Interacting Massive Binaries

Massive stars are important for many fields of astrophysics: they are thought to have provided most of the ionizing radiation that caused cosmic reionization, their remnants merge in gravitational wave events, and they even produced the oxygen we breathe. Understanding the evolution of massive stars is therefore a key component, for example, when interpreting the radiation from distant galaxies or estimating the merger rates of neutron stars.

However, massive stars are not born alone. In fact, they spend their lives in so close orbit with at least one companion star that dramatic interaction is bound to happen as the stars evolve and swell. Through interaction, many solar masses of material can be exchanged between the two stars, X-rays can be produced, and the stars can even coalesce. In this lab, we will use MESA's binary module to model a variety of interactions in massive binaries.

Modeling binaries in `MESA`

`MESA` has a `binary` module, which is very similar to the `star` module, but has the capability to evolve two stars simultaneously while also accounting for mass exchange between them. To accommodate both stars and the binary orbit, there is one inlist for the binary orbit (`inlist_project`) which points to one inlist for each of the stars (`inlist1`, `inlist2`). It is also possible to evolve just one of the stars in the system, while treating the other as a point mass, which can be handy when simulating systems composed of a star and a compact object, or when you are mostly interested in the evolution of one of the stars. Similar to the `star` module, there is a `binary_history.data` output file which tracks information about the binary system while there still are the `history.data` and profile files for the individual components in their respective `LOGS1` and `LOGS2` folders. There is also a `run_binary_extras.f90` file in the `src` directory, for when you need to access the two stars at the same time or interact with parameters of the binary orbit.

You can start a binary run using the `$MESA_DIR/binary/work` directory, which is similar to the `$MESA_DIR/star/work` directory. However, to avoid setting up many initial conditions, we will start from prepared work directories in these labs.

Outline

We have prepared three laboratory exercises that will familiarize you with how to use the `binary` module to model interacting binary stars, as well as some of the difficulties you might run into in the process.

Minilab 1 Get familiarized with the `MESA binary` module by modeling the evolution of a star that loses its fluffy hydrogen-rich envelope via mass transfer to a companion star. Also experimenting with changing the wind mass loss prescription.

Minilab 2 Explore what types of supernovae to expect from envelope-stripped stars and model the prior X-ray binary phase.

Maxilab Make exploratory models of binary stars that merge and further evolve the resulting merger products.

1 Minilab 1: Envelope-stripping in a binary

1.1 The evolution leading to a stripped star

About a third of all massive stars are expected to lose their hydrogen-rich envelopes either via mass transfer or the successful ejection of a common envelope. Here, we start exploring envelope-stripping via Roche-lobe overflow (RLOF, mass transfer) in a binary initially composed of a 14 $M_\odot$ star that orbits a 8 $M_\odot$ black hole every 20 days. To speed up the models, we will model stars at the sub-solar metallicity Z = 0.006 (similar to the metallicity in the Large Magellanic Cloud).

> ***Task 1: Prepare your model.*** *Start by downloading the folders called* `minilab1` *and* `ZAMS_models`*. Update the initial masses and initial period of the two stars. Locate the 14 $M_\odot$ zero-age main-sequence (ZAMS) model in the folder for ZAMS models and make sure your model will use it to start from. Remember that it has a different metallicity than the default in MESA. Compile and run the code.*

Solution Task 1: Update the initial conditions for the binary in the `inlist_project` and `inlist_donor`:

```
! Initial conditions for the binary
  m1 = 14.0d0  ! donor mass in Msun
  m2 = 8.0d0 ! companion mass in Msun
  initial_period_in_days = 20.0d0
```

```
! Load ZAMS model
  load_saved_model = .true.
  saved_model_name = 'zams_14Msun.mod'
```

```
use_Type2_opacities = .true.
Zbase = 0.006
```

```
! Metallicity
  initial_z = 0.006
```

A large pgstar dashboard should pop up. Study each panel and see if you can understand what they are showing you. Can you recognize the mass transfer phase in the Hertzsprung-Russell diagram? Can you distinguish the mass transfer from wind mass loss?

Task 2: Analyze the model. *At the time the simulation stops, the stellar model should be fusing helium to carbon to carbon and oxygen in its center. Identify the following characteristics for this binary stellar model. All quantities refer to the donor star unless otherwise indicated.*

- *What is its mass?*
- *What is its effective temperature?*
- *How large is the star?*
- *At what rate did the star lose mass at maximum? At what rate is the star losing mass now?*
- *What is the mass of the companion?*
- *How long is the orbital period?*

Solutions Task 2:

- $\sim 5.3\, M_\odot$
- $\sim 91,000$K
- $\sim 0.91 R_\odot$
- max: $\sim 10^{-2}\, M_\odot$/yr, now: $10^{-6}\, M_\odot$/yr
- $15\, M_\odot$
- ~ 38 days

As you notice from your model, this stripped star does not have regular stellar properties anymore! It is very small and extremely hot for its mass.

`MESA` pro-tip
The `inlist_pgstar_binary` inlist is instructing your model to dutifuly output pngs of pgstar frames to the folder `grid_png`. While you shouldn't need past frames to answer the questions in this task, you still might find it useful to create a movie out of these. Fortunately, that ability is baked in to the MESA SDK! Simply use the script `images_to_movies`.

`images_to_movies` takes a glob string and a name for an output video file. The glob string should be something like `'grid_png/grid_*.png'` (the quotes are important), which can match the files you wish to use as frames. Then the output file should end in `mp4` to indicate making an `mp4` movie, so something like `to_he_exhaustion.mp4`. So putting it all together:

```
images_to_movie 'grid_png/grid_*.png' to_he_exhaustion.mp4
```

1.2 Updating the wind mass-loss scheme

The default wind mass-loss rate for stripped stars in `MESA` is an extrapolation of the Nugis & Lamers (2000) empirical wind mass-loss recipe created from a sample of Galactic Wolf-Rayet (WR) stars. This extrapolation reproduces relatively accurately the wind mass loss rate from the only published intermediate mass stripped star, the $\sim 4\,M_{\odot}$ quasi-WR star in the binary HD 45166 (Groh et al., 2008). However, a recent theoretical model from Vink (2017) predicts that the wind mass loss rate from intermediate mass stripped stars should be significantly lower. This, because WR stars are so bright that radiation pressure contributes to the wind mass-loss, while it should be unimportant for intermediate mass stripped stars, which are less luminous. As a result, Vink (2017) proposed that the wind mass-loss rate from stripped stars is related to the luminosity and metallicity following the relation:

$$\log_{10} \dot{M} = -13.3 + 1.36 \log_{10}(L/L_{\odot}) + 0.61 \log_{10}(Z/Z_{\odot}) \tag{1}$$

If stellar winds are strong, they can significantly alter the stellar properties of a star, revealing deeper, chemically enriched layers. For stripped stars, the amount of hydrogen left-over after mass transfer not only affects the stellar properties, but also the possibility to interact again and the supernova type (see Minilab 2).

Task 3: Implement a custom wind *Make a copy of your first work directory that you used for the first two tasks. In the new directory, implement the Vink (2017) wind mass loss rate for stripped stars with mass $< 10\,M_{\odot}$ in the* `run_star_extras.f90` *file and activate the* `other_wind` *hook. Note that we have prepared a function into which you can implement the new wind mass loss scheme. Compile and run your model.*

Switch off the previous hot-wind scheme and switch on the other_wind hook in the inlist:
Go to run_star_extras.f90. Inside the subroutine extras_controls:

```
! 2) Hot stars
  !hot_wind_scheme = 'Dutch'
  !Dutch_scaling_factor = 1d0

  hot_wind_scheme = 'other'
  use_other_wind = .true.
```

```
      s% how_many_extra_profile_header_items => how_many_extra_profile_header_items
      s% data_for_extra_profile_header_items => data_for_extra_profile_header_items

      s% other_wind => weak_wind_stripped_stars

   end subroutine extras_controls
```

```
   subroutine weak_wind_stripped_stars(id, Lsurf, Msurf, Rsurf, Tsurf, X, Y, Z, w, ierr)
```

This is the function to implement the wind scheme in:
Define Mlim and set Mlim = 10 Msun

```
real(dp) :: Zsun, center_h1, Mlim
Mlim = 10.0*Msun
```

Here is where you call your new function - note the use of Mlim:

```
! Call the wind functions
if ((Tsurf >= 1.0d4) .and. (Xsurf >= 0.4)) then
   call eval_Vink_wind(w)
else if ((Tsurf >= 1.0d4) .and. (Xsurf < 0.4) .and. (Msurf > Mlim)) then
   call eval_Nugis_Lamers_wind(w)
! Stripped stars less massive than Mlim
else if ((Tsurf >= 1.0d4) .and. (Xsurf < 0.4) .and. (Msurf < Mlim)) then
   call eval_Vink17_wind(w)
else
   call eval_de_Jager_wind(w)
end if
```

And the function may look like this:

```
contains

  ! The Vink 2017 theoretical algorithm
  subroutine eval_Vink17_wind(w)
     real(dp), intent(out) :: w
     real(dp) :: log10w
     include 'formats'
     log10w = -13.3 + 1.36*log10(Lsurf/Lsun) + 0.61*log10(Zsurf/Zsun)
     w = exp10(log10w)
  end subroutine eval_Vink17_wind
```

Task 4: Explore how the properties of the stripped star changed since you updated the wind mass loss scheme:

- *What is its mass?*
- *What is its effective temperature?*
- *How large is the star?*
- *At what rate did the star lose mass at maximum? At what rate is the star losing mass now?*
- *What is the mass of the companion?*
- *How long is the orbital period?*

Solutions Task 4:

- $\sim 5.6\ M_\odot$
- $\sim 76,000$K
- $\sim 1.38 R_\odot$
- max: $\sim 10^{-2}\ M_\odot$/yr, now: $10^{-7}\ M_\odot$/yr
- 15 $M_\odot$
- $\sim$ 37 days

As you noticed, the weaker wind for stripped stars significantly alters the effective temperature and the radius of the star. This is because less hydrogen is lost via winds and since hydrogen is more fluffy than helium, the star can be larger and therefore also a little cooler.

1.3 Bonus: Ionizing radiation

Because stripped stars reach extremely high surface temperatures, they emit very energetic radiation. In fact, the majority of their emitted radiation is ionizing. To accurately compute the emission rate of ionizing photons, spectral models that match the properties of the stripped stars are necessary. However, the hydrogen-ionizing emission rate from intermediate mass stripped stars is quite well approximated by a blackbody radiation curve.

Bonus Task 5: Track ionizing radiation *In the text-file `Q0_function.txt` in the work directory, there is a function that calculates the emission rate of H-ionizing photons, Q_0. Copy this function and place it in the correct place in the `run_star_extras.f90` file so that you can add $\log_{10}(Q_0)$ as an output in the `history.data` file. Make sure to call the new history column `log_Q0`. Before compiling and running the model, un-comment the fourth panel in the pgstar window (tip: search for EDIT HERE) so that you can see the evolution of the emission rate of H-ionizing photons. It is also advantageous to set the minimum of the y-axis to 46.*

This function goes into data_for_extra_history_columns in the run_star_extras.f90 file:

```
subroutine data_for_extra_history_columns(id, n, names, vals, ierr)
   integer, intent(in) :: id, n
   character (len=maxlen_history_column_name) :: names(n)
   real(dp) :: vals(n)
   integer, intent(out) :: ierr
   type (star_info), pointer :: s
   ierr = 0
   call star_ptr(id, s, ierr)
   if (ierr /= 0) return

   ! note: do NOT add the extras names to history_columns.list
   ! the history_columns.list is only for the built-in history column options.
   ! it must not include the new column names you are adding here.

   ! Add the emission rate of H-ionizing photons to the history file
   names(1) = 'log_Q0'
   call calculate_Q0(vals(1))

   contains

   subroutine calculate_Q0(log10_Q0)
```

You also need to say that you have one extra history column here:

```
integer function how_many_extra_history_columns(id)
   integer, intent(in) :: id
   integer :: ierr
   type (star_info), pointer :: s
   ierr = 0
   call star_ptr(id, s, ierr)
   if (ierr /= 0) return
   how_many_extra_history_columns = 1
end function how_many_extra_history_columns
```

Un-comment the output variable here:

```
   ! The fourth panel
   ! ! ! ! EDIT HERE ! ! ! ! ! ! !
   History_Panels1_yaxis_name(4) = '' !'log_Q0'   ! Extra exercise, minilab 1
   !History_Panels1_ymin(4) = 46   ! Extra exercise, minilab 1
   History_Panels1_other_yaxis_name(4) = '' !'log_LX' ! Extra exercise, minilab 2
   ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !
```

OB-type stars are thought to provide most of the ionizing radiation in a stellar population. Let's compare our stripped star to published spectral models for main-sequence OB-stars from the

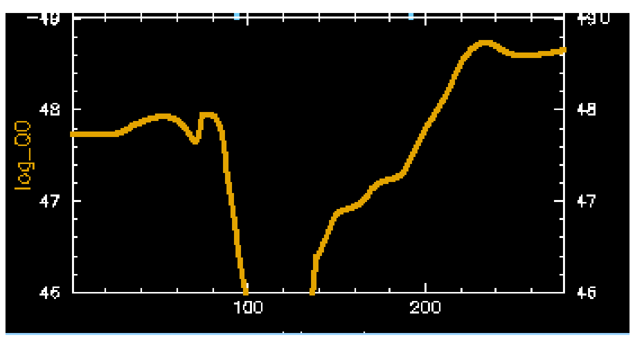
Figure 1: Emission rates of hydrogen, neutral helium, and ionized helium ionizing photons (Q_0, Q_1, and Q_2, respectively) for main-sequence OB-type stars (Smith et al., 2002).

Model Ref.	$T_{\rm eff}$ (kK)	log g (cm s^{-2})	$R_\star$ ($R_\odot$)	SpT	Z$_\odot$ log $\dot{M}$ / Q$_0$	Z$_\odot$ Q$_1$	Z$_\odot$ v_∞ / Q$_2$	0.2 Z$_\odot$ log $\dot{M}$ / Q$_0$	0.2 Z$_\odot$ Q$_1$	0.2 Z$_\odot$ v_∞ / Q$_2$	0.4 Z$_\odot$ log $\dot{M}$ / Q$_0$	0.4 Z$_\odot$ Q$_1$	0.4 Z$_\odot$ v_∞ / Q$_2$	0.05 Z$_\odot$ log $\dot{M}$ / Q$_0$	0.05 Z$_\odot$ Q$_1$	0.05 Z$_\odot$ v_∞ / Q$_2$	2.0 Z$_\odot$ log $\dot{M}$ / Q$_0$	2.0 Z$_\odot$ Q$_1$	2.0 Z$_\odot$ v_∞ / Q$_2$
OB#1	50.0	4.00	9.8	O3V	−5.85		3150	−6.41		2550	−6.17		2790	−6.89		2130	−5.61		3440
					49.5	48.8	45.8	49.5	49.0	45.6	49.5	48.9	45.7	49.6	49.1	45.4	49.5	48.8	45.6
OB#2	45.7	4.00	10.4	O4V	−6.08		2950	−6.63		2390	−6.39		2610	−7.12		1990	−5.83		3220
					49.4	48.7	45.0	49.4	48.7	45.2	49.4	48.7	45.2	49.4	48.8	45.0	49.4	48.6	44.8
OB#3	42.6	4.00	10.5	O5V	−6.24		2870	−6.80		2330	−6.55		2550	−7.28		1940	−6.00		3140
					49.2	48.5	44.4	49.2	48.6	44.2	49.2	48.5	44.3	49.2	48.6	44.2	49.2	48.5	44.5
OB#4	40.0	4.00	10.6	O7V	−6.34		2500	−6.90		2020	−6.66		2210	−7.38		1690	−6.10		2730
					49.0	48.2	43.8	49.0	48.3	43.4	49.0	48.3	43.5	49.0	48.4	43.6	49.0	48.1	37.7
OB#5	37.2	4.00	10.5	O7.5V	−6.47		2100	−7.03		1700	−6.79		1860	−7.51		1420	−6.23		2290
					48.7	47.6	37.3	48.8	47.8	42.3	48.8	47.8	42.4	48.7	47.8	42.5	48.7	47.4	38.5
OB#6	34.6	4.00	10.5	O8V	−6.64		1950	−7.20		1580	−6.96		1730	−7.68		1320	−6.40		2130
					48.5	46.6	36.3	48.4	46.9	36.7	48.4	46.8	36.7	48.4	46.7	38.7	48.4	46.7	35.4
OB#7	32.3	4.00	10.1	O9V	−6.74		1500	−7.30		1210	−7.06		1330	−7.79		1010	−6.50		1640
					47.9	45.7	34.3	48.0	45.5	34.8	47.9	45.5	34.9	48.0	45.6	34.8	48.0	45.6	33.6
OB#8	30.2	4.00	9.5	B0V	−6.92		1200	−7.48		970	−7.24		1060	−7.96		810	−6.68		1310
					47.4	44.8	32.7	47.4	44.6	33.3	47.3	44.8	33.4	47.2	44.5	33.2	47.4	44.9	32.7
OB#9	28.1	4.00	8.6	B0.5V	−6.74		800	−7.30		640	−7.06		710	−7.79		540	−6.50		870
					47.0	44.5	31.5	46.8	44.2	32.4	46.8	44.3	32.2	46.8	43.9	32.1	46.9	44.5	31.1
OB#10	26.3	4.00	8.1	B1V	−6.89		700	−7.45		560	−7.20		620	−7.93		470	−6.65		760
					46.5	43.9	30.5	46.3	43.7	31.6	46.4	43.8	31.4	46.2	43.3	30.9	46.4	44.1	...
OB#11	25.0	4.00	7.1	B1.5V	−7.08		600	−7.63		480	−7.39		530	−8.12		400	−6.83		650
					46.1	43.4	...	45.9	43.2	30.8	46.1	43.3	30.9	45.8	42.9	...	46.1	43.7	...

literature. Figure 1 contains a table from Smith et al. (2002) which shows how the emission rates of hydrogen-ionizing photons (Q_0) vary for different massive main sequence stars.

Bonus Task 6: Compare to OB-star models. *Look at Figure 1 and find out what the spectral type is of the main sequence star that has the most similar emission rate of H-ionizing photons as your stripped star? How does it compare in effective temperature and radius? (Remember that the metallicity of the stripped star is Z = 0.006, which is about 0.5 Z$_\odot$.)*

Here is the new panel where you can see that during the stripped star phase (last phase), the stripped star radiates H-ionizing photons at a rate of $\sim 10^{48.6}$ photons/second. The O7.5V type star



of the 0.4 Z$_\odot$ models is the most similar in terms of Q_0 and metallicity. This main sequence star is about ten times larger and is about 40,000 K cooler.

2 Minilab 2: Stripped-envelope supernova progenitors

Even though only one intermediate mass stripped star has been published, many supernovae thought to result from the collapse of stripped stars have been identified. These stripped-envelope supernovae constitute about a third of all core-collapse supernovae and are, unsurprisingly, hydrogen-poor (Smith et al., 2011, Graur et al., 2017ab).

2.1 Stripped-envelope supernovae

There are three main classes of stripped-envelope supernovae, distinguished by how much hydrogen or helium they contain. Following Hachinger et al. (2012), we make the following distinction:

- Type IIb: Hydrogen-poor ($> 0.03\, M_\odot$), helium-rich
- Type Ib: Hydrogen-free ($< 0.02\, M_\odot$)[1], helium-rich ($> 0.14\, M_\odot$)
- Type Ic: Hydrogen-free ($< 0.02\, M_\odot$), helium-free ($< 0.06\, M_\odot$)

As this list shows, the amount of leftover hydrogen at explosion is an important determinant for the supernova type. In this lab, we will explore how much hydrogen is left on the surface of stripped stars and what its effect is, not the least for what supernova the star produces.

Task 1: Prepare a progenitor.

a) *Navigate back to the original work directory from Minilab 1, before we updated the wind mass loss scheme.*

b) *Change the stopping condition to core carbon-depletion, which we will approximate as when the central mass fraction of ^{12}C drops below 10^{-2}. This isn't technically "right before" core collapse, but we don't anticipate the outer regions of the star changing substantially between this time and core collapse.*

c) *To infer the supernova type, we need to track how much hydrogen and helium is left in the star in total. Look for a parameter in the `controls.default` file that outputs data to the terminal and set it to display the total mass of ^1H and total mass of ^4He. Hint: search for the word "trace".*

d) *Restart your model from where it stopped last.*

b) The updated stopping criterion goes into the inlist of the star

c) Output the total masses of hydrogen and helium by adding the following to the controls in your inlist for the star:

d) To restart, you type ./re and press enter

[1]Dessart et al. (2012) found that this limit could be as low as 0.001 $M_\odot$.

```
! Central abundance limit
  xa_central_lower_limit_species(1) = 'c12'
  xa_central_lower_limit(1) = 1d-2
```

```
num_trace_history_values = 2
trace_history_value_name(1) = 'total_mass_h1'
trace_history_value_name(2) = 'total_mass_he4'
```

Task 2: Supernova taxonomy. *How much hydrogen and helium is left in the stripped star? What type of supernova do you expect it to result in?*

Total mass of hydrogen: $9 \times 10^{-9}\,M_\odot$, Total mass of helium: $1.41\,M_\odot$, I expect this star to result in a type Ib supernova.

Task 3: Back to weaker winds. *Now, let's look at the future evolution when we use the lower wind mass loss scheme you implemented in minilab 1. Go back to the work directory from minilab 1 in which you implemented the lower wind mass loss rate. Update it in the same way you updated the other model in Task 1 and restart this weaker wind model. For this model, do not wait for central carbon depletion, it will take too long to compute that far. Instead, once your model reaches a second phase of mass transfer (one of the panels shows mass transfer rate), you can analyze the output.*

Task 4: Supernova taxonomy revisited. *In this weaker wind model, how much hydrogen and helium is left in the stripped star when it enters the second mass transfer phase? What type of supernova do you expect? Is it different from what is expected when weaker wind is not implemented?*

Total mass of hydrogen: $0.05\,M_\odot$ (at C-depletion: $0.04\,M_\odot$), total mass of helium: $1.67\,M_\odot$ (at C-depletion: $1.60\,M_\odot$). I expect this star to result in a type IIb supernova. Because of the weaker winds, this star has more hydrogen left at these late stages.

Bonus Task 5: What about type Ic? *If the second phase of mass transfer initiated during the helium shell burning of the stripped star does not last long enough for a significant amount of helium to be stripped off, how can type Ic supernovae be created?*

1. They can come from single stars with very strong stellar winds (WC stars)
2. Low-mass versions can be created if mass transfer starts earlier for the stripped star, maybe during the central helium burning phase. This can occur if the star is stripped via common-

envelope ejection, then the post-interaction orbital period can be so short that the stripped star fills its tiny Roche lobe during central helium burning.

2.2 Supernova progenitors

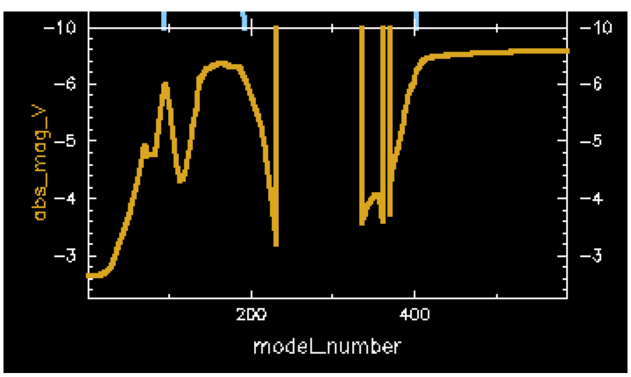
To identify what type of star is responsible for a certain type of supernova, archival images of the progenitor can sometimes be found. If the progenitor can be found, it is brighter than the magnitude limit of the observations. iPTF13bvn was a stripped envelope supernova for which the progenitor was detected (Cao et al., 2013). It was found to have V 25 mag and the host galaxy is located at 25 Mpc distance.

MESA can provide decent estimates for the absolute magnitudes of a large number of stars (Lejeune et al., 1998). However, as you have discovered, stripped stars are not standard stars and are therefore not part of standard spectral libraries.

Task 6: Detection distance *Use one of the panels in the pgstar window to find the absolute V-magnitude in the model you just created.*

a) *Can you see where MESA cannot find a realistic absolute magnitude?*

b) *Assuming that the model you just created is a good representation of a supernova progenitor, out to what distance would it be detectable if available images were limited to the magnitude $V < 25$ mag? Hint: the absolute magnitude, the apparent magnitude, and the distance are related to each other according to $M = m - 5\log_{10}(d_{pc}) + 5$, where M is the absolute magnitude, m is the apparent magnitude and d_{pc} is the distance in parsec.*



a) When the abs_mag_V spikes out of range of the plot there is no good solution.

b) When entering the second phase of mass transfer, the stripped star has $M_V = -6.6$ mag, which leads to the maximum distance ~ 21 Mpc for the apparent magnitude to be 25 mag. It would not have been detected in the archival data if it was the progenitor for iPTF13bvn because that galaxy is too far away.

The brightness in the V-band of the supernova progenitor is not only determined by the distance, but also by the temperature of the star – a cooler star emits a larger fraction of its photons in the optical band compared to a hotter star. Since your stripped star model dies during mass transfer, the Roche radius and therefore also the orbital period determines how large and cool the stripped star will be at core-collapse.

Task 7: Varying parameters

a) *Make a copy of your latest run directory and prepare it for modeling a different stripped star. Pick a combination of initial masses (M_1, M_2) and orbital period by signing up in the spreadsheet called `Ylva - Minilab 2`. Prepare your new run directory accordingly. Don't forget to find the new ZAMS model and import it in your inlist.*

b) *The stopping condition for central carbon depletion will no longer work because of the initial composition of the star. You can remove it, or stop and restart as previously.*

c) *Run your model and report the absolute V-band magnitude of the donor star and your supernova type prediction once it enters its second phase of mass transfer in the google sheet.*

2.3 Bonus: High-mass X-ray binaries

We have completely forgotten that the companion is a black hole!

Black holes cannot accrete at infinite rates, but their mass accretion rates are thought to be limited by the so-called Eddington accretion rate. When the accretion rate reaches the Eddington accretion rate, the radiation pressure produced by the accretion is so strong that it can prevent accretion.

The black hole is also moving through stellar wind material as it orbits its companion star. Therefore, the black hole can accrete material also when the stars are detached and mass transfer is not ongoing.

Bonus task 8: Updating black hole accretion

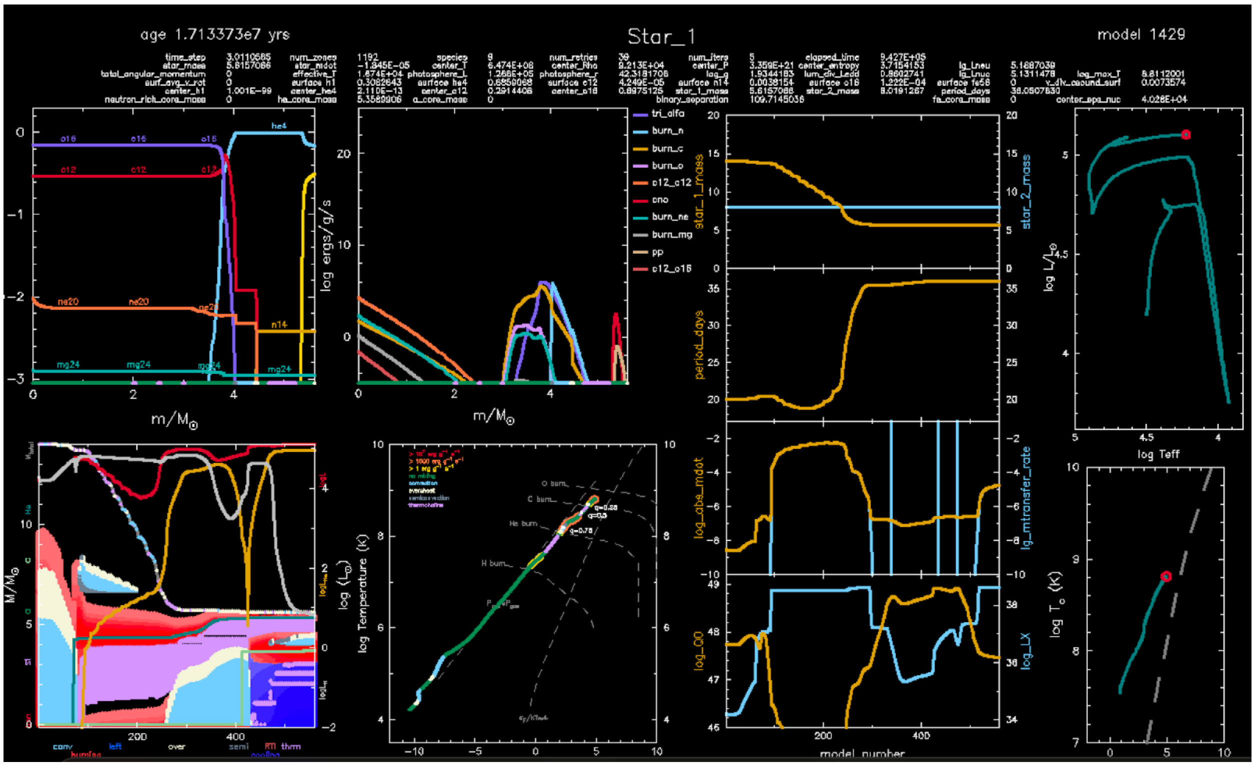
a) *Go back to your model from Tasks 3-6 and update the `inlist_project` so that the accretion is limited by the Eddington accretion rate. (Hint: look through the `binary_controls.defaults` file.)*

b) *Activate wind accretion onto the black hole. Assume that a maximum of 10% of the stellar wind mass loss is accreted at any time.* [a]

[a] In this exercise, to make sure that 10% of the wind mass is accreted, we set `use_radiation_corrected_transfer_rate = .false.` in `inlist_project`.

This goes into inlist_project:

```
! Limit accretion to the Eddington accretion rate
  limit_retention_by_mdot_edd = .true.

! Allow mass transfer via winds
  do_wind_mass_transfer_1 = .true.
  max_wind_transfer_fraction_1 = 0.1
```



As the black hole accretes material, it emits X-rays. In our model, we have the opportunity to explore different stages of the life of a high-mass X-ray binary (HMXB): accompanied by a main sequence star, a stripped star, and detached and semi-detached. First, let's just implement an approximation for how high the X-ray luminosity from the system is.

The X-ray luminosity, L_X, can be approximated by the function:

$$L_X = \epsilon \frac{G M_{\mathrm{BH}} \dot{M}_{\mathrm{acc}}}{R_{\mathrm{acc}}}, \tag{2}$$

where M_{BH} is the mass of the black hole, $\dot{M}_{\mathrm{acc}}$ is the accretion rate, R_{acc} is the radius at which accretion occurs, and ϵ describes how efficient the conversion from accretion to luminosity is. For black holes, we can assume that accretion occurs at $R_{\mathrm{acc}} = 3 \times R_{\mathrm{Schwarzschild}}$, where $R_{\mathrm{Schwarzschild}} = 2GM_{\mathrm{BH}}/c^2$.

Bonus task 9: Tracking X-ray luminosity

a) *Following the simplified calculation for X-ray luminosity above, and assuming that $\epsilon = 0.1$, write a function in* `run_binary_extras.f90`*, which produces the output* `log_LX` *(that is, $\log_{10} L_X$) to the* `binary_history.data` *file. Tip: make sure to keep track of the units and use constants that are available!*

b) *Un-comment the X-ray luminosity in the pgstar window so you can follow the X-ray emission rate with time. Hint: search for EDIT HERE.*

c) *Wipe out the contents of* `grid_png` *and then compile and run your model.*

This function goes into data_for_extra_binary_history_columns:

```
subroutine data_for_extra_binary_history_columns(binary_id, n, names, vals, ierr)
   type (binary_info), pointer :: b
   integer, intent(in) :: binary_id
   integer, intent(in) :: n
   character (len=maxlen_binary_history_column_name) :: names(n)
   real(dp) :: vals(n)
   integer, intent(out) :: ierr
   real(dp) :: epsilon, R_schwarzschild, Racc, LX, log10_LX
   ierr = 0
   call binary_ptr(binary_id, b, ierr)
   if (ierr /= 0) then
      write(*,*) 'failed in binary_ptr'
      return
   end if

   ! Implement the X-ray luminosity estimate for accreting BH
   epsilon = 0.1
   R_schwarzschild = 2*standard_cgrav*(b% m(2))/(clight*clight)
   Racc = 3*R_schwarzschild
   LX = epsilon*standard_cgrav*(b% m(2))*(b% component_mdot(2))/Racc
   if (LX < 1.0) then
      log10_LX = 0.
   else
      log10_LX = log10(LX)
   end if
   names(1) = 'log_LX'
   vals(1) = log10_LX
   !print *, 'LX =', LX

end subroutine data_for_extra_binary_history_columns
```

And you also need to activate the extra binary_history column here:

```
integer function how_many_extra_binary_history_columns(binary_id)
   use binary_def, only: binary_info
   integer, intent(in) :: binary_id
   how_many_extra_binary_history_columns = 1
end function how_many_extra_binary_history_columns
```

Bonus task 10: Mass transfer efficiency *What happens to the mass transfer? Is it conservative (100% of the mass lost from the donor star is accreted by the black hole) or non-conservative? How does it compare to your previous models?*

Mass transfer is almost completely non-conservative, which is completely opposite to the earlier models.

Bonus task 11: X-ray analysis *What different rough X-ray luminosities do you find that the black hole accretion produces for*

a) The main sequence + black hole stage?

b) During the first mass transfer phase?

c) When the black hole orbits a stripped star?

d) During the second mass transfer phase?

a) $\log_{10} L_X \sim 34 - 37$ erg/s

b) $\log_{10} L_X \sim 38.5$ erg/s

c) $\log_{10} L_X \sim 35.5$ erg/s

d) $\log_{10} L_X \sim 38.5$ erg/s

Bonus task 12: Ultraluminous X-ray sources *Ultraluminous X-ray sources (ULXs) have X-ray luminosities above 10^{39} erg/s, does your model ever reach such a stage? What kinds of systems do you expect can become ULXs?*

The model never becomes quite as X-ray bright as a ULX. To be able to reach higher X-ray luminosities, for example either (1) the BH can accrete at super-Eddington mass accretion rates, (2) the BH is more massive and therefore has a higher Eddington accretion rate, or (3) the efficiency parameter is higher than 0.1.

3 Maxilab: Stellar mergers[2]

In the minilabs, we modeled the evolution of systems undergoing mass transfer and the associated envelope stripping. But, about a quarter of all massive stars are expected to merge with their companions, for example when the mass transfer becomes unstable. `MESA` cannot model common envelope evolution, so when a common envelope is expected to be initiated, `MESA` stops. Even though we don't have the full evolution leading to the merger product, we can come up with a possible outcome of a common envelope evolution and then model the future evolution of the merged binary.

In the maxilab, we will:

1. Evolve both stars in a binary starting from their zero-age main-sequences until a common envelope should develop.
2. Make an assumption for how much mass is lost during the merger and how to arrange the interior of the merger product.
3. To prepare the merger product, we will first relax the mass to the mass corresponding to the mass we decided for the merger product. Then, we will relax the interior composition to match our assumptions.
4. Finally, we will evolve the merger product, exploring its properties during life and before death.

3.1 Modeling a merged star

3.1.1 Evolving a binary until coalescence is initiated

Task 1: Get set up. *Download the maxilab starting directory, which contains four folders. Go into* `1_evolve_binary`*, which is the first of several working directories we will use. Switch on the evolution for both stars and choose masses of* $16\,M_\odot$ *for* M_1 *and* $8\,M_\odot$ *for* M_2 *(make sure to use the right ZAMS models for them, you can find other ZAMS models in the folder ZAMS_models). Update the initial period to 500 days. Compile and run the model. During what evolutionary stage and for what reason does your model stop?*

Update the initial conditions in inlist_project:

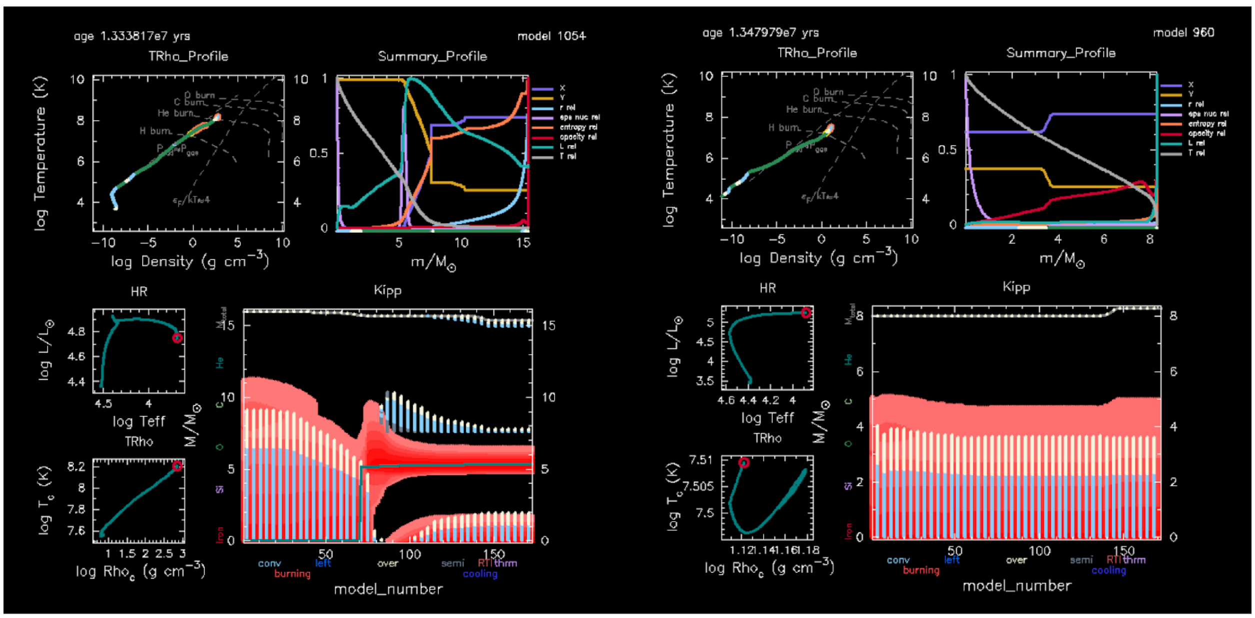
Also update the loaded ZAMS models in inlist1 and inlist2 like this (after having copied them from the folder with ZAMS models):

The donor star has started helium burning in the center and is red and large (340 Rsun). It also has started to develop a convective envelope because it is close to the Hayashi track. The accreting star has become brighter and bloated from the accretion, causing it to also fill its Roche lobe. Because the accretor star fills its Roche lobe, MESA stops.

[2] This lab was inspired by the evolution of merger products described in Renzo et al., (2020).

```
&binary_job

  ! What inlist to use for the first star
    inlist_names(1) = 'inlist1'
    inlist_names(2) = 'inlist2'

  ! Whether to evolve both stars in the system or treat one as a point mass
    evolve_both_stars = .true.

/ ! end of binary_job namelist


&binary_controls

  ! Initial conditions for the binary
    m1 = 16.0d0  ! donor mass in Msun
    m2 = 8.0d0 ! companion mass in Msun
    initial_period_in_days = 500.0d0
```

```
&star_job

   ! Load ZAMS model
     load_saved_model = .true.
     saved_model_name = 'zams_16Msun.mod'
```



Whether accretor stars can stably accrete the material they are fed during Hertzsprung gap Roche lobe overflow is uncertain. The later during the Hertzsprung gap the donor fills its Roche lobe, the faster is the flow of material as the thermal timescale of the donor decreases, making it more and more difficult for the accretor star to accrete the material. Here, let's assume that the two stars in the binary you just created actually will lead to the development of a common envelope and a subsequent merger.

> ***Task 2: Merger mass*** *Assuming no mass is lost during the merger, what will be the mass of the merger product?*

23.7 $M_\odot$

3.1.2 Constructing a fake merger product

`MESA` has capabilities to create and evolve many different types of structures, not all necessarily physically motivated. To mimic a merger product, let's use this capability and fake the resulting star.

To be able to evolve a merger product, we must first create the model to start from. To do this, we begin with making a model with the right mass, and then improving it to also have the composition we want.

> ***Task 3: Faking a merger.*** *We need to create a starting model that has the total mass of the merger product. To do that, use the work directory* `2_relax_mass`*. Start with studying the* `inlist_project` *file.*
> *We need a model to start from, it doesn't matter very much which one, but we have a model with quite close mass created from the previous step – the final model for the donor. Start the mass relaxation from this model and also adjust the total mass that you would like to reach. Compile and run the model. See how you reach the appropriate mass in the terminal.*

```
save LOGS/profile1.data for model 1055
save photos/x055 for model 1055
 saved to relaxed_mass.mod
termination code: max_model_number
---------------------------------------------------------------------------------------------------------------------------

       step    lg_Tmax     Teff     lg_LH      lg_Lnuc     Mass      H_rich     H_cntr     N_cntr     Y_surf   eta_cntr   zones  retry
   lg_dt_yr    lg_Tcntr    lg_R     lg_L3a     lg_Lneu     lg_Mdot   He_core    He_cntr    O_cntr     Z_surf   gam_cntr   iters
     age_yr    lg_Dcntr    lg_L     lg_LZ      lg_Lphoto   lg_Dsurf  C_core     C_cntr     Ne_cntr    Z_cntr   v_div_cs       dt_limit
---------------------------------------------------------------------------------------------------------------------------

       1055   8.209733   3943.130   4.977102   5.120229  23.700000  18.373522   0.000000   0.003426   0.252000  -3.909521   1524      0
  -2.476449   8.209733   2.890977   3.731701   3.810187 -99.000000   5.326478   0.993923   0.000079   0.006000   0.031750      5
 1.3338E+07   2.840863   5.120012   4.500233 -99.000000  -8.671538   0.000000   0.000230   0.001119   0.006077  0.000E+00  max increase
```

The entropy inside stars typically increases with radius. Entropy can be resembled with the ability to float – higher entropy material floats easier and therefore ends up on the surface. When assembling the two merging stars and deciding what the composition structure of the merger product should be, sorting it according to the specific entropy of the zones is, therefore, a method that has been used when modeling merger products (e.g., Lombardi et al. 2002).

Task 4: Adjusting the composition. *You have now produced a model with the appropriate mass, but the composition is still not updated. To do that, we need to also relax the composition according to a predetermined structure that is provided in a text file. For this exercise, use the work directory `3_relax_composition`. Begin with studying the `inlist_project` file and locate where the relaxation is treated.*

a) Fill in and set up so that the simulation will start from the model with correct mass.

b) You can see that we need to provide a text file with a composition structure. You can either create it yourself if you have python installed on your computer - then run the python script called `construct_merger_composition.py` inside your work directory `1_evolve_binary`. Move the newly created composition file to `3_relax_composition`. You can also use pre-created composition files located in the folder `merger_composition_files`. If you use the pre-created composition file for this exercise, pick the one called `merger_composition_16_8_HG.txt`.

c) Compile and run the model in `3_relax_composition`. See how the composition of the star changes.

Entropy sorting can give rise to an erratic composition structure, which can be seen in the pgstar window. This is a known behavior, but probably not completely physically realistic. However, as you notice during the composition relaxation, the merger product develops a deep convective envelope (look at the Kippenhahn diagram), which quickly will smoothen the strange composition profile in the next step when we evolve the model with time.

3.1.3 Evolving the merger product

Task 5: Evolve the merger product. *Now your model is ready to be evolved with time, so pick the resulting model from your composition relaxation, `merger_start.mod`, and place it inside the directory `4_evolve_merger`. Import the model as a starting model. Set the model to stop when the central helium mass fraction reaches 0.5. Compile and run your model.*

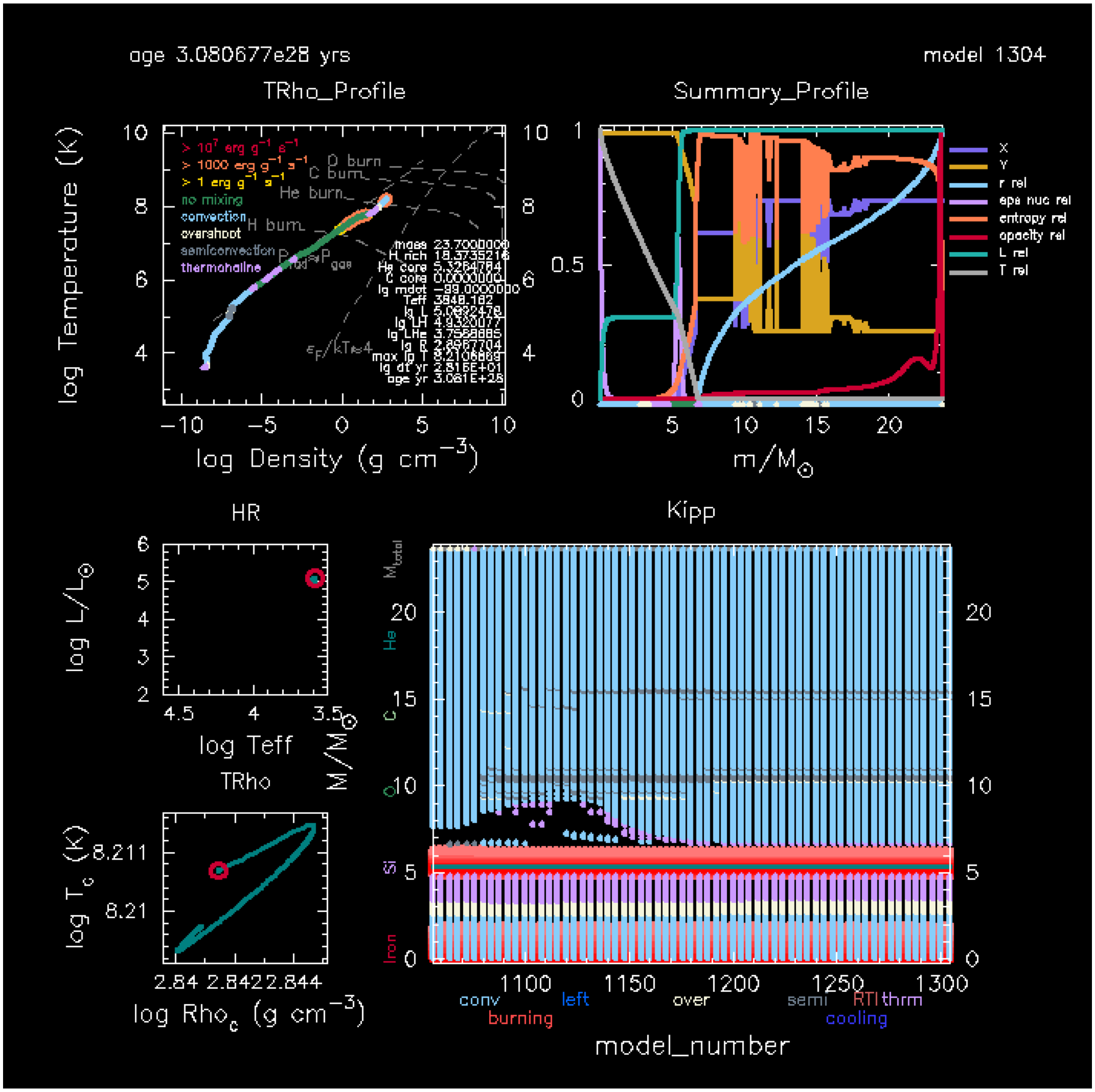
Task 6: Analyze the merger product.

a) What happened to the merger product?

b) What regions of the star are convective?

c) What is the temperature? What is the radius? What is its surface gravity?

d) Do you think the merger product could be confused with a single main sequence star?

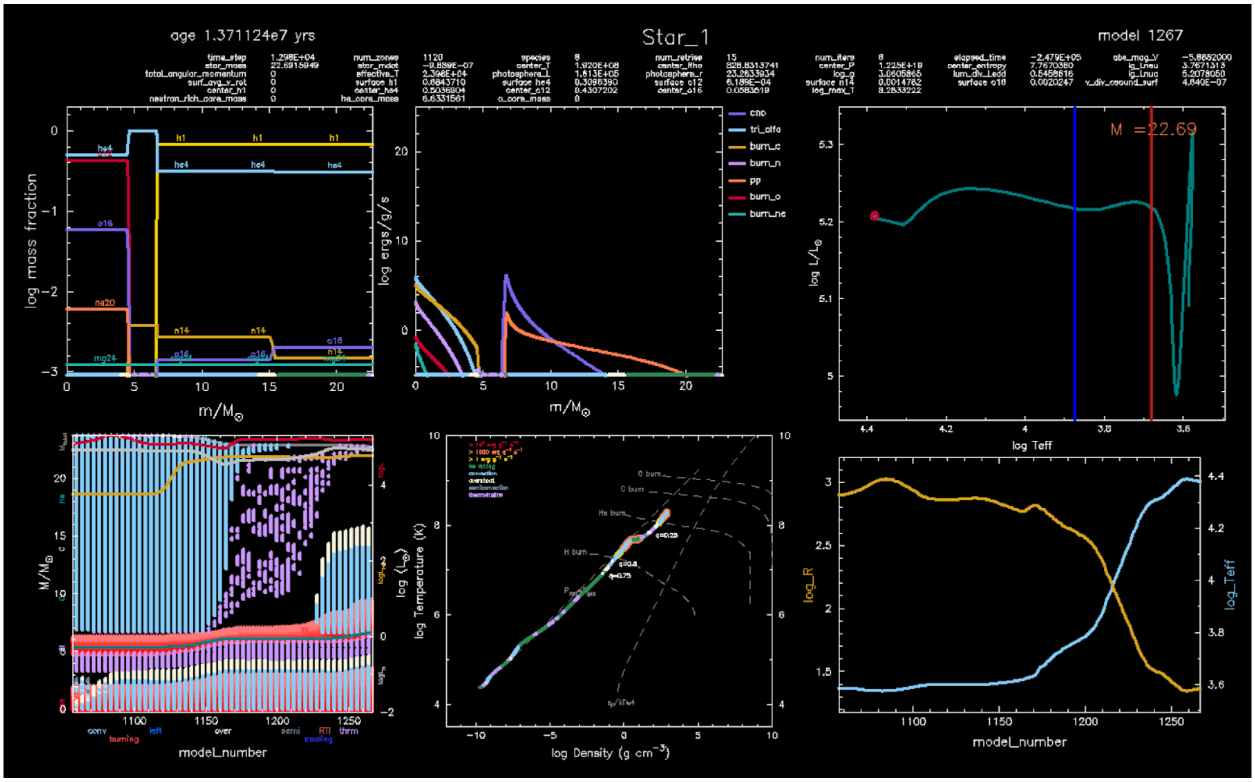


a) The star contracted and heated up. The composition profile has smoothened.

b) During central helium burning, the helium-burning core is convective and also the hydrogen-burning shell. The convective layer covering the hydrogen-burning shell extends over $\sim 8\,M_\odot$ and therefore reaches into the non-burning envelope as well.

c) The effective temperature is $\sim 24,000$ K, the radius is $\sim 23R_\odot$ and the surface gravity is $logg \sim 3$.

d) Apart from the surface gravity being somewhat low for main sequence stars, the effective temperature and radius match well with the expectations for main sequence stars. The luminosity is also similar. This star could most likely be confused with a single main sequence star.



The contraction of merger products is thought to result from the mass gain: suddenly the helium core constitutes a much lower fraction of the total stellar mass compared to before. A thick convective layer has also been discovered in stellar evolutionary models that burn helium as blue supergiants (e.g., Klencki et al. 2020). These characteristics to the interior structures of merger products differ from what is expected for single stars with the same temperatures and luminosities.

Bonus task 7: How to recognize it? *The helium-core burning stage is the second longest evolutionary stage of a star, after the main sequence when the star burns hydrogen in the center. Assuming merger products contract like yours, can you think of ways to distinguish them from regular, single, main-sequence stars?*

On the surface, the merger products seem similar to main sequence stars, possibly with a little low surface gravity. With data from large spectroscopic surveys it may be possible to identify a subgroup of hot stars with lower surface gravity. However, this could be hard. Asteroseismology could provide the best constraints, since with pulsations it could be possible to characterize the core size and possible convective layers.

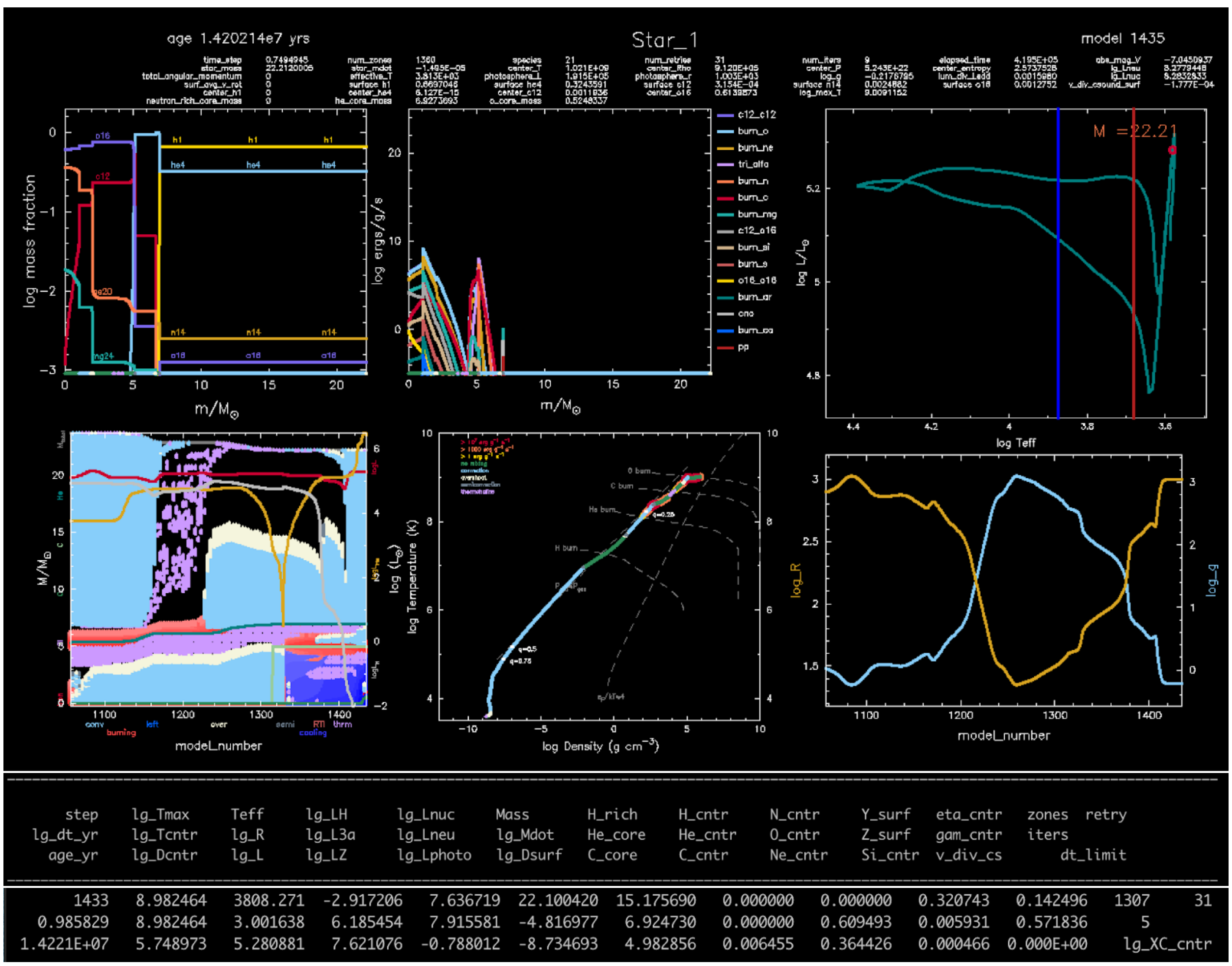
3.1.4 Final evolutionary stages

The merger product is massive enough to eventually undergo core-collapse. How does this supernova progenitor look like? Here are some rough characteristic temperatures for massive stars (adopted from Drout et al., 2009):

- Blue supergiant (BSG): $>$ 7,500 K
- Yellow supergiant (YSG): 4,800 - 7,500 K
- Red supergiant (RSG): $<$ 4,800 K

Task 8: Evolve towards death. *Change the stopping condition in your model to central carbon depletion and restart your merger product. What happens to the star? According to the giant star definitions above, what type of star do you think it is at death?*

The star expands and cools again, reaching $\sim 1,000 R_\odot$ and 3,800 K. The star develops again a deep convective envelope. At death, this merger product should be a RSG. Maybe there is extra ejecta mass?



3.2 Blue supergiants as merger products - from life to death

Do all merger products behave like the model you just created? This could mean that regular single stellar population synthesis models under-predict the number of BSGs, since massive single stars are thought to burn helium as RSGs, while your merger product burns helium as BSG. Merger products have even been suggested as responsible for the blue supergiant progenitors of some supernovae, like SN 1987A (e.g., Podsiadlowski et al., 1990, Menon et al., 2017). Let's explore how common these blue supergiant phases may be and whether they can last to the end stages of stellar life.

Task 9: Change component masses *Make a new merger product with different initial component masses: each person at a table chooses a new combination of masses among 18 & 10, 18 & 9, 16 & 10, 16 & 6, and 11 & 8* $M_{\odot}$*. Follow the same steps as previously. For the models to run smoothly, make sure to use the masses from the terminal output when calculating the total mass of the merger product. If you use the pre-computed composition profiles, make sure to pick the one corresponding to your initial component masses. What is the stellar type of your merger product (1) during helium core burning, and (2) at death?*

3.3 Coalescence at different evolutionary stages

Binary stars can merge, for example, because of too long or too short an orbital period, or because of extreme mass ratio. We have explored scenarios when the merger occurs because the binary has a very wide orbit - now let's look at how the merger product evolves if the stars merged already during the main sequence evolution.

Task 10: Evolve a main-sequence merger *Start from your first merger model, but change the initial period to 1.5 days and follow the same steps (tasks 1-5). There is a pre-computed composition file for this main sequence merger in the folder* `merger_composition_files` *called* `merger_composition_16_8_MS.txt`*, but you can also compute your own with the python script. Does the merger product still burn helium as a blue supergiant? Does the evolution in the HR diagram remind you of something? Note: this model takes a minute to find a good starting point, but it will get going.*

It does not burn helium as BSG - in fact, its evolution is very similar to that of a regular single massive star.

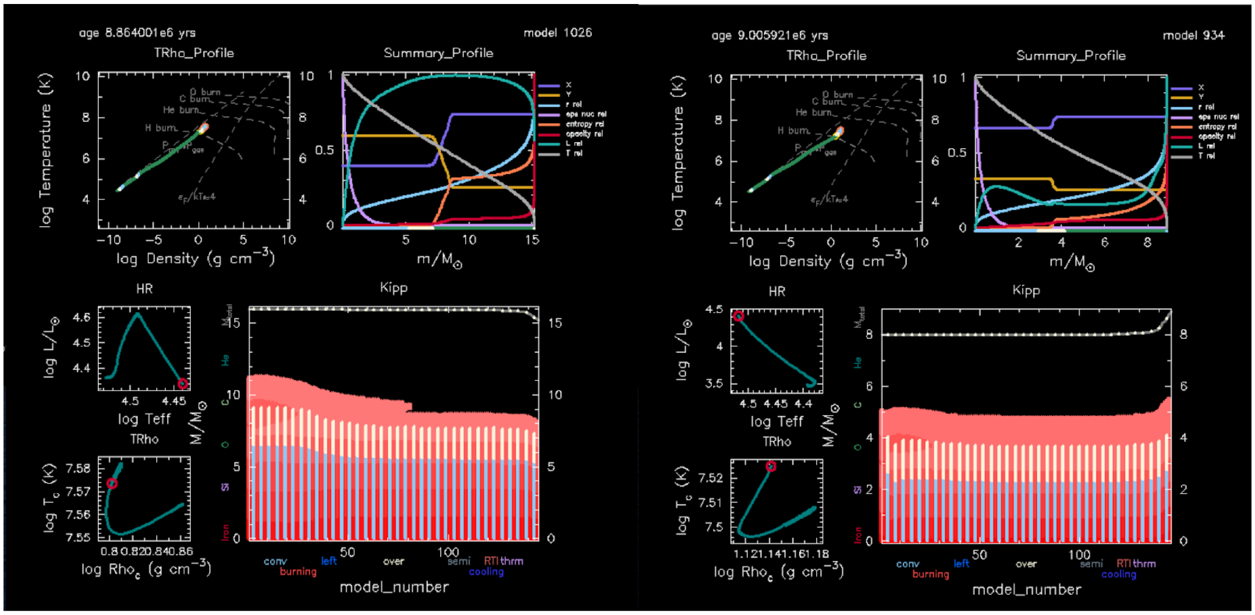
Binary at model termination:

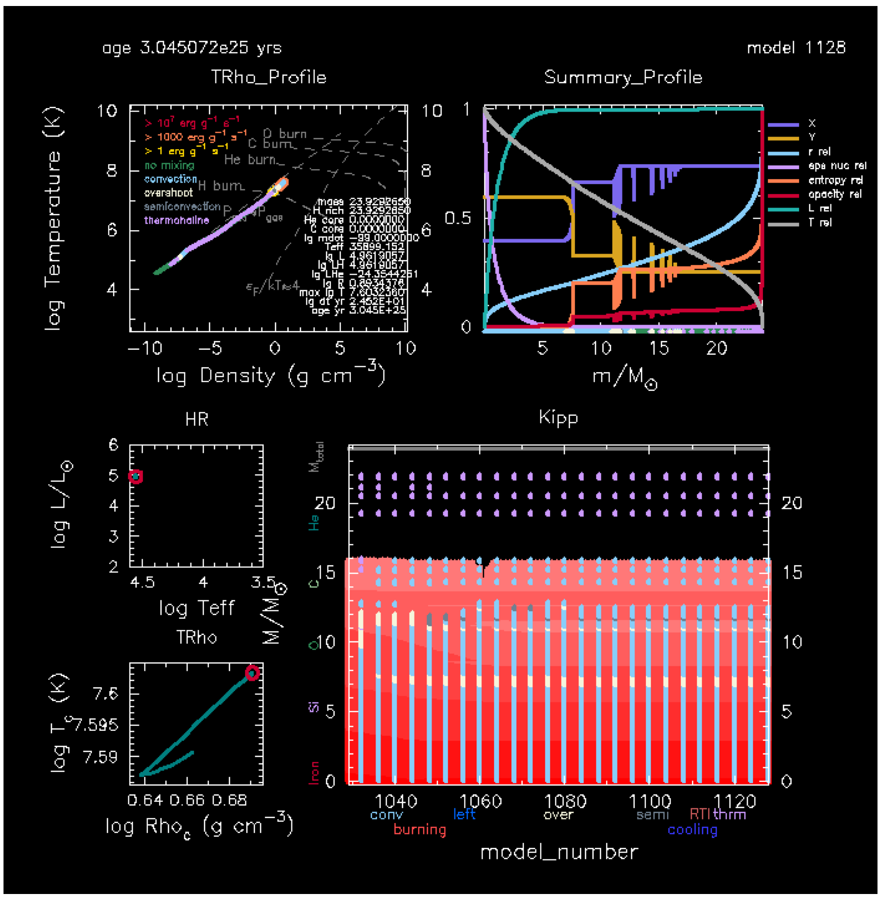
During composition relaxation:

The evolution of the merger product (during He burning - slow because of wind Mdot):

3.4 Bonus: Rotation of merger products

After the coalescence of two stars, some angular momentum is thought to remain, causing the merger product to spin. Let's invoke some rotation in the merger product before evolving it further.

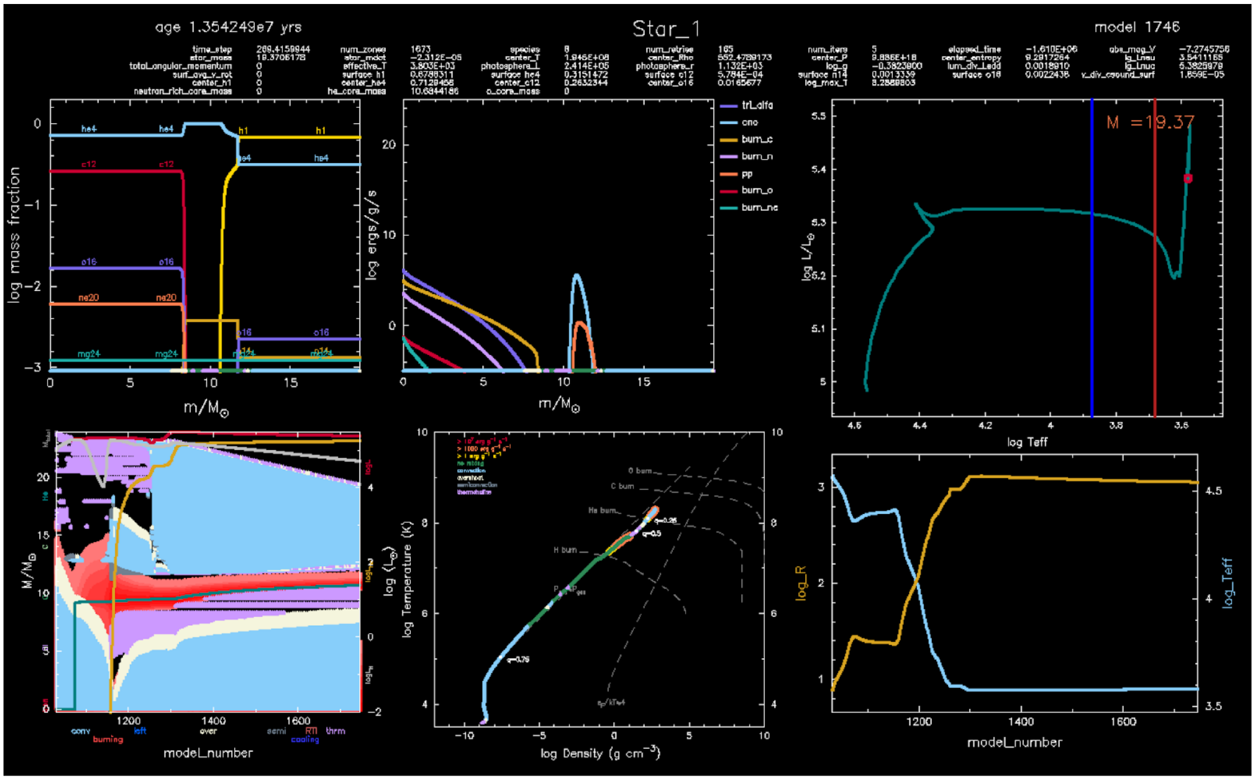




Bonus task 11: Invoke rotation.

a) To make sure that the model we invoke rotation in has fully adjusted, start with evolving `merger_start.mod` *for 10 model numbers and save a model at the end of the run.*

b) Look through `star_job.defaults` *for how to make the star rotate. Change the rotation flag to a new rotation flag. Set the spin rate (in omega/omega_crit) to 0.5. Assume 60 steps of relaxation for the model to reach the right spin rate.*



a) My new model is called rot_prep.mod - it starts from the mass- and composition-relaxed model merger_start.mod. It only runs for 10 models - enough to smoothen out weird composition gradients before invoking rotation.

```
&star_job

  ! Load saved model
    load_saved_model = .true.
    saved_model_name = 'merger_start.mod'

    save_model_when_terminate = .true.
    save_model_filename = 'rot_prep.mod'

  ! display on-screen plots
    pgstar_flag = .true.
    pause_before_terminate = .true.

    set_initial_model_number = .true.
    initial_model_number = 0

/ ! end of star_job namelist
```

```
&controls

  ! Output
    profile_interval = 1000
    history_interval = 1
    terminal_interval = 1
    write_header_frequency = 10

    max_model_number = 10
```

b) Inside star_job: Start from rot_prep.mod and also initiate rotation.

c) The content is commented in the standard inlist_pgstar.

```
&star_job

  ! Load saved model
    load_saved_model = .true.
    saved_model_name = 'rot_prep.mod'

    save_model_when_terminate = .true.
    save_model_filename = 'end.mod'

  ! display on-screen plots
    pgstar_flag = .true.
    pause_before_terminate = .true.


  ! Invoke some rotation
    change_rotation_flag = .true.
    new_rotation_flag = .true.
    new_omega_div_omega_crit = 0.5
    set_omega_div_omega_crit = .true.
    relax_omega_div_omega_crit = .true.
    num_steps_to_relax_rotation = 60


/ ! end of star_job namelist
```

```
! ! ! ! ! DISPLAY ROTATION  ! ! ! ! ! !
! The second panel
History_Panels1_yaxis_name(2) = 'surf_avg_v_rot'
History_Panels1_other_yaxis_name(2) = ''
! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !
```

d)

Bonus task 12: Spin-down. *What was the initial equatorial rotation rate of the merger product? What is it at the end of the central hydrogen burning phase?*

Initially: 150 km/s, at TAMS: 25 km/s.