ARtful modelling Code

ARCiS

for

exoplanet Science

User Guide to ARCiS

Michiel Min

June 4, 2019

1 Introduction

1.1 Terms of use

By using ARCiS you agree to the following:

- You are not permitted to pass (parts of) the code to anyone else. If anyone else is interested, let him/her drop me an email: M.Min@sron.nl

- You offer me co-author rights on any paper that uses results computed with ARCiS

2 Basics

3 Installing ARCiS

Before installing ARCiS you need:

A Fortran compiler: This can be either `gfortran` or `ifort` (any other might work but is not tested).

`cfitsio` library: This is needed to allow fortran to read and write binary fits files.

`MultiNest`: This allows ARCiS to perform Bayesian retrievals.

On a Mac the easiest is to install cfitsio through HomeBrew (google 'homebrew' for installation instructions. After that:

```
# brew install cfitsio
```

Next fetch MultiNest:

```
# git clone https://github.com/JohannesBuchner/MultiNest.git
# cd MultiNest/build
# cmake ..
# make
# sudo make install
```

Next fetch the git source code from:

```
# mkdir ARCiS ; cd ARCiS
# git clone https://github.com/michielmin/ARCiS.git ./src
# cd src
# make gfort=true multi=true
```

This creates the ARCiS binary, which you can put in any path accessible (/usr/bin or something like that).

Now you also need all the data files. These need to be stored under `$(HOME)/ARCiS/Data/`

4 Using ARCiS

The files are:

`ProDiMoForFLiTs.fits.gz` This file contains all the details of the model needed by FLiTs. It contains the density and temperature structure, the opacities of the dust, the abundances of the molecules, and the non-LTE level populations. It also contains some keywords needed by FLiTs such as the radius and mass of the star, and the distance to the source.

`inputFLiTs.dat` This is the input file needed by FLiTs where the details of the FLiTs run can be specified. By default ProDiMo provides a file where all molecular species are included in the FLiTs run.

`*Lambda.dat` These are files that contain for each molecule the levels, and Einstein coefficients for the transitions. It is important to keep the right Lambda files with the accompanying ProDiMoForFLiTs.fits.gz file, since the level numbering has to be consistent between the two.

These are all files needed by FLiTs. In principle directly after ProDiMo a default run of FLiTs can be done by using the command

```
# FLiTs inputFLiTs.dat
```

which creates the file `specFLiTs.out` containing the default wavelength interval.

There are several options included in FLiTs. These are given as keywords in the `inputFLiTs.dat` file (or whatever you call it). Keywords are always given as `key=value` and can be anywhere in the file (order does not matter). Also, you can overwrite keywords set in the input file from the command line in the following way

```
# FLiTs inputFLiTs.dat -s key1=value1 -s key2=value2
```

Any number of keys can be set on the command line. Just make sure the first argument of the command line is the name of your input file. Note that FLiTs always takes the last keyword value it encounters, first reading the input file, next the command line keywords one by one.

The possible keywords are described below.

5 Most important keywords

FLiTsfile

This keyword sets the FLiTsfile to use. When using ProDiMo output, this is likely `ProDiMoForFLiTs.fits.gz`, but you can rename this file, for example to store different models.

linefile

This keyword can be provided as often as you like (maximum up to 100). Each entry specifies a Lambda file with the information on a particular species that has to be included. These files are provided by ProDiMo with level numbering consistent with that used in the `ProDiMoForFLiTs.fits.gz` file. However, you can choose to provide different Lambda files. However, you have to make sure that in this case you switch to LTE mode (see below) to avoid inconsistencies in the level populations.

`LTE`

When this keyword is set to `.true.` (default is `LTE=.false.`), the level populations are computed in LTE. This can be useful when you want to use a different Lambda file as the ones provided by ProDiMo.

`lmin` and `lmax`

These keywords set the minimum and maximum wavelength used (in micron). Default is `lmin=5` and `lmax=50`. FLiTs always computes all lines in the given wavelength interval.

`vres`

This keyword sets the velocity resolution of the output spectrum in cm/s. Default value is `vres=1e5`, i.e. 1 km/s. Note that increasing the resolution by a factor of 10 increases the computation time by more than a factor of 10. This is because for higher resolution velocity spectra, also a finer spatial sampling of the rays is required.

`inc`

This keyword sets the inclination angle of the disk in degrees with respect to pole on (0 degrees). Default value is `inc=30`.

`accuracy`

With this keyword you can set the level of accuracy. The default is set for speed (`accuracy=1`). The different options are:

`0` - Go for speed! (time ∼ 0.12 s per line)

`1` - Default. Increased radial sampling of the rays (time ∼ 0.20 s per line)

`2` - Increased azimuthal sampling of the rays (time ∼ 0.57 s per line)

`3` - Further increased azimuthal sampling (time ∼ 0.88 s per line)

`4` - Increased sampling of radial and azimuthal rays (time ∼ 1.48 s per line)

5 - Maximum sampling of radial and azimuthal rays (time very long)

The times here are very rough indications for one particular model and depend a lot on the model setup, the number of species, and the amount of line-blending in the model. What accuracy is needed depends on the application. Often `1` or `2` are the best options.

6 Optional keywords

While all keywords are in principle optional (there are always defaults). The keywords below are probably not useful to the general user.

`blend`

Default `blend=.true.`, i.e. compute line blends consistently. With this keyword you can switch off the computation of line blends. This means all lines are treated separately, and in the `specFLiTs.out` file you may find some wavelengths multiple times, since they were computed for different lines. This does increase speed a bit (up to a factor of 2 for very specific cases).

`cylindrical`

By default FLiTs uses a cylindrical grid setup (`cylindrical=.true.`). However, when you create a FLiTs file using another program (one that has a spherical grid setup for example. you can switch it to `.false.`, in which case the grid will be spherical.

`vres_profile`

This keyword can set the velocity resolution of the sampling of the profiles intrinsically. It has no influence on the output resolution. By default it is set to `vres_profile=1e4`, i.e. 0.1 km/s. The only reason to increase this is when you have very narrow lines.

`tau_max`

This keyword sets the maximum optical depth to which FLiTs raytraces. Default is `tau_max=15`, so it stops a ray when an optical depth of 15 is encountered. This is quite generous, in principle regions shielded behind 15 optical depths don't contribute to the output flux. Decreasing this threshold might improve speed only a bit, but in general it is recommended not to change this value.

7 Output file

FLiTs creates one single output file `specFLiTs.out`. This file contains the spectrum computed including all lines. The different columns in this file are:

column 1 wavelength in micron

column 2 flux in Jy

column 3 velocity with respect to the center of the closest line in the blend

column 4 flux of the continuum

column 5 comment which species contributed to this blend

8 Examples

As an example we run the standard TTauri disk setup provided with ProDiMo. We first run ProDiMo to get the radiative transfer and chemistry. Afterwards we run FLiTs using all standard lines and the entire spectral range.