# fu_ori

A repository containing instructions and parameter files for running FU Ori type models with Python.

Tested with the latest commit to the dev branch of agnwinds/python https://github.com/agnwinds/python/commit/f61670e9e8a2fda11fb001cbe

Relevant references mentioned throughout this README:

- SLD05: Sim et al. 2005, describes YSO model
- KWD95: Knigge, Woods and Drew 1995, describes actual wind prescription and geometry
- LK02: Long & Knigge 2002, describes the code.

## Files in the repository

- yso_a.pf: YSO model based on Stuart Sim's 2005 model but with some adjustments described below
- yso_ionization.dat: sample ionization file
- yso_a.pf: YSO model based on Stuart Sim's 2005 model but uses a readin temperature profile
- yso_tprofile.dat: completely fudged temperature profile file. please adjust, but tells you the format.
- yso_example_na_line.png: example output from running yso_a.pf. You see nice Na I lines! :)
- py_read_output.py: routines for reading output files
- plot.py: an example of how to plot the spectrum at 85 degrees

## About the model

The model here is based on model A from SDL05. However, there are some differences. SDL05 are interested in modelling the hydrogen profiles. To do that, you’d need to use the “macro-atom” line transfer mode, and the “recalc_bb” ionization mode. You could do that by changing these lines in a parameter file.

```
Atomic_data                                             data/h20
Wind_ionization(0=on.the.spot,1=LTE,2=fixed,3=recalc_bb,6
=pairwise_bb,7=pairwise_pow,8=matrix_bb,9=matrix_pow)   3
Line_transfer(0=pure.abs,1=pure.scat,2=sing.scat,3=escape
.prob,6=macro_atoms,7=macro_atoms+aniso.scattering)   7
```

Other than that, everything should pretty similar.

## Key parameters

**IMPORTANT – To run with fixed temperature:** Use the -f flag. This will keep the Electron temperature fixed to the wind t_init value.

Wind prescription: Parameters starting kn correspond to the variables defining the KWD95 wind prescription. This are described by KWD95 and the fiducial
values are given by SDL05. I have adopted the values of SDL05, which makes for a very equatorial wind, so you’ll only see absorption lines at 85 degrees.

Disk: The disk is set up to use a Shakura-Sunyaez temperature profile. This can be changed by providing your own file as described (note slightly odd units of 1e11cm and 1e3K for temperature profile file.) I haven’t tested this mode so use with caution and check it produces the spectrum you expect.

Ionization mode: the parameter files I’ve given used “fixed” ionization mode. This means they require an additional concentrations file name which has format “Atomic number, ionization stage, relative abundance”. Alternatively, you can use recalc_bb to adopt the Mazzali & Lucy (1993) ionization method as described by LK02

Line transfer mode: This is set to 5, which is the escape probability formalism as described by LK02 among others. To use macro-atoms, use 7. No need to worry about the other modes.

Atomic Data: You’re using the standard set, but it won’t matter match in fixed ionization mode. Contact me if you want to change things up.

# Running the model

First, Setup the environment with

```
Setup_Py_Dir
```

Then, once the code is compiled, simply run with

```
py -f yso_a
```

For a fixed temperature run.

# Processing the outputs

Python produces a number of output files. The most important are the ".spec" files, which contain either flambda or fnu spectra at each angle at 100 parsecs. The bins are in frequency space. the ".log_spec" file instead uses log frequency, which is sometimes more useful. I've provided a
number of scripts here which are based on those in the $PYTHON/py_progs/ folder. They allow processing of the output files.

the file plot.py shows you how to plot a basic flambda spectrum.

Any questions get in touch.