Fun with MESA

J. Jackiewicz

February 5, 2024

1 Introduction

1.1 What is MESA?

Modules for Experiments in Stellar Astrophysics (`MESA`[1]) is an open source software package (written mostly in FORTRAN) that computes the physics of stellar interiors, from nuclear reaction networks to diffusion and convection. The most useful aspect of it for our purposes will be the module `MESA star`, which is a state-of-the-art one-dimensional stellar evolution solver. To get some familiarity, try to read over the 5 "instrument papers" for `MESA` (or at least the first two), which are collected in an ADS library[2] for your convenience (there are some other papers in there too). Start with first ApJS one: Paxton+2011.

1.2 Where to find MESA

You will be using `MESA` in class to help you understand stellar structure and evolution. Version r15140 has been installed on a local server. This is NOT the most current release, but it will suit us just fine if you choose to use it. All you need to do is copy a "working directory" to the location on some disk on the network where you would like to run it (more later).

So you can copy this directory right from where they live in my filesystem on the html server at:[3] `/home/httpd/html/jasonj/MESA/WORK_DIR/`. If you are on one of the department machines, go somehwere and type:

```
>cp -r /home/httpd/html/jasonj/MESA/WORK_DIR/   dirname
```

where dirname is any name you choose (for example, homework1 or whatever).

Alternatively you can just go directly to the course website and download the folder (or tar.gz file) each time you start a new run from there.[4]

1.2.1 Relevant files

The `MESA` working directory stucture is set up so that every directory has a replica of all the necessary codes needed to run a model, or links to such codes. In your working directory the only thing you will likely be editing are the "inlists;" `inlist_project` and `inlist_pgstar`. These are both called by `inlist`, and it is set up in such a way that you can have multiple inlists. Also present in the working directory are make commands, clean commands, and run commands. In src/ you have the main fortran code that calls the modules in the mesa/star directory. The `LOGS/` and `photos/` directories are where the outputs are stored (more later).

[1] https://docs.mesastar.org/
[2] https://ui.adsabs.harvard.edu/public-libraries/GNAgYNMhRiOegMbBCXsdfg
[3] http://astronomy.nmsu.edu/jasonj/MESA/
[4] http://astronomy.nmsu.edu/jasonj/MESA/

The `inlist_project` file contains the input parameters to the model. There are many, many parameters you can change from their default values, such as the equation of state prescription, initial model mass, data output frequency, etc. The `inlist_pgstar` file describes how output gets plotted as the code is running (`MESA` uses software called pgplot). There are only 3 "sections" of commands, called "namelists", that do all of this control: (1) `&star_job` lets you set things like the source directory, starting model, if it should plot, etc, you won't do much in here most of the time; (2) `&controls` is where you change the initial mass, output directory and filenames, when to stop, abundances, etc.; (3) `&pgstar`, found in the `inlist_pgstar` file, lets you have precise control over the plotting windows that pop up, by changing axis limits or window size. Commands here can be changed while the code is running for realtime adjustments.

The parameters for these 3 namelists have many default values, as mentioned above. They are listed in the 3 files `contols.defaults`, `star_job.defaults`, and `pgstar.defaults`, which you can find in the `DEFAULTS/` directory in my top-level `MESA` location[5]. If you want to change the default values for a particular quantity for a run, you can do that in two different ways. You can add the parameter and value in the inlist files under the appropriate namelist section. Or, you can copy one of the defaults files into your working directory, and change the parameter in the file. This will override the values found in the `MESA` source directory (which you cannot edit). Obviously, the first option is more straightforward, but you should definitely look through the .defaults files to get an idea of what the code has available.

As `MESA` runs, it saves output. There are two types of output, a "history" of the evolution, i.e., gross stellar properties (mass, luminosity, $T_{\rm eff}$, etc., that you can control) as a function of time. Then, at certain time steps, you can save the structure "profile" of the 1D model parameters, such as $m(r)$, $\rho(r)$, gradients of elements, etc. Not all quantities that are computed are saved as functions of time or radius. If you look in your `inlist_project` under the `&star_job` namelist, you'll find the names of custom files for the history and profile output. In your working directory, you can find these files and can see what will be saved. All of the possibilities are given in the `DEFAULTS/history_columns.list` and `DEFAULTS/profile_columns.list` files. To change what is saved, again you have 2 basic options. If you would like to add or remove quantities, you can simply edit these "custom" files. Or, you can use the .list files as default or copy them into your directory and edit them directly. Note that with the default settings, not many profile variables are actually saved, and you'll definitely be modifying this. Note: editing the .list files inside the `DEFAULTS` directory will not do anything ... `MESA` does not see that path.

1.3 Running MESA

Now that you have a working directory and a basic understanding of `MESA`, it is easy to get it running, and there are only a few other things you need to know.

1.3.1 Machines

`MESA` is installed on our main astronomy server, which all of our network machines can "see." Log into a machine, such as your desktop computer, or virgo, praesepe, hyades, astrogpu, etc. Open a terminal (using tcsh is required) and go to a location where you can copy a working directory. At the prompt, type:

```
> init_mesa
```

which will point your environment to where `MESA` is installed. It also shows you how many threads you will be using, based on the machine. You can change that number manually (`setenv OMP_NUM_THREADS 10`, for example). Please note that in that particular terminal you are using, once you type the initialization command some of the compilers and things will be different than the ones you might normally use. For example, the fortran compiler `MESA` is using might be more recent than the one installed on your machine. All you need to do is open a different terminal and type `> tcsh` to use the default one again.

[5]again, just go here to find it: http://astronomy.nmsu.edu/jasonj/565

If you're off campus, you can ssh into one of our machines and everything explained above still pertains. HOWEVER, to do so you now need to be connected to the NMSU VPN.

It's probably not a great idea to run from the astronomy.nmsu.edu server (it's slow and not meant for computations), nor will it be a good idea to save your output under /home/users/, since space there is limited.

You may also try to download, install, and get it running on your personal computer/laptop if you wish. If that works, fine. If it doesn't, I can give limited guidance. There's plenty of help on the `MESA` website[6] to accomplish this.

1.3.2 Initial test

`MESA` is really easy to run and now you're ready. Copy a fresh working directory if you need to and give it a name. Don't change anything in `inlist_project` or `inlist_pgstar` just yet (but have a look at it). You shouldn't have any object files in your fresh directory, but just for fun run

```
> ./clean
```

This is a good habit. Now you want to run

```
> ./mk
```

which will compile your code. If there is an error here it's probably because of your environment variables not being set properly.

Now simply type

```
> ./rn
```

and away it should go. Early on you should notice if PGplot is working as you will see a window with various plots like an HR diagram. If this happens you are set. You can kill the job any time (control c).

I now recommend learning a few of the plotting controls. Start a fresh run and change the controls to create a pre-main sequence star. Do a $1\,M_\odot$ model. Remove any stopping criteria so it runs for a while. Then read through the "hands-on tutorial" in the `DEFAULTS/pgstar.README` file,[7] which describes how you can change plotting features on-the-fly. You'll see the power of pgplot as you go through this tutorial.

1.3.3 Output

Output is saved in the `LOGS/` directory, as well as in the `photos/` directory (whose files are only used to restart a run from a model). The `history.data` file contains the basic stellar parameters with time, and the `profile_n.data` file contains the 1D model profiles (with depth) for model #n. The inlist file allows you to control *how often* you save models and photos. Saving the history at every model is useful, but saving the profile at every model step is not, since those files get big. Try not to take up too much disk space with your output.

The key thing is having software that can parse these output files and give you what you need. They are just text files, but have a complicated structure. There are already some community contributed codes in certain languages that you may be able to use. Look here,[8] I know there are `MATLAB`, `Python`, and `IDL` codes to do this, at least. Otherwise you may have to write your own, I'm not sure exactly how the output .ascii files are structured, and I haven't seen this documented anywhere, but haven't really looked too hard either. Most `Python` users can use `mesa_reader`.[9]

[6] `https://docs.mesastar.org/`
[7] also online here: `https://docs.mesastar.org/en/latest/using_mesa/using_pgstar.html#hands-on-tutorial`
[8] `https://cococubed.com/mesa_market/add-ons.html`
[9] `https://docs.mesastar.org/en/latest/using_mesa/output.html#what-is-mesa-reader`

1.3.4 More tests

The `MESA` people have provided a wide range of interesting test suites.[10] To see a list of them that are installed at the top level,

```
> ls $MESA_DIR/star/test_suite/
```

If you'd like to run one of them, copy that directory to your local machine as before. After making a few tiny path edits to the files in the directory, as described on the help page linked above, you are ready to go. These are very instructive.

1.3.5 Getting help

- Visit the sourceforge site[11] for any general problems you have initially. It describes all the parameters and capabilities.
- Most of the documentation is migrating to docs.mesastar.org. For example, if you want to see what all the default setting are and all the things that can be changed in your inlist, see the defaults.
- The "Mesa Marketplace"[12] has tools for you, such as video explanations, plotting tools in most software languages, a blog, and inlists of published papers that used `MESA`.
- Come seee me.
- Read the ApJS papers for the physics!

2 First assignment: a model of the Sun

Computer problem 0: You are going to make a roughly calibrated model of the Sun ($1\ M_\odot$) that you can use for other purposes if ever needed. It's always useful to have interior solar profiles lying around.

1. Copy a fresh working directory for yourself.
2. Make sure a pre-main sequence model is created for your Sun (i.e., you are not starting from a saved starting model).
3. Set the initial metallicity mass fraction $Z = 0.02$ and initial helium $Y = 0.25$ (this is roughly the primordial He fraction).
4. Add the appropriate stopping condition, to run to an age of exactly 4.57 Gyr (this will go in the &controls namelist).
5. Make a plot of an H-R diagram of the star's track (use the history file that's saved). Also plot the Sun (from known values) as a symbol on your H-R diagram. If your track does not "end" very near the observed solar values (it shouldn't), continue below.
6. How does the model star's radius and luminosity compare to the values we estimate for the real Sun? Compute the relative error of each quantity as $(x_{\rm model} - x_\odot)/x_\odot \times 100$.

[10] https://docs.mesastar.org/en/r15140/test_suite.html
[11] http://mesa.sourceforge.net/
[12] https://cococubed.com/mesa_market/index.html

7. What can be "tuned" to make the star more Sun-like? We want R and L to agree within about 2% of the Sun. Try some modification in the controls, and rerun to get a model a bit closer to the Sun. What did you do? What did you get? Make one modification to get the radius and/or luminosity a bit closer and show your recomputed relative difference. Then overplot the new evolution track on top of your original one.

What to submit in Canvas (20 pts):

1. An H-R diagram (**properly plotted** with labels and using standard conventions) containing the two evolutionary tracks of your runs, as well as a point symbol indicating the position of the current Sun. **Makes sure your plots are sufficiently zoomed into the area around where the models stopped so we can see what's happening** . . . we don't really care right now about the pre-main sequence evolutionary history when the Sun was orders of magnitudes brighter.
2. Your relative percentage differences of the radius and luminosity in each run.
3. A brief description of what you did and the inputs you used in each case.
4. A brief conclusion about what effects the changes you made to the starting parameters have on the model over the course of the evolutionary period. This does not have to be detailed at all and can be just pointing out the obvious.

Upload files and enter text (or upload a document) directly in Canvas. You can certainly make a latex document with everything embedded in it if you choose.

Make sure not to overwrite either your "history" file for each run, or your final "profile" file. You can either rename the directories after each run, or, make a modification in the inlist that uses a different base name, or, use a whole new working directory. I tend to rename the directories. For example, after a run, I will do

```
> mv LOGS/ LOGS_RUN1/
```

or something more descriptive. I might copy my inlists in there too so I know what parameters I used. However, it's easy to forget to do this.

The final "profile" of your last run should be a decently calibrated solar model. You should save it for future use. You might even consider setting the `save_model_when_terminate = .true.` and `save_model_filename = 'solar_model.mod'` controls, for example. If you do this, make sure you save any relevant interior variables that you might need. We will probably "add" to your inlist throughout the semester to make the model even more "Sun like" as we learn about other physical details.