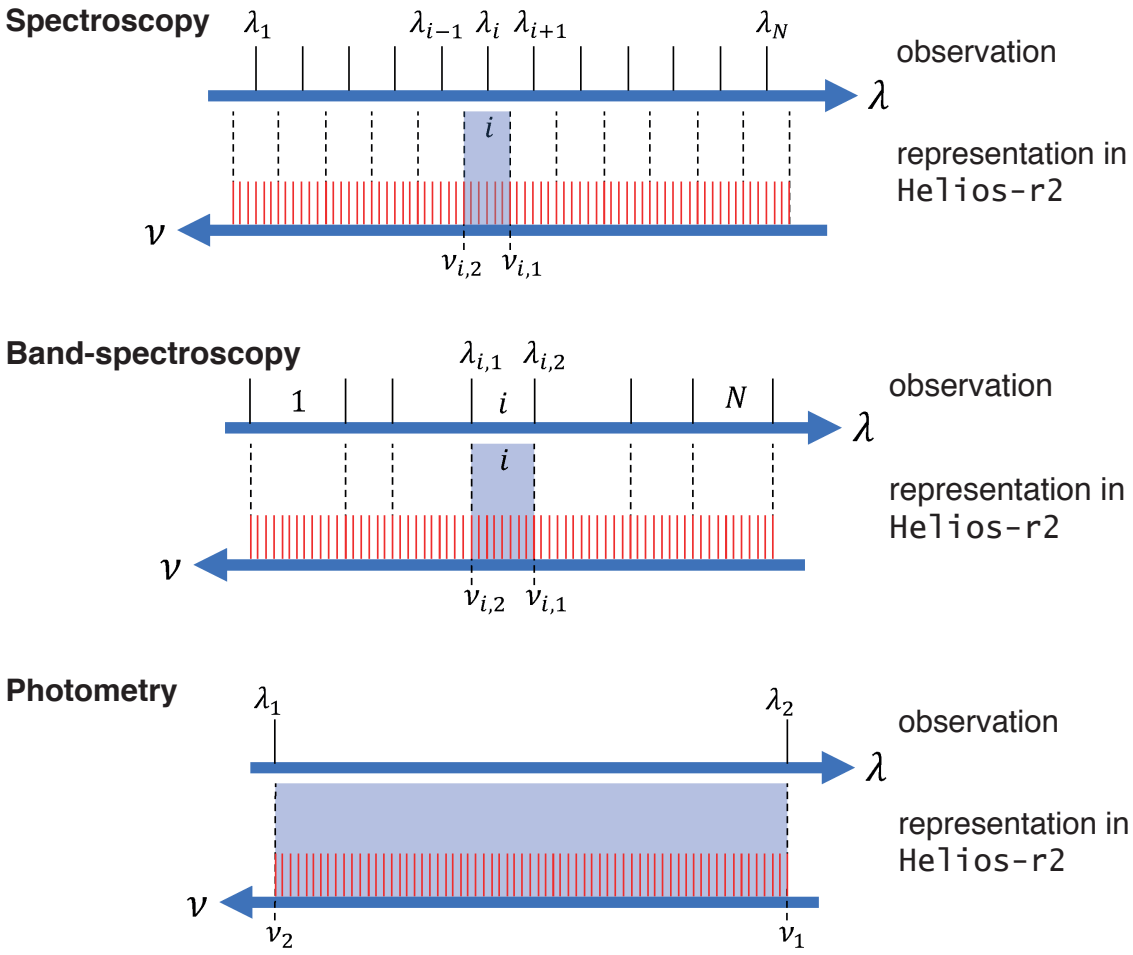

Figure 4.1: The three observational types supported by `Helios-r2`. The upper panels in each figure show the observations in wavelength space, with vertical, black lines denoting specific wavelengths. The lower panels depict the representation within `Helios-r2` in wavenumber space. The vertical, red lines correspond to the high-resolution grid that is used by `Helios-r2`, the dashed-lines resemble the band structure it employs to simulate the observations.

4.1.3 Photometry

Photometry is essentially band-spectroscopy with just one broad band between two wavelengths λ_1 and λ_2 (see Fig. 4.1). The high-resolution spectrum calculated by `Helios-r2` will be integrated over the bandpass in wavenumber space to obtain the mean flux in the filter.

The conceptual difference between band-spectroscopy and photometry within `Helios-r2` is that unlike the former, a photometry observation does not have an instrument line profile because it's supposed to cover a broader wavelength range. Instead, it can be processed through a filter transmission function to simulate the observation through a specific filter. Note, however, that this is not yet fully implemented in `Helios-r2`. For now, the two boundary wavelengths λ_1 and λ_2 should be used as the effective filter limits. The ability to read in and use specific filter transmission curves will be added in the future.

4.2 Input file structure

The following subsections describe the structure of the input files for the different observational types. The observational types that are currently supported are: `spectroscopy`, `band-spectroscopy`, and `photometry`. If you use any other keyword for the observational type, `Helios-r2` will quit with an error message.

4.2.1 Spectroscopy

An input file for a spectroscopy observation has the following structure:

```
#observation description
#name
Generic Brown Dwarf

#type
spectroscopy

#wavelength (mu) flux (W m-2 mu-1) error
8.546470e-01    7.608414e-18    2.168767e-17
8.577970e-01    1.707642e-18    2.168767e-17
8.609660e-01    1.426162e-17    2.168767e-17
8.641550e-01    1.166071e-17    2.168767e-17
8.673640e-01    4.046967e-18    2.168767e-17
8.705910e-01    2.501110e-17    2.168767e-17
8.738380e-01    2.093669e-17    2.168767e-17
8.771040e-01    4.879053e-18    2.168767e-17
8.803890e-01    3.160485e-17    2.168767e-17
```

It consists of a *name*, that the observation will be referenced to in the code and output, followed by the *type* of the observation - in this case `spectroscopy`. The observational data is organised in columns, starting with the wavelength in μm, followed by the observed flux in units of $\mathrm{Wm}^{-2}\mu\mathrm{m}^{-1}$ and its absolute error in the same units.

As mentioned earlier, `Helios-r2` can also take an instrument line profile into account. If that is the case for a given observation, another column is added to the file, right after the flux error. This column should contain the FWHM of the profile in units of μm. An example for such a file would look like this:

```
#observation description
#name
GJ570D SpeX

#type
spectroscopy

#wavelength (mu) flux (W m-2 mu-1) error line profile fwhm (mu)
8.546470e-01     1.067250e-16     4.268400e-17     1.039500e-02
8.577970e-01     1.554700e-17     6.090700e-17     1.045770e-02
8.609660e-01     2.235860e-16     1.419440e-16     1.052370e-02
8.641550e-01     9.457600e-17     1.087830e-16     1.058970e-02
8.673640e-01     2.628640e-16     1.121910e-16     1.064910e-02
8.705910e-01     2.536920e-16     3.449900e-17     1.071510e-02
8.738380e-01     1.915240e-16     4.122700e-17     1.077780e-02
8.771040e-01     2.369350e-16     1.440910e-16     1.084050e-02
8.803890e-01     2.678520e-16     1.319020e-16     1.090320e-02
```

4.2.2 Band spectroscopy

An input file for a band-spectroscopy observation has the following structure:

```
#observation description
#name
WASP121b WFC3

#type
band-spectroscopy

#lower wavelength (mu) upper wavelength (mu)  flux (W m-2 mu-1)  error
1.122000e+00     1.141000e+00     1.949532e+06     1.714572e+05
1.141000e+00     1.159000e+00     1.666056e+06     1.926314e+05
1.159000e+00     1.178000e+00     1.904628e+06     1.772200e+05
1.178000e+00     1.196000e+00     1.807765e+06     1.689500e+05
1.196000e+00     1.215000e+00     2.096797e+06     1.424307e+05
1.215000e+00     1.234000e+00     2.099312e+06     1.362282e+05
1.234000e+00     1.252000e+00     1.860565e+06     1.459596e+05
1.252000e+00     1.271000e+00     1.594426e+06     1.273606e+05
1.271000e+00     1.289000e+00     1.724892e+06     1.207273e+05
1.289000e+00     1.308000e+00     1.451919e+06     1.322659e+05
```

It consists of a *name*, that the observation will be referenced to in the code and output, followed by the *type* of the observation - in this case `band-spectroscopy`. The observational data is organised in columns, starting with two columns that describe the edges of the wavelength bands in μm, followed by the band's flux in units of $\mathrm{Wm}^{-2}\mu\mathrm{m}^{-1}$ and its absolute error in the same units.

As mentioned earlier, `Helios-r2` can also take an instrument line profile into account. If that is the case for a given observation, another column is added to the file, right after the flux error. This column should contain the FWHM of the profile in units of μm.

4.2.3 Photometry

An input file for a photometry observation describes a single photometric measurement and has the following structure:

```
#observation description
#name
WASP121b Spitzer Photometry

#type
photometry

#lower wavelength (mu)   upper wavelength (mu)    flux (W m-2 mu-1)    error
3.129400e+00     3.964390e+00     1.769131e+05     5.473023e+03
```

It consists of a *name*, that the observation will be referenced to in the code and output, followed by the *type* of the observation - in this case `photometry`. The observational data is organised in columns, starting with two columns that describe the edges of the photometric band in μm, followed by the band's flux in units of $\mathrm{Wm}^{-2}\mu\mathrm{m}^{-1}$ and its absolute error in the same units. Note that `Helios-r2` will only read in one line of observational data.

Future versions of `Helios-r2` will also add the option to read in an optional filter transmission file here.

5 Opacity data

`Helios-r2` uses tabulated absorption cross-sections for its radiative transfer calculations. The cross-sections have to be given in units of cm^2.

You can of course use your own opacity data. If you choose to do that, you need either convert your data to the format described below or change the corresponding code within `Helios-r2`. More in-depth details about the implementation in the code can be found in Part III.

5.1 Required format of the tabulated data

Cross-sections are required to be tabulated as a function of wavenumber. All cross-sections have to be given on the same wavenumber grid. `Helios-r2` expects to find this global wavenumber grid in a special file. The location of the opacity data and the wavenumber file are handled by the specific forward model. The configuration of the forward models will usually have an option to specify specific folders where the data for the different species can be found by `Helios-r2`.

Global wavenumber grid

The first file that `Helios-r2` will look for is the global wavenumber grid in a file named `wavenumber_full.dat` that has to be located in the opacity folder set in the config file `retrieval.config`. The first, few lines of this file are shown below.

```
5000001
0.0000000000e+00
1.0000000000e-02
2.0000000000e-02
3.0000000000e-02
4.0000000000e-02
5.0000000000e-02
6.0000000000e-02
7.0000000000e-02
8.0000000000e-02
9.0000000000e-02
1.0000000000e-01
1.1000000000e-01
1.2000000000e-01
1.3000000000e-01
1.4000000000e-01
1.5000000000e-01
```

The file starts with an integer value that is equal to the total number of wavenumbers in the file. After that, the single wavenumbers are tabulated in ascending order. In the example above, the wavenumber grid has a constant step size of $0.01\,cm^{-1}$. This is not a requirement

for `Helios-r2`, the grid does not have to be equidistantly spaced.

One important thing to note is, that all cross-sections **have** to be tabulated at the wavenumbers from this file.

Structure of a single cross-section file

The absorption cross-sections are all saved in single files, one for each tabulated temperature-pressure point. Forward models will usually provide configuration options to select the opacity species and specify the folders where the corresponding data is stored. Within a specified folder for a given species, `Helios-r2` will first look for the file `filelist.dat` that lists the individual data files with the temperatures and pressures.

The general structure of this file is as follows:

```
1.00000e-08     5.00000e+01     H2O_00050_n800.bin
2.18776e-08     5.00000e+01     H2O_00050_n766.bin
4.67735e-08     5.00000e+01     H2O_00050_n733.bin
1.00000e-07     5.00000e+01     H2O_00050_n700.bin
2.18776e-07     5.00000e+01     H2O_00050_n666.bin
4.67735e-07     5.00000e+01     H2O_00050_n633.bin
1.00000e-06     5.00000e+01     H2O_00050_n600.bin
```

The first column is the pressure in bar, the second the temperature in K, and the third the corresponding file where the cross-sections for this p-T combination are stored. The list does not have to be ordered in any special way. The absorption cross-sections also don't need to be tabulated on a regular, square p-T grid, nor do all species have to use the same p-T grid. `Helios-r2` will not load all the cross-sections files at once. Instead, it will always only load the data it currently needs.

All individual cross-section files are saved in binary format to decrease the time it requires to read them in. The first entry in each file contains an `integer` value that describes the total number of entries in a specific file. All remaining entries are `float` values for the cross-sections in cm^2.

As mentioned earlier, the cross-sections have to be tabulated on the global wavenumber grid, i.e. the first cross-section (the one right after the integer value) corresponds to the first wavenumber in `wavenumber_full.dat`. However, not all chemical species have absorption lines available that cover the entire, global wavenumber range. Some, for example, might only absorb in the infrared but not in the UV. Therefore, the cross-sections don't need to be tabulated over the entire global wavenumber range. Instead, they may stop once no absorption lines are present any more. Internally, the missing cross-sections are set to zero. Note, however, that there must be no holes in the data files. If a molecule has holes in its absorption data, the corresponding cross-sections have to be stored as zeros in the binary file.

5.2 Available pre-tabulated data

We provide pre-tabulated absorption cross-sections in the format required by `Helios-r2` for a selected number of molecules. This data has mostly been calculated by Simon Grimm using the `HELIOS-k` model (https://github.com/exoclime/HELIOS-K). The folder includes all files needed to start a `Helios-r2` model with this opacity data and can be found here: https://chaldene.unibe.ch/data/helios-r2/.

We also provide the raw `HELIOS-k` data for a larger number of other molecules and line lists. The data can be found under https://chaldene.unibe.ch/

However, to use this data, it first has to be converted into the `Helios-r2` format. Most importantly, the `HELIOS-k` data is given as opacities in units of $\mathrm{cm^2 g^{-1}}$, whereas `Helios-r2` employs absorption cross-sections in $\mathrm{cm^2}$. Additionally, the `HELIOS-k` data is split in wavenumber space across different files, each with a range of $1000\,\mathrm{cm^{-1}}$. For more details on the `HELIOS-k` data, please check the documentation in the repository https://github.com/exoclime/HELIOS-K.

6 Output files

Once a retrieval and the subsequent postprocess are finished, all output files can be found in the retrieval folder that has been supplied in the command line argument.

`MultiNest` itself will provide the following, important output files:

- `post_equal_weights.dat`
  The posterior distributions of the model parameters and likelihood values.
- `summary.dat` and `stats.dat`
  A basic summary and some statistics of the nested sampling results.
- `live.points` and `phys_live.points`
  The list of all live points during the last iteration in both, the hypercube dimensions in the range of [0, 1] and the corresponding points in the physical dimension of the retrieval parameters.

For a detailed description of these files and all other output that `MultiNest` produces, please consult the `MultiNest` documentation.

`Helios-r2` itself will generate posterior spectra for each observation that has been used during the retrieval. These posterior spectra will be saved in files named `spectrum_observation_name.dat`, where `observation_name` is the name that has been used in the observation data files. Spaces in these names will be replaced by underscores. These files contain the spectral fluxes for each set of posterior points found in `post_equal_weights.dat`. Each column in the files correspond to one posterior set. The order of the spectral fluxes correspond to the ones from the associated observational data file.

In addition to the posterior spectra, specific forward models will generate their own posterior data. The Brown Dwarf forward model, for example, will generate temperature profiles and effective temperatures for all posterior samples. Information on the posterior files of the forward models can be found in Part II.

7 BrownDwarfSim example

The `Helios-r2` repository contains an example for a retrieval that can be found in the folder `BrownDwarfSim`. This folder contains all configuration files and data files to start a retrieval. This example is a simulated observation of a brown dwarf atmosphere that has been calculated by the `Helios-r2` forward model itself, with some noise added on top of the calculated spectrum and degraded to the resolution and wavelength coverage of the SpeX instrument.

The basic parameters for this model are: a $\log g$ value of 4.75 (with g in cgs units), a scaling factor f of 1.0, and a distance of 10 pc. The temperature profile has been taken from the `Sonora` grid of brown dwarf atmosphere models, provided by Mark Marley. This model has an effective temperature of 700 K and a $\log g$ value of 4.75.

For this example, the isoprofile chemistry was used, with the following species and mixing ratios: H_2O: $5 \cdot 10^{-4}$, CH_4: $5 \cdot 10^{-4}$, NH_3: $3 \cdot 10^{-5}$, CO_2: 10^{-9}, CO: 10^{-9}, H_2S: 10^{-9}, and K: 10^{-7}.

A retrieval of the spectrum should yield roughly these mixing ratios as posteriors, except for species the spectrum is not very sensitive to in the wavelength range used in this example.

Part II

Forward Models

8 General information

`Helios-r2` has been written to be as flexible as possible regarding adding new forward models or adapting old ones. Many aspects of a forward model are modularised and can, therefore, be easily changed. The chemical composition, the radiative transfer, and the description of the temperature profile are all contained in separates modules. Each module has a standardised interface that allows to easily exchange, for example, one chemistry module for another. Available modules are discussed in Part III.

The central task of a forward model is essentially to calculate a high-resolution spectrum based on a number of free parameters it receives during the nested sampling iteration. It is also responsible for e.g. selecting the opacity species and setting up the prior distributions for its parameters. Lastly, a forward model might also provide additional postprocessing methods to calculate specific output from the resulting posterior distributions. Normally, a forward model will use configuration files, in which important model parameters can be set. Those configuration files are forward-model specific and might differ between different models.

8.1 List of available forward models

Currently `Helios-r2` has only one fully developed forward model that is focussed on emission spectra of Brown Dwarfs and exoplanet atmospheres. Since there is only one forward model at the moment, there is no actual configuration parameter in the main retrieval configuration file to choose a specific one. This will change in the future once new forward models are added to `Helios-r2`.

9 The Brown Dwarf forward model

9.1 Overview

This specific forward model was written to perform retrieval on emission spectra of brown dwarfs. It was first introduced in Kitzmann et al. (2019), where a full theoretical description of the model can be found. The model has also already been successfully applied to secondary eclipse spectra of exoplanets, as demonstrated in Bourrier et al. (2019).

As mentioned in Kitzmann et al. (2019), this forward model can use two different radiative transfer schemes: a short characteristic method and the general multistream radiative transfer code `CDISORT`[1].

`Helios-r2` also contains two different chemistry modules: an equilibrium chemistry model based on the open-source `FastChem` code (Stock et al. (2018), https://github.com/exoclime/FastChem) and a free chemistry that assumes that the species' mixing ratios are constant throughout the atmosphere. While the equilibrium chemistry requires two free parameters (metallicity and C/O ratio), the free parameters of the isoprofile chemistry are the constant mixing ratios of the considered species (see Kitzmann et al. (2019) for details).

This forward model requires two additional configuration files: `forward_model.config`, a general config file with model parameters, and `priors.config` for the prior distributions. Both files need to be located in the retrieval folder that is passed as a command line argument when starting `Helios-r2`.

9.2 General parameter file

The general parameter file `forward_model.config` for the Brown Dwarf model has the following structure:

```
#Number of levels
70

#Bottom of atmosphere pressure (bar)
100

#Top of atmosphere pressure (bar)
1e-3

#Temperature profile, number of elements
6

#Temperature profile, polynomial degree
1

#Use grey cloud layer (N/Y)
N
```

[1] Note that `CDISORT` cannot be run on the GPU.

```
#Radiative transfer (scm/disort)
scm

#Retrieved chemical species
iso H2O CH4 NH3 CO2 CO H2S K

#Opacity species&folders
H2    H2
Na    Na
K     K
H2O   H2O_standard_parameter
CO    CO
CH4   CH4_standard_parameter
CO2   CO2
NH3   NH3_standard_parameter
H2S   H2S
```

It includes the following options that need to be specified:

- **Number of levels**
  The number of pressure/altitude grid points to be used in the one-dimensional atmosphere model.
- **Bottom of atmosphere pressure**
  The pressure at the bottom of the atmosphere in bar.
- **Top of atmosphere pressure**
  The pressure at the top of the atmosphere in bar.
- **Number of elements for the description of the temperature profile**
  An integer value that sets the number of elements for the temperature profile. See Kitzmann et al. (2019) for a description.
- **Polynomial degree for the temperature profile**
  Sets the polynomial degree for each temperature element. Note that only degrees up to order 6 are currently implemented.
- **Use grey cloud layer**
  Determines if a grey cloud layer should be used. If set to `Y`, the priors for this cloud layer must be added to the prior configuration file.
- **Radiative transfer**
  Choose between the short characteristic method (`scm`) or `CDISORT` (`disort`). Note that if you choose `CDISORT` and set to use the GPU in the main config file to `Y`, `Helios-r2` will quit with an error because `CDISORT` cannot be run on the GPU.
- **Chemical species**
  This determines the species that the retrieval will be performed for and also chooses

which chemistry module should be used. In the example above, the isoprofile chemistry is used (module keyword `iso`). The species for the retrieval are added after the chemistry module. Note that for the `iso` chemistry module, H_2 and He do not need to be specifically added to the list of species because they are assumed to form the background gas of the atmosphere (see Kitzmann et al. (2019)).

- **Opacity species**
  The species for which the absorption cross-sections are used. The first column contains the species' chemical symbols, the second the folder where the absorption cross-sections for each species are stored. `Helios-r2` will look within the opacity folder from the `retrieval.config` for these species' folders.

Note that the list of opacity species and the species for the free chemistry module don't need to be equal. However, species that are listed as opacity species but don't appear in the chemistry list, however, will have no impact on the spectrum because their mixing ratios are zero[2]. On the other hand, if you list a species for the chemistry module but do not add it to the list of opacity species, it also won't have a direct impact on the spectrum. It might have indirect effects, though, for example due to its impact on the mean molecular weight of the atmosphere or as a collision partner for collision induced absorption of another species.

In case the equilibrium chemistry model should be used instead, the corresponding configuration option looks as follows:

```
#Retrieved chemical species
eq fastchem_parameters.dat
```

The module is selected by using the keyword `eq`, followed by the name of the `FastChem` parameters file `fastchem_parameters.dat`. `Helios-r2` will look within the retrieval folder for that file. `fastchem_parameters.dat` contains the configuration parameters used by `FastChem`. Please consult the `FastChem` guide at https://github.com/exoclime/FastChem and the `FastChem` publication by Stock et al. (2018) for more details on these parameters. Note that you don't have to directly specify any chemical species. Instead, `Helios-r2` will use all the species that are contained within `FastChem`.

9.3 Prior configuration file

Besides the general forward model configuration `forward_model.config`, an additional file describing the prior distributions is required. The file is named `priors.config` and has to be located in the retrieval folder. The basic structure that corresponds to the options from the `forward_model.config` file shown above looks like this:

```
uniform     log_g           3.5     6.0
uniform     scaling_factor  0.1     5.0
```

[2] If the mixing ratio of a species is 0, `Helios-r2` won't even load and calculate the absorption cross-sections.

```
gaussian        distance          10.0    0.003
log_uniform     mr_h2o            1e-12   0.1
log_uniform     mr_ch4            1e-12   0.1
log_uniform     mr_nh3            1e-12   0.1
log_uniform     mr_co2            1e-12   0.1
log_uniform     mr_co             1e-12   0.1
log_uniform     mr_h2s            1e-12   0.1
log_uniform     mr_k              1e-12   0.1
uniform         temperature1      3000    1000
uniform         temperature2      0.3     0.95
uniform         temperature3      0.3     0.95
uniform         temperature4      0.4     0.95
uniform         temperature5      0.5     0.95
uniform         temperature6      0.5     0.95
uniform         temperature7      0.5     0.95
```

The file contains three columns, the first describing the type of the prior, a second column with a name for the corresponding prior that is used for output purposes and the last columns with the parameters for the chosen prior.

Three types of priors are currently supported by `Helios-r2`: `log-uniform`, `uniform`, and `gaussian` priors. The names in the second column should not contain whitespace characters.

It is important to note, that the Brown Dwarf forward model expects the priors to appear in a specific order. The first three parameters are the logarithm of the surface gravity[3], the scaling factor for the radius-distance relationship and the distance of the brown dwarf to the observer in pc. Please note that `Helios-r2` has no possibility to check if the order in which these priors appear in the configuration file is correct.

The next priors are the parameters for the chemistry, in this case the free mixing ratios for the isoprofile chemistry. Note that the priors have to be listed in exactly the same order as they appear in the list of the chemical species in `forward_model.config`. While `Helios-r2` can check that number of chemistry priors agrees with the number of species from the list, it cannot check if the lists are in the same order.

The last rows are the priors for the temperature profile, starting from the bottom of the atmosphere. Please consult Kitzmann et al. (2019) for details in how the profile is described. The total number of temperature parameters is equal to $K_e\, q + 1$, where K_e is the number of temperature elements and q the polynomial order set in `forward_model.config`. `Helios-r2` will check that the total number of temperature parameters equals the expected one. It can, however, not check if the priors make sense or are listed in the correct order.

In case the equilibrium chemistry and the grey cloud layer have been selected in `forward_model.config`, this file would look like this:

```
uniform         log_g             3.5     6.0
uniform         scaling_factor    0.1     5.0
gaussian        distance          10.0    0.003
uniform         metallicity       0.1     10
```

[3] The surface gravity g is assumed to be given in units of $\mathrm{cm\,s^{-2}}$

```
uniform         c/o             0.1     5.0
uniform         temperature1    3000    1000
uniform         temperature2    0.3     0.95
uniform         temperature3    0.3     0.95
uniform         temperature4    0.4     0.95
uniform         temperature5    0.5     0.95
uniform         temperature6    0.5     0.95
uniform         temperature7    0.5     0.95
log_uniform     cloud_top_p     1e-2    50
log_uniform     cloud_bottom    1       10
log_uniform     cloud_tau       1e-5    20
```

The equilibrium chemistry requires two parameters: the metallicity and the C/O ratio. The parameters for the cloud layer are located just below the temperature priors. The grey cloud layer is described by three parameters: the cloud top pressure in bar, the cloud bottom, defined as a fraction of the cloud top pressure, and the cloud's optical depth.

9.4 Postprocess output files

The Brown Dwarf forward model will perform its own postprocessing and write special output files. As described in Kitzmann et al. (2019), the forward model will estimate the effective temperature of the brown dwarf by calculating a high-resolution spectrum over a wider wavelength range and use the total flux together with the Stefan-Boltzmann law to compute a corresponding effective temperature. Currently, the Brown Dwarf forward model will write the following postprocess output:

- `temperature_structures.dat`
  This file contains the temperature profiles for all posterior samples. The first column is the pressure grid in bar. All subsequent columns are the temperature profiles corresponding to the posterior data from the `MultiNest` file `post_equal_weights.dat`.

- `effective_temperatures.dat`
  Contains the estimated effective temperatures for all posterior samples.

Part III

Helios-r2: Detailed Code Descriptions

TBD

Bibliography

Bourrier, V., Kitzmann, D., Kuntzer, T., et al. 2019, arXiv e-prints, arXiv:1909.03010

Feroz, F., & Hobson, M. P. 2008, MNRAS, 384, 449

Feroz, F., Hobson, M. P., & Bridges, M. 2009, MNRAS, 398, 1601

Grimm, S. L., & Heng, K. 2015, ApJ, 808, 182

Hamre, B., Stamnes, S., Stamnes, K., & Stamnes, J. J. 2013, in American Institute of Physics Conference Series, Vol. 1531, American Institute of Physics Conference Series, 923–926

Kitzmann, D., Heng, K., Oreshenko, M., et al. 2019, arXiv e-prints, arXiv:1910.01070

Lavie, B., Mendonça, J. M., Mordasini, C., et al. 2017, AJ, 154, 91

Stamnes, K., Tsay, S.-C., Jayaweera, K., & Wiscombe, W. 1988, ApOpt, 27, 2502

Stock, J. W., Kitzmann, D., Patzer, A. B. C., & Sedlmayr, E. 2018, MNRAS, 479, 865