Making Kippenhahn diagrams using MESA output

1 Introduction

Kippenhahn diagrams (KHDs), as discussed in the lecture notes and the lectures, are plots showing the internal structure of a star with time. These plots can be useful for your final reports of the MESA projects you will be conducting for this course. Instead of making a screenshot of the KDHs that MESA outputs on the screen via PGSTAR as you run it, which might not always illustrate the information you want or not be very clear, you can use a Python script known as mkipp[1] that makes plotting your own KHDs quite easy. However, you do have to turn on the right output columns for your history.data and profile.data files. This document will help you set up mkipp, get the right MESA output, and become familiar with the scripts.

2 Setting up

Download the zip file from https://github.com/orlox/mkipp by pressing the green Code button and choosing the Download ZIP option. Extract the 3 Python scripts titled mkipp.py, mesa_data.py, and kipp_data.py from this zip file to the directory you plan to make plots in. In addition, extract the Python script example.py there too, along with the directory LOGS[2]. The first 3 of these scripts are integral parts of mkipp, the latter example.py file gives you example lines of code on how to make KHDs employing mkipp while using the data from its LOGS directory. If you want to clean up your working environment a bit, you can put mkipp.py, mesa_data.py, and kipp_data.py each into their own directory of the same name (without the .py extension of course) inside your plotting directory, and changing the names of each of the files to __init__.py where __ is a double underscore.

2.1 Setting MESA output

As you can read in mkipp.py, to use mkipp you need to have the following MESA output available:

```
Requirements: history.data and profiles.data containing
              History (star_age,model_number,star_mass,photosphere_r,
              mixing_regions,mix_relr_regions)
              Profile (mass,radius,eps_nuc)
```

This entails editing the two files inside your MESA model titled history_columns.list and profile_columns.list[3]. These files specify what columns of data MESA will output into the history.data and profile.data files inside the LOGS directory. Open these files and uncomment the columns mentioned above by removing the '!' in front of the variable. When you uncomment mix_relr_regions, it will most likely say <integer> after it. You should change this to 10, so that it has the same value as for mixing_regions.

[1] https://github.com/orlox/mkipp

[2] Do not confuse this with the other LOGS directories mentioned later on, this is example.py's LOGS directory.

[3] Examples of these files are included in the session2.tar file.

For MESA to use the columns you have specified in these 2 files instead of the default set of columns, we have to make sure MESA knows that it needs to read these 2 files. To do this, we have to open inlist_project and include the following under &star_job:

```
! to specify which output columns we want in history.data and profile.data
    history_columns_file = 'history_columns.list'
    profile_columns_file = 'profile_columns.list'
```

3 Making plots

Now that we know the required output will actually be present in the output files, we have to write scripts to make the KHDs. The MESA output inside the LOGS directory after you have run your model needs to be moved to the same directory as where your 3 Python scripts of mkipp reside. It is recommended to run example.py, to see if everything is working properly for you and to understand how to write the code that makes KHDs. You can do this by copying its content, pasting it into a Jupyter Notebook[4], and running it. Furthermore, read through mkipp.py to see how the input parameters and functions are designed. The most important object you need to look at is Kipp_Args in mkipp.py, because this shows the default input parameters for mkipp.kipp_plot as shown in example.py. If you just want to make a KHD with these parameters, you should execute example 1 in example.py. If you want to plot, for example, time in Myr instead of model number along the x-axis, this is done in example 3 in example.py (this one shows the He abundance instead of the energy production). Try to understand at least the first four examples in example.py, the other examples you probably will not need.

Any Python script you write that uses mkipp needs to be in the same place as where the three Python scripts mkipp.py, mesa_data.py, and kipp_data.py are located (or in the directory containing the correspondingly named subdirectories that contain these three files, as previously mentioned). In your Python script you will need to at least add the following line:

```
import mkipp
```

Furthermore, to use the functions inside the other 2 scripts for yourself, you need the following lines of code too:

```
import kipp_data
import mesa_data
```

As shown by example.py, the mkipp scripts, in particular Kipp_Args, expect the MESA output data to be in a directory titled LOGS where these scripts are. However, if you want to plot more models with the same scripts in the same location, you can put your MESA output data in a directory with a name pertaining to the MESA model (e.g. model1) inside a directory titled for example data as to have a clear overview of what data you are exactly plotting. You will only need to define the logs_dir argument of the Kipp_Args function as the path of the output data from the script's location as a string inside a list. An example of this is given below:

```
mkipp.Kipp_Args(logs_dirs = ['data/model1'])
```

3.1 Binary stars

If you want to make a KHD of one of the stars in a binary system, you will also have to change the folder in which the code looks for the history.data and profile.data files, e.g. defining the proper path for logs_dir. The primary of the binary is found in LOGS1, and the secondary in LOGS2. You will see this first-hand during MESA session 3.

[4]Accessed on the university computers via https://jupyterhub.science.ru.nl/.