dc2rad: a script to generate RADEX input files from UDFA dark cloud models

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

June 2015

Dc2rad.py is a Python script which will take standard output files from the UDFA dark cloud model[1] and produce input files suitable for the offline version of RADEX[2]. dc2rad.py runs under Python 2.x (tested on 2.7). It requires some setup before it can be run on your system, which can be done interactively in a terminal window or by editing the script itself.

Using dc2rad.py

Initial configuration

Several adaptations must be made to the `dc2rad.py` script in order to use it. The user can either edit the script directly, or switch on the interactive mode (`interaction = 1`) at the start of the script. The user must specify:

1. The molecule of interest OR specify 'ALL' molecules
2. The system path to the RADEX installation
3. The system path to the collisional data files (e.g., from the LAMDA database).

The user may also specify other parameters, which are given default values if no user-defined values are supplied:

1. The typical line width (in km s^{-1}) of the cloud in question
2. The diameter of the cloud in question, in parsecs
3. The cloud age at which outputs are required, in years
4. The filenames of the dark cloud model data files.

See below for further details.

Choice of molecule

`dc2rad.py` may be run for a single molecule, or for all molecules for which there are suitable collisional data. Typically molecular data files for RADEX 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 `dc2rad.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 dark cloud model output file, e.g.,

```
specString = 'SiS'
```

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

```
specString = 'ALL'
```

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

[2] [1] and `http://www.sron.rug.nl/~vdtak/radex/`

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

Path to the RADEX installation

The user must specify the absolute path to their RADEX installation, e.g.,

```
radexPath = '/Users/abc/Desktop/Radex/'
molecPath =
    '/Users/abc/Desktop/Radex/molec/'
```

This is the place where dc2rad.py will write output files and may also be where you instruct to look for molecular data files (in the molec directory). Please be sure to include a trailing '/' in the path.

Optional parameters

Not all necessary parameters for RADEX are specified in the UDFA dark cloud model output files, and so they must be specified. Sensible defaults are set in the script, but they may be altered for specific cases, or input interactively at the command line. The user may wish to change the line width expressed in $\mathrm{km\,s^{-1}}$:

```
db = 1.0
```

The diameter of the cloud, in parsecs:

```
cloudDiam = 1.0
```

The time (cloud age) at which the outputs are written (in years). Note this is a string variable and must maintain the formatting in order to match the model output files:

```
outputTime = '1.000E+06'
```

The dark cloud model output files themselves are given defaults, but of course may be adapted as necessary:

```
dataFilename = 'dc.out'
ssDataFilename = 'rate13steady.state'
```

Running dc2rad.py

Once dc2rad.py has been configured to your requirements, it should be placed in the directory with the output of the UDFA DC code. By default, dc2rad.py will look for the file named rate13steady.state in order to read the temperature of the cloud, and dc.out to read the remaining information it needs.

The script can be run from the command line:

```
$ python dc2rad.py
```

or from the Python command line:

```
>>> execfile('dc2rad.py')
```

dc2rad.py will then generate the *.inp files RADEX requires.

The script will specify the electronic transitions for each molecule according to which transitions fall into the various ALMA bands from Band 3 to Band 10 (excluding Band 5). Thus for CO, dc2rad.py will specify that RADEX should produce intensities for the transitions 1–8 (excluding 5) of CO, which in this case correspond to the J=1–0 to J=8–7 rotational transitions of CO. Transition numbers relate to the transitions listed in RADEX's molecular data files. If the user adds collisional data files to RADEX (those which are not from the LAMDA database), (s)he should ensure that dc2rad.py's fileToMol dict is updated with the filename, the species formula as present in the Rate12 database, and all possible transitions.

Running RADEX

RADEX can be run as normal. From the RADEX directory:

```
$ bin/radex < CO.inp
```

Alternatively, one can run the 'ALL' script, which will run RADEX for all possible molecules:

```
$ bin/radex < ALL.inp
```

The user can, of course, edit the RADEX input files as they normally would.

References

[1] Van der Tak, F.F.S., Black, J.H., Schöier, F.L., Jansen, D.J., van Dishoeck, E.F., (2007). A computer program for fast non-LTE analysis of interstellar line spectra. With diagnostic plots to interpret observed line intensity ratios. *A&A*, 468, 627.

Acknowledgments

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