Introduction to running MESA

If you can't stand reading anything that isn't on the web, skip this and go directly to https://docs.mesastar.org. Even if you do read this file, when you are done you should still go to that site!

These directions assume you have already installed MESA. Double check if you have set your paths correctly by running:

```
echo $MESA_DIR
echo $MESASDK_ROOT
```

and checking if they indeed point to your MESA installation.

Copy this work directory to somewhere outside the mesa directory tree and name it anything you like.

Compile by executing the `clean` and `mk` scripts:

```
./clean
./mk
```

This compiles the files in `src/`, links them against MESA, and produces the `star` executable file. In this first lab, you do not need to worry about what is in that `src/` folder, we took care of it ;)

By convention, run the program using the `rn` script:

```
./rn
```

When MESA runs, it first reads the `inlist` file. This file can point to other inlist files. Here, it points to `inlist_project` and `inlist_pgstar`.

You can control MESA by editing the options in the various sections of the inlist. The full set of parameters and their default values can be found in the defaults files listed in the inlists.

To restart MESA from a saved photo file, run the program using the `re` script:

```
./re [photo]
```

where `[photo]` is one of the saved snapshot files in `photos`. If no file is specified, MESA restarts from the most recent photo.

Monday MaxiLab 1: Modelling core overshooting in main-sequence stars

In this lab, you will learn how to set up a MESA model from scratch, learn how to monitor the run, customise its output and how to choose reasonable values for model parameters. Our science case is focused on the effects of core overshooting on the core of main-sequence stars.

As a maxilab, this lab will take two 1.5h sessions. Each session is organised in three sections, structured as below with an estimate of the time you are expected to spend on each. Please do not hesitate to ask your TA and/or the other people at your table for assistance if you notice you are falling behind on schedule.

SESSION 1

1. Setting up your MESA work directory [20']
2. Modifying the input physics and saving your final model [30']
3. Monitoring the run and customising output [40'] (move some stuff into session 2)

SESSION 2 4. Adapting the input parameters [10'] (and play more with the pgstar plots?) 5. Making your own plots [20'] 6. BONUS: automatically run multiple MESA models sequentially [20']

In this lab, we will examine how overshooting of a convective core affects a star's evolution together. In particular, we will examine how the numerical simplifications in this modelling may affect the model. In doing so, we will learn how to find reasonable values for model parameters, namely the initial mass of the star $M_\rm{ini}$, its initial metal mass fraction Z, the overshooting scheme, overshoot parameter $f_\rm{ov}$ which describes how far from the core the overshoot can reach and f_0 which describes how deep in the convection zones the model switches from mixing by convection to overshoot. To this end, everyone picks one set of parameter values to simulate, after which we will collect everyone's results and look for trends together. You will also learn how to navigate the

MESA output in greater detail and make fully custom plots in Python, both using the dedicated `mesa_reader` Python package as well as 'by hand'. If you have time left over, you can look into some prepared scripts that automatically write parameter values into your inlist and runs different sets of parameters sequentially.

SESSION 1

Setting up your MESA work directory

1. We will start from the mostly empty default MESA work directory and slowly build it up until we have a properly fleshed-out main-sequence model. Make a new empty directory somewhere on your machine, go into the empty directory and copy over the default MESA work directory:

   ```
   cp -r $MESA_DIR/star/work/ .
   ```

Do a quick `ls` to check what is included in this default work directory. You'll see a number of executables, namely *clean*, *mk*, *re* and *rn*. You can look at the summary at the start of this README to see what these executables do. The subdirectories *make* and *src* contain the Makefile and extra code to include, but you don't have to look into that today. For now, let's take a look at the inlists *inlist*, *inlist_pgstar* and *inlist_project*. These files describe what you want MESA to do. In particular MESA will always look for *inlist*. Using your favourite text editor, take a look at what is in *inlist*.

What this *inlist* essentially does is redirect MESA to the other two inlist files for all the real content, with *inlist_project* containing most of the fields describing how the MESA run should go and *inlist_pgstar* describing what visuals MESA should produce. For now, let's focus on *inlist_project*.

2. To start, let's run a very simple main-sequence model of a star with an initial mass of 5 solar masses and metallicity of 0.014 with some strong step-wise mixing due to core overshooting. To do so, open *inlist_project* and find and change the following parameters to the given values:

- `initial_m = 5d0`
- `initial_z = 0.014d0`

Next, to add the core overshooting, we need to add in some new fields. Before you try to do so, have a look at the questions below.

- `overshoot_zone_type(1) = 'burn_H'`
- `overshoot_zone_loc(1) = 'core'`

- `overshoot_bdy_loc(1) = 'top'`
- `overshoot_scheme(1) = 'step'`
- `overshoot_f(1) = 0.30`
- `overshoot_f0(1) = 0.005`

Question: The first three `overshoot_` fields describe where the overshooting should take place. Go into the MESA documentation and look up what each of these fields means. What other values are available? Meanwhile, `overshoot_scheme` describes what shape the overshoot mixing profile should take. Again, what are the alternatives to our 'step'?

▶ Show answer

Question: `overshoot_f` and `overshoot_f0` describe how large the overshooting region should be. How are they defined?

▶ Show answer

Question: Where should you add these fields?

▶ Show answer

Bonus Question: Why does each overshoot field in our example have that `(1)` at the end?

▶ Show answer

3. While looking around your *inlist_project*, you may have noticed the field called `ZBase` under the `&kap` namelist. This describes the reference metallicity used in the calculation of the opacities. For consistency, you should set `ZBase` to the same value as `initial_z`.

4. Before you run your model, you should consider when the model is terminated. Since we want to simulate the main-sequence evolution, we should place our stopping condition around the terminal age main-sequence (TAMS). Look under `! when to stop` in `&controls` of your *inlist_project*.

You'll note there are two conditions that can trigger the model to end. The first is designed to stop the model at the zero-age main-sequence (ZAMS), which we obviously do not want. Therefore, set

`stop_near_zams = .false.`

The second condition is meant to stop the model around the TAMS. Bear in mind that different people define the TAMS in different ways, so you should always think about how you want to define it.

Question: How does the default *inlist_project* define the TAMS?

▶ Show answer

5. Run your model by cleaning any executables in your work directory using

```
./clean
```

Then making a new executable using

```
./mk
```

and finally running that executable by

```
./rn
```

You'll notice that MESA writes a bunch of numbers describing your model to your terminal, followed by regular updates on some key parameters summarising the current state of your model. These are highly useful in examining how your run is going. For instance, keep an eye on the central hydrogen fraction (`H_cntr`) field, which will tell you how far along the main- sequence evolution your model is. You'll note that it initially changes extremely slowly. This is because MESA starts with a very small time step which gradually increases, as shown by the `lg_dt_yrs` field.

After a while, two panels pop, one showing the evolutionary track of your model on a Hertzsprung-Russell diagram and one the internal temperature and density profiles. Like the terminal output, these help you keep an eye on your model.

6. From all these numbers MESA wrote to your terminal, we've already identified a way to improve the efficiency of our models. The first 50 or so steps accomplish very little because the time steps are very small. We can tell MESA to start with a time step of one year, hence decreasing the required number of steps and speeding up the run. To do so, add the following to the `%star_job` section of *inlist_project*:

```
set_initial_dt = .true.
years_for_initial_dt = 1d0
```

Note that this will only take effect once the star has reached the ZAMS.

Moreover, you will have noticed that before the run truly stars, MESA first creates a pre-main-sequence model and lets it relax for 300 steps. For our purposes today, this relaxation is not critical, so let's reduce the number of relaxation steps to speed up the initialisation of our model to 100. To do so, add the following to your `%star_job`:

```
pre_ms_relax_num_steps = 100
```

Upgrading the inlist

7. *inlist_project* is currently mostly empty. This means that most settings are using MESA's default values. You should always check whether these are appropriate for your models. As an example, let's consider the composition of the model. In steps 2 and 3, you already set the metallicity mass fraction of the model. However, you also need to think about how that mass is distributed across different elements and isotopes.

Question: What is the default metal composition of MESA?

▶ Show hint

▶ Show answer

Say you have decided that you would like to use the more recent solar metal composition found by Asplund et al. (2009). Check the MESA documentation for how you can set that composition.

▶ Show hint

▶ Show answer

8. As you just altered the composition of your model, you should make sure you are using an appropriate opacity table, like you did by setting `ZBase` after changing `initial_z`. Navigate to the documentation of the kap module, which describes the opacities.

▶ Show hint

Look through the available opacity tables. Can you find the appropriate tables to be consistent with your metal compositions?

▶ Show hint

9. After deciding on how the metal mass fractions are, let's take a look at how the hydrogen and helium fractions are set.

Question: What is currently the initial value of the helium abundance in your model? How did MESA compute this initial value?

▶ Show hint

The Y–Z relation described in `initial_y`'s documentation is a linear one, with $Y = Y_p + \frac{dY}{dZ}Z$, wherein Y_p is the galaxy's primordial helium content. Aver et al. (2021) suggest the primordial helium abundance Y in the Galaxy is $Y_p = 0.2453$. Scaling to the solar composition, we find $\frac{dY}{dZ} = 2.193$.

Using these values, compute an appropriate initial helium fraction for your model and implement it in your *inlist_project*.

▶ Show hint

10. As this lab concerns fairly massive stars, mass loss by winds may play a considerable role. Check the documentation of `&controls` to see what implementations of mass loss are available. And what is the default mass loss?

▶ Show hint

▶ Show answer

You will see in the documentation that there is a wealth of wind mass loss schemes available, all of which can be scaled up or down. Each scheme is appropriate in particular regimes of the surface temperature, composition, etc. The so-called Dutch scheme attempts to merge some of these schemes into a cohesive whole. Add it into your *inlist_project* without scaling it down.

▶ Show hint

11. As this lab is concerned with the overshooting around a convective core, we naturally needs a good description of the convective zones as well. To that end, we would like for MESA to use the Ledoux criterion. Search through the documentation how to activate this criterion and add it into your *inlist_project*.

MESA uses the mixing-length theory (MLT) to describe the transport by convection. This theory relies on a scaling factor α_{MLT} which is in general quite poorly calibrated. As such, you should check what MESA's default value of this α_{MLT} parameter is.

When you are working on your real science cases, you should test a few different values for this α_{MLT} to gain an understanding of its effects. However, to save some time in this lab, we will stick to just one value, namely 1.8. Add this into your *inlist_project*.

12. To check if you made any mistakes, run your model again. You do not need to let it continue all the way to the TAMS, just check it does not crash. Plus, you should be able to use the terminal output to check if some of your changes are working as intended.

▶ Show hint

Despite how much you already added into your *inlist_project*, there are still many empty headers. Indeed, when building an inlist for your real science cases, you should still look into your atmosphere settings, equation of state tables, spatial and temporal resolution, and much more besides. However, for the sake of time and not making this lab too repetitive, we'll stop here and move on to adapting MESA's output and tracking the model's evolution using PGPLOT.

Customising output

13. Now let's turn to these animated plots, often called the pgstar plots. These are incredible useful in understanding what is going on in your model while its running, helping you spot potential problems early. Therefore, it is worthwhile to customise your pgstar panels to show those quantities that are the most important to your work. To this end, MESA has a bunch of prepared windows you can easily add by adding one flag to your *inlist_pgstar*. You can find these and how to edit your *inlist_pgstar* in [this documentation page]{https://docs.mesastar.org/en/24.08.1/reference/pgstar.html}.

For the purposes of this lab, we have prepared a specialised *inlist_pgstar* for you. Download that *inlist_pgstar* here [TO DO] and move it into your MESA work directory.

Run your model again to see what the new pgstar plots look like. You don't have to wait for the run to be finished. You can interupt it using ctrl+C if you're on Linux and Cmd+C if you're on Mac.

For some of you, this new panel may look terrible, either being very small or overflowing out of your screen. This is because the width of the pgstar window is dependent on your system and the size of your screen. If the panel is too large or small for you, open

inlist_pgstar, find the two lines shown below near the start of the inlist and play around with the values until it looks nice.

```
Grid1_win_width = 10
Grid1_win_aspect_ratio = 0.7
```

▶ Show hint

14. We have merged all the plots in one panel for a better overview. We also included some key quantities at the top, similar to MESA's terminal output. The plots are the HRD, a plot relating the star's age to the model number and another MESA default panel: the mixing panel. Finally, there is a mysterious mostly empty panel. We'll get back to that empty panel later.

For now, focus on that mixing panel.

Question: What is the panel showing exactly? What does the colour of each line indicate?

▶ Show answer

15. So far, so good! Now let's think about the age plot. This is an example of a history panel, where we plot two history quantities, i.e. quantities that vary over time. Since we will explore the effect of overshooting on the core in the second half of this lab, it would be more interesting to change the y-axis of the history panel to something more relevant such as the core mass. However, to do so, you first need some idea of what history quantities are available.

You may have noticed that since running your model, a new subdirectory named *LOGS* has appeared. That is the default directory to which MESA writes its output. In there, you will find files named *history.data* and *profile{i}.data*. These track how a number of quantities vary over time and the star's radius respectively.

So far, your history and profile output only contain the default columns. To see what history output is included, open *LOGS/history.data* with a text editor. You'll note a header on the first few lines describing some essential aspects of your run and then on line 6 the names of your history columns. The meaning of some columns will be mostly clear from the name, but some are a bit obtuse.

Both to find out what these columns mean and to include extra columns, we need to examine the file listing which history columns our model should output. First, copy the default history columns list to your work directory. Let's also give it a new name:

```
cp $MESA_DIR/star/defaults/history_columns.list my_history_columns.list
```

Open *my_history_columns.list* and take some time to scroll through the wealth of possible output MESA offers. Can you find the meaning of some of the columns you saw in your *history.data* file? You'll notice that some columns are not yet documented. Feel very free to ask the MESA developers present what they mean.

In this lab, we are interested in the effects of overshooting on the stellar core. The mass of the convective core is already included in the defaults. Now add the stellar radius and radius of the convective core to the history output. Look for and uncomment the appropriate fields.

▶ Show hint

▶ Show hint

▶ Show hint

Once you have uncommented the relevant lines, you need to tell your MESA inlist that you want to include the output columns in *my_history_columns.list* under `&star_job` :

```
history_columns_file = 'my_history_columns.list'
```

16. Now we can finally turn back to the pgstar history plot. Open up *inlist_pgstar* and navigate to the section where the history panel is defined. Change the left y-axis to the default quantity of the convective core mass.

```
History_Panels1_yaxis_name(1) = 'mass_conv_core'
```

Bonus: You could also use this panel to directly compare the two different definitions we might use. How would you go about that?

▶ Show hint

Bonus Question: Which of these two definitions seems more convenient to you? And what about the definition of the core radius?

What history and profile quantities we ask MESA to include not only changes the output of our model, but also impacts pgstar's plotting options. For instance, in order to plot a full Kippenhahn diagram, MESA needs to keep close track of all the mixing regions. In order to enable that, go into *my_history_columns.list* and find the option named `mixing_regions`. As you can read in its description in *my_history_columns.list*, this is not one quantity, but automatically adds a number of history columns. Try uncommenting this `mixing_regions` and give it a fairly large integer to ensure it includes all the relevant mixing zones, e.g.

```
mixing_regions 20
```

Your pgstar window should now include fully functional panels. Briefly run your model again to double check everything works as it should.

17. The history files tell you how chosen quantities vary over time. But what about quantities that vary over the star's radius? Those are described by the files *profile{i}.data* in the LOGS folder. As with the history, let's check study what is included by default and add a few columns. First copy over the list file:

```
cp $MESA_DIR/star/defaults/profile_columns.list my_profile_columns.list
```

Again, take a few minutes to check out what it has to offer. Later in this lab, we will examine the mixing profile at different times. To this end, you will need the profile of the (logarithmic) diffusive mixing coefficient and some way to tell what process is causing that mixing. Find and uncomment some appropriate fields.

▶ Show hint

Remember to add your profile column list to your inlist:

```
profile_columns_file = 'my_profile_columns.list'
```

Bonus Question: How often does MESA produce a profile file? How could you increase this resolution?

▶ Show answer

18. In the other labs today, you will learn how to run models that continue after the main-sequence evolution. When doing so, it is quite annoying to have to simulate the main-sequence again every time you tweak something in your inlist. Instead, we can tell MESA to save a model at the end of a main-sequence run so we can load that model in next lab. Add this to your `%star_job` and name your model:

```
save_model_when_terminate = .true.
save_photo_when_terminate = .true.
! Give a name to the model file to be saved including your parameter value
! 'M{your_M}_Z{your_Z}_fov{your_f_overshoot}_f0ov{your_f0_overshoot}.mod'
save_model_filename = ! Add your name here
```

19. Now let's run the model all the way to the end. As the model runs, keep an eye on your new mixing panel in particular. Compare it to those of the other people at your table.

SESSION 2

Trying different the overshoot parameters

You now know how to navigate your work directory and build up a main-sequence model. That's great. However, so far we have limited ourselves to simply adding in pre-chosen parameter values, choices of tables etc. In real scientific applications, you should always consider the impact of these settings, for instance by trying a few different values. In particular, there are a number numerical schemes and poorly calibrated physical parameters for which you should think carefully about the appropriate value. You already encountered some of these today, namely the mixing length parameter α_{MLT} and the mixing by overshooting.

In this session, we'll explore the impact of overshooting in your model. Through your experiments and the lecturer's discussion of everyone's result, you will learn how you can find reasonable values and settings for overshooting in your model. The plan is that everyone gets a unique set of overshooting parameters, initial mass and initial metallicity to try out. You will then compare the results of these parameter settings to the model you produced in lab 1. Meanwhile, we will collect some basic results from everyone's model and examine the correlations between different parameters together.

20. Go into this spreadsheet and put your name next to one set of parameters to claim it as yours. Modify your inlist accordingly.

If you selected the **'no overshoot'** scheme from the spreadsheet, you should leave the overshoot scheme as an empty string, i.e.

```
overshoot_scheme(1) = ''
```

21. Before you run your model again, you should make sure you are not overwriting your previous results. To do so, you should first adapt `save_model_filename`, ideally with some new name that reflects the new parameter set.

Next, to not overwrite your history and profile data, you could tell MESA to write the history and profile data to differently named files. However, there is another, easier option, which is to simply tell MESA to save the output in another directory than *LOGS/*. Check the documentation or user forums to discover how you can do that. Like the final model name, it is generally recommended to use a name that reflects the settings of your model, rather than something generic such as *model2*.
Here's the cleaned-up, collapsible version of that double hint block:

▶ Show hint

▶ Show hint

22. Now run your model again. Keep a close eye on your pgstar plots, particularly the mixing panel. Compare it with those of the other people at your table.

23. Using your favourite text editor, open the history.data file and find the line describing the TAMS. Add the values of the following parameters to the second page of the spreadsheet. Take care to check your units!

- log(Teff)
- log(L)
- core mass
- core radius
- age in Myr

MATHIJS TO TEAM: What output would be most useful? Teff and L are no- brainers and the core conditions are relevant as well. What else?

24. Now let's wrap up this lab by reading your MESA output in using Python and making some custom plots.

MATHIJS TO TEAM: What kind of plots should we have them make? One idea is to make them plot the mixing profile of every profile together in one plot. Another obvious option is to plot core mass/radius against (a proxy of) time.

BONUS: Batch Parameter Studies with MESA

If you've completed the main lab activities and have time remaining, explore the automated parameter study framework in the `batch_runs/` directory. This framework enables systematic exploration of overshooting effects across multiple stellar models.

Why Run Batch Studies?

1. **Efficiency**: Run dozens of MESA models without manual intervention
2. **Thoroughness**: Test how overshooting parameters affect stellar evolution across different masses and metallicities
3. **Reproducibility**: Generate standardized output for consistent analysis

Quick Start Guide

```
# Generate parameter-specific inlists
python batch_runs/make_batch.py batch_runs/MESA_Lab.csv

# Execute models sequentially
python batch_runs/run_batch.py

# Visualize results across parameter space
python batch_runs/plot_hr.py
python batch_runs/plot_ccore_mass.py
```

Available Parameter Space

The provided parameter grid explores:

- Stellar masses: 2, 5, 15, 30 M☉
- Metallicities: Z = 0.014, 0.0014
- Overshooting schemes: None, Exponential, Step
- Overshooting parameters: 0.01-0.3
- Penetration depths: 0.001-0.01

For complete documentation and additional analysis tools, see `batch_runs/README.md`.