

Bildsten Maxilab: Climbing the RGB in a Binary

Friday, August 15, 2014

Setup

In this exercise, we'll see what conditions lead to unstable or stable mass transfer from a star climbing the red giant branch onto companions of varying mass. To do this, we'll make elementary use of MESA's `binary` module, which can track two stars simultaneously, handling changes in orbital parameters due to mass transfer through Roche lobe overflow, winds, magnetic braking, gravitational wave radiation, and some other options.

Whirlwind Tour of the Binary Work Directory (if familiar, skip)

Before we begin, we should take a whirlwind tour of how a binary work directory is set up. Open up `mesa/binary/work` and take a look inside. For the most part, things should look similar to a standard `star` work directory. However, there are two “bonus” inlists called `inlist1` and `inlist2` as well as two logs directories called `LOGS1` and `LOGS2`. To understand these, let's first look at the basic `inlist`. At first you might think it's just a familiar inlist, but look at the namelists: `&binary_job`, `&binary_controls`, and `&binary_pgstar`. The first two are analogous to the `&star_job` and `&controls` namelists in a regular `mesa/star` directory. Ignore the third one as it is not used as of the writing of this guide.

Perhaps unsurprisingly, all of these namelists are read from another inlist, `inlist_project`. Open that up now and take a look inside. The controls therein will largely be unfamiliar to you, but in `mesa/binary/defaults` you can find some documentation for most of these commands (though not as complete as that for `mesa/star`).

First let's look at the `&binary_job` namelist. The main gist of what's going on is that inlists for two stellar models are being read from `inlist1` and `inlist2`. Those are regular inlists for `mesa/star`, wherein you can load/save models, change nuclear networks, and whatever niceties you're used to doing in `mesa/star`. Inside those inlists, you'll see that the log directories, photo directories, and log output have all been set for each star (where the `LOGS1`, `LOGS2`, `photos1`, etc. directories are coming from). The `evolve_both_stars = .false.` is important since it says we are only evolving star 1. That is, if this is set to false, `inlist2` is essentially ignored, and `LOGS2` will remain empty. You can think of this as calling the secondary a black hole whose evolution is uninteresting, though its mass and orbital separation may change depending on the binary's evolution. Finally, the `pgstar` command determines which star or stars will get `pgstar` plots. To show plots for the first star, this should be set to 1.

Now let's look at the `&binary_controls` namelist. Here we set the masses of the two stars as well as their initial period. This sets the initial orbit (assumed to be circular, but this can be changed). Look in `mesa/binary/defaults/binary_controls.defaults` for more controls to determine how mass and angular momentum are transferred (by default all transfer is conservative) and other setup options. Of particular note is that the initial binary separation can be set by setting `initial_period_in_days` to something negative and then setting `initial_separation_in_Rsuns`

to your binary separation.

To get things going, it's just the same as a normal work directory. You would run `./mk` and `./rn` (No need to run this basic work directory, as you'll be creating your own shortly). If all is well, your run should commence, and if you've allowed `pgstar` in your binary inlist as well as the appropriate star inlists (and actually turned on some plot windows!), you should see plots for individual stars showing up.

Setting up the lab directory

Just like `mesa/star`, `mesa/binary` has a test suite. We'll use one of the test suite cases as a basis for this lab. Copy over the `binary_donor_only_implicit.mdot` directory via

```
cp -r $MESA_DIR/binary/test_suite/binary_donor_only_implicit.mdot ./
```

assuming your current directory is where you want the directory. Immediately delete the line setting `MESA_DIR` in `make/makefile`. This is something you [currently] must always do with all test cases in either `mesa/star` or `mesa/binary`.

Next, if you haven't done so already, download and unzip the directory for the maxilab from the MESA Summer School 2014 repository. The file should be called `bildsten_maxilab.tar.gz` or something like that. This contains the starting models for this lab, but is not a fully-functional work directory since part of the exercise is about getting familiar with `mesa/binary`. Move these saved models into your work directory. There should be six of them:

```
0.8M_RGB_10L.mod
0.8M_RGB_100L.mod
1.0M_RGB_10L.mod
1.0M_RGB_100L.mod
1.2M_RGB_10L.mod
1.2M_RGB_100L.mod
```

Notice that your work directory has pretty much the same structure as the standard binary work directory, so this should be familiar ground. Open up `inlist_project` and examine it. You should notice that once again, we are only evolving one star (that's the "donor only" part of the test suite name). Additionally, at the bottom we have the command `max_tries_to_achieve = 50`. This instructs `binary` to use an "implicit" scheme to determine mass loss rates in the case of Roche lobe overflow (the "implicit" part of the test suite case).

Implicit vs. Explicit Mass Loss The implicit scheme (turned on by `max_tries_to_achieve`), will have `binary` do a root find for a mass loss rate to ensure that at the end of the timestep, the radius is within some tolerance of the Roche radius. If it is too large, it retries the timestep with a higher mass loss rate. It will do this readjustment up to the number of times you specify before doing a more full-blown retry at a lower timestep. In an explicit scheme, there is a concrete formula involving the stellar and binary parameters that determines the mass loss rate, regardless of whether or not it keeps the star confined to its Roche radius.

Prompt

Randomly pick a companion mass for each model provided to you that is between 1 and 3 times as massive as the giant model. Adjust `inlist_project` to have the correct donor (`m1`) and accretor (`m2`) masses for your binary configuration, and set the initial binary period or separation so that the donor’s Roche radius is just slightly larger than its initial radius. You can use Eggleton’s Roche radius approximation,

$$\frac{R_{\text{RL,donor}}}{a} = \frac{0.49q^{2/3}}{0.6q^{2/3} + \ln(1 + q^{1/3})} \quad (1)$$

where $q \equiv M_{\text{donor}}/M_{\text{accretor}}$ and a is the semi-major axis of the orbit (i.e. the orbital separation), to get your orbital separations by setting $R_{\text{RL,donor}} = R_{\text{donor}}$. The initial radii of the giant models are given in Table 1. The orbital period or separation you choose should be a few percent (or

Model Name	Mass	Luminosity	Radius
0.8M_RGB_10L.mod	$0.8 \mathcal{M}_{\odot}$	$10 \mathcal{L}_{\odot}$	$4.95 \mathcal{R}_{\odot}$
0.8M_RGB_100L.mod	$0.8 \mathcal{M}_{\odot}$	$100 \mathcal{L}_{\odot}$	$20.1 \mathcal{R}_{\odot}$
1.0M_RGB_10L.mod	$1.0 \mathcal{M}_{\odot}$	$10 \mathcal{L}_{\odot}$	$4.7 \mathcal{R}_{\odot}$
1.0M_RGB_100L.mod	$1.0 \mathcal{M}_{\odot}$	$100 \mathcal{L}_{\odot}$	$19.0 \mathcal{R}_{\odot}$
1.2M_RGB_10L.mod	$1.2 \mathcal{M}_{\odot}$	$10 \mathcal{L}_{\odot}$	$4.5 \mathcal{R}_{\odot}$
1.2M_RGB_100L.mod	$1.2 \mathcal{M}_{\odot}$	$100 \mathcal{L}_{\odot}$	$18.2 \mathcal{R}_{\odot}$

Table 1: Initial radii for provided giant models.

possibly more) larger than what you get by setting $R_{\text{RL,donor}} = R_{\text{donor}}$. This is because we want to allow the donor to expand into its Roche lobe rather than starting the run off with the donor already larger than its Roche lobe (undesirable things happen if you do this, but it will probably run).

You will also need to adjust `inlist1` to load the proper giant model and remove the lines containing the model number, retries, and backup stopping conditions as well as the incorrect setting of `mesa_dir` (all relics of this being a test suite case).

Additionally, I’d recommend turning on some pgstar plots to see what the star’s doing. My personal favorite general pgstar option is `Grid1`. You can turn it on by putting `Grid1_win_flag = .true.` in the pgstar namelist of `inlist1`. Additional pgstar tips in the **Recommendations** section.

Your task is to run your binary configurations for each of the six models and determine whether or not the resulting mass transfer is stable or unstable. In the crowdsourcing Google spreadsheet [here](#), there will be sheets/tables for each of the six starting models. In these tables, please report your accretor mass and its stability. If it is definitely stable, report “1”. If it is definitely unstable, report “−1”. If it is unclear, report “0”. That’s it! If you finish all six with time to spare, try finding the critical mass ratio that splits the stable and unstable configurations for one of the giant models by trying many companion masses for your selected donor model. Is this ratio what you expected?

For some tips on picking a good orbital separation and determining stability, read on.

Recommendations

Getting a Useful Orbital Separation

In setting up your `inlist_project`, you'll need to set up an initial orbital period. It's easier to instead set an orbital separation (see above whirlwind tour on how to do this instead).

Using the known radius of the giant star from Table 1 as the Roche lobe radius in the Eggleton formula, you can solve for a binary separation. However, this may still be too small. We want the giant to expand to fill its Roche lobe, but we don't want to start the model with the Roche lobe already overfull lest unexpected things happen. You are recommended, then, to calculate an initial orbital separation and then increase it by a few percent or so before putting it in your `inlist_project`.

In general, if you start your model and it has trouble converging, look at the mass loss rate (`lg_mdot`; should be on-screen during the run for each iteration in a timestep) that is being attempted. If it is much higher than $10^{-10} \mathcal{M}_{\odot}/\text{yr}$, your star is probably already overflowing its Roche lobe. Try increasing your initial orbital separation and running it again.

Conversely, if your model starts off happily, but mass loss rates remain low ($< 10^{-20} \mathcal{M}_{\odot}/\text{yr}$) for more than 50 or so timesteps, you probably need to reduce your initial orbital separation (or wait a long time for the giant to grow in radius).

Determining stability

As usual, the best way to get intuition for a run in MESA is to use plots. You are encouraged to make a custom history plot to see the mass loss rate in real time (using, perhaps, `History_Track2`). For the x -axis, use some measure of time, such as `star_age` or `model_number`, and for the y -axis, use `log_abs_mdot`. You also may want to specify sensible limits on the y axis like -12 to -6 . Mass loss rates less than $10^{-12} \mathcal{M}_{\odot}/\text{yr}$ or above $10^{-6} \mathcal{M}_{\odot}/\text{yr}$ are likely due to not filling the Roche lobe yet or being unstable, respectively. Figure 1 shows an example of a stable mass loss scenario and Figure 2 shows an example of an unstable mass loss scenario.

What you're looking for is the general shape of stability. As the giant approaches Roche lobe filling, the mass loss rate ramps up rather quickly. Once it's full, the mass loss rate should either flatten out (the stable case) or skyrocket (unstable case). However, even unstable cases will eventually flatten out as the matter is quickly removed from the donor until the mass ratio is brought down. The telltale spike in Figure 2 is easy to recognize, but more intermediate companion masses may have a more subtle bump at more modest mass loss rates. In general, the mass loss rate should not exceed $10^{-6} \mathcal{M}_{\odot}$ and the star shouldn't lose more than 5% of its initial mass before stabilizing, but these are guidelines, not rules. There is no obvious bifurcation between stability and instability.

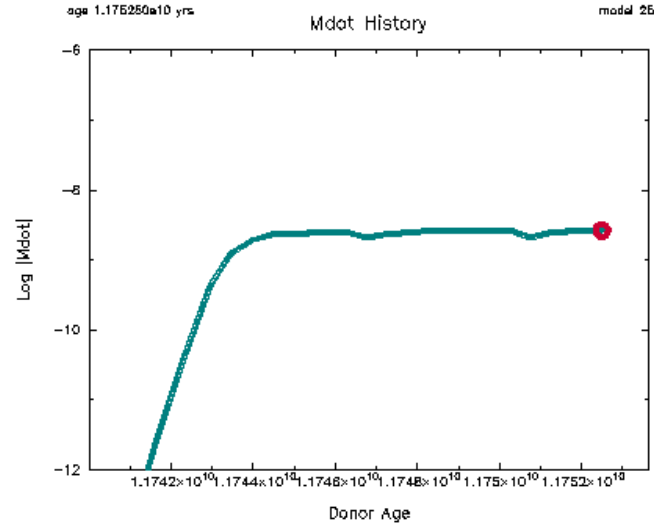


Figure 1: Stable mass transfer. Mass loss ramps up until Roche lobe is filled, and then stabilizes.

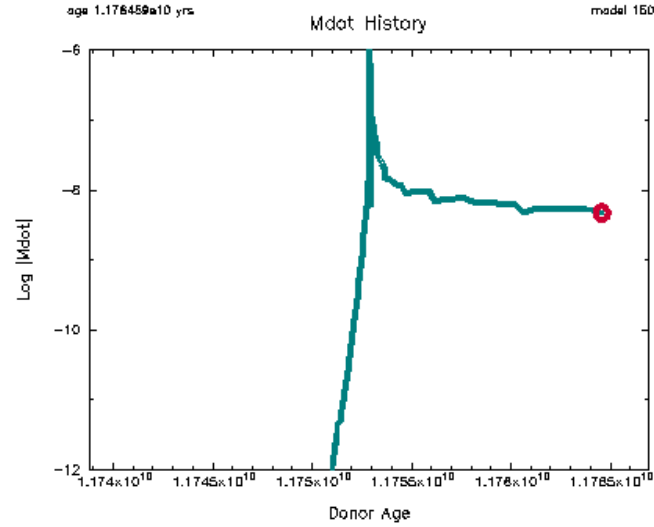


Figure 2: Unstable mass transfer. Mass loss ramps up until Roche lobe is filled, then it skyrockets for a short time, losing a large amount of mass until the donor is light enough for stable mass transfer to resume.