cse2rad: a script to generate RADMC-3D input files from UDFA circumstellar envelope models

Paul M. Woods, *Astrophysics Research Centre, Queen's University Belfast*

May 2015

Cse2rad.py is a Python script which will take standard output files from the UDFA circumstellar envelope model[1] and produce input files suitable for the RADMC-3D[2] radiative transfer code. cse2rad.py runs on Python 2.x (tested on 2.7). It requires very little setup, needing just four inputs in order to run on your system.

Using cse2rad.py

Initial configuration

RADMC-3D can be used to perform line radiative transfer calculations. As such, the user must specify which molecular line or lines they are interested in. In addition, there are three other changes that must be made to the `cse2rad.py` script in order to adapt it to the user's system. Thus the user must specify:

1. The molecule of interest OR specify 'ALL' molecules
2. The expansion velocity (in km s^{-1}) of the circumstellar envelope in question
3. The system path to the RADMC-3D installation (or other directory in which to store input files)
4. The system path to a directory containing collisional data for molecules of interest.

See below for further details.

Choice of molecule

`cse2rad.py` may be run for a single molecule, or for all molecules for which there are collisional data. Typically molecular data files for RADMC-3D can be downloaded from the LAMDA database[3] at Leiden. At the time of writing, there are 67 collisional data files in LAMDA, however, some species have multiple files (according to different specifications) and some species are not available in the standard UDFA Rate12 species list (e.g., DCO$^+$).

The user must specify the value of the `specString` variable at the start of the `cse2rad.py` script. For a single atom, ion or molecule, the user must enter the species in the exact manner in which it is present in the UDFA CSE model output file, e.g.,

```
specString = 'SiS'
```

Alternatively, the user may desire to produce RADMC-3D input files for all potential species, in which case the user must enter 'ALL', i.e.,

```
specString = 'ALL'
```

Envelope expansion velocity

Unfortunately, the envelope expansion velocity is not written as an output in the standard UDFA CSE model, and so the user must enter this into `cse2rad.py` manually. Units for this quantity are km s^{-1}, and this is converted to cgs units (RADMC-3D standard) in the script.

[1] `http://udfa.net/index.php?mode=downloads`

[2] `http://www.ita.uni-heidelberg.de/~dullemond/software/radmc-3d/`

[3] `http://www.strw.leidenuniv.nl/~moldata/`

```
vexp = 14.5
```

The user may also wish to change the Doppler width, also expressed in $\mathrm{km\,s^{-1}}$.

```
db = 1.0
```

Path to the RADMC-3D installation

The user must specify the absolute path to their RADMC-3D installation, and to the location of the collisional data files for each molecule of interest, e.g.,

```
radmcPath = '/Users/abc/Desktop/radmc-3d/'
molecPath =
        '/Users/abc/Desktop/Ratran/molec/'
```

These are the places where `cse2rad.py` will write output files and look for molecular data files (in the `molec` directory), respectively. Please be sure to include a trailing '/' in the path.

Running cse2rad.py

Once `cse2rad.py` has been configured to your requirements, it should be placed in the directory with the output of the UDFA CSE code. By default, `cse2rad.py` will look for the file named `csnum_rate13.out`. This of course can be modified at the beginning of `cse2rad.py`:

```
dataFile = open('csnum_rate13.out','rb')
```

The user should be sure to use the CSE model output file which contains the **number density** of species, not that which contains fractional abundances.

The script can be run from the command line:

```
$ python cse2rad.py
```

or from the Python command line:

```
>>> execfile('cse2rad.py')
```

`cse2rad.py` will then generate the `*.inp` and `*.dat` files RADMC-3D requires.

Running RADMC-3D

RADMC-3D can be run as normal. Users should consult the extensive RADMC-3D documentation. Please note that `cse2rad.py` will generate all possible files from the UDFA CSE model output. However, the user must create (or adapt) additional files to run RADMC-3D successfully. These include (but are not limited to) files describing the central star, the external radiation field, the dust properties, the wavelength grid, geometry, etc. Some template files are generated by `cse2rad.py`, but these must be adapted by the user.

From the RADMC-3D directory:

```
$ radmc3d mctherm
$ radmc3d spectrum [options]
```

e.g.,

```
$ radmc3d spectrum iline 2 widthkms 15
```

would calculate a line profile of the second transition (`iline`) of molecule in the `lines.inp` file.

Alternatively, one can run the 'ALL' script, which will run RADMC-3D for all possible molecules, by overwriting `lines.inp` with `lines_all.inp`. Then the user would type:

```
$ radmc3d spectrum imolspec 2 iline 2
        widthkms 15
```

which would calculate a line profile of the second transition (`iline`) of the second molecule (`imolspec`) in the `lines.inp` file. The user can, of course, edit the RADMC-3D input files to their liking as they normally would.

Acknowledgments

PMW would like to thank STFC for financial support for this work.