

# MESA tutorial: session 1

## 1 What is MESA?

MESA stands for Modules for Experiments in Stellar Astrophysics. It is an open-source stellar astrophysics code consisting of different modules. Each module is responsible for a different aspect of the numerics or physics required to construct computational models for stellar astrophysics. At the core of it all is the most important module: `MESA star`. `MESA star` is a one-dimensional stellar evolution code and is designed for a wide range of stellar physics applications. You can find more information on the website <https://docs.mesastar.org/>, and also in the first MESA instrument paper by Paxton et al. [2011]

MESA is one of (if not the most) the most widely used stellar evolutionary codes today, with applications ranging from binary systems to planet structure, nuclear burning in the accretion layers of neutron stars, and asteroseismology. In this course, we will learn some of the basics of using MESA, as well as how to set up your own numerical experiment. By the end of the course, you will be able to set up and run your own stars!

## 2 Getting started

MESA has already been installed on the Science Faculty computer system of Radboud University. This section will explain how to access this. The MESA installation is not the newest version, but an older one named `r12778` which is found at <https://zenodo.org/record/3698354#.Yyxmg3ZBxPZ>, because this release runs stably for the computer exercises you will perform during this course. <sup>1</sup>

To access the MESA installation, you first need to log in on the Linux environment of the university computers at the Science Faculty. If you cannot manage to log in with your science account, you can find helpful information here: <https://wiki.cncz.science.ru.nl/Login>.

Once logged in, open a terminal and either run the following lines of code, or directly add them to your Linux shell start-up file (i.e., `~/.bashrc`).<sup>2</sup> The latter approach avoids having to run these commands manually each time. Otherwise, **you will have to run them in every new terminal you open** to make MESA work.

```
1 #This line defines the path where MESA is located
2 export MESA_DIR=/vol/cursus/astro/ASBE/MESA/mesa-r12778
3
4 # Number of cores the computer has for MESA to use
5 export OMP_NUM_THREADS=4
```

---

<sup>1</sup>If you want to install MESA on your own personal computer in your own time, follow the instructions on the MESA website (<https://docs.mesastar.org/en/stable/installation.html>). The corresponding MESA SDK (software development kit) can be downloaded from <http://user.astro.wisc.edu/~townsend/static.php?ref=mesasdk>, making sure that your SDK matches the ‘MESA Version at Release’.

<sup>2</sup>More information can be found in the subsection **Set your environment variables** at <https://docs.mesastar.org/en/release-r22.05.1/installation.html#>.

```

6
7 # path to the MESA SDK
8 export MESASDK_ROOT=/vol/cursus/astro/ASBE/MESA/mesasdk
9 source $MESASDK_ROOT/bin/mesasdk_init.sh
10
11 # Change the MESA caches directory to a local scratch directory
12 mkdir -p /scratch/$USER/MESA/caches
13 export MESA_CACHES_DIR=/scratch/$USER/MESA/caches

```

Line 2 specifies the installation path for MESA. Line 5 sets the number of threads for MESA using the `OMP_NUM_THREADS` variable, related to OpenMP for parallel programming. We set it to 4 to match university computers. Use `lscpu` to check your machine's CPU count if unsure. Lines 8 and 9 define and initialize the MESA SDK path.

Lines 12 and 13 set the path where MESA stores its cache data. The `mkdir -p` command creates a cache directory in the local `scratch` area (e.g., `/scratch/$USER/MESA/caches`). The `-p` flag ensures that you will also create any missing parent directories, plus that command succeeds even if the directories already exist. Note that using `scratch` instead of `$HOME` significantly improves performance because it leverages fast, local storage. And there is more space available on `scratch` than on your `home` directory. This path may vary per machine and may not persist, which is why it is good practice to run these lines on any new system or after a wipe of the local `scratch` directory.

The block of code mentioned above have been put for your convenience in a text file titled `MESA_startup_script`, which is on Brightspace with this document. However, we strongly recommend to add these lines to your `~/.bashrc` file, so that they are automatically run every time you open a new terminal window.

### Pro Tip

If (or when) you have multiple versions of MESA installed, you can use shell functions to easily switch between them. For example, you can create a function in your `~/.bashrc` for each version.

## 2.1 Running MESA

It is a good idea to create a new work folder for every new MESA calculation you do. To do this initially, you copy the model work folder from the MESA installation to a new directory with a name of your choice (in this example `project1`) inside your MESA directory on the local `scratch` disk. This is done via the recursive copy command invoked via `cp -R`. An example line is given below:

```

1 cp -R $MESA_DIR/star/work /scratch/$USER/MESA/project1

```

For each MESA project, create a separate work folder in this way. To run your new MESA project, first go to its directory. This is done via the change directory command invoked via `cd`. An example line is given below:

```

1 cd /scratch/$USER/MESA/project1

```

Inside this directory, you first need to assemble the code because MESA uses a compiled programming language called **Fortran**. This is done via a `make` command using the following line of code:

```

1 ./mk

```

Afterwards, the code can be run using the following line of code:

```
1 ./rn
```

You will see a bunch of text appearing in the terminal window, which is the output of MESA as it runs. Congratulations, you are now running MESA! There is no need to complete this simulation for now, so you can stop it by pressing **Ctrl + C** in the terminal window.

The **&** symbol at the end allows you to keep the editor window open while typing further Linux commands in the terminal.

For more on how to use MESA, you can find information here: [https://docs.mesastar.org/en/release-r22.05.1/using\\_mesa.html](https://docs.mesastar.org/en/release-r22.05.1/using_mesa.html). Furthermore, more information on simulating different stellar evolution problems in MESA can be found here: [https://docs.mesastar.org/en/release-r22.05.1/test\\_suite.html#](https://docs.mesastar.org/en/release-r22.05.1/test_suite.html#).

## 2.2 Saving your work

If you inspect your `project1` directory, you will see that MESA has created output files in both the `LOGS` and `photos` directories. Note that content on scratch older than 30 days **is automatically deleted every night**. Hence, when you have finished using MESA for the current session, you will need to migrate your project directories from the local `scratch` directory to prevent you from losing them. This can be done using the file manager and copying directories from `scratch` to your `home` directory, or via the terminal using the following example lines of code, which create a MESA directory in your `home` directory and copies your project there:

```
1 mkdir -p $HOME/MESA
2 rsync -avu --progress /scratch/$USER/MESA/project1/ $HOME/MESA/project1
```

The `rsync` command is preferred over `cp -R` because it only copies files that are new or have changed, which saves time when you have large output files. If you would like to know what the option flags `-avu --progress` mean, you can check the `rsync` manual by typing `man rsync` in the terminal.

Note that you will have to make a copy for *every* project directory you have worked on during the session! It is also important to note that you only have limited space in your `home` directory (about 7 Gb). Thus, it will be useful to migrate your MESA projects to a personal external storage device or to cloud storage (e.g. the OneDrive that comes with your Radboud e-mail, or a Google Drive).

### Pro Tip

Separating your code from your data is generally *very* good practice. We recommend to keep your code version controlled by making a `git` repository for your project folders, and to keep your data (i.e. the output files) separate from your code. You can do this by adding a `.gitignore` file to your project folder that ignores the `LOGS`, `photos`, and `png` directories, as well as any other large output files you do not want to track with `git`.

## 3 Evolving a 1 $M_{\odot}$ star

1. **Create a copy of the work folder** and give it a suitable name, then compile the code. For example:

```
1 cp -R $MESA_DIR/star/work /scratch/$USER/MESA/session1
2 cd /scratch/$USER/MESA/session1
3 ./mk
```

2. **Change the MESA model**, by editing the inlist files that live inside the project directory (e.g. `inlist_project` and `inlist_pgstar`). You can edit `inlist_project` (and other inlists like `inlist_pgstar`) using the file manager: navigate to **Other Locations** → **On This Computer** → **Computer** → **scratch**. Alternatively, open the file from the terminal with:

```
1 xdg-open inlist_project &
```

`xdg-open` command opens a file with the machine's preferred application, and the `&` symbol at the end allows you to run further Linux commands in the terminal.

- (a) Edit the `inlist_project` file to evolve a  $1 M_{\odot}$  model by setting:

```
1 initial_mass = 1 ! in Msun units
```

- (b) With the default settings, the model will start on the pre-main sequence and will stop when it reaches the zero-age main sequence (ZAMS), where hydrogen burning starts. Let's evolve the model a bit further than that, to the current age of the Sun: 4.6 Gyrs. Edit `inlist_project` and change the stopping conditions. First set:

```
1 stop_near_zams = .false.
```

and add the lines:

```
1 ! stop when the age of the Sun is reached
2 max_age = 4.6d9
```

*Tip:* to be able to see the plots after the run finishes, also add these lines to `inlist_project` in the `&star_job` section:

```
1 ! stop before terminating such that you can see the plots
2 pause_before_terminate = .true.
```

When adding new lines to your inlist files, make a habit out of also adding a comment (by using the `!`-sign) to explain what the parameter setting does. This will help you in the future.

3. Now compile again (`./mk`) and run the model (`./rn`).

- (a) First concentrate on the HR plot. You will notice the model (red circle) starts at the bottom of the plot and quickly disappears out of sight. To remedy this, change the limits of the plot window by editing the file `inlist_pgstar`. Good values for this model are  $\log L$  between  $-1$  and  $3$ , and  $\log T$  between  $3.5$  and  $4$ . Save the changes to `inlist_pgstar` and you should see the changes taking effect immediately!

#### Pro Tip

If you want to temporarily pause your model to have more time to look at a plot, you can suspend the job by pressing `Ctrl + Z`. To resume the model, type `fg` in the prompt. If you want to terminate your model, press `Ctrl + C`. If you want to terminate a paused model, first type `fg` in the terminal and then press `Ctrl + C`; this prevents the unclosable MESA plot windows from staying open and piling up.

- (b) After adjusting the HR plot, try to understand what happens. Can you explain the movement (i.e. the evolution history) of the star in the HR diagram? Is the star expanding or contracting? (remember the relation between  $L$ ,  $R$ , and  $T$  from the Stefan–Boltzmann law) And how do you explain this behaviour? (Note: ignore the

first horizontal part in the HR diagram where the star is still relaxing into hydrostatic equilibrium.)

- (c) The `TRho_Profile` plot shows the entire stellar model, from centre to surface, as it evolves. Can you identify the centre and the surface? What is the meaning of the other (dashed) lines in the plot? Does the ideal gas law hold everywhere in the star?
- (d) Colours indicate in which regions of the star convection (or some other mixing processes) occurs, and where nuclear energy is produced. Initially, almost the entire model is light blue, i.e. convective. Can you explain why? And how is this related to the star's location in the HR diagram?
- (e) At later times, the central portion of the star becomes green, i.e. radiative. At the same time the effective temperature in the HR diagram increases. Can you explain this? Consult section 9.1 in the lecture notes to answer these questions, if necessary.
- (f) When the model stops after 4.6 Gyrs, the star is undergoing hydrogen fusion in its core. How can you identify this from the plots? Also compare the location in the HR diagram to the current parameters of the Sun. Do they agree?

You will have noticed it takes several hundreds of steps to reach the main sequence (start of hydrogen fusion), while subsequently it takes much fewer steps to reach the current age of the Sun. The reason is that the structure of the star changes rapidly when the star contracts to the main sequence, so many small time steps are needed for MESA to resolve these changes. Once the star is on the main sequence, its structure hardly changes, so the code can take much longer time steps to evolve the star through the main sequence phase.

What would you have expected if the number of steps is proportional to the actual time the star takes to evolve? (i.e. compare to the appropriate stellar time scales.)

4. Now let's evolve the model further to the red giant branch phase. Remove the stopping condition for the age in `inlist_project` and instead add the lines:

```
1 ! stop when helium core reaches this limit
2 he_core_mass_limit = 0.25
```

Do not forget to remove (or comment out) all other stopping condition that we do not need! We can also examine some other PGSTAR plots. Edit `inlist_pgstar` and add:

```
1 Summary_Burn_win_flag = .true.
2 Abundance_win_flag = .true.
```

Now restart the model from a saved 'photo' from the previous run:

```
1 ./re
```

to start from the last saved photo in the `photos` directory. You can start from any saved photo inside your MESA model by running `./re x***`, where `***` are the final three digits of the photo of your choice.

- (a) Take some time to understand what is plotted in these two new panels, and identify each line in the plots. Note that the vertical scales are logarithmic in both panels, and that they represent many orders of magnitude, especially in `Summary_Burn`. You can correlate the  $T$  and  $\rho$  profiles in `Summary_Burn` with the curve plotted in the `TRho_profile` panel. The blue and green colours along the bottom of these panels correspond to those plotted in `TRho_profile`, and can be used to identify convective regions. Note that most of the change in density and temperature occurs in the outer few percent of mass of the star.

- (b) As the star evolves through the main sequence, the abundance profiles change. Which processes cause these abundance changes? Consult/refresh chapter 6.4 of the lecture notes if necessary. Can you explain all the changes from the nuclear reactions taking place? (Take into account the logarithmic scale: the changes in the bottom part of the plot look dramatic but only involve small values of the mass fraction.)
- (c) At what age (approximately) does the star reach the end of the main sequence? How can you tell from the PGSTAR panels? (Several plots show an indicator of this!)
- (d) Observe the changes in the  $T$  and  $\rho$  profiles as the star moves from the main sequence to the giant branch. Can you explain these changes? Does the core become degenerate, and if so, in which point in the evolution?
- (e) As the star evolves along the red giant branch, convection reaches deeper and deeper in the envelope of the star. Note how this affects the abundance profiles. Which elements/isotopes are most affected? Do their surface abundances increase or decrease? This process of convection changing the surface abundances is called ‘dredge-up’.

As the helium core mass increases, notice how the burning shell becomes thinner in mass. This requires MESA to take smaller time steps, so that it will take thousands of steps to reach the end of the RGB. The observable changes between each time step become smaller and smaller. That is why we stop the model at a core mass of  $0.25 M_{\odot}$ .

## 4 Exploring PGSTAR options

Many other details of the models can be shown using PGSTAR. Read the MESA website page [Using PGSTAR \(https://docs.mesastar.org/en/release-r22.05.1/using\\_mesa/using\\_pgstar.html#\)](https://docs.mesastar.org/en/release-r22.05.1/using_mesa/using_pgstar.html#) to get an idea of the possibilities, and experiment with different options. Simply run the same model again, or try a different stellar mass. (In the latter case, create a new work folder for your stellar models, as in step 1 above.)

In order to be able to easily inspect the plots created in real-time by the PGSTAR module *after* the end of the simulation you can opt to save every  $N$ 'th plot to a file in a separate directory. This can be done by adding several controls to the `inlist_pgstar` file, for example:

```
1 HR_file_flag = .true.
2 HR_file_dir = 'png'
3 HR_file_prefix = 'hr_'
4 HR_file_interval = 5
5 HR_file_width = 16
6 HR_file_aspect_ratio = 1
```

This will save a HR diagram every 5 steps into a directory `png` giving it a name starting with `hr_`. You can then combine a series of png images into a short movie using `ffmpeg`, for instance by executing the following in terminal when in the `png` directory (make sure this is 1 long line in the terminal):

```
1 ffmpeg -f image2 -pattern_type glob -framerate 5 -i "*.png" -filter:v
  scale=1680*1080 -preset:v slow -pix_fmt yuv420p -c:v libx264 -b:v 4M
  -f mp4 movie.mp4
```

Note that if you want to have several graphs in one png output file (e.g. not only the HR diagram but also `TRho_profile` or a Kippenhahn plot) then you can make use of the `Grid_*` controls of the PGSTAR module.

## References

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