CAP - report of assignment 3

Lennart van Sluijs, Marco TvdB

November 20, 2017

1 Introduction

For assignment 3 we evolved the triple system θ Muscae in time where both stellar evolution and gravitational dynamics were incorporated.

2 Methods

Our time scheme uses the interlaced temporal discretization method: Firstly, we evolve the stars at time t_0 for a certain time step $dt/2$ and obtain their new masses. The masses are used to run the gravitational integrator for a time dt as from t_0. Lastly, we evolve the stars again for a time $dt/2$ such that both the gravity code and stellar code have now been fully evolved up to a time $t_0 + dt$. We ran the simulations for a total of 0.1 Myr.
Now in our implementation dt is variable: we perform much smaller time steps $d\tau$, where $d\tau \ll dt$, as long as the mass loss $\dot{m} < \dot{m_{\rm lim}}$ does not exceed the limiting mass loss of $\dot{m_{\rm lim}}$. For example if after n iterations $\dot{m} < \dot{m_{\rm lim}}$ we use $dt = nd\tau$. Therefore, one can think of $d\tau$ as the minimal time step we use. We used $d\tau = 15 \times 10^{-9}$ Myr as our minimal time step. We ran three simulations in total and named them A, B and C. For these simulations we only changed the limiting mass loss $\dot{m_{\rm lim}}$. A higher limiting mass loss will use larger time steps dt. The parameters used per simulation are summarized in Table 1.
All simulations A, B and C run with an initial inclination of 30 degrees. However, in reality we do not know the inclination of the system. To test the system for it's dependency on the inclination we ran another simulation, D, with the same specs as A, except for the initial inclination which is now set as 70 degrees (thus bigger than 50 degrees).

3 Results

Our results are summarized in the Figures below.

4 Discussion

The total runtime, time in AMUSE framework and percentage of time spent in the AMUSE framework are shown in Table 2. As expected A runs the longest since it uses the smallest time steps and C the largest, since the mass loss rate limit is the highest. We can also see that the longer the simulations runs relatively more time is spent in the integrator rather than the AMUSE framework. First we want to check if the time stepping scheme worked. Figure 1 shows indeed the time per timestep differs between the simulations. The shortes timesteps are for simulation A where no mass loss limit is specified and the smallest timesteps of size $d\tau$ are used all the time. As expected simulation B uses slightly bigger timesteps, because it has a higher mass loss rate

Simulation	$\dot{m_{\rm lim}}$
A	0
B	1×10^{-6}
C	3×10^{-6}

Table 1: Maximum mass loss rates per simulation. The mass loss rate is given in $M_\odot$/Myr.

Simulation	Runtime [s]	Time in AMUSE framework [s]	% in AMUSE framework
A	51040	11106	21.7
B	32527	8486	26.1
C	10712	4149	38.7

Table 2: Global parameters of all three simulations.

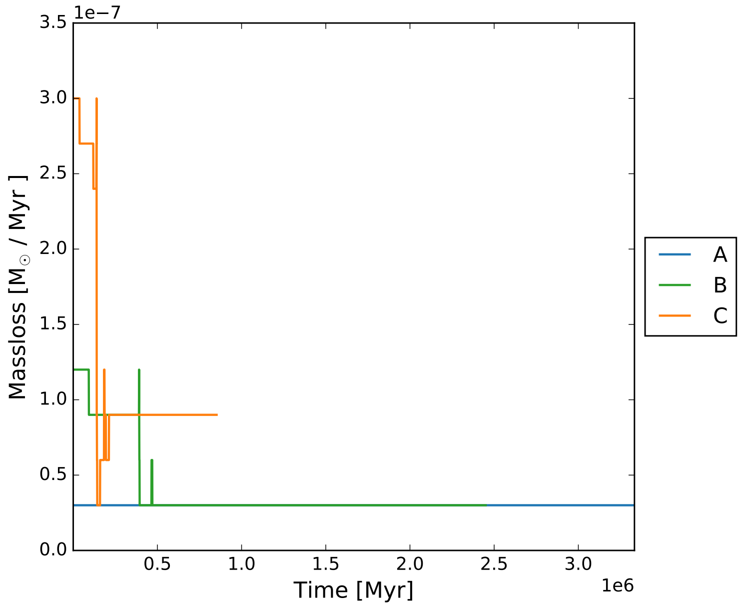

Figure 1: Time per timestep as a function of the timesteps for all simulations.

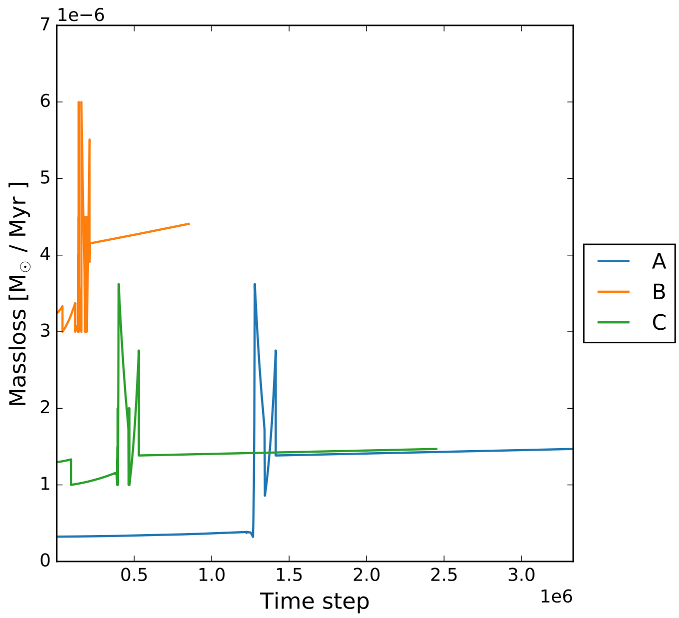

Figure 2: Massloss per timestep as a function of the timesteps for all simulations.

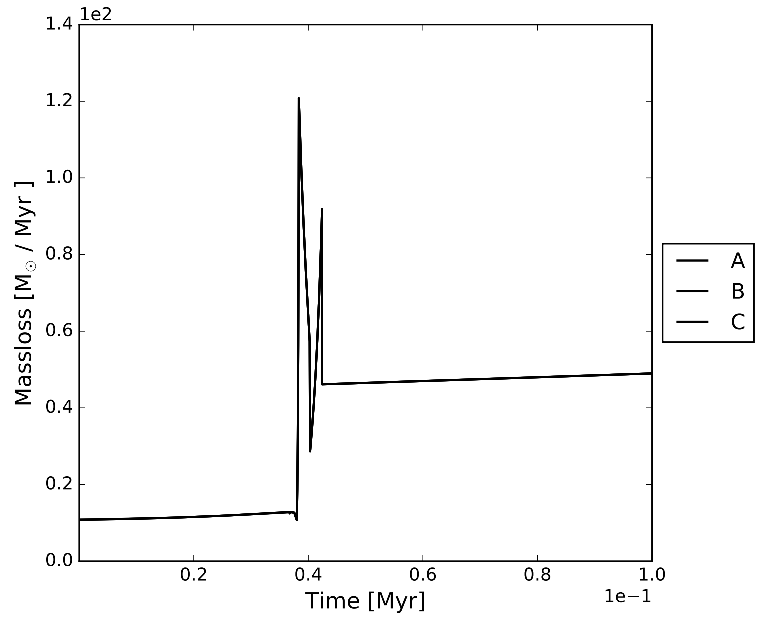

Figure 3: Massloss as a function of the simulation time for all simulations.

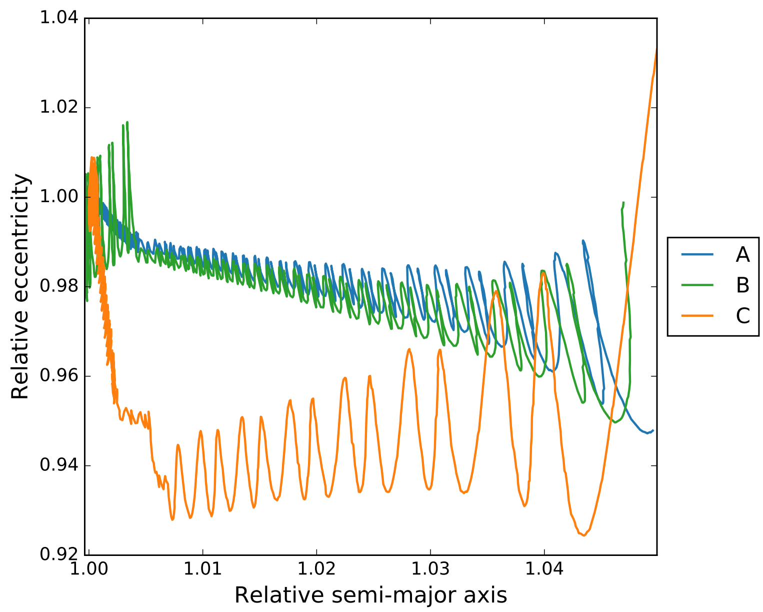

Figure 4: Inner relative eccentricity versus semi-major axis for all simulations.

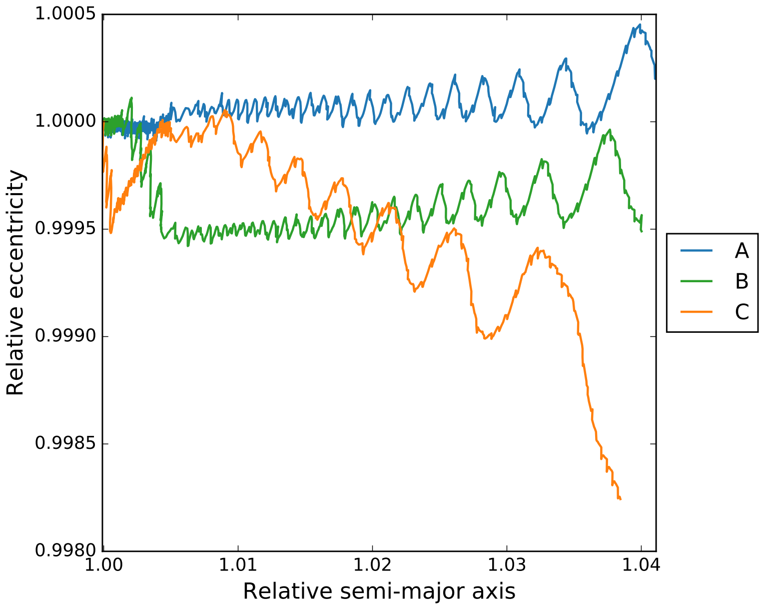

Figure 5: Outer relative eccentricity versus semi-major axis for all simulations.

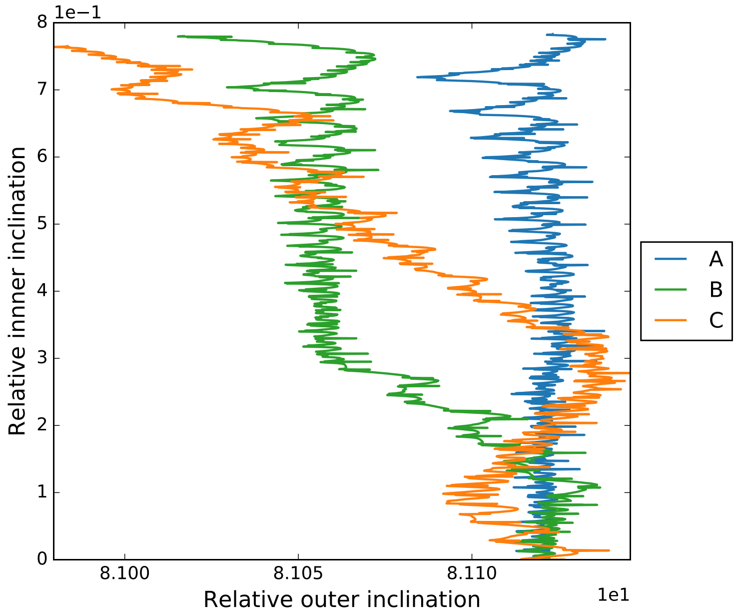

Figure 6: Inner inclination versus outer inclination for all simulations.

threshold. C has the highest maximum mass loss rate per time step and therefore uses the biggest timesteps. It can be seen in Figure 1 simulation A ran much longer than C, because of the smaller time steps.

Figure 2 shows the mass loss per simulation as a function of the timestep. Indeed mass loss per timestep is much lower on average for simulation A compared to B and C respectively. All simulation show peaks in the mass loss rate for some time steps. This deviates from expected continous mass loss for normal stellar evolution indicating that the parameterizes steller evolution code SeBa might not be good enough for this problem.

To check if the mass loss rate per time (not timestep) was the same for all simulations we also plotted this for all simulations in Figure 3. Indeed all simulation have the same mass loss rate per unit time, thus we can trust the results presented above.

Figures 4 and 5 show how the relative semi-major axis and eccentricity change over time for the inner and outer pair. None of the simulation converge to a stable system. We can see that as expected simulation A has a result that is closed to a stable system. For the inner relative eccentricity and semi-major axis simulation A and B almost converge to the same result. Thus in this case we are able to speed up the simulation and still converge to the same result. For the outer semi-major axis and eccentricity however, the results between A and B already differ more. For simulation C the results differ significantly from A and B: simulation C has likely a too high mass loss rate threshold and therefore very large time steps are used which reduce the accuracy of the results.

Figure 6 shows the inner and outer inclination angles during the simulations. Again here A performs the best where at least one of the inclination angles stays quite stable over the full simulation lifetime.

After running simulation D with a high inclination angle the system quickly destabilizes. If the system. Therefore, if θ Muscae is indeed a stable system, lower mutual inclinations are expected.