Tricking MESA into making Exoplanets

Konstantinos Kilmetis

November 2023, Leiden

Hi, here I'll detail to the best of my ability what one needs to do to convince MESA to simulate exoplanets with escaping atmospheres instead of stars. These insets and the associated changes to MESA were developed by Dr. Daria Kubyshkina. I am only documenting my efforts on installing and using them.

I am going to assume that you are using a UNIX based system, be it a Linux distribution or OSX. I am partial to Linux Mint and currently use version 21.1. No windows users allowed! Just kidding, if you ~~like having Microsoft spy on you~~ prefer Windows, I recommend that you make either a virtual machine and use Linux there or use WSL.

In the[1] case that you find an error here, e-mail me at kilmetis@strw.leidenuniv.nl

1 MESA and its court

The bit of software in the heart of it all is Modules for Experiments in Stellar Astrophysics. The eagle eyed reader will have noted that it says, Stellar; not planetary. MESA is an old but exceptionally successful of piece of software. It is written in FORTRAN by a team led by B. Paxton[2]. (Paxton et al., 2011)

MESA is very good at what it does. It is one of the most widely used codes throughout the world of computational astrophysics and enjoys all there is to enjoy from having a vast community. Through the years, there have been a great many modifications for tackling all sorts of scenarios about simulating the structure of all sorts of vaguely spherical objects. What you really need to understand about it is:

- It is 1-D. It solves all the PDEs it needs to in the r direction. It assumes spherical symmetry. This means that it can't *really* account for multi-dimensional effects like convection and tidal forces. It is very clever about how it tackles these, but they just inherently require more than 1 dimension to simulate properly.
- It has plenty of versions, each of them fixing some bugs and introducing new functionalities, which in turn give rise to new bugs and so on and so forth.
- It is finicky. Do *exactly* as it asks you. Or else...

1.1 Downloading

We care about **version 12115**. It can be found here.

For MESA to work, it needs to be sure that all it's dependencies are in one place. This is known as the *software development kit* (hereafter SDK). It contains a bunch of compilers, debuggers, plotters and some very fast libraries handling all the linear algebra. It is a great idea to make sure that have a version of the SDK that came out at about the same time as our version did.

[1] very probable

[2] who after "retiring" due to being an integral part of the Adobe team which made the basis of the .pdf format, chose to develop a stellar evolution code as an external researcher at UC Berkley instead of drinking Malibus in Hawaii. Thanks Bill!

I use **20190830**. It can be found here.

1.2 Setup

To have MESA work, you need to tell your computer where exactly it is located. To do that, on Linux, you need to:

- Go to your home folder
- Right click, select show hidden files. (This may change depending on what file manager you use)
- Find the file .bashrc, open it with a text editor.
- Paste the following:

```
# Path to Mesa version 12115
export MESA_DIR=~/your/folder/mesa-12115

# Number of cores on your machine
export OMP_NUM_THREADS=6

# Path to SDK
export MESASDK_ROOT=~/your/folder/mesa-sdk
source $MESASDK_ROOT/bin/mesasdk_init.sh

# Path to Spada models
export path/to/spads/fs255_grid/
```

Where I assume that you have unpacked the MESA version into a folder named mesa-12115 and the SDK in mesa-sdk. When you edit these paths, the tilde means that the path starts ftom the home folder. Do not put spaces in either your path or the .bashrc

1.3 Install and Testing

To install, go to the folder where you put mesa-12115 in and type

```
./install
```

If the machine spirit is pleased with you, you should be greeted by this:

```
************************************************
************************************************
************************************************

MESA installation was successful

************************************************
************************************************
************************************************
```

To make sure it works, go to mesa-12115/test-suite/1.3M_ms_high_Z (or any other folder in the test suite) and try to run it by typing:

```
./mk
./rn
```

2 Daria's Insets

If you explore a bit around the test suite, you'll notice some files called insets. These specify the physics that we want MESA to care about for every step. In our case this entails a complicated sequence of steps which convince it to simulate a planet instead of a star. The exact sequence of these steps and reasoning behind them is given in (Kubyshkina et al., 2020).

Download the insets from here

2.1 Folder Structure

I recommend that all of your code lives on the same folder. Here is the structure that I am using:

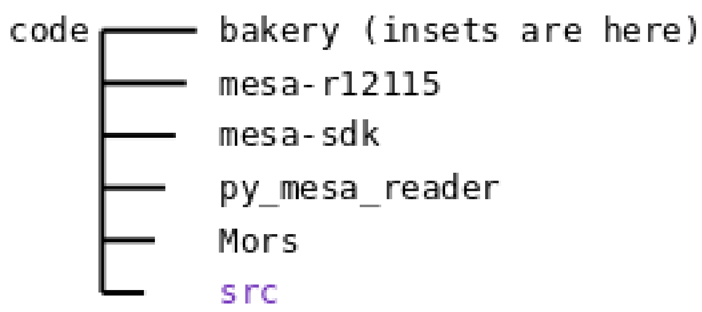

Figure 1: My folder structure, `src` has my own code that I use for post-processing MESA's output

In general, it is a neat idea to start a new `python` enviroment for every new project you start. I recommend you do that. Because we are using some really boutique packages, I also recommend that you stay as close to the metal as possible and stick with `pip`[3].

2.2 MESA Reader

To manipulate MESA output with `python` I recommend using the package Mesa Reader.

To install, make sure you have activated your new environment and then navigate to the `code` directory specified above and then:

```
git clone https://github.com/wmwolf/py_mesa_reader.git
pip install .
```

2.3 MORS

A crucial part of the atmospheric evolution is the knowing the amount of FUV radiation which irradiates the planetary atmosphere. In (Kubyshkina et al., 2020) they provide a prescription for it, but the new version of the code goes on step further and uses MORS.

The specifics of how MORS works are besides the point, we can treat it as a function which works as shown below. We provide the mass and initial rotation rate of the stars, let MESA tell us how far along its evolutionary path we are and MORS will tell us the X-ray and UV flux of the star.

[3] If you know enough about `conda` to know that you can use `pip` within it, you are not in the target audience of this document. Either way, `conda` has failed me in that regard in the past and I've learnt to avoid it.

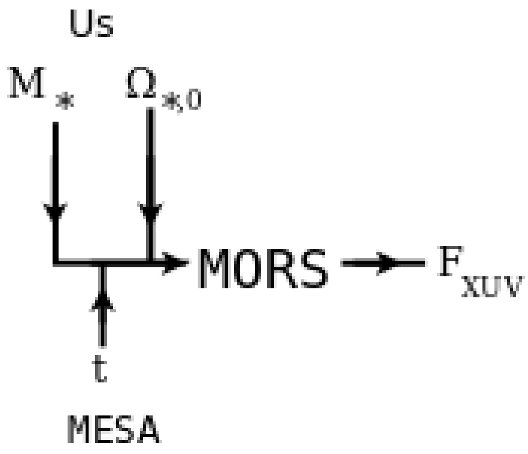

Figure 2: Diagram depicting how I think MORS works

Again, make sure you're in the `code` directory and have your python environment activated, use the following to install the package

```
git clone https://github.com/ColinPhilipJohnstone/Mors
pip install .
```

MORS makes use of some tabulated stellar models, from (Spada et al., 2013). You may download them from here. Make sure to place these somewhere where they'll not be touched. Then, point to them through the in `.bashrc`, as shown in section 1.2

3 Baking Planets

With all that done we are ready to start baking planets! By baking we refer to the simulating. Why this name stuck around, is left as an exercises to the reader.

Each simulation is controlled -mostly- by the following 7 parameters

1. Atmospheric Escape Prescription.
2. Planetary Mass, M_p
3. Initial Atmospheric Fraction, $f_{at,0}$. How much of the planet's mass is in it's atmosphere
4. Core Radius, R_p
5. Entropy recipe.
6. Host Star FUV flux (through MORS)
7. Orbital Separation, α
8. (bonus!) Observed Radius at some age

3.1 Atmospheric Escape Prescription

There are 4 choices:

1. `HD`. Hydrodynamic Escape. This should be the most accurate one, since it employs the results of detailed hydrodynamic simulations (Kubyshkina et al., 2018). Use it when you can. Only works for planets with
   - $M_p \in [1, 45]\ M_\oplus$
   - $\alpha \in [0.002, 1.3]$ AU
   - $M_* \in [0.4, 1.3] M_\odot$
2. `EL`. Energy Limited Approximation. The old way of doing things. Used as a baseline to compare other perscriptions against.

3. ELR. Energy limited + radiation-recombination limited escape as described in (Chen and Rogers, 2016). Don't use for small planets. $M_p < 109\ M_\oplus$

4. zero. No escape!

3.2 Planetary Mass

You can experiment a lot with these, I've reliably pushed till 1000 $M_\oplus$. Below are some reference values

- Venus: 0.9 $M_\oplus$
- Earth: 1 $M_\oplus$ (duh)
- Jupiter: 317 $M_\oplus$
- Saturn: 95 $M_\oplus$
- Uranus: 14 $M_\oplus$
- Neptune: 17 $M_\oplus$

3.3 Atmosphere Fraction

This is the envlope fraction, what percentage of the planet's mass is contained in its atmosphere. It ranges between 0 and 1, but pushing it to too low or too high values will either break the code or have it take too long. For comparison:

- Venus:
- Earth:
- Jupiter: 0.4-0.6
- Saturn
- Neptune: 0.3
- Uranus: cite this from fortney and nettelman.

3.4 Core Radius

You may allow MESA to set it by chooising -1. Alternatively, you may use a perscription from (cite this) by choosing -2.

3.5 Entropy

You may allow MESA to set it by setting to -1. Again, you may use a -really arbitrary- prescription from **(?)** by choosing -2. This choice is more stable, but you should not trust its results early on. It is okay if you care about the latter stages of evolution.

In general -1 is more accurate but -2 is more stable and fast. If you want to generate a lot of profiles, choose -1. I do not know why but the -2 option just doesn't generate as many. Probably the timestep becomes too big.

If you pass a positive float here, the code will treat this as the value of entropy in [k_b/baryon]. If you do this, you probably know what you're doing. For what it's worth, I've never done this.

3.6 Host Star

MORS accepts the following ranges of values:

- $M_* \in (0.1, 1.2) M_\odot$
- $T_* \in (0.05, 60)$ days

UV and X ray fluxes increase (in weird and non-linear ways, read (Johnstone et al., 2021) for more) with rotation and mass. Mass is -usually- the more important factor.

3.7 Orbital Seperation

I've had runs succeed using orbital separations from 0.05 AU to 100 AU. I don't know if I trust the results yet, but it's gonna compile :D

4 Tips and Tricks

- `screen_log.txt` is your friend. That's all the relevant output. Read `MESA` output only when you *absolutely* have to.
- If you're trying to re-do a planet, it often pays to to a soft reset by,

```
    rm p3*
    rm p4*
    rm p5*
    rm p6*
    rm p7*

```

- Do not delete everything in LOGS, the history files that are part of the `.zip` are required for putting the core in. If you need to do a hard reset, just copy-paste everything from the `.zip` again. It is the easiest way.
- I prefer to work by editing `main.py` and then calling it from the terminal. It is the most robust way. Some IDEs have trouble with `MESA`'s massive output.
- Your simulation may stop due to Roche Lobe Overflow, even if a very rarefied part of the planets atmosphere actually overflows. Use -2 for Entropy. If that fails, move the planet further away, tis not meant to be...
- You may want more profiles. If this is the case, you'd logically go to `mesa-r12115/star/controls/conrols.defaults` and change `profile_interval`. Amateur mistake. Why would this be controlled by `MESA`'s global settings?

  What you need to do is change all inlist 7 (EL-HD-zero choices for atmospheric evaporation) to include the line

```
        profile_interval = whatever value

```

  I usually add it beneath history and terminal interval.

- Don't decrease this too much. When you exceed the maximum allowed profiles saved - which is changed by adding `max_num_profile_models = your value` - `MESA` **will overwrite the existing profiles, starting by 1**. To illustrate this, assuming a profile interval of 2, it going to save like:

```
        model 196 profile 98
        model 198 profile 99
        model 200 profile 100
        model 202 profile 1
        model 204 profile 2

```

Lovely.

- Anytime you make a change to the inlists, you probably also want to go delete the `tmp_inlists` directory.

- A message like:

```
          At line 119 of file ../src/run_star_extras.f (unit = 14, file = 'mors_tracks/
    track_1_150.0_1.dat')
          Fortran runtime error: End of file

```

  while getting normal successful output is okay. It means that your star died! It's normal, they tend to do that. G stars die at about 10 Gyrs. Just go make a new one.

References

Howard Chen and Leslie A. Rogers. Evolutionary Analysis of Gaseous Sub-Neptune-mass Planets with MESA. *Astrophysical Journal*, 831(2):180, November 2016. doi: 10.3847/0004-637X/831/2/180.

C. P. Johnstone, M. Bartel, and M. Güdel. The active lives of stars: A complete description of the rotation and XUV evolution of F, G, K, and M dwarfs. , 649:A96, May 2021. doi: 10.1051/0004-6361/202038407.

D. Kubyshkina, L. Fossati, N. V. Erkaev, P. E. Cubillos, C. P. Johnstone, K. G. Kislyakova, H. Lammer, M. Lendl, and P. Odert. Overcoming the Limitations of the Energy-limited Approximation for Planet Atmospheric Escape. *Astrophysical Journal Letters*, 866(2):L18, October 2018. doi: 10.3847/2041-8213/aae586.

Daria Kubyshkina, Aline A. Vidotto, Luca Fossati, and Eoin Farrell. Coupling thermal evolution of planets and hydrodynamic atmospheric escape in MESA. *Monthly Notices of the Royal Astronomical Society*, 499(1):77–88, November 2020. doi: 10.1093/mnras/staa2815.

Bill Paxton, Lars Bildsten, Aaron Dotter, Falk Herwig, Pierre Lesaffre, and Frank Timmes. Modules for Experiments in Stellar Astrophysics (MESA). *The Astrophysical Journal Supplement*, 192(1):3, January 2011. doi: 10.1088/0067-0049/192/1/3.

F. Spada, P. Demarque, Y. C. Kim, and A. Sills. The Radius Discrepancy in Low-mass Stars: Single versus Binaries. , 776(2):87, October 2013. doi: 10.1088/0004-637X/776/2/87.

5 Trials and Errors

Name	Mass [M_earth]	Init. Atm. Frac.	Atm. Escape	Rcore [R_earth]	Entropy	Orb Sep [AU]	Stellar M [M_sun]	Stellar Ω [days]	Step	Output
klassikos1	12,5	0,1	HD	-1	-1	0,05	0,83	1	6b	min_timestep_limit. Evo ter
klassikos2	12,5	0,1	HD	-1	-2	0,05	0,83	1	6b	dt_is_zero, max_age_reache
klassikos3	12,5	0,1	EL	-1	-1	0,05	0,83	1	all	max_age
klassikos4	12,5	0,1	EL	-1	-2	0,05	0,83	1	6b	dt_is_zero, max_age_reache
klassikos-zero	12,5	0,1	ZERO	-1	-1	0,05	0,83	1	all	stop because star_age >= ma
klassikos-zero2	12,5	0,1	ZERO	-1	-2	0,05	0,83	1	all	max_age
xodros	320	0,1	EL	-1	-1	0,05	0,83	1	all	stop because star_age >= ma
xodros2	320	0,1	EL	-1	-2	0,05	0,83	1	all	termination code: max_age
xodros-puffy	320	0,5	EL	-1	-1	0,05	0,83	1	all	max_age
xodros-puffy2	320	0,5	EL	-1	-2	0,05	0,83	1		