Tidal Evolution of Close–In Extra–Solar Planets

Fran Bartolić

Supervisors: izv.prof.dr. Dejan Vinković, dr.sc. Alexander James Mustill

Split, September 2015

Bachelor Thesis in Physics

Department of Physics
Faculty of Science
University of Split

SVEUČILIŠTE U SPLITU
SPLIT

Abstract

Observed population of giant planets around evolved stars shows a paucity of planets with semi-major axes below 0.5 au, in contrast to main sequence stars where those planets are present. This lack of close-in planets has been attributed to either planet engulfment due to tidal forces or substantially higher masses of evolved host stars implying different initial semi-major axis distributions, because of different parameters of planet formation and migration. However, there is recent evidence that the masses of evolved stars have been systematically overestimated and that they should not differ considerably from the main sequence sample, casting doubt on the latter hypothesis. In this project, to investigate the observed lack of close-in planets around evolved stars, I develop a tidal evolution code coupled with a population synthesis model which is based on probability distributions derived from observational data. I find that the observed paucity of planets cannot be attributed to tidal engulfment alone. One possibility is that the lack of close-in planets is the combination of inherently different samples of stars, tidal engulfment, and a statistically insignificant sample of planets. However, further observations are necessary to reach any firm conclusion.

Contents

List of Figures

Chapter 1

Introduction

The number of detected exoplanets and planet candidates has increased considerably in recent years. About two thirds of all confirmed planets have been found using the transit method, the rest were discovered by radial-velocity (RV) surveys. The *Kepler* mission alone found 3697 candidates based on the data collected during the first three years of the mission (Rowe et al. 2015).

Observations show a lack of planets in close orbits around giant stars (ex. Johnson et al. 2007; Sato et al. 2007; Bowler et al. 2010) despite the fact that those planets (so called hot Jupiters, Jupiter like planets orbiting within 0.1 au) are found around main-sequence (MS) stars. The prevailing hypothesis explaining the deficit of hot Jupiters around giant stars is tidal engulfment of planets by the evolving star during the subgiant and red-giant branches of stellar evolution.

Recent radial-velocity surveys of subgiants with masses greater than $\sim 1.5 M_\odot$ show that there are no detected planets in a region that extends all the way up to 0.5 au. It is not clear that tidal engulfment is the only mechanism responsible for this second deficit. In this project, I verify the tidal engulfment hypothesis as an explanation for the lack of observed close-in planets, and investigate the possible if it could also be responsible for the lack of giant planets around evolved stars up to 0.5 au.

The outline of the thesis is as follows. In Chapter 2 I discuss the theoretical background necessary for the project, namely, evolution of stars up to the end of the asymptotic giant branch (AGB), and tidal evolution, which describes how planetary orbits change when subjected to a gravitational potential of a tidally deformed star. Observational data relevant to the project are discussed in Chapter 3. In Chapter 4 I describe the code simulating the orbit of a planet around an evolving star and the statistical methods used for population synthesis. Finally, in Chapter 5 the results are summarized and compared to observations, and the work is put into context.

Chapter 2

Theory

The theory chapter is divided into three parts. Section 2.1 describes the general theory of stellar evolution, concentrating on the evolution of stars in the $1-3M_\odot$ mass range, up to the end of the AGB. Stars more massive than $3M_\odot$ are not interesting for the project since Reffert et al. (2015) showed that the probability of finding a planet around giant stars with masses greater than $3M_\odot$ is near zero. The discussion mostly follows the stellar evolution textbook Prialnik (2009).

Section 2.2 concerns the formation of tides on two gravitationally interacting bodies. Two different models are discussed. The first one is a relatively simple model based on a so called Q parameter, well described in the literature. The second is a more complex model based on the theory developed by Zahn (1977), which gives a bit more insight into the internal properties of the star.

2.1 Stellar evolution

The Hertzsprung–Russell diagram

In the early 20th astronomers Ejnar Hertzsprung and Henry N. Russell independently noticed a certain trend among stars in our galactic neighbourhood. Most of the stars they studied lay on a single narrow line in a diagram of L (stellar luminosity) vs. T, the surface temperature of the star. Unsurprisingly, these diagrams are called Hertzsprung–Russell (H-R) diagrams (Fig. 2.1) and they provide an excellent overview of the stellar population and evolution.

There are two ways to interpret Figure 2.1. Either each branch on the H-R diagram belongs to an intrinsically different type of star or we are seeing roughly the same types of stars (i.e. differing in mass), but at different evolutionary stages. The latter interpretation turns out to be correct.

Observations of single stars can tell us nothing about stellar evolution because they are like a snapshot in the life of a star. However, by statistically analyzing samples of stars we can infer the relative durations of evolutionary phases. Thus we expect that the stars with

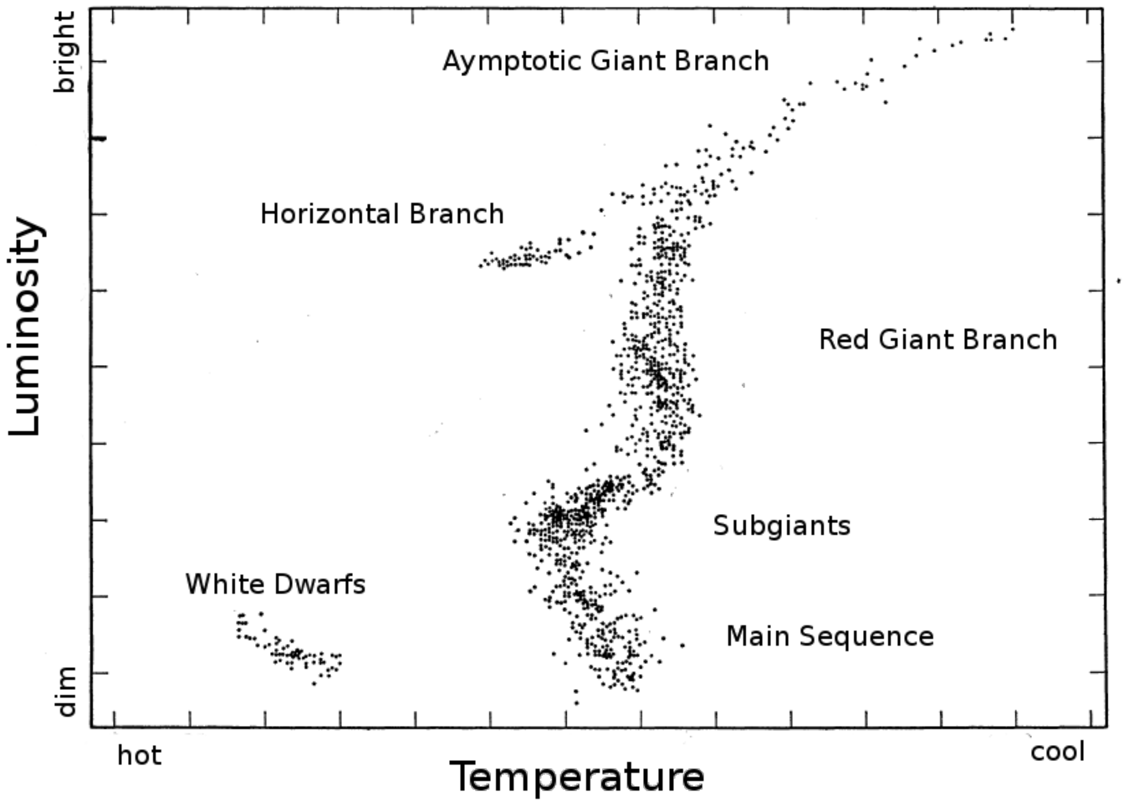

Figure 2.1: H-R digram showing a population of low to intermediate mass stars. Figure taken from https://www.mps.mpg.de/3664169/stars

properties most commonly observed (such as those belonging to a thin strip in Fig. 2.1, called the main sequence (MS)) correspond to longest evolutionary phases.

Equations of stellar structure

In order to develop a theory of stellar structure and evolution, several basic assumptions are needed:

- stars are *isolated* in space, their structure and evolution depends only on intrinsic properties such as mass and composition. This is true for most stars except for stars in dense clusters and binaries, which require a more complex theory.

- stars are formed with a *homogeneous composition*. The molecular cloud from which stars form is made of almost entirely hydrogen and helium, with traces of heavy elements, and it is well-mixed so this is a reasonable assumption.

- *spherical symmetry*. Gravity is a central force. The collapse of a sufficiently massive molecular cloud always results in a spherically symmetric object. There are however small deviations from spherical symmetry due to magnetic fields and centrifugal forces which can usually be neglected. Thus, all stellar properties can only change with the radial distance r from the centre. Departure from spherical symmetry becomes vitally important in Section 2.2, where the star cannot be treated as a sphere.

We assume that the star is in *local thermodynamic equilibrium* (LTE), that is, the radiation in stellar interiors is very nearly blackbody radiation described by the Planck distribution. The assumption of LTE is justifiable because the mean free time of photons inside the star is very small (typically on the order of $1cm$), such that the gas and radiation may be assumed to be in thermodynamic equilibrium locally, that is, the temperature of matter and radiation at any given point inside the star is the same. It is a great simplification because it implies that all thermodynamics properties can be calculated in terms of the temperature T, density ρ, and composition. The structure of a star with given mass M is thus determined at any time t if ρ, T, and the mass fractions of elements are known at each point.

The first equation of stellar structure is simply the statement of mass conservation.

$$\frac{\partial r}{\partial m} = \frac{1}{4\pi r^2 \rho} \tag{2.1}$$

The second equation results from conservation of momentum.

$$\ddot{r} = -\frac{Gm}{r^2} - 4\pi r^2 \frac{\partial P}{\partial m}$$

Because of the long lasting stellar evolution phases, in most cases there is no sudden acceleration of the gas element, and the star is considered to be in hydrostatic equilibrium. The above equation then reduces to

$$\frac{dP}{dm} = -\frac{Gm}{4\pi r^4} \tag{2.2}$$

A direct consequence of this is that the pressure inside of a star always deceases outwards.

Eqs. 2.1 and 2.2 determine the mechanical structure of the star in hydrostatic equilibrium, they constitute a set of two equations in three unknown functions of mass, r, P, and ρ. To solve them, we need a third condition, usually between P and ρ, called the *equation of state*.

The third equation is a consequence of the conservation of energy.

$$\frac{\partial l}{\partial m} = \epsilon_{nuc} - \epsilon_{\nu} \tag{2.3}$$

Where l is the stellar luminosity, ϵ_{nuc} is the rate at which nuclear energy is produced per unit mass, and ϵ_{ν} is the rate at which energy is removed from the stellar interior by the release of energetic neutrinos.

Finally, the fourth equation of stellar structure deals with energy transfer inside the star, it is valid in the case of radiative transfer, provided radiative diffusion constitutes the only means of energy transfer.

$$\frac{\partial T}{\partial m} = -\frac{3}{4ac}\frac{\kappa}{T^3}\frac{l}{(4\pi r^2)^2} \tag{2.4}$$

Where κ is the *opacity coefficient*, a measure of transparency of stellar interior to radiation, a is the radiation constant($a = \frac{4\sigma}{c}$ where σ is the Stefan–Boltzmann constant), and c is the speed of light.

These four equations constitute a set of four coupled differential equations. Once the equation of state $P(\rho, T, X_i)$, the opacity $\kappa(\rho, T, X_i)$ and the nuclear energy generation rates $\epsilon(\rho, T, X_i)$ (where X_i are the mass fractions of various elements) are given, the above equations involve four independent functions of m: ρ, T, r and l. A simple choice of boundary conditions is $r = 0$, $l = 0$ at $m = 0$ and $\rho = 0$, $T = 0$ at $m = M$.

This is then a well defined system of differential equations, however, it is highly non-linear, coupled, and it constitutes a two-point boundary value problem, as such, it cannot be solved analytically without drastically simplifying assumptions. The stellar structure equations have to be solved numerically, this will be further discussed in Section 4.1. The solutions to these equations are not necessarily unique as there are some special cases when the solutions are degenerate.

Formation of stars

Stars form with the collapse of an interstellar gas cloud. An interstellar cloud is a diffuse medium, composed of around 70% hydrogen by mass, the rest is almost entirely helium, with only traces of heavy elements. A molecular cloud is in hydrostatic equilibrium as long as the kinetic energy of the gas pressure is balanced by gravitational potential energy of the system. According to the virial theorem, the criterion for stability is that the gravitational potential energy must be equal to twice the thermal energy. A small disturbance such as an acoustic wave can cause the collapse of the cloud. The collapse results in a formation of a protostar, surrounded by a cloud of dust. If the mass of the protostar is high enough(higher than about $0.08 M_\odot$), the gravitational attraction is strong enough to initiate nuclear fusion.

Shortly after the formation of the protostar, around most stars (Cassan et al. 2012), planet formation begins in the protoplanetary disk which surrounds the star.

Evolution on the main sequence

Stars on the MS slowly burn hydrogen as a nuclear fuel in the core. The nuclear energy is transported outward by radiation or convection. In low mass stars ($M \lesssim 0.3 M_\odot$) energy is transfer mostly by convection, as they are fully convective. More massive stars have smaller and smaller outer convective envelopes. Stars more massive than the Sun, which burn hydrogen by the temperature-sensitive CNO (carbon-nitrogen-oxygen) cycle have convective cores and radiative envelopes. Some mass-loss occurs during MS but rates are usually very small.

What happens after MS depends on the stellar mass. Stars can be divided into two categories: stars with masses below $9 - 10 M_\odot$ end up as white dwarfs while the more massive ones undergo supernova explosions. The first category is divided into *red dwarfs* ($M \lesssim 0.7 M_\odot$), which reach the end of MS after tens of billions of years, *low-mass stars* ($0.7 \lesssim M \lesssim 2 M_\odot$) and *intermediate-mass stars* ($2 \lesssim M \lesssim 9 - 10 M_\odot$).

The Red Giant Branch

During the MS, a hydrogen-depleted core gradually grows in mass and hydrogen continues to burn in a shell surrounding the core. Mario Schönberg and Chandrasekhar derived a theoretical limit for the core mass M_c above which the pressure within the core would not be able to support the weight of the hydrogen envelope. The stability criterion is given by

$$\frac{M_c}{M} \lesssim 0.37 \left(\frac{\mu_{env}}{\mu_c}\right)^2 \tag{2.5}$$

Where μ_{env} and μ_c are the mean molecular weight of the envelope and the core respectively.

When the mass of the core reaches this limit, which is roughly 10% of the total mass for a solar type star, it starts to contract rapidly. The Schönberg-Chandrasekhar (S-C) limit is only valid for ideal gases. Cold and dense gases, in which the degenerate electrons supply most of the pressure, are capable of supporting the envelope even for massive cores. This electron degeneracy pressure is sufficiently high in the helium cores of stars with masses below about $2M_\odot$. The RGB evolution of low-mass stars thus differs qualitatively from intermediate-mass stars and it will be described separately.

Hydrogen-shell burning in intermediate-mass stars

The first phase of hydrogen-shell burning occurs slowly, until the S-C limit is reached and the contraction of the core speeds up, thus causing the envelope to expand. While this may seem counterintuitive, it is simply a consequence of the virial theorem. The exact opposite happens if instead the core expands. The expansion of the envelope occurs on a Kelvin-Helmholtz timescale (see for ex. Prialnik 2009, for definition), that is, very rapidly, causing a rightward trajectory of the star in the H-R diagram. Expansion continues until Helium is ignited in the core, at that point the star reaches its maximum luminosity and radius.

Hydrogen-shell burning in low-mass stars

Compared to intermediate-mass stars, low-mass stars($M \lesssim 2M_\odot$) have small or no convective cores. When they leave MS their cores are close to becoming degenerate. By the time the helium core has grown to $\approx 0.1M_\odot$, its density is large enough that the electron degeneracy pressure dominates and the S-C limit is either irrelevant or it has a very small influence on subsequent evolution of the star. Because of this, the star remains in thermal equilibrium throughout evolution, until it reaches the base of the RGB. This phase lasts for around 2 Gyr and it is called the *subgiant branch*.

As the helium core increases in mass, it slowly contracts and the radius and luminosity increase. Due to higher luminosity, the rate at which hydrogen is burned in the shell increases, this in turn accelerates the core mass growth leading to a feedback loop. The burning shell looses mass rapidly and at the same time the luminosity is increasing, this means that the nuclear energy generation rate per unit mass ϵ_{nuc} increases, and with it the temperature in the shell and the degenerate helium core.

When the core mass reaches about $0.5M_{\odot}$, independent of the total stellar mass, the temperature in the core is around 10^8K and helium ignites. For degenerate matter, pressure is not a function of temperature, thus, temperature in the core continues to grow, and since Helium fusion is highly dependant on the temperature ($\epsilon_{nuc} \propto T^{40}$) this results in a *thermonuclear runaway* process. This runaway process leads to a huge increase in luminosity, for a few seconds the luminosity of the core is equal to $10^{10}L_{\odot}$, comparable to that of an entire galaxy. However, this energy is completely absorbed in the envelope. The degeneracy in the core is lifted when the core temperature reaches around 3×10^8K. In the process, the core expands from a degenerate to a non-degenerate state and this caused the envelope to contract substantially.

The Horizontal Branch and Early Asymptotic Giant Branch

Following the RGB a phase of helium burning (HB) starts in the core while hydrogen continues to burn in the shell. The fusion of helium into carbon and oxygen produces only a tenth of the energy per unit of mass produced by hydrogen burning and the stellar luminosity is substantially higher. Consequently, the HB phase is significantly shorter than that of core hydrogen burning during MS.

Low-mass stars which started the HB occupy the so called *horizontal branch* in the HR diagram, a horizontal strip located between the main sequence and red giant branch. Since all these stars have equally massive cores at the end of the RGB, the different positions they occupy along the horizontal branch are due to differences in envelope mass and metallicity. In general, stars with higher metallicity and higher envelope mass occupy the red end of the branch (cooler effective temperature), the so called *red clump*. Observationally, it is difficult to distinguish between the first ascent red giant stars and helium burning red clump giants.

Intermediate-mass stars start burning helium gradually and end up at the left end of the HB.

Stellar radius as a function of time for various stellar masses in the $1 - 3M_{\odot}$ range is shown in Figure 2.2. Several features are apparent. First, more massive stars live shorter lives. Second, the stellar radius at the tip of the RGB reaches substantially higher values for low-mass star (those which undergo the helium flash) than intermediate-mass stars, and finally, the duration of the HB phase increases with stellar mass.

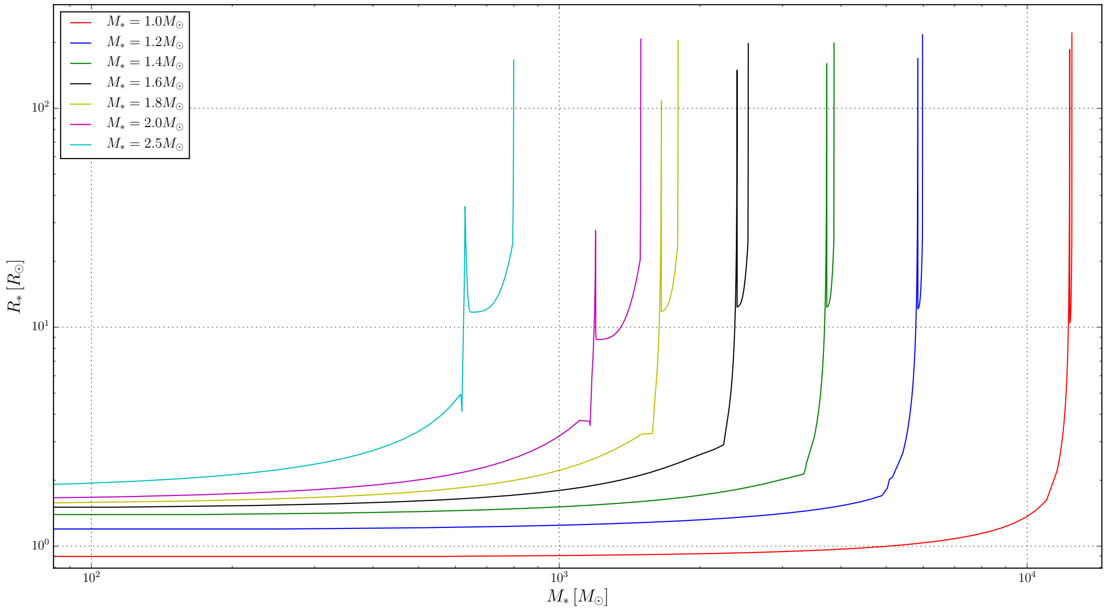

Figure 2.2: Stellar radius as a function of time up to the AGB phase, output of the stellar evolution code discussed in Section 4.1.

2.2 Tides and tidal evolution

So far we have concentrated only on the star, since this thesis in concerned with star-planet interaction, we need to investigate the influence of stellar evolution on the planetary system around it. The mechanism with which an evolving star interacts with planets is tidal dissipation. To understand tidal evolution, we first need to understand how and why tides form. The following discussion mostly follows the book by Murray & Dermott (1999).

Formation of tides

Tides form when two massive spherical bodies interact gravitationally. Probably the most familiar example of tidal phenomena is the Earth-Moon system. Both the Earth and the moon exert forces on each other, since one side of the Earth (or the Moon) experiences a greater force than the other, and the bodies are not perfectly rigid, a *tidal bulge* will form on both bodies.

In our the description of tide formation, we will call one body the primary - the one being subject to tidal forces, and other the secondary - the one exerting tidal forces. If we designate the mass of the primary by m_p and that of the secondary by m_s and assume (for now) that both are point-mass objects, then Newton's law of gravitation gives the mean mutual force $\langle F \rangle$ as

$$\langle F \rangle = G\frac{m_p m_s}{r^2} \tag{2.6}$$

where r is the separation between the two bodies. Assuming both bodies orbit their common centre of mass in circular orbits, the semi-major axis of the orbits are related to the masses by

$$\frac{a_s}{a_p} = \frac{m_p}{m_s} \tag{2.7}$$

where the constant separation between the bodies is $a = a_p + a_s$. This equation follows from the fact that the centre of mass is always on the line joining the two point-masses.

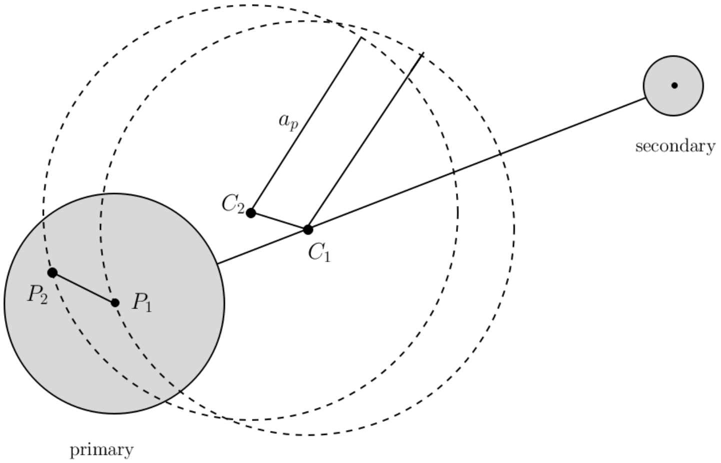

Figure 2.3: All the particles in the primary move in similar circles of radii a_p, but with different centres. The particles P_1 and P_2 are on circles with centres C_1 and C_2 respectively. Figure taken from Murray & Dermott (1999) (Fig. 4.2)

Designating the centre of mass of the two bodies by C_1, the centre of the primary by P_1, and some point inside the primary by P_2, we see from Fig. 2.3 that if P_1 circles around C_1 the every point P_2 moves in a circle of same radius a_p with a centre C_2 displaced from C_1 by the distance $|P_1P_2|$. It follows that every point in the primary is acted upon by equal (in magnitude and direction) centrifugal force, but not by equal gravitational forces $\mathbf{F}$. The centrifugal force is equal to the mean gravitational force $\langle \mathbf{F} \rangle$. Thus, tidal force that deforms the planet is given by

$$\mathbf{F_{tidal}} = \mathbf{F} - \langle \mathbf{F} \rangle \tag{2.8}$$

To determine the shape of the tidal bulge on the primary, we consider the potential V at some point P on the surface of the primary due to the secondary (treated as point mass). This is illustrated in Fig. 2.4. Thus, we have

$$V = -G\frac{m_s}{\Delta} \tag{2.9}$$

where Δ is the distance from the point P to the centre of the secondary. Applying the cosine rule, we have

$$\Delta = a\left[1 - 2\left(\frac{R_p}{a}\right)\cos\psi + \left(\frac{R_p}{a}\right)^2\right]^{\frac{1}{2}} \tag{2.10}$$

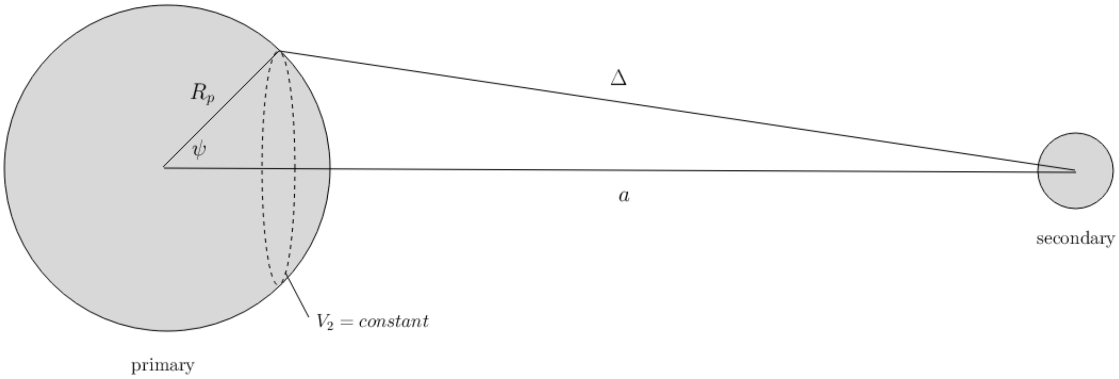

Figure 2.4: The relationship among the radius of the primary R_p, the semi-major axis of the secondary a and the distance Δ from a point P to the secondary. The dashed line denotes an equipotential surface. Figure taken from Murray & Dermott (1999) (Fig. 4.3)

Assuming $R_p/a \ll 1$ (a reasonable assumption in most cases) we can expand Eq. 2.10 binomially. Neglecting higher order terms, we have

$$\begin{aligned} V(\psi) &= -G\frac{m_s}{a}\left[1 + \left(\frac{R_p}{a}\right)\cos\psi + \left(\frac{R_p}{a}\right)^2 \frac{1}{2}(3\cos^2\psi - 1) + \ldots\right] \\ &\approx V_1 + V_2 + V_3 \end{aligned} \tag{2.11}$$

The first term V_1 in Eq. 2.11 is constant and it does not produce any force on the primary (its gradient is zero). The second term V_2 gives rise the centripetal force on the particle at the point P, needed for motion in a circle. The potential due to the third term V_3 can be written as

$$V_3(\psi) = -\zeta g \mathcal{P}_2(\cos\psi) \tag{2.12}$$

where

$$\zeta = \frac{m_s}{m_p}\left(\frac{R_p}{a}\right)^3 R_p \tag{2.13}$$

and

$$g = \frac{Gm_p}{R_p^2} \tag{2.14}$$

is the surface gravity of the primary, $\zeta g \mathcal{P}_2(\cos\psi)$ is said to be the amplitude of the *equilibrium tide* on the planet's surface. It is maximum for $\psi = 0$ or $\psi = \pi$ and minimum for $\psi = \pi/2$ or $\psi = 3\pi/2$. This explains why the Moon produces two high tides and two low tides on the Earth approximately every day.

So far, we have calculated the tidal deformation of the primary body due to the gravitational influence of the secondary. The surface of the deformed primary body is defined by

$$R(\theta) = C[1 + \epsilon_2 \mathcal{P}_2(\cos\theta)] \tag{2.15}$$

where θ is the azimuthal angle in spherical coordinates, $\epsilon_2 \ll 1$ is a constant and C is the mean radius.

We now wish to calculate the resulting gravitational potential created by the tidally deformed primary body, at a point P which can can be either interior ($r < C$) or exterior ($r > C$) to the deformed body and has spherical coordinates (r, μ, Φ), where $\mu = \cos\theta$ and θ is measured from the axis of symmetry of the tidal bulge.

The derivation of the resulting potential is lengthy and it has been omitted (see Murray & Dermott 1999, Chapter 4.3). The resulting potential for a point exterior to the body is

$$V_{ext}(r,\theta) = -\frac{4}{3}\pi C^3 \gamma G \left[\frac{1}{r} + \frac{3}{5}\frac{C^2}{r^3}\epsilon_2 \mathcal{P}_2(\cos\theta)\right] \tag{2.16}$$

where γ is the density. The first term in the parentheses corresponds to a potential of the spherical part of the body, the second one to the thin distribution of mass between the spherical and the deformed surface.

The constant Q model of tidal dissipation

Up to this point, we have assumed equilibrium tides and ignored any dissipative mechanisms. In this section, we consider a more realistic case, one in which tidal dissipation causes energy loss and a phase lag between the driving tidal force due to the secondary, and the response of the primary body. This means that the axis of symmetry of the tidal bulges no longer lies on the line connecting the two bodies.

The simplest way to model this phase lag is using a forced harmonic oscillator model. From elementary mechanics, we know that the phase lag of a harmonic oscillator can be expressed in terms of a single dimensionless number Q, called the *specific dissipation function* of the harmonic oscillator.

$$\sin\delta = -1/Q \tag{2.17}$$

The Q factor tells us how efficient is the dissipation of energy in a damped forced harmonic oscillator. Higher Q means less efficient energy dissipation, *i.e.*, less energy lost during one cycle. It is important to note that Q is independent of the frequency.

Consider a tide raised on the primary body by a secondary body moving in a circular, equatorial orbit with mean motion n (defined as $n = 2\pi/T$ where T is the orbital period) around the primary body, which is rotating with angular speed Ω. The secondary body raises a tide of frequency $2(\Omega - n)$ (two high tides per day in the case of the earth) on the primary. If $\Omega > n$ the tidal bulge is carried ahead of the secondary by an angle $\epsilon = \delta/2 = 1/2Q$. The opposite happens if $\Omega < n$, in that case the tidal bulge lags behind the tide-raising body by the same angle ϵ. This is illustrated in Fig. 2.5.

The misalignment of the axis of symmetry of the tidal bulges and the line connecting the two bodies results in a net *tidal torque* Γ being exerted on the secondary, given by

$$\mathbf{\Gamma} = \mathbf{r} \times \mathbf{F} \tag{2.18}$$

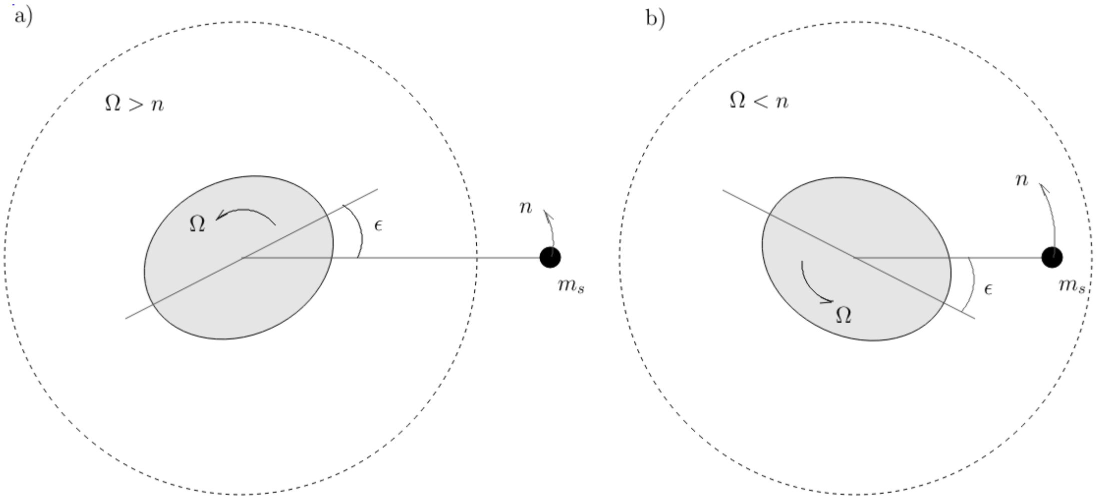

Figure 2.5: Tidal friction causes a negative phase lag δ in the response of the primary. Te figure shows a secondary body in circular orbit around the primary, raising a tide of frequency $2(\Omega - n)$ on the primary. (a) If $\Omega > n$ the secondary is above synchronous height (dashed line) and the tidal bulge is carried ahead of the secondary by an angle ϵ. (b)If $\Omega < n$ the secondary is below synchronous height and the tidal bulge lags behind the secondary by the same angle ϵ.

where

$$\mathbf{F} = -m_s \nabla V_{ext} \tag{2.19}$$

V_{ext} is the previously mentioned gravitational potential of the primary. Only the component of the force perpendicular to the line connecting the two bodies, $F_\psi = -(m_s/r)(\partial V_{ext}/\partial \psi)$ contributes to the torque and only the noncentral part of the potential $V_{nc,ext}$ contributes to that force component. Thus, the magnitude of the torque on the secondary is given by

$$\Gamma = -m_s \frac{\partial V_{nc,ext}}{\partial \psi} \tag{2.20}$$

If $\Omega > n$ the work done by the torque acts to increase the orbital energy of the system at a rate Γn. At the same time an equal and opposite torque works at a rate $\Gamma\Omega$ to decrease the rotational energy of the planet. Since $\Omega \neq n$ the total mechanical energy of the system, E, decreases at a rate

$$\dot{E} = -\Gamma(\Omega - n) < 0 \tag{2.21}$$

Conversely, if $\Omega < n$, the orbital energy of the system decreases, while the rotational energy of the planet increases. The total mechanical energy of the system still decreases at a rate

$$\dot{E} = \Gamma(\Omega - n) < 0 \tag{2.22}$$

In both cases, the energy is dissipated as heat in the planet and the rate of dissipation determines the rate of orbital evolution.

The total mechanical energy of the system is the sum of the rotational energy of the primary, $\frac{1}{2}I\Omega^2$, where I is the moment of inertia of the planet, and the orbital energy of the system is $-Gm_pm_s/2a$ (see Murray & Dermott 1999, Sect. 2.7). Thus, the rate of change of mechanical energy is

$$\dot{E} = \frac{d}{dt}\left(\frac{1}{2}I\Omega^2 - G\frac{m_pm_s}{2a}\right) = I\Omega\dot{\Omega} + G\frac{m_pm_s}{2a^2}\dot{a} \tag{2.23}$$

Total angular momentum of the system,

$$L = I\Omega + \frac{m_pm_s}{(m_p + m_s)}a^2n \tag{2.24}$$

is conserved. Thus, $\dot{L} = 0$, substituting the derivative of L into into Eq. 2.23, we have

$$\dot{E} = -\frac{1}{2}\frac{m_pm_s}{(m_p + m_s)}na\dot{a}(\Omega - n) \tag{2.25}$$

and because $\dot{E} < 0$ we have

$$\text{sign}(\dot{a}) = -\text{sign}(\dot{\Omega}) = \text{sign}(\Omega - n) \tag{2.26}$$

Therefore, if $\Omega > n$, the semi-major axis of the secondary increases while the rate of rotation of the planet decreases. This is what happens for the Earth-Moon system. Conversely, if $\Omega < n$, the semi-major axis of the secondary decreases while the rate of rotation of the primary increases, this is the case relevant to the thesis project since the rotation rate of the vast majority of subgiant and giant stars is substantially slower than the orbital period of their planets.

In order to calculate the timescale of orbital evolution, we need an expression for the magnitude of the tidal torque Γ as a function of the phase lag angle. From Eq. 2.16 it follows that the noncentral part of the external gravitational potential due to the deformed primary body at some point $P(\psi)$ is given by

$$V_{nc,ext} = -k_2\zeta\left(\frac{C}{r}\right)^3\mathcal{P}_2(\cos\psi) \tag{2.27}$$

The dimensionless coefficient k_2 is called a *Love number*. It is used as a way of disguising our ignorance of the body's internal structure. From equations (2.20), (2.27), and (2.13), using $\partial\mathcal{P}_2(\cos\psi)/\partial\psi = -\frac{3}{2}\sin 2\psi$, we have

$$\Gamma = \frac{3}{2}k_2\frac{Gm_s^2}{a^6}C^5 sin2\epsilon \tag{2.28}$$

Finally, from equations (2.21), (2.22),(2.25), and (2.28), we obtain

$$\dot{a} = \text{sign}(\Omega_p - n)\frac{3k_2}{Q}\frac{m_s}{m_p}\left(\frac{C}{a}\right)^5 na \tag{2.29}$$

We simplify Eq. 2.29 further by incorporating the Love number and the numerical factor in front of it into $Q' \equiv 3k_2/Q$. Using the Keplerian relation $n = \sqrt{G(m_p + m_s)/a^3}$, and

$m_p = M_*$, $m_s = m$, $C = R_*$ (where m is the mass of the planet, R_* and M_* are the stellar radius and mass respectively) and assuming a non-rotating star ($\Omega_p = 0$), we obtain the following equation which describes the orbital evolution of a planet orbiting a star experiencing tidal forces.

$$\dot{a} = -\frac{1}{Q'}\frac{m}{M_*}\sqrt{G(M_* + m)}\left(\frac{R_*}{a}\right)^5 a^{-1/2} \tag{2.30}$$

This is a nonlinear first-order differential equation in a, it can be solved analytically only in the case of constant stellar radius, with solution:

$$a(t) = \left(1 - \frac{13m\, R_*^5\, t\sqrt{G(M_* + m)}}{2M_* Q'}\right)^{2/13} \tag{2.31}$$

for an initial condition $a(t = 0) = 1$. In general however (and in this project especially), R_* is a function of time and in that case Eq. 2.30 has to be solved numerically.

Using this simple model, we can get some insight into the orbital evolution of a planet around an evolving star. Firstly, Eq. 2.30 is highly dependant on the stellar radius, it is by far the most important stellar parameter that is influencing the orbital evolution of the planet, it is thus important to accurately model the increase in stellar radius during and after the RGB phase. Secondly, this model has only one parameter directly dependant on stellar structure and independent of the orbital frequency of the planet, Q'. Lower Q' (*i.e.*,more efficient tidal dissipation) results in faster orbital decay. For MS stars, Q' is usually on the order of $10^8 - 10^9$ (Penev & Sasselov 2011), though the value is still debated. Lastly, the model is dependant of planet mass, more massive planets decay more quickly.

However, this model assumes that the Q parameter is independent of the stellar structure and the driving frequency (the mean motion of the planet), a rather naive assumption considering the complex nature of dissipation processes in the star. Thus, I have decided to use a more detailed model which incorporates the dependence on the driving frequency of the tidal force and the internal structure of the star. The model is described in the next section.

Zahn's model of tidal dissipation

This model is described in the paper by Zahn (1977), Zahn investigated tidal dissipation mechanism in binary stars, a case not too different from the one in this thesis, as the underlying physics is roughly the same. All the details of Zahn's theory are beyond the scope of this thesis and are not in direct connection with the analysis I have done. For this reason I merely describe his approach and state the final result. An interested reader is encouraged to read Zahn (1977).

Zahn's derivation of the tidal evolution equation follows roughly the same approach as the one we have outlined in the previous section. He expands the tidal potential due to the secondary body in spherical harmonics and then further decomposes each spherical harmonic in Fourier series of the mean anomaly (the secondary body is no longer assumed to be fixed in space). The reason for breaking the external gravity potential into such Fourier components is that one can then study the oscillations of a star forced by a potential which varies sinusoidally in time. If we then assume that the amplitudes of oscillations are small enough to be approximately linear, then the total response of the star is just the sum of its responses to each Fourier component. The problem is then reduced to determining the resulting gravitational potential of the deformed primary.

The outer potential created by the deformed primary body is then assumed in a most general form, as Fourier series with complex coefficients and it is found that the exchange of energy and angular momentum due between the two bodies is caused only by the imaginary part of the outer potential. Thus, by evaluating those imaginary coefficients, we completely describe the resulting outer potential. The resulting force acting on the secondary is then easily determined by calculating the gradient of the outer potential. Several different dissipation mechanisms are then evaluated:

- viscous dissipation
- radiative dissipation
- turbulent dissipation in stars possessing a convective envelope
- turbulent dissipation in stars possessing a convective core
- turbulent dissipation generated by the tide itself

The most effective mechanism is determined to be turbulent dissipation in stars possessing a convective envelope and the resulting differential equation for a secondary body orbiting a star with a convective envelope is

$$\dot{a} = -\frac{f}{\tau_d}\frac{M_{env}}{M_*}\,q(1+q)\left(\frac{R_*}{a}\right)^8 a \tag{2.32}$$

where

$$\tau_d = \left[\frac{M_{env}(R_* - R_{env})^2}{3L_*}\right]^{\frac{1}{3}} \tag{2.33}$$

$q = m/M_*$, M_{env} and R_{env} are the mass and radius of the stellar envelope respectively, a is the semi-major axis of the planet and m is its mass. τ_d is called the friction time, it measures the efficiency of dissipation (the smaller the time τ_d the more efficient the dissipation), it is expressed in terms of stellar radius R_*, envelope mass and radius, and stellar luminosity L_*. The dimensionless factor f is introduced to weaken the stellar tide when the orbital period of the planet is short compared to response time of the star. It is usually given by

$$f = \min\left[1, \left(\frac{P}{2\tau_d}\right)^n\right] \tag{2.34}$$

where n is 0,1, or 2. If $P > \tau_d$ then $f = 1$, otherwise $f = (P/2\tau_d)^n$, that is, the tide is weakened in proportion to the orbital period. There is some debate over what value of n is appropriate. I have chosen the use the value $n = 0$ (that is, $f = const. = 1$) to avoid abrupt changes in a resulting in unphysical step-functions for the semi-major axis. constant Note that Equation 2.32 is qualitatively similar to the constant Q model equation, Equation 2.30; $\dot{a}$ is again a very strong function of stellar radius. It cannot be solved analytically (even in the case of constant stellar radius), it must be integrated numerically, this is discussed in Section 4.2.

Stellar mass–loss

Tidal dissipation is not the only phenomenon influencing the orbit of a planet around an evolving star. During the RGB, a star experiences mass-loss in the form of a stellar wind due to its decreased surface gravity. The mass-loss is most severe at the tip of the RGB when the star rapidly expands and then contracts. Angular momentum of the planet-star system is given by $L = \sqrt{G(M_* + m)a(1-e^2)}$ (Landau & Lifshitz 1976), where e is the eccentricity of the planet. Using the fact that we conservation of angular momentum holds and assuming a circular orbit ($e = 0$), we have

$$\begin{aligned} \dot{L} &= \frac{d}{dt}\left(\sqrt{G(M_* + m)a}\right) \\ &= -\frac{1}{2}\left[G(M_* + m)a\right]^{-1/2} G\left[(\dot{M}_* + \dot{m})a + (M_* + m)\dot{a}\right] = 0 \end{aligned} \tag{2.35}$$

thus

$$\dot{a}_m = -\frac{\dot{M}_* + \dot{m}}{M_* + m}a \tag{2.36}$$

where the subscript m denotes that the term is due to mass-loss. The change in the planet's mass is not modelled in the project as it has been evaluated to be of negligible importance by Kunitomo et al. (2011). Setting $\dot{m} = 0$, $M_* + m \approx M_*$ and combining the mass-loss term with the tidal dissipation term, we obtain the final equation for the orbital evolution of a planet in orbit around an evolving star experiencing mass-loss.

$$\dot{a} = -\frac{f}{\tau_d}\frac{M_{env}}{M_*}q(1+q)\left(\frac{R_*}{a}\right)^8 a - \frac{\dot{M}_*}{M_*}a \tag{2.37}$$

Note that $\dot{M}_*$ is always negative, thus, we have two competing effects influencing orbital evolution: tidal dissipation and mass-loss.

Chapter 3

Observations of exoplanets

There are several methods of detecting exoplanets. The most prominent are transit photometry, radial velocity (RV), gravitational microlensing, direct imaging and astrometric detection. All these methods have different detectable physical quantities and different observational biases. The most recent available exoplanet catalogue is the one on www.exoplanets.org and it is the one I have used.

The vast majority of planets in the catalogue are detected through transit methods using the Kepler spacecraft. However, a significant number of these planets are false positives. Fressin et al. (2013) estimate that approximately 10% of the candidates are false positives, more recent work suggests that the percentage could be as high as 50% for single planetary systems. Another flaw of the Kepler planets is poorly constrained stellar radius, which is of crucial importance for the project. Thus, we have restricted ourselves to the planets detected with the more reliable RV method.

The RV method works by measuring the Doppler shift in the spectral lines of the observed planet-hosting star. As the star orbits the system's centre of mass, its radial velocity relative to the earth changes periodically, this causes a Doppler shift in observed spectra. Periodic trends in the data are then used to estimate the planet's orbital period. The RV method is also used to estimate the mass of the planet, however, only up to an inclination angle of the orbit as the detected quantity is the minimum mass of the planet $M = M_{true} \sin i$, not true mass.

The RV method has several observational biases. In order to confirm a planet, the star has to be observed for the time comparable to the planet's period. Since the longest RV surveys so far lasted about 7 years, RV observations are biased towards short period planets. For precise RV measurements, the targeted star must be bright (low apparent magnitude) and a slow rotator. Fast rotating stars have broad spectral lines which make RV observations very difficult. More massive stars on the main sequence are hot, which means that they have both fewer absorption lines, and they rotate quickly. For this reason there are very few detected main sequence stars more massive than about $1.5\,M_\odot$ in the exoplanet catalogue. However, it is possible to detect planets with RV surveys of more massive stars once the stars evolve off the main sequence. After MS they both cool down and decrease their rotational speed due to the increased moment of inertia.

The most important observational plot in this thesis is the the one showing the semi-major axis distribution of giant planets ($m > 1\,M_j$) as a function of stellar radius, shown in Fig. 3.1. The left group of planets corresponds to main sequence stars. They are mostly solar-type stars, with an average mass of around $1.1\,M_\odot$. A noticeable feature of this planet population is a semi-major axis gap (equivalently, a period gap) between around 0.08 au and 0.6 au. Burkert & Ida (2007) used Monte-Carlo simulations of planet formation and migration to reproduce this gap around stars with mass greater than $1.2\,M_\odot$. They found that the gap is more pronounced for higher mass stars since more massive stars have protoplanetary disks with shorter disk-depletion timescales (Kennedy & Kenyon 2009), which prevent Jovian planets migrating too far inwards.

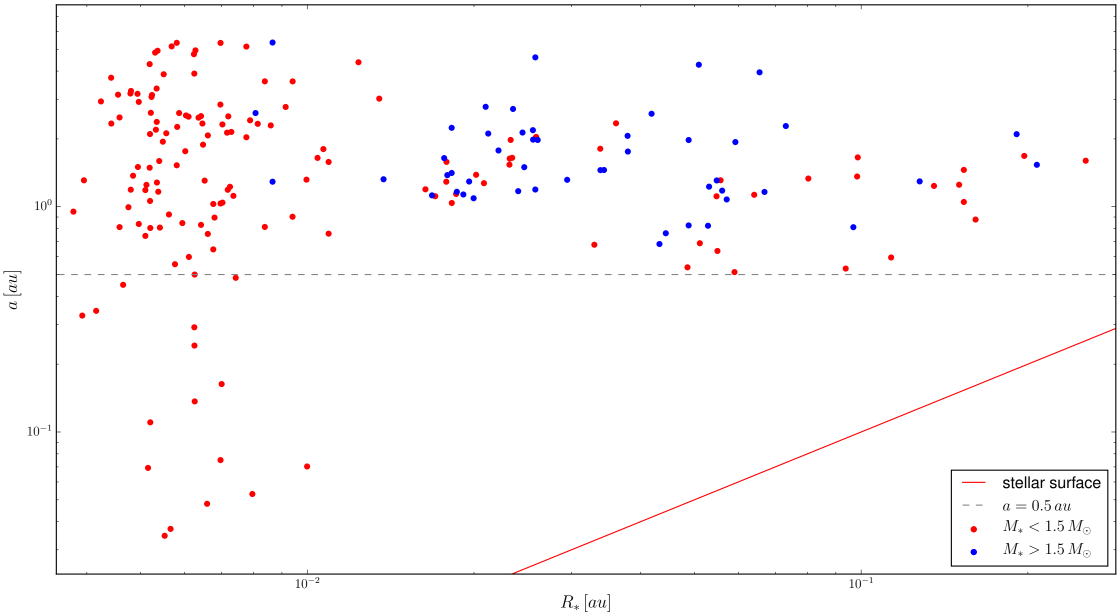

Figure 3.1: Observed semi-major axis distribution of $m > 1\,M_j$ planets as a function of stellar radius, each dot represents a planet. Black dots denote planets orbiting lower mass stars ($m < 1.5\,M_\odot$) and blue dots denote planets orbiting higher mass stars ($> 1.5\,M_\odot$). The red line represents the stellar surface, where $a = R_*$

The group of planets to the right in Fig. 3.1 corresponds to planets orbiting evolved stars. A striking difference between the evolved population and the MS population is visible, there are no detected giant planets orbiting evolved stars (mostly subgiants) below about 0.5 au. Another difference compared to the MS population is that the majority of detected evolved planet-hosting stars are significantly more massive (blue dots correspond to stars more massive than $1.5\,M_\odot$). We also see that there are no planets close to the stellar surface (the red line).

Two scenarios have been proposed to explain the observed lack of planets below 0.5 au around subgiant and giant stars. The first one is due to Currie (2009). Using Monte-Carlo simulations similar to those of Burkert & Ida (2007), he concludes that the observed deficiency of close in Jovian planets can be explained by stellar-mass-dependent protoplanetary disk lifetimes, which prevent giant planet migration in stars more massive than about $1.5\,M_{\odot}$. However, there is evidence based on analysis of selection effects (Lloyd 2011) and galactic kinematics (Schlaufman & Winn 2013) that the masses of the stars in the evolved sample have been greatly overestimated and that the mass distribution of evolved host stars should roughly the same as that of the MS host stars.

The second proposed scenario, and the one which is investigated in this thesis is that of tidal capture (Villaver & Livio 2009; Schlaufman & Winn 2013). Assuming that the stellar masses of evolved host stars are indeed overestimated, the two samples of host stars should differ only in age, thus providing a snapshot of the exoplanet population before and after tidal evolution. As mentioned in Sect. 2.1, a considerable percentage of the evolved host stars belong to the helium-burning branch. In fact, if we look closely at Fig. 2.2, we see that a star of given mass spends more time on the HB branch than on the ascending RGB, thus, it is more likely that the observed host stars with radii greater than the initial radius in the HB phase (about $10\,R_{\odot}$) belong to the HB branch. This implies that most planets in the evolved sample have undergone significant tidal evolution.

Chapter 4

Methods

4.1 Stellar evolution code

In order to simulate the post-main sequence evolution of stars, we need a stellar evolution code which outputs all the variables present in the tidal evolution equation (Eq. 2.37) as a function of time. There are many stellar evolution codes available and they work by solving fully coupled stellar structure and composition differential equations, a computationally intensive task. I have opted for a simpler code called SSE, described in the paper by (Hurley et al. 2000). The SSE code does not actually solve the system of stellar structure equations given in Section 2.1, it fits analytical formulae to the outputs of detailed numerical simulations (which do solve the stellar structure equations) to within 5% accuracy. At each time step, an analytical function is evaluated, The code is thus very fast and easy to use.

The code is written in FORTRAN, it takes several input parameters and it outputs a text file containing the values of stellar variables for each time step (which varies with time). The input parameters relevant to the project are

- stellar mass
- metallicity z
- maximum evolution time
- Reimers mass-loss coefficient

Stellar mass and the evolution were varied as needed while the metallicity and the mass-loss coefficient were kept constant throughout the project. I have chosen a metallicity value of $z = 0.02$, corresponding to Population I stars, because this is a fairly common value among planet-hosting stars. The Riemers mass-loss coefficient is a dimensionless number which describes how quickly the star loses mass (see Prialnik 2009, for definition), for it, I have chosen a value of 0.5, a common value in the literature and past papers on tidal evolution.

4.2 Tidal evolution code

Next step is to develop a code which reads the stellar evolution data, generates random planets with properties according to observed distributions, and finally, integrates equation 2.37. I have chosen to write the code in C++ programming language because it is flexible, computationally efficient, and a programming language I am most familiar with. The code (included in Appendix A) is well documented and flexible, it is designed to be easily adapted to a different stellar evolution code or different differential equations of orbital evolution.

The structure of the code is as follows. First, the stellar evolution data is read into several arrays and converted into units suitable for further computation. Then, it is interpolated using the *spline* interpolation algorithm, taken from Press et al. (1992). Interpolation is necessary because the time step necessary to integrate the orbital evolution equation is substantially smaller than that used in the stellar evolution code. Next, we have several functions which generate system variables consistent with observed planet frequency distributions. Finally, the input parameters are passed to function `EvolvePlanet` which integrates Eq. 2.37 using an adaptive Runge-Kutta integrator based on the Dormand-Prince algorithm (Dormand & Prince 1980). An adaptive integrator was used because significant orbital decay occurs on very small timescales which requires small time steps, using a fixed small time step would thus be computationally inefficient considering that for the most part of stellar evolution almost no significant change in orbit occurs.

In Section 2.2 I mentioned that an analytical solution to Eq. 2.30 exists if we assume constant stellar radius. I used that analytical solution to test the Runge-Kutta integrator and found that the relative error between the two solutions is completely negligible.

4.3 Population synthesis

In order to attempt to recreate the observations, we first have to generate initial parameters for the tidal evolution code. The goal is to evolve the MS population of planets, with properties matching observations as closely as possible, up to a specific target stellar radius (and thus also time), distributed according to the evolved host stars radius distribution. The initial parameters are the semi-major axis of the planet a at time $t = 0$, the stellar radius R_* at which the integration of Eq. 2.37 is stopped, the planet mass m, and finally the stellar mass M_*.

The semi-major axis sample is restricted to stars with radii less than $2\,R_\odot$, which corresponds to the MS sample of planets. The targeted stellar radius distribution is restricted to planets orbiting stars with radii greater than $2\,R_\odot$, since those stars are evolved off the main sequence. The two constraints valid for all parameters are to planets orbiting stars with $M_* > 1\,M_\odot$ (because there are virtually zero lower mass stars in the evolved sample), and $m > 1\,M_j$ (because there are no detected lower mass planets around evolved stars). As mentioned before, the whole sample is restricted to planets discovered with the RV method.

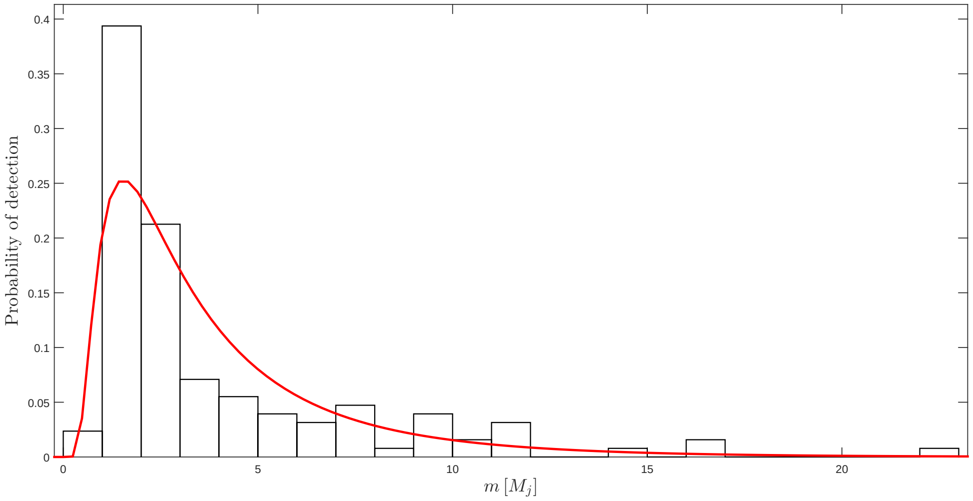

Figure 4.1: Inverse Gaussian fit to observed planet mass distribution around MS host stars

Now that we know the necessary initial parameters and the conditions imposed on each, we have to analyze the data from `exoplanets.org`, fit it with an appropriate probability distribution and develop a random number generator (RNG) which generates random numbers according to the fitted distributions.

Figure 4.1 shows the fitted distribution of planet mass. The distribution peaks around $2\,M_j$, and then declines towards higher masses.

The distribution of stellar radii for the evolved host stars ($R_* \gtrsim 2\,R_\odot$) is shown in Fig. 4.2. There is a peak in occurrence rate around $4\,R_\odot$, followed by a small dip before about $10\,R_\odot$ and a decline towards larger radii. The small dip is likely due to the fact that after about $10\,R_\odot$ (the minimum stellar radius for helium burning solar like stars), the sample also contains planets orbiting helium burning stars. The decline towards higher radii is due to the fact that evolved stars expand rapidly to peak radius on the red giant and helium burning branches and are thus less likely to be observed during that time.

Most of the detected stars in the sample are subgiants with radii of around $5\,R_\odot$. This means that majority of stars in the evolved sample are hydrogen-burning, however, those which have radii greater than about $10\,R_\odot$ are more likely to be helium burning as the helium burning branch has a less steep ascent to higher radii which implies higher probability of detection in a certain interval of R_* above $10\,R_\odot$. Thus, once a random number is drawn for the stellar radius, a probability is assigned for the star to be helium burning. It has the following form

$$\mathcal{P}_{HB}(R_*) = \begin{cases} 0 & , R_* < R_{HBmin} \\ 1 - t_{RGB}/t_{HB} & , R_* > R_{HBmin} \end{cases} \tag{4.1}$$

where $\mathcal{P}_{HB}$ is the probability for a star to be HB, R_{HBmin} is the initial radius of the star after the red-giant branch (before it starts increasing again), and t_{RGB}/t_{HB} is the

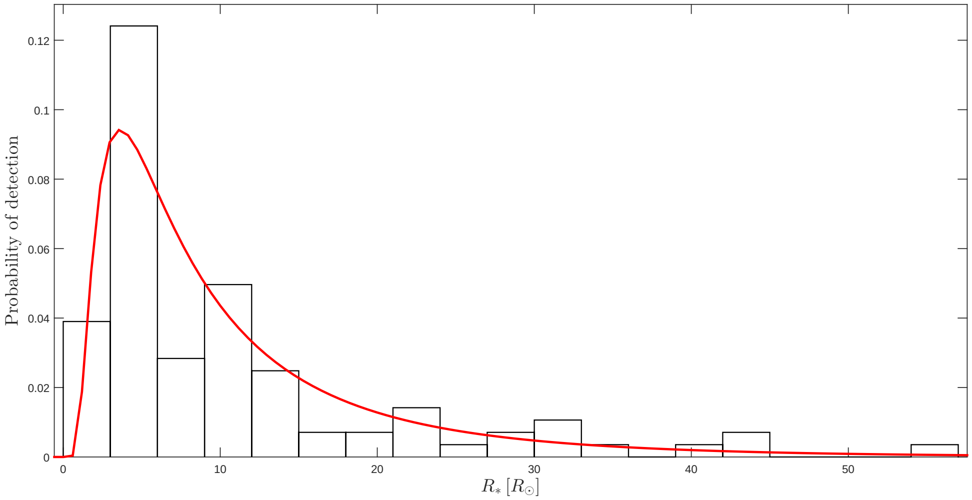

Figure 4.2: Inverse Gaussian fit to observed stellar radii distribution of evolved host stars

ratio between the time a star spends on the ascending red-giant branch and the time it spends on the HB branch within some small interval of R_*. The probability is then simply $1 - t_{RGB}/t_{HB}$ and for solar mass stars its value is around 60%. This is implemented in the code with a simple `if` statement, if the star is not HB the integration stops once the star reaches the targeted radius or the planet is engulfed . If it is HB the integration continues past the tip of the RGB and up to the targeted radius on the HB branch.

The stellar mass distribution of the main sequence stars is shown in Fig. 4.3. The occurrence rate of giant planets drops with stellar mass, and peaks around $1.1\,M_\odot$.

Finally, the most important distribution is the semi-major axis distribution of the MS sample of planets, shown on a logarithmic scale in Fig. 4.4. Looking at the figure, we see low occurrence rate up to about $a = 0.5\,au$, then a sharp increase in occurrence rate and a decline towards larger semi-major axes, most likely a result of observational bias. It is difficult to fit any probability function to this observed distribution. Cumming et al. (2008), based on the analysis of sample of planets with periods less than 2000 days (which corresponds to a semi-major axis of 3 au for a Keplerian orbit) observed during an RV survey. He suggests a power law fit to the logarithmic period distribution, of the form

$$\frac{dN}{d\ln P} = CP^{0.26} \tag{4.2}$$

where C is a normalization constant. An alternative description of the distribution mentioned in Cumming et al. (2008) is a flat distribution in $\ln P$ with a rapid increase in planet fraction at orbital periods of around 300 days (0.8 au in semi-major axis). When extrapolating the proposed to 5 au, I have found that a flat distribution in $\log a$ with a rapid increase around 0.7 au reproduces the observations most accurately. It is defined as

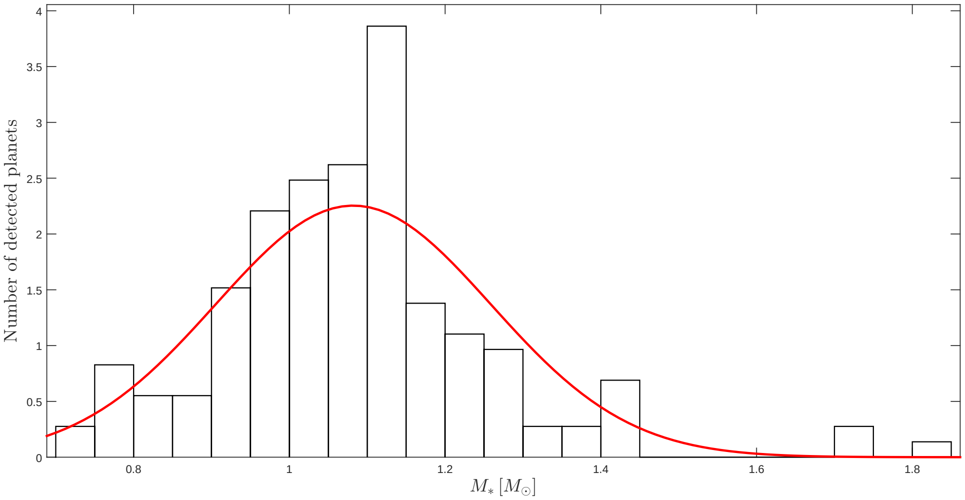

Figure 4.3: Gaussian fit to stellar mass distribution of planet-hosting MS stars

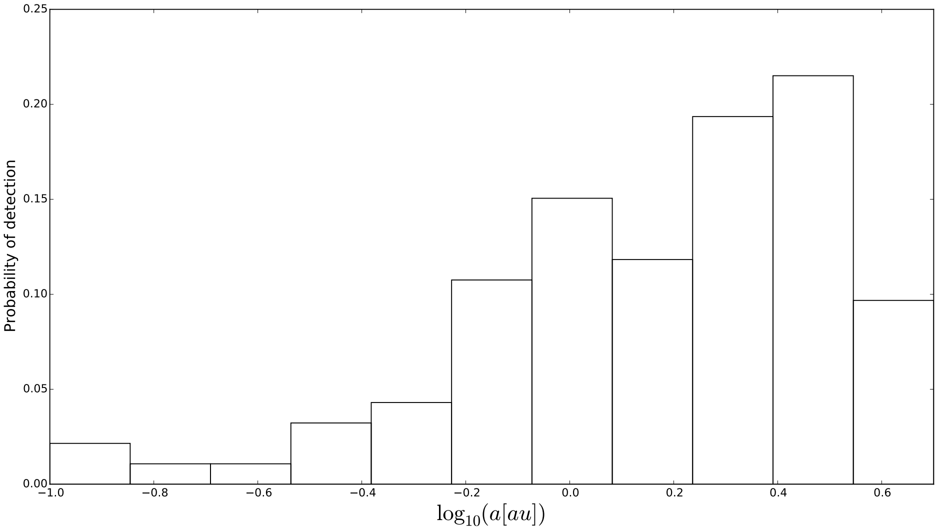

Figure 4.4: The observed semi-major axis distribution of planets orbiting MS stars. Plotted on a logscale.

follows

$$\frac{dN}{d\log a}(a) = \begin{cases} C & ,0.1 < a < 0.7 \\ \gamma C & ,0.7 < a < 5.0 \end{cases} \tag{4.3}$$

where where γ is 5, and C is an appropriate normalization constant.

To generate random numbers from a probability distribution I used the inverse transform method (see for ex. Riley et al. 2006), defined as follows. Given a general probability distribution $f(x)$, for $a < x < b$, the cumulative probability density function $F(x)$, where

$$F(x) = \int_a^b f(x)dx \tag{4.4}$$

is a monotonically increasing function bounded in the interval (0,1). Given a uniformly generated random number $u \in (0,1)$, there exists a number $F^{-1}(u)$ such that if u is distributed uniformly, $F^{-1}(u)$ is distributed according to $f(x)$. The inversion method assumes that we have a readily available uniform random number generator (which we do) and that there exists a closed form expression for the cumulative distribution $F(x)$ (if it doesn't we can always solve it numerically). Applying the inversion method to the piecewise probability distribution for the semi-major axis (Eq. 4.3) results in the following expression for the random number $F^{-1}(u)$

$$F^{-1}(u) = \begin{cases} 0.1\exp(u/C) & ,u < C\ln(0.7/0.1) \\ 0.7\exp[u - Cln(0.7/0.1)/5C] & ,u > C\ln(0.7/0.1) \end{cases} \tag{4.5}$$

where the normalization constant C is given by

$$C = \left(\ln\frac{0.7}{0.1} + 5\ln\frac{5}{0.7}\right) \tag{4.6}$$

The resulting histogram for 200 draws from the probability distribution for the semi-major axis is shown in Fig. 4.5. The probability distributions for the remaining three parameters are either Gaussian or Inverse Gaussian for which the same procedure was used.

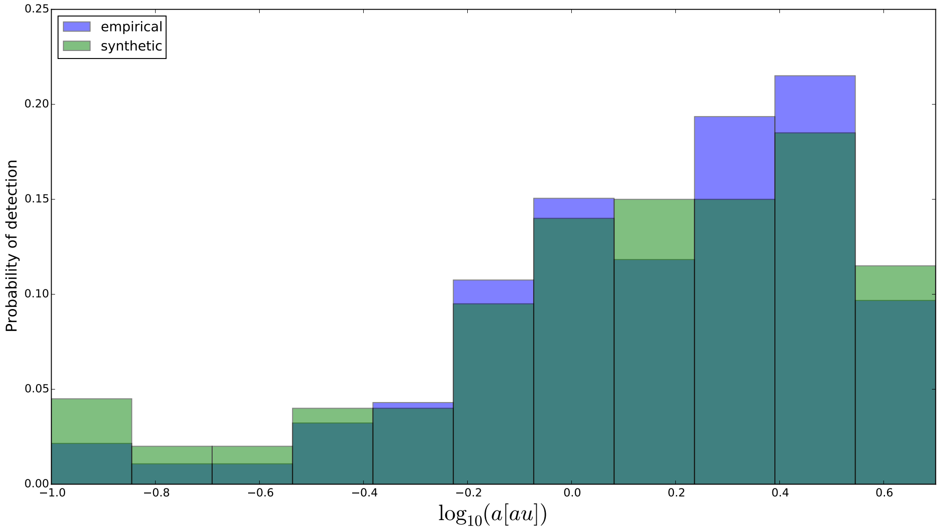

Figure 4.5: Comparison of randomly generated semi-major axis to the observed sample shown in Fig. 4.4

Chapter 5

Results and Discussion

5.1 Tidal evolution of a single planet

Before modeling the tidal evolution of many planets, it is useful to look how the semi-major axis of a single planet evolves in time. Figure 5.1 shows the solution of Eq. 2.37 for a Jupiter-mass planet, orbiting a $1 M_{\odot}$ star, plotted on the same graph is the stellar radius. We see that there are three possible outcomes. First, the blue trajectories show the planets which are engulfed by the star at some point on RGB, most significant decay occurs on a timescale of around 1 Myr. Then, there is a narrow range of a (green) in which planets start to decay rapidly at the tip of the RGB and then barely avoid engulfment as the star rapidly contracts. Finally, the black trajectories show planets which are safe from engulfment and actually increase their semi major axis, due to stellar mass-loss.

An natural question which then occurs is what is the minimum semi-major axis ($a_{critical}$) for which the planets avoid engulfment. The critical semi-major axis as a function of stellar mass is plotted in Fig. 5.2. We see a declining value of $a_{critical}$ with increasing stellar mass, a minimum around $2\,M_{\odot}$, and gradual increase towards high stellar masses. This plot is closely related to Fig. 2.2, showing the stellar radii for stars of various masses as a function of time. Decreasing $a_{critical}$ corresponds to a smaller RGB tip radius for more massive stars and a minimum at $2 M_{\odot}$ corresponds to a point at which stars start burning helium quietly (no helium flash). Increasing planet mass would effectively translate the curve upwards since more massive planets tend to be engulfed more easily.

These results do not agree with those of Villaver & Livio (2009) and Kunitomo et al. (2011) who predicted somewhat lower values of $a_{critical}$, the reason is likely due to the fact that they used different stellar evolution models and different values of the f parameter in Eq. 2.37. While the strength of the tidal forces may be overestimated, it is still not enough to reproduce the observed lack of close-in planets, as will be discussed in Section 5.2.

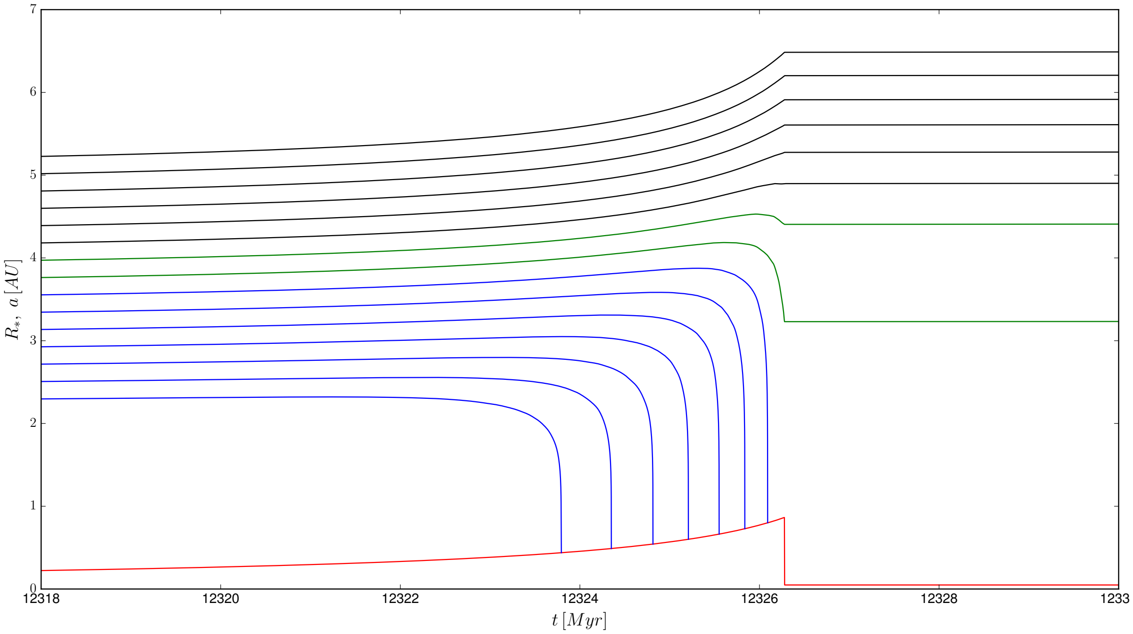

Figure 5.1: The evolution of the semi-major axis of a $1\,M_j$ planet orbiting a solar-mass star. The red line shows the stellar radius. Blue lines show the evolution of planets which are engulfed, green lines those which barely avoid engulfment, and black lines show the mass-loss dominated evolution of planets which are not in danger of being engulfed.

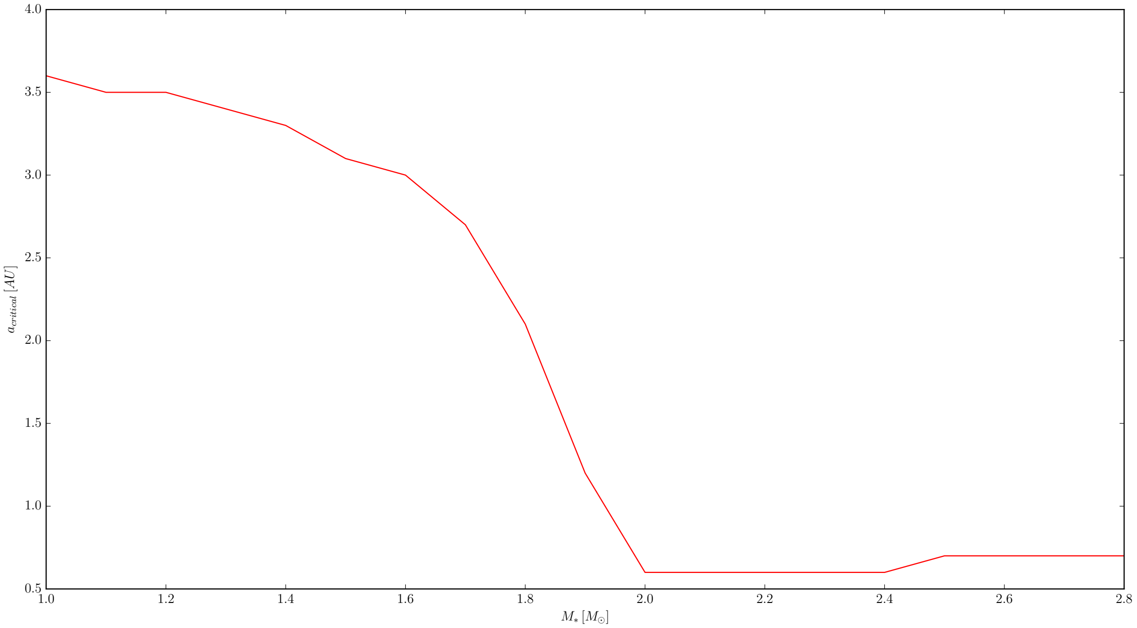

Figure 5.2: Critical semi-major axis below which a planet is engulfed as a function of stellar mass

5.2 Population synthesis results

The results of the population synthesis code for an initial population of 500 planets plotted on an $\log a$ vs. $\log R_*$ graph are shown in Fig. 5.3. Out of 500 initial planets, 374 have survived the tidal evolution up to their targeted stellar radius. It is useful to plot the output of the code for a smaller number of planets, comparable to that of observed evolved sample, which is comprised of 91 planets, this is shown in Fig. 5.4.

Both plots clearly show what one would naturally assume based on the nature of the tidal interactions. A region of some width parallel to the stellar radius is cleared out due to tidal interaction. Only among planets orbiting giants with large radii do we see considerable effects of tidal evolution, since the majority of those planets had to survive the tip of the red giant branch. Even when we reduce the sample size of initial planets such that the number of those which survive tidal evolution matches the number of observed planets around evolved stars, the differences between the two distributions are visible and it is clear that tidal engulfment alone cannot explain a sort of step function we see in the observational data.

It might be useful to see there is a set of initial parameters not consistent with observational data which change the plot significantly. If we assume that the observed mass distribution of evolved host stars is real, and indeed much higher than that of the MS sample, we can do a Gaussian fit of the stellar masses of the evolved population and use that distribution to generate the synthetic data. The resulting plot is shown in Fig. 5.5. No significant changes are visible.

We can also modify the probability for a star to be on the HB branch. Changing it from 60 to 90% results in Fig. 5.6. We see very poor agreement with observations, as almost all planets orbiting stars with radii greater then about $10\,R_\odot$ are now engulfed (no black dots at large solar radii). Lastly, we can arbitrarily increase the size of the step in the probability density function for the initial semi-major axis, that is changing the γ parameter in Eq. 4.3 from 5 to 12. This is plotted in Fig. 5.7. The resulting evolved population of planets looks quite similar to that of the observed data. Indeed, it seems that the only way to recreate the observational data is to use an initial semi-major axis distribution which is strongly biased towards large values of a, different from that of the main sequence sample.

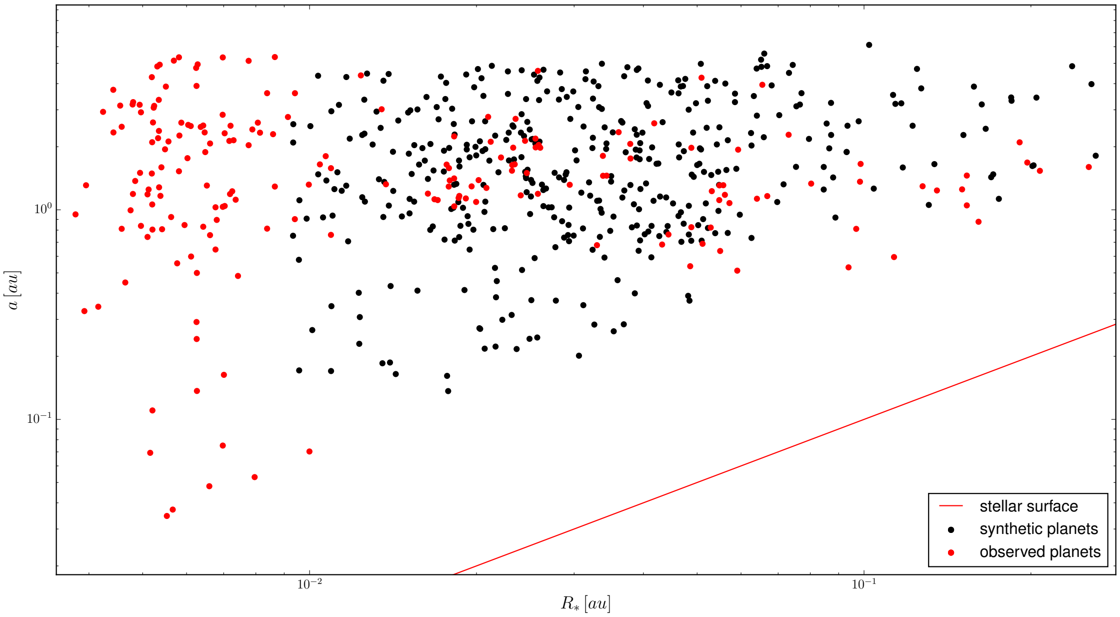

Figure 5.3: Output of the tidal evolution code for an initial population of 500 planets. Each dot represents a planet. Red dots are the observed planets from exoplanets.org database. Black dots are the synthetically generated planets which underwent tidal evolution. The red line represents the stellar surface, where $a = R_*$

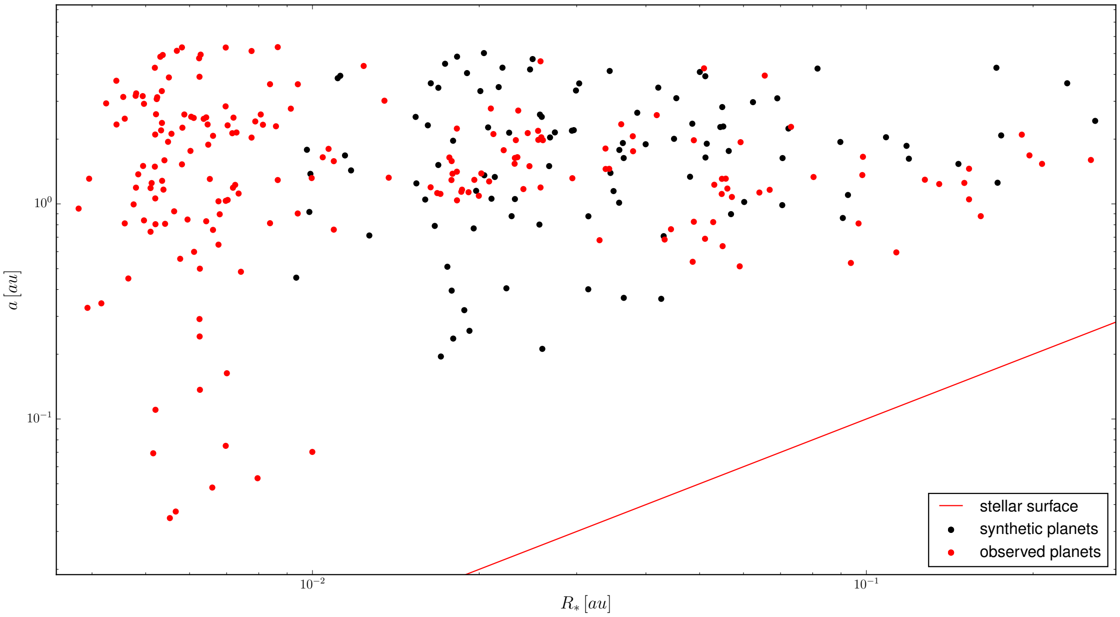

Figure 5.4: Plot showing the output of tidal evolution code for an initial sample of 130 planets, otherwise identical to Fig. 5.3

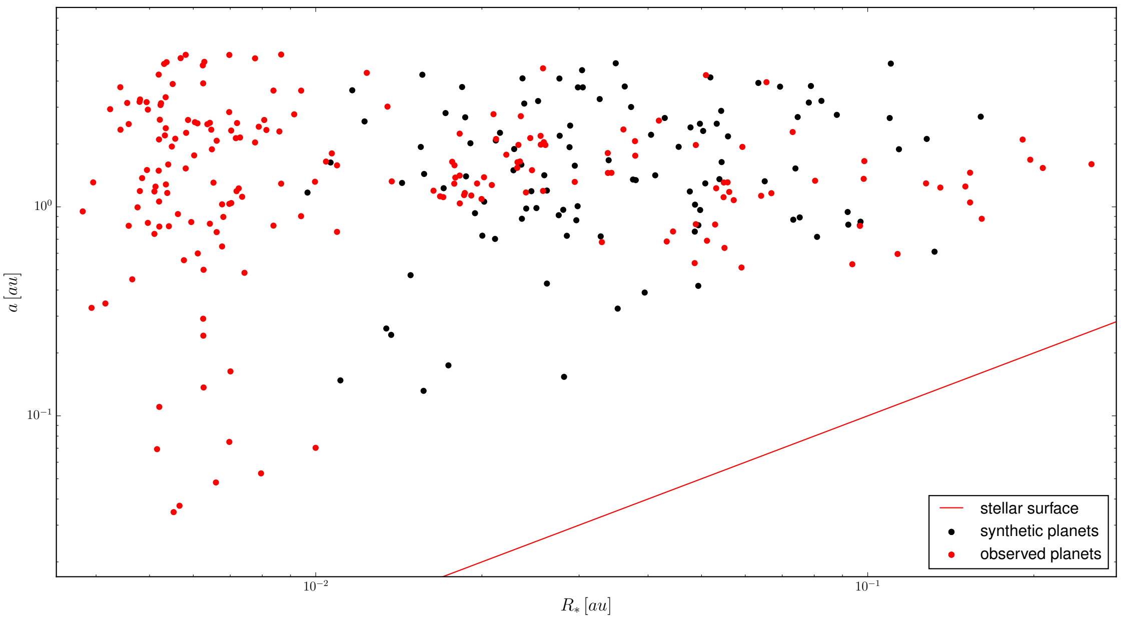

Figure 5.5: Plot showing the output of tidal evolution code for an initial sample of 130 planets with a different initial stellar mass distribution. Plot otherwise identical to Fig. 5.3

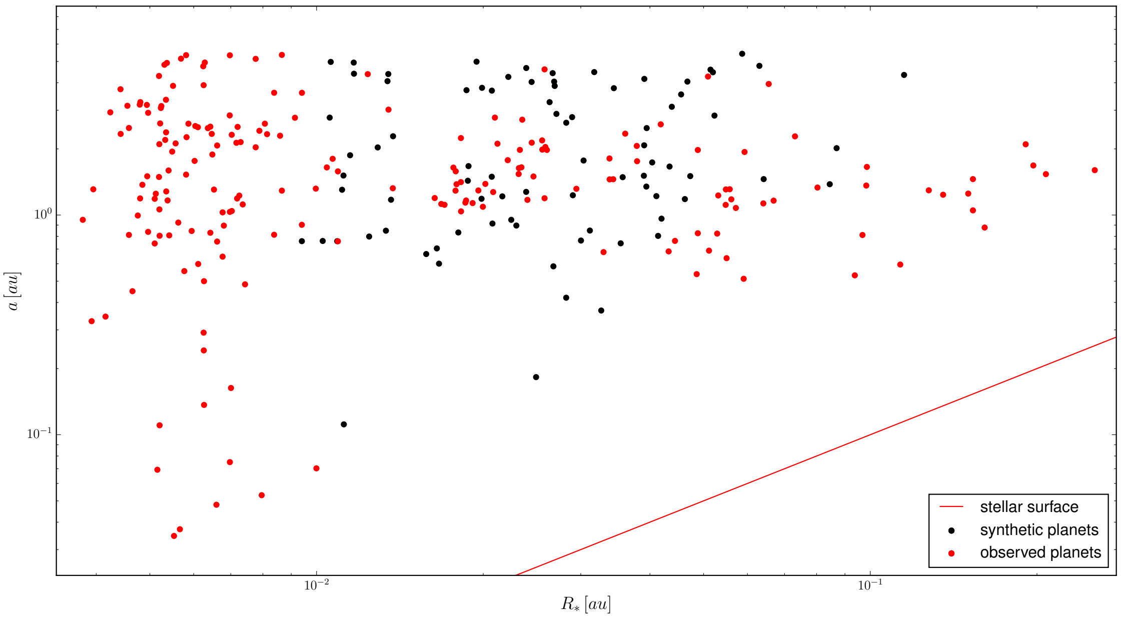

Figure 5.6: Plot showing the output of tidal evolution code for an initial sample of 130 planets with a high probability (90%) probability for a stars to be on the HB branch. Plot otherwise identical to Fig. 5.3

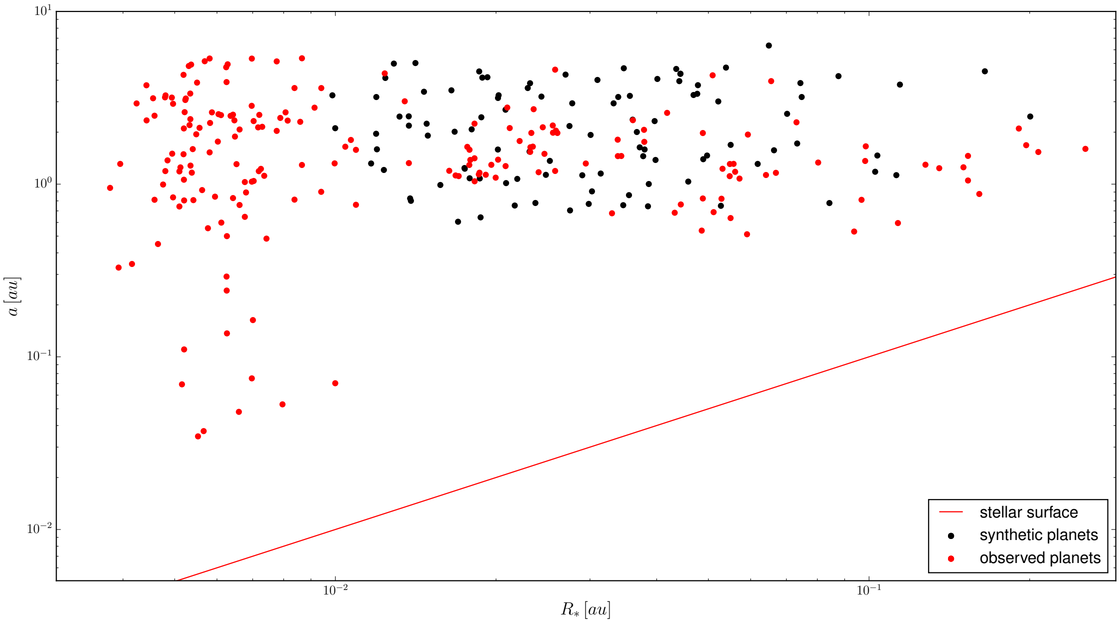

Figure 5.7: Plot showing the output of tidal evolution code for an initial sample of 130 planets with a different initial semi-major axis distribution. Plot otherwise identical to Fig. 5.3

5.3 Conclusions

In this project, I have developed a code which generates an initial sample of stars based on probability distributions derived from the main sequence planet population. I have assumed that the evolved planet population shares the same initial semi-major axis and stellar mass distribution as the MS sample in order to test if tidal engulfment can explain the observed data. This assumption was based on the work of Lloyd (2011) and Schlaufman & Winn (2013) who argue that the two observed planet populations share the same initial conditions and differ only in age of the host star. My results show that even if that assumption is true, the only remaining mechanism for clearing out inner planets cannot explain the observations. Thus I arrive at a conclusion that the observed lack of close-in planets around evolved stars could possibly be a combination of tidal engulfment, differences between the two observed populations (even if the stellar masses of the evolved population are overestimated, they need not be identical to that of the MS population) and perhaps also a consequence of a small statistical sample of planets.

Further observations of planets around giants combined with accurate determination of stellar masses and ages using the Gaia satellite should provide us with data needed to form a coherent picture of the interaction between evolving stars and planets.

References

Bowler, B. P., Johnson, J. A., Marcy, G. W., et al. 2010, ApJ, 709, 396

Burkert, A. & Ida, S. 2007, ApJ, 660, 845

Cassan, A., Kubas, D., Beaulieu, J.-P., et al. 2012, Nature, 481, 167

Cumming, A., Butler, R. P., Marcy, G. W., et al. 2008, PASP, 120, 531

Currie, T. 2009, ApJ, 694, L171

Dormand, J. & Prince, P. 1980, Journal of Computational and Applied Mathematics, 6, 19

Fressin, F., Torres, G., Charbonneau, D., et al. 2013, ApJ, 766, 81

Hurley, J. R., Pols, O. R., & Tout, C. A. 2000, MNRAS, 315, 543

Johnson, J. A., Butler, R. P., Marcy, G. W., et al. 2007, ApJ, 670, 833

Kennedy, G. M. & Kenyon, S. J. 2009, ApJ, 695, 1210

Kunitomo, M., Ikoma, M., Sato, B., Katsuta, Y., & Ida, S. 2011, ApJ, 737, 66

Landau, L. & Lifshitz, E. 1976, Mechanics, Butterworth Heinemann (Butterworth-Heinemann)

Lloyd, J. P. 2011, ApJ, 739, L49

Murray, C. D. & Dermott, S. F. 1999, Solar System Dynamics (Cambridge University Press)

Penev, K. & Sasselov, D. 2011, ApJ, 731, 67

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 1992, Numerical Recipes in C (2Nd Ed.): The Art of Scientific Computing (New York, NY, USA: Cambridge University Press)

Prialnik, D. 2009, An Introduction to the Theory of Stellar Structure and Evolution (Cambridge University Press; 2nd edition)

Reffert, S., Bergmann, C., Quirrenbach, A., Trifonov, T., & Künstler, A. 2015, A&A, 574, A116

Riley, K., Hobson, P., & Bence, S. 2006, Mathematical Methods for Physics and Engineering: A Comprehensive Guide (Cambridge University Press)

Rowe, J. F., Coughlin, J. L., Antoci, V., et al. 2015, ArXiv e-prints

Sato, B., Izumiura, H., Toyota, E., et al. 2007, ApJ, 661, 527

Schlaufman, K. C. & Winn, J. N. 2013, ApJ, 772, 143

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

Zahn, J.-P. 1977, A&A, 57, 383

Appendix A

The code

```
#include <iostream>
#include <iomanip>
#include <cmath>
#include <stdio.h>
#include <stdlib.h>
#include <iomanip>
#include <fstream>
#include <vector>
#include <time.h>
#include <cstring>

using namespace std;

#define   ZERO       1.0E-10
#define   pi         3.14159265358979323846264338327950288419 68L
#define   G          4.*pi*pi

typedef vector<double> vec;

ifstream input;
ofstream output;

/*Given arrays x[1..n] and y[1..n] containing a tabulated function, i.
   e., yi = f(xi), with
x1 <x2 < :: : < xN, and given values yp1 and ypn for the first
   derivative of the interpolating
function at points 1 and n, respectively, this routine returns an
   array y2[1..n] that contains
the second derivatives of the interpolating function at the tabulated
   points xi. If yp1 and/or
```

```
ypn are equal to 1 1030 or larger, the routine is signaled to set the
    corresponding boundary
condition for a natural spline, with zero second derivative on that
   boundary.*/
void spline(vec x, vec y, int n, double yp1, double yp2, vec &y2)
{
    int          i,k;
    double       p, qn, sig, un, *u;

    u = new(nothrow) double [n];
    if(!u) {
        printf("\n\nError in function spline():");
        printf("\nNot enough memory for u[%d]\n",n);
        exit(1);
    }

   if(yp1 > INFINITY) { y2[0] = u[0] = 0.0;}
   else {
      y2[0] = -0.5;
      u[0]  = (3.0/(x[1] - x[0])) * ((y[1] - y[0])/(x[1] - x[0]) - yp1
         );
   }

   for(i = 1; i < (n - 1); i++) {
      sig    = (x[i] - x[i - 1])/(x[i + 1] - x[i - 1]);
      p      = sig * y2[i - 1] + 2.0;
      y2[i] = (sig - 1.0)/p;
      u[i]  = (y[i + 1] - y[i])/(x[i + 1] - x[i]) - (y[i] - y[i - 1])
         /(x[i] - x[i - 1]);
      u[i]  = (6.0 * u[i]/(x[i + 1] - x[i - 1]) - sig*u[i - 1])/p;
   }

   if(yp2 > INFINITY)  qn = un = ZERO;
   else {
      qn = 0.5;
      un = (3.0/(x[n - 1] - x[n - 2])) * (yp2 - (y[n - 1] - y[n - 2])
         /(x[n - 1] - x[n - 2]));
   }
   y2[n - 1] = (un - qn * u[n - 2])/(qn * y2[n - 2] + 1.0);

   for(k = n - 2; k >= 0; k--) {
      y2[k] = y2[k]*y2[k+1]+u[k];
   }
   delete [] u;
}

/*Given the arrays xa[1..n] and ya[1..n], which tabulate a function (
   with the xai's in order),
and given the array y2a[1..n], which is the output from spline above,
   and given a value of
x, this routine returns a cubic-spline interpolated value y.*/
void splint(vec xa, vec ya, vec y2a, int n, double x, double *y)
{
   int       klo,khi,k;
   double    h,b,a;

   klo = 1;
   khi = n - 1;
   while((khi - klo) > 1) {   // binary search
      k = (khi + klo) >> 1;
      if(xa[k] > x)    khi = k;
      else             klo = k;
   }
   h = xa[khi] - xa[klo];
   if(fabs(h) < ZERO) {
      printf("\n\n Error in function splint(): ");
      printf("\n The difference h = %4.1E -- too small\n",h);
      exit(1);
   }
   a  = (xa[khi] - x)/h;
   b  = (x - xa[klo])/h;
   *y =   a * ya[klo] + b * ya[khi] + ((a * a * a - a) * y2a[klo] + (b
       * b * b - b) * y2a[khi]) * (h * h)/6.0;
}
/*
    Based on the equations described in the Zahn paper
*/
double dadt(double a, double R, double M, double Mdot, double L,
   double Menv, double Renv, double m)
{
    double tau_d = pow(Menv * pow(R-Renv,2) / (3*L), 1/3.);
    double q = m / M;
    return -1E06 * Menv * q/(tau_d * M) * (1 + q) * pow(R/a, 8) * a  -
        Mdot * a / M;
}

/*
    Runge kutta differential equation solver
    implements Dormand-Prince method
    Inputs:
        &y - dependent variable
         t - time
        &h - time step
         epsilon - error tolerance
        R - parameter
    Outputs:
        y - new value of y
        h - new time step
*/
void rk45_adaptive(double &y, double t, double &h, double epsilon,
   double R,
                    double M, double Mdot, double L, double Menv,
                       double Renv, double m)
{

    vec k(7);
    double z, h_new;
    double h_min = 0.001; //Myr
    double h_max = 0.1;     //Myr

    k[0] = h * dadt(y, R, M, Mdot, L, Menv, Renv, m);

    k[1] = h * dadt(y + (1/5.)*k[0], R, M, Mdot, L, Menv, Renv, m);

    k[2] = h * dadt(y + (3/40.)*k[0] + (9/40.)*k[1], R, M, Mdot, L,
       Menv, Renv, m);

    k[3] = h * dadt(y + (44/45.)*k[0] - (56/15.)*k[1] + (32/9.)*k[2],
       R, M, Mdot, L, Menv, Renv, m);

    k[4] = h * dadt(y + (19372/6516.)*k[0] - (25360/2187.)*k[1] +
       (64448/6561.)*k[2] - (212/729.)*k[3], R, M, Mdot, L, Menv, Renv
       , m);

    k[5] = h * dadt(y + (9017/3168.)*k[0] - (355/33.)*k[1] -
       (46732/5247.)*k[2] + (49/176.)*k[3] - (5103/18656.)*k[4], R, M,
        Mdot, L, Menv, Renv, m);

    k[6] = h * dadt(y + (35/384.)*k[0] + (500/1113.)*k[2] + (125/192.)
       *k[3] - (2187/6784.)*k[4] + (11/84.)*k[5], R, M, Mdot, L, Menv,
        Renv, m);

    //calculation by Runge Kutta method of order 5
    z = y + (5179/57600.)*k[0] + (7571/16695.)*k[2] + (393/640.)*k[3]
       - (92097/339200.)*k[4] + (187/2100.)*k[5] + (1/40.)*k[6];
    //calculation by Runge Kutta method of order 4
    y += (35/384.)*k[0] + (500/1113.)*k[2] + (125/192.)*k[3] -
       (2187/6784.)*k[4] + (11/84.)*k[5];
    //err = |z-y_new|
    double err = abs((71/57600.)*k[0] - (71/16695.)*k[2] + (71/1920.)*
       k[3] - (17253/339200.)*k[4] + (22/525.)*k[5] -(1/40.)*k[6]);

    double s = pow(epsilon*h / (2*err), 1/5.);

    h_new = h*s;

    if(h_new < h_min)
        h_new = h_min;

    if(h_new > h_max)
        h_new = h_max;
    //in the case that the new time step is much less than the old,
       y_new is updating using the new value immediately
    if(h_new < h/2)
        rk45_adaptive(y, t, h_new, epsilon, R, M, Mdot, L, Menv, Renv,
           m);
    //to prevent infinities
    if(isnan(h_new) == true)
        h_new = 0.1;
    //update h
    h = h_new;
}

/*
This function reads the data from the stellar evolution code into two
   different arrays for each of the variables,
 one for the giant branch and the other for asymptotic giant branch.
   The reason for this is that the radius of the
 star is not a smooth function of time at the point when the star
   leaves the RGB phase. This can cause interpolation
 problems, thus, to increase accuracy, we split the variable arrays
   into two parts.
 */
void LoadStellarData(string filename, vec &t_GB, vec &R_GB, vec &t_AGB
   , vec &R_AGB,
                     vec &M_GB, vec &M_AGB, vec &L_GB, vec &L_AGB, vec
                        &Renv_GB,
                     vec &Renv_AGB, vec &Menv_GB, vec &Menv_AGB)
```

```
{
    //load file
    input.open (filename.c_str());

    //read data from file
    double dummy, t, L, R, M,Renv, Menv;
    double evo_phase = 3;
    int i = 0;

    while(input.good())
    {
        /*
        For G to be equal to 4pi^2, all variables must be expressed in
            units solar masses, years, and astronomical units.
        The stellar evolution code outputs time in Myr, radius in
           log10(R_solar), luminosity in log10(L_solar)
        and mass in solar masses, the following conversion factors are
            used:
            1 MGyr = 1E06 yr
            1 solar luminosity = 2.7*1E-04 Msun*AU^2/yr^3
            1 solar radii = 0.004649 AU
        */
        if(evo_phase < 4) {
            input>>t>>evo_phase>>dummy>>M>>L>>R>>dummy>>dummy>>Menv>>
               dummy>>Renv;
            t_GB.push_back(t);
            R_GB.push_back(pow(10, R) * 0.004649);
            M_GB.push_back(M);
            L_GB.push_back(pow(10, L) * 2.7*1E-04);
            Menv_GB.push_back(Menv);
            Renv_GB.push_back(Renv * 0.004649);
        }

        else {
            //remove last element from GB arrays and transfer it to
               AGB arrays
            if(i == 0) {
                t_AGB.push_back(t_GB[t_GB.size()-1]);
                R_AGB.push_back(R_GB[R_GB.size()-1]);
                M_AGB.push_back(M_GB[M_GB.size()-1]);
                L_AGB.push_back(L_GB[L_GB.size()-1]);
                Menv_AGB.push_back(Menv_GB[M_GB.size()-1]);
                Renv_AGB.push_back(Renv_GB[Renv_GB.size()-1]);
                t_GB.pop_back();
                R_GB.pop_back();
                M_GB.pop_back();
                L_GB.pop_back();
                Menv_GB.pop_back();
                Renv_GB.pop_back();
            }

            i++;
            //continue reading data for AGB phase
            input>>t>>evo_phase>>dummy>>M>>L>>R>>dummy>>dummy>>Menv>>
               dummy>>Renv;
            t_AGB.push_back(t);
            R_AGB.push_back(pow(10, R) * 0.004649);
            M_AGB.push_back(M);
            L_AGB.push_back(pow(10, L) * 2.7*1E-04);
            Menv_AGB.push_back(Menv);
            Renv_AGB.push_back(Renv * 0.004649);
        }
    }

    for(i = 0; i < 2; i++) {
        t_AGB.pop_back();
        R_AGB.pop_back();
        M_AGB.pop_back();
        L_AGB.pop_back();
        Menv_AGB.pop_back();
        Renv_AGB.pop_back();
    }

    input.close();
}

//returns either success (true) or failure (false), p is probability
   of success
bool BernoulliRNG( double p )
{
    return ((double) rand() / (RAND_MAX)) < p;
}

//the piecewise probability function for the semi-major axis
   distribution. Outputs numbers in the range (a,b) with a certain
   probability,
//that probability increases by a factor of gamma in the range (b,c)
double FlatRNG(double a, double b, double c, double gamma)
{
    double r, N, output;
    N = 1 / (log(b/a) + gamma * log(c/b));

    r = ((double) rand() / (RAND_MAX));
    if(r < N * log(b / a))
        output = a * exp(r / N);
    else
        output =  b * exp((r - N * log(b / a) ) / (gamma*N));

    return output;
}

//Gaussian random number generator
double GaussianRNG( double mu = 1.12488, double sigma = 0.473735, bool
    selection = true) //for target_r distribution
{
    double p, p1, p2, output;

    if(selection == false) {
        do {
            p1 = ((double) rand() / (RAND_MAX))*2. - 1.;
            p2 = ((double) rand() / (RAND_MAX))*2. - 1.;
            p = p1 * p1 + p2 * p2;
        } while ( p >= 1. );
        output = mu + sigma * p1 * sqrt( -2. * log( p ) / p );
    }

    else {
        do {
            do {
                p1 = ((double) rand() / (RAND_MAX))*2. - 1.;
                p2 = ((double) rand() / (RAND_MAX))*2. - 1.;
                p = p1 * p1 + p2 * p2;
            } while ( p >= 1. );
            output = mu + sigma * p1 * sqrt( -2. * log( p ) / p );
        } while(output < 1 || 3 < output);
    }

    return output;
}

//Inverse Gaussian random number generator
double inverseGaussianRNG(double mu, double lambda, double lower_limit
   , double upper_limit)
{
    double output, r, y, x, test;
    do {
            r = GaussianRNG(0,1,false);
            y = r*r;
            x = mu + (mu*mu*y)/(2*lambda) - (mu/(2*lambda)) * sqrt(4*mu
               *lambda*y + mu*mu*y*y);
            test = ((double) rand() / (RAND_MAX));
            if (test <= (mu)/(mu + x))
               output = x;
            else
               output = (mu*mu)/x;
        } while(output < lower_limit || upper_limit < output);

      return output;
}

//Function which takes initial parameters generated by the statistical
    model and evolves a planet up to a target radius
void EvolvePlanet(double a_initial, double m, double target_R, string
   filename)
{
    //memory allocation
    vec t_GB, t_AGB, R_GB, R_AGB, M_GB, M_AGB, MdotGB, MdotAGB, L_GB,
       L_AGB, Renv_GB, Renv_AGB, Menv_GB, Menv_AGB;

    //load stellar evolution data
    LoadStellarData(filename, t_GB, R_GB, t_AGB, R_AGB, M_GB, M_AGB,
       L_GB, L_AGB, Renv_GB, Renv_AGB, Menv_GB, Menv_AGB);

    int GB_sz = t_GB.size();
    int AGB_sz = t_AGB.size();

    //calculate approx. mass loss rate dM/dt based on the stellar
       evolution code data
    MdotGB.resize(GB_sz);
    MdotAGB.resize(AGB_sz);

    MdotGB[0] = (M_GB[1] - M_GB[0]) / (t_GB[1] - t_GB[0]);

    MdotGB[GB_sz-1] = (M_GB[GB_sz-1] - M_GB[GB_sz-2]) / (t_GB[GB_sz-1]
        - t_GB[GB_sz-2]);

    for(int i = 1; i < GB_sz-1; i++)
        MdotGB[i] = (M_GB[i+1] - M_GB[i-1]) / (t_GB[i+1] - t_GB[i-1]);

    //AGB
```

```
MdotAGB[0] = (M_AGB[1] - M_AGB[0]) / (t_AGB[1] - t_AGB[0]);

MdotAGB[AGB_sz-1] = (M_AGB[AGB_sz-1] - M_AGB[AGB_sz-2]) / (t_AGB[
    AGB_sz-1] - t_AGB[AGB_sz-2]);

for(int i = 1; i < AGB_sz-1; i++)
     MdotAGB[i] = (M_AGB[i+1] - M_AGB[i-1]) / (t_AGB[i+1] - t_AGB[i
         -1]);

/*spline interpolation of stellar evolution data*/
double deriv_1 = 0;  //value of derivative at first point
double deriv_n = 0;  //value of derivative at last point

//arrays for storing second derivatives to be used in splint
    function
vec deriv_M_GB(GB_sz), deriv_M_AGB(AGB_sz), deriv_Mdot_GB(GB_sz),
    deriv_Mdot_AGB(AGB_sz),
deriv_R_GB(GB_sz), deriv_R_AGB(AGB_sz), deriv_L_GB(GB_sz),
    deriv_L_AGB(AGB_sz),
deriv_Menv_GB(GB_sz), deriv_Menv_AGB(AGB_sz), deriv_Renv_GB(GB_sz)
    , deriv_Renv_AGB(AGB_sz);

//calculate second derivatives using spline function
spline(t_GB, R_GB, GB_sz, deriv_1, deriv_n, deriv_R_GB);
spline(t_AGB, R_AGB, AGB_sz, deriv_1, deriv_n, deriv_R_AGB);
spline(t_GB, M_GB, GB_sz, deriv_1, deriv_n, deriv_M_GB);
spline(t_AGB, M_AGB, AGB_sz, deriv_1, deriv_n, deriv_M_AGB);
spline(t_GB, MdotGB, GB_sz, deriv_1, deriv_n, deriv_Mdot_GB);
spline(t_AGB, MdotAGB, AGB_sz, deriv_1, deriv_n, deriv_Mdot_AGB);
spline(t_GB, L_GB, GB_sz, deriv_1, deriv_n, deriv_L_GB);
spline(t_AGB, L_AGB, AGB_sz, deriv_1, deriv_n, deriv_L_AGB);
spline(t_GB, Menv_GB, GB_sz, deriv_1, deriv_n, deriv_Menv_GB);
spline(t_AGB, Menv_AGB, AGB_sz, deriv_1, deriv_n, deriv_Menv_AGB);
spline(t_GB, Renv_GB, GB_sz, deriv_1, deriv_n, deriv_Renv_GB);
spline(t_AGB, Renv_AGB, AGB_sz, deriv_1, deriv_n, deriv_Renv_AGB);

//simulation parameters
double t,a, R, M, Mdot, L, Menv, Renv;
double h = 50;  //initial step size in Myr
double tswitch = t_GB[GB_sz-1];
double tmax = t_AGB[AGB_sz-1];
bool planet_engulfed, HB;
//initial values
t = 0;
R = 0;
```

```
a = a_initial;
planet_engulfed = false;

//estimate the probability that the star with observed target
   radius is in HB phase
if(target_R < R_AGB[0])
    HB = false;  //impossible for it to be a HB star since minimum
        HB radius is R_AGB[0]
else
    HB = BernoulliRNG(0.6);  //slightly biased towards it being a
       HB star

while(t < tmax)
{
    if (a < R || a > 10) {
        planet_engulfed = true;
        cout<<”Planet engulfed”<<endl;
        break;
    }

    if(R > target_R && HB == false) {
        cout<<”R* greater than target radius”<<endl;
        break;
    }


    else if(R > target_R && HB == true && t > t_AGB[0]) {
        cout<<”R* greater than target radius”<<endl;
        break;
    }


    if(t <= tswitch) {
        splint(t_GB, R_GB, deriv_R_GB, GB_sz, t, &R);
        splint(t_GB, L_GB, deriv_L_GB, GB_sz, t, &L);
        splint(t_GB, Menv_GB, deriv_Menv_GB, GB_sz, t, &Menv);
        splint(t_GB, Renv_GB, deriv_Renv_GB, GB_sz, t, &Renv);
        splint(t_GB, M_GB, deriv_M_GB, GB_sz, t, &M);
        splint(t_GB, MdotGB, deriv_Mdot_GB, GB_sz, t, &Mdot);
    }

    else {
        splint(t_AGB, R_AGB, deriv_R_AGB, AGB_sz, t, &R);
        splint(t_AGB, L_AGB, deriv_L_AGB, AGB_sz, t, &L);
        splint(t_AGB, Menv_AGB, deriv_Menv_AGB, AGB_sz, t, &Menv);
```

```
            splint(t_AGB, Renv_AGB, deriv_Renv_AGB, AGB_sz, t, &Renv);
            splint(t_AGB, M_AGB, deriv_M_AGB, AGB_sz, t, &M);
            splint(t_AGB, MdotAGB, deriv_Mdot_AGB, AGB_sz, t, &Mdot);
        }
        //in case interpolation errors give near zero negative values
           of necessarily possitive variables
        if(Menv < 0)
            Menv = 1E-10;

        if(Renv < 0)
            Renv = 1E-10;

        if(L < 0)
            L = 1E-10;

        //update diff. eq variable
        rk45_adaptive(a, t, h, 10E-10, R, M, Mdot, L, Menv, Renv, m);

        //output<<setprecision(13)<<t<<setw(30)<<setprecision(13)<<a<<
           setw(30)<<setprecision(13)<<R<<endl;

        t += h;
    }

    //output to file if planet survived
    if(planet_engulfed == false)
        output<<setprecision(8)<<a<<setw(15)<<setw(15)<<setprecision
           (8)<<R<<endl;
}

int main()
{
    //RNG seed
    srand ( time(NULL) );

    //output file
    output.open (”ispis.txt”);

    double a, M, m, target_R, a_final, R_final;
    int N = 500; //number of synthetic planets

    //evolve synthetic planets
    for(int i = 1; i < N; i++)
    {
        //generate simulation parameters distributed according to
           empirical data
        M = GaussianRNG(1.0781,0.176642, true);
        a = FlatRNG(0.1,0.7,5,5);
        m = inverseGaussian(3.93317, 5.46111, 1, 20)* 0.00095458;//
           conversion to solar masses
        target_R = inverseGaussian(12, 12.0146, 2, 200) * 0.004649; //
           conversion to AU

        //choose correct stellar evolution output file named according
            to stellar mass in M_sun. Example: 1.1, 2.6,...
        string filename;
        M = roundf(M * 10) / 10;
        char buffer[32];
        snprintf(buffer, sizeof(buffer), ”evo_data/%.1f.dat”, M);
        filename = buffer;
        cout<<filename<<endl;

        //evolve planet
        EvolvePlanet(a, m, target_R, filename);

    }

    output.close();
}
```