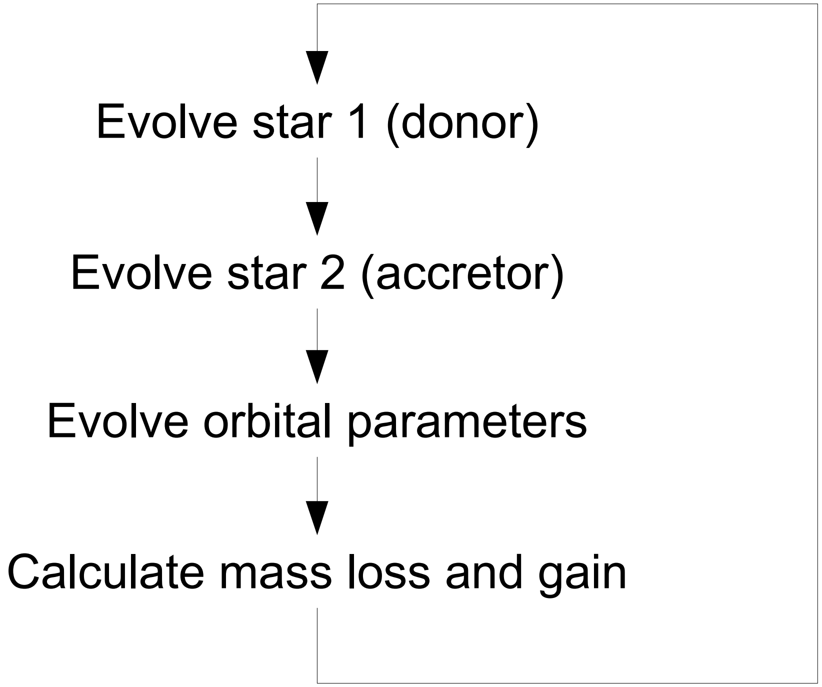
MESA

Week 3: Binary evolution

MESA Binary - simplified

MESA binary combines the single star evolution code found in `mesa/star` for two stars, and adds a binary module on top of this:



Evolve star 1 (donor)

Evolve star 2 (accretor)

Evolve orbital parameters

Calculate mass loss and gain

The binary part of MESA can be found in: `mesa/binary`, and has the same structure as the single star part.

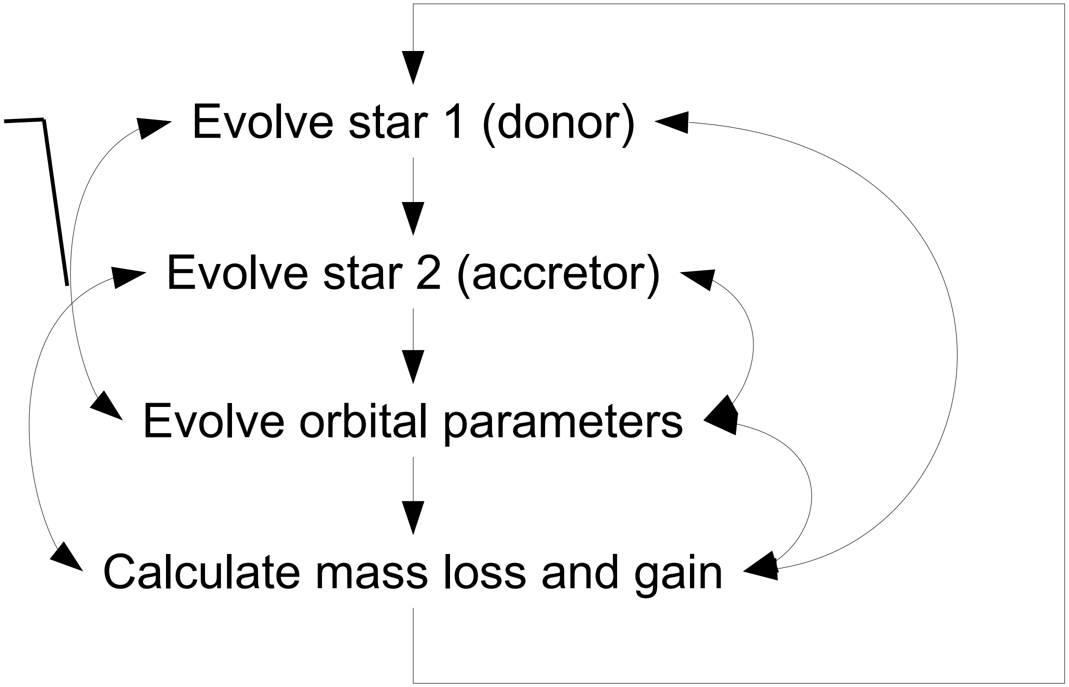
A standard working directory: `mesa/binary/work`

List of examples (test suite): `mesa/binary/test_suite`

MESA Binary - reality

MESA binary combines the single star evolution code found in `mesa/star` for two stars, and adds a binary module on top of this:

There is a lot of switching between modules to set variables.



- In theory donor and accretor can switch roles, but in practice it is easier to assign each star a fixed role.

- In theory Mesa can evolve 3 or more stars, but in practice this is not implemented.

Binary: file structure

Evolve 2 stars at the same time → double the amount of input files + binary controls

```
inlist                   # base inlist for mesa
inlist_project           # inlist with binary parameters

binary_history_columns.list # binary parameters to save

history_columns.list        # star parameters to save for history
profile_columns.list        # star parameters to save for profiles

binary_history.data     # mesa output for binary
```

Star 1 files - donor

```
LOGS1
photos1
inlist1
```

star 2 files - accretor

```
LOGS2
photos2
inlist2
```

Binary: file structure

Evolve 2 stars at the same time → double the amount of input files + binary controls

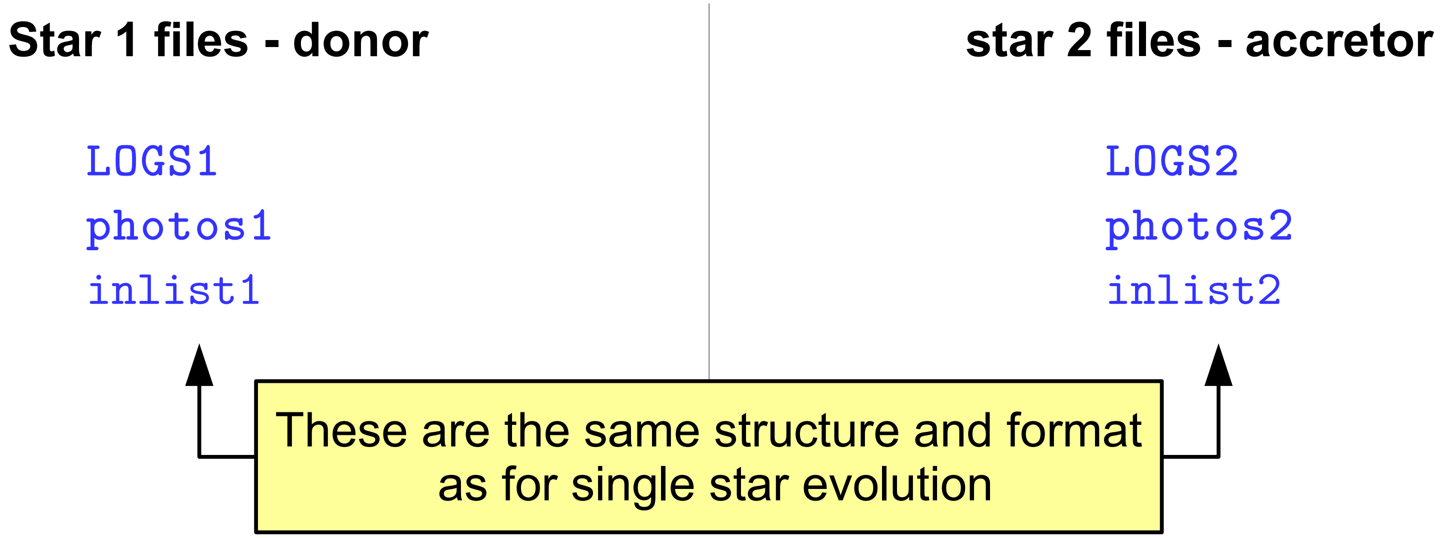
```
inlist                # base inlist for mesa
inlist_project        # inlist with binary parameters

binary_history_columns.list # binary parameters to save

history_columns.list        # star parameters to save for history
profile_columns.list        # star parameters to save for profiles

binary_history.data    # mesa output for binary
```

Not in the LOG directory!



Star 1 files - donor

```
LOGS1
photos1
inlist1
```

star 2 files - accretor

```
LOGS2
photos2
inlist2
```

These are the same structure and format as for single star evolution

Binary: input

In the inlist_project file: define the binary parameters and the links to the stellar input parameters

Different names then single star environments!

```
&binary_job

   inlist_names(1) = 'inlist1'  ! input file for star 1
   inlist_names(2) = 'inlist2'  ! input file for star 2

   evolve_both_stars = .true.   ! you can evolve star 1, or both stars

/ ! end of binary_job namelist

&binary_controls

   m1 = 1.0d0  ! donor mass in Msun
   m2 = 0.8d0  ! companion mass in Msun
   initial_period_in_days = 2d0

/ ! end of binary_controls namelist
```

~~&binary_pgstar~~

Does not exists yet!

Binary: input

`&binary_job`: Only define inlists for stars, and whether to evolve both stars or not. There are no other settings here.

```
&binary_job

   inlist_names(1) = 'inlist1'   ! input file for star 1
   inlist_names(2) = 'inlist2'   ! input file for star 2

   evolve_both_stars = .true.    ! you can evolve star 1, or both stars

/ ! end of binary_job namelist
```

MESA can tread the companion (accretor) as a point mass.

```
&binary_controls

   m1 = 1.0d0   ! donor mass in Msun
   m2 = 0.8d0   ! companion mass in Msun
   initial_period_in_days = 2d0

/ ! end of binary_controls namelist
```

Binary: input

`&binary_controls`: Initial setup, output parameters, mass loss settings, ...

```
&binary_job

   inlist_names(1) = 'inlist1'   ! input file for star 1
   inlist_names(2) = 'inlist2'   ! input file for star 2

   evolve_both_stars = .true.    ! you can evolve star 1, or both stars

/ ! end of binary_job namelist

&binary_controls

   m1 = 1.0d0   ! donor mass in Msun
   m2 = 0.8d0   ! companion mass in Msun
   initial_period_in_days = 2d0

/ ! end of binary_controls namelist
```

Remember: mesa/binary/defaults   and   http://mesa.sourceforge.net/

Binary: Output

Binary parameters are only history parameters (There is nothing to make a profile of)

Indicate in `binary_history_columns.list` which parameters to save

Binary parameters are saved in `binary_history.data`

```
model_number ! model number of donor star
age          ! age of donor star

period_days        ! orbital period in days
binary_separation ! orbital separation in rsun
eccentricity       ! orbital eccentricity
J_orb              ! orbital angular momentum

rl_1             ! roche lobe radius of first star in rsun
lg_mstar_dot_1 ! log10 of first star abs(mdot) in Msun/secyer
lg_wind_mdot_1 ! log10 of first star abs(mdot) due to winds in Msun/secyer
...
```

Binary: pgplot

Plotting parameters are defined in the pgstar section of the first star: **inlist1**

Binary parameters are history parameters → use history plot

```
&pgstar

      History_Panels1_win_flag = .true.
      History_Panels1_win_width = 5

      History_Panels1_title = 'Orbital evolution'
      History_Panels1_num_panels = 2

      History_Panels1_yaxis_name(1) = 'period_days'
      History_Panels1_other_yaxis_name(1) = 'lg_mstar_dot_1'
      History_Panels1_yaxis_name(2) = 'Jdot'
      History_Panels1_other_yaxis_name(2) = 'binary_separation'

/ ! end of pgstar namelist
```

Binary: pgplot

Plotting parameters are defined in the pgstar section of the first star: **inlist1**

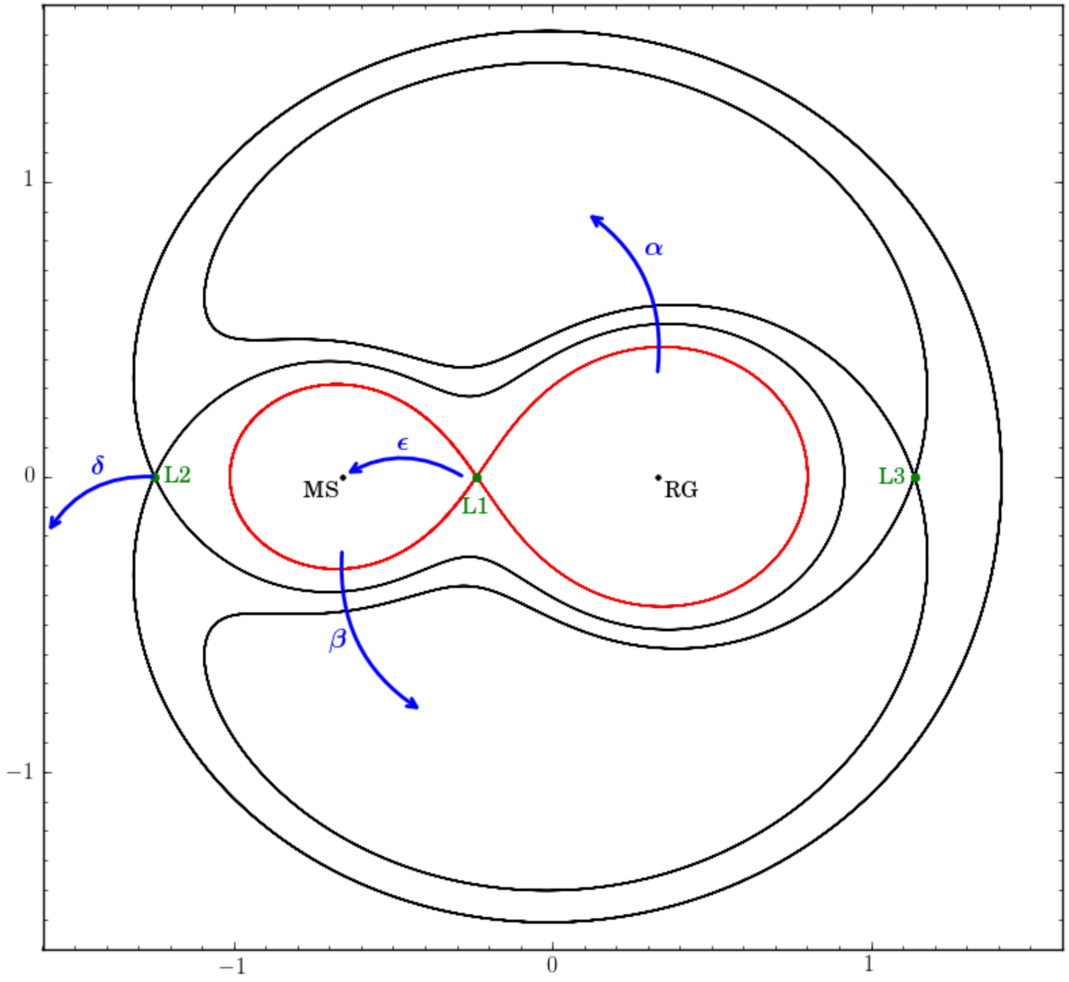
Binary parameters are history parameters → use history plot

For pgstar to access the history output, you need to also write the history output of the binary module to the stellar history files:

```
    append_to_star_history = .true.
```

This is turned on by default.

Binary: mass-loss fractions



$$1 - \alpha - \beta - \delta = \epsilon$$

`Alpha` : Mass lost in fast wind around donor

`Beta` : Mass transferred to companion, and then lost in a fast wind

`Delta` : Mass lost through the outer lagrange point L2

`Epsilon` : Accreted mass

Exercise 3: Roche lobe overflow

Roche lobe overflow (RLOF) during the RGB is an important process that creates many ‘special’ types of systems. In this exercise you will start from a system containing a main sequence star and a red giant on a wide orbit (P ~ 600 days), and study the effect of the mass loss fractions on the orbital parameters.

Pick a set of alpha, beta and gamma so that their sum is 1 (no accretion). Evolve the system till after RLOF starts, and check what happened with the orbital period and the mass loss rate.

Hints:

We will only evolve the donor star (to speed up)

You need to setup the initial parameter so that the RGB model is loaded for the donor star, and the companion is a regular main sequence star.

Also set:

```
terminate_if_initial_overflow = .false.
mdot_scheme = 'Kolb'
max_explicit_abs_mdot = 1d-2
mass_transfer_gamma = 1.2
```