FALL BACK DISKS AND THE END OF THE COMMON ENVELOPE PHASE

By

Rajika Kuruwita

A THESIS SUBMITTED TO MACQUARIE UNIVERSITY
FOR THE DEGREE OF MASTER OF RESEARCH
DEPARTMENT OF PHYSICS & ASTRONOMY
DECEMBER 2014

MACQUARIE
UNIVERSITY
SYDNEY ~ AUSTRALIA

EXAMINER'S COPY

Typeset in LaTeX 2_ε.

Except where acknowledged in the customary manner, the material presented in this thesis is, to the best of my knowledge, original and has not been submitted in whole or part for a degree in any university.

Rajika Kuruwita

Acknowledgements

First and foremost I would like to thank my supervisors Associate Professor Orsola De Marco and Dr Jan Staff. They have been supportive and patient throughout my masters research project. Thank you Jan for running my simulations and providing invaluable feedback for all my work. Thank you Dr. De Marco for listening to me rattle off my train of thoughts, and believing in me. Orsola is Superwoman in my eyes and I hope to one day achieve as much as she has.

Thank you to Jean-Claude Passy, whose work has been fundamental to my thesis. It has been a pleasure to work with you.

I would also like to thank the Think Tank Kids: Blake, Glen, Matt and Reece, and other masters colleagues, Brint, Christina and Tiffany for making this time enjoyable. I will miss the late nights of assignments, Game of Thrones and Coon and Goon throughout our masters degree. Thank you to all my other friends at Macquarie University including Joao, Andrew, Jacob, James, Shane and Vincent. A special thank you to Joao, for always being around, supporting me through tough times and for all the fun we've had together. Thank you for believing in me and pushing me a little bit more, getting me one step closer to being like Lara Croft.

Not that they would really appreciate this, but I would like to thank my cat Ginger (I guess I should thank Abby too). Pets seem to always know when you need a friend and companion. Growing up with Ginger for the last 10 years, he has always been my big fat fuzzy baby and I can't imagine life without him.

Finally thank you to my father, mother and brother, who despite my unorthodox career choice, have always stood behind me and helped provide the opportunities to follow my dreams. They have admired and enjoyed all my hobbies from dancing and cosplay to music. I am grateful for their support.

Abstract

The common envelope interaction scenario was first proposed by Paczynski (1976) to explain the existence of cataclysmic variables. The interaction involves a binary system, where the more massive primary star extends its gas envelope as it becomes a giant, engulfing its companion. The companion star is believed to help with the expulsion of this envelope, resulting in the post common envelope binaries that we observe today in cataclysmic variables and central stars of planetary nebula. The existence of such systems can not be explained without the need of a common envelope interaction, and for many decades this was the only support for this interaction. A common envelope interaction is expected to be short, and hence very difficult to observe. The duration of an event is on the order of the dynamical time scale of the giant. Due to the short time scale of this event and how little the composition of the giant is expected to change, it is ideal to simulate such an event using hydrodynamical code.

As computers have become powerful enough to run common envelope simulations to sufficient accuracy, many attempts have been made at simulating this. However these simulations fall short in producing the characteristics of observed post common envelope binaries. In particular, they do not succeed in unbinding much of the envelope, and the final separation in the simulations are larger than what is observed in post common envelope binaries. It has been propose that physics that has not been accounted for in simulations may be necessary to reproduce observations, for example, recombination energy.

In this thesis we propose another mechanism to make up for the shortfall of previous studies, and investigate it via simulations. As previous simulations have found that most

($\geq 60\%$) of the giant envelope is still bound to the system, our hypothesis is that the fall back of bound gas onto the binary system can lead to a further reduction of the orbital separation while unbinding more of the envelope gas. Our simulations show that the orbital separation is effectively reduced by a fall-back event. However, only a small amount of mass is unbound. We propose that a number of fall-back events are necessary (and likely inevitable) but that another energy source may be needed to fully unbind the envelope.

Contents

1

Introduction

Stellar evolution has been studied for decades and today a fairly sound understanding of how a single star evolves has been achieved. However, it has been known for centuries that stars can exist in systems where they are gravitationally bound to another star (a binary star; Herschel 1802). The fraction of all stars that are members of binary systems depends on the mass of the system. As shown in Figure 1.1 (Bouy, 2011) low mass stars (≤ 1.0 M$_\odot$) are more likely to be single and higher mass stars (≥ 1 M$_\odot$) are more likely to be in binaries. Sana et al. (2012) found 71% of O-type stars belonged in binary systems that will interact as some point in the life of the system. Fifty percent of solar-type stars (with effective temperature between 4800 and 6550K and luminosity between 0.2 to 3.0L$_\odot$) are in binary systems (Raghavan et al., 2010). Lower mass stars have a multiplicity of 30% (Lada, 2006).

If two stars in a binary system are sufficiently close together, interactions between the two stars will occur. This will usually happen when one component evolves and expands

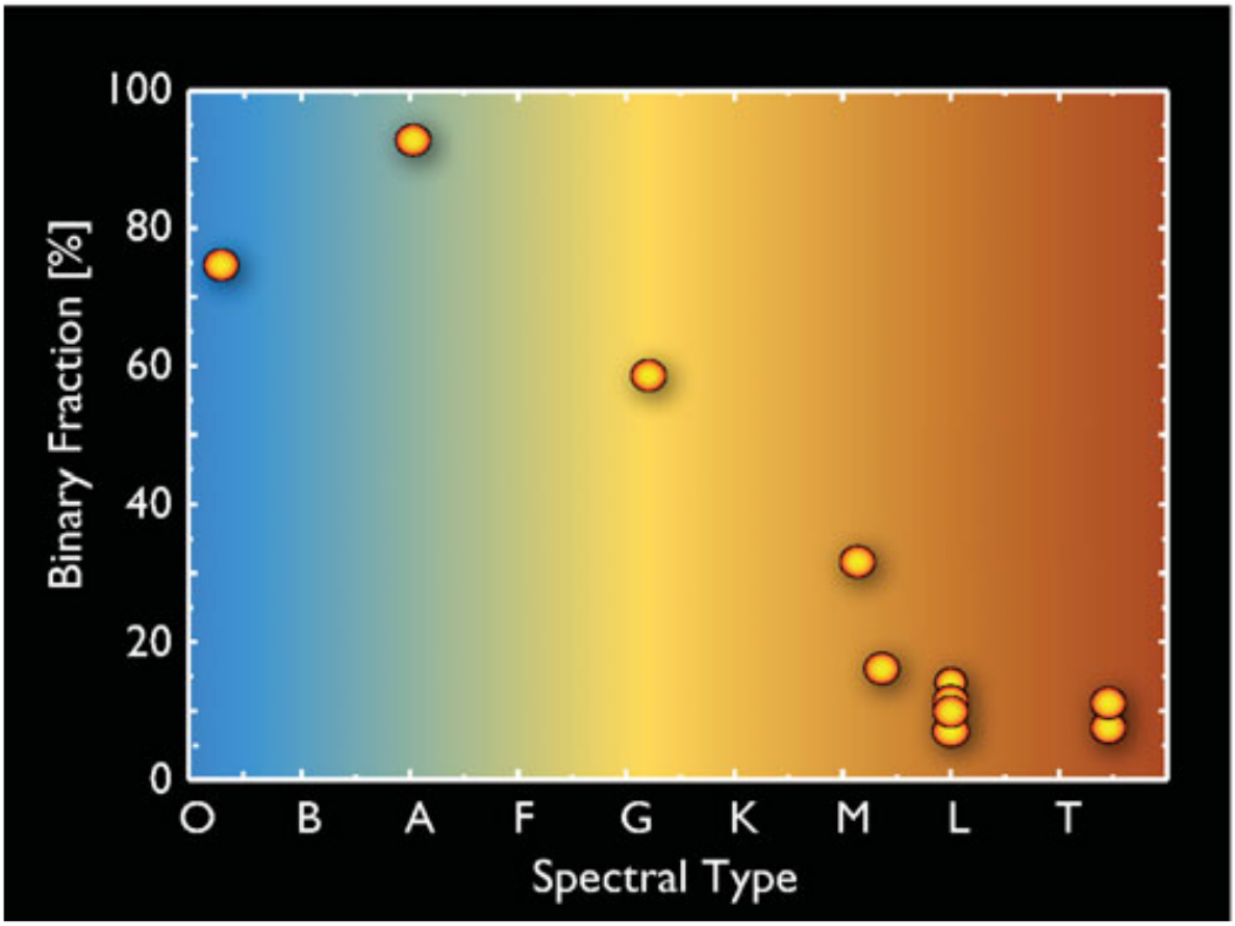


Figure 1.1: Multiplicity frequency as a function of Spectral Type found from a number of different studies. (Bouy, 2011).

on to the red giant branch (RGB) or asymptotic giant branch (AGB). These interactions may involve the stable transfer of mass between the components, or if one star overflows and transfers more mass than the companion can accrete, a common envelope can occur (common envelope formation will be discussed later).

Even if a star is solitary, it may harbour planetary companions (over 1000 stars have been found to host close-in planets so far with the *Kepler Space Telescope*, and more are being discovered every day). When these parent stars evolve and expand, they may engulf or interact with their planets. This means that these stars are also likely to experience some sort of interaction some time during their life. The influence of planets on the evolution of stars and binary systems is discussed further in Section 1.1.4.

While we have studied the evolution of stars extensively, it has often been under the assumption that they are solitary. With our current understanding of binary stars and planet populations, we should realise we have ignored scenarios that will lead to a larger number of evolutionary pathways for a star: the interaction with a companion whether it be stellar or planetary. Understanding these interactions is crucial for understanding a variety of observed systems and events. Such events include outbursts, mass transfer and engulfing of companions, all leading to poorly understood physical phenomena.

In this chapter we focus on the common envelope binary interaction and investigate the envelope removal process which is a prerequisite to the survival of binary systems. First we will briefly describe some astronomical phenomena that are best explained by this phase. Particularly we will discuss cataclysmic variables (CVs), Type Ia supernovae and how the common envelope may play a role in a relatively new class of phenomena known as intermediate luminous optical transient (ILOTs). CVs and Type Ia supernovae have been known for decades, and have been studied extensively, however it is only recently that ILOTs have been actively searched for and studied. This is partly due to recent time-domain surveys such as the Catalina Real-time Transient Survey (CRTS), (Drake et al., 2009) and the Palomar Transient Factory (PTF), (**?**), which are actively searching for these objects.

1.1 Compact, Evolved Binaries and Merger Products

There is a variety of systems and events that are best explained by binaries interacting or having interacted, such as cataclysmic variables, X-ray binaries and Type Ia supernovae progenitors. Some interacting binaries may result in a merger of the two stars into a single star. Mergers have been proposed as an explanation for other stars and events, such as ILOTs, R Coronae Borealis (RCB) stars, and massive WDs. Here we will describe the common envelope binary interaction and how it explains systems such as those that have been listed.

1.1.1 Cataclysmic Variables

CVs are systems that increase in luminosity by a few orders of magnitude and may do so multiple times, irregularly or semi-regularly in outburst events known as novae or nova-like. These objects typically consist of a white dwarf (WD) primary, and a companion (usually a main sequence star) that is transferring mass to it. In order for mass transfer to occur between the stars the two components must be sufficiently close to each other, so these variables have periods usually of the order of hours. It is impossible for a system like this to have always existed, because the WD is an evolved star. Therefore it must have had a radius larger than the current orbital separation of the system. Therefore, the companion must have been farther away in the past than it is today. The question is then how did the orbital separation shrink so much?

The unusually small separation of the binaries that produce CVs can be explained with a common envelope in the following way (Paczynski, 1976). The binary initially had a wider separation, wide enough that the system did not interact. However, when the primary expanded during one of its giant phases, it filled its Roche lobe. At that point gas from the primary would flow through the first Lagrangian point (the point between the primary and secondary where the co-rotating gravitational potential forms a "saddle") leading to mass transfer onto the companion. If this mass transfer is sufficiently fast, the secondary may not accrete it and eventually it may expand to fill its own Roche lobe. If this continues, the mass overflows both Roche lobes resulting in a common envelope. In this envelope the secondary and primary's core orbit one another and lose angular momentum and energy to the envelope gas due to gravitational drag, also known as dynamical friction (Ostriker, 1999). What method of energy and angular momentum exchange dominates in these interactions has been the matter of some debate (Ricker & Taam, 2008). The loss of angular momentum and energy to the envelope results in the secondary and primary's core falling into a tighter orbit. The energy given to the envelope "lifts" it, increasing the luminosity of the system. The envelope becomes unbound, it leaves behind the secondary and the primary core, which is now a white dwarf, in the close orbit observed today. These types of systems with a WD in a tight binary are called post-common envelope binaries.

Paczynski (1976) proposed that if the envelope of the primary is lost in a common envelope interaction, it may result in planetary nebula with a close binary in the center. The hot WD in the center would be ionising the expanding envelope. With this prediction, Paczynski proposed a simple observational test of this hypothesis: to look for a short period binaries as a nuclei of planetary nebulae. Short period binaries as central stars of planetary nebulae have been observed, with approximately 15-20% of all central stars being post common envelope binaries (Bond 2000; Miszalski et al. 2009), showing that 1 in 5 to 1 in 6 planetary nebulae are ejected common envelopes. These binaries have periods between a few hours and ~3 days. Further evolution of the period at the hand of magnetic breaking and gravitational wave radiation is expected to reduce the periods to the values observed in CVs.

1.1.2 Type I Supernovae

Cataclysmic variables are not the only phenomena to be explained by Paczynski's common envelope hypothesis. The production of Type Ia supernovae also requires a common envelope phase to produce their progenitors.

Type Ia supernovae were first classified based on their spectral features and light-curves. Type I supernovae do not have hydrogen in their spectra. The Type Ia is distinguished from types Ib and Ic because of singly ionised silicon lines in its spectrum. The current understanding of the physical mechanism that generates Type Ia supernova is the growth of a WD to the Chandrasekhar mass limit. There are currently two scenarios to achieve this: in the first, the progenitor is a single degenerate, or a carbon-oxygen WD with a main sequence or giant star companion. The second, is a double degenerate progenitor, or two WDs in a close orbit (Webbink 1984; Nomoto et al. 1985, Iben & Tutukov 1999).

The single-degenerate scenario was proposed because Type Ia supernovae lack hydrogen in their spectra and this implies that the progenitor must have lost its hydrogen envelope. Additionally, the elements produced in this type of explosion result from the ignition of degenerate material (Hoyle & Fowler, 1960). From these criteria a WD is an ideal candidate as a progenitor. However, getting the WD to explode posed another problem, as there is no reason a stable degenerate star should explode. Hence a close binary system was proposed, as it is capable of stripping a giant of its envelope and the components are close enough that mass transfer onto a degenerate WD occurs. This progenitor is similar to the binaries that go through CV outbursts. However, it is not the heated accreting mass, nor the surface burning that creates the outburst in Type Ia Supernovae. Nomoto et al. (1985) describe how the rate of mass accretion determines how the WD responds; accretion rates of less than 10^{-8} $\mathrm{M}_{\odot} yr^{-1}$ will produce a strong hydrogen flash on the surface, ejecting accreted material in a nova-like fashiom. Accretion rates larger than this value will instead produce a weaker flash, leading to fusion into helium, increasing the WD mass. Even larger mass transfer rates will not allow mass accretion onto the white dwarf and instead leads the Roche lobe of the companion is filled. The weaker hydrogen flashes occur multiple times as mass is accreted. Once the WD reaches the Chandrasekhar limit (≈ 1.4 $\mathrm{M}_{\odot}$) it explodes as a Type Ia supernova. This explosion leaves no remnant star, unlike core-collapse supernovae, but the companion star will remain. In the double-degenerate scenario the two WDs spiral in towards each other and eventually merge. This scenario would leave no remnant star. Some advocate this scenario because in several supernovae remnants the companion star was absent (Schaefer & Pagnotta, 2012).

The ability to standardise supernova light-curves has made these objects pivotal in measuring cosmological distances and hence contribute to our showing the accelerating expansion of the universe (Riess et al. 1998, **?**). However, despite the wide use of Type Ia supernovae, there is still significant variation in their light-curve peaks, and duration, which are all dependent on the progenitor of the supernova. Historically, these supernovae were believed to be relatively consistent in luminosity and duration, as the progenitor is limited by the Chandrasehkar mass. However, the existence of sub-luminous, sub-Chandrasehkar detonations and possible super-Chandrasehkar detonations and super-luminous Type Ia supernovae has made the matter complex (Sullivan et al. 2011; Garca-Senz et al. 2007; Woosley & Weaver 1994). These variations in the progenitors translate into uncertainties in our extra-galactic measurements using Type Ia supernovae. It has also been realised that our understanding of binary stellar evolution and nuclear physics is badly lacking.

The rate of WD-WD mergers (1.4×10^{-13} $\mathrm{yr}^{-1}\mathrm{M}_{\odot}^{-1}$ predicted using Monte Carlo simulations; Badenes & Maoz 2012) is found to be similar to the total Type Ia supernova rates found observationally (1.1×10^{-13} yr^{-1}M$_\odot^{-1}$; (Maoz & Mannucci, 2012)) for a galaxy like the Milky Way. However this calculated rate does not distinguish between those merging systems that detonate or not. For super-Chandrasekhar mergers, Badenes & Maoz (2012) predicted the rate to be 1.0×10^{-14} yr^{-1}M$_\odot^{-1}$, which is significantly smaller than the overall calculated merger rate mentioned. As a result Badenes & Maoz (2012) concluded that super-Chandrasekhar mergers cannot account for most of the Type Ia supernovae caused by WD-WD mergers, hence sub-Chandrasekhar detonations must occur. The error bars of the predicted rates tend to be ill defined and highly reliant on the prescription of the common envelope interaction in the population synthesis codes.

With such uncertainty in the mechanism causing Type Ia supernovae, we need to calculate accurate single and double degenerate supernovae rates. All these theoretical estimates are affected by our ignorance of the common envelope interaction, which prevents us from knowing the orbital separation after the common envelope: a smaller separation leads to a shorter time to interaction. Predicting small separations leads to more binary mergers and to larger estimated supernova Type Ia rates.

1.1.3 Intermediate Luminosity Optical Transients

Supernova and nova outbursts can be separated on a luminosity-time diagram. In Figure 1.2 we can see that there is a clear distinction between the two types of event (Soker & Kashi 2012; Kasliwal 2012). The supernovae have larger luminosities, starting at $M_V = -14$ mag for a period of a few tens of days to higher luminosities of < -17 mag. Novae fill a luminosity band on the luminosity-time diagram between $M_V = -9$ to $M_V = -7$ mag, with durations of a few days. This leaves a strip between the two events called the Optical Transient Strip (OTS), which is populated by objects and events that do not fall into the supernova or nova categories. These objects are classified as Intermediate Luminosity Optical Transients (or ILOTs; also known as Red Novae, or Intermediate Luminous Red Transients). ILOTs are characterised by their cool temperature and steady rise in luminosity. The duration of an ILOT event is of the order of few weeks ($\sim 10 - 100$ days; Kasliwal 2012).

Evidence is mounting that at least some ILOTs are the product of stars interacting

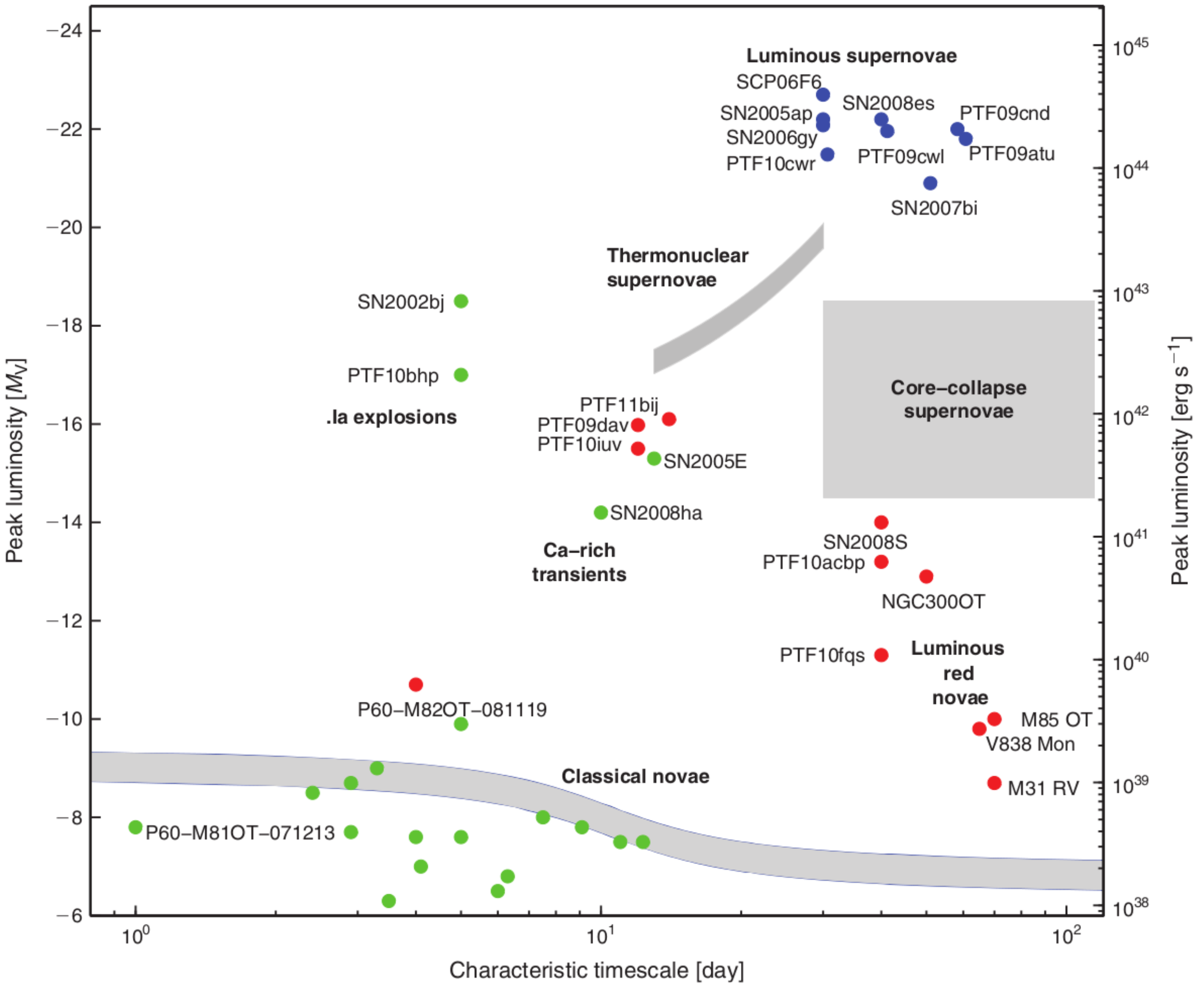


Figure 1.2: Framework for optical transients in 2011. Note that until 2005 we only knew of about three classes (denoted by grey bands). In the past six years, systematic searches, serendipitous discoveries and archival searches have uncovered a plethora of novel, rare transients. Several new classes are emerging and the governing physics is being widely debated (Kasliwal, 2012).

or merging. Examples include V838 Mon, M85 OT 2006 (Rau et al., 2007) and NGC300 OT2008 (Kashi et al., 2010). Here we shall focus on the example of V1309 Sco, whose outburst was discovered in September 2008 (Figure 1.3) and is not only a perfect example of an ILOT, but it also provides proof of having been caused by a merger between two stars.

V1309 Sco's progenitor was observed before the outburst and its light-curve shows a periodic variability with an amplitude of ∼0.2 mag and a period of ∼1.4 days (Tylenda

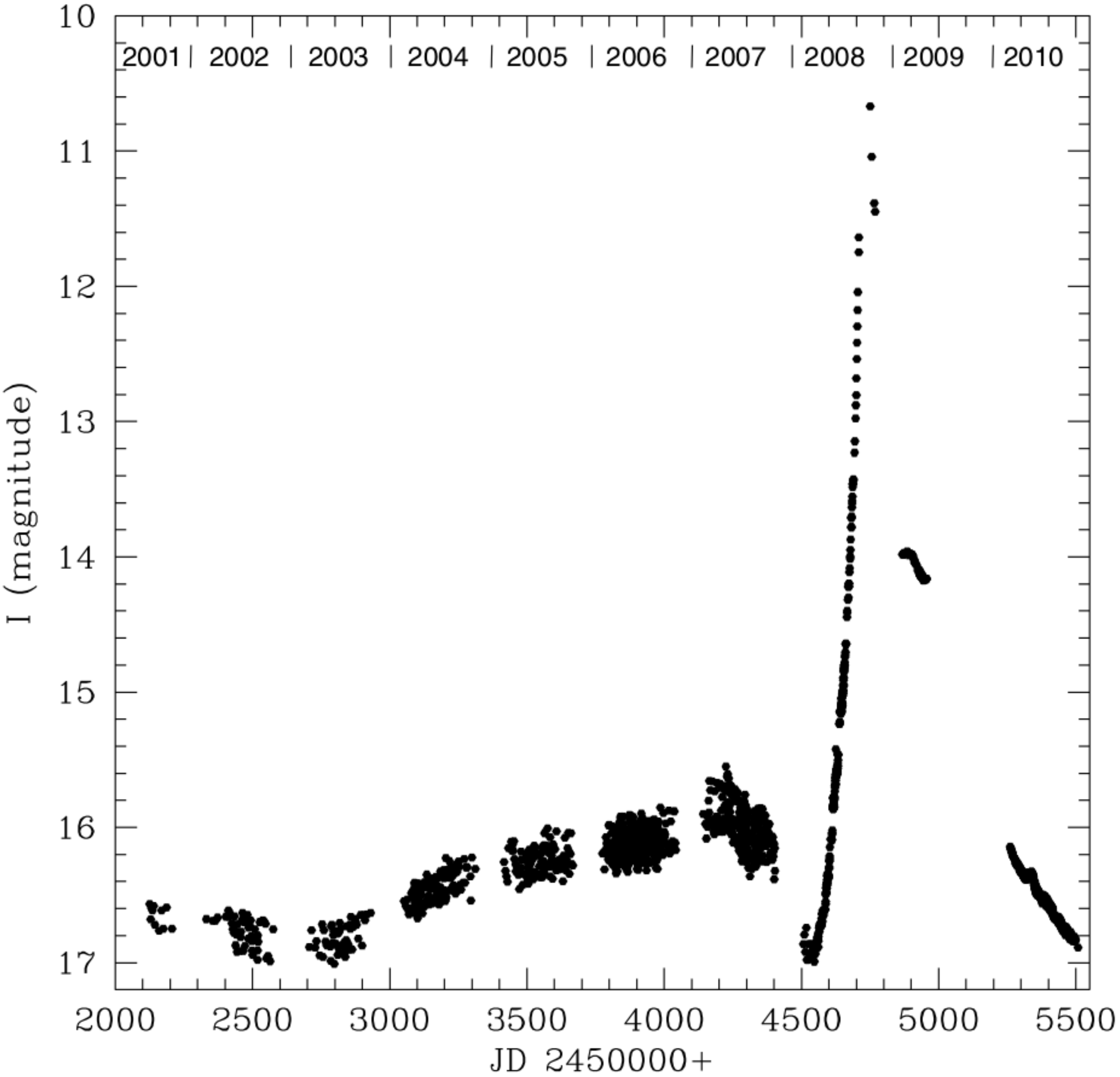


Figure 1.3: The I-band light-curve of V1309 Sco from the OGLE-III and OGLE-IV projects: Time in years is marked on top of the figure. At maximum light the object attained $I = 6.8$. (Tylenda et al., 2011).

et al., 2011). However, the variability behaviour observed in the outburst progenitor could not be explained by a pulsating or a rotating single star. The classical nova scenario, which involves stellar interactions, also failed as an explanation for the outburst as it results in a hot compact object, instead of a cool giant observed in the case of the post-outburst V1309 Sco and other events of this type (Tylenda & Soker, 2006). By chance this object was in the field

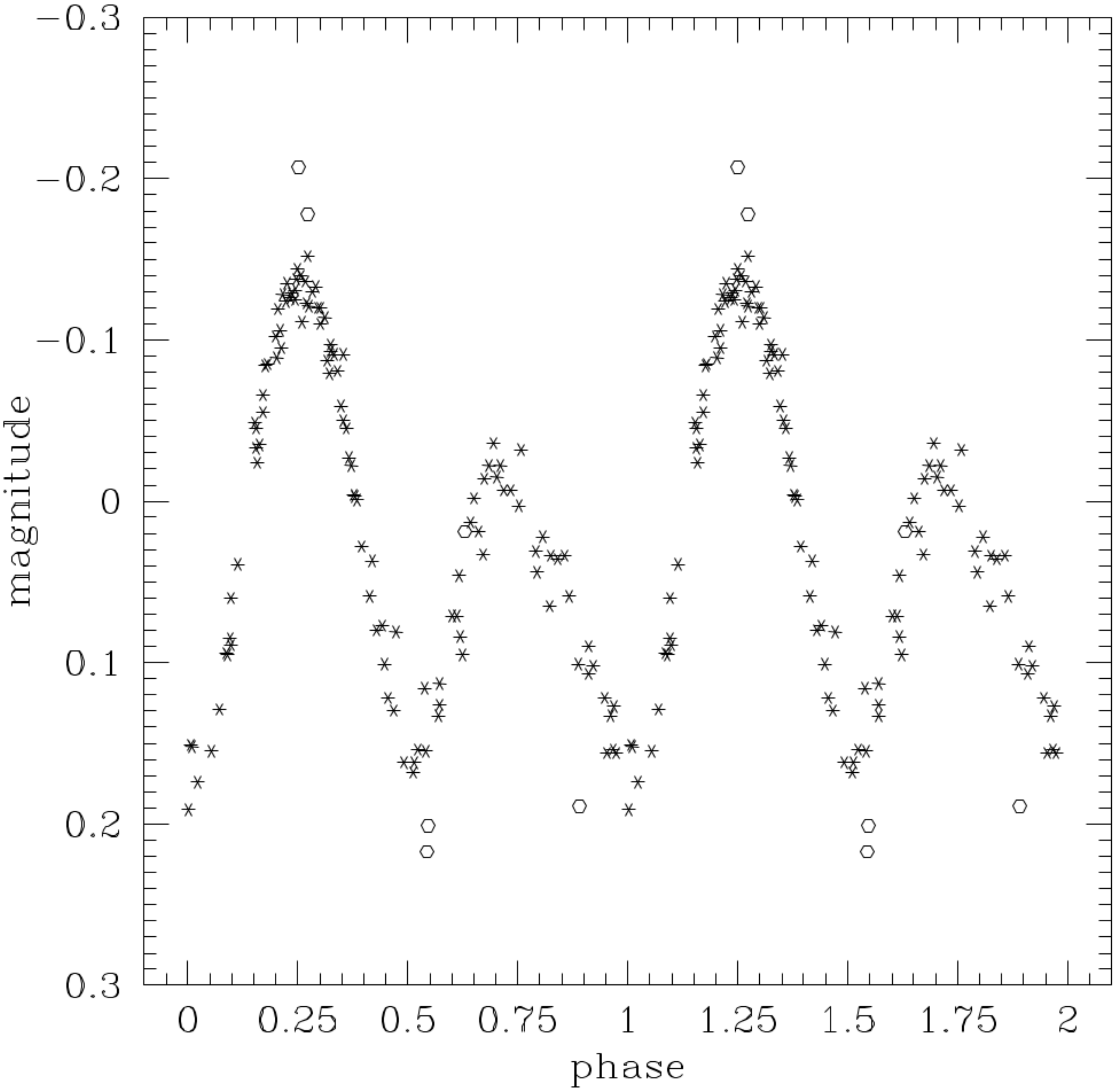


Figure 1.4: V (open points) and I (asterisks) measurements obtained in JD 2 453 8802 453 910 (season 2006; Tylenda et al. 2011).

of the OGLE survey, meaning photometric data of the object had been collected since august 2001. By plotting the light-curve of the object over time (Figure 1.4), it was found that the progenitor was a variable with a period of 1.4 days from the years 2002 to 2007. This period was not constant, but slowly decreasing by 1.2% from the the first observations in 2002 up until the outburst in 2008. The best explanation of the pre-outburst light behaviour is that the object was a contact binary, as the light-curves showed two different luminosities at the peak of its sinusoidal behaviour (Figure 1.4). This is due to the companion interacting with the envelope of the primary, such that gas curves around toward the back of the companion. Here the gas is heated as it collides with other gas, producing a higher luminosity at the leading edge. This translates to two different peak luminosities as the system orbits, a lower peak luminosity when the companion moves towards us and a higher luminosity when it moves away.

Looking at Figure 1.3, we see that the luminosity steadily increases between approximately 2003 and 2007. This is likely due to heating of the entire system as the components interact. As the companion falls into the primary, the system losses angular momentum, heating and lifting the envelope. This is likely to be the cause of the rise in luminosity from the beginning of 2008, to the eruption in August. The final outburst is then the result of the two cores merging into one star, bringing the maximum luminosity up to $\sim 3 \times 10^4$ $L_\odot$. After the event a cool (~3000 K) star is observed. During V1309's eruption, it had released $\sim 10^{46}$ erg over one month (Soker & Kashi, 2012).

V838 Mon also had an outburst of around two months length, cooling to a temperatures of ~3000 K and releasing $\sim 10^{47}$ erg. M85 OT 2006 had an outburst lasting ~70 days, with a final effective temperature of ~4000 K also releasing $\sim 10^{47}$ erg (Kulkarni et al., 2007).

The current consensus is that the V1309 Sco ILOT was the direct result of a common envelope leading to a merger of two stars. Other ILOTs such as V838 Mon are similar to V1309 Sco that it is plausible that it too would have been caused by a binary merger or interaction. Other phenomena also populate the optical transient strip, such as luminous blue variables and events such as brown dwarf and planet mergers (Bear et al., 2011). Only further studies will tell whether some or all of the phenomena populating the diagram in Figure 1.2 are also the result of stellar interactions.

1.1.4 Interactions Between Stars and Planets

For the past two decades exoplanets have been a hot topic for discussion and many surveys have been carried out to search for them. Only recently have we realised that their presence offers the star an opportunity for an interaction.

The first exoplanet was discovered around an evolved star: a pulsar (Wolszczan & Frail, 1992). Though most surveys have found planets around main sequence stars, to date a few exoplanets have been discovered around evolved stars such as giants (Sato et al. 2008; Jones et al. 2014; Mitchell et al. 2013) and recently around white dwarfs (Marsh et al., 2014). The existence of planets in small orbits around evolved stars may suggest that the planets survived the expansion of their parent star. Alternatively these planets are second generation planets formed from debris disks created due to stellar interactions such as a common envelope (Volschow et al., 2014). Both proposed explanations require some interaction to have occurred.

The influence of planets on stellar evolution and their engulfment has been recently considered (Siess & Livio 1999a; Siess & Livio 1999b; Villaver & Livio 2007; Alikutty et al. 2011; Mustill & Villaver 2012). It is likely that these planets would interact with their parent star when it expands and this may lead to a common envelope phase. Planets are probably not capable of unbinding the envelope, however it may fall into a tighter orbit with the core, or likely evaporate. The engulfment of a planet may also cause some pollution in the atmosphere of the parent giant star. Adamow et al. (2012) uses this reasoning to explain a giant star with very high lithium abundance and the presence of a planet in a highly eccentric orbit as indicators for a possible recent engulfment of a second planet. Exoplanets tend to have low eccentricities around giants due to tidal interactions (Villaver & Livio, 2009), hence this planet with high eccentricity must have achieved this via the orbital decay and engulfment of another planet in the system.

Another possible system is described by Charpinet et al. (2011). This system consists of two Earth-sized bodies orbiting a hot B subdwarf star KIC 05807616, at distances of 1.2 and 1.6 $R_\odot$. Subdwarf B (sdB) stars are believed to be the core of RBG stars that suffered a mass loss event that stripped most of their envelope. It is possible that significant mass loss was caused by an interaction with a companion. Approximately two thirds of sdB stars are in binary systems (Saffer et al., 2001). However, for the other third of stars, interaction with planets may provide another avenue to their formation (Soker, 1998). The 2 planets around KIC 05807616 may have survived such processes (Charpinet et al. 2011; Passy et al.

2012).

In recent years characterisation of exoplanets using planetary disks around white dwarfs has also become a very precise science. White dwarfs with atmospheres polluted by relatively large abundances of metals are believed to be caused by the accretion of planetesimals (Zuckerman et al., 2007). By studying the metal abundances in the atmospheres of these polluted white dwarfs, the composition of the engulfed asteroids or planetary bodies can be determined. It is even possible to deduce whether swallowed bodies have been gravitationally differentiated (Melis et al. 2011; Zuckerman et al. 2010).

More complex planetary systems have also also been discovered. For example planets around main sequence binary stars (**?**; **?**). These planets tend to be orbiting binaries which are in tight orbits. All of these circumbinary planets have been found around main sequence binaries until very recently, when a planet has been found around the post common envelope binary NN Serpentis (Marsh et al., 2014). The discovery of planets around post common envelope binaries suggests that the common envelope event may lead to the production of second generation planets. This idea is briefly address in our research.

1.2 Epilogue

In this chapter we have highlighted that we should not consider all stars as solitary objects and their evolution is likely to involve interactions with companions, be they stellar or planetary. In particular we focused on the need for a common envelope interaction to explain many of the mentioned phenomena.

The common envelope phase is a crucial event in the evolution of binaries. However, our understanding of this event is very poor. This lack of understanding translates into difficulty in determining Type Ia SNe rates, the rates of mergers or understanding if and when planets can disturb stellar evolution. Here we aim to investigate this phase. One avenue to research the common envelope phase is via computer simulations and this is the approach we have taken. The history of common envelope simulations will be described further in Chapter 2.

No simulation to date (e.g., Rasio & Livio 1996, Sandquist et al. 1998, Ricker & Taam 2008, Ricker & Taam 2012, Passy et al. 2012) has been able to successfully reproduce the entire event that leads to a close binary, namely an unbound envelope and small binary separation to match those observed (Paczynski, 1976). Common envelope simulations typically end with most of the primary's envelope still bound albeit lifted to large distances from the core. Because of this, we decided to investigate what might happen in the event that the bound gas returns to the binary system forming a disk. After all the bound gas is bound to fall back!

A fall-back interaction phase between the core, the companion and the gas may provide a new opportunity for renewed energy and angular momentum transfer from the stars to the gas (Kashi & Soker, 2011). This would result in more gas gaining energy and potentially becoming unbound, while the core and companion fall into a tighter orbit. This fall-back event may occur multiple times with varying efficiencies, eliminating the entire envelope and converging to separations as found in observations.

2

Theoretical and Numerical Background

The idea of the common envelope (CE) interaction was first proposed by Paczynski (1976) to explain the existence of cataclysmic variables and has since been extensively studied both via the use of simulations and observations of post common envelope binaries, amongst which some are the central stars of planetary nebulae (PNe). In the past two decades computers have become powerful enough to run complex, 3D hydrodynamic simulations of astronomical events and this has contributed to our knowledge and our understanding of binary interactions and the common envelope phase. In this chapter we describe the theoretical underpinning of the common envelope interaction, the simulation techniques that have been used to model this interaction and, finally, the past attempts at simulating the interaction.

2.1 The Theoretical Underpinning of the Common Envelope Interaction

It was briefly described in Chapter 1 how a common envelope interaction is thought to occur. Here we will discuss it in more detail. It is understood that Roche lobe overflow can lead to a common envelope. We also think that via some mechanism energy and angular momentum are transferred from the core and companion to the giant envelope and that the result is an unbound envelope. van den Heuvel (1976) originally formulated the assumption that the post common envelope orbital separation of the giant's core and companion could be determined by equating the change in orbital energy and the binding energy of the envelope as shown in Equation (2.1) (Webbink 1984, de Kool 1992):

$$-G\frac{M_1 M_{1,e}}{\lambda R_{1,L}} = \alpha_{CE} G \left[\frac{M_{1,c} M_2}{2A_f} - \frac{M_1 M_2}{2A_i} \right], \tag{2.1}$$

where $M_1, M_{1,c}, M_{1,e}$ and M_2 are the primary, the primary's core, the primary's envelope and the companion masses, respectively. $R_{1,L}$ is the radius of the primary's Roche lobe at the onset of the mass transfer, where λ depends on the mass distribution of the primary's envelope. A_i and A_f are the initial and final separation of the system, respectively. Finally, α_{CE} is the unbinding efficiency of the common envelope. This equation is expected to hold irrespective of the energy transfer mechanism. This mechanism assumed the conversion of gravitational potential energy to the kinetic energy of the envelope, however this ignores other reservoirs of energy such as the recombination energy of the gas.

Gravitational drag is currently thought to be the major contributor to the decay in orbital separation during a common envelope event (Ricker & Taam, 2008). This "drag" force has nothing to do with the hydrodynamic drag experienced by a body moving through a medium. Rather, it works on the basis that as a body travels through a medium, the gravitational force from the wake produced from the interaction slows down the mass. The strength of this drag depends on whether the mass is subsonic or supersonic. Analytically, Ostriker (1999) found the gravitational drag force is strongest at speeds close to the speed of sound.

Many simulations of the common envelope phase have been carried out and will be discussed in Section 2.2. A simulation of a common envelope event is best conducted using 3-dimensional hydrodynamics codes. This is because the interaction is inherently asymmetric. There are two main methods of simulating fluid dynamics in astrophysics: Eulerian grid codes and Lagrangian Smoothed Particles Hydrodynamics (SPH). Both these methods solve the fluid dynamic equations. These equations describe the changes in density, momentum and energy for a particular volume of fluid using the appropriate conservation laws and fluxes of quantities through the surface of the volume. The fluid equations are the continuity equation (Equation 2.2), the momentum equation (Equation 2.3) and the energy equation (Equation 2.4):

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \tag{2.2}$$

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v} = -\frac{1}{\rho}\nabla P - \nabla \phi, \tag{2.3}$$

$$\frac{\partial e}{\partial t} + \mathbf{v} \cdot \nabla e = -\frac{1}{\rho}\nabla \cdot (P\mathbf{v}) - \mathbf{v} \cdot \nabla \phi, \tag{2.4}$$

where, ρ is the density, $\mathbf{v}$ is the velocity of the fluid, P is the pressure, ϕ is the gravitational potential, e is the specific total energy of the fluid and t is time. We have included the gravitational potential in our equations because in common envelope simulations self-gravity needs to be accounted for. One could optionally include magnetic fields or other physical phenomena such as radiative transfer of energy, heating and cooling. However this comes at the expense of computation time. Any suitable equation of state can be used. In our simulations we have assumed the gas is ideal, hence the equation of state is:

$$P = nk_bT, \tag{2.5}$$

where n is the gas number density, T is the temperature of the fluid and k_b is the Boltzmann constant,($k_b = 1.38 \times 10^{-16}$ erg K^{-1}). The specific total energy of the gas is the sum of the specific internal and kinetic energies, i.e.,:

$$e = u + \frac{\mathbf{v}^2}{2} \tag{2.6}$$

where u is the specific internal energy u is given by:

$$u = \frac{1}{\gamma - 1}\frac{P}{\rho}, \tag{2.7}$$

where, in our work, we have adopted an adiabatic index $\gamma = 5/3$. In adiabatic hydrodynamic simulations where cooling is ignored, the composition of the gas is irrelevant. Many simulations do not even specify temperature, preferring to work with specific total energy. Only when temperature is plotted, does the composition need to be specified.

2.2 Simulation Techniques and Codes

SPH, invented by Lucy (1977) and Gingold & Monaghan (1977), uses a Lagrangian approach to fluid dynamics, which means the code follows a parcel of fluid as it moves. SPH uses *mass coordinates*, where each parcel of fluid corresponds to a "particle" or, rather, an interpolation point. Each particle is defined by a particular mass so these particles change separation, or the interpolation points can become more or less rarefied. This means the resolution of the simulation is sensitive to the density of the fluid, having higher resolution in higher density regions. In low density regions the number of particles in a volume may be very low, making it difficult to gather detailed information on the evolution of gas in these regions. SPH also requires two to three times as more particles than the number of cells in a grid simulation to produce to same resolution of physics such as shocks (Hubber et al., 2013). These are some downsides of using SPH, however there are some advantages in using this technique. The particles in SPH are defined by mass, hence this method intrinsically conserves mass, and angular momentum. As SPH simulations do not constrain the size of the computational domain, mass can be tracked up to large distances. This is a useful characteristic in simulating the common envelope phase where the gas quickly gets pushed outwards. This would be a phenomenally useful characteristic for our project and could allow us to simulate a full fall-back event, something that our grid code did not allow (see Chapter 3).

The other hydrodynamic method which is also the technique we have used, is the grid method. This method uses a Eulerian approach to fluid dynamics. We have used the grid code *Enzo* (**?**), using the ZEUS hydrodynamic method (Stone & Norman, 1992). Other hydrodynamic methods are available in *Enzo*, such as the piecewise parabolic method (Colella & Woodward, 1984). Passy et al. (2012) (in private communication) had tested ZEUS against PPM and found they return similar results however the ZEUS method was faster, hence why we have selected to use the same method for our simulations. The Eulerian approach to fluid dynamics uses *volume coordinates* instead of mass coordinates, which is basically a grid. This method advects mass and other quantities through grid cells, and the resolution is not affected by the density of the gas, but it is affected by the size of the cells. Techniques such as nested grids (e.g. Sandquist et al. (1998)) or adaptive mesh refinement (AMR, e.g. Ricker & Taam (2008)) have been developed to help improve resolution where needed.

The grid method is limited by the size of the grid, which means if mass leaves the grid, it cannot return. Also, with this method, there are losses due to cell boundary approximations. This leads to local non-conservation of energy and momentum. Re-zoning algorithms with AMR can also contribute to this non-conservation, if they are not implemented carefully (**?**). The nested grid technique has some disadvantages in that the regions of higher resolution must be pre-defined, for example, the central region near the core and companion in a common envelope simulation (Sandquist et al., 1998). As long as regions that need higher resolution remain in the nested regions, the simulation can produce accurate data. However, the center of mass may wander in and out of the highest resolution regions resulting in loss of information. In our simulations using *Enzo*, we do not implement any nested grids, or AMR techniques, but use a simple uniform grid.

Hydrodynamic codes use the fluid dynamics equations by discretising them. In each fluid dynamic equation, there is only one quantity time derivative. For the continuity equation, this is the density, for the momentum equation, this is the velocity, and for the energy equation this is the internal energy. As velocity is a vector in three dimensions, the full set includes 5 equations. When discretising equations, differential equations become difference equations.

Using the discretised fluid equations, simulations of fluids can be carried out, where the resolution is dependent on Δx, i.e., on the size of each cell in the grid, and Δt, i.e., the time steps taken. The time step is itself dependent on the spatial resolution via the Courant condition given in Equation (2.8):

$$\Delta t < f_{Cour} \frac{\Delta x}{|v + c|}, \tag{2.8}$$

where v is the velocity of the parcel of fluid, c is the sound speed of the fluid, and f_{Cour} is

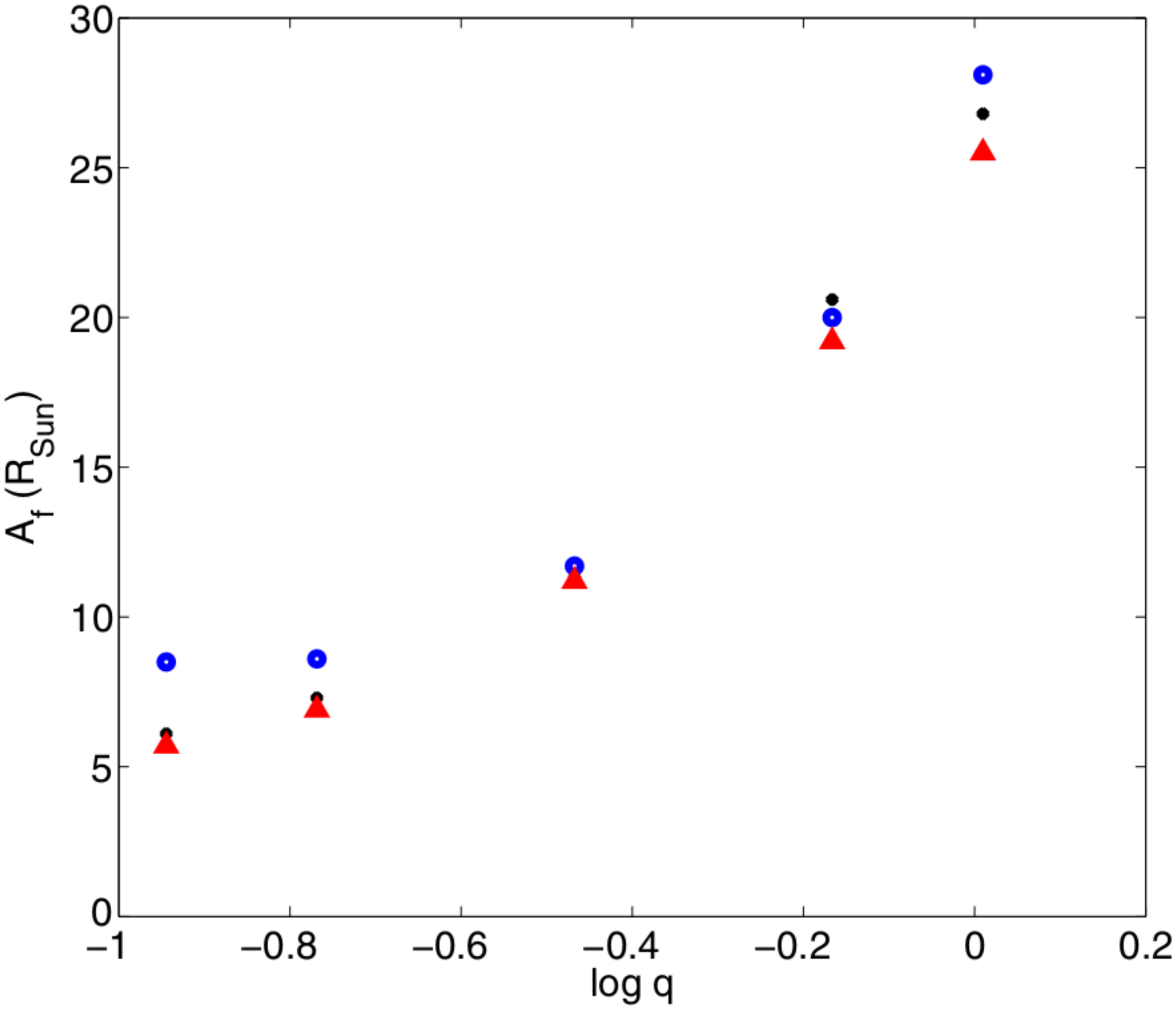


Figure 2.1: The final separation of the core and companion from a number of common envelope simulations of systems with varying mass fraction. The small black dots indicate separations found with SPH simulations, while the blue dots and red triangles are from fluid simulations of resolution 128^3 and 256^3 respectively. Figure 15 from Passy et al. (2012).

the Courant factor. The Courant condition determines the upper limit for a time step in numerical calculations such that sufficient information can be gathered for a wave travelling through some spatial step (in our case Δx). We see from Equation (2.8) that the time step is dependant on the size of cells, therefore a improvement in spatial resolution of two implies a factor of two smaller time step, so will take two times longer to calculate the flow of quantities. In 3D this increase in spatial resolution by two will take sixteen times longer to run. Generally, the smaller Δx is, the smaller the necessary Δt is.

Figure 2.1, shows figure 15 of Passy et al. (2012) where they demonstrate the dependence of final orbital separation on the resolution, found from SPH and a grid code. To determine an appropriate resolution for simulating a common envelope event, the orbital separation should converge to a certain value for increasing values of the resolution. Passy et al. (2012) (Figure 2.1), using a combination of 500 000 particle SPH simulations and 128^3 and 256^3 cell grid simulations deemed 256^3 cells to be sufficient for their simulations.

2.2.1 Mapping Stellar Evolution Models to Hydrodynamic Simulation

The common envelope simulations in Passy et al. (2012) are between a giant and a more compact star. The structure of the giant must be calculated separately by a computer code able to include much more physics than would be possible in our hydrodynamic simulations. The primary star must be pre-calculated by a stellar evolution code such as STARS (Eggleton, 1972), as used by Sandquist et al. (1998) and Ricker & Taam (2012), or EVOL (Herwig, 2000) as used by Passy et al. (2012). Another publicly available code is the versatile and accurate 1D stellar evolution code MESA (Paxton et al. 2011; Paxton et al. 2013).

EVOL is a one dimensional stellar evolution code and in the work of Passy et al. (2012) it was used to evolve a 1 $M_\odot$ main sequence star to a 0.88 $M_\odot$ red giant star with a radius of 83 $R_\odot$ and 0.392 $M_\odot$ core. As this stellar model is in one dimension and the *Enzo* fluid simulations are in three dimensions, the model must be mapped under the assumption that the star is initially spherically symmetric. As giant stars typically have a very large radius ($\sim$ 100 $R_\odot$) and large diffuse envelope, and the core is very small ($\sim$0.01 $R_\odot$) and dense in comparison, the core is modelled as a point mass particle. The same is done for the companion, which also has a relatively small radius (∼0.5 $R_\odot$) in comparison to the radius of the giant star. Modelling the core and companion as point mass particles is rather non-physical. This can affect the gravitational drag, but one can insure that the effect is minimal by carrying out convergence tests. Treating the core and companion as particles is the only way to carry out these computations, which are already very demanding due to the large range in size and time scales.

Due to a much lower resolution of the 3D hydrodynamic code compared to the 1D stellar evolution code and to the different equations of state used, the mapped star is slightly out of hydrostatic equilibrium. In order to make the star hydrostatically stable once the star is mapped into the 3D code it is allowed to evolve for a few dynamical times (∼0.1 year for the RGB star of Passy et al. (2012)), while initially damping the velocities to reach an equilibrium state. This process usually results in an RGB star coming into an equilibrium state with a slightly larger radius of 91 $R_\odot$. The extent of this expansion is dependent on the resolution of the fluid simulation.

This is the mapping process used by Passy et al. (2012) and is adopted commonly by 3D common envelope simulations. Once the giant star has been mapped and stabilised, the companion is introduced either on the surface of the giant (Passy et al., 2012), or at some distance (Sandquist et al. 1998; Ricker & Taam 2012). Once the companion is introduced, the system is allowed to evolve. These simulations are reviewed in the next section.

2.2.2 Past Simulations of the Common Envelope Interaction

Simulations of the common envelope phase have been carried out as computers became powerful enough to run them with sufficient resolution. All simulations (e.g., Meyer & Meyer-Hofmeister 1979, Rasio & Livio 1996, Sandquist et al. 1998, Ricker & Taam 2008, Ricker & Taam 2012, Passy et al. 2012) replicate only what is known as the fast in-spiral phase, when orbital energy and angular momentum are transferred to the envelope gas, aiding in its removal. It is impractical to run simulations for much longer than these initial stages partly because the gas leaves the computational domain (for grid simulations) and partly because simulating longer time scales means having to add physics such as cooling. All simulations of the dynamical in-fall phase have found that most of the envelope is still bound to the system by the time the orbital separation stabilises. Rasio & Livio (1996) using SPH with 5×10^4 particles found 90% of the gas was still bound. Sandquist et al. (1998), using a stationary nested grid method found approximately 60% of the envelope was still bound to the system. Ricker & Taam (2012) improved on this simulation by using an AMR grid code and with this method they found that 75% of the envelope remained bound to the system. Passy et al. (2012), comparing single grid and SPH simulations concluded that approximately 85% of the envelope remains bound to the system. In all simulations most of the envelope is lifted to large distances away from the orbiting cores, but remains mostly bound to them.

The final outcome of the common envelope interaction in nature must be the total removal of the envelope or else we would not observe post common envelope binaries. Whether this discrepancy is physical and the first fast in-spiral phase truly does not unbind the envelope, or whether the simulations ignore some physics, is to be determined. Missing physics is likely as these simulations include only self gravity. Han et al. (1995) and more recently Ivanova et al. (2014) suggested that the giant star uses some of the energy in its chemical and atomic bonds to help unbind its envelope. When the gas recombines upon expanding it recombines and this energy is released as radiation. If this energy is released within optically thick layers it can transfer momentum to the gas and help unbind it. In addition, molecules and grains may form increasing the optical thickness. The combination of grains and ionisation energy being released is calculated to be of the order necessary to unbind the remaining bound gas found in simulations (Webbink 2007; Harpaz 1998; Passy et al. 2012).

Though recombination may contribute to a more efficient removal of the envelope, it would not help to reduce the final separations obtained in simulations. To date the separations measured in binary central stars of planetary nebulae are significantly smaller ($\leq 4R_\odot$;

Zorotovic et al. 2010; De Marco et al. 2011; Schreiber & Gnsicke 2003) than what the simulations give at the end of the fast in-spiral phase (between 27 $R_\odot$ to 6 $R_\odot$; Passy et al. 2012). As the simulations find, the rapid in-fall phase varies between systems, with systems with larger $q = M_2/M_1$ resulting in larger separations (see Figure 2.1), but this trend is not reflected in the observations. Looking at figure 17 of Passy et al. (2012), which compares the simulation separations of Figure 2.1 with observed post common envelope binaries, we see that few systems have mass ratios $\frac{M_2}{M_1} \geq 0.5$. Simulations showed that the lower mass ratio systems resulted in smaller separations. However, these separations remain generally larger than observed separations. In the simulations, the system stops in-spiralling when the density around the core and companion decreases below a certain threshold and energy and angular momentum transfer stops. This seems to happen at larger orbital separations for larger mass ratios, q. The time taken to reach orbital stability also varies, with systems with a lower mass ratio taking more time to reach the plateau in the orbital decay.

With these discrepancies between observations and simulations in mind we wonder why the simulations are failing to remove the envelope completely. We have already discussed within this section that alternative energy sources, such as recombination energy, not accounted for in simulations could lead to envelope ejection. However in this thesis we argue that there could be another way to unbind the remaining envelope *and* bring the core and companion closer together. What happens when the bound gas falls back onto the binary system? Logically, a new interaction should take place and the binary separation should decrease further and more of the envelope should become unbound. These are the questions that we try to answer in this thesis. Our hypothesis is that the fall-back of the gas will allow the core and companion to lose more angular momentum via the interaction with the gas. This transfer of angular momentum will shrink the orbit further and provide the gas with more orbital energy to allow it to escape the system.

2.3 Analysis of the Simulation Outputs

As briefly mentioned previously, the code we use for our simulations is *Enzo*, which is a Eulerian grid code. We have not applied any grid refinement but used a simple uniform grid with 256^3 cells.

The outputs from our simulations are analysed using `yt`, version 2.6 (Turk et al., 2011), which is written in python. The software `yt` reads the *Enzo* data cubes and produces useful visualisations. *Enzo* fields are values of quantities at each location (grid cell center) in the computational domain. *Enzo* uses a number of basic fields (e.g., density, pressure, velocity, gravitational potential, total energy) from which other fields can be derived. There are many pre-constructed, derived fields (e.g., number density, pressure, sound speed, etc.)[1]. Custom derived fields can also be constructed within `yt`. An example is the temperature field (see code in Appendix A.1.1.).

We derive the temperature of the gas using its internal energy by substituting Equation (2.5) into Equation (2.7), which gives:

$$T = \frac{(\gamma - 1)\,\rho U}{n k_B}. \tag{2.9}$$

When creating this field, some values can be input manually such as γ, which is 5/3 as we are in an adiabatic regime. These values can also likely be read directly from the *Enzo* setup file. This setup file contains information on the simulation options (such as time step, data dump times, grid dimensions, etc.), as well as input data about the fluid (e.g., adiabatic index, mean molecular mass) and options for including certain physics such as radiative transfer and cooling or dust formation.

We can also read and manipulate data from subregions of the entire domain. These regions have to be predefined within the box. An example is the codes used to determine the amount of gas falling onto our binary system (see code in Appendix A.2.1) in Chapter 3. In this case a region with x, y and z dimension of $1.0, 1.0$ and 0.3 of the full length of the computational domain is defined. This region is centred on the center of the box, and is used to monitor the total mass falling into the orbital plain.

---

[1] A full list of all fields can be found at http://yt-project.org/docs/2.6/reference/field_list.html

We can also read in individual cell data. This is used numerous times in the code to determine the center of mass, the angular momentum of gas, the bound and unbound mass within the grid, fall-back time of gas and Mach number in selected cells (see Appendix A.2.2 to A.2.6). The *Enzo* gravitational potential field only returns the potential of the gas on itself. *Enzo* obviously does recognise the presence of the particles and accounts for their gravity when running the simulation. However, the particle's potential is just not considered when returning the gravitational potential of the gas. The potential of the particles on the gas had to be calculated manually. This is done by simply reading through all cells, iterating through all three dimensions, using nested for-loops. The range of the x, y, and z dimensions is the grid dimension, read in from the setup file.

2.4 Epilogue

In this chapter, we have reviewed the theoretical underpinning of the common envelope phase in binary interactions and the tools that have been used to study this phase. We have highlighted some of the issues with our current understanding of this phase and some of the downfalls of simulations. In particular we have described the problem all simulations have in unbinding the envelope bringing the core and companion to within observed separations. We have proposed that lifted, bound material will fall back onto the core and companion providing a new opportunity to reduce the orbital separation and unbind further the envelope gas. Next we will look at the setup for the simulations we have run to test our hypothesis of the fall-back disk.

3

Simulation Setup

In the previous chapters we have reviewed the significance of the common envelope phase, the studies conducted to better understand this phase and the tools used. We have then explained how a fall-back of bound envelope gas may help resolve some of the outstanding issues found in simulations. Here, we lay out the basis for our simulations and the iterations conducted to produce a successful fall-back disk.

3.1 Initial Conditions

The set up for our simulations is based on the finishing conditions of the simulations of Passy et al. (2012). Passy et al. (2012) conducted a series of common envelope simulations between a red giant and a main sequence star for varying mass fractions $q = \frac{M_2}{M_1}$ with different computational methods and different resolutions. The simulations resulted in the envelope being lifted, but only a small amount of the envelope mass being unbound. The separation between the core of the RGB star and the main sequence star stabilised, so no more orbital energy would be injected into the system from a decaying orbit, and therefore no more mass can be unbound. The envelope is therefore destined to fall back. We decided to produce a fall-back disk that would be the logical continuation of Passy et al. (2012)'s simulation of a system with a mass fraction of $q = 0.6$. In Section 2.2.1 we described the mapping of the evolved giant star from the 1D stellar evolution code to the 3D fluid dynamic simulation. In that section we justified modelling the core of the giant and the companion as point mass particles. In the system that we simulate the companion is a point mass with 0.6 $M_\odot$, and the core is a point mass with 0.392 $M_\odot$. At the end of their simulation, Passy et al. (2012) found that the system stabilised with a separation of ~20 $R_\odot$ and that 0.44 $M_\odot$ of the envelope, which has mass of 0.48 $M_\odot$, remained bound. Our simulation begins with the core and companion point mass particles with a separation of 20 $R_\odot$ and aims to have approximately 0.44 $M_\odot$ fall back onto the system.

Looking at the end of the simulation of Passy et al. (2012), the final density of the gas within an AU of the core and companion was approximately 5×10^{-7} g cm^{-3}. The final temperature was found to be 50 000 K. The size of the domain we simulated is 10^{13} cm on a side, which corresponds to approximately 144 $R_\odot$. This length is a third of that used by Passy et al. (2012), which allows a better resolution of the core and companions' immediate surroundings. For most of our simulations we have selected a resolution of 256^3, which results in cells having an approximate dimension of 0.5 $R_\odot$ on a side.

Now that we have described the initial conditions needed for our simulation, we will explain our approach to simulating the fall-back of gas.

3.2 Our Fall-back Setup

The initial simulation setup was designed such that the gas would fall onto the core and companion as a disk coplanar with the orbital plain. This decision was made as previous studies of common envelope interactions show that most of the mass is expelled preferentially along the orbital plain (e.g., Sandquist et al. 1998, Passy et al. 2012, Ricker & Taam 2012, who found that gas left within 30° of orbital plain). It is therefore reasonable to assume that the gas would fall back along this plain.

We attempted to set up a disk with the required parameters, but found it impossible to generate a stable configuration. To overcome this, the computational domain was filled with a diffuse constant density gas and the initial velocities were imposed such that the gas above and below the orbital plain would be evacuated quickly, while the gas near the plain would fall onto a disk. In the 3-dimensional grid, a column of grid points going through the orbital plain (i.e., with the same x and y value, but variable z value), had constant initial velocity, which was initialised to be a given fraction of the keplerian velocity on the orbital plain. This would create concentric, vertical cylinders of gas centred in the middle of the grid, moving with the same rotational velocity. The velocity was selected such that the gas in these cylinders nearer the orbital plain would be sub-keplerian and hence fall onto the core and companion, while farther above and below the plain it would be super-keplerian and escape. This allowed to, essentially, create a fall-back disk that has the correct mass, angular momentum and energy (checks comparing angular momentum and energy to Passy et al. (2012) are discussed in the next chapter). The fall back disk acts as if the bound gas lifted to large distances in the simulations of Passy et al. (2012) were to return towards the binary.

This set up is explained in Figures 3.1 and 3.2. In Figure 3.1 we see, in the left panel, the keplerian velocities above and below the orbital plain for x and y values, such that $\sqrt{x^2 + y^2} = 70\ \mathrm{R}_\odot$. As expected, the keplerian velocity decreases above and below the orbital plain ($z = 0$). Along the same column, the velocity is initialised to be constant as shown in the middle panel of Figure 3.1. As a consequence the gas within approximately $\pm 15\ \mathrm{R}_\odot$ of the orbital plain is initially sub-keplerian as is demonstrated on the right panel of Figure 3.1, where the red dashed line marks where the velocity is keplerian. The sub-keplerian gas will fall onto the core and companion. At larger $|z|$ positions, the gas is initially super-keplerian. However, this does not mean that this gas will leave the box immediately. Some of this gas may fall onto the core and companion in elliptical orbits.

The keplerian velocity throughout the grid was calculated by using the gravitational potential of the core and companion:

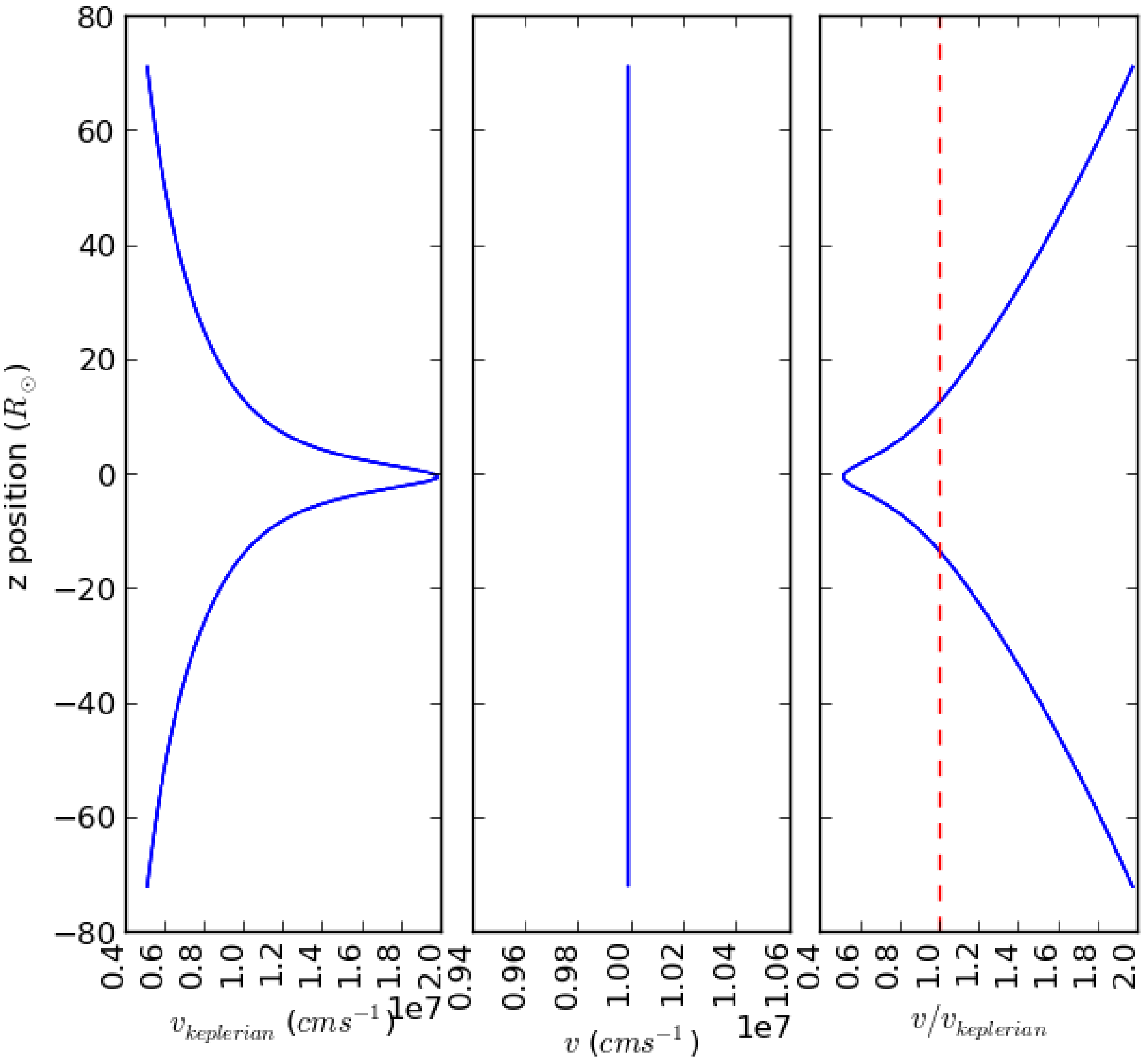


Figure 3.1: Initial velocity values for gas at positions of varying z coordinate for a column that passes perpendicularly through the orbital plain at a distance of ∼70 $R_\odot$ from the center of the grid. The left panel shows the keplerian velocity at that distance. The middle panel shows the initial velocity given to the gas. The right panel shows velocity relative to keplerian velocity. The red dashed line marks boundary between super-keplerian and sub-keplerian gas.

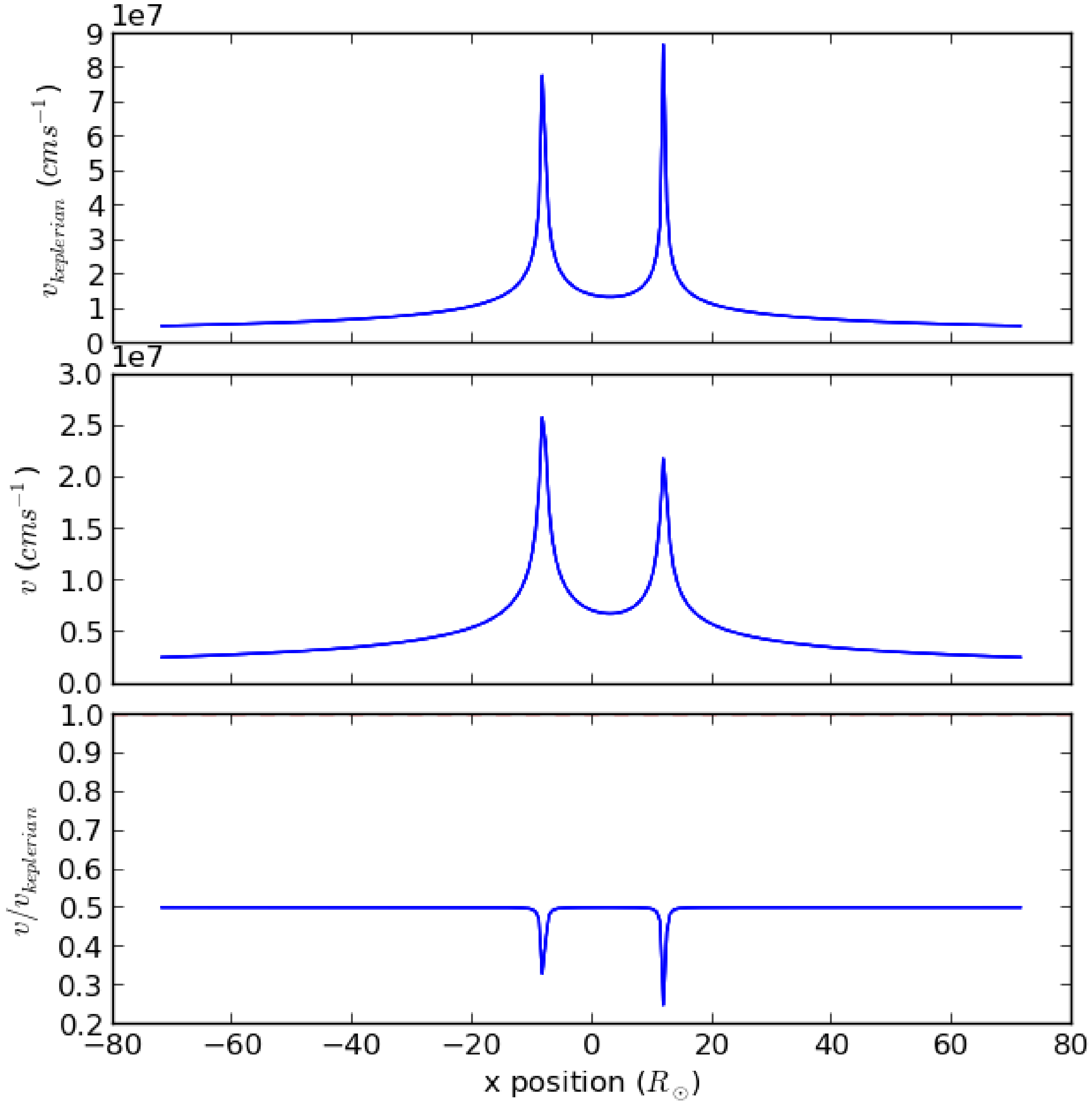


Figure 3.2: Initial velocity of the gas for a cross section along the orbital plane passing through the positions of the core and companion. The top panel shows the keplerian velocity. The middle panel shows the velocity given to the gas at $t = 0$. The bottom panel shows velocity relative to the keplerian value.

$$v_{kep} = \sqrt{-\left(\phi_{core} + \phi_{comp}\right)}, \tag{3.1}$$

where ϕ_{core} and ϕ_{comp} are the potentials of the core and companion, respectively. The potential due to the gas itself was not considered. If it had the resulting velocities would have been larger and we would have needed a smaller fraction of the keplerian velocity to obtain the desired fall back mass. What mattered in our setup was the outcome, not necessarily how to was achieved.

Due to the large value of the potential near the core and companions, the keplerian velocity is expected to be very high. This is shown in Figure 3.2, where a column of gas lying on the orbital plain connecting the core and companion has been displayed. In the top panel of this figure the keplerian velocity at the varying positions away from the core and companion along the orbital plain is shown. Here we see two large spikes near the positions of the core and companion. The middle panel shows the velocity values at these distances. The bottom panel shows the velocities relative to the keplerian values. In this example we have selected the velocities on the plain to be $0.5v_{kep}$. As demonstrated by Figure 3.1, sub-keplerian velocities along the plain translate to super-keplerian velocities at a certain distance above and below the plain.

3.3 Preliminary Simulations

Now that we have established our initial conditions and fall-back set up we will review the simulations that were run to achieve the best fall-back disk. That simulation will be discussed in Chapter 4. The initial parameters of all simulations that were attempted have been summarised in Table 3.1.

All simulations were run by Jan Staff. The set up configuration was determined in collaboration with Orsola De Marco and Jan Staff. Changes in setup parameters were decided by myself following the analysis of the simulations' outputs.

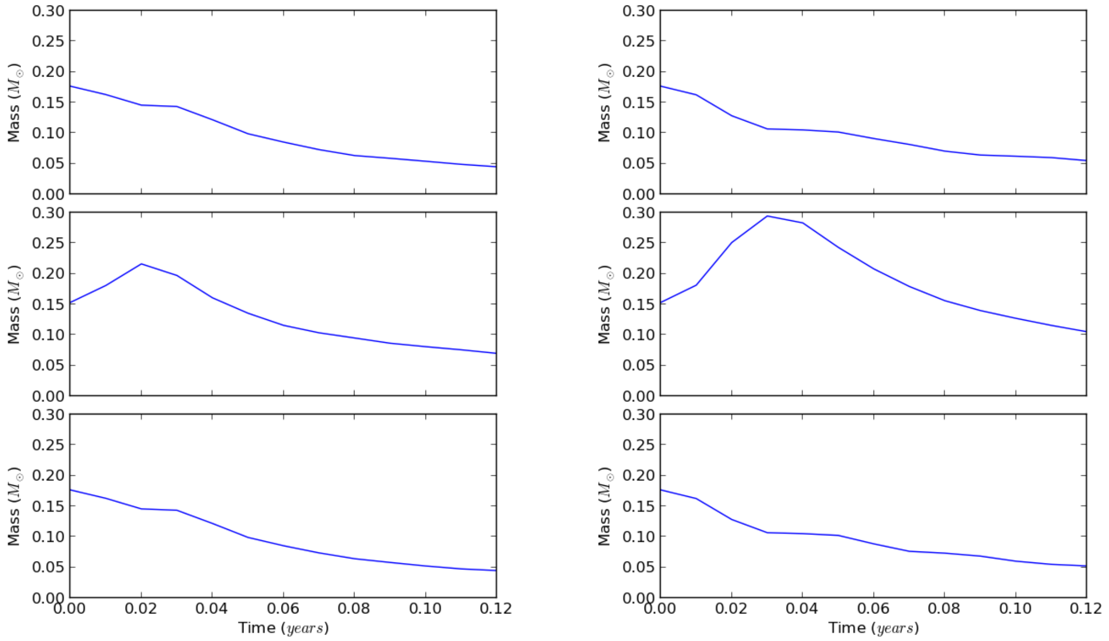
3.3.1 Non-rotating Simulations

Before establishing that a fall-back disk was best represented by rotating gas, we ran simulations that were not initially rotating. For the first of these simulations (*nonrot_hot* in Table 3.1) the grid was filled with diffuse gas with a constant density of 10^{-6} g cm^{-3} and zero initial velocity. This density was selected because the density near the core and companion at the end of the simulation in Passy et al. (2012) was of the same order of magnitude (5×10^{-7} g cm^{-3}). In addition, this density resulted in a total mass of ∼0.29 M ⊙ falling back onto the core and companion. Another simulation with higher density was also run (*nonrot_highden* in Table 3.1). These simulations allow us to adjust the amount of mass falling onto the core/companion system and allowed us, in theory, to test the dependences of the results on density and total mass.

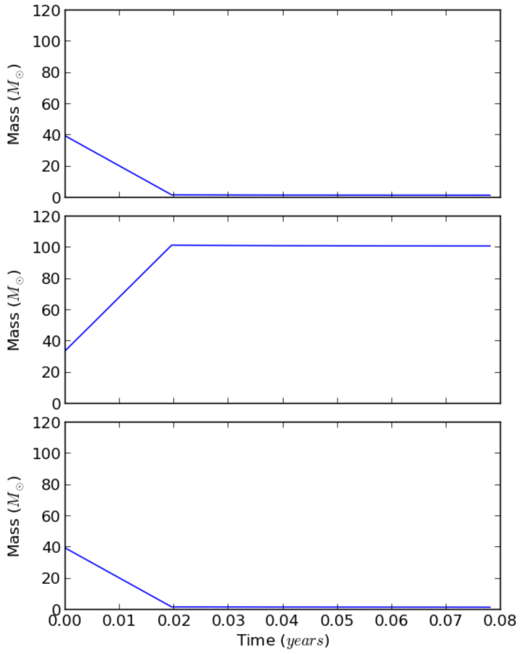
We also run another simulation with an initial density of 10^{-6} g cm^{-3}, but with a lower temperature (*nonrot_cool*). In hindsight there was no clear reason for this extra simulation, although, pre-empting the work presented in Chapter 5, *nonrot_hot* and *nonrot_cool* can provide some insight into the effect of temperature on the fall-back mass.

To measure the mass of the fall-back gas we measured the mass of gas in each of 3 sections of the computational domain (See Appendix A.2.1). The middle section is defined by a box that spans 30% of the z-axis (~ 43 R$_\odot$) and is centred on the orbital plain. The regions above and below are the remaining two sections. By creating a mass time series for each of these three regions we can estimate the fall-back disk mass as the largest value in the central region. We apply this to the *nonrot_hot* and *nonrot_cool* simulations shown in Figures 3.3a and b, respectively. Both of these systems have the same initial density, but have different temperature. We see that *nonrot_cool* has a greater mass of fall back into the central region (∼0.29 M$_\odot$), compared to *nonrot_hot* (∼0.22 M$_\odot$). This is expected as more gas in the hotter simulation expanded away leaving the computational domain before managing to fall into the system.

Comparing *nonrot_hot* with *nonrot_highden* we again see that the higher density simulation has a larger fall-back mass (Figure 3.3c). The high density simulation has a fall-back mass of 100 M$_\odot$, which is unrealistically high. Nonetheless, comparing all three simulations



(a) For *nonrot_hot* simulation. (b) For *nonrot_cool* simulation.



(c) For *nonrot_highden* simulation.

Figure 3.3: The middle panel of (a), (b) and (c) shows the total mass within 30% of the z-axis centred on the origin (i.e., $0.35 \leq z \leq 0.65$, where 1.0 is the full length of grid). Top and bottom panels show the mass above and below the central region.

we see in Figure 3.4, that the higher the fall-back mass, the greater the in-spiral. This is expected, because if there is more mass around the core and companion to interact with, then there will be greater drag.

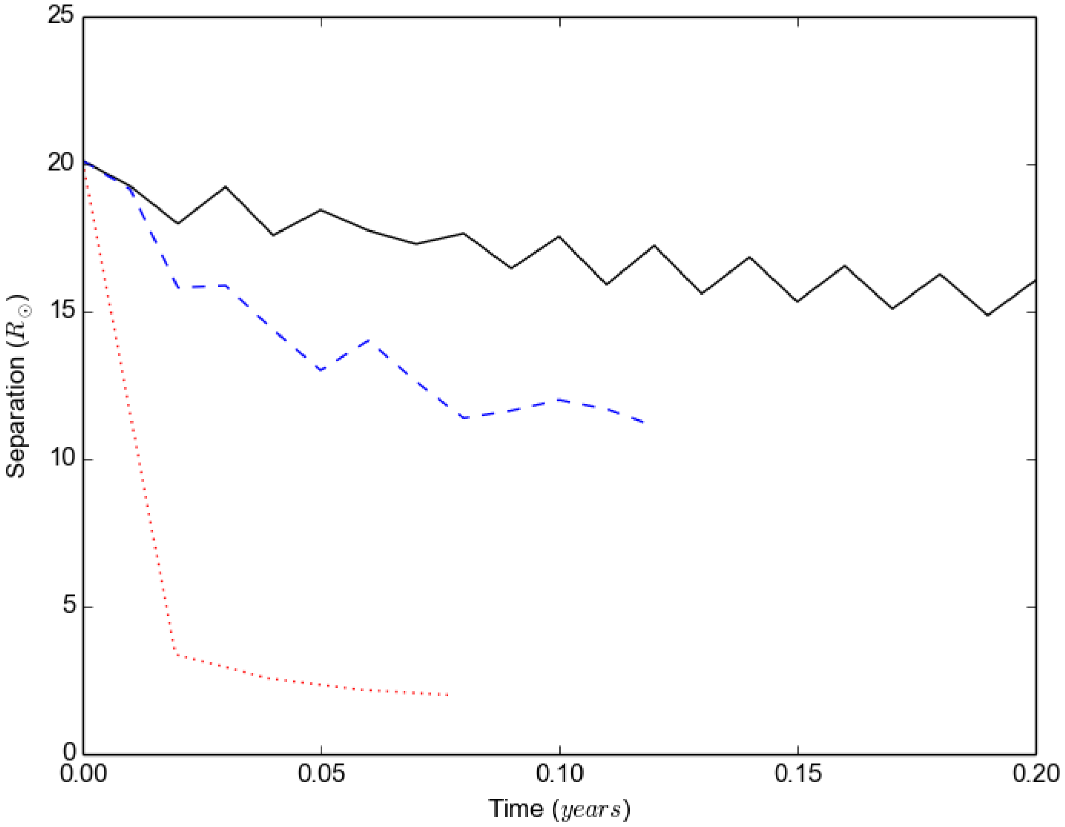


Figure 3.4: Separation evolution for the *nonrot_hot* (solid black line), *nonrot_cool* (dashed blue line) and *nonrot_highden* (dotted red line) simulations. The oscillation in the separation of the core and companion is sure to eccentricity in their orbit.

3.3.2 Rotating Simulations

Next, we implemented the velocity structure discussed in Section 3.2 to produce a disk. These simulations were all calculated with an initial density of 10^{-6} g cm^{-3}. At this time we had not given much thought to the initial temperature, such that *rot_hot* was given the same internal energy, and hence the same temperature, as *nonrot_hot*. The *nonrot_hot* and *rot_hot* simulations provide a simple comparison between mass falling in from all directions and mass falling into a disk and its effect on the in-spiral of the system.

A comparison of the separation over the course of the rotating and non-rotating simulations is shown in Figure 3.5. We see that in both simulations there is a decrease in separation, but the decrease is significantly smaller in the rotating simulation. This is likely due to the fact that the gas can fall further into the center of the system for non-rotating setups, creating a higher density environment around the core and companion, leading to higher drag. With the rotating simulation the gas has angular momentum making it difficult for the gas to fall into the centre to create a higher density environment. Also with this initial velocity

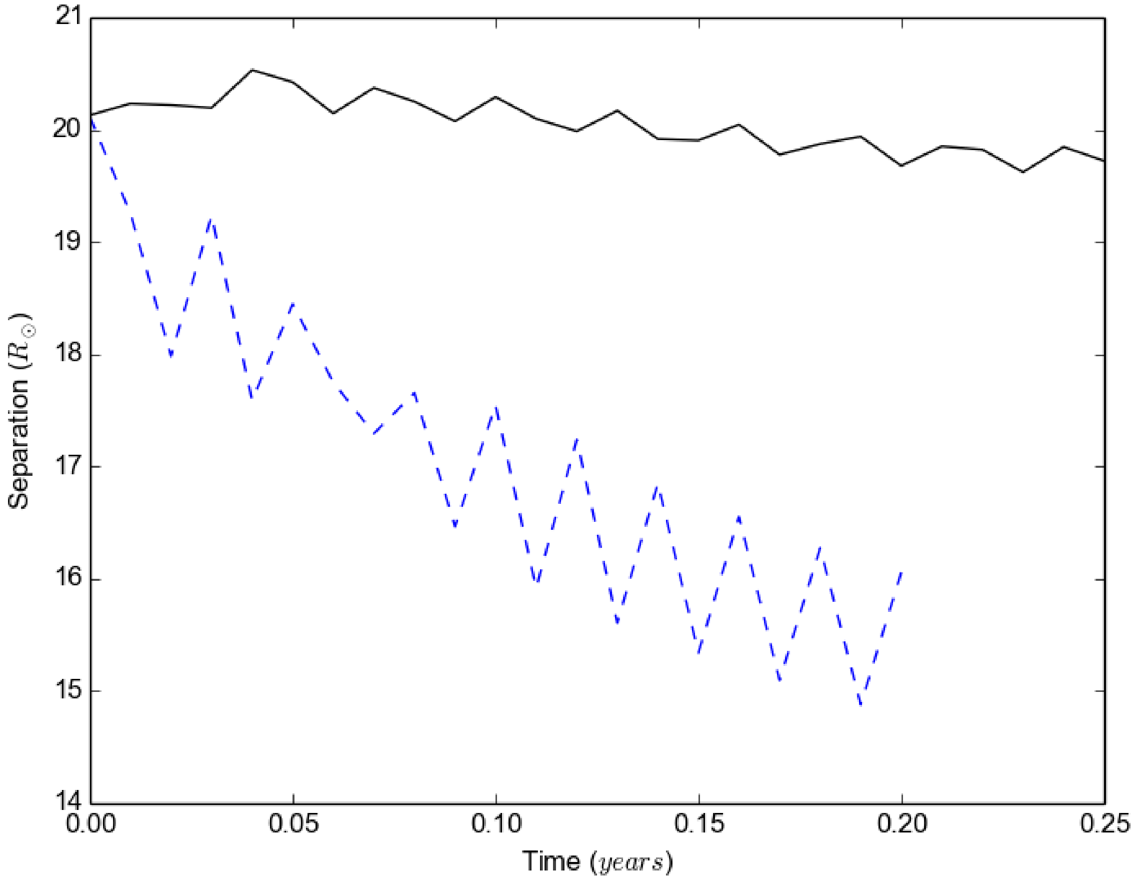


Figure 3.5: Separation time series for two simulations (*nonrot_hot* and *rot_hot*; Table 3.1). The solid black line shows the separation of the core and companion in the simulation with gas having been given an initial rotational velocity (*rot_hot*). The dashed blue line shows the separation of the core and companion in the simulation where gas was given no initial velocity (*nonrot_hot*).

there is a lower velocity contrast between the core and companion and the gas, which results in less gravitational friction as shown by Ostriker (1999).

It was decided that the initial temperature of the gas was unjustifiably high (this is discussed further in Section 5.1). As such the temperature was decreased. This simulation is called *rot_cool* in Table 3.1 and the initial velocity along the orbital plain was set to 0.5 v_{kep}. We found, by looking at the fall-back disk of *rot_cool*, that an initial velocity of 0.5 v_{kep} on the orbital plain was not enough to make mass escape the grid. This also showed that the gas fell into the binary along highly eccentric orbital paths and was not able to produce a disk before interacting with the core and companion. Because of this, we increased the initial velocity to 0.75 v_{kep} in *Cool* and all following simulations. This velocity distribution appeared to produce a shallower fall-back which will be looked at in Chapter 4. The mass of the fall-back disk was $\sim$0.42 $M_{\odot}$, which is very close to our target fall-back mass.

3.4 Production Simulations

The *Cool* simulation satisfactorily produced a fall-back disk, and it was this simulation which was further analysed. This velocity distribution was also used in the simulations *Hot1*, *Hot2* and *Hot3* which were run to study the effects of temperature on the fall-back outcome. These hot simulations will be discussed in Chapter 5. Ideally resolution convergence tests should be run for all our production simulations. However, due to time constraints only one test was run. This simulation was called *HighRes*, and will also be analysed in Chapter 5. We felt justified in taking this short cut by the fact that our resolution was the highest used by Passy et al. (2012) who did carry out a convergence test.

Simulation name	Initial temperature (K)	Initial background density (g cm^{-3})	Run time ($days$)	Comments
nonrot_hot	1.8×10^5	10^{-6}	73	non-rotating
nonrot_cool	3.6×10^4	10^{-6}	44	non-rotating
nonrot_highden	1.8×10^5	2×10^{-4}	28	non-rotating
rot_hot	1.8×10^5	10^{-6}	526	rotating, error in initial velocity distribution.
rot_cool	3.5×10^2	10^{-6}	47	rotating $0.5v_{keplerian}$
Cool	3.5×10^2	10^{-6}	179	rotating $0.75v_{keplerian}$
Hot1	1.8×10^4	10^{-6}	179	rotating $0.75v_{keplerian}$
Hot2	3.5×10^4	10^{-6}	179	rotating $0.75v_{keplerian}$
Hot3	1.8×10^5	10^{-6}	179	rotating $0.75v_{keplerian}$
HighRes	3.5×10^4	10^{-6}	88	high resolution: 384^3, rotating $0.75v_{keplerian}$

Table 3.1: All simulations began with a core-companion separation of 20 R$_\odot$ and with core and companion masses of 0.6 M$_\odot$ and 0.39 M$_\odot$, respectively. All simulations were carried out with a resolution of 256^3 cells (unless specified the in comments column), with a linear size of 144 R$_\odot$. The run time specified is time in the simulation, not the wall-time of the simulation. The temperature was calculated assuming an ideal gas using Equation 2.14.

3.5 Epilogue

The initial simulations of this chapter have been used to optimise our production simulations, the first of which, the *Cool* simulation, will be analysed in detail in the next chapter. These preliminary simulations have provided some basic insight on the in-spiral efficiency and its dependence on fall-back mass. These simulations also informed us on the relationship between temperature of gas and fall-back mass.

4

Simulating a Cool Fall Back Disk

The simulations described in Chapter 3 aimed to test a range of conditions to produce a suitable fall-back disk. Here, we will analyse in detail one of the simulations of a successful fall-back event. This simulation that we analyse is *Cool* as described in Table 3.1. *This chapter and the one that follows are being prepared as a paper manuscript.*

4.1 The Disk's Mass and Angular Momentum

The simulations' setup discussed in Chapter 3 appear to satisfactorily produce a disk falling onto the core and companion. Figure 4.1 shows the first 0.03 years of the fall-back event. In the top right panel of this figure we show a contour that indicates the positions where the gas velocity is equal to the keplerian velocity. Within the cones outlined by these contours the gas is super-keplerian, while outside the cone it is sub-keplerian. The gas within the

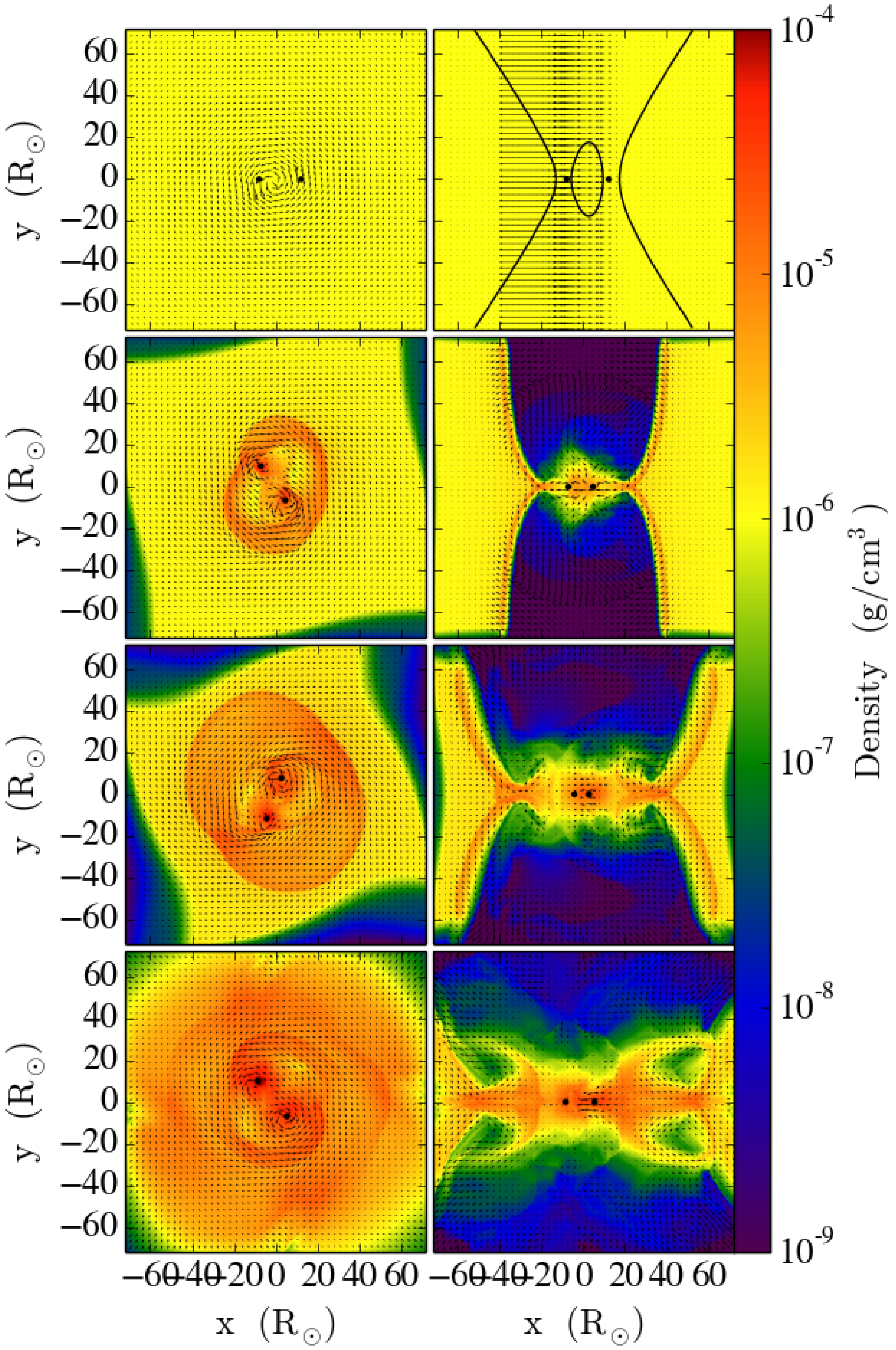


Figure 4.1: Density slices along the orbital plain (left panels) and perpendicular to orbital plain (right panels). Time steps 0.0 (top row), 0.01 (second row), 0.02 (third row), 0.03 yrs (last row). From the right panels we can see that we essentially obtain a disk falling onto the orbital plain.

cones is rapidly evacuated as seen at 0.01 years. Meanwhile, the initially sub-keplerian gas falls onto the orbital plain and towards the core and companion, as seen at 0.01, 0.02 and 0.03 years in Figure 4.1. By 0.03 years the density along the orbital plain is approximately 4 orders of magnitude higher than above and below the plain, so we can confidently say we have created a disk.

The box shape at the edge of the computational domain seen in the left panels of Figure 4.1 as the the gas falls towards the centre is an artefact that results from our chosen boundary condition. The boundary condition applied to our domain is the outflow boundary condition designed for the ZEUS method. These condition does not allow the inflow of gas from outside the box. As a result, when gas falls inward towards the center and there is no gas entering the box filling the region that the in-falling gas has just evacuated, a low density region on the boundary develops. This artifact is quickly overwhelmed by the out flowing gas caused by the particles as see at time 0.03 years.

We estimate the mass of the fall-back disk (Figure 4.2) by dividing our grid into 3 sections, as explained in Section 3.3.1. The middle section is defined by a box that spans 30% of the z-axis and is centred on the orbital plain. The regions above and below contain the remaining computational domain. By creating a mass time series for each of the three regions we can estimate that the fall-back disk contains 0.38 $M_\odot$, as this is the total mass in the central region shortly after the initial evacuation of gas from the grid at 0.04 years. This value is very close to the 0.44 $M_\odot$ of bound mass found by Passy et al. (2012), which is expected to fall back.

Throughout this thesis, time series figures will have 10 vertical reference lines. The blue dashed line indicates what we call the "unbinding event". This corresponds to the time of 0.06 years, determined in Figure 4.7(i). The black solid lines indicate 3 major mass loss events over the course of the simulation. These correspond to times $0.01, 0.04$ and 0.1 years, which are determined in Figure 4.7(iv). The red dotted lines each indicate times of $0.02, 0.03, 0.05, 0.07, 0.08$ and 0.09 years. This is to allow easy comparison between figures.

Passy et al. (2012) calculated that, the total z-component of the angular momentum for the bound gas and core and companion at the end of their simulation was 1.5×10^{52} g cm^2 s $^{-1}$ and 0.6×10^{52} g cm^2 s $^{-1}$. In our simulations, the total angular momentum of the gas and

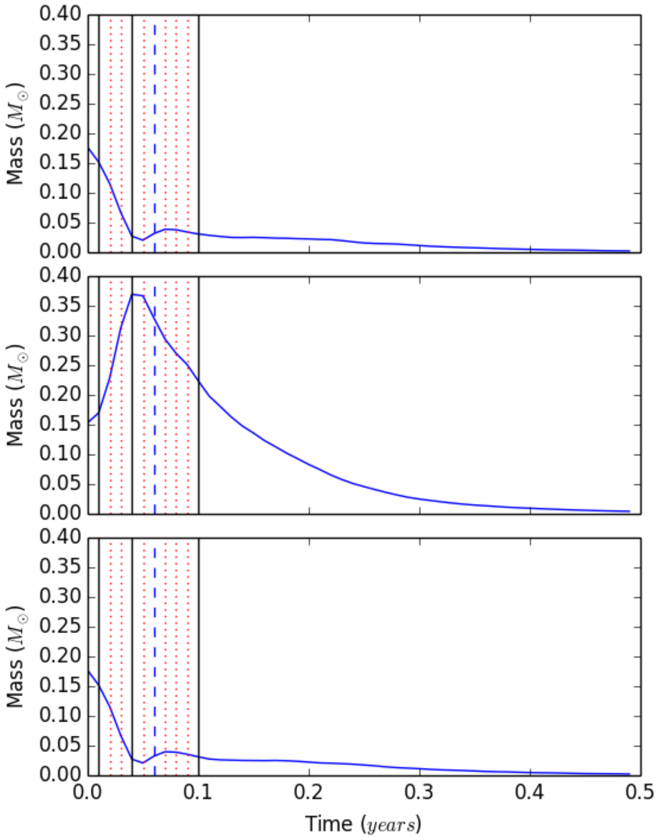


Figure 4.2: Total gas mass within three regions in the grid. The middle panel shows the mass within the 30% of the orbital plain (i.e., $0.35 \leq z \leq 0.65$, where 1.0 is the full length of grid). Top and bottom panels show the mass above and below the central region. The vertical lines are time reference points and are described in Section 4.1.

core and companion begins at 1.5×10^{52} g cm^2 s $^{-1}$ and 0.6×10^{52} g cm^2 s $^{-1}$ respectively (see Figure 4.3; see Appendix A.2.3 and A.2.4 for relevant scripts). This is in very good agreement with Passy et al. (2012).

At the end of the simulations of Passy et al. (2012) the total energy of the system was -0.4×10^{47} ergs. In our simulations, the total energy of the system begins with -0.4×10^{47} ergs (Figure 4.4). Concerning the core and companion, we aimed to have them begin with a kinetic energy of 0.3×10^{47} ergs and a potential energy of -0.5×10^{47} ergs. In our simulation the core and companion begin with a kinetic energy of $\sim 0.2 \times 10^{47}$ ergs, and a potential energy of $\sim -0.4 \times 10^{47}$ ergs. While not in exact agreement, the total energy (kinetic + potential energy) of the core and companion is the same between both simulations.

Concerning the energy of the gas, in our set up our gas has approximately ten times the final kinetic and potential energy of the bound envelope in Passy et al. (2012). We are able to have the same total energy as the simulations of Passy et al. (2012) because our initial thermal energy is about 170 times smaller than that Passy et al. (2012). The consequence of our lower internal energy is addressed in Chapter 5.

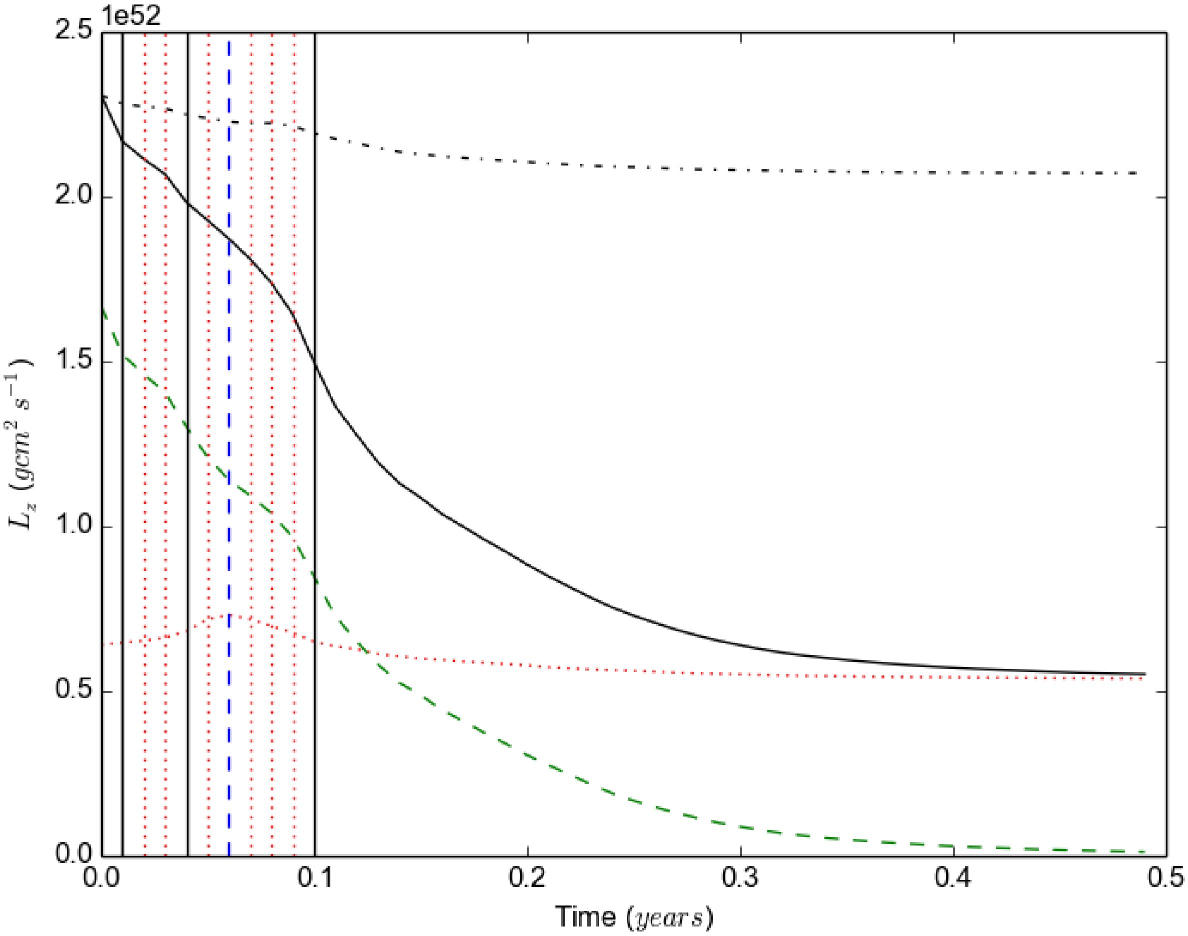


Figure 4.3: The evolution of the total angular momentum throughout the simulation. The solid black line is the total angular momentum of the system. The green dashed line is the angular momentum of the gas and the red dotted line is the angular momentum of the core and companion. The broken black line is the corrected angular momentum for losses from the box. The angular momentum does not remain constant as much gas is flowing off the grid taking angular momentum with it. The vertical lines are time reference points and are described in Section 4.1.

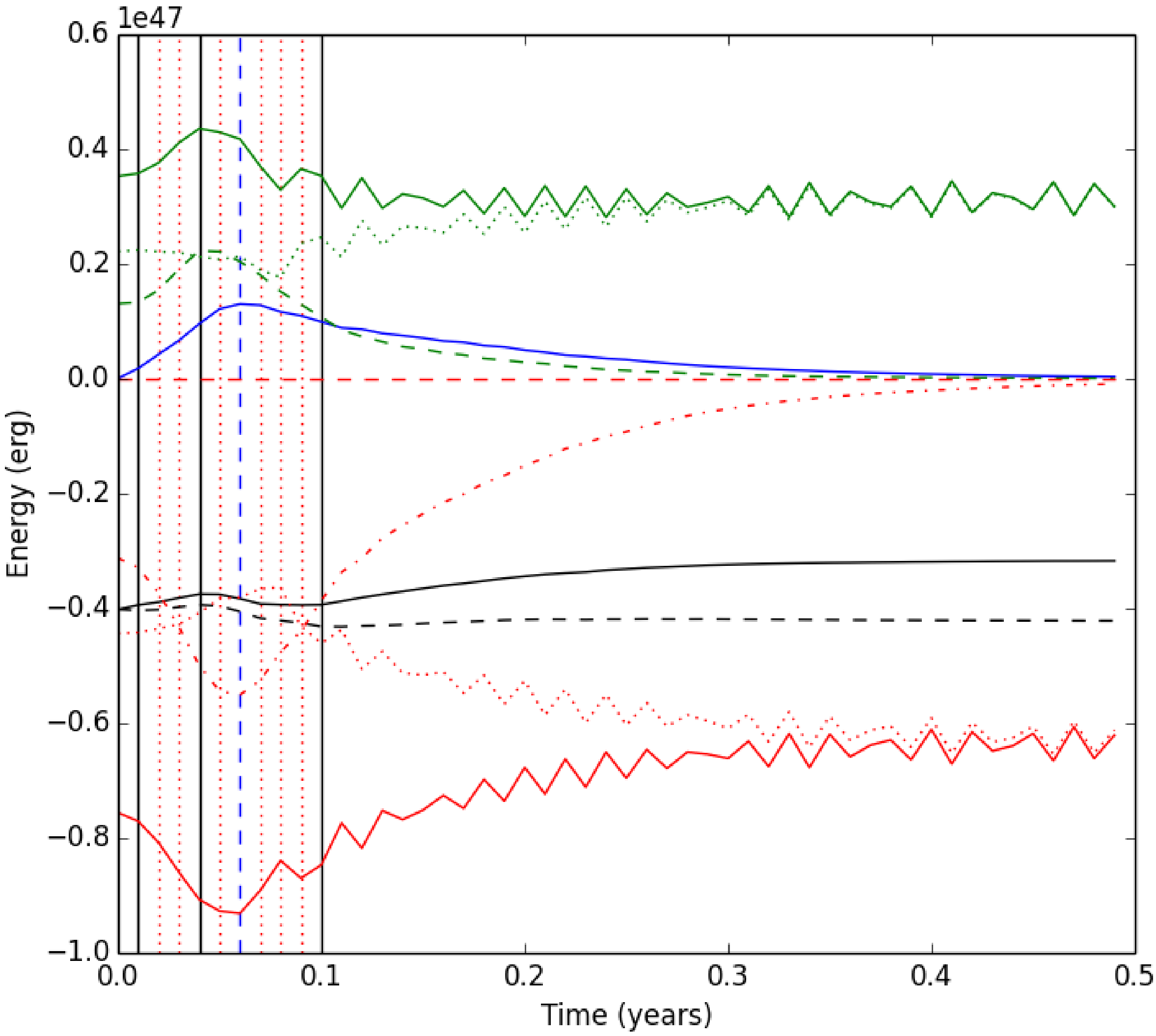


Figure 4.4: Total energies within the box. Plotted are the total energy (solid black line), total energy corrected for mass loss from the box (dashed black line), thermal energy (blue line), kinetic energy of core and companion (dotted green line), gas (dashed green line), the sum of both (solid green line), the potential of the core and companion with respect to each other (dotted red line), the gas on itself (dashed red line), core and companion with respect to the gas (dash-dot red line) and total potential energy (solid red line). The vertical lines are time reference points and are described in Section 4.1.

The energies and angular momenta of Passy et al. (2012) were taken from their figures 8 and 9 and were calculated for their SPH simulations carried out with the same initial set up, and with the same final outcomes as with their grid simulations used here as reference.

4.2 Results

The results of the simulation show promise in that the separation of the core and companion decrease significantly and gas becomes somewhat unbound due to the interaction.

4.2.1 The Separation of the Particles

The interaction resulted in the decrease in orbital separation by ∼30% as shown in Figure 4.5. However, before the decrease in separation occurs, there is an initial increase. This increase in separation occurs over ∼1.5 orbits. This is determined by the orbital periods given by

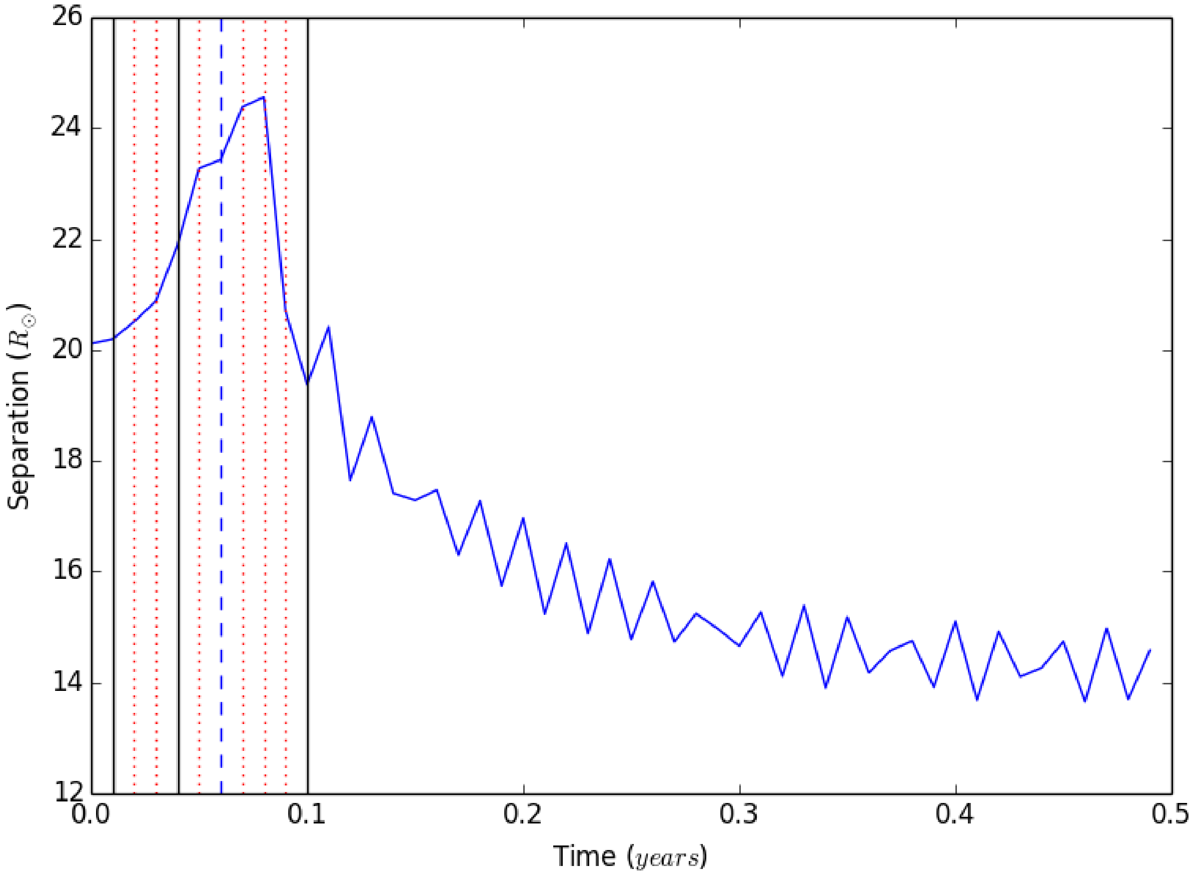


Figure 4.5: The separation of core and companion over time. Over the course of the simulation we see a ∼30% decrease in the separation of the core and companion. The initial increase in separation is due to the increase in eccentricity and is discussed in the text. The vertical lines are time reference points and are described in Section 4.1.

Figure 4.6. This increase in separation is likely due to the sudden eccentricity the system develops due to the highly asymmetric fall back of the mass into the binary. The initial increase in orbital separation may also be due to the relative velocity of the particles and the gas resulting in a gravitational drag that is non uniform in intensity or direction during the early part of the simulation. The extent to which this occurs is discussed further in Chapter 5. We measure the amount of in-spiral from the initial separation of 20 $R_\odot$.

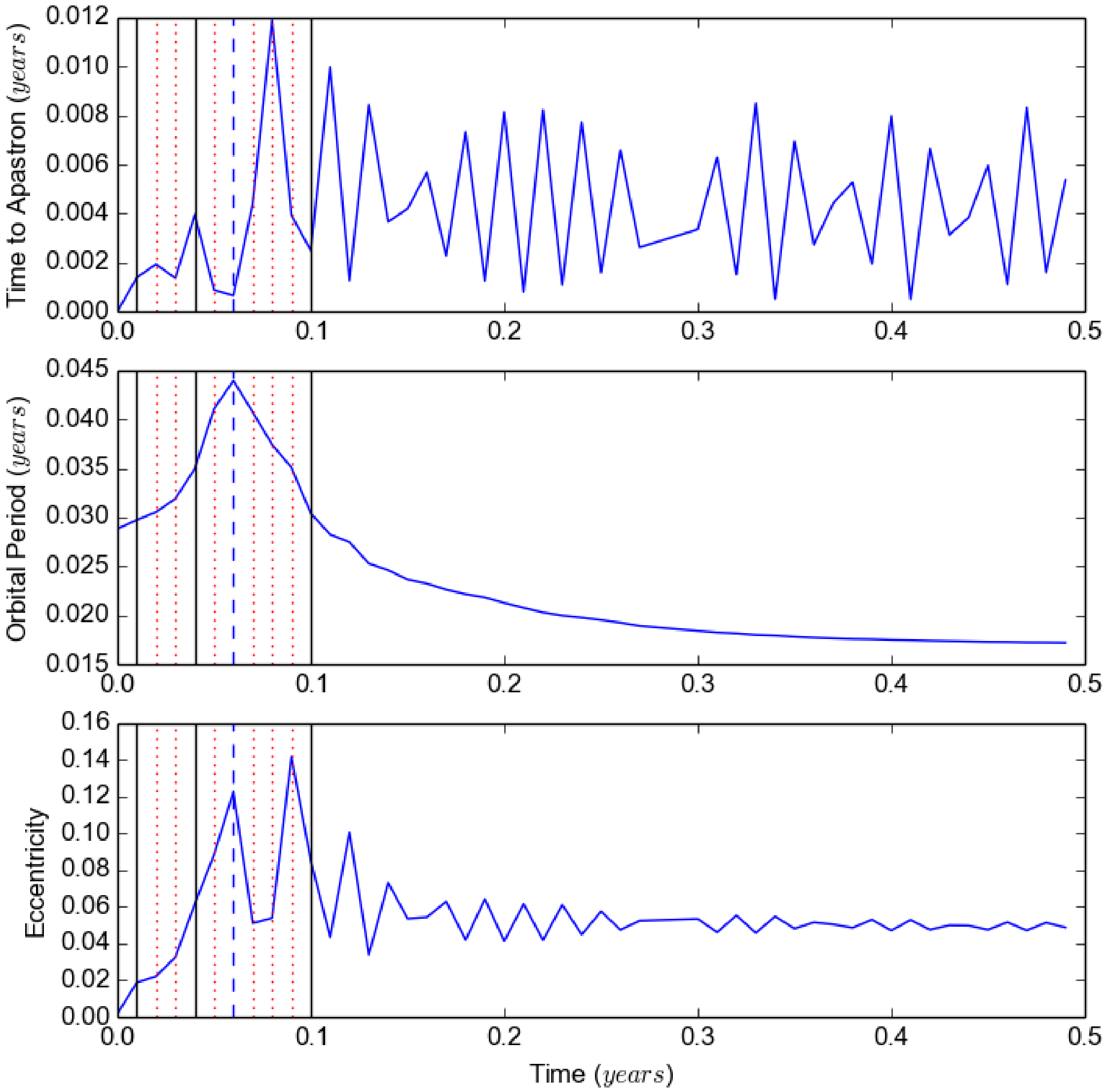


Figure 4.6: The time to apastron at each data output is shown in the top panel. The middle panel shows the orbital period of the system, and the bottom panel shows the eccentricity of the system.

4.2.2 The Determination of Unbound Material

Whether gas has been unbound is determined by summing the potential, kinetic and thermal energies. The bound and unbound gas within the grid is shown in Figure 4.7(i). There we see a peak in unbound mass within the box at 0.06 years. Using the fraction of unbound

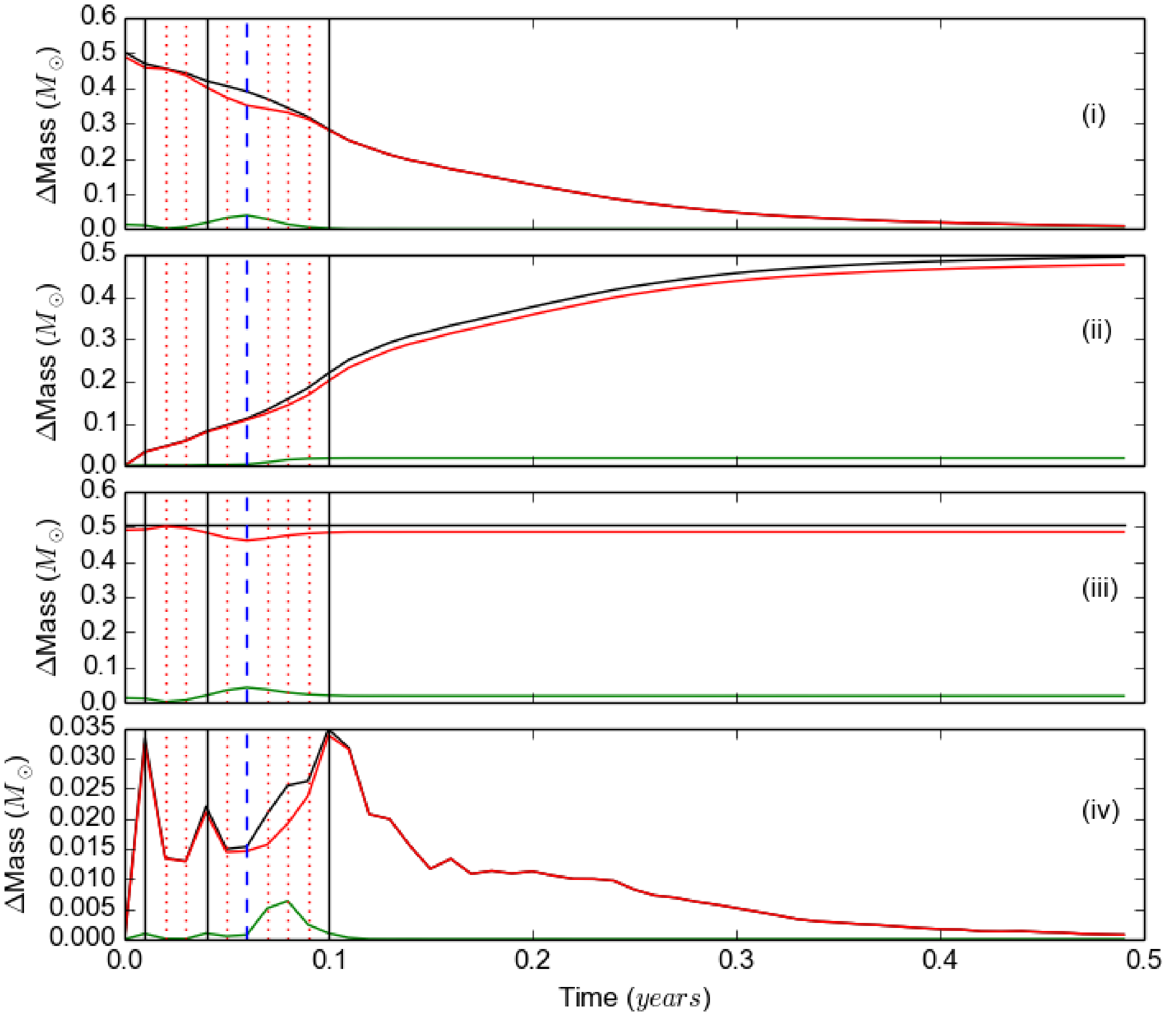


Figure 4.7: All plots show mass within certain regions. (i) Mass within the box, (ii) cumulative mass out of the box, (iii) total mass in the simulation, (iv) mass lost per data output. In all graphs, the black line shows total mass. Red and green lines indicate the bound and unbound mass, respectively. The unbinding event is defined as the peak unbound mass within box as indicated in (i). The major mass loss events are defined by the peaks in (iv). The vertical lines are time reference points and are described in Section 4.1.

mass in a layer one cell thick at the edge of the grid, and the mass lost at the time of each output, the unbound mass leaving the grid was estimated, assuming the unbound ratio remained constant between outputs. The bound and unbound gas out of the grid are shown in Figure 4.7(ii). The mass-loss per data output is shown in Figure 4.7(iv). From these plots we see that the unbound mass from the unbinding event appears to leave the grid 0.01 to 0.02 years later. Summing together the unbound gas fraction within the box and the mass that has left the box (see Figure 4.7(iii)) we see that the unbound gas fraction levels off to around 4% of the total mass of the gas system. This translates to 5% of the fall-back disk mass becoming unbound. How we measure unbinding efficiency is discussed in detail in Chapter 5. The decreases in the total unbound mass in the system as seen in Figure 4.7(iii) seems unusual, as it would be assumed that once mass becomes unbound it remains unbound. This is due to our decision of including the thermal energy in the calculation of whether mass is bound or not. The consequence of this inclusion of thermal energy is a parcel of unbound gas will dissipate its energy to the surrounding bound gas and on average result in bound material.

4.3 Energy and Angular Momentum

The energy conservation is reasonable (Figure 4.4). The increase in energy by 8.5×10^{45}erg or 21% is due to the loss of bound gas from the grid. This loss was estimated by calculating the average specific energy over the entire grid boundary layer (i.e, the layer one cell thick on all 6 domain faces). We refined our analysis further by determining what amount of energy was lost perpendicular to the orbital plain (the faces of the domain above and below the orbital plain), and along the plain (the other four box faces). The fraction of the total mass within the 4 box boundary faces perpendicular to orbital plain, f_{perp}, and the fraction of the mass within the two box faces parallel to the orbital plain, f_{plain}, were then calculated ($f_{perp} + f_{plain} = 1$). Assuming these mass and energy fractions remained relatively constant over the period between data outputs, the energy lost perpendicular and parallel to the orbital plain were estimated in the following way:

$$E_{loss,perp} = E_{spec,perp} \times f_{perp} \times M_{loss}, \tag{4.1}$$

$$E_{loss,plain} = E_{spec,plain} \times f_{plain} \times M_{loss}, \tag{4.2}$$

where M_{loss} is the lost mass between data outputs. This gives the energy loss at the time of each data output to be that shown in Figure 4.8. From this we see that most of the energy leaves the system along the orbital plain. This is likely because most of the gas leaves along the orbital plain, as observed by Passy et al. (2012) and Sandquist et al. (1998), which would carry a large amount of energy. Adding the energy lost calculated with this method to the energy within the box, brings the total energy to a fairly constant value (the black dashed line in Figure 4.4), with a maximum excursion ∼5% from the initial total energy. This is comparable to the degree of conservation seen in Sandquist et al. (1998) and Passy et al. (2012).

Next we considered the conservation of angular momentum. Only the z-component of the angular momentum was considered, due to the rotation of the system, this component is representative of the total angular momentum. This also provides an easy comparison with the work of Passy et al. (2012), who also calculated this. The angular momentum (**L**) was calculated using:

$$\mathbf{L} = \mathbf{r} \times \mathbf{p} \tag{4.3}$$

which is simply the cross product of **r**, the location of the parcel of gas from the center of mass and **p**, the linear momentum of that mass. Figure 4.3 shows the core and companion increase in angular momentum until the unbinding event (blue dashed reference line), then decreases to a plateau. It is expected that the core and companion gain angular moment because as we see in Figure 4.5, the separation of the core and companion increase. The increase in angular momentum of the core and companion must come from the in-falling gas, but this is difficult to see in Figure 4.3, as mass loss from the computational domain is also carrying a lot of angular momentum out.

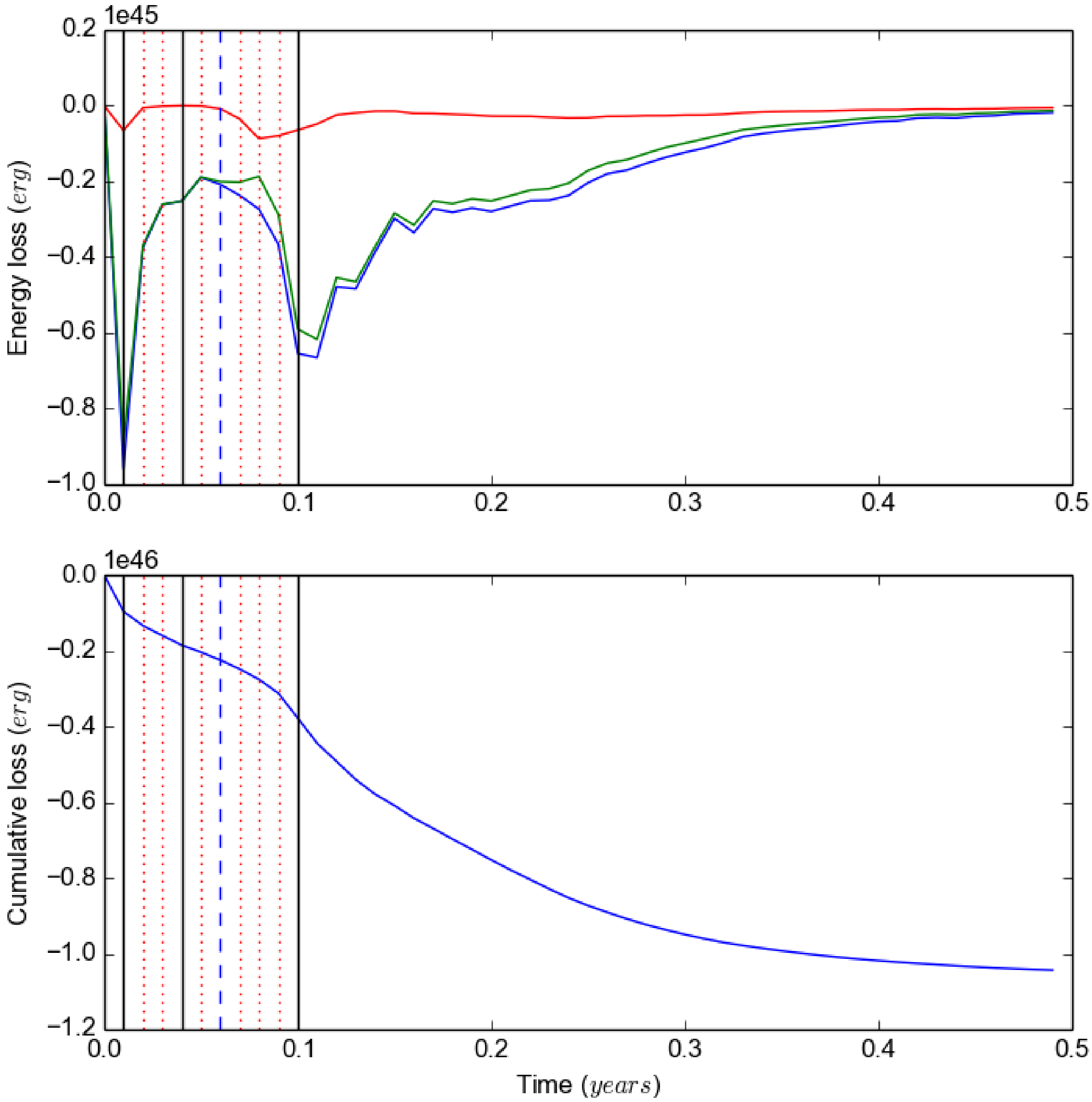


Figure 4.8: Energy loss per data output as estimated using Equation (4.1) and Equation (4.2). The vertical lines are time reference points and are described in Section 4.1.

We can still get an idea of when the major mass loss events occur as the gradient of angular momentum loss for the gas is greatest during these events. We can also see that the gradient of gas momentum loss is at its most shallow at the time of the unbinding event, where the core and companion begin to lose angular momentum to the gas. Ideally the angular momentum of the gas should increase at the time of unbinding, but the simulations is still losing significant mass at that time, so this is poorly captured.

We calculate angular momentum loss similarly to the calculation of energy loss from the grid. The specific angular momentum per cell was calculated for each face of the computational domain. This method assumed that the angular momentum lost between data outputs is relatively constant. The resultant angular momentum loss is added to the total angular momentum shown in Figure 4.3. We see that even the corrected total angular momentum is still decreasing, with a maximum excursion of ∼5% from the initial total angular momentum. This is probably due to the assumption that angular momentum loss is constant between data outputs, also angular momentum is not explicitly conserved with our chosen simulation method.

4.4 Time to the Next Fall-Back

We showed in Section 4.2.2 that most of the fall-back mass remained bound to the system by the end of the simulation. As is the case with the initial common envelope simulations, this bound gas is also bound to come back to the system. It was indeed unlikely that a single fall-back event would completely unbind the remaining bound envelope, and more likely that a series of fall-back events would occur. This is discussed further in Chapter 6. However, suspecting that the gas will return to the system several times, it may be desirable to simulate another fall-back event. When to begin simulating another fall-back event would be determined by the time taken for the gas from this fall-back event to return to the computational domain.

Here we decided to calculate the time for gas to fall back into the computational domain for our fall-back event. For each data output, we determined the velocity of the gas in each cell along the border of the grid (see Appendix A.2.5). It was assumed that the majority of the gas within the border was leaving the grid, hence had a positive radial velocity. The fall-back time was calculated numerically using simple projectile motion. Looking at Equation (4.4), the change in distance (Δr) from the center of the computational domain for the mass in a cell was calculated over a time step of $\Delta t = 1$ week, where v is the velocity of the parcel of mass. The acceleration was taken to be only due to core and companion

$(M_{core} + M_{comp})$ as calculated in Equation (4.5). At each step the velocity and position of the mass was updated, until the mass would fall back into the computational domain.

$$\Delta r = v\Delta t + \frac{1}{2}a\left(\Delta t\right)^2 \tag{4.4}$$

$$a_{grav} = -\frac{G\left(M_{core} + M_{comp}\right)}{r^2} \tag{4.5}$$

This calculation is meant to give an estimate of the fall-back time of the bound gas. This was carried out for all the cells on the border of the computational domain. We have plotted the estimated fall-back time to the computational domain vs. the mass in each at 0.08 years in Figure 4.9. This time was selected because, looking at panel (iv) in Figure 4.7, this appears to be the time when the gas from the unbinding event leaves the grid.

From Figure 4.9 we see that the return time to the computational domain has a large spread from a few weeks, to more than two years. Determining when the bulk of the gas from this interaction will return is uncertain. We see that the cells have lower mass above and below the orbital plain (blue triangles in Figure 4.9), and this is expected as the density is very low in these regions. This low mass gas appears to have a wide range of fall-back times from a few weeks to over 2 years. Along the orbital plain the gas is generally denser, hence the cells have a higher mass. With this understanding, we can treat the cells with high mass as that which is expelled along the orbital plain, and this mass has a generally shorter fall-back time of ≤ 10 weeks. Some intermediate mass cells that leave the domain at some angle appears to have been lifted further and fall-back times for this gas is as high as one year.

With this large spread in fall-back times of the gas, determining when the next fall-back event would take place requires some careful thinking.

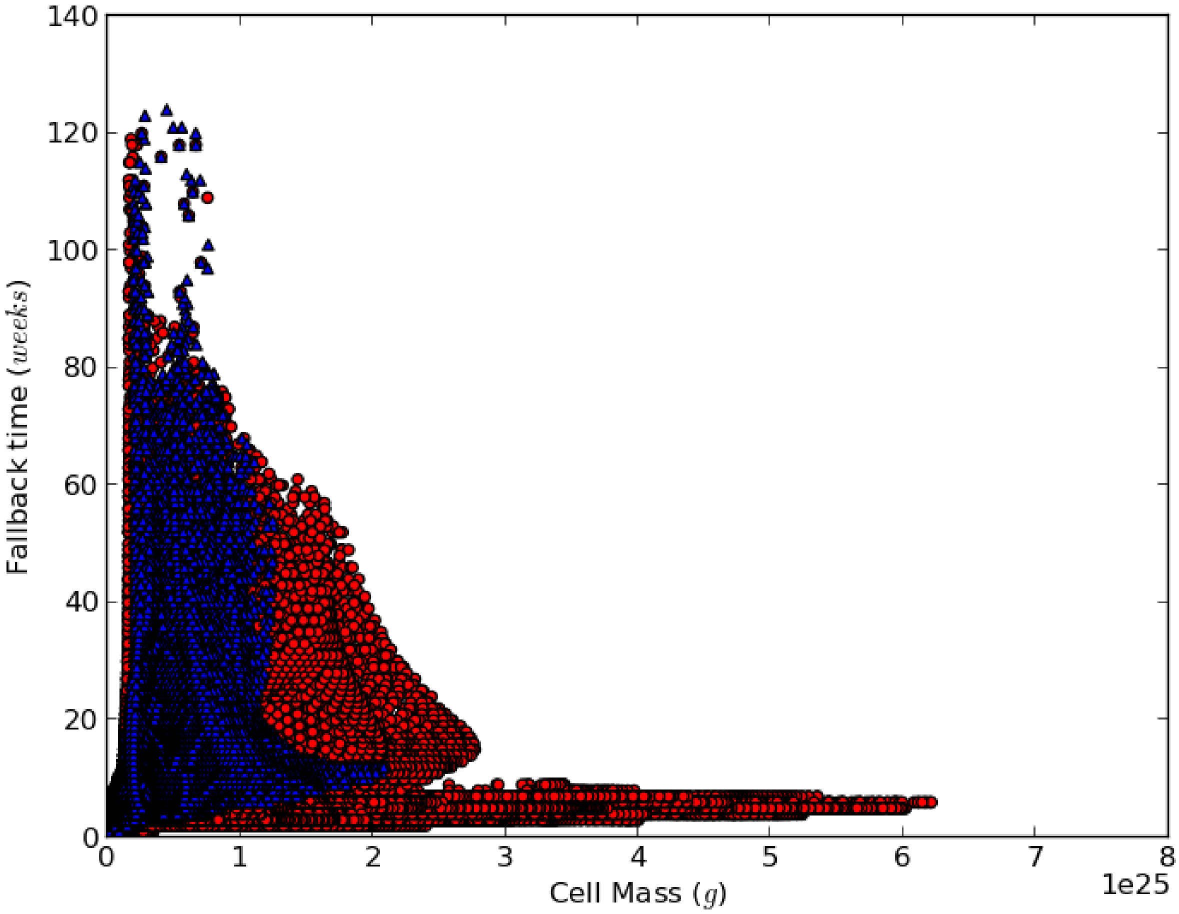


Figure 4.9: The estimated fall-back time to return to the grid for each cell on the border of the computational domain at time 0.08 years vs. the mass in the cell. The fall-back time for the gas on the orbital plain faces are shown by red circles, and for the gas on the perpendicular faces, by blue triangles.

4.5 Conclusions

Our simulation of a fall-back disk supports this as a mechanism to unbind additional envelope gas while decreasing the orbital separation. The fall-back disk produced a decrease in binary separation by ~30% (From 20 $R_\odot$ to 14 $R_\odot$), while unbinding about 5% of the fall-back gas (approximately $0.022 M_\odot$). This single fall-back event is clearly not sufficient to reproduce the observations, but multiple fall-backs may result in the system converging to the separations observed in post common envelope binaries, while at the same time unbinding the envelope.

4.6 Epilogue

This simulation indicated some characteristics which need further investigation. The consequence of our lower internal energy was the initial temperature of the gas in our simulation was 350 K. This resulted in major shock heating of the system very early in the simulation. This issue will be addressed in Chapter 5.

5

Investigation of the Role of Temperature on Fall-Back Disk Dynamics

The simulations presented in Chapter 4 aimed to reproduce a fall-back disk that would have the appropriate parameters such as mass and angular momentum. Of all the parameters that could be varied, temperature is one that may make a difference to the fall-back event. We base this statement on the observation that shocks develop readily in our simulation of Chapter 4 (called *Cool* in Table 3.1) and that these may alter the in-fall since the strength of the gravitational drag changes substantially for supersonic motions (Ostriker, 1999). Here we therefore investigate how the initial temperature of the gas falling back affects the outcome of the fall-back event.

5.1 The Thermal Energy of the Fall-Back Disk

In the simulation of Chapter 4 the initial temperature of the gas was a cold 350 K. This was calculated using the method described in Section 2.3. The simulations carried out by Passy et al. (2012) end with gas temperatures of ∼50 000 K in a box 2 AU on a side centred around the core and companion. In our *Cool* simulation initial shocks develop because the sound speed is very low (see Figure 5.1) and heat up the gas to temperatures of ∼50 000 K, close to the final temperature reached in the simulations of Passy et al. (2012) because that is the virial temperature of them system.

According to Tocknell et al. (2014) the fall-back of bound gas in the Passy et al. (2012) simulation would take ∼2 − 14 years, starting from an average distance of 4.5×10^3 R$_\odot$. Would this time be sufficient for the gas to cool? Or would the gas still be hot when it falls back onto the core and companion? The expanding common envelope is expected to cool adiabatically and will heat back up adiabatically during the in-fall. The question is whether it would have time to cool radiatively when it reaches maximum expansion?

To estimate this we first calculate the temperature of the envelope when it expands to 4.5×10^3 R$_\odot$. If we expect the gas to expand adiabatically, the final temperature can be estimated using:

$$T_f = T_i \left(\frac{V_i}{V_f} \right)^{\gamma - 1}, \tag{5.1}$$

where T_i and T_f are the initial and final temperature, and similarly V_i and V_f are the initial and final volumes. γ is the adiabatic index, which in our case is $\frac{5}{3}$. We assume the final volume to be a sphere of radius 4.5×10^3 R$_\odot$, which gives a volume of $V_f = 3.8 \times 10^{11}$ R$_\odot^3$. It is not entirely clear what the initial volume would be. Assuming the radius of the gas volume expands at a constant rate, and using the estimated fall-back time to also be the time taken to expand to 4.5×10^3 R$_\odot$, we can estimate the initial volume. As the calculated fall-back time for gas from a mean distance of 4.5×10^3 R$_\odot$ is $2 - 14$ years, we take a mean expansion time of 8 years. The simulation of Passy et al. (2012) stabilises after half a year, therefore half a year of the full expansion time is approximately 1/16. We can estimate the volume at the end of the simulation of Passy et al. (2012) to have a radius approximately 1/16 of

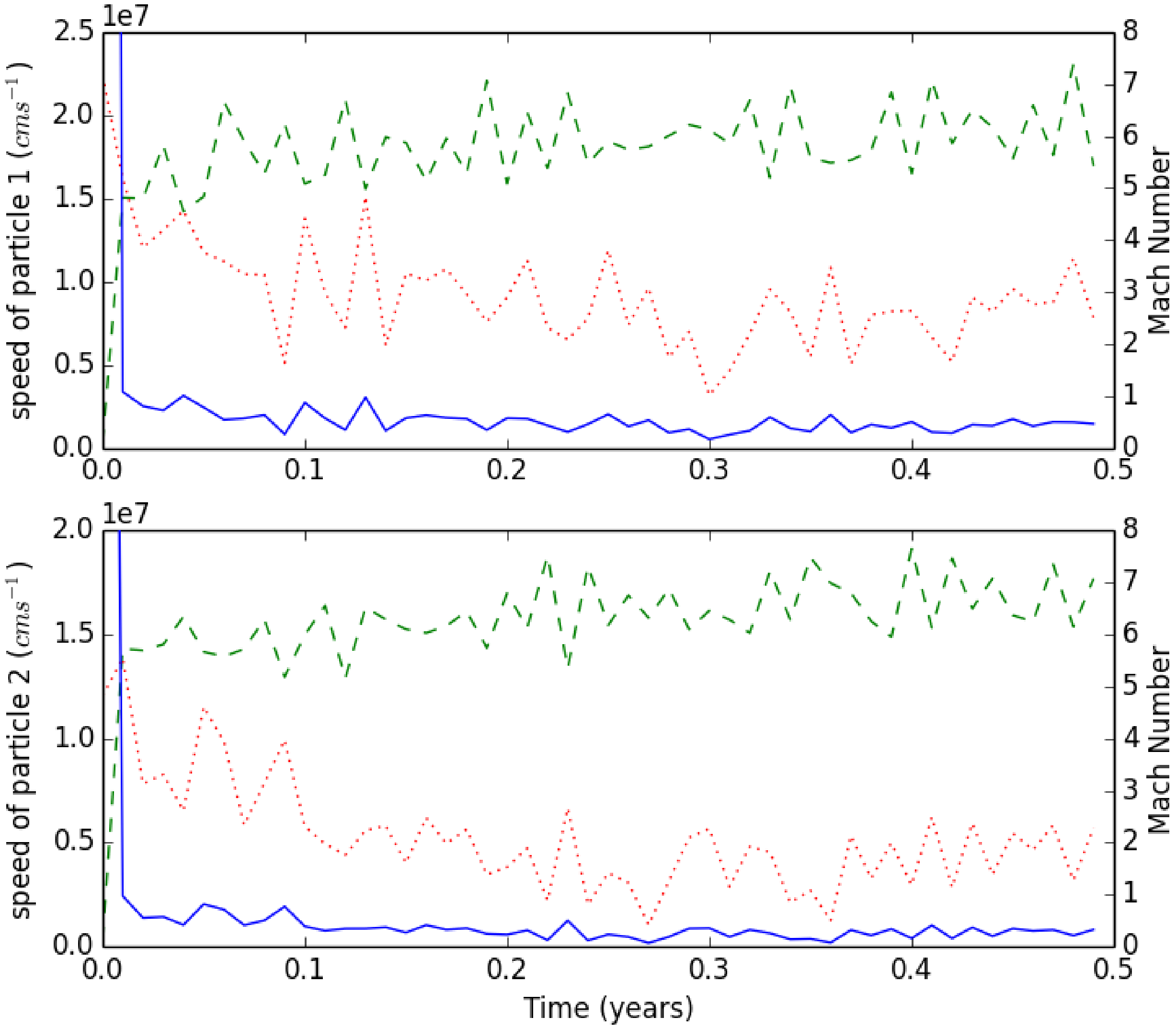


Figure 5.1: In both panels, the dashed green line shows the sound speed in the sphere around the core and companion. The red dotted line shows the speed of the core and companion, relative to its surrounding gas. The values of the speed are indicated on the left axis. The solid blue line shows the Mach number of the core and companion. The values are indicated on the right axis.

the final radius. This gives an initial radius of 281 $R_\odot$, which corresponds to a volume of $V_i = 9.3 \times 10^7$ $R_\odot^3$. We take $T_i = 50\,000$ K, as it is the final temperature of the remaining gas in the simulation of Passy et al. (2012). With these values, the temperature of the gas at maximum expansion is found to be ~ 195 K.

Using this value of temperature and assuming isotropic expansion we can estimate the luminosity and the thermal (or Kelvin-Helmholtz) time scale of expanded gas. The luminosity is calculated by using the Stefan-Boltzmann law:

$$L = 4\pi R^2 \sigma T_{eff}^4, \tag{5.2}$$

where R is the radius of the emitting surface, T_{eff} is the effective temperature and σ is the Stefan-Boltzmann constant. With the radius of the expanded gas being $R = 4.5 \times 10^3$ $R_\odot$ and the adiabatically cooled temperature of 195 K, we find the luminosity of this object to be approximately $L = 26$ $L_\odot$. The thermal time scale is then found using:

$$\tau_{therm} = \frac{U}{L}, \tag{5.3}$$

where U is the internal energy of the envelope of gas. Looking at figure 9 of Passy et al. (2012), we find that $U = 10^{46}$ ergs at the end of their simulation. Some of this energy is used to expand the gas to 4.5×10^3 $R_\odot$, but we find this amount to be negligible, and we therefore take U to be 10^{46} ergs also at this larger volume. Using our calculated luminosity the thermal time scale of the expanded gas is found to be ~3100 years. Given the calculated fall-back time of the common envelope gas is at most a couple of decades it would seem that the envelope would not have enough time to radiate away a significant amount of energy before returning to the binary system. However, this calculation assumes that the expanding envelope is optically thick, hence only radiating thermally from the outer layers. If the gas is optically thin, more energy can be radiated as it expands, producing different results.

Our calculation of the temperature of the expanded envelope may not be accurate. We used the initial temperature of 50 000 K as this is the final temperature of the simulations of Passy et al. (2012), but at this stage there has been significant mass loss from the grid that could also carry away thermal energy. Also, the initial volume is not very well constrained due to our lack of knowledge of the gas outside of the computational domain. Nonetheless, using our calculations, the expanded gas would radiate away its internal energy in the space of a decade if it had a luminosity of $L = 8000$ $L_\odot$. Maintaining a radius of $R = 4.5 \times 10^3$ $R_\odot$, the gas would need a temperature of ~820 K to achieve this luminosity. This temperature seems reasonable. There are many unknown factors in determining the thermal time scale of the common envelope gas making it difficult to determine whether the gas will return to the system cold or hot.

Initially the core and companion have high Mach numbers ($\mathcal{M} > 40$) as shown in Figure 5.1. Mach numbers were calculated using the sound speed of the gas surrounding the core and companion (see Appendix A.2.6). This was calculated by making a 1 $R_\odot$ sphere around the core and companion and taking the average sound speed within that sphere. We also used a second method because we considered that the gas in front of the particle had not interacted with the particle yet, while the gas behind the particle had. With this in mind, we calculated the sound speed of the gas using a box of 3^3 cells, centred on a cell located two cells in front of the cell containing the particle. In *Enzo*, the position of the particle is not necessarily at a cell's centre. The particle's actual position could be on the boundary of two cells, hence the presences of the particle may be felt equally by the gas in both cells. So instead of taking the cell right in front of the rounded off cell position of the particle, we chose the next cell in front of the particle, just to be sure we selected gas that had not interacted with the particle.[1] Also when calculating the Mach number of the core and companion, the velocity of core and companion was corrected to be relative to the velocity of gas used to calculate the sound speed. This was done simply by subtracting the bulk velocity of the gas from the velocity of the particle.

The initial Mach numbers of the 0.39 $M_\odot$ giant core and 0.6 $M_\odot$ companion in the *Cool* simulation were $\mathcal{M} = 79$ and $\mathcal{M} = 43$, respectively. These high Mach numbers caused dramatic shock heating of the gas in the box (shown in Figure 5.2), as we have explained. The Passy et al. (2012) simulations end with a gas temperature within the box of approximately 50 000 K. If the gas ejected from the computation domain was hot, and it did not have time to cool before it fell back, then maybe it would still be hot. We therefore wondered whether starting with a hot gas would alter our conclusions.

---

[1] this is strictly correct only for supersonic particles. However choosing one or two cells in front did not change our conclusions appreciably

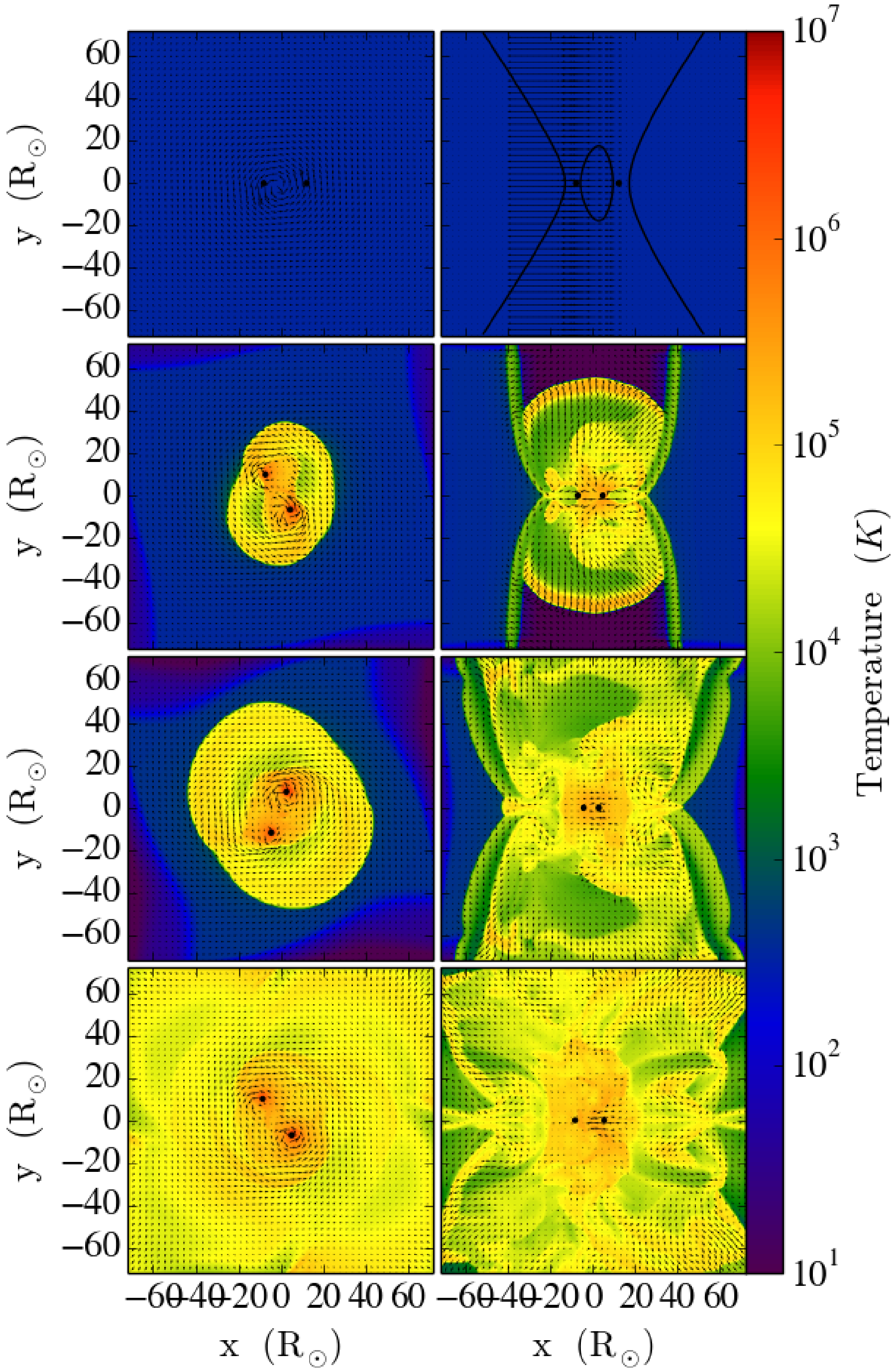


Figure 5.2: Left panels show the temperature evolution on a slice along the orbital plain and the right panels show the temperature evolution on a slice perpendicular to the orbital plain. Time steps 0 yrs (top row), 0.01 yrs (second row), 0.02 yrs (third row), 0.03 yrs (last row).

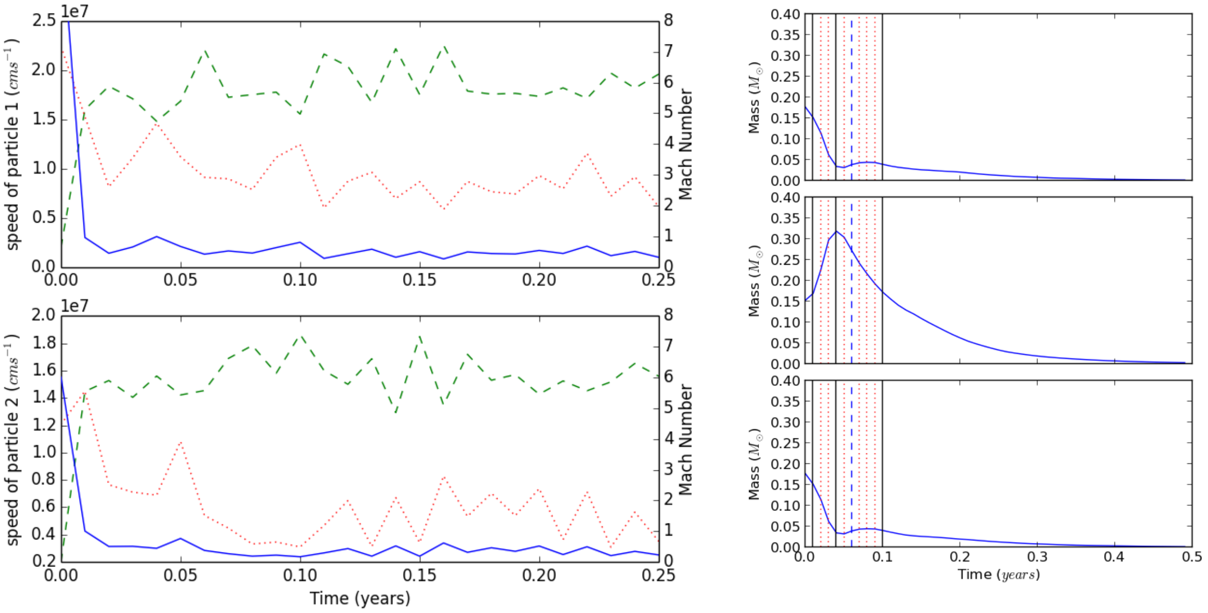
5.1.1 Setup for Hotter Simulations

The falling gas may still be hot when it falls back. We therefore ran 3 additional simulations with higher initial temperature, but maintained the same initial density of 10^{-6} g cm^{-3} and initial velocity setup (see Section 3.2). Maintaining the same velocity distribution results in less fall-back mass with higher temperature. Despite not changing this velocity, these simulations provided the opportunity to investigate general relationships between the temperature of the gas and evolution of the system. These simulations were called *Hot1*, *Hot2* and *Hot3*, and they are listed alongside the other simulations in Table 3.1. Below we discuss them further, and in Section 5.2 we carry out a comparison of the hot simulations with the cool, along with a resolution test.

The ***Hot1*** Simulation

This simulation has an initial constant temperature of 1.7×10^4 K calculated from a value of $U = 3.6 \times 10^{45}$ erg with the same initial mass and number density as the *Cool* simulation. The main goal of increasing the initial temperature of the simulation was to bring the core and companion into a subsonic regime and to prevent the initial shock heating. With hindsight we could have pre-calculated the target temperature. However, initially we had not zeroed in on the required criteria and were more generally interested in understanding the effects of a higher initial temperature on the results. Hence we calculated 3 "hot" simulations.

The Mach number evolution of the core and companion is shown in Figure 5.3a. The initial Mach numbers of the core and companion are $\mathcal{M} = 6$ and $\mathcal{M} = 11$, respectively, so this system also experienced initial shock heating. The amount of mass falling onto the orbital plain is less than for the *Cool* simulation of Chapter 4, being approximately 0.33 $\mathrm{M}_{\odot}$ (shown in Figure 5.3b, cf. 0.38 $\mathrm{M}_{\odot}$ and Figure 4.2, depicting the *Cool* simulation). This is expected because hot gas has a higher pressure and expands away.



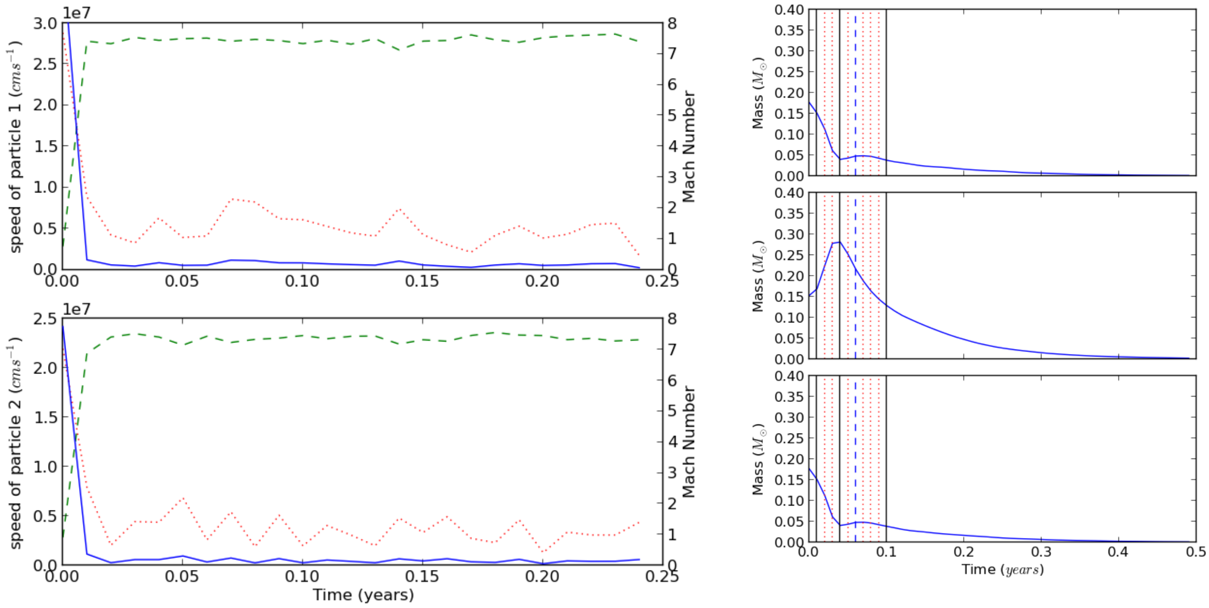
(a)

(b)

Figure 5.3: (a) The dashed green line shows the sound speed of the gas for the *Hot1* simulation.. The red dotted line shows the speed of the core and companion, relative to its surrounding gas. The values of the speed are indicated on the left axis. The solid blue line shows the Mach number of the core and companion. The values are indicated on the right axis. (b) The middle panel shows the mass within the 30% of the orbital plain. Top and bottom panels show the mass above and below the central region. The vertical lines are time reference points and are described in Section 4.1.

The *Hot2* Simulation

In this simulation, we doubled the internal energy of the system, hence doubling the initial temperature. Therefore *Hot2* was set up with an initial temperature of 3.5×10^4 K. This temperature is still lower than the final temperature of the simulations of Passy et al. (2012), and also resulted in shock heating. The initial Mach numbers of the core and companion were $\mathcal{M} = 5$ and $\mathcal{M} = 10$, respectively. The Mach number evolution of the core and companion is shown in Figure 5.4a. The fall-back disk mass is shown in Figure 5.4b, with the disk having approximately 0.28 $M_\odot$.

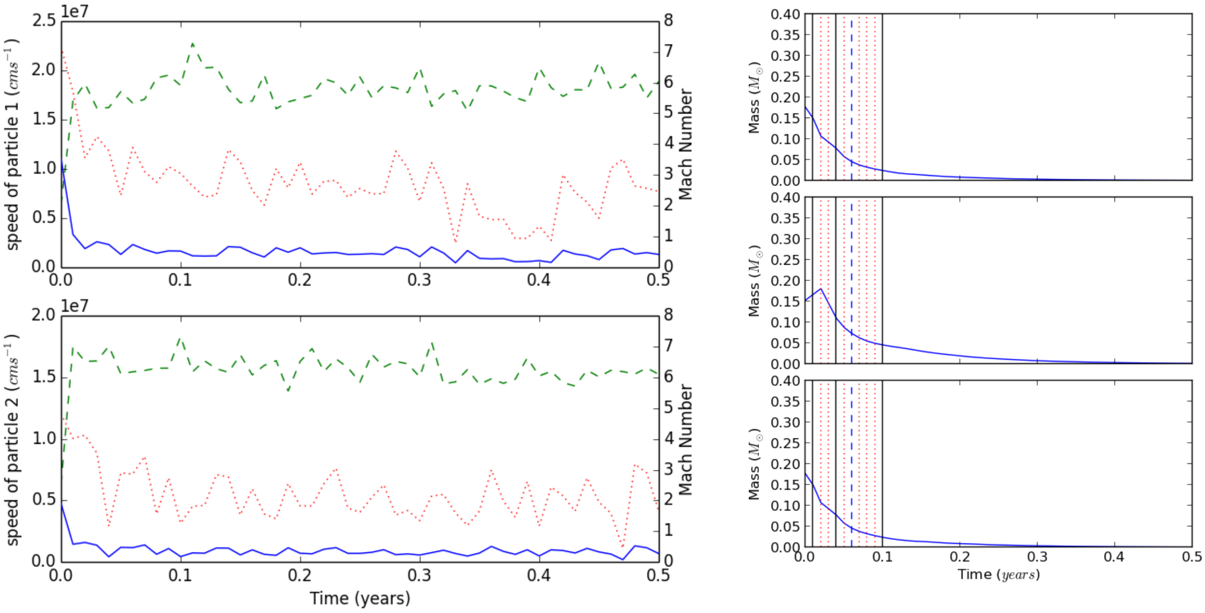


(a) (b)

Figure 5.4: (a) The dashed green line shows the sound speed of the gas for the *Hot2* simulation. The red dotted line shows the speed of the core and companion, relative to its surrounding gas. The values of the speed are indicated on the left axis. The solid blue line shows the Mach number of the core and companion. The values are indicated on the right axis. (b) The middle panel shows the mass within the 30% of the orbital plain. Top and bottom panels show the mass above and below the central region. The vertical lines are time reference points and are described in Section 4.1.

The *Hot3* Simulation

The core and companion in the simulations *Hot1* and *Hot2* were initially supersonic despite the larger temperatures of the gas. We therefore decided to increase the temperature further. This time we increased the internal energy tenfold compared to the *Hot1* simulation. This resulted in an initial temperature of 1.7×10^5 K. This temperature is a factor of 3 higher than the 50 000 K of Passy et al. (2012), but with this initial temperature the Mach numbers of the core and companion are $\mathcal{M} = 0.6$ and $\mathcal{M} = 1.0$, respectively. The Mach number time series is shown in Figure 5.5a. Despite these very high temperatures the system is not completely subsonic. With these initial higher temperature, much gas expands away and even less mass falls into the disk (see Figure 5.5b), with the fall-back disk having about 0.18 M$_\odot$. Below we examine the outcome of these three simulations and compare them to the *Cool* simulation.



(a)

(b)

Figure 5.5: (a) The dashed green line shows the sound speed of the gas for the *Hot3* simulation. The red dotted line shows the speed of the core and companion, relative to its surrounding gas. The values of the speed are indicated on the left axis. The solid blue line shows the Mach number of the core and companion. The values are indicated on the right axis. (b) The middle panel shows the mass within the 30% of the orbital plain. Top and bottom panels show the mass above and below the central region. The vertical lines are time reference points and are described in Section 4.1.

5.2 Results and Comparisons

These simulations provided the opportunity to investigate the dependence of the evolution of the system on the initial gas temperature.

5.2.1 The Evolution of the Orbital Separation

Comparing the separation evolution of each of these three hot simulations with that of the *Cool* simulation in Chapter 4, we see that the core and companion in the cold simulation in-spiral the farthest (see Figure 5.6). Orbital separation changes depend on the force experienced by the core and companion. This force is primarily due to gravitational friction or drag (Ricker & Taam, 2012). Gravitational friction depends on the ambient density and the

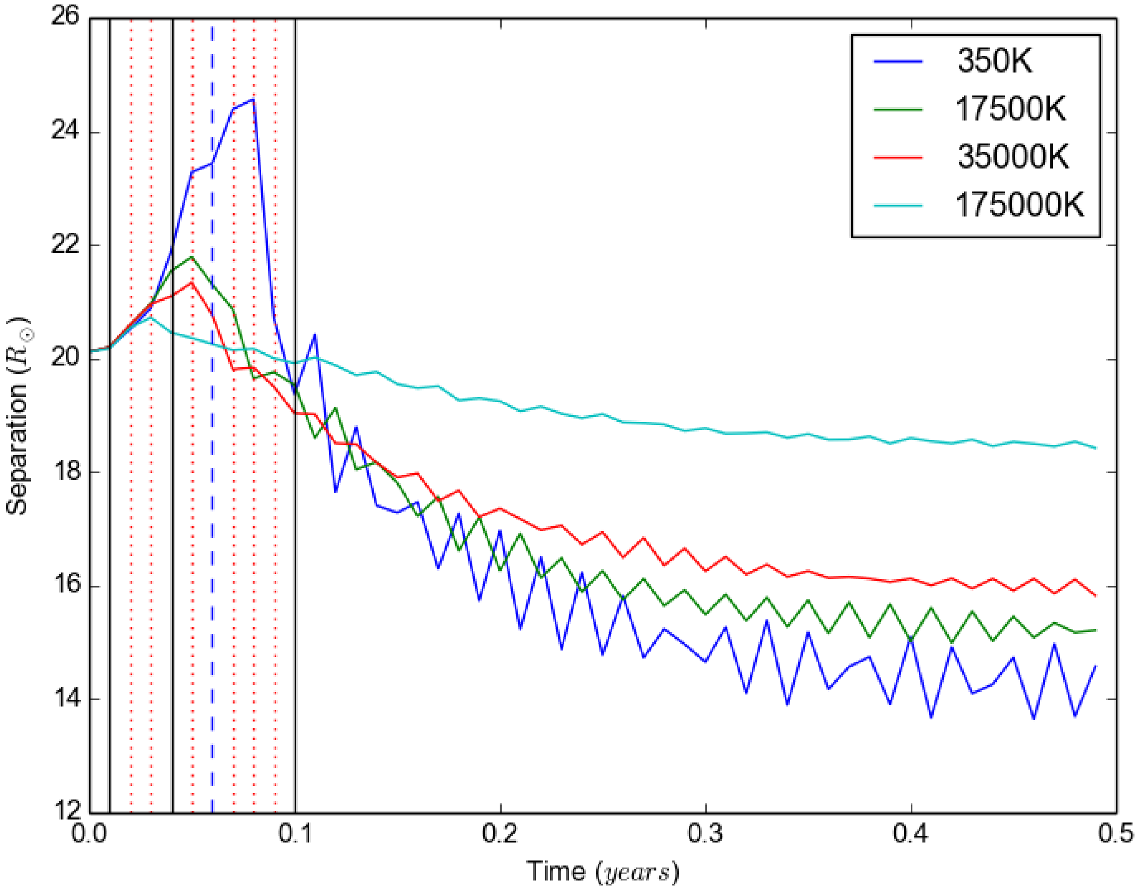
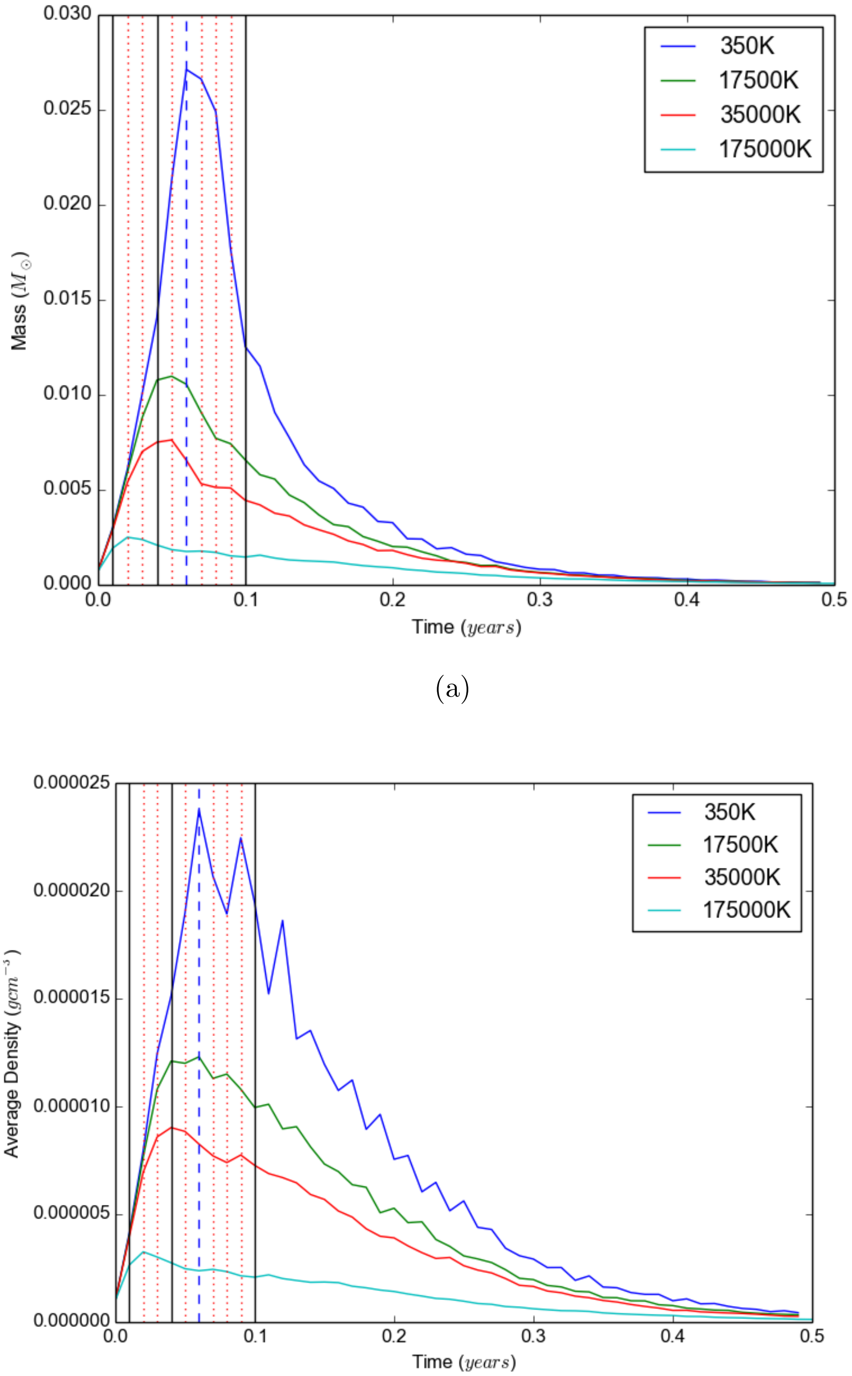


Figure 5.6: Evolution of the orbital separation for simulations with different initial temperatures. (*Cool*: 350 K, *Hot1*: 17 500 K, *Hot2*: 35 000 K, *Hot3*: 175 000 K).



(a)

(b)

Figure 5.7: Comparison (a) mass and (b) average density of the gas enclosed within the core and companions' orbits for *Cool* (350 K, dark blue line), *Hot1* (17500 K, green line), *Hot2* (35000 K, red line) and *Hot3* (175000 K, light blue line).

relative velocity of the core and companion and the gas (Ostriker, 1999). It is also possible that the orbital in-fall would depend on the density gradient, something we would like to study further (see Chapter 6). Our three hot simulations result in lower mass disks, compared to the *Cool* simulation. Although there is no reason to think that the initial force on the core and companion is different in the four simulations (densities and relative velocities are the same), the ambient parameters change rapidly. Below we investigate the effects of density and velocity on the separation evolution.

We first compare the mass enclosed within the orbit (see Figure 5.7a). The *Cool* simulation has core and companion with a comparatively smaller separation and hence a smaller volume within the orbit. Despite this, more mass is enclosed in the orbit. This is probably expected because the mass of the fall-back disk was the largest in the *Cool* simulation, so there is more mass that might provide greater drag due to greater ambient density. The average density enclosed within the orbit for the four simulations is shown in Figure 5.7b, once again demonstrating a trend of higher values for the *Cool* simulations. In the *Cool* simulation we see an oscillation in the average density within the orbit over time. This is because of the orbital eccentricity that develops preferentially in the cooler simulations and is due to the density gradient: as the volume enclosed increases when the core and companion are near apastron, the amount of extra mass enclosed does not increase as much and therefore the average density within the orbit decreases. In conclusion, although all simulations show a decrease in separation due to fall back of mass, we conclude that the higher the mass of the fall-back disk, the greater the in-spiral.

All simulations also show an initial increase in separation with the cooler one showing the largest increase. For the colder simulations this was interpreted in Chapter 4 as due to the system acquiring an eccentricity. The hot simulations display a smaller initial separation increase, the hotter they are. We conjecture that the acquisition of an early eccentricity may be due to the fall-back disk mass. The disk mass distribution is not symmetric so an eccentricity is expected. Such eccentricity may therefore be more pronounced for more massive disks. This is supported by Figure 5.8, where we show that the hotter simulation *Hot2* develops a far smaller eccentricity than *Cool* (see Figure 4.6). In all cases the orbit seems to circularise with time.

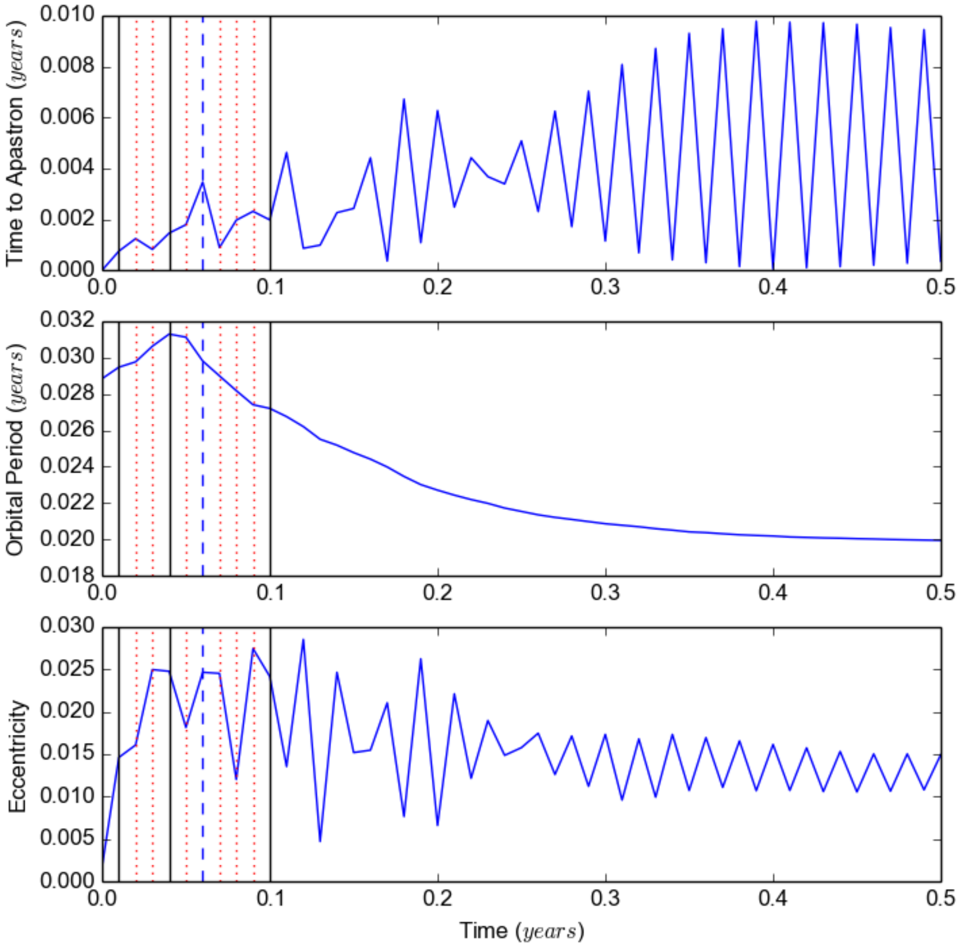
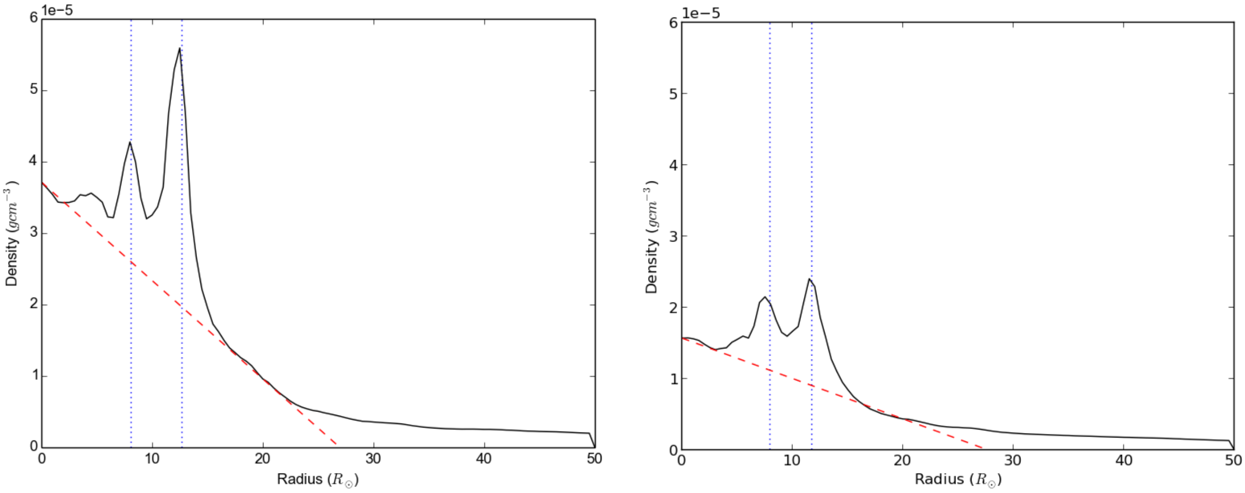
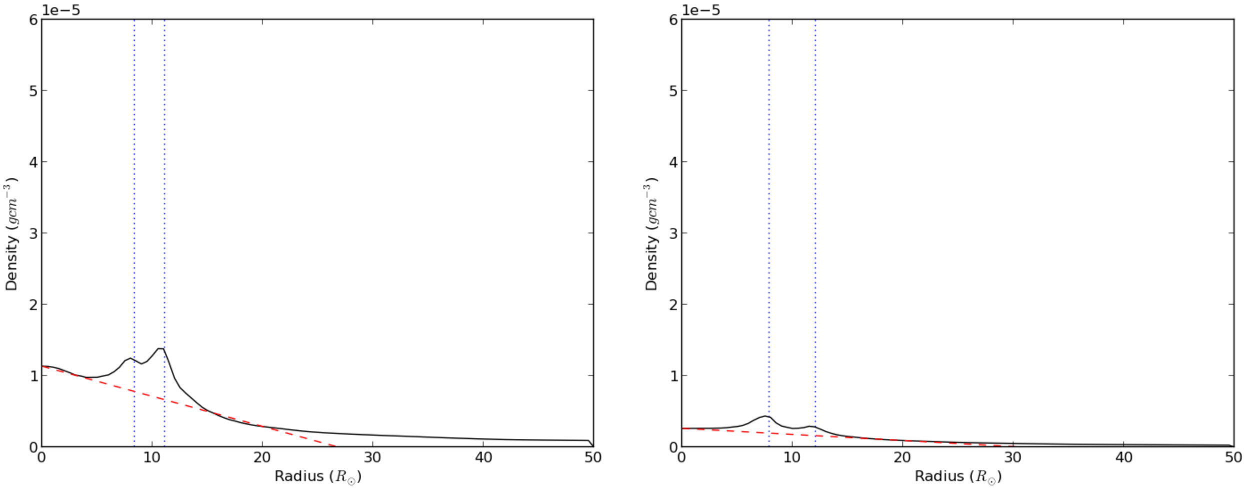


Figure 5.8: Top panel: time to apastron; middle panel: orbital period evolution; bottom panel: eccentricity evolution for the *HighRes* simulation, which is the high resolution counterpart of the *Hot2* simulation.

We conjecture that orbital in-spiral is also promoted by the presence of a density gradient. We think that in the presence of a density gradient the gravitational drag vector will point slightly towards the centre, where there is more mass (i.e., higher density). We attempt to show this in Figures 5.9 and 5.10. In Figure 5.9 density profiles were created for each of the production simulations at time 0.09 years. This time was selected because looking at Figure 5.6, the simulations appear to have the steepest in-fall around this time. We see that fitted density gradient becomes shallower, dropping by two orders of magnitude as we move to hotter simulations, which experienced less in-fall. The density profiles were created by binning the averaged density up to a radius of 50 R$_\odot$. The gradients of the profile were determined by taking the central density and the binned density at 20 R$_\odot$, and fitted the line. The gradient of this line is taken to be the gradient of the density profile.



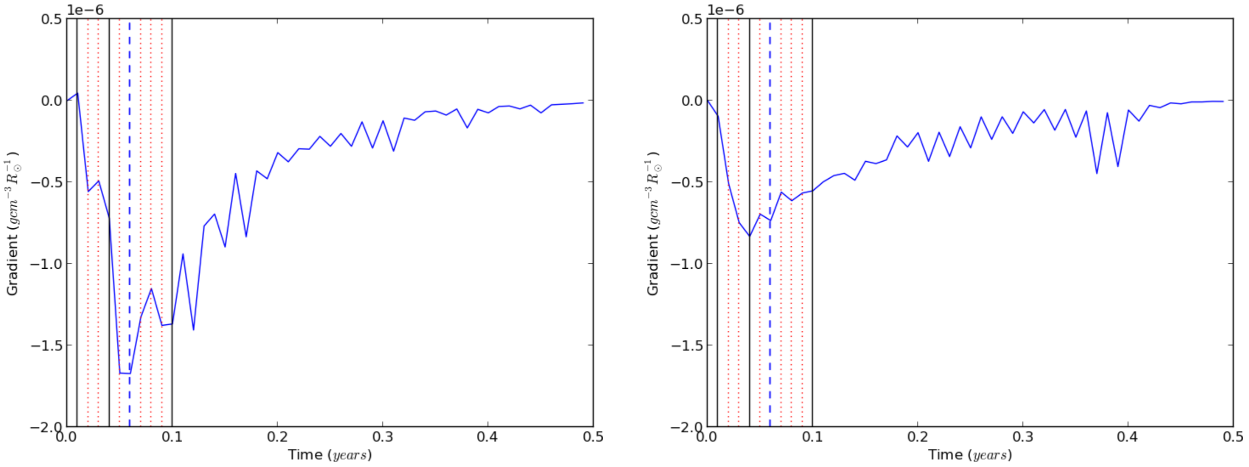
(a) *Cool* (b) *Hot1*



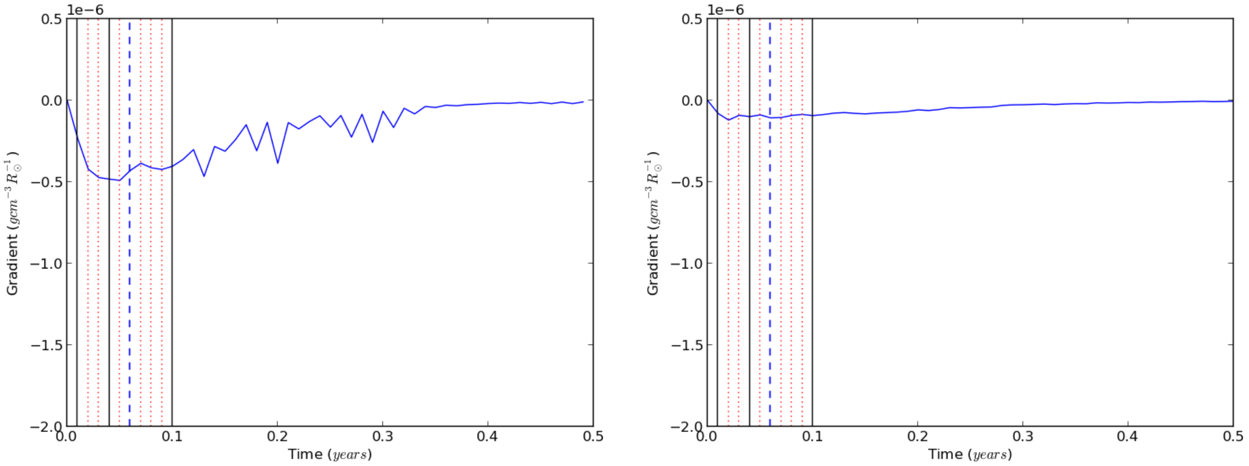
(c) *Hot2* (d) *Hot3*

Figure 5.9: Density profiles for a sphere with radius of 50 R$_\odot$ centred on the center of computational domain, taken at time 0.09 years. The blue dotted lines indicate the position of the core and companion. The red dashed line is fitted to the profile with gradient (a) -1.4×10^{-6}, (b) -5.5×10^{-7}, (c) -3.9×10^{-7} and (d) -8.4×10^{-8} g cm^{-3} R$_\odot^{-1}$

Time series of the gradients of the red dashed lines in Figure 5.9 are shown in Figure 5.10. Each simulation reaches the steepest density gradient $\sim$ 0.01 years before the peak separation is reached. When the gradient is at its steepest the gravitational drag force becomes the greatest, and counters the out-spiral of the core and companion due to the eccentric orbit. This leads to an orbital decay. We see that the density gradient is consistently steeper as



(a) *Cool* (b) *Hot1*



(c) *Hot2* (d) *Hot3*

Figure 5.10: Density gradient time series for our production simulations. The gradients are those of the red dotted line, as described in Figure 5.9. The time of the minimum gradient is (a) 0.06, (b) 0.04, (c) 0.05 and (d) 0.02 years

the temperature decreases, and the density gradient vanishes as the simulations stabilise.

Since we concluded that the orbital separation increase is due to an acquired eccentricity and the orbital decay is due to the presence of a density gradient across the core and companion, we decided to measure the orbital reduction starting from an initial separation of $20R_{\odot}$. Thus We find that in the *Cool*, *Hot1*, *Hot2* and *Hot3* simulations, the reduction of the orbital separation is 30%, 23%, 20% and 8%, respectively.

5.2.2 Unbound Mass

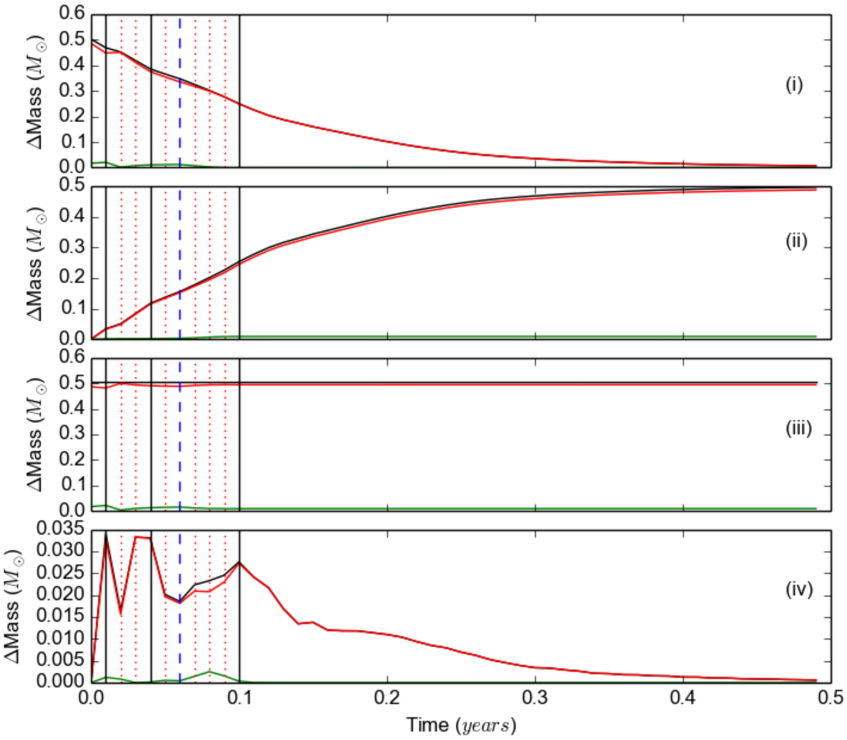
The analysis to determine the fraction of unbound gas was carried out in the same fashion as described in Chapter 4. To determine whether gas in a cell was unbound we calculated the total energy of that cell. To determine whether gas that left the computational domain was unbound, we determined the fraction of unbound gas at the computational domain boundary for every code output and multiplied it by the mass that left the domain between outputs.

Total mass time series similar to Figure 4.7 were created for the hot simulations, shown in Figure 5.11(a), (b) and (c). Looking at panel (i) in all of these figures, it can be seen that the hotter the initial temperature the more mass leaves the box. This is expected as more gas expands away out of the box in the initial time steps in the hotter simulations. In the *Hot3* simulation much of the gas is initially unbound due to the high temperature. In the colder simulations, on the other hand, most of the gas begins in a bound state. We remind the reader that the initial setup is an artificial configuration aiming to create a disk with characteristics consistent with what we expect based on the final conditions of Passy et al. (2012). We are therefore not interested in whether the gas in the box at time zero is bound or not. We just care about the amount of gas that becomes unbound as a result of the renewed interaction between core and companion and gas *caused* by the fall-back event.

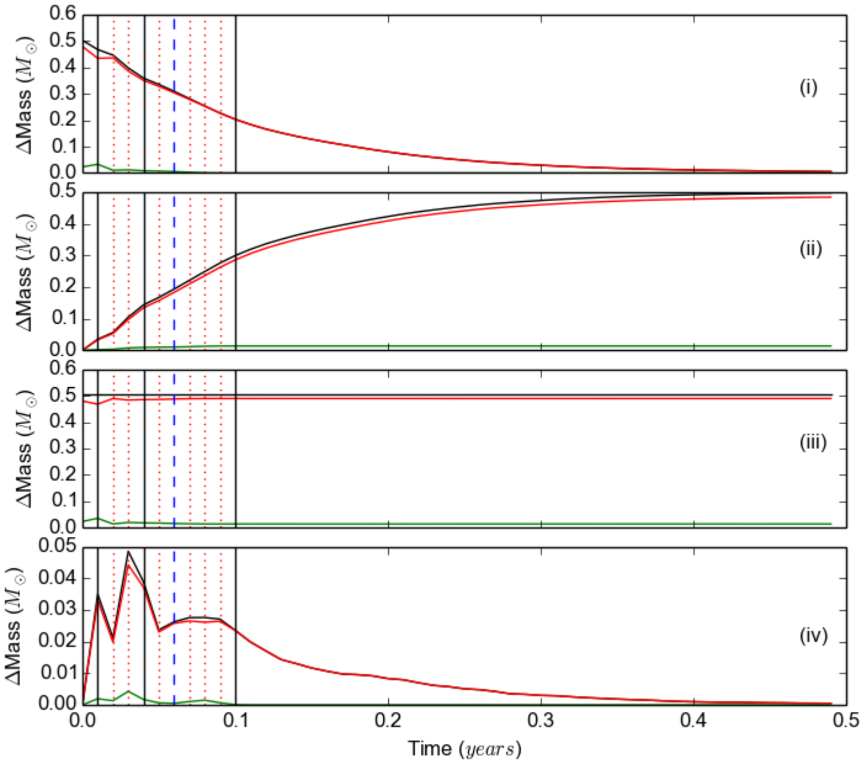
For the *Cool*, *Hot1* and *Hot2* simulations, minor unbinding events occur after the infalling gas starts to interact with the core and companion. To show these events more clearly, the percentage of unbound gas within the box is plotted in Figure 5.12. From this figure we can see that the *Cool* and *Hot1* simulations appear to have an unbinding event at 0.06 years. The *Hot2* simulation does not seem to have a clear unbinding event, and in the

Hot3 simulation the gas expands away before any significant unbinding can occur.

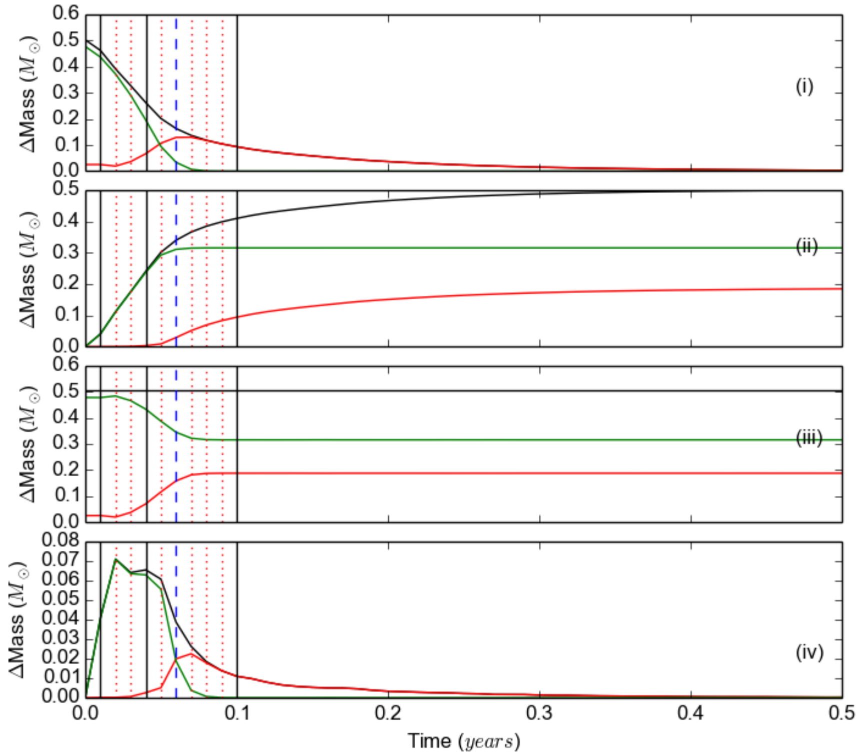
Judging by the total unbound mass at the end of the simulation and the mass of the fall-back disk, these events appear to have very poor unbinding efficiencies. To calculate unbinding efficiencies we cannot simply subtract the initial unbound mass within the box from the final unbound mass as we would obtain a negative number for all the hot simulations. This is because adiabatic cooling decreases the internal energy of the gas converting some of the unbound gas into bound gas. However, we note that in the hot simulations more of the initially unbound mass resides within the cone of super-keplerian gas, and therefore leaves the box almost immediately. Hence, a more accurate measurement of the unbinding efficiency would be to only account for the gas that falls back. Looking at Figure 5.12, the *Cool*, *Hot1* and *Hot2* simulations reach a minimum unbound mass in box at 0.02 years (the first red dashed line). This point is interpreted as the moment when most of the initially unbound gas has left the box and the fall-back disk is beginning to interact with the core and companion.



(a) For *Hot1*.



(b) For *Hot2*.



(c) For *Hot3*.

Figure 5.11: (i) Mass within the box, (ii) cumulative mass out of the box, (iii) total mass in simulation, (iv) mass loss per data dump. The black line shows the total mass. The red and green lines indicate the bound and unbound mass, respectively. The vertical lines are described in the caption of Figure 4.7.

With this in mind, we decided to start the unbound mass analysis after 0.02 years. We take the overall gas that becomes unbound due to the fall-back and the final amount of unbound mass in the simulation (i.e., the value of the green line in (iii) in Figure 4.7 and Figure 5.11(a), (b) and (c)). The mass of the fall-back disk is determined by measuring the mass that falls within 30% of the orbital plain as shown in Figures 4.2, Figure 5.3b, Figure 5.4b and Figure 5.5b. We calculated the unbinding efficiency by dividing the total unbound mass (inside and outside of the box, panels (iii) in Figures 4.7, Figure 5.11(a) and (b)) by the mass of the fall-back disk. The efficiencies thus calculated are 5% for *Cool*, 3% for *Hot1* and 5% for *Hot2*. This calculation cannot be carried out for the *Hot3* simulation.

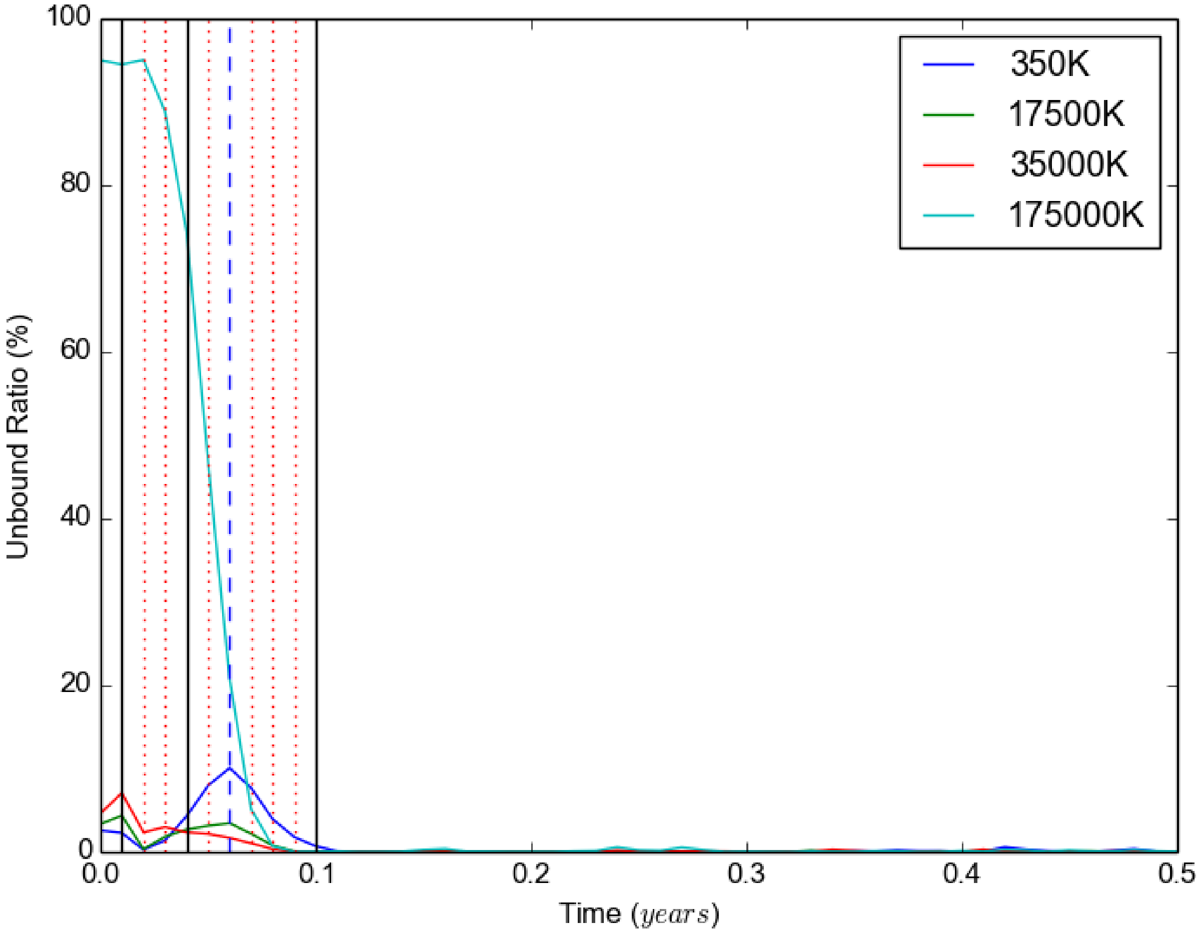


Figure 5.12: Unbound gas as percentage of the total gas mass in the box at current time for the *Cool* (350 K, dark blue line), *Hot1* (17500 K, green line), *Hot2* (35000 K, red line) and *Hot3* (175000 K, light blue line) simulations.

In that case the amount of initial unbound mass is so large that any unbinding at the hand of the orbiting core and companion is confused. Despite the difficulties of estimating the unbinding action of the core and companion for the hotter simulations, the conclusion is that not much mass is unbound before the core and companion in-spiral levels off again, signalling that the interaction is effectively over. If this persists in further fall-back episodes, the inevitable conclusion is that the core and companion would merge before the entire envelope is unbound. We discuss this further in Chapter 6, where we also propose further tests.

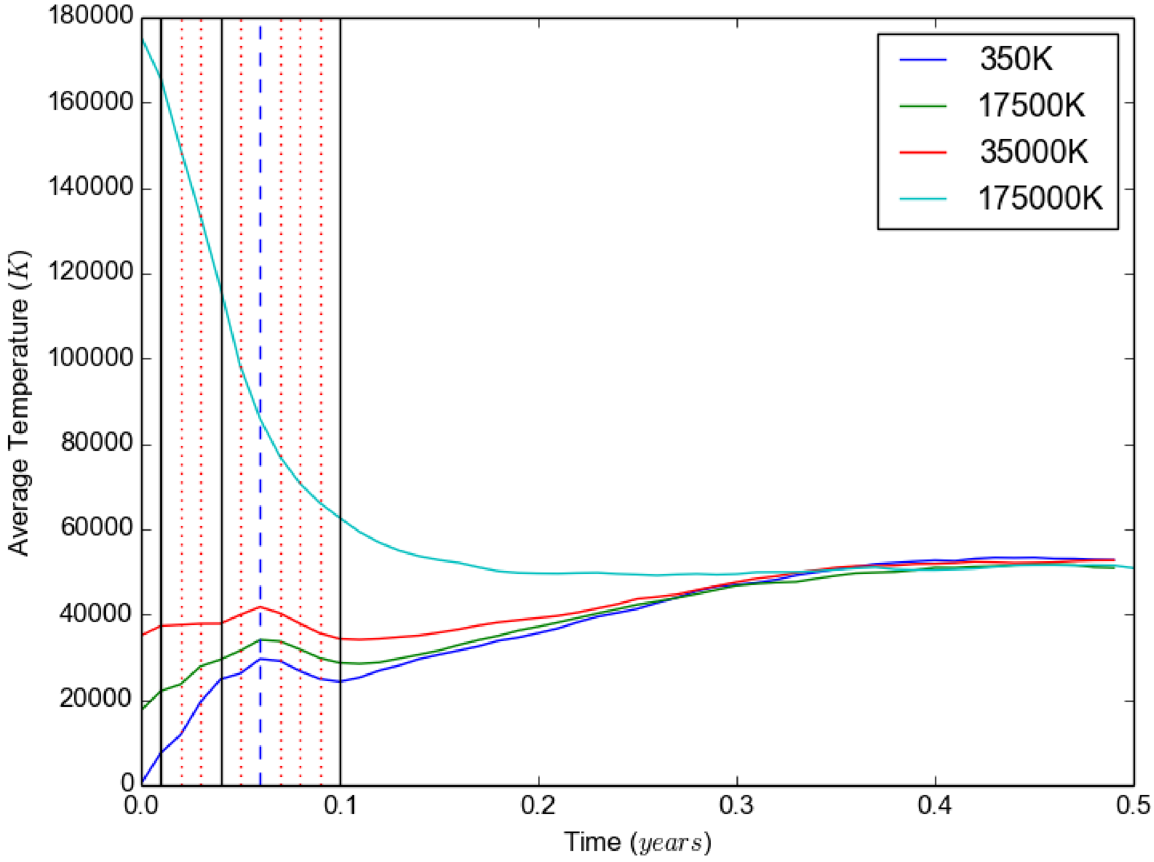


Figure 5.13: Evolution of the average temperature over the entire grid for different initial temperature simulations. Vertical lines are described in the caption of Figure 4.7.

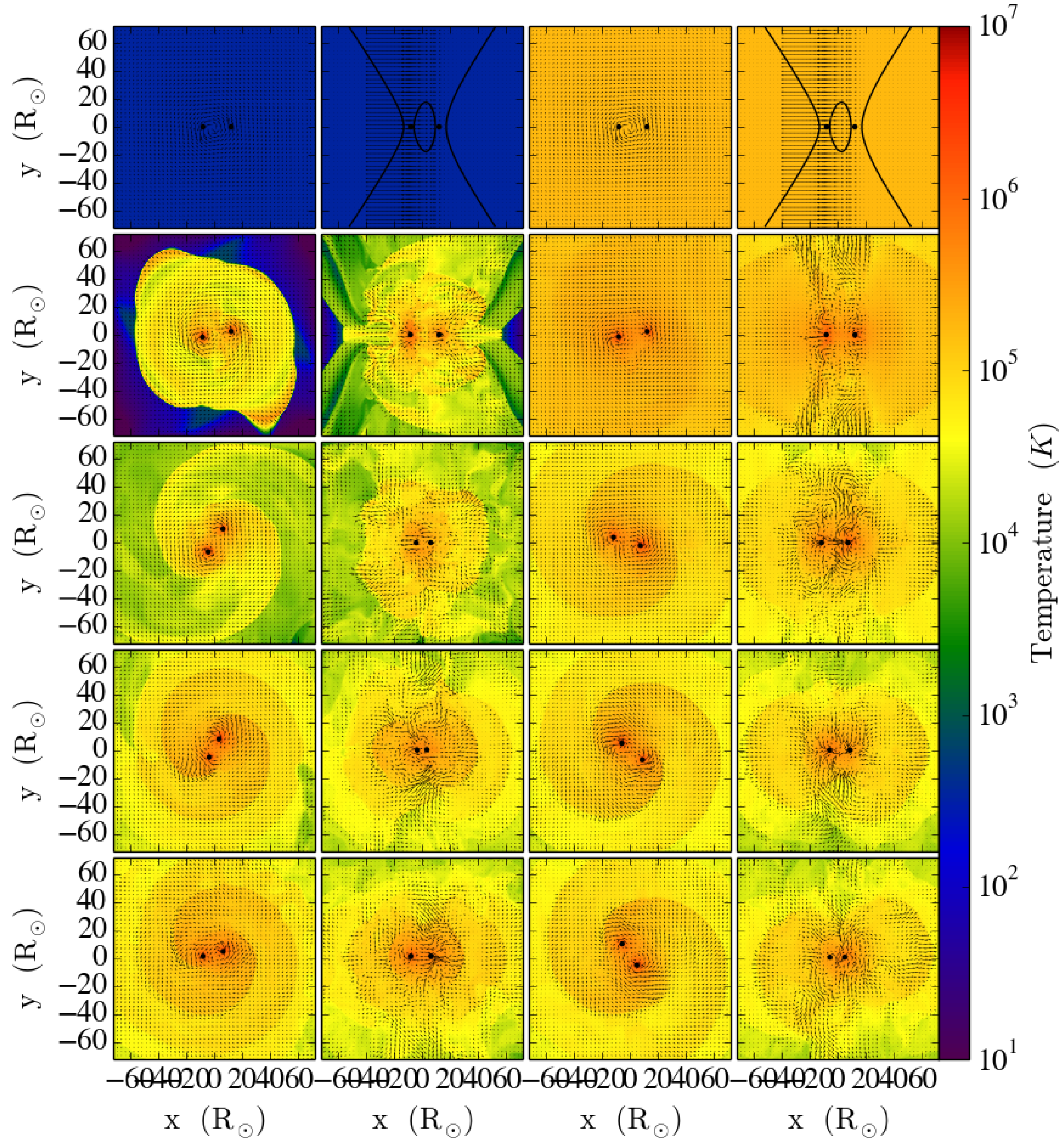


Figure 5.14: Temperature slices in the orbital (columns 1 and 3) and perpendicular (columns 2 and 4) plains for the *Cool* (columns 1 and 2) and *Hot3* (columns 3 and 4) simulations. From the top row to the bottom row, the times correspond to 0.0, 0.03, 0.1, 0.3 and 0.5 years.

5.2.3 Temperature Evolution

In this chapter we aim to compare the impact of varying the initial temperature on the fall-back disk effectiveness. Irrespective of the initial gas temperature of each simulation, three simulations heat up (*Cool*, *Hot1* and *Hot2*) and one cools down (*Hot3*) to approximately 50 000 K. We show this behaviour by plotting the average temperature in the box as a function of time for each simulation in Figure 5.13. To show this behaviour more clearly, we also show temperature slices for the *Cool* and *Hot3* simulations in Figure 5.14. In this figure we can see that, by the end of the simulations, both simulations become virtually indistinguishable in terms of temperature. However, the evolution to this temperature is very different. From the first few snapshots, we see that the *Cool* simulation heats up significantly due to shock heating, with the gas around the core and companion jumping very quickly from a few hundred degrees to around 50 000 K. However, in the *Hot3* simulation the gas quickly expands away and cools down to 50 000 K. The final temperature converged towards the system's virial temperature.

5.3 Resolution Test

Ideally throughout this thesis resolution tests should have been conducted on all production simulations. We rely primarily on the resolution tests of Passy et al. (2012). However, here we present one resolution test for the *Hot2* simulation.

All simulations were conducted on a grid with 256^3 cells, arranged in a cube 143 $R_\odot$ on a side. Therefore, each cell had a length of approximately 0.5 $R_\odot$. With this resolution test, the *Hot2* simulation was run again on a grid of 384^3 cells. We call this simulation *HighRes* on Table 3.1.

A comparison of the separation evolution for the two resolutions is shown in Figure 5.15. Both simulations end at approximately 16 $R_\odot$, but it would be important to determine whether this value actually converges by carrying out a third test with a 512^3 grid. Judging by these two simulations the agreement is relatively good. The separation increased less for *HighRes* in the early part of the simulation. The lower resolution of the *Hot2* simulation

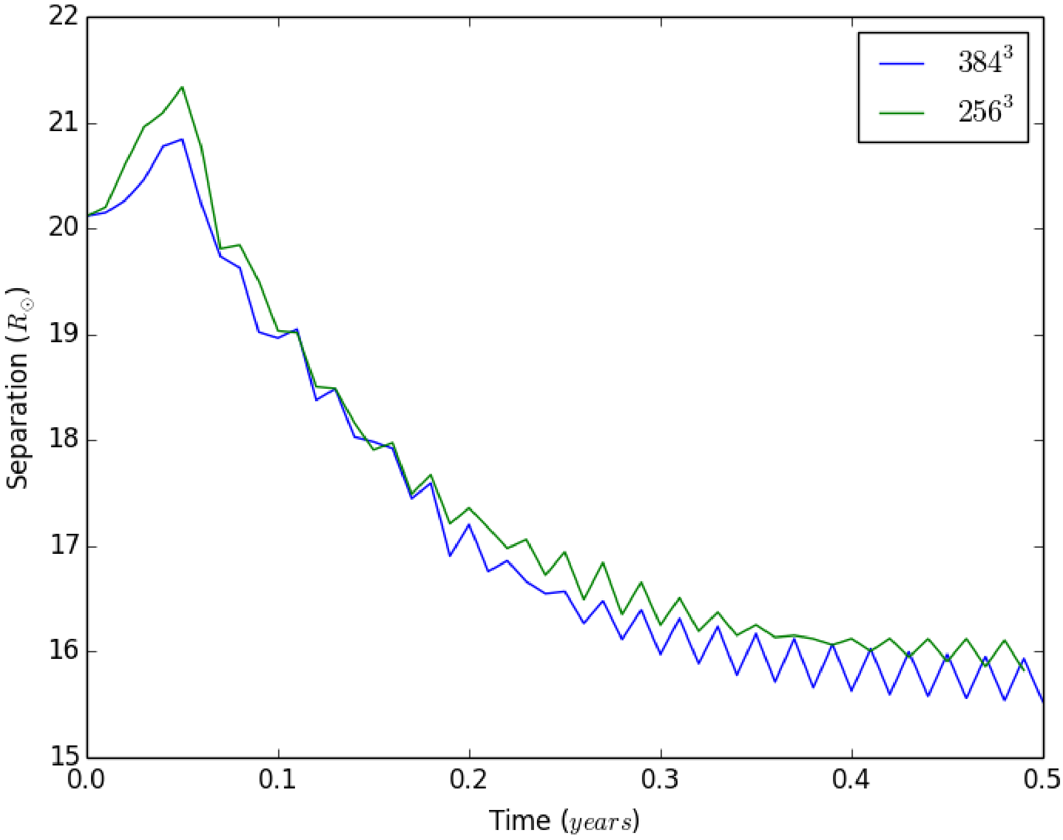


Figure 5.15: Separation time series for the *Hot2* simulation (resolution: 256^3 cells) and its high resolution counterpart, *HighRes* (resolution: 384^3 cells).

probably amplified the effect of the sudden increase in eccentricity due to the fall-back. The other minor difference is the final eccentricity, which is three times smaller ($e = 0.013$) for *HighRes* than for *Hot2* ($e = 0.05$, see Figure 5.8 and compare it with Figure 4.6). On the other hand both these values are quite low and their difference is not of much consequence.

The mass evolution of the two resolutions is also very similar. The mass of the fall-back disk for *HighRes* is also 0.28 $M_\odot$ and there is very little difference in the amount of unbinding in the the box. The mass in and out of the box varies slightly, but by very small amounts usually smaller than 0.007 $M_\odot$, which corresponds to ∼0.4% (see Figure 5.16). The *HighRes* simulation appears to retain some more mass within the box after 0.1 years, ∼0.4%. By the end of both simulations, the *HighRes* simulation unbound about 0.04% more mass than its lower resolution counterpart. This is very small and does not affect the efficiency of the unbinding, which was also 5% for the *HighRes* simulation.

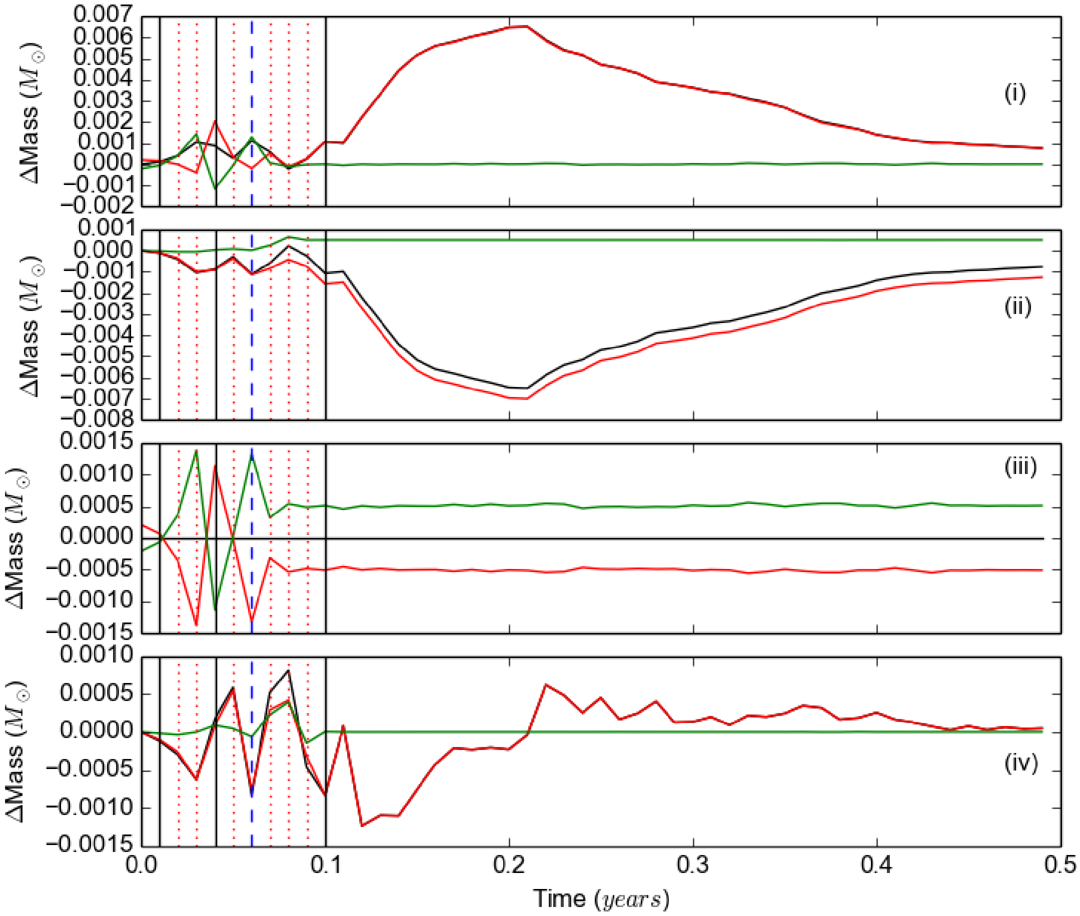


Figure 5.16: This figure is equivalent to Figure 5.11b. However, for each panel we show the difference in mass between the *Hot2* and *HighRes* simulations, where *HighRes* is the higher resolution version of *Hot2*. As with Figure 5.11b, (i) corresponds to the mass inside the box, (ii) to the mass outside the box, (iii) to the total mass in the simulation, and (iv) to the mass leaving the box per output. Black lines show the total mass, while red and green lines show the bound and unbound mass, respectively. Vertical lines are described in the caption of Figure 4.7.

Overall, although a more rigorous resolution convergence test would be desirable, the lower resolution simulation *Hot2* produced similar results to the *HighRes* simulation. Assuming the other simulations to have similar behaviour, this test shows that there is no significant variation in terms of unbinding efficiency and in-spiralling of the system with increased resolution.

5.4 Conclusions

In this chapter we carried out three “hot” simulations because of the realisation that the fall-back gas may be hotter than the temperature we had used initially (see simulation *Cool* in Chapter 4) and because the gas that is still in contact with the core and companion at the end of the simulation of Passy et al. (2012) had a temperature of 50 000 K. We were also suspicious of the high Mach numbers of the core and companion at the beginning of the *Cool* simulation, which resulted in high levels of shock heating. Such strong shocks can be problematic of their own right in hydrodynamic calculations.

Increasing the temperature resulted in different amounts of fall-back gas, which altered the parameters of the fall-back disk. This may have contributed more than anything else to the different results we have observed in the four simulations. Despite these differences, the mass unbound by the interaction between the fall-back gas and the core and companion remains low, while the amount of in-spiral does vary somewhat with the hottest simulation exhibiting the least amount of in-spiral.

5.5 Epilogue

Here ends the analysis of our four production simulations. In Chapter 6 we will discuss further how the reduction of the separation and the unbinding of mass work together towards a successful common envelope event, where mass is unbound and a close binary survives. At that time we will also discuss how to fine tune the simulations to gain the most insight out of the current set up.

6

Conclusion

After running a series of simulations to produce a fall-back disk, we summarise our results, critique our methods and generally comment on the efficiency of a fall-back event for unbinding further common envelope gas and decrease further the orbital separation.

In this thesis we aimed to address a major issue in our understanding of the common envelope binary interaction phase. Previous studies could not simulate a common envelope phase that is capable of unbinding the RGB or AGB stars' envelopes nor did they result in orbital separations comparable to those observed in post common envelope binaries. In all simulations, most of the envelope remains bound to the system although it is lifted to large distances from the binary, so we hypothesised that this bound gas returns to the system and provides a new opportunity to unbind more gas and reduce further the orbital separation of the core and companion.

In this work we have attempted to simulate such a fall back of bound gas. We based our simulations' initial conditions on the final conditions of the SPH2 simulation of Passy et al. (2012). Of all the simulations we have run (Table 3.1), four are particularly meaningful. We have referred to them as *Cool*, *Hot1*, *Hot2*, and *Hot3* and we have discussed them in Chapters 4 and 5.

6.1 The Orbital Separation of the Core and Companion During the Fall Back Event

In all our simulations there was a decrease in the orbital separation. The amount of in-spiral varied for each case, but the cooler the initial temperature, the greater the in-fall. We hypothesize this is due to the cooler simulations having a larger fall-back disk mass, hence more gas was available to interact with the core and companion rather than the temperature itself.[1] Ideally, we should have compensated for the lower fall-back mass of the higher initial temperature simulations by adjusting the initial gas velocity, so as to obtain disks with approximately the same mass in all simulations. Doing so would have given a better understanding of the influence of temperature of the fall-back gas on the amount of in-spiral.

In none of our simulations do we observe sufficient in-spiral to bring the core and companion to within the observed separations, but if a number of fall-back events occur after a common envelope phase the separation would be reduced further. Below we calculate the number of fall-back events necessary for the orbital separation of the core and companion to decrease to within the observed range of $\lesssim 4$ $R_\odot$. This number is based off the work of Zorotovic et al. (2010) and De Marco et al. (2011) who found, from a survey of 61 post common envelope binaries, that 87% of them had a separation of $\lesssim 4$ $R_\odot$. In order to calculate the number of necessary fall-backs, we noticed that the relative orbital reduction appears to be larger, the greater the mass of the fall-back disk. This is shown in Figure 6.1, where based on our simulations the relationship between in-fall efficiency and fall-back mass is demonstrated.

---

[1] while applying corrections we have conducted further simulations of fall back events and found this interpretation to be correct.

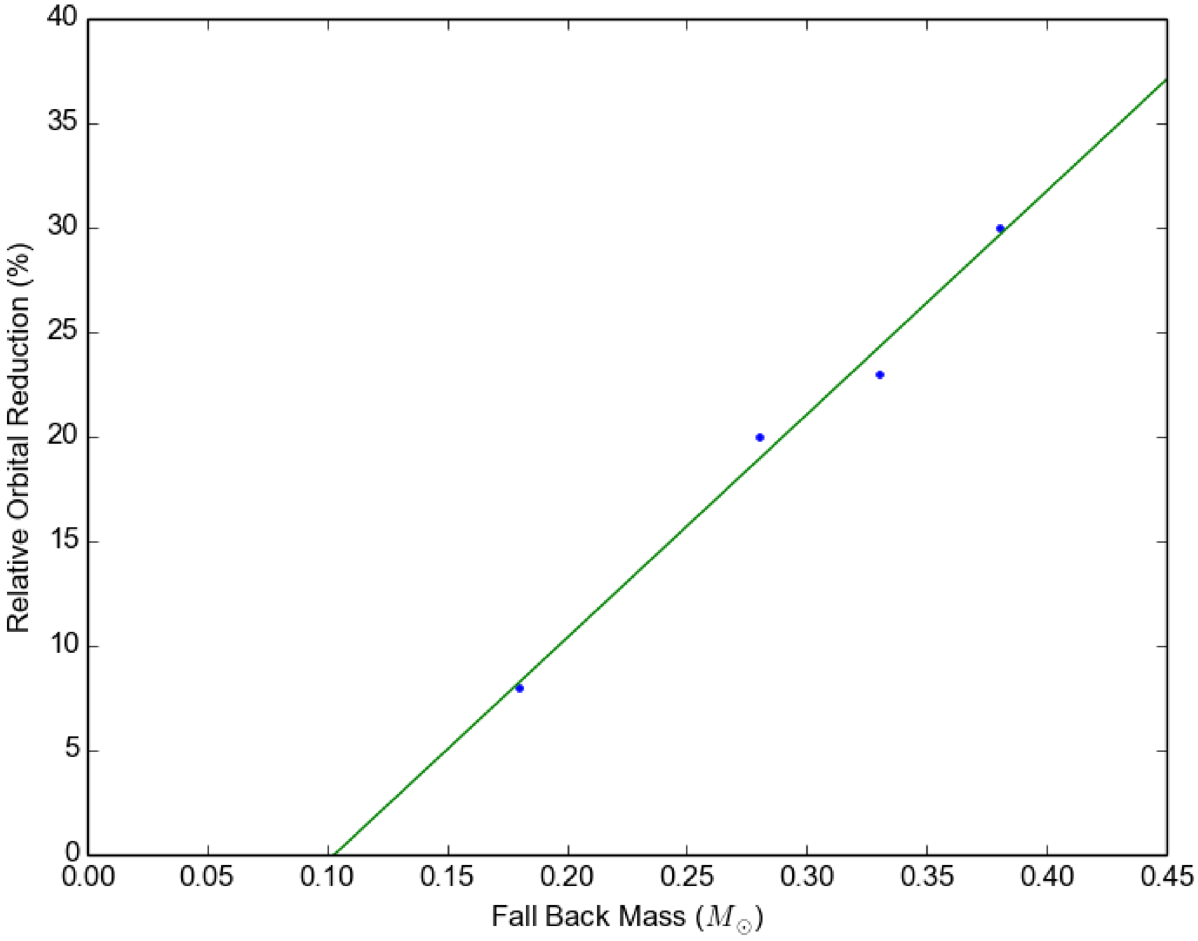


Figure 6.1: Relative separation reduction for different fall-back disk masses, based on our four main simulations. This fits an approximate linear relationship: $y = 107x - 11$.

However, to account for the a variable mass of subsequent fall-back events, we would need to know the fall-back mass for every successive fall-back event. Determining the fall-back mass of successive fall-back events requires a calculation of the unbound mass, something that we tackle in Section 6.2. For now, we attempt to estimate a lower limit on the number of necessary fall-back events by assuming that all events are equally efficient in reducing the orbit, as opposed to a decreasing efficiency. With this assumption the orbital separation after n fall-back events is given by:

$$R_n = R_0(1 - e_{in})^n, \tag{6.1}$$

where R_0 is the initial orbital separation, which in our case is 20 $R_\odot$, e_{in} is the efficiency with which the fall-back makes the core and companion in-spiral, or in other words, the amount of in-spiral divided by the separation at the beginning of each event, n is the number of fall-back events and R_n is the orbital separation after n fall-back events. Rearranging this for n, we can calculate the lower limit for the number of fall-back events using:

$$n = \frac{\log \frac{R_n}{R_0}}{\log(1 - e_{in})}. \tag{6.2}$$

Using a target final separation of 4 $R_\odot$, the ratio $\frac{R_n}{R_0} = 0.2$. Now, using our simulations we can calculate an approximate number of fall-back events. For the *Cool* simulation which has an orbital reduction of ~30%, at least 5 fall-backs are needed. For *Hot1*, which has an orbital reduction of ~22%, then at least 7 fall-backs are needed. For *Hot2*, which has an orbital reduction of ~20%, then at least 8 fall-backs are needed. Finally, although we could not calculate the unbinding efficiency for *Hot3* the orbital reduction of *Hot3* is ~7%, then we find that at least 23 fall-backs are needed to bring the orbital separation to 4 $R_\odot$.

As we have determined in Chapter 4, the fall-back time is short, of the order of weeks to months. For example, using the velocity of the bound gas leaving the computational domain, the return time to the center of the domain is estimated (see Figure 4.9). For the *Cool* simulation, at the time that mass from the unbinding event appears to leave (approximately 0.08 yrs, or the 5th red dashed reference line, in Figure 4.7), the gas has a fall-back time ranging from weeks to years, but the denser gas had a fall-back time of approximately 7 weeks. Common envelope ejection events are dynamical, therefore of the order of months to years for RGB or AGB stars. Adding the interaction time of a few months, say 20 weeks, 4 events will take place in approximately 2 years. From the point of view of orbital separation reduction, all our simulations show great promise that fall-back events can help significantly decrease the orbital separation in a short time.

6.2 Unbinding of Gas

The simulations have successfully produced some significant in-spiral. However, none of them resulted in a great deal of unbound gas. As mentioned in Chapter 5, the unbinding efficiencies of all simulations were of the order of a few percent, with *Cool* unbinding approximately 5% of the fall-back mass.

Each unbinding event may have varying efficiency, but based on our simulations there does not seem to be any clear trend between any of the variables and unbinding efficiency, with all simulations unbinding a small fraction of the fall-back envelope. However, the closer the core and companion get, the more the orbital energy that will be released. This is because the potential of the core and companion grows steeper the closer they are to each other. Therefore it may be argued that successive fall-backs may result in a higher unbinding efficiency.

As in the previous section, below we make the assumption that the unbinding efficiency is constant for each fall-back. Then the fall-back mass of event number n is given by:

$$M_n = M_0(1 - e_{ub})^n, \tag{6.3}$$

where M_0 is the mass of the initial fall-back disk, e_{ub} is the unbinding efficiency of the event, n is the number of fall-back events, and M_n is the mass of the remaining bound gas after n fall-back events. Similarly to the previous section, we can calculate the number of fall-back events necessary to achieve a certain M_n value using:

$$n = \frac{\log \frac{M_n}{M_0}}{\log(1 - e_{ub})}. \tag{6.4}$$

To carry out the calculation we require a final bound mass value. We adopt $M_n = 0.1$ M$_\odot$, on the assumption that some final amount of envelope will remain in a circumbinary disk, and in Figure 6.1 we estimate fall-back mass of 0.1 M$_\odot$ will not produce any further orbital reduction for this system.

For the *Cool* simulation, the initial fall-back mass is 0.38 M$_\odot$ and had an unbinding efficiency of 5%, therefore this simulation requires an estimated 26 fall-back events. For *Hot1*, the initial fall-back mass is 0.33 M$_\odot$ and unbinding efficiency is 3%, which gives 39 events. Finally for *Hot2*, the initial fall-back mass is 0.28 $M_\odot$ and the unbinding efficiency of 5%, which required 20 fall-back events. Although the unbinding efficiency could change for further events, it does at present appear that the number of fall-backs required to eliminate the envelope would be larger than the number required to reduce the orbital separation to the required values. This would indicate that common envelope interactions result in mergers.

It does appear that for the fall-back mechanism to work we need fine tuning, resulting in approximately the same final separation no matter how massive the envelope is. If a mechanism that results in tuning is not implicit within the gravitational drag that drives the energy exchange, then we will have to appeal to additional physical mechanisms that can mine energy out of the stellar envelope. In Section 6.4 we discuss further these possibilities.

6.3 The Effect of Temperature and the Strength of the Interaction

In Chapter 5 we have investigated the effect of temperature on the fall-back event. The temperature of the fall-back gas matters because as Ostriker (1999) evaluated analytically, the maximum gravitational drag experienced by a perturber occurs at $\mathcal{M} \simeq 1$. As the temperature dictates the sound speed, it should affect the strength of the interaction between the core and companion, and the gas. The investigation was initiated when we realised that the stabilisation temperature of the simulation of Passy et al. (2012) was 50 000 K within 1 AU of the core and companion. This temperature is significantly hotter than the initial temperature of our *Cool* simulation (∼350 K), which showed strong shocks at the onset of the fall-back event. It was then considered that the gas from the common envelope event would not have enough time to cool significantly in the short return time scales as calculated by Tocknell et al. (2014) and hence would come back as hot as when it started expanding away. If so, it would avoid shock heating.

Guessing the temperature of the falling back envelope turned out to be difficult, as shown by our attempt in Chapter 5. This is because there are a number of factors to consider. These factors include whether the envelope gas is optically thin or thick, and the cooling time function. Though the gas near the core and companion was very hot at the end of the simulations of Passy et al. (2012), it is possible that the gas farther out would have cooled adiabatically, so there is probably a temperature gradient over the extended envelope. In our simulations we only considered the gas being all hot or all cold.

We calculated the initial sound speed of gas in the finishing conditions of Passy et al. (2012). Using an initial temperature of 50 000 K initial density of 5×10^{-7} g cm^{-3}, mean molecular mass of $\mu = 0.6$ and adiabatic index $\gamma = \frac{5}{3}$ as we treat the gas as adiabatic, the sound speed of the gas would be $c_s = 3.38 \times 10^6$ cm s^{-1}. With the initial velocity of the core and companion being 3.82×10^6 cm s^{-1} and 5.89×10^6 cm s^{-1}, respectively, the core and companion would begin supersonic. So even if we started at the temperature of 50 000 K, with either our density of 10^{-6} g cm^{-3}, or the finishing density of Passy et al. (2012) of 5×10^{-7} g cm^{-3}, the core and companion would be supersonic, and hence it appears some shock heating would occur anyway.

One peculiarity that was found is that all four production simulations reached the same final temperature of $\sim$50 000 K, which is the same as the stabilisation temperature of the simulation of Passy et al. (2012). It was also found that, the manipulation of the initial temperature had a large effect on the fall-back mass. By changing the initial temperature we aimed to investigate the changing gravitational drag on the system. We do find that the simulations that started more super-sonic showed a greater in-spiral, however it is not certain whether it was the greater fall-back mass or the higher temperature that were the main contributors to this. To better understand the relationship between temperature of the envelope gas and the fall-back event evolution, the fall-back mass should be better constrained by changing the initial velocity distribution to allow the same amount of mass to fall back into the system with varying temperature. Doing so would allow us to carry out a more thorough study on the strength of the gravitational drag in hydrodynamic computations. It would also give us the opportunity to determine whether the analytical study of Ostriker (1999) is representative of our simulations.

6.4 Conclusions and Future Work

The question we asked in this thesis was whether the return of bound gas from a common envelope event would provide another opportunity for further energy and angular momentum transfer from the core and companion to the gas, resulting in a smaller orbital separation and more unbound gas. Concerning the orbital separation, it does appear that fall-back events provide another opportunity for further orbital separation reduction, and the reductions we have witnessed in our fall-back simulations are significant and leading to the conclusion that a handful of events would reduce the separation to within observed values. However, only a small fraction of the bound gas becomes unbound in the fall-back process. Assuming that the unbinding efficiency remains the same, it would take a much larger number of fall-backs to unbind the entire envelope than to reduce the orbital separation to zero. This would imply that this common envelope event would result in a merger.

Why do our simulations fail to significantly unbind gas? As suggested in previous studies (Han et al. 1995; Ivanova et al. 2014), accounting for recombination of the gas may provide the energy to unbind most of the envelope. We should also consider that dust likely forms in the expanding common envelope as observed by Tylenda et al. (2011) and Nicholls et al. (2013) for the only witnessed common envelope event, V1309 Sco. Such dust can help transfer momentum from the radiation field to the envelope. We know, however, that the solution to the problem cannot be only an extra energy source, such as ionisation energy. Such extra energy may unbind the envelope but it will not necessarily reduce the orbital separation in anything it would widen it due to the high unbinding efficiency of the interaction. On the other hand a fall-back event, or multiple fall-back events, can reduce the orbital separation and unbind further envelope gas. However, if our results hold, then even in the presence of a fall-back event some extra energy source in addition to orbital energy would be needed.

We should repeat our hot simulations and modify the velocity distribution to obtain the same fall back mass in all simulations only so we would determine whether the temperature (and hence the Mach number of the core and companion) plays a role in the interaction. Modifying the velocity distribution may alter the angular momentum of the system, therefore we should approach this problem carefully.

With our hydrodynamic simulation code *Enzo* we had difficulties setting up a disk, and controlling the amount of mass falling onto the core and companion. As producing fall-back disks have proven to be difficult it might be desirable to simulate a fall-back disk with smoothed particle hydrodynamic code. This would provide us with an opportunity to determine whether our results for a fall-back event can be confirmed via other computational methods and whether our initial set up may have influences our calculations.

Further resolution tests would also be necessary to verify our results even if Passy et al. (2012) have carried out several convergence tests with parameters similar to ours. As it is, we have only carried out resolution tests for the hotter simulations. From these tests we have concluded that with a higher resolution the eccentricity that develops at the beginning of the *Cool* simulation might not be so severe. A further increase in resolution may decrease it further.

Concerning shocks in our simulations, a better understanding of gravitational drag would be desirable. This would be carried out by comparison with analytical values of Ostriker (1999). This will help better understand where the major contributions to gravitational drag come from in our simulation. Concerning drag force and particle dynamics, magnetic fields are likely very important in this process and should be considered in future work. We expect that kilo-Gauss magnetic fields would permeate the gas by the end of the common envelope in-spiral (Regos & Tout 1995; Tocknell et al. 2014). During the expansion the magnetic field strength would locally decrease, but likely increase again during the in-fall. This would make the gas more buoyant than simulated by us with a resulting different dynamic and drag force on the particles.

Simulating a second fall-back may be insightful. Simulating consecutive fall-back events may produce an understanding on how the in-fall and unbinding efficiency evolves with the system. This can be used to confirm whether the assumptions made in Sections 6.1 and 6.2 for estimating the number of necessary fall-back events were valid.

After carrying out a series of simulations of fall-back disks with grid code, we conclude that our hypothesis is partially confirmed. The results of our simulations found that a fall-back event does decrease separation of the core and companion. A single fall-back event is not capable of producing the separations observed in post common envelope binaries, but multiple fall-back events will likely continue to shrink the orbital separation to the observed values. Our results also show that this event does not significantly unbind more gas from the binary system. Other energy sources, such as that due to recombination, may still be required to further unbind the envelope. While the envelope energetics need to be studied further, extra energy alone would not solve the problems of large simulated final separations. One or more fall-back events are likely needed to bring the binary system closer and we therefore conclude that these events likely play an important role in the life of a common envelope interaction.

A
Appendix

In this appendix I report all the Python codes with `yt` libraries that I have written to analyse the outputs of the *Enzo* simulations.

A.1 Derived Fields

A.1.1 Temperature

Our custom temperature field. The temperature is determined using Equation 2.16.

```
def _Temperature(field, data):
    #This find the temperature, assuming the gas is ideal (ie gamma = 5./3.)
    #Thermal energy is given in erg/g,
    #Density is given in g/cm^3
    #Number density is given in cm^{-3}
    gamma = data.pf[‘Gamma"]
    Boltzmann_constant = 1.3806488e-16 #erg/K
    top= (gamma - 1.) * data["Density"] * data["ThermalEnergy"]
    bottom = data["NumberDensity"] * Boltzmann_constant
    temperature = top / bottom
    inf = np.all(np.isfinite(data["NumberDensity"]))
    return temperature

add_field("Temperature", function=_Temperature, units=r"K")
```

A.2 Analysis Codes

A.2.1 Fall-Back mass

```
#!/usr/bin/env python
# This script determines the mass of the fall back by simply
# monitoring the mass that falls within the central
# 30% (approximately 43Rsun) of the z-axis, about the orbital plain.

from yt.mods import *

# Import all the time steps for the series:
pf = load("Data Directory")
ts = TimeSeriesData.from_filenames("Data Directory")

# Define arrays:
time = []              # Time in years in the simulation
fall_back_mass = 0.0       # Peak mass within orbital plain region
Total_Mass_above = []        # Total mass in the regions
```

```
Total_Mass_plain = []
Total_Mass_below = []

# Find region bounds:
center = 0.3           # Thickness of central slab in grid units. Set to 30%
lower_upper_bound = 0.5 - center / 2.
upper_lower_bound = 0.5 + center / 2.
lower_center = lower_upper_bound / 2.
upper_center = 1. - (upper_lower_bound / 2.)

it = 0
for pf in ts:
    if it < 100:
        dd = pf.h.all_data()

        # Get current time:
        time.append(pf.current_time * pf["years"])

        # Define regions in orbital plain and above the plain:
        region_above = pf.h.region([0.5, 0.5, upper_center],
[0.0, 0.0, upper_lower_bound], [1.0, 1.0, 1.0])
        region_plain = pf.h.region([0.5, 0.5, 0.5],
[0.0, 0.0, lower_upper_bound], [1.0, 1.0, upper_lower_bound])
        region_below = pf.h.region([0.5, 0.5, lower_center],
[0.0, 0.0, 0.0], [1.0, 1.0, lower_upper_bound])

        # Find the total mass in these regions:
        total_mass_above_value = sum(region_above["CellMassMsun"])
        total_mass_plain_value = sum(region_plain["CellMassMsun"])
        total_mass_below_value = sum(region_below["CellMassMsun"])
        if total_mass_plain_value > peak:
            fall_back_mass = total_mass_plain_value

        # Append to mass arrays:
        Total_Mass_above.append(total_mass_above_value)
        Total_Mass_plain.append(total_mass_plain_value)
        Total_Mass_below.append(total_mass_below_value)
        it = it + 1

print("Fallback disc mass = %s Msun" % fall_back_mass)

# Create Plot
```

A.2.2 Center of Mass

```
#!/usr/bin/env python
# Calculates the position of the center of mass (CoM)
# Uses the total mass in the box

from yt.mods import *
import csv

# Import Data
ts = TimeSeriesData.from_filenames("Data Directory")
pf = load("Data Directory")
f = open("coms.csv","r+")
f.write("time, CoM:x, y, z\n")

# Define values:
dim = pf.domain_dimensions[0]
```

```
lu = pf["LengthUnits"]     # Length of the box in cm
gl = lu / dim              # Length of a cell in cm

#Find the mass of the particles:
dd = pf.h.all_data()
pm1 = dd["ParticleMass"]

for pf in ts:
    time_val = pf.current_time

    dd = pf.h.all_data()

    # Find the position of the particles:
    # Multiple the position but the length of the grid
    # to give the position in cm, not grid units
    pp1 = [dd["particle_position_x"][0] * lu,
dd["particle_position_y"][0] * lu, dd["particle_position_z"][0] * lu]
    pp2 = [dd["particle_position_x"][1] * lu,
dd["particle_position_y"][1] * lu, dd["particle_position_z"][1] * lu]

    # Set counters:
    # The CoM is found by (sum(mass * position)) / (total mass)
    # hence the _top variables are summing the (mass * position)
    TM = pm[0] + pm[1]
    x_top = pm[0] * pp1[0] + pm[1] * pp2[0]
    y_top = pm[0] * pp1[1] + pm[1] * pp2[1]
    z_top = pm[0] * pp1[2] + pm[1] * pp2[2]

    # Get the grids in the data output:
    g = pf.h.grids[0]
    # Now calculate CoM of gas:
    # This for-loop scrolls through all cells in the grid
    for x in range(dim):
        for y in range(dim):
            for z in range(dim):
                # We multiple the cell coordinates by the length of a cell in cm
                # We also add half a cell length so the position is
                # taken from the center of a cell.
                pos = [(x + 0.5) * gl, (y + 0.5) * gl, (z + 0.5) * gl]
                mass = g["CellMass"][x, y, z]

                # Sums total mass in the box:
                TM = TM + mass
                x_top = x_top + mass * pos[0]
                y_top = y_top + mass * pos[1]
                z_top = z_top + mass * pos[2]

    # Calculates the position of the CoM for total mass in the box:
    x_pos = (x_top / TM)
    y_pos = (y_top / TM)
    z_pos = (z_top / TM)
    com = [x_pos, y_pos, z_pos]

    f.write(str(time_val) + "," + str(com[0]) + "," + str(com[1]) +
"," + str(com[2]) + "\n")
```

A.2.3 Angular Momentum Conservation

```
#!/usr/bin/env python
```

```
# Calculate the z-components of angular momentum for the simulation,
# using Equation 4.3

from yt.mods import *
import matplotlib.pyplot as plt
import csv



def linear_momentum(mass, velocity):
    # Calculates linear momentum p=mv
    # Everything should be in cgs
    # mass in grams
    # velocity in cm/s
    # return in gcm/s
    px = mass * velocity[0]
    py = mass * velocity[1]
    pz = mass * velocity[2]
    p = [px, py, pz]
    return p

def rel_position(pos1, pos2):
    # Finds the relative position of the cell position to the CoM.
    # Output units are the same as input units.
    # This is just to work out the sign of the linear momentum.
    x = pos1[0] - pos2[0]
    y = pos1[1] - pos2[1]
    z = pos1[2] - pos2[2]
    rel = [x, y, z]
    return rel
```

```
def z_momentum(rel_position, momentum):
    # Works out the z-component of the angular momentum
    # Using the cross product of radius and linear momentum
    # i.e., L = r x p
    # Everything in cgs
    Lz = rel_position[0] * momentum[1] - rel_position[1] * momentum[0]
    return Lz

# Load data:
pf = load("Data Directory")
ts = TimeSeriesData.from_filenames("Data Directory")

#Define values:
dim = pf.domain_dimensions[0]
lu = pf["LengthUnits"]     # length units in cm
gl = lu / dim              # cm in grid length

#Define lists:
time = [] # Time in simulation
L_tot = [] # Total L_z momentum in box
L_p = [] # Total L_z momentum of particles
L_g = [] # Total L_z momentum of gas
coms = [] # Center of mass of system

header = 0
with open("coms.csv", "r+") as f:
    reader = csv.reader(f)
    for row in reader:
        if header != 0:
            x = float(row[1])
            y = float(row[2])
            z = float(row[3])
            com_val = [x,y,z]
            coms.append(com_val)
        else:
            header = 1

# Find the mass of the particles
dd = pf.h.all_data()
pm1 = dd["ParticleMass"]

it = 0
for pf in ts:
# Reset counters
    L_z = 0.
    Lp = 0.
    Lg = 0.
    time_val = pf.current_time
    time.append(time_val)

    dd = pf.h.all_data()

    # Find the position of the particles:
    pp1 = [dd["particle_position_x"][0] * lu,
dd["particle_position_y"][0] * lu, dd["particle_position_z"][0] * lu]
    pp2 = [dd["particle_position_x"][1] * lu,
dd["particle_position_y"][1] * lu, dd["particle_position_z"][1] * lu]

    # Find velocity of the particles:
    pv1 = [dd["particle_velocity_x"][0],
dd["particle_velocity_y"][0], dd["particle_position_z"][0]]
    pv2 = [dd["particle_velocity_x"][1],
dd["particle_velocity_y"][1], dd["particle_position_z"][1]]

    # Get the center of mass position
    com = coms[it]

    # Calculate angular momentum of the particles:
    p1 = linear_momentum(pm[0], pv1)
    p2 = linear_momentum(pm[1], pv2)
    rel1 = rel_position(pp1, com)
    rel2 = rel_position(pp2, com)
    L1 = z_momentum(rel1, p1)
    L2 = z_momentum(rel2, p2)
    Lp = L1 + L2
    L_p.append(Lp)

    # Get the grids in the data dump
    g = pf.h.grids[0]

    # Calculate angular momentum of the gas
    for x in range(dim):
        for y in range(dim):
            for z in range(dim):
                pos = [(x + 0.5) * gl, (y + 0.5) * gl, (z + 0.5) * gl]
                mass = g["CellMass"][x, y, z]
                velocity = [g["x-velocity"][x, y, z],
g["y-velocity"][x, y, z], g["z-velocity"][x, y, z]]
                p = linear_momentum(mass, velocity)
                rpos = rel_position(pos, com)
                L = z_momentum(rpos, p)
                # Adds up all the z-components
                # to find the total angular momentum of gas
                L_z = L_z + L

# Appends total angular momentum of the gas
    Lg = L_z
    L_g.append(Lg)

    # Calculates total angular momentum of system:
    # This is done by summing the particle and gas momentum
    Lt = Lg + Lp

    L_tot.append(Lt)
    it = it + 1

# Create plots
```

A.2.4 Energy Conservation and Determining Bound and Unbound Mass

```
#!/usr/bin/env python
# Calculates the total energy components.
# Also calculates bound and unbound gas within the box.

from yt.mods import *


# Define distance function
def distance(point1, point2):
```

```
# Uses Pythagorus theorem to calculate distance between two points
# Output units are the same as input units
    x_diff_sq = (point1[0] - point2[0]) ** 2.
    y_diff_sq = (point1[1] - point2[1]) ** 2.
    z_diff_sq = (point1[2] - point2[2]) ** 2.
    result = (((x_diff_sq) + (y_diff_sq) + (z_diff_sq)) ** (0.5))
    return result

# Define Gravitational Potential Energy function:
def Grav_Pot(Mass, cellMass, distance):
# Takes in the masses in g and the separation in cm
# gives energy in erg
    top = -6.67259e-8. * Mass * cellMass
    bottom = distance
    result = top / bottom
    return result

# Define Kinetic Energy function:
def Kinetic_Energy(mass, velocity):
# Takes in mass in g and the velocity in cm/s
# gives energy in ergs
    result = 0.5 * mass * (velocity ** 2.)
    return result

# Define lists:
time = [] # Time in simulation
bound = [] # Bound percentage of gas
unbound = [] # Unbound percentage of gas
KE_g = [] # Kinetic energy of gas
KE_p = [] # Kinetic energy of particle
```

```
KE = [] # Total kinetic energy in box
TE = [] # Thermal energy of gas (Particles carry no thermal energy)
GPE_g = [] # Gravitational potential of gas
GPE_pg = [] # Gravitational potential of particles on gas
GPE_pp = [] # Gravitational potential of particles
GPE = [] # Total gravitational potential energy
Total = [] # Total energy in box

# Import Data:
ts = TimeSeriesData.from_filenames("Data Directory")
pf = load("Data Directory")

# Find size of domain
dim = pf.domain_dimensions[0]

# Define values:
lu = init_pf["LengthUnits"]      # Length of box in cm
gl = lu / dim # Length of a cell in cm

# Find the mass of the particles
dd = pf.h.all_data()
pm = dd["ParticleMass"]

for pf in ts:
    time_val = pf.current_time
    time.append(time_val)

    dd = pf.h.all_data()

    # Find the position of the particles:
```

```
    pp1 = [dd["particle_position_x"][0] * lu,
dd["particle_position_y"][0] * lu, dd["particle_position_z"][0] * lu]
    pp2 = [dd["particle_position_x"][1] * lu,
dd["particle_position_y"][1] * lu, dd["particle_position_z"][1] * lu]

    # Find velocity of the particles:
    pv = (((dd["particle_velocity_x"]) ** 2. +
(dd["particle_velocity_y"]) ** 2. + (dd["particle_position_z"]) ** 2.) ** 0.5)

    # Find kinetic energy of particles:
    KEp1 = Kinetic_Energy(pm[0], pv[0])
    KEp2 = Kinetic_Energy(pm[1], pv[1])
    TotalKEp = KEp1 + KEp2
    KE_p.append(TotalKEp)

    # Find kinetic energy of gas
    # The Kinetic Energy field gives energy/volume,
    # hence why we multiply by the cell volume
    KEg = dd["KineticEnergy"] * dd["CellVolume"]
    TotalKEg = sum(KEg)
    KE_g.append(TotalKEg)

    # Calculate total kinetic energy in box:
    TotalKE = TotalKEp + TotalKEg
    KE.append(TotalKE)

    # Find gravitational potential energy of the particles on each other
    pp_sep = distance(pp1, pp2)
    TotalGPEpp = Grav_Pot(pm[0], pm[1], pp_sep)
    GPE_pp.append(TotalGPEpp)
```

```
# Find gravitational potential energy of the gas:
# Gravitational potential field returns the specific energy
GPEg = dd["Grav_Potential"] * dd["CellMass"]
#divide by two to avoid doubling up the potential measured between points.
TotalGPEg = sum(GPEg) / 2.0
GPE_g.append(TotalGPEg)

# Find thermal energy of gas:
 # Again, thermal energy field returns the specific energy.
TE_val = dd["ThermalEnergy"] * dd["CellMass"]
TotalTE = sum(TE_val)
TE.append(TotalTE)

# Get the grids in the data dump
g = pf.h.grids[0]

# Calculate GPE of particles on gas:
TotalGPEpg = 0
for x in range(dim):
    for y in range(dim):
        for z in range(dim):
            position = [(x+0.5)*gl,(y+0.5)*gl,(z+0.5)*gl]
            mass = g["CellMass"][x,y,z] #in grams

            #Calculate GPE in grid:
            radius1 = distance(position, pp1)
            radius2 = distance(position, pp2)
            GPE1 = Grav_Pot(pm[0], mass, radius1)
            GPE2 = Grav_Pot(pm[1], mass, radius2)
```

```
                GPEpg_val = GPE1 + GPE2
                TotalGPEpg = TotalGPEpg + GPEpg_val
                GPEpg.append(GPEpg_val)

    # Append total gravitational potential of the gas to list:
    GPE_pg.append(TotalGPEpg)

    # Calculate total gravitational potential energy in box
    TotalGPE = TotalGPEpp + TotalGPEpg + TotalGPEg
    GPE.append(TotalGPE)

    # Now calculate the total energy of the gas:
    totalg = KEg + GPEg + GPEpg + TE_val

    # Counter for bound and unbound mass
    bound_mass = 0
    unbound_mass = 0
    # Find unbound mass:
    for i in range(totalg.size):
        if totalg[i] < 0:
            bound_mass = bound_mass + dd["CellMass"][i]
        else:
            unbound_mass = unbound_mass + dd["CellMass"][i]
    bound_percentage = (bound_mass/(sum(dd["CellMass"]))) * 100.
    unbound_percentage = (unbound_mass/(sum(dd["CellMass"]))) * 100.
    bound.append(bound_percentage)
    unbound.append(unbound_percentage)

    #Finally calculate the total energy
    TotalE = sum(totalg) + TotalKEp + TotalGPEpp
    Total.append(TotalE)

#Create Plots
```

A.2.5 Fall-Back Time

```
# !/usr/bin/env python
# Calculates fall back time of gas from edge of grid
# uses Equations 4.4 and 4.5

from yt.mods import *



# Defines function:
def g_accel(mass, radius):
# Takes mass and radius in cgs and returns value in cgs
    g_top = (-6.67259e-8) * mass
    g_bot = (radius) ** 2.
    g = g_top / g_bot
    return g

def projectile_height(velocity, time_step, acceleration):
# Finds the change in height for a projectile over a time step.
# All units given in CGS
    first_term = velocity * time_step
    second_term = 0.5 * acceleration * (time_step ** 2.)
    h = first_term + second_term
    return h

def velocity(initial_velocity, acceleration, time_step):
```

```
#Calculates the velocity over a given time step. All units given in CGS.
    v = initial_velocity + acceleration*time_step
    return v

def escape_vel(mass, radius):
# All units are in CGS
    v = ((2. * 6.67259e-8 * mass) / radius) ** 0.5
    return v

def distance(point1, point2):
#Takes in the position as cm and give the separation in cm
    x_diff_sq = (point1[0] - point2[0]) ** 2.
    y_diff_sq = (point1[1] - point2[1]) ** 2.
    z_diff_sq = (point1[2] - point2[2]) ** 2.
    result = (((x_diff_sq) + (y_diff_sq) + (z_diff_sq)) ** (0.5))
    return result

def center_of_mass(mass1, mass2, position1, position2):
    # Everything in cgs
    x_top = mass1 * position1[0] + mass2 * position2[0]
    y_top = mass1 * position1[1] + mass2 * position2[1]
    z_top = mass1 * position1[2] + mass2 * position2[2]
    bot = mass1 + mass2
    x = x_top / bot
    y = y_top / bot
    z = z_top / bot
    com = [x, y, z]
    return com

# Load data:
```

```
pf = load("Data Directory")

# Define values:
lu = pf['LengthUnits']     # Length of box in cm
dim = pf.domain_dimensions[0]
edge = [0.0, dim-1]
gl = lu / dim              # Length of a cell in cm
dt = 60. * 60. * 24. * 7.  # Time step of a week given in seconds

# Find the mass of the particles:
dd = pf.h.all_data()
pm1 = init_dd["ParticleMass"][0]
pm2 = init_dd["ParticleMass"][1]
pmass = pm1 + pm2

#Get the grids in the data dump:
g = pf.h.grids[0]
dd = pf.h.all_data()

#Define Arrays:
cmass = [] # Mass of gas in cell (in grams) leaving along orbital plain
time = [] # Time for gas to fall back into the box
cmass_perp = [] # Mass of gas leaving perpendicular to orbital plain
time_perp = [] # Time for this gas to fall back into the box.

pp1 = [dd["particle_position_x"][0] * lu,
dd["particle_position_y"][0] * lu, dd["particle_position_z"][0] * lu]
pp2 = [dd["particle_position_x"][1]*lu,
dd["particle_position_y"][1] * lu, dd["particle_position_z"][1] * lu]
com = center_of_mass(pm1, pm2, pp1, pp2)
```

```
#Left and right edge:
for x in edge:
    for y in range(dim):
        for z in range(dim):
            position = [(x + 0.5) * gl, (y + 0.5) * gl, (z + 0.5) * gl]
            r = distance(position, com)
            v_x = g["x-velocity"][x,y,z]
            v_y = g["y-velocity"][x,y,z]
            v_z = g["z-velocity"][x,y,z]
            v_rad = (v_x ** 2. + v_y ** 2. + v_z ** 2.) ** 0.5
            ve = escape_vel(pmass, r)
            # Only selects bound gas to calculate fall-back time
            if v_rad < ve:
                t = 0.
                cell = g["CellMass"][x, y, z]
                cmass.append(cell)
                # Sets initial radius to the edge of the box for the cell
                r_0 = r
                a = g_accel(pmass, r)
                # While the gas is outside the computational domain
                while r > r_0:
                    height = projectile_height(v_rad, dt, a)
                    r = r + height
                    v_rad = velocity(v_rad, a, dt)
                    a = g_accel(pmass, r)
                    t = t + dt
                time.append(t / dt)

#Front and back edge:
```

```
print "For front and back edge"
for x in range(dim):
    for y in edge:
        for z in range(dim):
            position = [(x + 0.5) * gl, (y + 0.5) * gl, (z + 0.5) * gl]
            r = distance(position, com)
            v_x = g["x-velocity"][x,y,z]
            v_y = g["y-velocity"][x,y,z]
            v_z = g["z-velocity"][x,y,z]
            v_rad = (v_x ** 2. + v_y ** 2. + v_z ** 2.) ** 0.5
            ve = escape_vel(pmass, r)
            # Only selects bound gas to calculate fall-back time
            if v_rad < ve:
                t = 0.
                cell = g["CellMass"][x, y, z]
                cmass.append(cell)
                # Sets initial radius to the edge of the box for the cell
                r_0 = r
                a = g_accel(pmass, r)
                # While the gas is outside the computational domain
                while r > r_0:
                    height = projectile_height(v_rad, dt, a)
                    r = r + height
                    v_rad = velocity(v_rad, a, dt)
                    a = g_accel(pmass, r)
                    t = t + dt
                time.append(t / dt)

#Top and bottom edge:
print "For top and bottom edge"
```

```
for x in range(dim):
    for y in range(dim):
        for z in edge:
            position = [(x + 0.5) * gl, (y + 0.5) * gl, (z + 0.5) * gl]
            r = distance(position, com)
            v_x = g["x-velocity"][x,y,z]
            v_y = g["y-velocity"][x,y,z]
            v_z = g["z-velocity"][x,y,z]
            v_rad = (v_x ** 2. + v_y ** 2. + v_z ** 2.) ** 0.5
            ve = escape_vel(pmass, r)
            # Only selects bound gas to calculate fall-back time
            if v_rad < ve:
                t = 0.
                cell = g["CellMass"][x, y, z]
                cmass_perp.append(cell)
                # Sets initial radius to the edge of the box for the cell
                r_0 = r
                a = g_accel(pmass, r)
                # While the gas is outside the computational domain
                while r > r_0:
                    height = projectile_height(v_rad, dt, a)
                    r = r + height
                    v_rad = velocity(v_rad, a, dt)
                    a = g_accel(pmass, r)
                    t = t + dt
                time_perp.append(t / dt)

# Create Plot
```

A.2.6 Mach Number Calculations

```
# !/usr/bin/env python
# Find the velocity and Mach number of the particles, and local sound speed.

# Define lists:
time = [] # Time in the simulation
grids_1 = [] # Selected grids around particle 1 to calculate sound speed
grids_2 = [] # Selected grids around particle 2 to calculate sound speed
mach_1 = [] # Mach number of particle 1
mach_2 = [] # Mach number of particle 2
ss_1 = [] # Local sound speed of gas around particle 1
ss_2 = [] # Local sound speed of gas around particle 2
v_1 = [] # Corrected velocity of particle 1 for bulk velocity
v_2 = [] # Corrected velocity of particle 2 for bulk velocity

# Import data:
ts = TimeSeriesData.from_filenames("Data Directory")
pf = load ("Data Directory")

# Define values:
dim = pf.domain_dimensions[0]
lu = pf["LengthUnits"]      # Length of the box in cm
gl = lu/dim              # Length of a cell in cm

for pf in ts:
    time_val = pf.current_time
    time.append(time_val)

    dd = pf.h.all_data()
    g = pf.h.grids[0]

    # Find the position of the particles:
     # Round of particle position to nearest cell:
    pp1 = [int(round(dd["particle_position_x"][0] * dim)),
int(round(dd["particle_position_y"][0] * dim)),
int(round(dd["particle_position_z"][0] * dim))]
    pp2 = [int(round(dd["particle_position_x"][1] * dim)),
int(round(dd["particle_position_y"][1] * dim)),
int(round(dd["particle_position_z"][1] * dim))]

    # Find velocity of the particles:
    pv1 = [dd["particle_velocity_x"][0],
dd["particle_velocity_y"][0], dd["particle_velocity_z"][0]]
    pv2 = [dd["particle_velocity_x"][1],
dd["particle_velocity_y"][1], dd["particle_velocity_z"][1]]

# Find maginitude of velocity (i.e. speed)
    ps1 = ((pv1[0]) ** 2. + (pv1[1]) ** 2. + (pv1[2]) ** 2.) ** 0.5
    ps2 = ((pv2[0]) ** 2. + (pv2[1]) ** 2. + (pv2[2]) ** 2.) ** 0.5
    pv_1.append(ps1)
    pv_2.append(ps2)

    # find center of box around the particles with which the sound speed
    # is calculated based on the velocity vector of particle.
    # We want to select the cells IN FRONT of the particle.
    # For particle 1:
    if pv1[0] > 0.0:
        cc1x = pp1[0] + 2
        cc1xn = pp1[0] + 1
```

```
elif pv1[0] < 0.0:
    cc1x = pp1[0] - 2
    cc1xn = pp1[0] - 1
else:
    cc1x = pp1[0]
    cc1xn = pp1[0]
if pv1[1] > 0.0:
    cc1y = pp1[1] + 2
    cc1yn = pp1[1] + 1
elif pv1[1] < 0.0:
    cc1y = pp1[1] - 2
    cc1yn = pp1[1] - 1
else:
    cc1y = pp1[1]
    cc1yn = pp1[1]
if pv1[2] > 0.0:
    cc1z = pp1[2] + 2
    cc1zn = pp1[2] + 1
elif pv1[2] < 0.0:
    cc1z = pp1[2] - 2
    cc1zn = pp1[2] - 1
else:
    cc1z = pp1[2]
    cc1zn = pp1[2]
cc1 = [cc1x, cc1y, cc1z]
cc1n = [cc1xn, cc1yn, cc1zn]

# For particle 2:
if pv2[0] > 0.0:
    cc2x = pp2[0] + 2
```

```
        cc2xn = pp2[0] + 1
    elif pv2[0] < 0.0:
        cc2x = pp2[0] - 2
        cc2xn = pp2[0] - 1
    else:
        cc2x = pp2[0]
        cc2xn = pp2[0]
    if pv2[1] > 0.0:
        cc2y = pp2[1] + 2
        cc2yn = pp2[0] + 1
    elif pv2[1] < 0.0:
        cc2y = pp2[1] - 2
        cc2yn = pp2[0] - 1
    else:
        cc2y = pp2[1]
        cc2yn = pp2[0]
    if pv2[2] > 0.0:
        cc2z = pp2[2] + 2
        cc2zn = pp2[2] + 1
    elif pv2[2] < 0.0:
        cc2z = pp2[2] - 2
        cc2zn = pp2[2] - 1
    else:
        cc2z = pp2[2]
        cc2zn = pp2[2]
    cc2 = [cc2x, cc2y, cc2z]
    cc2n = [cc2xn, cc2yn, cc2zn]

    # Get the relevant grids, the sound speed, and bulk velocities:
    ssg_1 = []
```

```
ssg_2 = []
bvx_1 = []
bvy_1 = []
bvz_1 = []
bvx_2 = []
bvy_2 = []
bvz_2 = []
# For particle 1:
for x in range(cc1[0] - 1, cc1[0] + 2):
    for y in range(cc1[1] - 1, cc1[1] + 2):
        for z in range(cc1[2] - 1, cc1[2] + 2):
            if [x, y, z] != cc1n:
                ssg1 = g["SoundSpeed"][x,y,z]
                ssg_1.append(ssg1)
                bvx = g["x-velocity"][x,y,z]
                bvx_1.append(bvx)
                bvy = g["y-velocity"][x,y,z]
                bvy_1.append(bvy)
                bvz = g["z-velocity"][x,y,z]
                bvz_1.append(bvz)
# Find average sound speed and bulk velocity of gas around particle 1
ss1 = sum(ssg_1) / len(ssg_1)
ss_1.append(ss1)
bvx1 = sum(bvx_1) / len(bvx_1)
bvy1 = sum(bvy_1) / len(bvy_1)
bvz1 = sum(bvz_1) / len(bvz_1)
bv1 = [bvx1, bvy1, bvz1]

# For particle 2:
for x in range(cc2[0]-1, cc2[0]+2):
        for y in range(cc2[1]-1, cc2[1]+2):
            for z in range(cc2[2]-1, cc2[2]+2):
                if [x,y,z] != cc2n:
                    ssg2 = g["SoundSpeed"][x,y,z]
                    ssg_2.append(ssg2)
                    bvx = g["x-velocity"][x,y,z]
                    bvx_2.append(bvx)
                    bvy = g["y-velocity"][x,y,z]
                    bvy_2.append(bvy)
                    bvz = g["z-velocity"][x,y,z]
                    bvz_2.append(bvz)
    # Find average sound speed and bulk velocity of gas around particle 2
    ss2 = sum(ssg_2) / len(ssg_2)
    ss_2.append(ss2)
    bvx2 = sum(bvx_2) / len(bvx_2)
    bvy2 = sum(bvy_2) / len(bvy_2)
    bvz2 = sum(bvz_2) / len(bvz_2)
    bv2 = [bvx2, bvy2, bvz2]

    # Correcting velocity of particles for bulk velocity of the gas
    v1 = ((pv1[0] - bv1[0]) ** 2. + (pv1[1] - bv1[1]) ** 2. +
(pv1[2] - bv1[2]) ** 2.) ** 0.5
    v2 = ((pv2[0] - bv2[0]) ** 2. + (pv2[1] - bv2[1]) ** 2. +
(pv2[2] - bv2[2]) ** 2.) ** 0.5
    v_1.append(v1)
    v_2.append(v2)

    m1 = v1/ss1
    m2 = v2/ss2
    mach_1.append(m1)
    mach_2.append(m2)
```

References

Adamow M., Niedzielski A., Villaver E., Nowak G., Wolszczan A., 2012, The Astrophysical Journal Letters, 754, L15

Alikutty N. M., De Marco O., Nordhaus J., Wardle M., 2011, AIP Conference Proceedings, pp 319–323

Badenes C., Maoz D., 2012, The Astrophysical Journal Letters, 749, L11

Bear E., Kashi A., Soker N., 2011, MNRAS, 416, 1965

Bond H. E., 2000 Vol. 199, Binarity of central stars of planetary nebulae. eprint: arXiv:astro-ph/9909516, p. 115

Bouy H., 2011, Proceedings of the International Astronomical Union, 6, 41

Charpinet S., Fontaine G., Brassard P., Green E. M., Van Grootel V., Randall S. K., Silvotti R., Baran A. S., stensen R. H., Kawaler S. D., Telting J. H., 2011, Nature, 480, 496

Colella P., Woodward P. R., 1984, Journal of Computational Physics, 54, 174

de Kool M., 1992, Astronomy and Astrophysics, 261, 188

De Marco O., Passy J.-C., Moe M., Herwig F., Low M.-M. M., Paxton B., 2011, MNRAS, 411, 2277

Drake A. J., Djorgovski S. G., Mahabal A., Beshore E., Larson S., Graham M. J., Williams R., Christensen E., Catelan M., Boattini A., Gibbs A., Hill R., Kowalski R., 2009, The Astrophysical Journal, 696, 870

Eggleton P. P., 1972, Monthly Notices of the Royal Astronomical Society, 156, 361

Garca-Senz D., Bravo E., Cabezn R. M., Woosley S. E., 2007, The Astrophysical Journal, 660, 509

Gingold R. A., Monaghan J. J., 1977, Monthly Notices of the Royal Astronomical Society, 181, 375

Han Z., Podsiadlowski P., Eggleton P. P., 1995, Monthly Notices of the Royal Astronomical Society, 272, 800

Harpaz A., 1998, The Astrophysical Journal, 498, 293

Herschel W., 1802, Philosophical Transactions of the Royal Society of London, 92, 477

Herwig F., 2000, Astronomy and Astrophysics, 360, 952

Hoyle F., Fowler W. A., 1960, The Astrophysical Journal, 132, 565

Hubber D. A., Falle S. a. E. G., Goodwin S. P., 2013, MNRAS, 432, 711

Iben Jr. I., Tutukov A. V., 1999, The Astrophysical Journal, 511, 324

Ivanova N., Justham S., Podsiadlowski P., 2014, arXiv:1409.3260 [astro-ph]

Jones M. I., Jenkins J. S., Bluhm P., Rojo P., Melo C. H. F., 2014, Astronomy and Astrophysics, 566, 113

Kashi A., Frankowski A., Soker N., 2010, The Astrophysical Journal Letters, 709, L11

Kashi A., Soker N., 2011, Monthly Notices of the Royal Astronomical Society, 417, 1466

Kasliwal M. M., 2012, Publications of the Astronomical Society of Australia, 29, 482

Kulkarni S. R., Ofek E. O., Rau A., Cenko S. B., Soderberg A. M., Fox D. B., Gal-Yam A., Capak P. L., Moon D. S., Li W., Filippenko A. V., Egami E., Kartaltepe J., Sanders D. B., 2007, Nature, 447, 458

Lada C. J., 2006, The Astrophysical Journal Letters, 640, L63

Lucy L. B., 1977, The Astronomical Journal, 82, 1013

Maoz D., Mannucci F., 2012, PASA - Publications of the Astronomical Society of Australia, 29, 447

Marsh T. R., Parsons S. G., Bours M. C. P., Littlefair S. P., Copperwheat C. M., Dhillon V. S., Breedt E., Caceres C., Schreiber M. R., 2014, MNRAS, 437, 475

Melis C., Farihi J., Dufour P., Zuckerman B., Burgasser A. J., Bergeron P., Bochanski J., Simcoe R., 2011, The Astrophysical Journal, 732, 90

Meyer F., Meyer-Hofmeister E., 1979, Astronomy and Astrophysics, 78, 167

Miszalski B., Acker A., Moffat A. F. J., Parker Q. A., Udalski A., 2009, Astronomy and Astrophysics, 496, 813

Mitchell D. S., Reffert S., Trifonov T., Quirrenbach A., Fischer D. A., 2013, Astronomy and Astrophysics, 555, 87

Mustill A. J., Villaver E., 2012, The Astrophysical Journal, 761, 121

Nicholls C. P., Melis C., Soszyski I., Udalski A., Szymaski M. K., Kubiak M., Pietrzyski G., Poleski R., Ulaczyk K., Wyrzykowski ., Kozowski S., Pietrukowicz P., 2013, Monthly Notices of the Royal Astronomical Society, 431, L33

Nomoto K., Thielemann F.-K., Yokoi K., 1985 Vol. 21, Production of CNO isotopes in type i supernovae. pp 341–353

Ostriker E. C., 1999, The Astrophysical Journal, 513, 252

Paczynski B., 1976 Vol. 73, Common envelope binaries. p. 75

Passy J.-C., De Marco O., Fryer C. L., Herwig F., Diehl S., Oishi J. S., Mac Low M.-M., Bryan G. L., Rockefeller G., 2012, The Astrophysical Journal, 744, 52

Passy J.-C., Herwig F., Paxton B., 2012, The Astrophysical Journal, 760, 90

Passy J.-C., Mac Low M.-M., De Marco O., 2012, The Astrophysical Journal Letters, 759, L30

Paxton B., Bildsten L., Dotter A., Herwig F., Lesaffre P., Timmes F., 2011, The Astrophysical Journal Supplement Series, 192, 3

Paxton B., Cantiello M., Arras P., Bildsten L., Brown E. F., Dotter A., Mankovich C., Montgomery M. H., Stello D., Timmes F. X., Townsend R., 2013, The Astrophysical Journal Supplement Series, 208, 4

Raghavan D., McAlister H. A., Henry T. J., Latham D. W., Marcy G. W., Mason B. D., Gies D. R., White R. J., ten Brummelaar T. A., 2010, The Astrophysical Journal Supplement Series, 190, 1

Rasio F. A., Livio M., 1996, The Astrophysical Journal, 471, 366

Rau A., Kulkarni S. R., Ofek E. O., Yan L., 2007, The Astrophysical Journal, 659, 1536

Regos E., Tout C. A., 1995, Monthly Notices of the Royal Astronomical Society, 273, 146

Ricker P. M., Taam R. E., 2008, The Astrophysical Journal Letters, 672, L41

Ricker P. M., Taam R. E., 2012, ApJ, 746, 74

Riess A. G., Filippenko A. V., Challis P., Clocchiatti A., Diercks A., Garnavich P. M., Gilliland R. L., Hogan C. J., Jha S., Kirshner R. P., Leibundgut B., Phillips M. M., Reiss D., Schmidt B. P., Schommer R. A., Smith R. C., Spyromilio J., Stubbs C., Suntzeff N. B., Tonry J., 1998, The Astronomical Journal, 116, 1009

Saffer R. A., Green E. M., Bowers T., 2001 Vol. 226, The binary origins of hot subdwarfs: New radial velocities. ASP Conference Proceedings, p. 408

Sana H., de Mink S. E., de Koter A., Langer N., Evans C. J., Gieles M., Gosset E., Izzard R. G., Le Bouquin J.-B., Schneider F. R. N., 2012, Science, 337, 444

Sandquist E. L., Taam R. E., Chen X., Bodenheimer P., Burkert A., 1998, The Astrophysical Journal, 500, 909

Sato B., Toyota E., Omiya M., Izumiura H., Kambe E., Masuda S., Takeda Y., Itoh Y., Ando H., Yoshida M., Kokubo E., Ida S., 2008, Publ Astron Soc Jpn, 60, 1317

Schaefer B. E., Pagnotta A., 2012, Nature, 481, 164

Schreiber M. R., Gnsicke B. T., 2003, Astronomy and Astrophysics, 406, 305

Siess L., Livio M., 1999a, Monthly Notices of the Royal Astronomical Society, 304, 925

Siess L., Livio M., 1999b, Monthly Notices of the Royal Astronomical Society, 308, 1133

Soker N., 1998, The Astronomical Journal, 116, 1308

Soker N., Kashi A., 2012, The Astrophysical Journal, 746, 100

Stone J. M., Norman M. L., 1992, The Astrophysical Journal Supplement Series, 80, 753

Sullivan M., Kasliwal M. M., Nugent P. E., Howell D. A., Thomas R. C., Ofek E. O., Arcavi I., Blake S., Cooke J., Gal-Yam A., Hook I. M., Mazzali P., Podsiadlowski P., Quimby R., Bildsten L., Bloom J. S., Cenko S. B., Kulkarni S. R., Law N., Poznanski D., 2011, ApJ, 732, 118

Tocknell J., Marco O. D., Wardle M., 2014, MNRAS, 439, 2014

Turk M. J., Smith B. D., Oishi J. S., Skory S., Skillman S. W., Abel T., Norman M. L., 2011, The Astrophysical Journal Supplement Series, 192, 9

Tylenda R., Hajduk M., Kamiski T., Udalski A., Soszyski I., Szymaski M. K., Kubiak M., Pietrzyski G., Poleski R., Wyrzykowski ., Ulaczyk K., 2011, Astronomy and Astrophysics, 528, 114

Tylenda R., Soker N., 2006, Astronomy and Astrophysics, 451, 223 van den Heuvel E. P. J., 1976 Vol. 73, Late stages of close binary systems. p. 35

Villaver E., Livio M., 2007, The Astrophysical Journal, 661, 1192

Villaver E., Livio M., 2009, ApJ, 705, L81

Volschow M., Banerjee R., Hessman F. V., 2014, Astronomy and Astrophysics, 562, 19

Webbink R. F., 1984, The Astrophysical Journal, 277, 355

Webbink R. F., 2007, Short Period Binary Stars

Wolszczan A., Frail D. A., 1992, Nature, 355, 145

Woosley S. E., Weaver T. A., 1994, The Astrophysical Journal, 423, 371

Zorotovic M., Schreiber M. R., Gnsicke B. T., Nebot Gmez-Morn A., 2010, Astronomy and Astrophysics, 520, 86

Zuckerman B., Koester D., Melis C., Hansen B. M., Jura M., 2007, The Astrophysical Journal, 671, 872

Zuckerman B., Melis C., Klein B., Koester D., Jura M., 2010, The Astrophysical Journal, 722, 725