KU Leuven – Binary Stars
B-KUL-G0155A-2324
Department of Physics & Astronomy
Celestijnenlaan 200D

MESA: the binary module

Reinhold Willcox, Kunal Deshmukh, & Annachiara Picco

November 21, 2023

Introduction to MESA

MESA is a widely-used, state-of-the-art, 1D detailed stellar evolution code. Following years of extensive development, MESA can now model interactions in binary stars: mass and angular momentum transfer, tidal effects, circularization, etc.

If you have not already, you can download MESA, install the SDK, and set your environment variables. The instructions for this can be found here: https://docs.mesastar.org/en/latest/installation.html. To run MESA in the computer lab or remotely via `ssh`, instructions can be found on the next page.

Additionally, you will want to download or clone the github repo associated to this tutorial, which can be found here: https://github.com/reinhold-willcox/binary_stars_MESA_module_KUL_2023.

Useful resources

- Paxton B., et al. (2015) "Modules for Experiments in Stellar Astrophysics (MESA): binaries, pulsations, and explosions"
- Online documentation: https://docs.mesastar.org/en/r15140/reference.html
- MESA-users email list: https://lists.mesastar.org/pipermail/mesa-users/
- Python plotting tool: https://github.com/wmwolf/py_mesa_reader

Key Vocabulary

- **Inlists** FORTRAN input files that contain value definitions for all of the parameters of your run. Examples: `inlist` `inlist_donor` `inlist_pgstar`
- **Executable Binary** This is the program that is built by the compiler and runs your simulation. It is akin to a `.exe` file in Windows.

Running MESA on IvS Computers

The lab computers at IvS have multiple versions of MESA installed and we need to make sure we pick the latest version. To do so, run the following commands:

```
$ ml mesa/r23.05.1
$ echo $MESA_DIR
```

The first command loads the MESA r23.05.1 version and the second command is to verify the same. You can now start using MESA as instructed in following sections.

Note: If you are planning to run it remotely using `ssh`, make sure to add `-X` in the `ssh` command. This is required to show output plots from the remote computer on your screen. For example, your command should be:

```
$ ssh -X <username>@...
```

Running a Single Star

We will start by copying the basic `work` template from the MESA module. This contains the source code to compile and run MESA. Next we will copy into this folder the custom `inlists` we have provided for you. From the project root, run:

```
$ cp -R $MESA_DIR/star/work SINGLE
$ cp -R single_star/* SINGLE/
```

The second line is the standard method to initialize any new project. The last line replaces the pre-included `inlists` with those provided by us in `path_with_input_files`.

Default values

Many of the default values for the input parameters are contained throughout the MESA source code as '.default' files. To see a list of these, we'll use the 'find' command.

```
$ find $MESA_DIR/* -name ‘‘*.defaults’’
```

There are many more here than we will need for this tutorial. But it is useful to have these in your working directory, and it doesn't hurt to copy them all over. We can do this in a single line to save time. From the project root, run:

```
$ cp $(find $MESA_DIR/* -name ‘‘*.defaults’’) .
```

We can now inspect the provided inlists next to the default files. This will give us a sense of what parameters are available, what choices we have for them, and what the defaults are. More information can of course be found in the online documentation, but this is a good place to start.

Compiling and running MESA

MESA is written in FORTRAN, which is a compiled language. That means we need to compile the source code, which is human readable, into a cleaned up, optimized, machine readable binary. We run:

```
$ ./clean && ./mk
$ ./rn
```

The first command `./clean` gets rid of previously compiled binaries, and the second `./mk` compiles the source code. The double ampersand `&&` means that the second command is run only if the first command runs successfully. The second line `./rn` starts the simulation. If compilation works correctly, you should see a splash screen explaining some of the MESA input and version, and what variables will be contained in the output.

After a moment, you should see the state of the star following its evolutionary timesteps. You can set which variables are shown, which are saved to a file, and how often these printed and saved timesteps are reported (e.g. if you only want every 10 timesteps to be shown to the output). More steps printed / saved means more I/O, and thus a slower code. You should also see a popup window showing various plots that track the evolution of the star.

1. **Inspecting the plots**
   (a) What are each of the plots showing? Which ones can you name?
   (b) What do each of the raw values in the table at the top represent?
   (c) Can you explain what each curve represents, and why it has the shape that it does?
   (d) What phases of the stellar evolution can you identify?

2. **Varying the initial conditions**
   (a) How would these plots look different if we varied the initial mass? How do we do that?
   (b) What about initial metallicity? What are reasonable bounds on the metallicity?
   (c) What about initial rotation? What are reasonable bounds on the rotation? Given that the escape velocity of Earth is ~ 10 km/s and $R \sim M^{0.7}$ at ZAMS, can you derive a rough relation for the maximal rotation of stars as a function of mass on the MS?
   (d) How can we modify the plots without stopping the simulation?
   (e) How can we keep the plots from closing when the evolution finishes?
   (f) How do we change the stopping criteria for the simulation?

Running a Binary Star

We will start similarly to how we started with the Single star. Generally, you want to keep all the relevant `inlists` for each project in their own project directory.

This time, we will not look at the inlists before running MESA. As before, we start from the project root, and run:

```
$ cp -R $MESA_DIR/binary/work BINARY
$ cp -R binary_star/* BINARY/
$ cd BINARY/
$ ./clean && ./mk
$ ./rn
```

1. **Inspecting the plots**
   (a) What is different about the binary plots compared to the single star plots? What file(s) contain information that controls how the binary plotter looks?
   (b) Star 2 is not simulated. Why might this be beneficial? What type of binary is this?
   (c) How can you tell when mass transfer starts?

2. **Varying the initial conditions**
   (a) How would we run a binary with two MS stars?
   (b) How do you control accretion efficiency and angular momentum loss during mass transfer? What do these parameters mean? (Hint: see Soberman et al. 1997)

3. **MESA use cases**
   (a) Why is MESA not ideal for studying the formation channels of binary black holes?
   (b) What are some stellar systems or physics that MESA might be better suited for?

4. **Can you reproduce these same plots in python using the `pyMESA` package?**

A final tip for the report: be sure to have a strong understanding of the expected single stellar evolution theory pertaining your donor (and accretor) ZAMS: you will have to comment on the features that you can expect from single evolution, as compared to the ones which are exclusively due to binary evolution. In this sense, feel free to use this document as a guideline to comment on your system, *when applicable*.

The provided inlists for this session are a bit of involved: try to read them carefully and understand as much as you can. They are optimized to run **as fast as possible**, in the context of this brief lecture, so you may want to comment a lot of these lines and try to balance a larger computing time at home with a better resolution. They might also be not that suitable for your assigned binary system! Always try with the standard work directory's inlists first, and see what options work best for you. If you have questions, check the `*.defaults` files, or the MESA documentation, ask an expert (there are many at KU Leuven!), or contact one of the TAs:
reinhold.willcox@kuleuven.be
kunalprashant.deshmukh@kuleuven.be

Practice problems for binary stars

This next bit contains practice problems: try answering the questions in your free time, by yourself or (better!) discussing with a classmate.

For each of the following investigations, create a new directory identically to how you created the `BINARY` directory, but renamed appropriately.

1. **CASE A**
   DEF The donor star fills its Roche lobe during its main sequence evolution, that is, during core H-burning. These are the shortest period systems: they evolve through a rapid, $\tau \sim \tau_{\rm KH}$, phase of mass transfer, followed by a slow phase on the much longer nuclear timescale, $\tau \sim \tau_{\rm nuc}$.

   (a) **Evolve** the model binary from the input files in your `CASE_A` directory. Set the following initial conditions:

   $$\mathrm{M}_1 = 10\,\mathrm{M}_\odot, \quad \mathrm{M}_2 = 9\,\mathrm{M}_\odot, \quad \mathrm{P}_{\rm orb} = 2.5\ \mathrm{days}\,. \tag{1}$$

   [Run of $\sim$ 12 min].

   (b) **Look** attentively at the PGstar windows and:

      i. Recognize the first mass transfer episode. Why is it reasonable to make the assumption of conservativeness? Can you explain the behavior of $\mathrm{P}_{\rm orb}$? Why the H surface mass fraction is not changing much in this phase?

      ii. Another mass transfer episode starts after the TAMS, but prior to the core He-burning ignition: this is sometimes called *case AB*. What is its feature in the HR diagram?

iii. Comment on the H surface mass fraction of the donor *after* the two mass transfer episodes. What kind of object is it? Would you expect this outcome from single stellar evolution theory?

iv. Examine the Kippenhahn diagram of the accretor. What happens to the convective core during mass transfer? And what is going on with the thermohaline-mixed zones (light violet)? What is its main consequence for the accretor?

(c) **After** the run has finished, find in your `grid_png` and `png2` folders a snapshot capturing the end of the first, case A, mass transfer (look at the ages). Recalling the L − M relationship for massive stars in MS,

$$\textsc{Mass-Luminosity Relation} \qquad \frac{\mathrm{L}}{\mathrm{L}_\odot} = \left(\frac{\mathrm{M}}{\mathrm{M}_\odot}\right)^{3.5} ,$$

check if the accretor (which is still on the MS, right?) has an appropriate L for its increased mass. Look also at the donor's L: is it appropriate for an object with such a reduced mass?

If you arrived here, good job: you just have solved, through stellar evolution theory, the ***ALGOL paradox***: a system in which the more massive component seems to be the less evolved object, as compared to the less massive one.

2. **CASE B**

DEF The donor fills its Roche lobe after leaving the main sequence, but before core helium ignition. The mass transfer is mainly driven by the rapid, $\tau \sim \tau_{\mathrm{KH}}$, expansion of the donor star as it is crossing the Hertzsprung gap. Setup another directory, `CASE_B`.

(a) **Estimate** the minimum value, $\mathrm{P}_{\mathrm{orb}}^{\mathrm{min}}$, of the orbital period for the system of Eq. (2) to undergo case B mass transfer. To do this, remember the following:

$$\textsc{Kepler's III law} \qquad \frac{4\pi^2 a^3}{\mathrm{P}_{\mathrm{orb}}^2} = G\left(\mathrm{M}_1 + \mathrm{M}_2\right) ,$$

$$\textsc{Eggleton's fit} \qquad \frac{\mathrm{R}_\mathrm{L}}{a} = \frac{0.49 q^{2/3}}{0.6 q^{2/3} + \log\left(1 + q^{2/3}\right)} ,$$

and you can assume that, at TAMS, your model star with ZAMS $\mathrm{M}_1 = 10\,\mathrm{M}_\odot$ has $\mathrm{R}_{1,\,\mathrm{TAMS}} \simeq 40\,\mathrm{R}_\odot$. Change to the desired value the parameter

```
initial_period_in_days = 2.5d0        !days
```

(b) **Change** your `inlist_project` to model just one star, i.e. the donor; search for

```
evolve_both_stars = .true.
```

MESAbinary will evolve just the donor star, treating the companion as a *compact object* ∼ point source, accreting material in a HMXRB-like system. For practical purposes, these systems are computationally less expensive, so well suited for this short training session. Another trick to speed up the computing is to lower the initial metallicity Z of your models. Find the following lines in your `inlist_donor` and uncomment them:

```
!  zams_filename = './zams_z2m3_y244.data'
!  initial_z = 0.002
```

You are simply telling MESAbinary to use a new set of ZAMS to interpolate for the PMS models, and this set has lowered Z. (Question: how much lower?)

(c) **Evolve** the model massive binary with your renewed initial conditions:

$$\mathrm{M}_1 = 10\,\mathrm{M}_\odot, \quad \mathrm{M}_2 = 9\,\mathrm{M}_\odot, \quad \mathrm{P}_{\mathrm{orb}} = ?$$

[Run of ~ 5 min].

(d) **Look** attentively at the PGstar window and:

i. Recognize the first mass transfer episode (ignore the spike at $\tau \sim 2.5 \times 10^7$ yr: this is just numerical). Can you motivate its feature in the HR diagram? And how is the mass transfer rate, as compared to the CASE A?

ii. Convince yourself that such a strong mass transfer rate leaves behind an almost naked He core. Also, how much mass is accreted by the compact companion? Compute the efficiency of mass transfer as $\eta = \Delta \mathrm{M}_{\mathrm{CO}}/\Delta \mathrm{M}_{\mathrm{donor}}$.

(e) **After** the run has finished,

i. Speculate on the final fate of the *donor*: assume that your star explodes directly in a core collapse SN, right after He depletion. Which is the most appropriate observational class (II, Ib, Ic) of this SN? Is it reasonable to make such an assumption or do you think that there are some big uncertainties in the remaining lifetime of your star?

ii. Speculate on the final fate of the *system*: as you saw in class, for symmetric explosions the system unbinds if

$$\Delta \mathrm{M} \geq \frac{\mathrm{M}_1 + \mathrm{M}_2}{2}\,;$$

while for asymmetric explosions, a NS is expected to receive a **natal kick** $\simeq 10^2\,\mathrm{km\,s^{-1}}$. Assuming that the SN leaves a standard NS with $\mathrm{M} \simeq 1.4\,\mathrm{M}_\odot$, what happens for the case without a kick? And what changes with the kick (look at the lecture slides, or think about the observational counterparts of NS+CO...)?

iii. Speculate on the final fate of the *accretor*: if the natal kick succeeds in disrupting the system, the accretor is likely to be running away with a linear velocity that is approximately equal to the orbital velocity $\Omega = 2\pi/\mathrm{P}_{\mathrm{orb}}$ it had prior to the explosion. Unbound stars with velocities faster than 30 kms^{-1} are often referred to as ***runaway stars***. Have you found one?

iv. (optional) We have ignored that the companion is modeled as a point source, namely MESAbinary is modeling an actual black hole!
A BH cannot accrete at infinite rates, its accretion is (we think!) limited by the Eddington accretion rate; also, it can accrete material when the stars are detached and mass transfer is not ongoing.
You can add an option to your `inlist_project` which includes the physics:

```
limit_retention_by_mdot_edd = .true.
use_radiation_corrected_transfer_rate = .false.
do_wind_mass_transfer_1 = .true.
max_wind_transfer_fraction_1 = 0.1d0
```

Look for these options in your `binary_controls.defaults`. Now, run your simulation again and see what changes; look at the efficiency of mass transfer in particular.

3. **A LOW MASS Binary**

We have previously commented about the features of the binary evolution of a High Mass Binary. Now we are modeling a binary system composed of lower masses, so that you can familiarize with typical orders of magnitude and look at new features. Also, this time we are including a small initial eccentricity and a circularization scheme: this is useful for your report.

(a) **Evolve** the model binary from the input files in your `outcomes` directory. You are evolving a low mass binary with the following initial conditions:

$$\mathrm{M_1} = 1\,\mathrm{M_\odot}, \quad \mathrm{M_2} = 0.8\,\mathrm{M_\odot}, \quad \mathrm{P_{orb}} = 0.4\ \mathrm{days}\,. \tag{2}$$

[Run of ~ 17 min].
First, look at the terminal output and try to see what happens to your initial eccentricity `e`. Does your orbit quickly circularize?

(b) **Look** attentively at the PGstar window for the donor and:

i. Recognize the mass transfer episode. As you can see, it happens during the donor's MS: which case is it? On which timescale does it proceed? (*Hint*: look at the $\log \dot{\mathrm{M}}$−Age (yr) plot, and recall that you are dealing with a Sun-like donor).

ii. Look at the Equation of State $\log T - \log \rho$ diagram and recognize the region in which the core is entering. Do you expect that this will influence the later evolution?

iii. The mass transfer episode terminates very close to a well known feature in the HR diagram. What is it? And what is happening in the Kippenhahn diagram in the meantime?

iv. Acknowledge how deeply the donor has been stripped by the mass transfer episode, i.e. look at the mass M_1 after this phase. Find in your window the information about the mass M_{He} of the He core and compare the two.

(c) **After** the run has finished, find in your `grid_png` the snapshots capturing the ending phases of the run.

i. What is the donor star doing after the end of mass transfer? You can recognize a **big** excursion towards the blue and then a sudden drop of luminosity. Can you explain these?

ii. What is this kind of final object? Would you expect this outcome from single stellar evolution theory? (*Hint*: consider the surface properties...)

Now we are going to see two different situations that you already encountered in your theory lessons: we want to see how the modeling with MESAbinary behaves in such cases. Remember to always make a copy of your work directory, to not overwrite your outputs.

4. **CONTACT binary**

Recall that in the lectures we saw different situations in which a system can evolve into contact. Here below, we are going to reproduce one of these situations, and see what MESA tells us at the end of the run.

(a) **Change** the value of the conservative mass transfer control, `mass_transfer_beta`, to a lower one, let's say `0.2d0`:

`mass_transfer_beta = 0.2d0`

This is in your `binary_controls` namelist, in `inlist_project`. As you know, this is going to decrease the amount of mass which is lost as "re-emission" from the vicinity of your accretor.

(b) **Evolve** the model binary with your renewed initial conditions:

$$M_1 = 1\,M_\odot\,, \quad M_2 = 0.8\,M_\odot\,, \quad P_{orb} = 0.4\,\text{days}\,, \quad \beta = 0.2$$

[Run of ~ 7 min].

(c) **Look** attentively at the PGstar window and:

i. Acknowledge, in the donor's window, the various phases you just saw in the previous run. How is the mass ratio evolving, with respect to before?

ii. Find in the window the information about the radius R_2 of your accretor, and look at it attentively as it reaches its respective Roche Lobe R_L. When this happens, the system has become a ***contact binary***: did you expect this from what you saw in the lectures? Why is the accretor able to fill its Roche Lobe? (*Hint*: look at the mass ratio!).

(d) **After** the run has finished,

i. Look at the final lines printed in your terminal: these are `MESA`'s final words, to tell you that the run has been terminated. Remember these: some of you were assigned with systems which are expected to give such outcome!!

```
termination code:  Terminate because accretor (r-rl)/rl >
                   accretor_overflow_terminate
```

NB If you want to let your system evolve a bit further, there is a nice option in your `binary_controls.defaults` parameters library, namely `accretor_overflow_terminate`. It allows for a bit of contact from the accretor before terminating the run. You may want to play with it and see how it goes.

5. **UNSTABLE mass transfer**

You know that the response of a deep convective envelope to mass loss is an asymptotically *adiabatic expansion*, leading in the end to an unstable mass transfer episode. We are now going to produce a system showing such a behavior, so that we can look at how `MESAbinary` acts in this case.

(a) **Change** the value of the initial period P to 3 days and resume the original value of the mass transfer parameter β:

```
initial_period_in_days = 3.d0       !days
    mass_transfer_beta = 0.5d0
```

Starting from a loosened binary will allow the first contact to happen later in the evolution of our donor, namely when it starts ascending its RGB.

(b) **Evolve** the model binary with your renewed initial conditions:

$$\mathrm{M}_1 = 1\,\mathrm{M}_\odot\,, \quad \mathrm{M}_2 = 0.8\,\mathrm{M}_\odot\,, \quad \mathrm{P}_{\mathrm{orb}} = 3\text{ days}\,, \quad \beta = 0.5$$

[Run of ~ 7 min].

(c) **Look** attentively at the PGstar window and:

i. Recognize the deepening of the convective envelope during the ascension of the RGB.

ii. Follow the information about the radius R_2 of your donor, and look at it attentively as it reaches its respective Roche Lobe R_L. Look at the $\log \dot{\mathrm{M}}$−Age (yr) plot and see the correspondent abrupt rise of the mass transfer rate, which is also printed above in the Summary panel. This rate is *enormously* high for our Sun-like star: this is the start of an ***unstable mass transfer episode***.

(d) **After** the run has finished,

i. Look at the final lines printed in your terminal: these are `MESA`'s final words, to tell you that the run has been terminated. Remember these: you will encounter them sometimes!!

`termination code: min_timestep_limit`

`MESA` is not able to deal with an unstable mass transfer, and it is indeed expected to complain sooner or later: the complaining you are seeing now is about the minimum timestep being reached.
NB These final words can be encountered also in other situations, *not necessarily* linked to an unstable mass transfer episode.