Mémoire

Auteur : Guyot, Nathan
Promoteur(s) : Van Grootel, Valérie
Faculté : Faculté des Sciences
Diplôme : Master en sciences spatiales, à finalité approfondie
Année académique : 2023-2024
URI/URL : http://hdl.handle.net/2268.2/19850

Université de Liège

The pulsation spectra of hot subdwarf stars in the core-He burning stage

Masster In Space Sciences

Research focus

Author:
Nathan Guyot

Supervisor:
Valérie Van Grootel

Reading committee:
Martin Farnir
Denis Grodent
Michaël De Becker

Academic year 2023-2024

Acknowledgement

I would like to thank my supervisor, Valérie Van Grootel, without whom this master thesis would not exist. Her ability to transfer her insights and interest of asteroseismology has found a host and re-ignited a thirst for knowledge. I additionally want to highlight her availability, I cannot recall a single time when my mails and questions did not find a fast and clear answer. The journey to this manuscript was thrilling, without a doubt thanks to her wonderful supervision.

I thank as well the member of the reading committee, Martin Farnir, Denis Grodent and Michaël De Becker, for the time they will devote to reading this master thesis. I acknowledge as well Marc-Antoine Dupret, for his discussions and insights that greatly helped.

A special thanks go to Brandon, who I have known for many years, and whom I consider in all aspects as a brother. His unwavering support, through thick and thin, is a fundation of the person I am today. I will never forget the countless sleepless nights of work you have helped me get through, and the amount of care and support you provide daily. My gratitude also goes to Viridias, whom even from afar, provided years of moral support, friendship, and laughs, all of which have been key to keep going forward.

Finally, I would like to thank my family, they always believed in me, and continue to do so. I am forever grateful for them, and consider myself incredibly lucky to be blessed with such caring relatives. None of them have ever shyed away from a good phone call despite the distance, and have contributed heavily to keeping my spirits up.

Abstract

Studying pulsation spectra through asteroseismology allows probing the inner structure of stars. Core helium burning stars, specifically subdwarf type B stars, have been observed to harbour pressure and gravity mode pulsations, allowing respectively to probe the envelope and core of such stars. In particular, observations show a dichotomy in gravity modes pulsation spectra, with sometimes structures called trapped modes, which induce variable period spacings between observed periods, and other times no trapped modes at all, with a rather smooth pulsation spectra showing more or less constant spacings instead.

In this master thesis, we model subdwarf type B stars with both 4th generation static models and evolutionary models, using the STELUM and PULSE codes. Through this, we aim to gain insights on the influence of core helium burning on the pulsation spectra. We highlight as well the influence of the chemical and thermal structures on the behavior of pulsation spectra, in particular the mass of the core and envelope, as well as the thermal gradients prescriptions. A clear distinction is made between evolutionary and static models. The latter are studied first, and we discuss the origin of trapped modes from chemical transitions and temperature gradients in such models. In evolutionary models, we focus on the overshooting and semi-convection phenomena, which are not found in static models, and study their impact on the chemical and thermal structure of the star, as well as on the pulsation spectra.

This master thesis gives the theoretical basis of pulsation spectra computed from current available stellar models of subdwarf B stars, now to be compared in detail with observations of such stars, in particular those observed by the Kepler and TESS satellites.

Contents

Introduction

As our tools to look towards space developed greatly over the years, we have yet to gaze beyond the surface of a star. Despite this shortcoming, we have been able to develop a stellar structure theory, with, as a precursor, Eddington in 1926 with *The Internal Constitution of the Stars*. Already then, Eddington realized that even if his theories were to be true, there was no simple way of confirming them, and for quite some time, the discussion of the internal physics and structure of the stars remained mainly a theoretical exercise. While we could already get some constraints on stellar structure using exterior observables such as the effective temperature, or even in the particular case of eclipsing binaries, masses and radii, the advent of asteroseismology has opened a way to experimentally verify what was until then, only theories.

Variable objects in the sky have been noticed quite a long time ago, with the first ever variable star, Algol, an eclipsing binary, already remarked about 3200 years ago in Egypt. Despite this early discovery, this is only far later that the very concept of variable star has been introduced, with in the 17th century, the first observation of periodic changes in the luminosity of stars were made. This leads us to the concept of *extrinsic* and *intrinsic* variables, in which the former's luminosity varies from external conditions, that is for example, an eclipsing binary or stellar winds, while the latter finds the origin of its variability through the compression and dilatation of different parts of the star in precise geometrical patterns, so-called *stellar oscillations* or *pulsations*[1]. It is precisely in understanding the relation linking on the one hand the structure and quantities defining what an intrinsic variable star is, and on the other hand its pulsations' patterns and frequencies, that motivates asteroseismology.

The origin of asteroseismology is found here, on Earth, through seismology. Is it upon realizing that the Earth has vibration modes and seismic waves which could be used to probe its internal structure that we began to apply the same ideology to stars. Following this, the theory of stellar pulsation was applied to the Sun and succeeded in constraining not only its interior but even its internal rotation. While asteroseismology began using ground telescopes, new data from recent spatial missions such as CoRot (2006-2012) and Kepler (2009-2018) are now allowing us to expand the range of known pulsating stars and to detect new modes of lower amplitude from stars already known to pulsate.

Despite those upgraded capabilities, stellar interiors and their complex physics remain a challenge. If the foundations of our numerical models are now anchored and strongly tested, we can hardly call them complete. Effects such as tidal forces, magnetic fields and convection, the latter which shall be investigated in this master thesis, are hardly understood in detail with respect to their influence on pulsations in stars, and are, for now, still evading the grasp of our knowledge.

We divide this master thesis in five chapters. In chapter 1, we will introduce subdwarf B stars, a kind of core helium burning stars that will be the topic of our study, before giving the

[1] "Pulsations" and "oscillations" are equivalent in this master thesis

usual equations of stellar structure and then delving into adiabatic oscillation theory. We will then show the properties on both pressure and gravity modes, with a special focus on the latter as they will occupy the remainder of the master thesis.

In chapter 2, we present the numerical tools which are used for our study, that is the STELUM code and the PULSE code.

Then, in chapter 3, we show the results found for so-called “static models”, an approximation of fully evolutionary models, in particular for the pulsation spectra of gravity modes for subdwarf B stars. We study the influence of a few parameters on those pulsation spectra, and draw physical conclusions on the different types of pulsations we observe relative to the thermal and chemical structure of the star.

In chapter 4, we reproduce this study in the case of evolutionary models and highlight in particular the differences between static and evolutionary models. We focus on the semi-convection and overshooting phenomena, as those influence strongly the pulsation spectra we find.

Finally, in chapter 5, we tie our results together to draw conclusions on the limitations of static models. We underline that some regions of the thermal structure of the core of sdB stars have considerable uncertainties, and that the thermal structure prescriptions adopted in such regions have an especially strong influence on the gravity modes pulsation spectra found afterward.

Chapter 1

Theoretical framework

The pulsations that we observe at the surface of a star are in reality a consequence of its stellar structure and dynamics. The goal of this chapter is to lay the groundwork required to understand what stellar pulsations are, where in the star they take place, and how they are created. To realize this, inspiration was taken from the work of notably Aerts et al. (2010), Unno et al. (1989), in which many additional mathematical developments are made, and two PhD thesis which have helped in synthesizing information, from Charpinet (1999) and Van Grootel (2008).

In order to directly link the theory of stellar oscillations with the particular case of this master thesis, we will first introduce the concept of Helium burning core stars (CHeBs) with a particular focus on subdwarf B stars (sdBs). Following this, we will lay the basis of stellar structure theory, using hydrodynamics and thermodynamics. Then, we will narrow our focus onto adiabatic oscillations, that is, removing the thermal considerations in order to extract the dynamical mechanism of pulsations. Finally, we will come back to non-adiabaticity as a mean to understand part of the origin of stellar oscillations.

1.1 Core He-Burning stars

1.1.1 Generalities on CHeB stars evolution

During their evolution on the main sequence, stars progressively transform the hydrogen in their core into helium. As the star nears the end of this phase, this process becomes less efficient and eventually stops when the hydrogen is exhausted in the core. Afterward, the star will climb the red giant branch (RGB) during which its core will stay inert as the star's core temperature is not hot enough to fuse helium, while its hydrogen will burn in a shell surrounding the core. Because of this hydrogen shell burning, the helium core will progressively gain mass, until the critical point to fuse helium is reached, and we then stop climbing the red giant branch, instead beginning the horizontal branch (HB) where core helium burning takes place. While on the HB, we fuse helium into carbon through the triple-α process, and when there is enough carbon, we fuse helium with carbon to produce oxygen. For low and intermediate mass stars, which are the one we are interested in, the lack of mass of this new core made of carbon and oxygen will not be sufficient to induce carbon fusion reactions. This inert C-O core will then constitute the core of the white dwarf, the final stage of the star after it has expelled most of its external envelope.

In this master thesis, we are interested in the HB, which is composed of CHeB stars. The morphology of the HB requires a helium core of almost fixed mass of about $M_{\text{core}} = 0.47 M_{\odot}$, with little variation around that value. This can be understood through the so-called helium flash phenomenon. During the red giant phase, stars with a mass of less than about 2.2 $M_\odot$ will not be able to reach the critical temperature required to fuse helium in their core, despite the latter getting denser from the helium produced by hydrogen shell burning. Upon reaching a high enough core mass, the hydrostatic equilibrium is not sustained anymore, as no fusion reactions in the core are preventing a gravitational collapse. The star thus starts contracting and its temperature increases, until the center of the helium core becomes degenerate matter, at which point the degeneracy pressure will stop the inner part of the core from contracting further. Matter above this degenerated inner core will continue to contract, and we eventually reach the fusion temperature for helium. Because we fuse in degenerate matter in which pressure is weakly reliant on temperature, while the reaction rate of the triple-α process is extremely sensible to temperature ($\approx T^{40}$), we will raise the temperature of the core too fast for the degeneracy pressure to expand the core as a cooling mechanism. A higher temperature means more efficient fusion, and as such, we have a runaway reaction which lasts a few seconds and produces a tremendous amount of energy, until the degeneracy in the core is lifted, and the star can finally cool down and continue fusing helium. The minimum core mass at which we lose the hydrostatic equilibrium is in fact the core mass we find for stars on the HB branch. Because of this nearly fixed core mass, we understand that the parameter which will set the position of CHeB stars on the HB will be the mass of their envelope $M_{\rm env}$, which, contrary to the core's mass, is highly variable. This is used to identify the extreme horizontal branch (EHB), where we find the hottest stars of temperatures $T_{\rm eff} > 20\ 000$ K, corresponding to $M_{\rm env} < 0.02 M_\odot$.

The trajectory from a star on the HB to a white dwarf is not straightforward and also depends on $M_{\rm env}$ after the helium core has been consumed. As illustrated in Fig. 1.1, a star on the HB will generally follow the asymptotic giant branch (AGB then P-AGB), during which helium will be burned in shells, eventually producing a planetary nebula. A star on the EHB has two possible paths depending on the mass of its envelope. If it is sufficiently massive ($0.01 \lesssim M_{\rm env} \lesssim 0.02 M_\odot$), the star will develop a convective envelope and will follow the AGB for a time, but will leave it before the thermal pulses leading to a planetary nebula. Indeed, because of hydrogen shell burning at the bottom of the envelope, combined with stellar wind at its surface, convection will stop early on the AGB and the star will cool towards a white dwarf following the so-called post-early AGB (P-EAGB). If the EHB star does not have a massive enough envelope however ($M_{\rm env} \lesssim 0.01 M_\odot$), then the latter will not become convective, and the star will follow a path called AGB-Manqué, in which the star keeps a high temperature, unlike the AGB and P-EAGB branches, and finally reaches the white dwarf cooling track.

1.1.2 Importance of studying CHeB stars

Transport processes, either microscopic or macroscopic, have a strong influence on the stellar chemical stratification of CHeB stars during their evolution, hence on the resulting white dwarf. By performing asteroseismic modeling on CHeB pulsators such as EHB stars, blue HB stars, or Red Clump stars, one can hope putting constraints on those transport processes and on the internal structure of these stars. In particular, what happens at the boundary between convective core and radiative mantle (see next subsection Fig. 1.2 for the location of those zones), is not well understood and thus modeled approximatively through overshooting and semi-convection (partial mixing) phenomena, with different recipes from different authors, for instance in Castellani et al. (1971a) and Constantino et al. (2015) (see also a comparison of various prescriptions in Blouin et al. 2024). This lack of knowledge leads to some discrepancies between the pulsation periods of models with different prescriptions. For example, Giammichele et al. (2022) showed that the main differences we find for gravity mode pulsation periods in white dwarf stars are due to those core uncertainties in current models. Transport processes

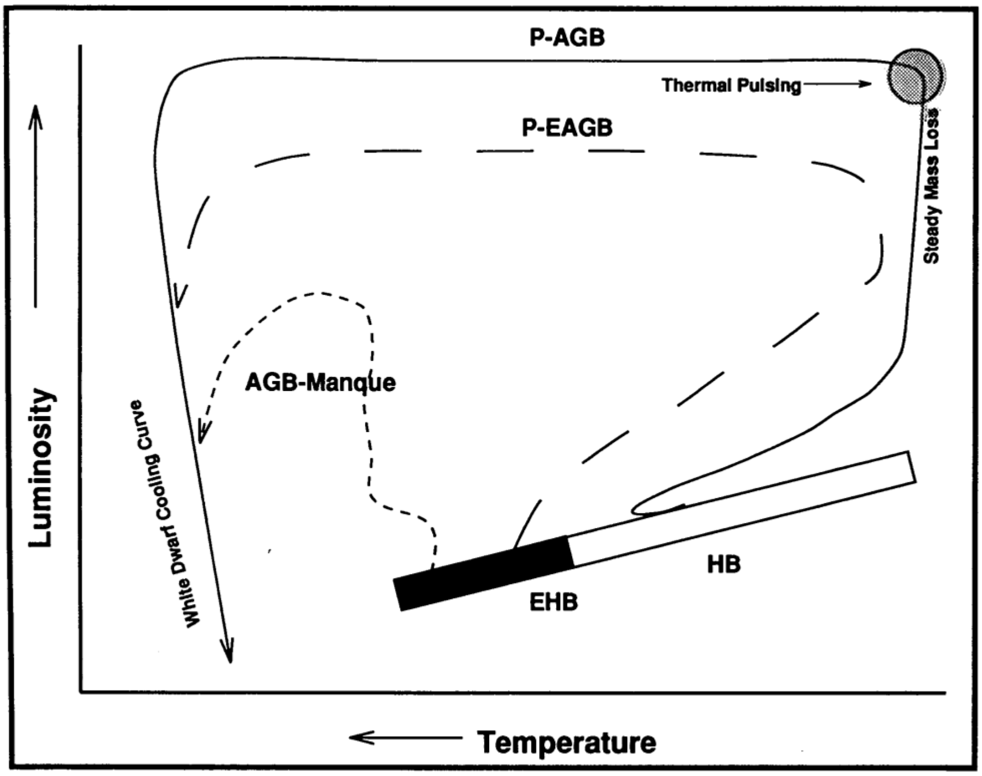


Fig. 1.1: Representation of post HB and EHB stars evolution, from Dorman et al. (1993) (Fig. 1)

such as overshooting and semi-convection, strongly influence the chemical composition of the core of CHeB stars and need to be studied in more details, which we aim to do through this master thesis.

1.2 Focus on sdB stars

In the context of this work, to study the influence of the chemical structure of CHeB stars on pulsation spectra, we will focus on hot B subdwarfs (sdBs). These particular stars have been discovered through spectroscopic observation by Humason and Zwicky (1947), having similar effective temperatures as OB main sequence stars, but with much lower luminosities in an HR diagram, hence the "subdwarf" denomination. sdB stars have been identified as HB stars, meaning core-He burning stars, lying on the blue end of the HB (aka the Extreme Horizontal Branch), by Heber (1986). Those stars have been theorized to pulsate by a team at Université de Montréal (Charpinet et al. 1996, 1997), which was coincidentally confirmed by observation at the same time by a team of astronomers at the South African Astronomical Observatory (Kilkenny et al. 1997).

A schematic chemical structure of sdB stars is illustrated in Fig. 1.2, in which gray zones are convective and white zones are radiative (details of the transition at the convective core boundaries are not represented here). We find first a convective core (I), in which we have triple-α processes burning the helium into carbon as well as, later, the fusion of carbon and helium into oxygen. Upon going toward the surface of the star, the temperature drops below what is needed for helium burning, and we find a radiative core (II) mainly composed of helium, which is the consequence of the former H-shell burning region during the red giant branch. We will call this region "radiative mantle" afterward, for a clear distinction between which part of the core we talk about. Above it, we find the envelope divided in two parts, a first part composed of a mix of helium and hydrogen (III), and a second part composed of near pure hydrogen (IV). The choice of a pure hydrogen layer is motivated by studies of the composition of the stellar atmosphere of sdBs, which show for the majority a deficiency in helium and heavy metals, specifically carbon, nitrogen and silicium. In addition, helium will also sink in the star over time in a radiative envelope due to being heavier than hydrogen and not supported by the radiative levitation process (which is efficient for some heavier elements, in particular those of the iron group).

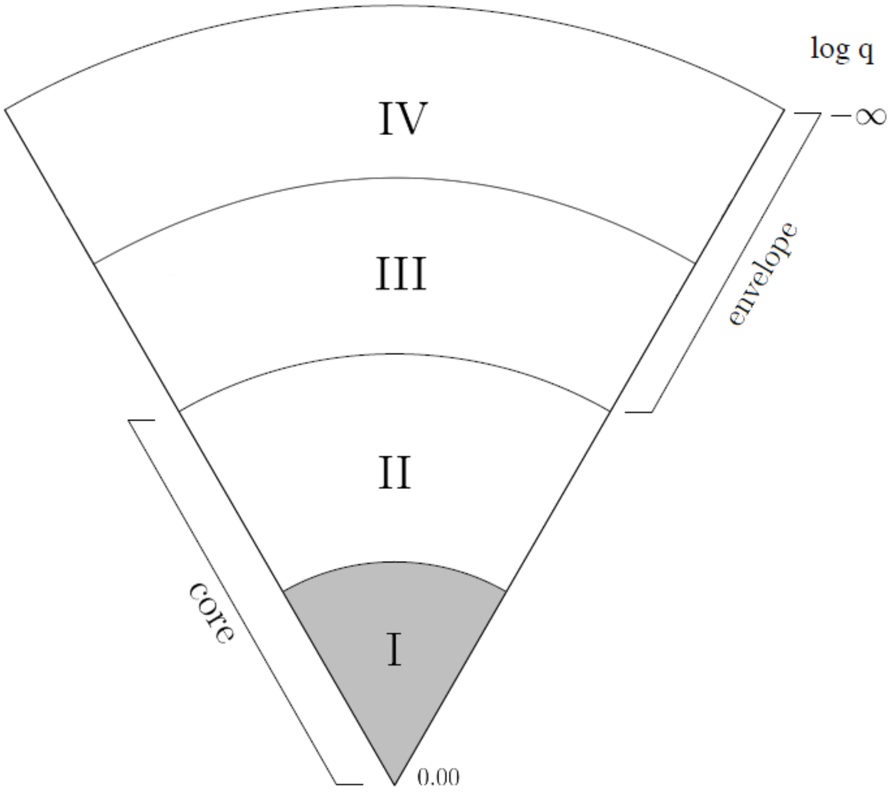


Fig. 1.2: A view of sdB structure from evolutionary models of EHB stars. Inspired from Charpinet (1999) (Fig. 2.3)

As said previously, most stars lying on the HB and EHB have a constraint on their core that has a mass around 0.47 $M_\odot$. sdBs follow this, and so will the models we will use later on. Additionally, the envelope's mass $M_{\rm env}$ will be our varying parameter, and will range from 0.02 $M_\odot$ to 0.5×10^{-6} $M_\odot$. It is customary to express the masses by a parameter called the mass fraction q, as $\log q = \log\left(1 - \frac{m(r)}{M_*}\right)$, where M_* is the total mass of the star and $m(r)$ is the mass contained in a sphere of radius r, both in $M_\odot$. This is the unit used in Fig. 1.2, and in theory, because of the definition of the mass fraction, we would find $\log q = 0$ at the center and $\log q = -\infty$ at the surface. In practice, we need to bind q in numerical simulations, and we find in Charpinet (1999) that we can do so and fix our surface with $\log q \in [-16, -14]$ without consequences on the pulsation spectrum afterward. We distinguish two types of pulsators for sdB stars. We first find fast period pulsating sdBs named “V361 Hya”, with pulsation periods between 80s and 600s discovered by Kilkenny et al. (1997), which will not be the focus of this master thesis. Instead, we will look at gravity modes with longer pulsation periods of the so-called “V1093 Her” stars, with observed periods between 3000 to 12000s and first found by Green et al. (2003).

1.3 Stellar structure

Now that we know which stars we are working with, we need to lay the groundwork of stellar pulsation theory. To do so, we will first define the equations of equilibrium structure of our star, then we will study the stability of perturbations around that equilibrium structure.

1.3.1 Equilibrium structure

We first consider the following equations of hydrodynamics :

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \tag{1.1}$$

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} = -\nabla \Phi - \frac{\nabla P}{\rho} \tag{1.2}$$

$$\rho T \left(\frac{\partial}{\partial t} + \mathbf{v} \cdot \nabla \right) S = \rho \epsilon_N - \nabla \cdot \mathbf{F_R} \tag{1.3}$$

$$\nabla^2 \Phi = 4\pi G \rho \tag{1.4}$$

This system is composed in descending order of the mass, momentum, and energy conservation equations, as well as the Poisson equation linking the gravitational field Φ to the density. The symbols are of usual meaning, with ρ the density, P the pressure, S the entropy and T the temperature.

In those equations, a few assumptions are made. The star is assumed to not be rotating, thus not accounting for inertial forces, and we do not account for a magnetic field as well. Note that those effects could be added to our models by considering a small perturbation from those effects in our equations. Additionally, convection and viscosity effects are not considered. While it is reasonable to neglect viscosity effects due to the Reynolds number being high in a star, effects induced by convection such as overshooting, semi-convection, and partial mixing zones, can have a strong impact and will need further investigation. The energy in the star is considered to be mainly produced by nuclear reactions, and we note the energy produced by unit mass from nuclear reactions as ϵ_N.

Two more equations are needed to close the system, which are the state equation and the energy transfer one. The former links the pressure, density, and temperature of our gas, and is seldom found in analytical form for stellar interiors. For the latter however, because the stellar interior is a relatively opaque medium, we assume to be in the diffusion approximation. This allows going from a stochastic process, that is, the movement of photons scattered in our medium, to a diffusion process, and the radiative flux $\mathbf{F_R}$ is then found as :

$$\mathbf{F_R} = -\frac{4acT^3}{3\kappa\rho} \nabla T$$

Where c is the speed of light, a is the radiation density constant and κ is the opacity. We will consider this radiative flux as the only factor in energy transport, as conduction plays only a minor part and is thus considered negligible.

We find the hydrostatic equilibrium by considering the aforementioned quantities as static in time as well as having $\mathbf{v} = \mathbf{0}$. Considering spherical symmetry and a system of classical coordinates (r,θ,ϕ), we have quantities dependent on r only, that is, the length from the center of the star. From this, we infer the stellar structure equations as :

$$\frac{dP}{dr} = -\rho g \text{ with } g = \frac{Gm}{r^2} \tag{1.5}$$

$$\frac{dm}{dr} = 4\pi\rho r^2 \tag{1.6}$$

$$\frac{dT}{dr} = -\frac{3\kappa\rho L}{16\pi acr^2T^3} \tag{1.7}$$

$$\frac{dL_r}{dr} = 4\pi\rho r^2\epsilon_N \text{ with } L_r = 4\pi r^2 F_r \tag{1.8}$$

In which all quantities are equilibrium quantities, and m is the mass contained in a sphere of radius r, while L_r is the luminosity through this same sphere. We shall now use this system to find the unperturbed equilibrium solutions for our star.

1.3.2 Small perturbation theory

Our goal is to study the oscillations of a star, and their stability. In order to do this, we cannot simply solve the hydrostatic equation system, as it is non-linear. However, since the amplitudes of the oscillations in a star are of small scale compared to the whole object, we can use the so-called small perturbation theory. This allows to act as if the deformations of the star due to oscillations are simply small perturbations around the equilibrium state, and we can then linearize the set of equations we have, then solving it analytically with a few extra steps. In order to do this, we consider the Eulerian and Lagrangian descriptions. On the one hand, an Eulerian description presupposes that we are a static observer, and we describe the various quantities as dependent on a position $\mathbf{r}$ and a given time t. This is a fixed system of coordinates independent of motion. On the other hand, a Lagrangian description is from the perspective of a moving observer, which would "sit" on the particle of fluid that is represented. The particle's movement is described using its initial position $\mathbf{r_0}$ and from it, we get its actual position $\mathbf{r}(\mathbf{r_0}, t)$. Here, the system of coordinates is thus following the motion. Using those descriptions, we express the perturbation in two forms, first an Eulerian form as :

$$X(\mathbf{r}, t) = X_0(\mathbf{r}) + X'(\mathbf{r}, t) \tag{1.9}$$

Where $X'(\mathbf{r}, t)$ is the Eulerian perturbation from the equilibrium quantity $X_0(\mathbf{r})$.
Then, a Lagrangian perturbation described as :

$$X(\mathbf{r}, t) = X_0(\mathbf{r_0}, t) + \delta X(\mathbf{r}, t) \tag{1.10}$$

Where $\delta X(\mathbf{r}, t)$ is the Lagrangian perturbation which shifted the fluid parcel from $\mathbf{r_0}$ to $\mathbf{r}$. As such, we define the Lagrangian displacement vector $\delta\mathbf{r}$ as : $\mathbf{r}(\mathbf{r_0}, t) = \mathbf{r_0} + \delta\mathbf{r}(\mathbf{r_0}, t)$, which we use to get the relation between Eulerian and Lagrangian perturbations since :

$$\delta X(\mathbf{r}, t) = X'(\mathbf{r}, t) + (X_0(\mathbf{r}) - X_0(\mathbf{r_0}, t)) \tag{1.11}$$

Using Taylor series to express the parenthesis and keeping only the first order, we have :

$$\delta X(\mathbf{r}, t) \simeq X'(\mathbf{r}, t) + \delta\mathbf{r} \cdot \nabla X_0(\mathbf{r}, t) \tag{1.12}$$

And from the stellar structure equations, we derive a system of linearized equations :

$$\frac{\partial\rho'}{\partial t} + \nabla\cdot(\rho\mathbf{v}) = 0 \tag{1.13}$$

$$\rho\frac{\partial\mathbf{v}}{\partial t} + \nabla P' + \rho\nabla\Phi' + \rho'\nabla\Phi = 0 \tag{1.14}$$

$$\rho T\frac{\partial}{\partial t}\left(S' + \delta\mathbf{r}\cdot\nabla S\right) = (\rho\epsilon_N)' - \nabla\cdot\mathbf{F}' \tag{1.15}$$

$$\nabla^2\Phi' = 4\pi G\rho' \tag{1.16}$$

Where $\mathbf{F}' = -K\nabla T' - K'\nabla T$.
This set of partial differential equations is now linear, and as we know from (1.5)-(1.8), the equilibrium quantities are functions of r only. This means, with the derivation details in Unno et al. (1989), that we can separate the time and spatial dependencies, such that we can write our quantities as what we call a normal mode :

$$f'_{klm}(r,\theta,\phi,t) = f'_{kl}(r)Y_l^m(\theta,\phi)e^{i\sigma_{kl}t} \tag{1.17}$$

Where σ_{kl} is the oscillation frequency and Y_l^m are the normalized spherical harmonics written as :

$$Y_l^m(\theta,\phi) = (-1)^m\left[\frac{(2l+1)(l+m)!}{4\pi(l+m)!}\right]^{\frac{1}{2}} P_l^m(\cos(\theta))e^{im\phi} \tag{1.18}$$

Where P_l^m is a Legendre polynomial, l is a positive integer including zero, and m is an integer between $-l$ and l such that for a given l, there are $2l+1$ values of m. We can make a distinction between non-radial pulsations, which are modes of $l > 0$, and radial pulsations of $l = 0$, which notably have no dependencies on θ and ϕ due to $Y_0^0(\theta,\phi)$ being a constant. As with all linear systems of equations, solutions are constructed by linear combinations, hence here of multiple normal modes. As such, the solution of our system takes the form :

$$f'(r,\theta,\phi,t) = \sum_{k,l,m}\alpha_{klm}f'_{klm}(r,\theta,\phi,t) \tag{1.19}$$

With α_{klm} an arbitrary coefficient representing the amplitude of the normal mode of indices k, l, m.

1.3.3 Geometry of stellar oscillations

In order to better our visualization of the solution (1.17), we can view on Fig. 1.3 a geometrical representation of spherical harmonics at the surface of a star. We characterize pulsations using the indices introduced previously. "k" represents the radial order, which corresponds to the number of nodes of a given mode between the center of the star and its surface. "l" is the degree of a mode, and gives the number of surface nodes on the star. Finally, "m" is the azimuthal order and $|m|$ is the number of surface nodes which are lines of longitudes. We thus understand that $l - |m|$ is the number of surface nodes that are lines of co-latitude, that is, lines aligned with the equator of the star, the latter being the line dividing the star in two with a plane perpendicular to its rotation axis. Additionally, the sign of m gives us which way the mode propagates. A positive azimuthal order is a prograde mode that travels in the direction of the rotation of the star, while a negative azimuthal order is a retrograde mode that travels opposite to the rotation. Note that in this master thesis, we will refer as "nodes" the nodes associated to the radial order k, in contrast to "surface nodes" associated to l and m. We can see on (1.17) that neither $f'_{kl}(r)$ nor σ_{kl} are dependent on m. This is because of our previous assumption that the problem is of spherical symmetry, which is inducing a $2l + 1$ degeneracy on both quantities. Lifting this degeneracy is equivalent to removing the spherical symmetry, which supposes that we consider a rotating star and/or magnetic effects, which is not done during this master thesis.

In practice, observing modes of $l \geq 4$ is hardly possible. Since l is giving the number of surface nodes of a star, the higher the degree of a mode, the more the star is divided into region of opposite velocities. Because the luminosity we receive is integrated over the surface of the star we observe, higher order modes features are harder to resolve spatially, and the

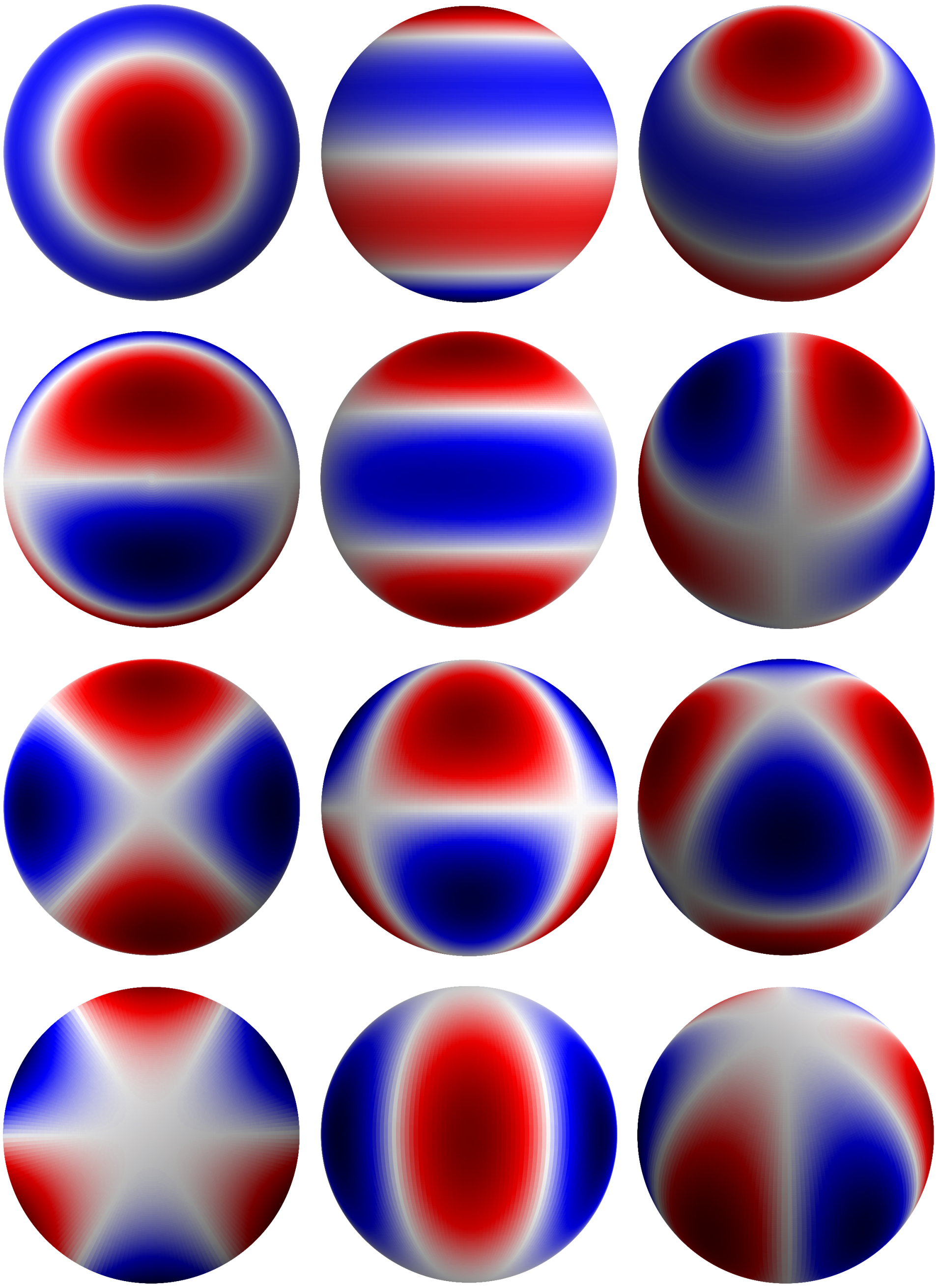


Fig. 1.3: Spherical harmonics for $l = 3$ order. The rows are in ascending m, from $m = 0$ at the top to $|m| = 3$ at the bottom. Red and blue colors represent a section coming from and towards us at a given time, while white lines represent the surface nodes.

effect of regions of opposite velocities near one another will tend to cancel out. This effect is called partial cancellation. Note however that the observability of each mode is also dependent on their intrinsic amplitude, which makes them more or less observable individually. We can roughly quantify this effect if we interest ourselves in the normalizing factor of equation (1.18) for a given l. As displayed in Aerts et al. (2010), for axisymmetric (m = 0) modes, we have the following :

$$Y_l^0(\theta, 0) = \left(\frac{2l+1}{4\pi}\right)^{\frac{1}{2}} P_l(\cos(\theta))$$

Integrating this quantity over the disk of the star facing the observer, that is for $\theta \in [0, \frac{\pi}{2}]$, we get the partial cancelling factor for axisymmetric modes. The closer to zero this factor is, the more the mode is affected by partial cancellation, as illustrated on Fig. 1.4.

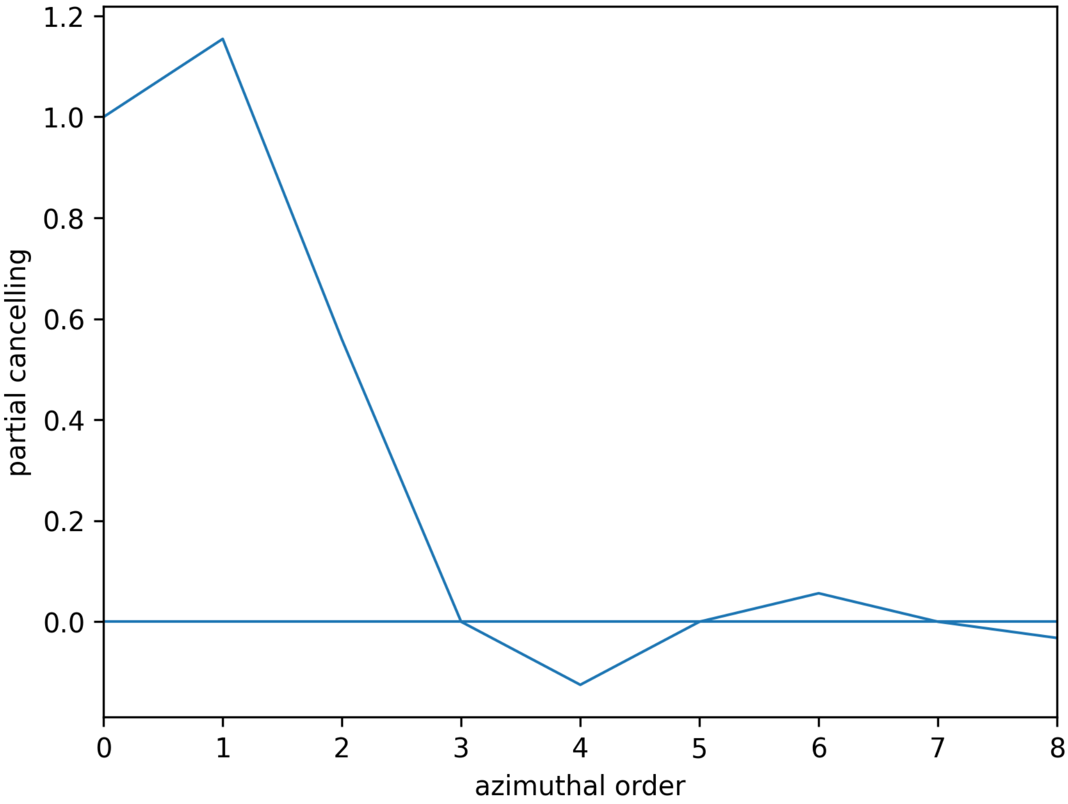


Fig. 1.4: Partial cancelling factor for modes of orders $l = 0$ to $l = 8$ and $m = 0$. The first mode is normalized to 1 as radial modes are not subject to partial cancellation in first approximation.

We first verify on Fig. 1.4 that the higher the degree of a mode, the less observable it is, as we understood earlier. In addition, we realize that even degree modes are easier to observe than odd degree modes, which is a direct consequence of the form of Legendre polynomials, except for $l = 1$. Indeed, despite the integral of P_1 being lower than the P_0 one, the normalization factor is higher for $l = 1$, and results in $l = 1$ being overall easier to detect. Additionally, the sign of the partial cancelling factor for even order modes is coming from the amount of zeros of the Legendre polynomials over the interval we are considering, which is half of the order of the polynomial since $\theta \in [0, \frac{\pi}{2}]$. An even number of zeros will lead to a positive integral, while an odd number of zeros will lead to a negative integral. It is however important to understand that it is not the sign that is important here, but the distance from zero of a given value of the partial cancelling factor. We understand that we will mainly observe modes of $l = 1$, $l = 2$ and even $l = 4$ for non-radial pulsations in practice, which is verified in the case of sdBs, although modes of higher degree have been observed occasionally with the Kepler satellite. As a result of this, we will limit our interest of pulsations up to modes of $l = 4$ in the models we will use. It is also important to note that the partial cancelling factor computed here is an approximation in so far that we do not take in account rotation, limb darkening effects, or the intrinsic amplitude of a given mode.

1.3.4 Stellar oscillations equations

As we showed earlier, we can express the quantities in the stellar structure equations as linear combinations of normal modes, composed of spherical integrals. Following this, we now express the displacement vector $\boldsymbol{\xi}$ as :

$$\boldsymbol{\xi} = \left[\xi_r(r), \xi_h(r)\frac{\partial}{\partial\theta}, \xi_h(r)\frac{1}{\sin(\theta)}\frac{\partial}{\partial\phi}\right] Y_l^m(\theta,\phi)e^{i\sigma t} \tag{1.20}$$

In which the natural frequency of oscillation σ_{kl} is noted σ and $\xi_h(r)$ is given by :

$$\xi_h(r) = \frac{1}{\sigma^2 r}\left(\frac{P'}{\rho} + \Phi'\right) \tag{1.21}$$

And assuming that we do have our quantities expressed as normal modes, then we obtain :

$$\frac{1}{\rho}\frac{dP'}{dr} + \frac{g}{\rho c_s^2}P' + (N^2 - \sigma^2)\xi_r + \frac{d\Phi'}{dr} = g\nabla_{ad}\frac{\rho T}{P}\delta S \tag{1.22}$$

$$\frac{1}{r^2}\frac{d}{dr}\left(r^2\xi_r\right) + \frac{1}{\Gamma_1}\frac{d\ln\rho}{dr}\xi_r + \left(1 - \frac{L_l^2}{\sigma^2}\right)\frac{P'}{\rho c_s^2} - \frac{l(l+1)}{\sigma^2 r^2}\Phi' = \nabla_{ad}\frac{\rho T}{P}\delta S \tag{1.23}$$

$$\frac{1}{r^2}\frac{d}{dr}\left(r^2\frac{d\Phi'}{dr}\right) - \frac{l(l+1)}{r^2}\Phi' - 4\pi G\rho\left(\frac{P'}{\rho c_s^2} + \frac{N^2}{g}\xi_r\right) = -4\pi G\nabla_{ad}\frac{\rho^2 T}{P}\delta S \tag{1.24}$$

$$K\frac{dT'}{dr} = -F_r' - K'\frac{dT}{dr} \tag{1.25}$$

$$i\sigma\rho T\delta S = (\rho\epsilon_N)' - \frac{1}{r^2}\frac{d(r^2F_r')}{dr} - \frac{l(l+1)}{r^2}KT' \tag{1.26}$$

$$\frac{\delta T}{T} = \nabla_{ad}\frac{\delta P}{P} + \frac{\delta S}{c_p} \tag{1.27}$$

In which c_s is the sound velocity and ∇_{ad} is the adiabatic gradient as :

$$c_s = \left(\frac{\Gamma_1 P}{\rho}\right)^{\frac{1}{2}} \text{ and } \Gamma_1 = \left(\frac{\partial\ln P}{\partial\ln\rho}\right)_S \tag{1.28}$$

$$\nabla_{ad} = \left(\frac{\partial\ln T}{\partial\ln P}\right)_S \tag{1.29}$$

In addition to those quantities, we identify two frequencies which are of capital importance, the Lamb frequency L_l and the Brunt-Väisälä frequency N, which are written as :

$$L_l^2 = \frac{l(l+1)c_s^2}{r^2} \tag{1.30}$$

$$N^2 = g\left(\frac{1}{\Gamma_1}\frac{d\ln p}{dr} - \frac{d\ln\rho}{dr}\right) \tag{1.31}$$

While we will come back to those quantities in section 1.4.2, we can already get a physical sense of what they represent. The Lamb frequency is the frequency of a horizontal sound wave. Because the radius r is lower towards the core of the star, and because one can generally link, through equation of state, the ratio $\frac{P}{\rho}$ as proportional to T, which is in fact higher in the deep regions of the star, then we infer that the Lamb frequency will be low in the superficial layers of the star, and high towards the center of it. The Brunt-Väisälä frequency is notably found when considering internal gravity waves in a gas layer horizontally stratified under the action of gravity. In this case, and this is how we shall view this frequency physically, a given perturbed parcel of fluid will oscillate vertically around its equilibrium position, with a frequency N, given that the density of our fluid parcel is higher than its surroundings.

We can express the Brunt-Väisälä frequency in a different form, which will be of much use later on :

$$N^2 = \frac{g^2\rho}{P}\frac{\chi_T}{\chi_\rho}\left(\nabla_{ad} - \nabla_T + B\right) \tag{1.32}$$

In which :

$$\chi_T = \left(\frac{\partial \ln P}{\partial \ln T}\right)_\rho \quad , \quad \chi_\rho = \left(\frac{\partial \ln P}{\partial \ln \rho}\right)_T \quad \text{and} \quad B = -\frac{1}{\chi_T}\left(\frac{\partial \ln P}{\partial \ln \mu}\right)_{\rho,T}\frac{d\ln \mu}{d\ln P} \tag{1.33}$$

∇ is the real gradient of temperature while B is the Ledoux term, or the compositional gradient. A demonstration of the expression of this Ledoux term can notably be found in Brassard et al. (1991a). This term will be extensively used to understand the models we compute. Indeed, we can see in particular that between two regions of different compositions, there shall be a rise in N^2, as B will be strong from the chemical gradient in what we call a transition zone. Because of this, we will have a direct way to probe the chemical composition of our star and better know the zones of interest to study.

1.4 Adiabatic oscillations

1.4.1 Equations of adiabatic oscillations

In the case of our work, we will study the so-called adiabatic oscillations, meaning that we neglect the transfer of energy between the kinetic energy of the oscillation and the internal energy of the gas. While we are not able to derive results on what drives the modes or their stability in this framework, we can instead focus on the dynamical behavior of oscillations. In practice, we assume in the system of equations (1.22) - (1.27) that $\delta S = 0$, in which case the perturbation of density and pressure is expressed in a way that allows to find a new set of basic equations corresponding to adiabatic oscillations :

$$\frac{1}{r^2}\frac{d}{dr}\left(r^2\xi_r\right) - \frac{g}{c_s^2}\xi_r + \left(1 - \frac{L_l^2}{\sigma^2}\right)\frac{P'}{\rho c_s^2} = \frac{l(l+1)}{\sigma^2 r^2}\Phi' \tag{1.34}$$

$$\frac{1}{\rho}\frac{dP'}{dr} + \frac{g}{\rho c_s^2}P' + (N^2 - \sigma^2)\xi_r = -\frac{d\Phi'}{dr} \tag{1.35}$$

$$\frac{1}{r^2}\frac{d}{dr}\left(r^2\frac{d\Phi'}{dr}\right) - \frac{l(l+1)}{r^2}\Phi' = 4\pi G\rho\left(\frac{P'}{\rho c_s^2} + \frac{N^2}{g}\xi_r\right) \tag{1.36}$$

The adiabatic framework is a double-edged sword. On one side, we lose the information associated with thermodynamic processes, that is, energetic considerations, meaning we cannot study the stability of the eigenfunctions we find. However, on the other side, the new equations are reduced in a way that we can understand the mechanisms and driving forces of different modes. This kind of system needs, for it to be solved, boundary conditions. We will use the boundary conditions used by Unno et al. (1989), such that as the center of the star :

$$\frac{d\Phi'}{dr} - \frac{l\Phi'}{r} = 0 \quad \text{and} \quad \xi_r - \frac{l}{\sigma^2 r}\left(\frac{P'}{\rho} + \Phi'\right) = 0 \tag{1.37}$$

And at its surface :

$$\frac{d\Phi'}{dr} + \frac{(l+1)}{r}\Phi' = 0 \quad \text{and} \quad \xi_r = \frac{P'}{g\rho} \tag{1.38}$$

It is important to realize that the “surface” of a star is ill-defined. Because of the stellar atmosphere, and more often than not, its non-adiabaticity, we need to be careful to not extend our surface too far off, such that we stay within the adiabatic boundaries of the star while applying this theoretical framework. This incurs again a loss of information and of feedback onto our modes. In our case, this becomes particularly relevant knowing that defining the surface point of the star in our models will serve also as a mean to know the extent of the star's envelope, which of course, affects mode propagation.

1.4.2 Overview of pressure and gravity modes

The Cowling approximation, that is, to take $\Phi' = 0$, provides an accurate approximation for high order modes, meaning modes of high k and l. This approximation, while not taking in account any real physical considerations, casts a needed light to gain a global understanding of which kind of modes propagate, and where. However, it will not be used in our models as this approximation requires high l orders, while we already established that we will look at $l \leq 4$. From $\Phi' = 0$, only (1.34) and (1.35) remain from our adiabatic oscillations equations, which take the form :

$$\frac{1}{r^2}\frac{d}{dr}\left(r^2\xi_r\right) - \frac{g}{c_s^2}\xi_r + \left(1 - \frac{L_l^2}{\sigma^2}\right)\frac{P'}{\rho c_s^2} = 0 \tag{1.39}$$

$$\frac{1}{\rho}\frac{dP'}{dr} + \frac{g}{\rho c_s^2}P' + (N^2 - \sigma^2)\xi_r = 0 \tag{1.40}$$

We can rewrite those equations using a change of variable, in which the new variables $\tilde{\xi}$ and $\tilde{\eta}$, are defined with :

$$\frac{d\tilde{\xi}}{dr} = h(r)\frac{r^2}{c_s^2}\left(\frac{L_l^2}{\sigma^2} - 1\right)\tilde{\eta} \tag{1.41}$$

$$\frac{d\tilde{\eta}}{dr} = \frac{1}{r^2 h(r)}(\sigma^2 - N^2)\tilde{\xi} \tag{1.42}$$

$$\tag{1.43}$$

In which :

$$h(r) = \exp\left(\int_0^r \left[\frac{N^2}{g} - \frac{g}{c_s^2}dr\right]\right) > 0 \tag{1.44}$$

Then, by neglecting the radius dependency in the above equations, we find a harmonic oscillator which is identical for both variables and the dispersion relation associated to it :

$$\frac{d^2\tilde{\xi}}{dr^2} + k^2\tilde{\xi} = 0 \quad \text{and} \quad k^2 = \frac{1}{c_s^2\sigma^2}(\sigma^2 - L_l^2)(\sigma^2 - N^2) \tag{1.45}$$

The solution of this harmonic oscillator is proportional to $\exp(ikr)$. This highlights two cases : either k is imaginary and the solution is called "evanescent", in which case the amplitude is dampened exponentially, or k is real, and the solution is oscillating and propagating radially instead. Using this, we have $k^2 > 0$ and is real in two cases. First, if $L_l^2 > \sigma^2$ and $N^2 > \sigma^2$ then we have what we call gravity modes, or g-modes, for which the restoring force is the buoyancy. Then, if $L_l^2 < \sigma^2$ and $N^2 < \sigma^2$, we have pressure modes, or p-modes, for which the restoring force is the pressure. Otherwise, we have that $L_l^2 < \sigma^2 < N^2$ or $N^2 < \sigma^2 < L_l^2$, and in those cases, we have evanescent waves.

This allows to identify where each wave is propagating in a star, and we can illustrate this with Fig. 1.5 showing a representative sdB model. We find in log scale the Brunt-Väisälä frequency and the Lamb frequency for $l = 1$, as well as two modes : a pressure mode of period $P_p = 200s$ and a gravity mode with $P_g = 10000s$, which are typical values we could find for either modes. We convert these periods in frequencies with $\sigma^2 = \frac{2\pi}{P}$, allowing us to find in log scale both horizontal lines on the figure, with $\log(\sigma_p^2) \approx -3$ and $\log(\sigma_g^2) \approx -6.40$. We have taken here the case of a sdB chemical structure, which informs us on where gravity and pressure modes are travelling in those stars. In Fig. 1.5, the horizontal lines are in red if the mode does not propagate, green if it does. Thus, we understand that pressure modes will propagate in the envelope, while gravity modes will propagate in the core, and also in the envelope for higher radial orders. It should be noted here as well that the convective core of our star is considered as formally evanescent and that no modes will propagate in it. Because a convective zone corresponds to $\nabla_{ad} \leq \nabla$ (Schwarzschild criterion for convective instability), then, in zones without chemical gradients where $B = 0$, we have that $N^2 < 0$ and as such that $\log(N^2)$ is undefined. We see this clearly on our figure, at the convective core as well as at $\log(q) \approx -9$ which correspond to a partial ionization zone of HeII/HeIII.

Propagation cavities are important to keep in mind, as a mode will mainly give information on the zone it propagates in, as such, it is for example not possible to have direct information on the core using pressure modes in sdBs. In some stars, notably red giant with burning helium cores, we also find mixed modes, which exhibit both the properties of p-modes or g-modes depending on their location in the star. This duality makes knowing where the mode propagates increasingly complicated. Those modes are however not found for high radial order modes, as one can demonstrate that g-modes frequencies become smaller with increasing radial order while p-modes frequencies follow the invert trend, meaning that both type of modes are fully decoupled at sufficient radial order k. As we interest ourselves mainly in those high radial orders modes, and because in sdB stars, g and p modes are well decoupled, mixed modes are not studied further.

An important concept of stellar pulsations which we will extensively study in this master thesis is mode trapping. As we have just seen, oscillations propagate in given cavities depending on their frequencies, while outside these cavities, we find evanescent regions where the oscillations are damped exponentially. The path of a given oscillation is not as straightforward as it may seem. Upon reaching the boundaries of its propagation cavities (there can be multiple cavities), a mode encounters an evanescent zone. When this happens, the mode is partially reflected and is thus considered trapped in the propagation cavity it is currently in. Those partial reflections can also occur at zones of high chemical gradient. Because of its expression, the Brunt-Väisälä frequency is strongly dependent on whether a region is convective or radiative, whether we have a strong chemical gradient, and where we are in the star because of the $\frac{\rho g^2}{P}$ factor. In practice, we call a mode trapped in a region of a star when it spends most of its time in said region, and in this case, the region in which the mode is considered trapped is a subregion of the propagation cavity itself. Such trapping often occurs as a consequence of sharp transitions coming from chemical gradients, or a sudden transition from radiative to convective zone. This is verified in the case of sdB stars. Indeed, going back to Fig. 1.5, we have a strong chemical transition from convective core to radiative mantle at $\log(q) \approx -0.25$ and from radiative mantle to the envelope at $\log(q) \approx -2$, both of which can be identified directly from the sharp local maxima at those positions coming from the Ledoux term in the Brunt-Väisälä frequency. For g-modes, located in the core, those chemical transitions can and will be responsible for trapping phenomena, as we will see in chapter 3 and 4 of this work.

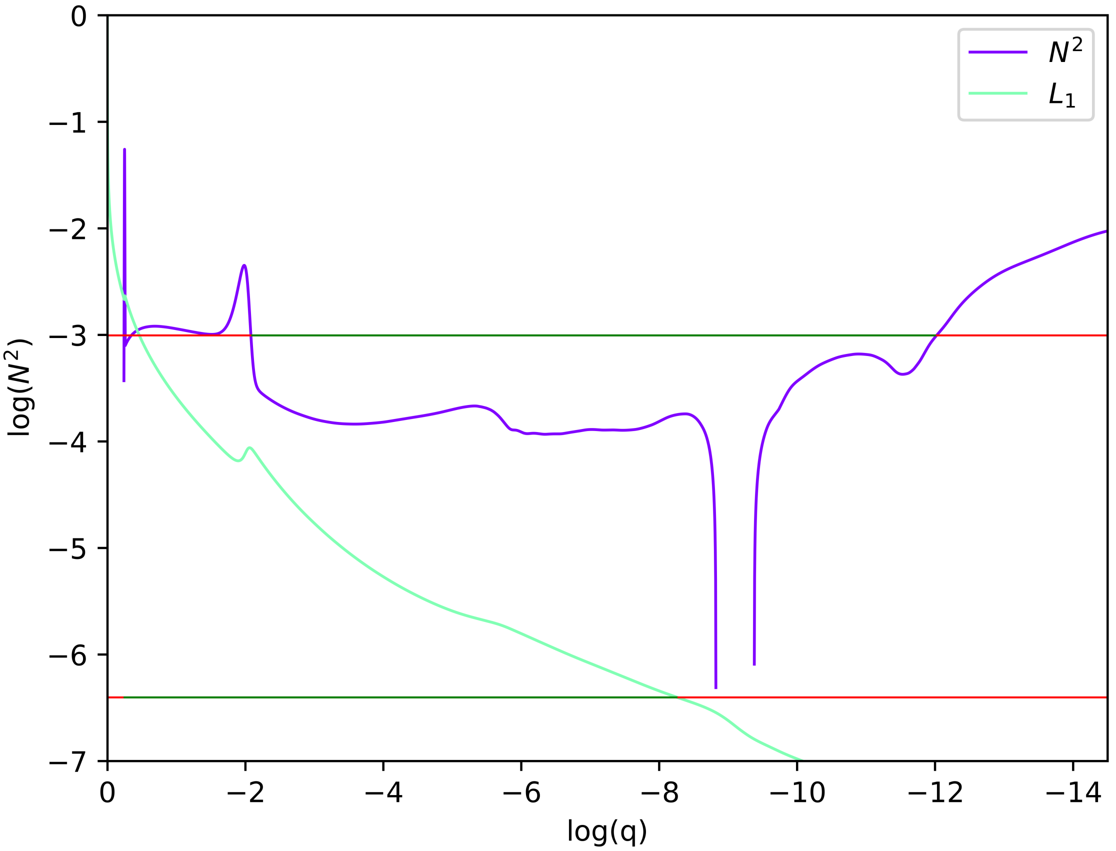


Fig. 1.5: Diagram showing the Brunt-Väisälä and Lamb frequencies in a sdB model. The horizontal lines are for one p-mode (top line) and one g-mode (bottom line), where the red zones are evanescent, while the green zones are propagation cavities.

1.5 Stability of oscillations

The adiabatic approximation allowed us to get a relatively accurate view of the dynamics of an oscillation in a star, by understanding where a given pulsation propagates, and the influence of the structure of the star on our modes. This approximation has its limits, and in particular, we cannot know the possible driving mechanisms that allow a pulsation to develop and grow through the adiabatic equations. A first look at non-adiabatic theory and its consequences is therefore presented in this section, to get a global idea of how modes occur in a star.

1.5.1 Work integral

To know how a mode could develop in a star, one needs to study its stability, as it is an unstable solution of the non-adiabatic system (1.22)-(1.27) that will allow a pulsation to grow and become potentially observable. Equation (1.26) directly induces that the solution of the non-adiabatic equations are in the complex plane. As such, for a given Lagrangian perturbation, we find :

$$\delta X(r,t) = \Re\left(\delta X(r)e^{i\sigma t}\right) \tag{1.46}$$

Because σ and $\delta X(r)$ are complex, we can express the perturbation as :

$$\delta X(r,t) = |\delta X(r)| \cos(\sigma_R t + \varphi(r))e^{\eta t} \tag{1.47}$$

In which we then have that $\sigma = \sigma_r + i\sigma_I$, and we call $\eta = -\sigma_I$ the growth rate of the mode considered, while $\varphi(r)$ is the phase of our quantity. There is then an oscillating and an exponential part to our solution, which means for the latter that if $\eta > 0$, the mode is unstable, and stable if $\eta < 0$.

This instability is viewed as a thermal instability in addition to having an oscillating part, and so, one can view those thermal oscillations as a sort of Carnot cycle in thermodynamics. Indeed, thermal energy is converted into mechanical energy as to produce an oscillation, and mechanical energy is in turn turned into thermal energy in the cycle. What we need to know, is if the repetition of such a cycle is potentially exciting our modes, and to compute this, we define the work integral, as follows :

$$W = \int_0^M \frac{dW}{dm} dm \tag{1.48}$$

In which $\frac{dW}{dm}$ is the local work from a singular cycle. If the process is supposed reversible and as coming back to its original state upon each oscillation, then we find (developments can be found in Unno et al. 1989) :

$$\frac{dW}{dm} = \oint \delta T \frac{d\delta S}{dt} dt \tag{1.49}$$

$$= \oint \frac{\delta T}{T} \delta \left(\epsilon_N - \frac{1}{\rho} \nabla \cdot \mathbf{F_R} \right) \tag{1.50}$$

Becoming after the integration :

$$\frac{dW}{dm} = \frac{\pi}{\sigma_R} \Re \left(\frac{\delta T^*}{T} \left[\delta\epsilon_N - \delta \left(\frac{1}{\rho} \nabla \cdot \mathbf{F_R} \right) \right] \right) \tag{1.51}$$

With this last expression, we understand that there should be locally driving zones and locally dampening zones for the oscillations. If the local work is positive, then the mode will gain kinetic energy in this local framework, while it will lose some if the local work is negative instead. Then, by computing W, we get the total contribution of each of those local effects on our mode. If $W > 0$, the mode is globally excited, unstable, and can potentially be observed, while if $W < 0$, the mode is globally dampened, stable, and will not be observed. It should be noted that even if we do know that our modes are subject to amplitude changes as they go through those different zones, we can only make relative comparisons of amplitudes. Linear theory as we use it does not induce a limit of amplitude on excited modes, however we do know that those modes do not grow infinitely in amplitude since this is not physical. This allows to pinpoint a limitation of the linear theory against the reality of the matter at hand.

1.5.2 Driving mechanisms

In the general case, one can split equation (1.51) into three parts, which are associated to the nuclear energy generation rate, the convective flux, and the radiative flux. In our case however, we have neglected the convective flux since the beginning, and this is reflected in our equation as we have $\mathbf{F_R}$, the radiative flux, instead of $\mathbf{F}$, the total flux. As a consequence, we distinguish not three, but two main driving mechanisms in our case. We can write both parts of our local work as W_N and W_F :

$$\frac{dW_N}{dm} = \frac{\pi}{\sigma_R} \Re \left(\frac{\delta T^*}{T} \delta\epsilon_N \right) = \frac{\pi}{\sigma_R} \Re \left(\frac{\delta T^*}{T} \epsilon_N \left[\epsilon_T \frac{\delta T}{T} + \epsilon_\rho \frac{\delta \rho}{\rho} \right] \right) \tag{1.52}$$

$$\frac{dW_F}{dm} = \frac{\pi}{\sigma_R} \Re \left(\delta \left[-\frac{1}{\rho} \nabla \cdot \mathbf{F_R} \right] \right) \tag{1.53}$$

In which we have :

$$\epsilon_T = \left(\frac{\partial \ln \epsilon_N}{\partial \ln T}\right)_\rho \quad \text{and} \quad \epsilon_\rho = \left(\frac{\partial \ln \epsilon_N}{\partial \ln \rho}\right)_T \tag{1.54}$$

The first mechanism, associated with W_N, is called ϵ-mechanism, and is related to the nuclear energy generation in stars. Since the nuclear reaction rate has a strong tendency to increase with temperature and density, one can infer that ϵ_T and ϵ_ρ will be positive quantities, and as such, we will have a mechanism able to excite our modes. This is true in nuclear burning regions, for example the core of a star, or regions of shell-burning. While the ϵ-mechanism is present in the majority of stars, its effects are low and quite negligible for almost all of them, except potentially for high mass stars. Currently, there is no known class of pulsating stars that have been shown to rely solely on the $\epsilon-$mechanism as their driving mechanism. However, works such as Battich et al. (2017, 2018) suggest that it could exist and might be observable.

The other mechanism, is known as the κ-mechanism. The opacity in a star is heavily reliant on temperature. When matter is compressed, the temperature rises, and the opacity diminishes, as such, radiative transfer will be more potent, effectively dampening the pulsation. In a partial ionization zone however, we have an advert effect, indeed, as highlighted in Cox (1980), the opacity will rise during compression, which means that radiative transfer is impeded, and as such, energy is gained and trapped locally. This triggers a positive feedback loop which excites the pulsations, and in particular, an opacity bump at the location of a partial ionization zone is generally associated with the driving of oscillations propagating in that zone. In sdB stars in particular, numerical models where we find an opacity bump due to the partial ionization of elements of the iron group are models which allow for pressure modes to develop (Charpinet 1999). It was shown by Fontaine et al. (2003) that the κ-mechanism associated to elements of the iron group is also responsible for the g-modes observed in sdB stars.

Chapter 2

Numerical tools and models

Now that we have seen the theoretical framework in which the master thesis takes place, we can introduce the numerical tools that we will use. In this chapter, we will first present the STELUM code used to compute the structure of white dwarfs and hot subdwarfs, in our case sdB stars, and we will distinguish two types of models : the static and evolutionary models. Then, we will introduce the tools to find the pulsations occuring in such a structure, using the PULSE code. We will introduce the dimensionless equations which are used in this code, as well as weight functions which are a powerful tool to distinguish how different regions of the stellar structure affect a specific mode.

2.1 Stellar structure models : STELUM

In order to model sdBs, we will use the STELUM code, standing for STELlar modeling from the Université de Montréal. This code was first introduced in Fontaine et al. (2001) and progressively updated, for example in Quirion et al. (2012) with more detailed diffusion processes and opacity treatment, and in Van Grootel et al. (2013) which introduce so-called third generation (3G) static models to allow for the study of g-modes in sdB stars. STELUM also computes evolutionary models and is fully described in Bédard et al. (2022), which allows understanding in particular how transport of elements takes place in the code through the following equation of transport (equation (8) in the paper):

$$\frac{dX_i}{dt} = S_{nuc,i} - \frac{1}{r^2\rho}\frac{d}{dr}\left(r^2\rho\left[(v_i + v_{wind} + v_{acc}X_i) - D\frac{dX_i}{dr}\right]\right) \tag{2.1}$$

Which describe the mass fraction X_i of element i at a given radius r. $S_{nuc,i}$ is the source/sink term which account for the creation or destruction of the element i by nuclear reactions, v_i is the diffusion coefficient for element i, v_{wind} the stellar wind velocity, v_{acc} the accretion velocity and D the macroscopic diffusion coefficient. As said in Bédard et al. (2022), the transport equation under this form highlights the distinction between microscopic and macroscopic transport, where the effects of atomic diffusion are contained in v_i, whereas the effects of large-scale motions are captured by v_{wind}, v_{acc}, and D. This diffusion coefficient D accounts in particular for the transport of elements in a convective zone, but also for convective overshooting, semi-convection and thermohaline convection, and we find the expression of the coefficient for each effect in Bédard et al. (2022), respectively with equations (14), (15), (16) and (17). In this master thesis, we will focus in particular on the effects of semi-convection on g-mode spectra in chapter 4, but we will see that overshooting can also have a strong effect on our modes.

2.1.1 Static and Evolutionary models

STELUM is designed as an evolutionary code targeting in particular white dwarfs and subdwarf hot stars, which, from an initial state, computes the evolution of the stellar structure of a given star through time. This requires a tremendous amount of computational power, and makes those evolutionary models poorly suited for parametric exploration. Because this is the case, we understand that evolutionary models are too rigid to use as-if for seismic analysis, that is, to explore parameters as to reproduce an observed pulsation spectrum, and thus infer the internal structure of a given pulsating star. To remedy this issue, static models were introduced in Brassard and Fontaine (1994) as a faster and flexible alternative to evolutionary models. In sdBs, triple-α processes occur in the convective core and the influence of shell burning hydrogen on energy generation is negligible, thus, the luminosity is essentially constant above the core region where nuclear reactions are occurring. Of course, it is important to verify that we do get similar structures for both static and evolutionnary models in order to justify that this approximation is applicable in our case, which was done extensively in Charpinet et al. (2002a). We learn in particular in this paper that static models reproduce quantities such as the Brunt-Väisälä frequency of evolutionary models with good precision.

Static models have been iteratively upgraded, starting from 1G to 4G models. In 1G models, we find stellar structure models of sdBs with uniform metallicity, which are found to be stable against pulsation in Charpinet et al. (1996) for sdB stars of solar metallicity. This induced the development of 2G models, lifting the uniform metallicity hypothesis. In those, we find that a κ-mechanism, presented in subsection 1.5.2, can excite p-modes and produce oscillations. In those 2G models, the stellar structure model of the sdB is not complete in so far that these suppose a static envelope above a completely inert hard sphere. Since g-modes are sensible to core conditions, this was inducing a strong discrepancy on the period founds for g-modes between evolutionary and static models. However, it is notable that 2G models are very good at computing p-modes which are less sensible to core conditions. This was done for a few stars such as PG 1336-018 (Charpinet et al. 2008), PG 1219+534 (Charpinet et al. 2005b) and Feige 48 (Charpinet et al. 2005a). Because g-mode pulsators exist for sdB stars, 3G models were developed as to enable the study of g-modes, and are described in Van Grootel et al. (2013). This time, we have a complete stellar structure in which the core is represented, and those models produced encouraging results for the seismic probing of sdB pulsators. Finally, we find 4G models, which, instead of a sharp transition between the He-rich radiative mantle and H-rich envelope, presents a transition zone of hydrogen and helium, as was presented in Fig. 1.2. Those 4G models are the ones we will investigate in the chapter 3 of this master thesis. It is important to note however that they do not include overshooting or semi-convection effects, and as such, will contrast heavily with the evolutionary models of chapter 4 which do include them.

2.2 Stellar pulsations code : PULSE

2.2.1 Adiabatic pulsation code equations

In order to study non-radial adiabatic oscillations of white dwarfs, P. Brassard et al. have published a series of papers, in which is described a numerical way to study the pulsations in those stars using a finite element code called PULSE (in particular Brassard et al. 1992c). More recently, this work has been expanded upon in Brassard and Charpinet (2008), in which the PULSE code is now presented and used for multiple stellar objects, including sdBs. The PULSE code equations are solved in a dimensionless way where several new variables y_1 to y_4

are introduced, and are defined as :

$$y_1 = \frac{\xi_r(r)}{r} \quad \text{and} \quad y_2 = \frac{1}{gr}\left(\frac{P'}{\rho} + \Phi'\right) = \sigma^2 \frac{\xi_h(r)}{g} \tag{2.2}$$

$$y_3 = \frac{1}{gr}\Phi' \quad \text{and} \quad y_4 = \frac{1}{g}\frac{d\Phi'}{dr} \tag{2.3}$$

As well as dimensionless quantities for the frequencies and the stellar structure variables, in which M and R shall be the total mass and radius of the star :

$$\omega^2 = \frac{\sigma^2 R^3}{GM} \quad ; \quad U = \frac{4\pi r^3 \rho}{m} \quad ; \quad V = \frac{\rho g r}{P} \tag{2.4}$$

$$C_1 = \left(\frac{r}{R}\right)^3 \left(\frac{M}{m}\right) \quad ; \quad A = -\frac{N^2}{g} \quad ; \quad x = \ln\left(\frac{r}{P}\right) \tag{2.5}$$

Which allows to recast the adiabatic equations set into a dimensionless one, as follows :

$$(V+1)\frac{dy_1}{dx} = \left(\frac{V}{\Gamma_1} - l - 1\right) y_1 + \left[\frac{l(l+1)}{C_1\omega^2} - \frac{V}{\Gamma_1}\right] y_2 + \frac{V}{\Gamma_1} y_3 \tag{2.6}$$

$$(V+1)\frac{dy2}{dx} = (C_1\omega^2 + rA)y_1 + (3 - l - U - rA)y_2 + rAy_3 \tag{2.7}$$

$$(V+1)\frac{dy_3}{dx} = (3 - l - U)y_3 + y_4 \tag{2.8}$$

$$(V+1)\frac{dy_4}{dx} = -rAUy_1 + \frac{UV}{\Gamma_1}y_2 + \left[l(l+1) - \frac{UV}{\Gamma_1}\right] y_3 + (2 - U - l)y_4 \tag{2.9}$$

With the boundary conditions at the star's center :

$$C_1\omega^2 y_1 - l y_2 = 0 \tag{2.10}$$

$$l y_3 - y_4 = 0 \tag{2.11}$$

And at its surface :

$$(V - 4 - C_1\omega^2)y_1 + \left[\frac{l(l+1)}{C_1\omega^2} - V\right] y_2 + (V - l - 1)y_3 = 0 \tag{2.12}$$

$$U y_1 + (l+1)y_3 + y_4 = 0 \tag{2.13}$$

Finally, we normalize our eigenfunction as :

$$y_1^2 + y_2^2 = 1 \text{ at the surface of the star} \tag{2.14}$$

Those equations, the boundary conditions, and the normalization, are the basis of the code with which we shall pulsate our models of stellar structure.

2.2.2 Kinetic energy and weight function

It is shown in Chandrasekhar (1964) that the eigenvalues of the system of adiabatic equations (1.34)-(1.36) can be found through a variational approach. This can be used notably to find the modification on frequencies of our modes from effects such as rotation. Here however, the star is supposed to not be rotating, and the variational approach in this case gives :

$$\sigma^2 = \frac{D}{A} \tag{2.15}$$

Where A is directly proportional to the kinetic energy. We can express the latter as :

$$E_{kin} = \frac{1}{2}\int_V \rho v^2 dV = \frac{\sigma^2}{2}\int_V \xi \cdot \xi^* \rho dV \tag{2.16}$$

Using the definition of the displacement vector (1.20), we thus have :

$$E_{kin} = \frac{\sigma^2}{2}\int_0^R \left[\xi_r(r)^2 + l(l+1)\xi_h(r)^2\right] \rho r^2 dr \tag{2.17}$$

We need to be careful as to how we approach the results of this quantity. Indeed, we find here the same issue as with the amplitude of a given mode, since the eigenfunctions of our problem are normalized arbitrarily, we can only make relative comparisons of kinetic energy between different modes. The D term is found in Brassard (1991), and again in Charpinet (1999), as :

$$D = \int_0^R \left[\xi_r^2 N^2 + \frac{(p')^2}{p\Gamma_1\rho} + \Phi'\left(\frac{p'}{\Gamma_1 p} + \xi_r \frac{N^2}{g}\right)\right] \rho r^2 dr \tag{2.18}$$

In which the integrand acts as a weighting term of the different contribution of the eigenfunctions in the term D, according to where we are in the star from the term ρr^2. We can show using now the dimensionless variables y_1 to y_4 introduced earlier that we thus have :

$$\sigma^2 \propto \int_0^R F(y_1, y_2, y_3, y_4, r) dr \tag{2.19}$$

Where F is called *weight function.* The weight function gives us the contribution of the different regions of the star on the value of σ^2. The higher the weight function is in a given region, the strongest its weight on the frequency of the mode. We can show how we use this in practice with Fig. 2.1.

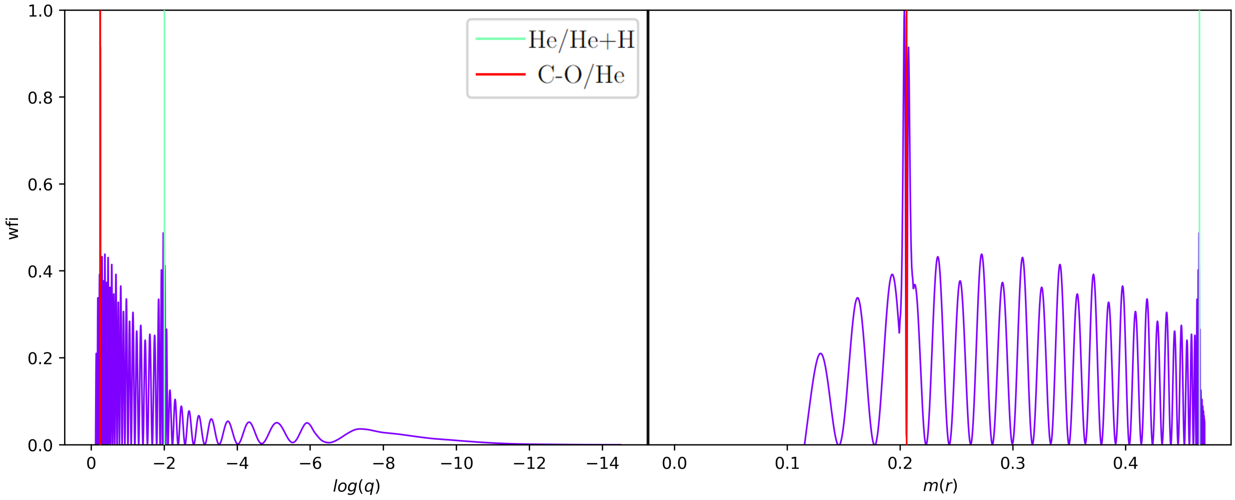


Fig. 2.1: Weight function in log(q) scale and m(r) scale, for g-mode l = 1, k = 40, and for a representative sdB model of total mass $0.47M_\odot$. The green vertical line shows the He/He+H transition from radiative mantle to the hydrogen-rich envelope, here taken $\log(q) = -2.0$. The red vertical line shows the C-O/He transition from convective core to radiative mantle, here taken at $\log(q) = -0.25$. The red and green vertical lines in the right part show the same transitions in $m(r)$ scale.

We show two different scales, the $\log(q)$ scale described earlier which allows seeing what happens in the envelope of the sdB star we are considering, and a scale using $m(r)$, which allows seeing what happens in the core of the star, since, as we said earlier, the core of an sdB represents the majority of its mass. The transition between radiative mantle and envelope is denoted by the red vertical line, and is helping with the mental representation of what each scale represents. As we will investigate only g-modes in this master thesis, here is represented such a mode, for $l = 1$ and $k = 40$, with is weight function normalized arbitrarily to unity. We can clearly see that the mode oscillates in the part highlighted in green as a propagation cavity in the previous Fig. 1.5, and stops oscillating afterward, when the mode goes beyond Lamb's frequency, with a downward trend on the weight function.

In general, the contribution of the core of a given sdB to the frequencies of g-modes is stronger than the envelope, as the core has a higher density with respect to the envelope. We also see here that local maxima of the weight function are found exactly where we have a strong gradient of the chemical composition of the star, that is, when going from mixed core to radiative mantle ("C-O/He transition", $\log(q)$ = -0.25, $m(r) = 0.205\ M_{\odot}$) and going from radiative mantle to the envelope ("He/H+He transition", $\log(q)$ = -2, $m(r) = 0.465\ M_{\odot}$). This highlights an effect we will further develop in chapter 3, concerning the pinching of nodes around zones of chemical gradient, effectively constraining our modes to certain regions of the star.

It is also important to understand that the downward trend we see in the evanescent zone on the weight function is not representative of the fact that the amplitude of the mode is exponentially decreasing in said zone, in fact, the weight function does not give any information for the absolute amplitude of a given mode in a given region. It is only possible to say that we anticipate the amplitude of this g-mode to be higher in the core than in the envelope, given that the weight function is higher in the core. Additionally, it can also be noted that g-modes are not propagating in convective zones, as mentioned earlier. It is better seen in Fig. 2.1 (right) in $m(r)$ scale, where we do not find any oscillations under $m(r) \approx 0.1$.

Chapter 3

Static models, a first look into pulsation spectra

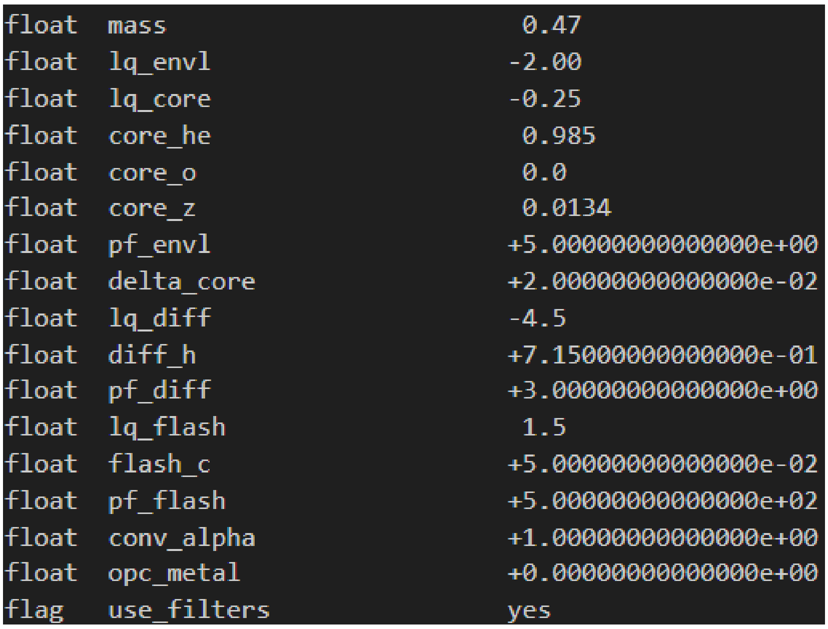
In this chapter, we aim at displaying the results obtained from 4G static models. In particular, we will focus our efforts on highlighting the differences between the modes appearing for different parameters of our numerical models. This will be done through a careful analysis of the radial nodes of those modes, their weight functions, and of course their pulsation spectra.

3.1 Basis for the study of static models

As a mean of introduction for 4G static models, it is important to display the parameters related to the STELUM and PULSE code, as those will directly influence our pulsation spectrum. Rather than looking at the influence of those parameters one by one, which has been previously done in the master thesis of Sebastien Faes (Uliège, 2019), we isolate the parameters that have the most important impact on the pulsation spectrum. In addition, we give the basis as to how we display the data found from both STELUM and PULSE with a model's pulsation spectrum and kinetic energy.

3.1.1 Overview of 4G models

In order to find the pulsation spectrum of a given model, we need to specify the parameters composing our sdB model, and then the parameters of the PULSE code we run on those models. We show in Fig. 3.1 a sample of the different parameters which we can potentially choose from in STELUM. Among those parameters, we expand here on those that have the most important impact on pulsation properties. **mass** allows us to simply fix the total mass of the star in $M_{\odot}$, note that we are around $0.47M_{\odot}$, as per the average core mass of stars on the HB. **lq_env** and **lq_core** allow respectively to fix the He/H+He transition (radiative mantle to envelope) and C-O/He transition (mixed core to radiative mantle), both in $\log(q)$ scale. In our configuration, models are computed from a helium core abundance ranging from 0.05 to **core_he** $= 0.985$, with increments of 0.05 between each model. As such, each time we run this configuration file, we find 197 static models, each independent of one another. In the future, we will refer as "core_he" the helium core abundance of the current model we are working on. **core_o** gives the abundances of oxygen in the core, but in our static models, we fix this to zero at all core helium abundances. Similarly, **core_z** gives the metallicity abundance in the core, and is fixed for all our model to the value we see in Fig. 3.1. **lq_diff** controls the extent of the H+He transition zone in the envelope, as such, **lq_diff** $<$ **lq_env** necessarily. To emulate possible pollution in

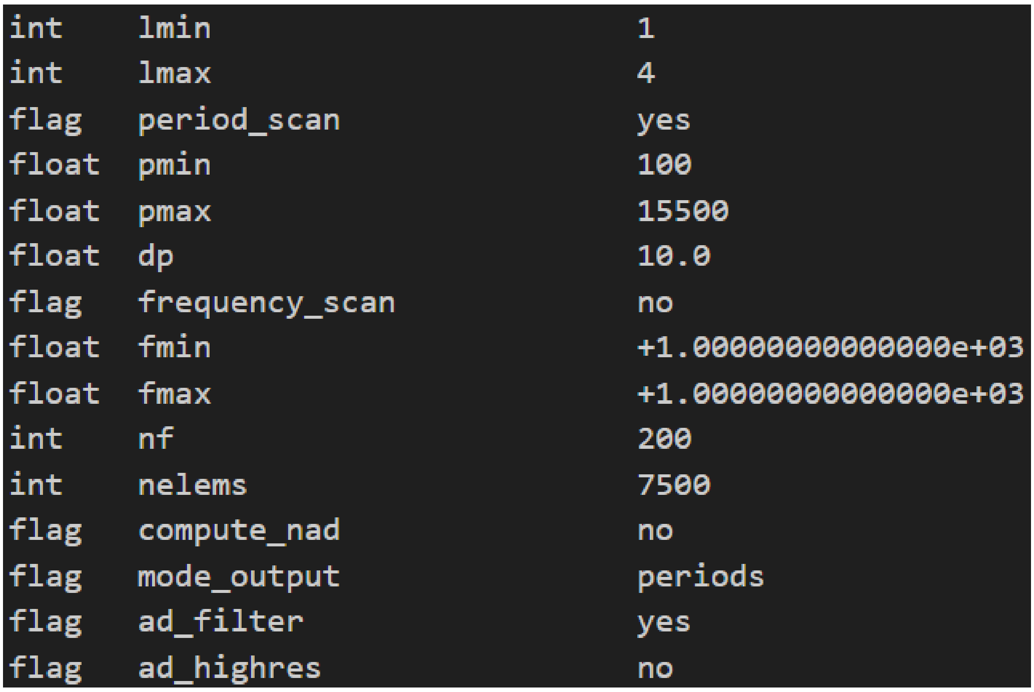


```
float  mass                        0.47
float  lq_envl                    -2.00
float  lq_core                    -0.25
float  core_he                     0.985
float  core_o                      0.0
float  core_z                      0.0134
float  pf_envl                    +5.00000000000000e+00
float  delta_core                 +2.00000000000000e-02
float  lq_diff                    -4.5
float  diff_h                     +7.15000000000000e-01
float  pf_diff                    +3.00000000000000e+00
float  lq_flash                    1.5
float  flash_c                    +5.00000000000000e-02
float  pf_flash                   +5.00000000000000e+02
float  conv_alpha                 +1.00000000000000e+00
float  opc_metal                  +0.00000000000000e+00
flag   use_filters                yes
```

Fig. 3.1: List of parameters to specify in STELUM 4G models.

carbon from the helium flash event, we fix a constant abundance of carbon (here 5%) outside the mixed core of value **flash_c** from **lq_core** to **lq_env**.

Following this, we can now make this 4G model pulsate using PULSE, which of course, takes another row of parameters which are as displayed on Fig. 3.2. Again, I describe here the relevant parameters for our study. We can choose the range of degree l of the modes with **lmin** and **lmax**. In our case, we will always compute all l from $l = 1$ to $l = 4$, as we have seen that above this, modes are harder to observe in practice. Using **period_scan** = **yes**, we indicate that we want to only compute modes within a definite period range, the boundaries of which are defined with **pmin** and **pmax**, over which we scan for modes by increments of **dp**. We take here the minimum period at **p_min** = 100s, and eliminate afterward pulsations



```
int    lmin              1
int    lmax              4
flag   period_scan       yes
float  pmin              100
float  pmax              15500
float  dp                10.0
flag   frequency_scan    no
float  fmin              +1.00000000000000e+03
float  fmax              +1.00000000000000e+03
int    nf                200
int    nelems            7500
flag   compute_nad       no
flag   mode_output       periods
flag   ad_filter         yes
flag   ad_highres        no
```

Fig. 3.2: List of parameters to specify in PULSE code.

corresponding to pressure modes. The maximum period is set at **p_max** = 15500s, however in all of our figures, we take **k_max** = 70 as the maximum radial order, whatever the actual maximum period. We recall that g-modes are observed in sdB stars between 3000 and 12000s, typically. Finally, as we said earlier, since PULSE is a finite element code, we need to fix the number of elements with **nelems**, which allows us to get a better precision on the modes we compute, at the price of higher computation time. Tests at the beginning of the master thesis have shown that nelems = 7500 is a good compromise.

3.1.2 First look into 4G model quantities

First, in order to represent the parameters of STELUM choosen in Fig. 3.1 visually, we can take a look at the chemical structure found Fig. 3.3, in which we display the different chemical transitions at helium core abundance core_he = 0.9. The vertical orange dotted line represents the C-O/He transition from convective core to radiative mantle at lq_core = -0.25[1], the vertical green dotted line represents the He/H+He transition from radiative mantle to H+He envelope at lq_env = -2 and finally the vertical yellow dotted line shows lq_diff = -4.5, which can be viewed as a near half point of the transition from the H+He mixed envelope to the pure H envelope. We also show the pollution of carbon due to the helium flash with the horizontal red dotted line, going from the center of the star to lq_env, and having the carbon abundance value of flash_c = 5×10^{-2}.

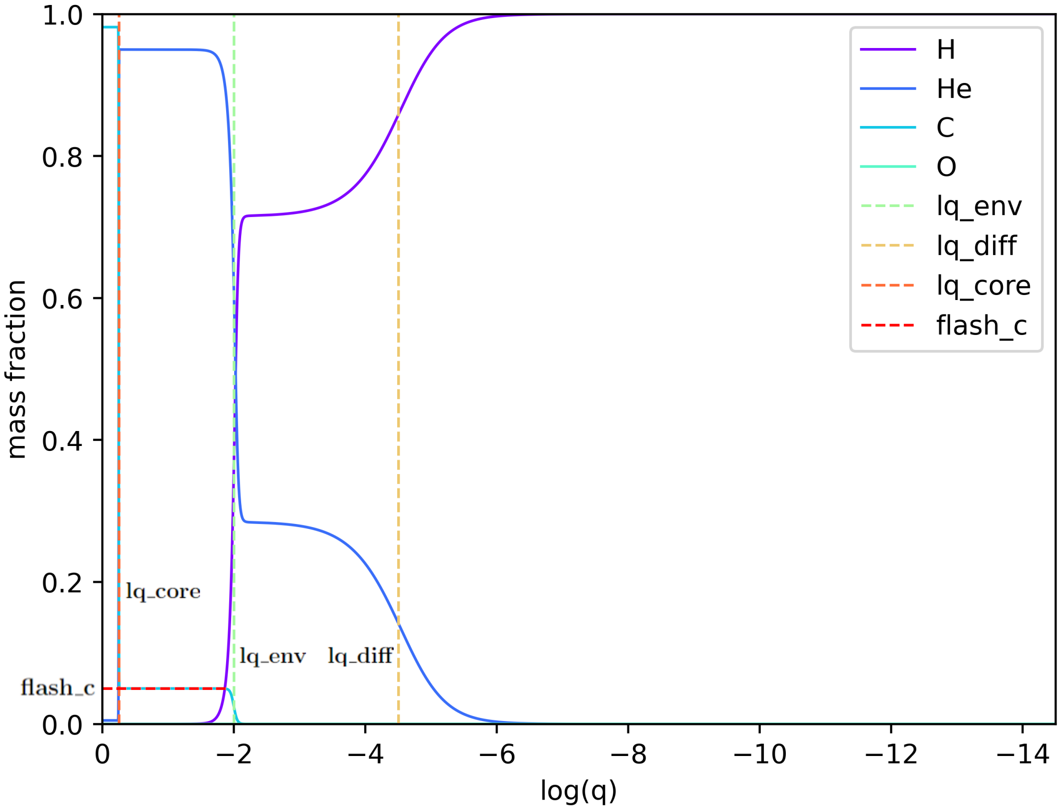


Fig. 3.3: Visual representation of the main 4G models parameters in the context of this master thesis, following the parameters from Fig. 3.1, at core_he = 0.05.

In Fig. 3.4 we show that each transition has a different strength in terms of the Ledoux term B. The Ledoux terms of the He/He+H (radiative mantle to mixed envelope) and He+H/H (mixed envelope to pure H envelope) transitions, for a given set of STELUM parameters, will

[1] let us recall that 4G models do not implement semi-convection and overshooting phenomena.

not change as the chemical conditions of our models are only changing in the core, and thus the conditions before and after aforementioned transitions are the same at all core helium abundances. Because the helium abundance in the radiative mantle does not change while the core helium abundance diminishes, the gradient of helium abundance between convective core and radiative mantle will increase with diminishing core_he, and as such, the C-O/He transition will gain in strength as the abundance of helium in the convective core diminishes. The influence of such transitions on the Brunt-Väisälä frequency can also be seen by putting in parallel Fig. 3.4 with Fig. 3.5, where we find the influence of the trend seen for the Ledoux term (and so the compositional gradient) in the Brunt-Väisälä frequency. In particular, we see on Fig. 3.5 that it is precisely the influence of chemical transitions on the Brunt-Väisälä frequency that will delimit the propagation cavities of our modes, which were viewed in Fig. 1.5. It is clear that while the He+H/H transition around lq_diff at $\log(q) = -4.5$ in the envelope has little influence on the Brunt-Väisälä frequency due to the low value of its associated Ledoux term, the convective core to radiative mantle transition C-O/He and the radiative mantle to envelope transition He/H+He are both clearly visible, and their impact will be further investigated.

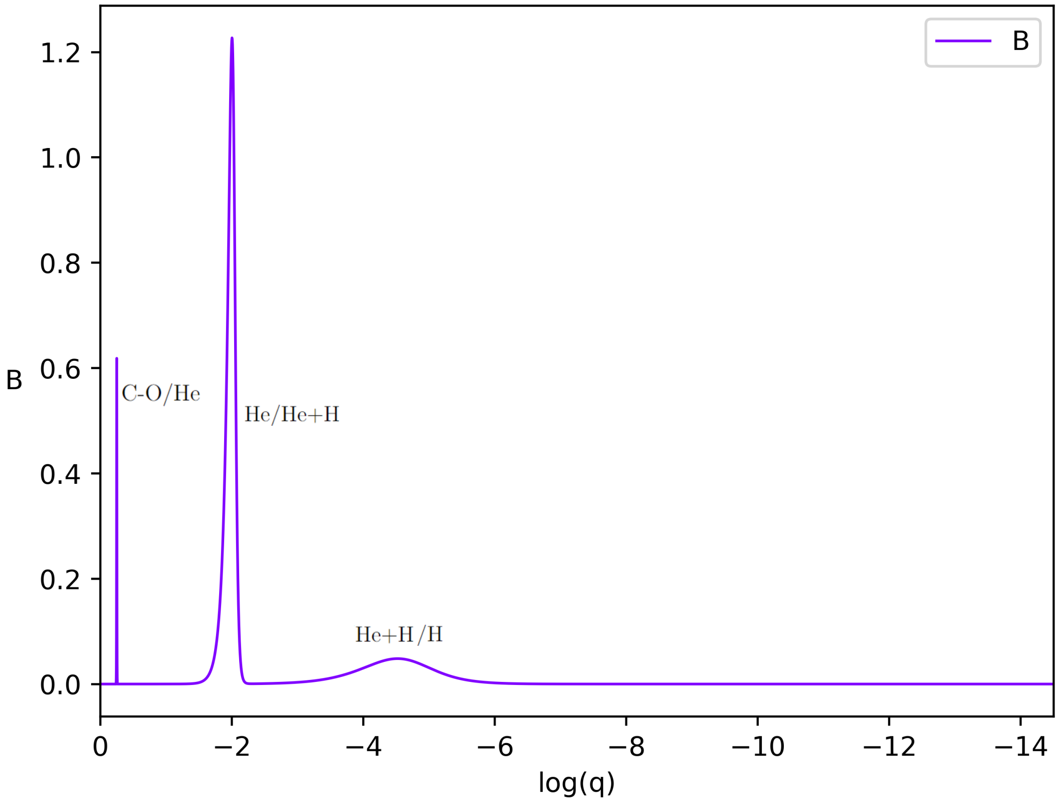


Fig. 3.4: Ledoux term for the 4G model shown in Fig. 3.1, at core_he = 0.9.

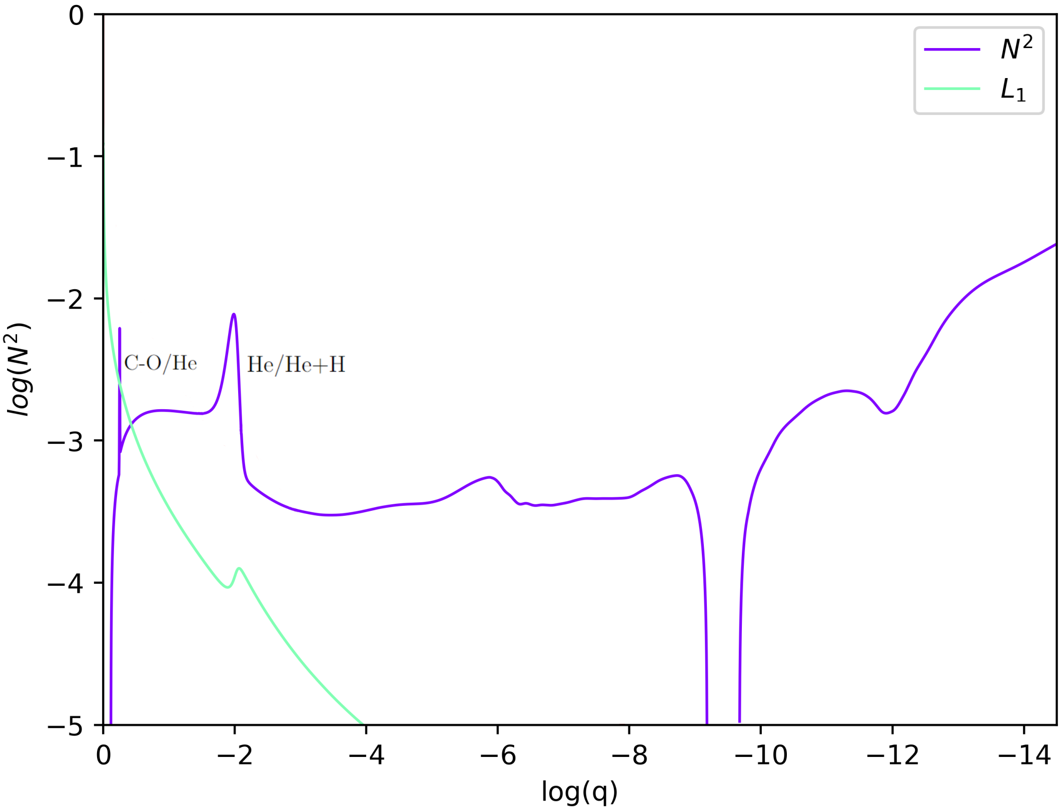
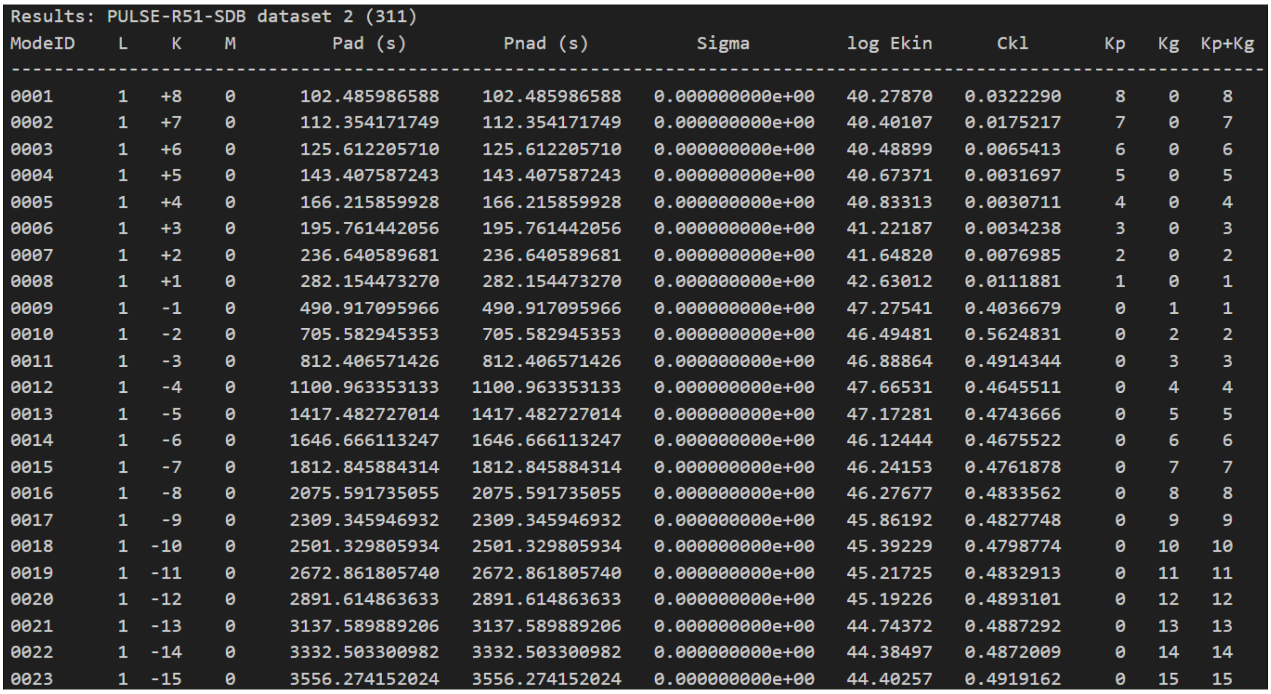


Fig. 3.5: Brunt-Väisälä frequency for the 4G model shown in Fig. 3.5, at core_he = 0.9. We highlighted in particular the chemical transition associated to this model.

3.1.3 Pulsation spectra and associated kinetic energy

In our study, the PULSE code outputs a file of the type displayed in Fig 3.6. Each lines of the file is associated to a different mode, of degree l as "L", radial order k as "K" and azimuthal order m as "M" in said file. The radial order sign gives us the nature of the mode, which is either a p-mode with positive k, or a g-mode with negative k. This is not in reality the case, as we established that all modes have a positive k radial order (a positive number of nodes along the radial direction), but it helps visibly distinguishing both type of modes in the output file. The azimuthal order m is always zero, since we do not consider effects lifting the degeneracy in m of our modes, that is rotation or magnetic effects which would break the spherical symmetry of our model. We find two columns for the periods of the modes, named "Pad" and "Pnad", for respectively adiabatic periods and non-adiabatic periods. In our case, because we only carry out an adiabatic study, both columns display equal values. Finally, the last column which is interesting to us is the "Ekin" one, which displays the value of the kinetic energy associated to each mode, using equation (2.17). While the figure does not show modes above g-modes $k = 15$, the full PULSE file contains every degree from $l = 1$ to $l = 4$, and up to radial order $k = 70$.

In Fig. 3.7, we find a usual representation of a pulsation spectrum, for the PULSE file parameters displayed in Fig. 3.6. We have an abscissa showing the reduced period, defined for a given mode of indices l, k as $P_{l,k} = P_k \times \sqrt{l(l+1)}$. In ordinate, we find the reduced period spacing, which is composed of the period spacing between two modes of consecutive degree k, such that $\Delta P = P_{k+1} - P_k$, combined with the aforementioned reduced period, and we thus define the reduced period spacing as $\Delta P \times \sqrt{l(l+1)}$. This way of displaying the pulsation spectra allows to have a superposition between the spectrum of modes of different l, as displayed



```
Results: PULSE-R51-SDB dataset 2 (311)
ModeID    L    K    M        Pad (s)         Pnad (s)          Sigma         log Ekin      Ckl        Kp   Kg  Kp+Kg
-----------------------------------------------------------------------------------------------------------------------

0001      1   +8    0     102.485986588    102.485986588   0.000000000e+00   40.27870   0.0322290     8    0    8
0002      1   +7    0     112.354171749    112.354171749   0.000000000e+00   40.40107   0.0175217     7    0    7
0003      1   +6    0     125.612205710    125.612205710   0.000000000e+00   40.48899   0.0065413     6    0    6
0004      1   +5    0     143.407587243    143.407587243   0.000000000e+00   40.67371   0.0031697     5    0    5
0005      1   +4    0     166.215859928    166.215859928   0.000000000e+00   40.83313   0.0030711     4    0    4
0006      1   +3    0     195.761442056    195.761442056   0.000000000e+00   41.22187   0.0034238     3    0    3
0007      1   +2    0     236.640589681    236.640589681   0.000000000e+00   41.64820   0.0076985     2    0    2
0008      1   +1    0     282.154473270    282.154473270   0.000000000e+00   42.63012   0.0111881     1    0    1
0009      1   -1    0     490.917095966    490.917095966   0.000000000e+00   47.27541   0.4036679     0    1    1
0010      1   -2    0     705.582945353    705.582945353   0.000000000e+00   46.49481   0.5624831     0    2    2
0011      1   -3    0     812.406571426    812.406571426   0.000000000e+00   46.88864   0.4914344     0    3    3
0012      1   -4    0    1100.963353133   1100.963353133   0.000000000e+00   47.66531   0.4645511     0    4    4
0013      1   -5    0    1417.482727014   1417.482727014   0.000000000e+00   47.17281   0.4743666     0    5    5
0014      1   -6    0    1646.666113247   1646.666113247   0.000000000e+00   46.12444   0.4675522     0    6    6
0015      1   -7    0    1812.845884314   1812.845884314   0.000000000e+00   46.24153   0.4761878     0    7    7
0016      1   -8    0    2075.591735055   2075.591735055   0.000000000e+00   46.27677   0.4833562     0    8    8
0017      1   -9    0    2309.345946932   2309.345946932   0.000000000e+00   45.86192   0.4827748     0    9    9
0018      1  -10    0    2501.329805934   2501.329805934   0.000000000e+00   45.39229   0.4798774     0   10   10
0019      1  -11    0    2672.861805740   2672.861805740   0.000000000e+00   45.21725   0.4832913     0   11   11
0020      1  -12    0    2891.614863633   2891.614863633   0.000000000e+00   45.19226   0.4893101     0   12   12
0021      1  -13    0    3137.589889206   3137.589889206   0.000000000e+00   44.74372   0.4887292     0   13   13
0022      1  -14    0    3332.503300982   3332.503300982   0.000000000e+00   44.38497   0.4872009     0   14   14
0023      1  -15    0    3556.274152024   3556.274152024   0.000000000e+00   44.40257   0.4919162     0   15   15
```

Fig. 3.6: Portion of the output file from the PULSE code associated to parameters on Fig. 3.2, here at core_he = 0.9.

here with in blue modes of $l = 1$ and in red modes of $l = 4$. Each dot corresponds to a radial order, for up to $k = 70$, increasing with the reduced period. We show in subsection 4.3 that the reduced period spacing between two modes is, at high k order and with the hypothesis of our framework (adiabaticity, spherical symmetry), a constant value for any degree l. This however is only true asymptotically in k order, and a general approximation. The pulsation spectrum displayed on Fig. 3.7 is indeed not just a simple horizontal line, but rather we have regular minima of ΔP, which as we will see later, correspond to modes trapped in some regions of the star.

We display the kinetic energy in Fig. 3.8 associated to the pulsation spectrum presented in Fig. 3.7. The figure shares the same reduced period abscissa as Fig. 3.7, and the ordinate shows the value of the kinetic energy for each mode. Here again, we present in blue the kinetic energy of modes of $l = 1$, and in red of modes of $l = 4$, for up to radial order $k = 70$, increasing with the reduced period. As showed in equation (2.17), the kinetic energy of a mode directly depends on its location in the star, due to the ρr^2 term. In particular, deep regions of the star hold more weight in the kinetic energy than superficial regions, that is, the envelope. We focus on the high radial order here, which presents a slow downward trend on the kinetic energy, for any l. The downward trend of the kinetic energy is a direct consequence of higher radial order modes propagating further up in the envelope. Indeed, because the period of a g-mode is increasing with radial order, which means that the value of their frequency $\sigma^2 = \frac{2\pi}{P}$ diminishes with radial order, and because the Lamb frequency is higher towards the surface with increasing frequency (as displayed in Fig. 3.5), a mode of higher radial order has a propagation cavity which is higher toward the surface. This directly implies that the higher the radial order of a g-mode, the further up it propagates in the surface, and thus the less its kinetic energy.

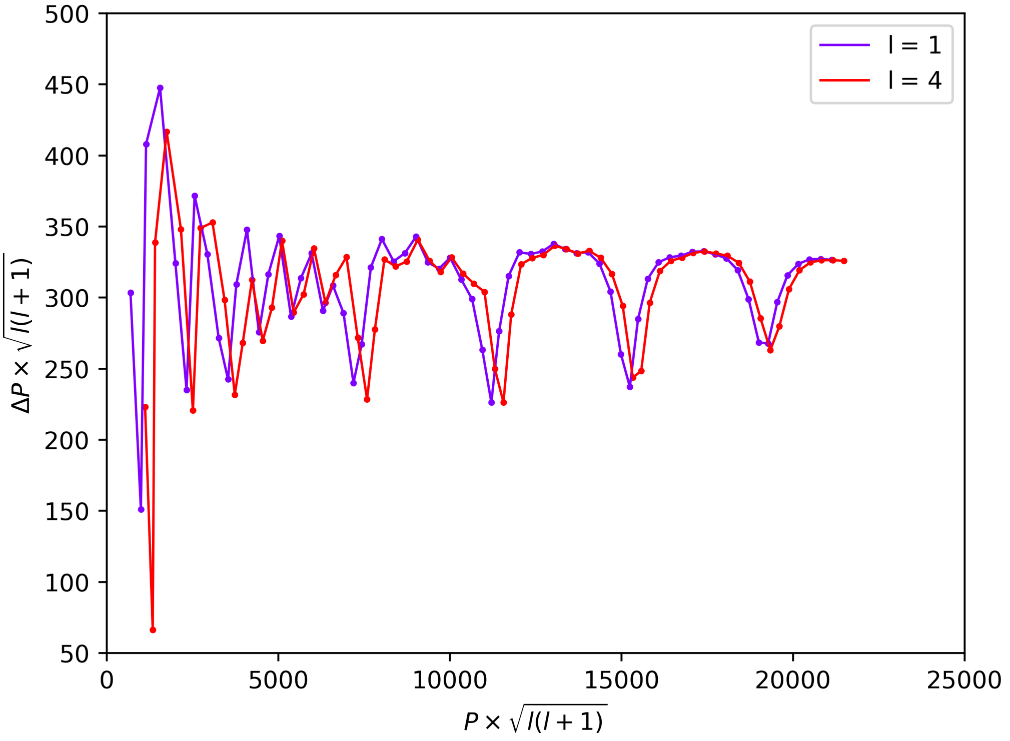


Fig. 3.7: Pulsation spectrum representation of the PULSE and STELUM parameters used in Fig. 3.1 and Fig. 3.2, taken at helium core abundance core_he = 0.9.

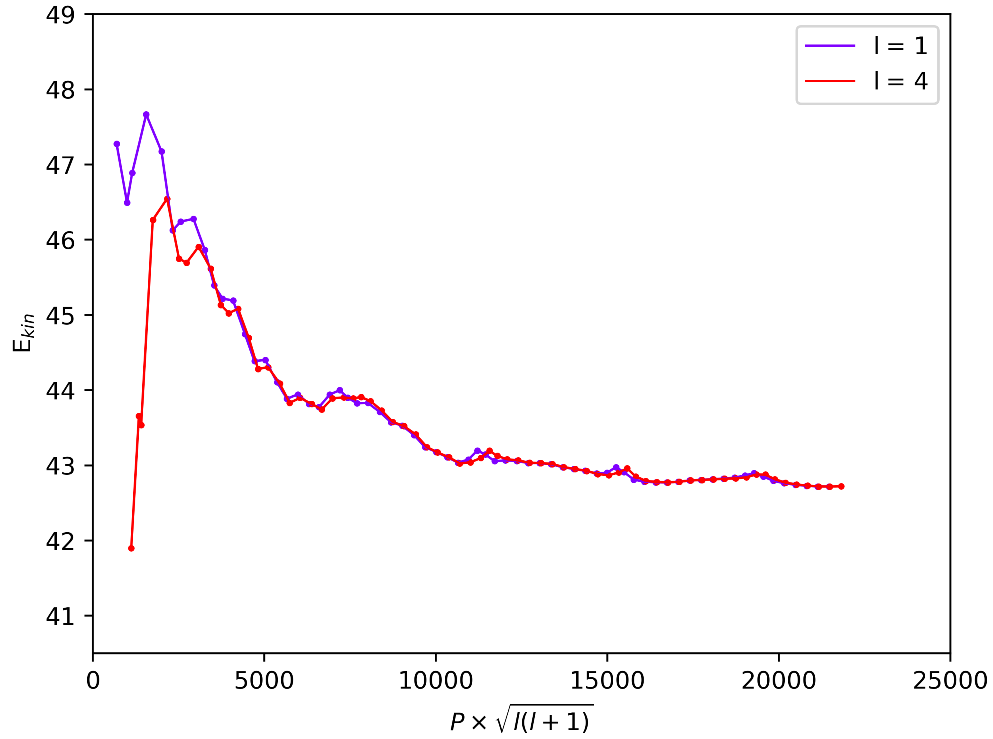


Fig. 3.8: Kinetic energy representation of the PULSE and STELUM parameters used in Fig. 3.1 and Fig. 3.2, taken at helium core abundance core_he = 0.9.

3.2 Origin of mode trapping from chemical transitions

3.2.1 Normal and trapped modes

The general cavity in which g-modes propagate is bounded by the Brunt-Väisälä and Lamb frequencies. However, some modes display a trapping phenomenon in a subset region of their propagation cavities, which alters the way a mode propagates with partial reflections at the boundaries of said subset. In Fig. 3.9, we give the juxtaposition of both the reduced period spectrum of Fig 3.7 and the kinetic energy of Fig 3.8, with which we aim to highlight that trends in the reduced period spectrum can be associated with trends in the kinetic energy. We zoom on the kinetic energy from Fig 3.8, as it highlights the local maxima of kinetic energy of high order modes while only loosing information on low order modes.

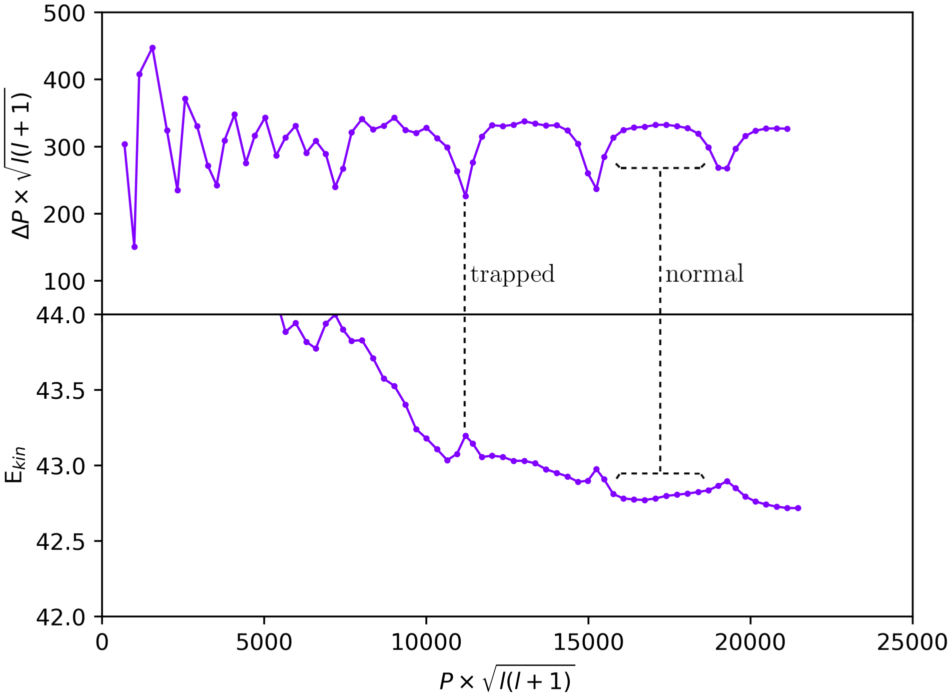


Fig. 3.9: Subplot of the reduced period spacing spectra and the kinetic energy spectra for modes of $k = 0$ to $k = 70$, mass $= 0.47M_{\odot}$, lq_env = -2, core_he = 0.9.

In this master thesis, we will contrast between trapped and normal modes. Distinguishing one from the other is first done through the properties of the kinectic energy and reduced period spacing spectrum. As mentioned in the previous subsection, the kinetic energy of a mode is directly dependent on how a given region of a star acts on the mode, in particular, if the mode is propagating more in the envelope, its kinetic energy will be lower than if it is propagating in the deeper regions of the star. Because of this, as shown in Brassard et al. (1992b), a mode trapped in the envelope, and for which the envelope thus has a strong influence on its frequency, will have lower kinetic energy the more the trapping phenomenon is effective, which will show as a local minimum of the kinetic energy spectrum. In our case, we do not find those local minima, but instead local maxima, which can be well seen on Fig. 3.8 and Fig. 3.9. For trapped modes, being associated to such local maxima thus means that the deeper regions of the star have a stronger influence, and that the mode trapping phenomenon occurs in those deeper regions, instead of the envelope. We will investigate this in more detail and determine the propagation cavity of these trapped modes in section 3.3.1.

By juxtaposing the reduced period spacing spectrum and the kinetic energy, we see that the local minima found in the reduced period spacing spectrum are in fact corresponding to local maxima of the kinetic energy, as shown for one trapped mode with the black dotted line in Fig. 3.9 denoted as "trapped". We thus understand that in the reduced period spacing spectrum, modes at a local minima are those which are trapped. This behavior contrasts with the one of what I call in this master thesis "normal modes". Normal modes are those for which the trapping phenomenon is not occuring, and as such, they do not correspond to either a minima or maxima on the kinetic energy spectrum, instead, those modes follow only the slow downward trend of the kinetic energy evoked earlier. On the reduced period spacing spectrum, those modes correspond to the dots on the upper part of the "arch-like" structure formed in between two trapped modes, and an example of which modes those are is displayed using the black dotted line denoted "normal" in Fig. 3.9.

As we can also tell on Fig. 3.9, some modes are between the trapped and normal modes. Actually, we have a continuous array of modes which are more or less trapped, with the bounds of this array being the trapped modes on one side, and normal modes on the other. Those modes, which can be found neighboring trapped ones, are subject in some respect to a trapping phenomenon, albeit less strong than trapped modes corresponding to an actual local minimum of the pulsation spectrum. In fact, this highlights that the trapping phenomenon induces a "compression" effect on the period of the modes surrounding trapped ones. This in turn results in the arch-like structures and pits founds in Fig. 3.9, since we alternate modes with close periods to one another, forming the pits in period spacing, and modes with periods spaced evenly, forming the top of the arches, that is normal modes.

3.2.2 Chemical transitions influence on node positioning

In subsection 1.4.2, we briefly touched on the fact that mode trapping is a consequence of a mode being partially reflected in a subset region of their propagation cavities, bounded by sharp transitions in the Brunt-Väisälä frequency. Such sharp transitions occur either when transitioning from a chemical composition to another, or when going from a convective region to a radiative one in the stellar structure. In our case, those sharp transitions are located when going from the convective core to the radiative mantle with the C-O/He transition, or from the radiative mantle to the envelope with the He/He+H transition. This subsection aims at studying the effects of the two aforementioned chemical transitions on the genesis of trapped modes.

Node pinching

A visualization tool which helps understand the global effects of a given chemical transition on modes is the so-called "propagation diagram", which can be found in Fig. 3.10 for a sdB model using mass = $0.47 M_{\odot}$, a C-O/He transition at lq_core = -0.25, a He/He+H transition at lq_env = -2, and for core helium abundance core_he = 0.9.

On a propagation diagram, we represent as red horizontal lines the frequency $\log(\sigma^2)$ of each mode of $l = 1$ from $k = 1$ to $k = 70$, as well as the zeros of the y_1 eigenfunction as red dots. Each dot is thus corresponding to one radial node of the mode we are considering. We first see that nodes are not evenly spaced, in fact, we find a greater number of nodes in the radiative mantle right before the He/H+He transition at lq_env = -2 than lower in the radiative mantle. Similarly, we find a low density of nodes after this same transition, in the envelope. This shows that while g-modes are able to propagate up until the Lamb frequency limit, here denoted by L_1 (Lamb frequency for $l = 1$), they are rather confined in the radiative mantle of the star. It is important to keep in mind that the node positioning of a mode directly tells us where the mode is propagating. Note in particular the complete absence of nodes above the

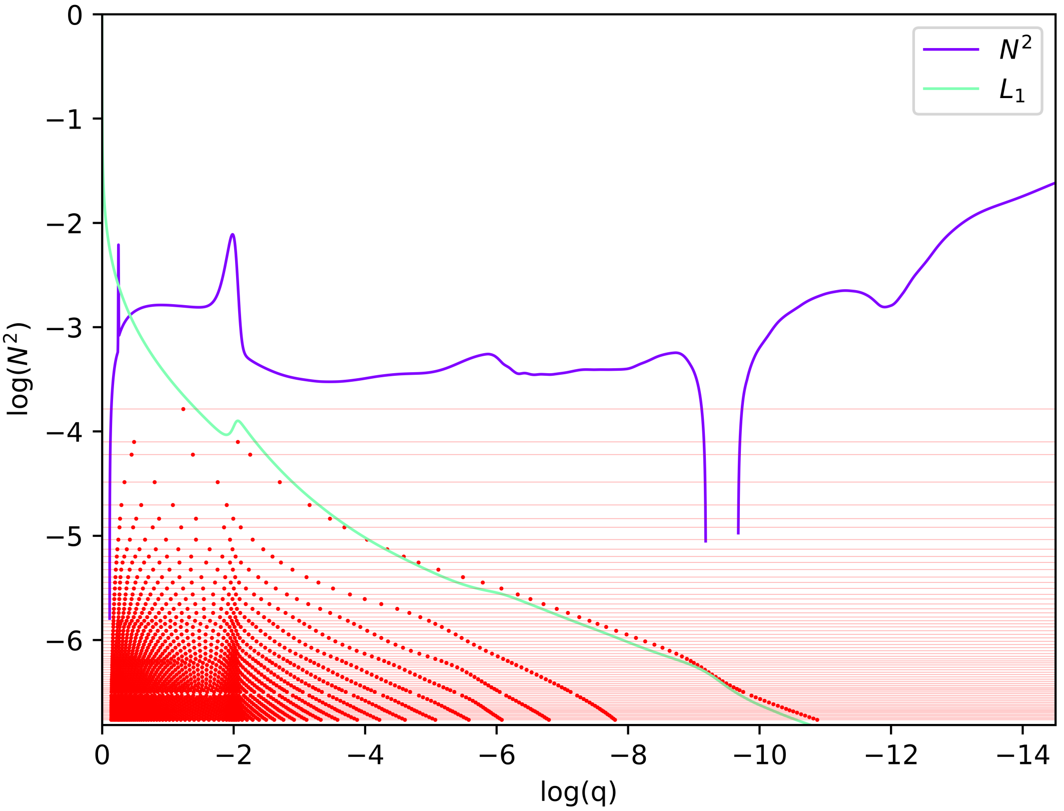


Fig. 3.10: Propagation diagram for mass = $0.47 M_\odot$, lq_core = -0.25 and lq_env = -2, at core_he = 0.9 and modes of $l = 1$ ranging from order $k = 0$ to $k = 70$. Red horizontal lines are the $\log(\sigma^2)$ of each mode and red dots indicate zeros of the y_1 eigenfunction.

Lamb frequency showing that the region above this frequency is an evanescent zone, in which we do not find an oscillating behavior for the modes, corroborating the theory seen in chapter 1.

We can view in more details the He/H+He transition zone using Fig. 3.11, which displays the same propagation diagram, this time centered and zoomed on said transition at $\log(q) = -2$. We can clearly see what I will call in this master thesis the "node pinching" phenomenon. This phenomenon occurs at chemical transitions, and we find that the density of nodes increases in the vicinity of such transitions, while relaxing away from them. In particular, this effect is at the origin of g-modes being confined in the radiative mantle of the star.

The downward trend of the kinetic energy at high radial orders (Fig. 3.8, Fig. 3.9) is well understood through the node pinching phenomenon as well. Indeed, in percentage, on Fig. 3.10 and Fig. 3.11 higher order modes have an increased number of nodes around the He/H+He transition and in the envelope, thus leading to lower kinetic energy, as those regions have a lesser density compared to deeper ones. It is shown in Charpinet (1999) that as a general trend, the n^{th} node of a g-mode of order k is further away from the surface than the $(n+1)^{\text{th}}$ node of a g-mode of order $k+1$. For the static models we are studying, it can be seen on Fig. 3.10 that while this trend is well verified in the envelope above $\log(q) = -2$, the node pinching occuring around the C-O/He and He/H+He transitions introduce a more chaotic behavior on the nodes positions in the radiative mantle. It is important to understand that the He/H+He transition at lq_env = -2 acts this way on every high order g-modes we compute, as such, we understand that this chemical transition is likely to not produce the mode trapping phenomena we observed in the pulsation spectrum, as we are looking for effects which are different for

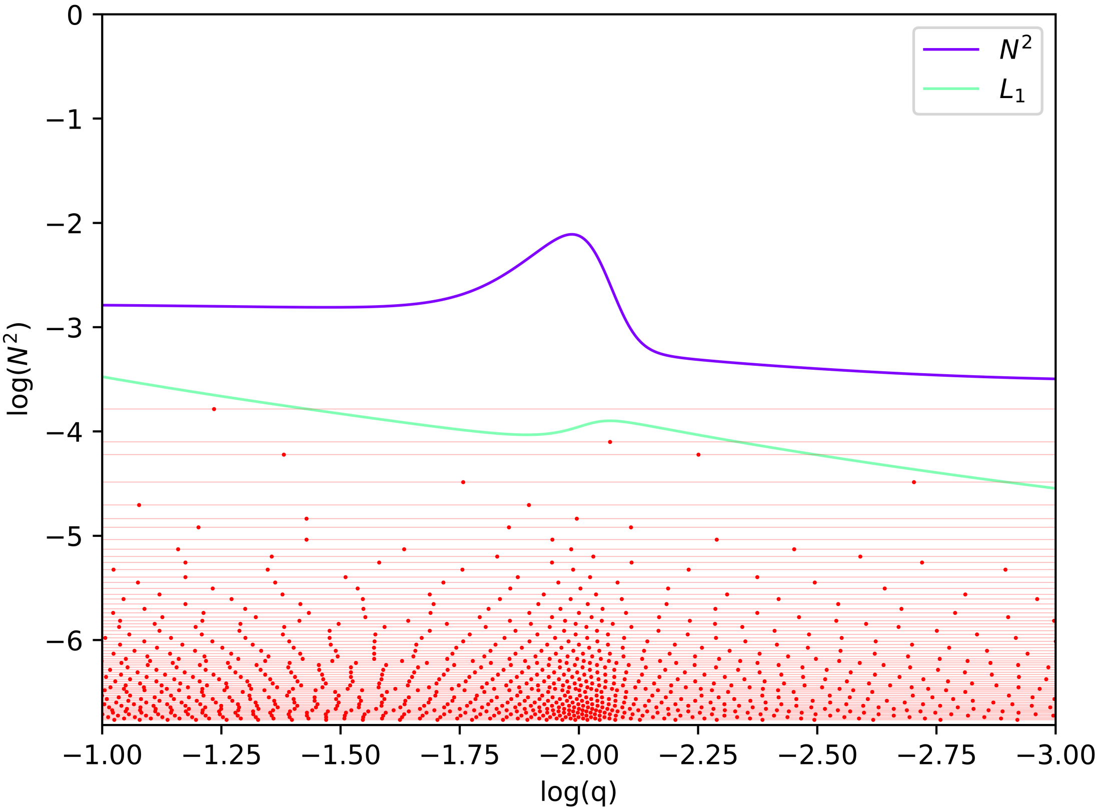


Fig. 3.11: Propagation diagram around He/H+He transition for mass $= 0.47M_{\odot}$ and lq_env = -2, at core_he = 0.9, for modes of $l = 1$ ranging from order $k = 0$ to $k = 70$. Red horizontal lines are the $\log(\sigma^2)$ of each mode and red dots indicate zeros of the y_1 eigenfunction.

normal and trapped modes.

We now investigate the C-O/He transition, in Fig. 3.12. In the top left part of this figure, we use the $m(r)$ scale to show the same propagation diagram as Fig. 3.10. This allows to highlight the C-O/He transition, located at $m(r) = 0.205M_{\odot}$, at the bump in the Brunt-Väisälä frequency. The top right part of the figure shows a zoom on the high radial order modes near the C-O/He transition. In this zoom, we highlight that the period compression due to the trapping phenomena is found again in the propagation diagram. We give numbers on the right of the zoomed part to show the correspondance between this period compression effect and the trapped mode associated to it, and those same numbers are given on the pulsation spectrum found at the bottom of Fig. 3.12. Since red lines in the propagation diagram correspond to lines of $\log(\sigma^2)$, when we have a period compression, aka a minimum of ΔP in reduced pulsation spectra, those lines will be closer to one another locally, which is what we observe.

Let us call n a given node of a radial order k, such that n varies from $n = 1$, which is the first node located at the center of star, to $n = k$, which is the last node toward the surface, and the nearest to the Lamb frequency. On Fig. 3.12, the propagation diagram shows that between $m(r) \approx 0.1M_{\odot}$ and $m(r) \approx 0.2M_{\odot}$, before the C-O/He transition, nodes of a given n follow each other in continuous lines. For example, the nodes $n = 1$ which are located at the very left of the propagation diagram, clearly follow each other across modes of different radial orders k. However, upon crossing the C-O/He transition, this continuous line is broken and forms a wave like pattern instead, where we alternate a vertical trend (sharp ascendant diagonal, but we use "vertical" as a mean of contrast) and a horizontal trend. An example of this is represented in green on the zoomed part of the propagation diagram (Fig. 3.12, top right), which links nodes

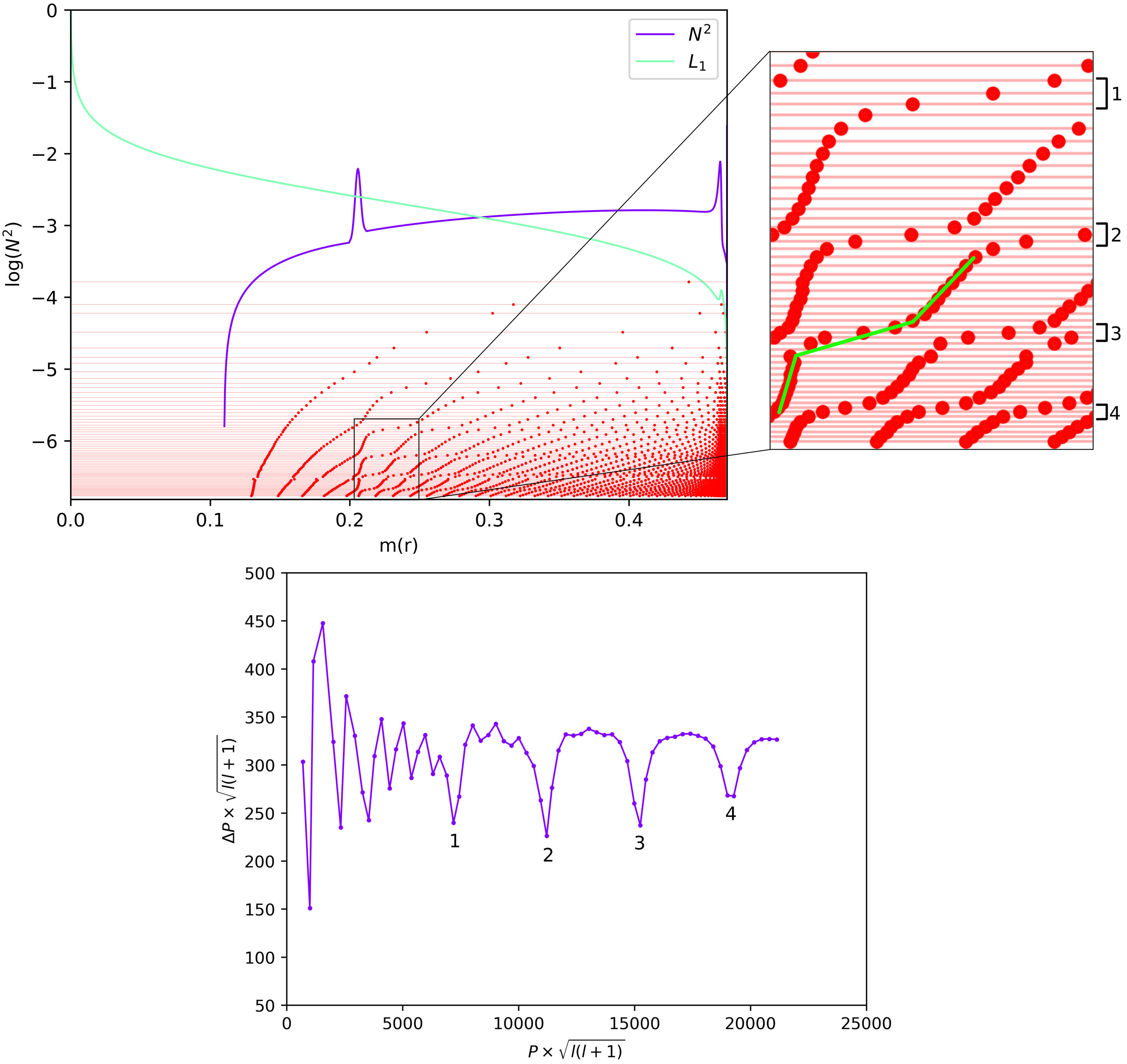


Fig. 3.12: Propagation diagram around C-O/He transition for mass = $0.47 M_\odot$ and lq_env = -2, at core_he = 0.9 and modes of $l = 1$ ranging from order $k = 0$ to $k = 70$. Red horizontal lines are the $\log(\sigma^2)$ of each mode and red dots indicate zeros of the y_1 eigenfunction.

$n = 5$. We first have the aforementioned vertical trend, followed by a near horizontal line, itself followed by a vertical trend again. The horizontal trend systemically occurs accross the period compression effect, which means it is associated with trapped modes, while the vertical trend is then associated with normal modes. The full propagation diagram of Fig. 3.12 also shows that this wave like pattern induces an offset of the nodes associated with trapped modes toward the surface, compared to the nodes associated with normal modes.

Figure. 3.13 displays the same type of figure as Fig. 3.12, except at core helium abundance core_he = 0.4 instead of core_he = 0.9. This gives a reference of what a model with a less strong trapping looks like (as we will see in section 3.3.1). The pulsation spectra of Fig. 3.13 (bottom) shows that we still find a form of trapping, but a less selective one (the minima are less deep compared to normal modes) than at core_he = 0.9. We still find the wave like pattern from the C-O/He transition, and an example of it is highlighted in green on the zoomed part of the propagation diagram. Again, we associate the horizontal trends that we see on these wave like patterns to the trapped modes on the pulsation spectra of Fig. 3.12 (bottom), with numbers. Comparing both zoomed parts of Fig. 3.12 and Fig. 3.13, we understand that the less a mode is trapped (aka, the less deep its minimum is in the pulsation spectrum), the less "horizontal" is the trend between two vertical trends (normal modes). Because of this effect, the nodes on the horizontal trends from Fig. 3.13 (core_he = 0.4) show less spread, meaning they are closer to each others than nodes on the horizontal trends from Fig. 3.12 (core_he = 0.9). Additionally, we do not have a visible offset of the nodes from trapped modes compared to normal modes anymore on Fig 3.13. We conclude from our study of the different propagation diagrams that the influence of the C-O/He transition is different according to the type of mode (trapped or normal), and thus that this chemical transition should be further investigated if we want to better understand mode trapping in the context of 4G models.

Chemical gradient influence

While propagation diagrams are helping to understand the general impact of chemical transitions on the pulsation spectra, we can visualize this in a more immediate way. To do this, we pulsate two models, one where we remove the C-O/He transition, and one where we remove the He/H+He transition, in order to see what impact this can have on our modes, and if mode trapping is still present after removing one or the other chemical transition. Here, "removing" a transition means that I force the Ledoux term B to zero in the region where we originally have a chemical transition. In Fig. 3.14 we show the results of removing those transitions against the normal spectrum (with both the C-O/He and He/H+He transitions) at core_he = 0.9. B(He/H+He) = 0 means we removed the He/H+He transition at lq_env = -2, while B(C-O/He) = 0 means we removed the C-O/He transition at lq_core = -0.25. As anticipated, the He/H+He transition leaves the trapped modes at high k orders unaffected, and mostly disturbs low orders modes. At the same time, we also see that removing the C-O/He transition drastically affects trapped modes, and they completely disappeared from our spectra.

Finally, we show the propagation diagram associated with the removal of the C-O/He transition in Fig. 3.15. The horizontal trends associated with trapped modes on the propagation diagram of Fig 3.12 (strongly trapped modes), which were becoming increasingly vertical in Fig. 3.13 (shallower trapped modes), are now fully vertical, and there is no wave like pattern anymore. This experiment of removing transitions reinforces our first investigation on the influence of each chemical transition on trapped modes through propagation diagrams, and reinforces that further studies of trapped modes should be focused around the C-O/He transition.

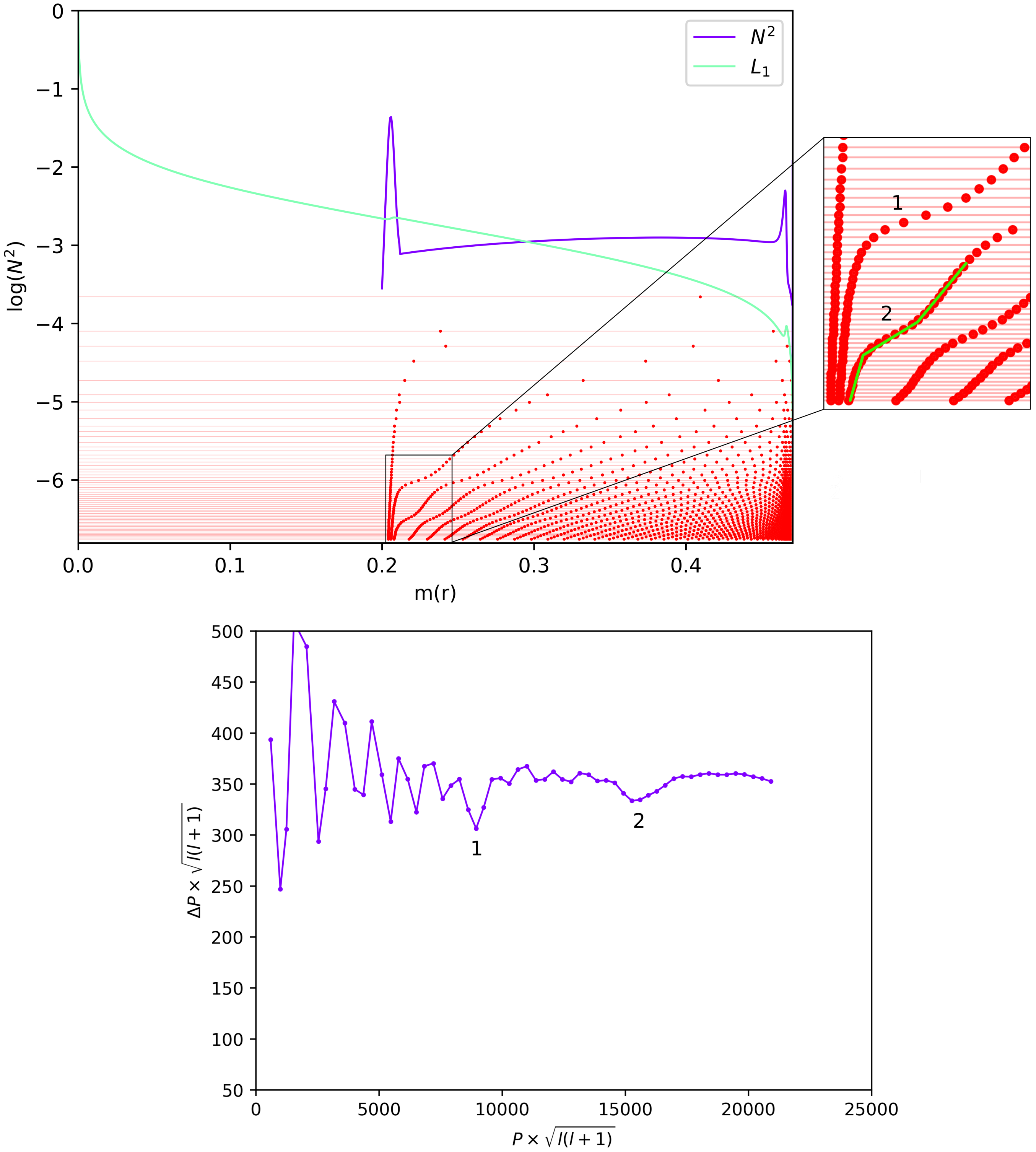


Fig. 3.13: Propagation diagram around C-O/He transition for mass = $0.47 M_\odot$ and lq_env = -2, at core_he = 0.4, for modes of $l = 1$ ranging from order $k = 0$ to $k = 70$. Red horizontal lines are the $\log(\sigma^2)$ of each mode and red dots indicate zeros of the y_1 eigenfunction.

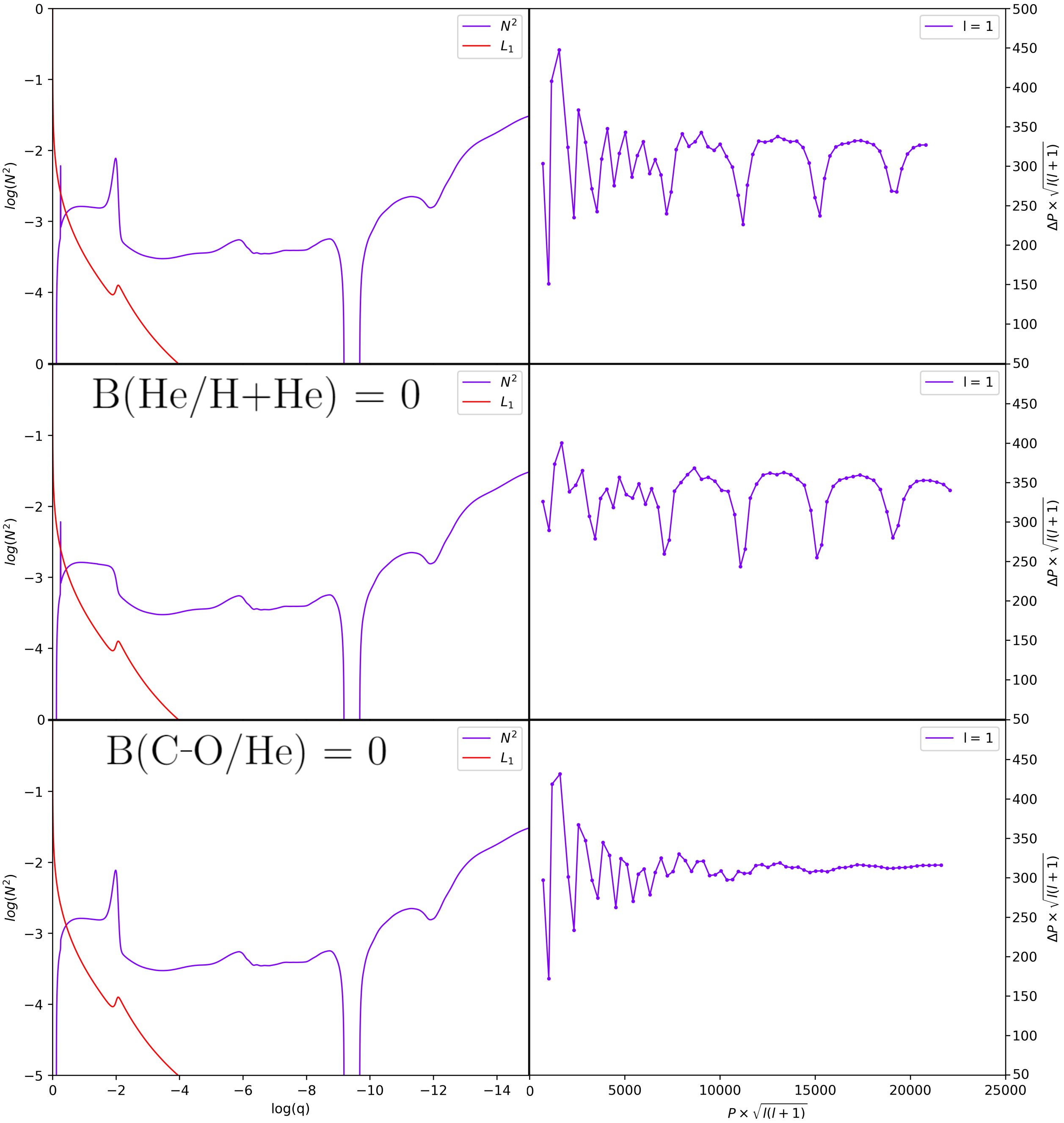


Fig. 3.14: Brunt-Väisälä frequencies and their associated reduced period spacing spectra. From top to bottom, we find : the unmodified model ; a model where the He/H+He transition is removed ; a model where the C-O/He transition is removed. Parameters are mass = $0.47 M_\odot$, lq_core = -0.25 and lq_env = -2, at core_he = 0.9.

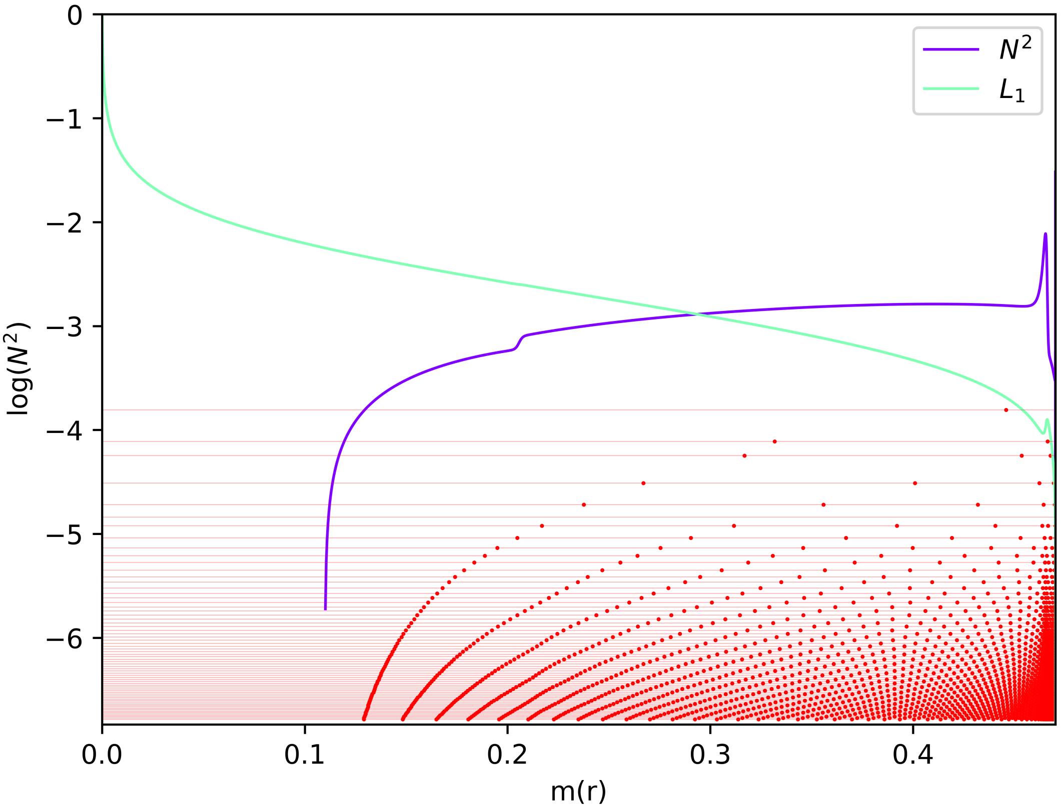


Fig. 3.15: Propagation diagram where the effects of the C-O/He transition have been removed. We use mass = $0.47M_\odot$, lq_core = -0.25 and lq_env = -2, at core_he = 0.9 and modes of $l = 1$ ranging from order $k = 0$ to $k = 70$. Red horizontal lines are the $\log(\sigma^2)$ of each mode and red dots indicate zeros of the y_1 eigenfunction.

3.3 Evolution of the pulsation spectrum

3.3.1 Influence of the core helium abundance

As we have seen, mode trapping is a direct consequence of the C-O/He transition. In particular, we could anticipate that the stronger this transition is, then the stronger the trapping becomes if it is the chemical gradient itself that is producing mode trapping. During the evolution of the star, helium is fused in the core to produce carbon, as such, in our models, the lower the helium core abundance core_he is, the higher the Ledoux term B(C-O/He) becomes. We would thus anticipate that lower core_he is then equivalent to pulsation spectra with more strongly trapped modes. While this is the case for evolutionary models as we will see in chapter 4, for static 4G models, this tendency is not verified. We give in Fig. 3.16 the evolution of pulsation spectra, from core_he = 0.1 to core_he = 0.9, and their associated kinetic energy directly below, for lq_env = -2, lq_core = -0.25, and mass = $0.47M_{\odot}$. This indicates that while it is now firmly established that mode trapping originates from the C-O/He transition (see Fig. 3.14), the detailed mechanism is slightly more subtle than the strength of the chemical transition alone.

While the C-O/He transition becomes stronger, Fig. 3.16 shows that minima of reduced period ΔP in pulsation spectra are progressively becoming shallower from core_he = 0.9 to core_he = 0.4. The aspect of the pulsation spectrum then remains globally the same from core_he = 0.4 to 0.1. This is an indication that the mode trapping phenomena is progressively weakening with decreasing core helium abundance in 4G models, up until core_he = 0.4. In order to find the explanation to this phenomenon, we need to interest ourselves in the weight function of trapped modes and normal modes, both at high core_he > 0.5 and at lower core_he < 0.5. As explained in the previous chapter, the weight function will allow us to know which regions are contributing most to the frequency of our modes, and by extension, which zones have the strongest influence on the period of our modes. Those weight functions are displayed in Fig. 3.17 at core_he = 0.8 and core_he = 0.1.

For the two weight functions at core_he = 0.8, we directly see the difference between a trapped and a normal mode. For the trapped mode, there is a zone below the C-O/He transition that has a strong influence on the frequency of the modes. Keep in mind however that when discussing weight functions, one can only make relative comparisons, as such, what is actually important to notice is that the weight function in the zone below the C-O/He transition at $m(r) \approx 0.2M_{\odot}$ has a stronger amplitude than the weight function in the radiative mantle for the trapped mode, while it is the opposite for the normal mode at core_he = 0.8. For core_he = 0.1, we instead find that the amplitude under the C-O/He transition is nearly zero.

The study of the size of the core that is convective as a function of the core helium abundance reveals that it grows as the core helium abundance diminishes, which is showed on Fig. 3.18. On this figure, the red horizontal line represents the C-O/He transition at lq_core = -0.25, while the green vertical line indicates when core_he = 0.5. When almost no helium has been burned at core_he = 0.985, the convective core ranges from the center of the star at log(q) = 0 to log(q) $\approx$ -0.12. As the core helium abundance decreases, we observe a slow growth of the convective core until about core_he = 0.5, where we observe a jump in convective core size from log(q) $\approx$ -0.16 to log(q) $\approx$ -0.25. The blue dots in the figure each represent a given model, ranging from core_he = 0.985 to core_he = 0.05, with decrements of 0.05, as such the jump in convective core size occurs over a single model. This jump has seemingly no influence on the pulsation spectra we observe, since the trapped modes are already well dampened when this jump occurs. While the convective core grows, the C-O/He transition remains fixed at lq_core = -0.25 in static models, and does not follow this growth. Because this is the case, we understand that the

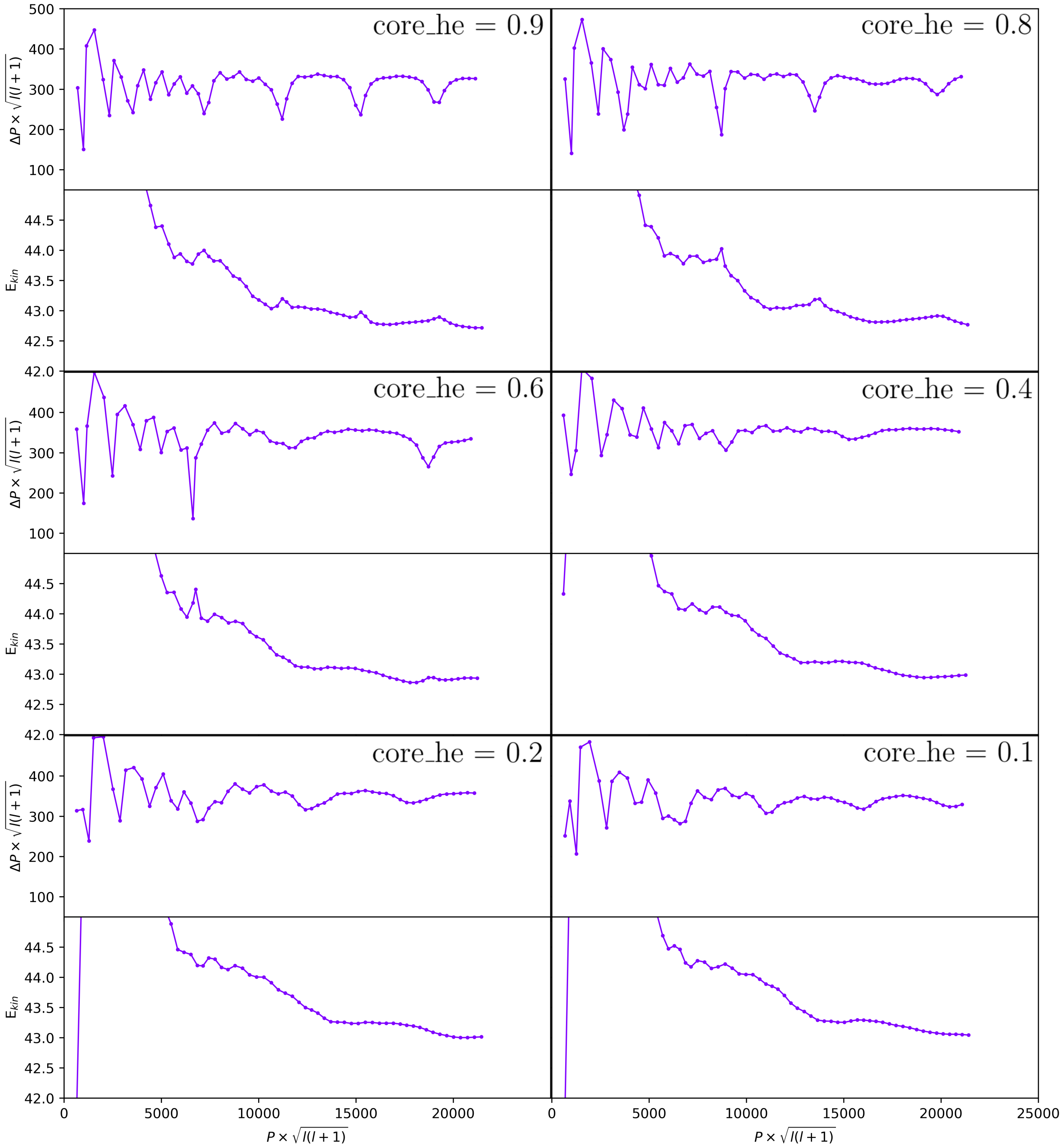


Fig. 3.16: Pulsation spectra and associated kinetic energy for different core_he, indicated on the figure.

radiative zone under lq_core, which is the main contributor to the frequencies of our trapped modes, is progressively replaced by the convective core, in which no modes can propagate (see theory of oscillations in chapter 1). The vanishing of strongly trapped modes (as opposed to the weaker trapping at low core helium abundance) is thus entirely due to the disappearence of the radiative zone under the C-O/He transition.

We noted earlier that we can still observe a weak form of trapping from the C-O/He transition for low core_he. We can indeed pinpoint that this weak form of trapping is still from the C-O/He transition, as the pulsation spectra at low core_he found in Fig. 3.16 present this weak form of trapping, while those displayed when the C-O/He transition is removed (with the Ledoux term B(C-O/He) = 0) in Fig. 3.14, do not. This weak trapping, with respect

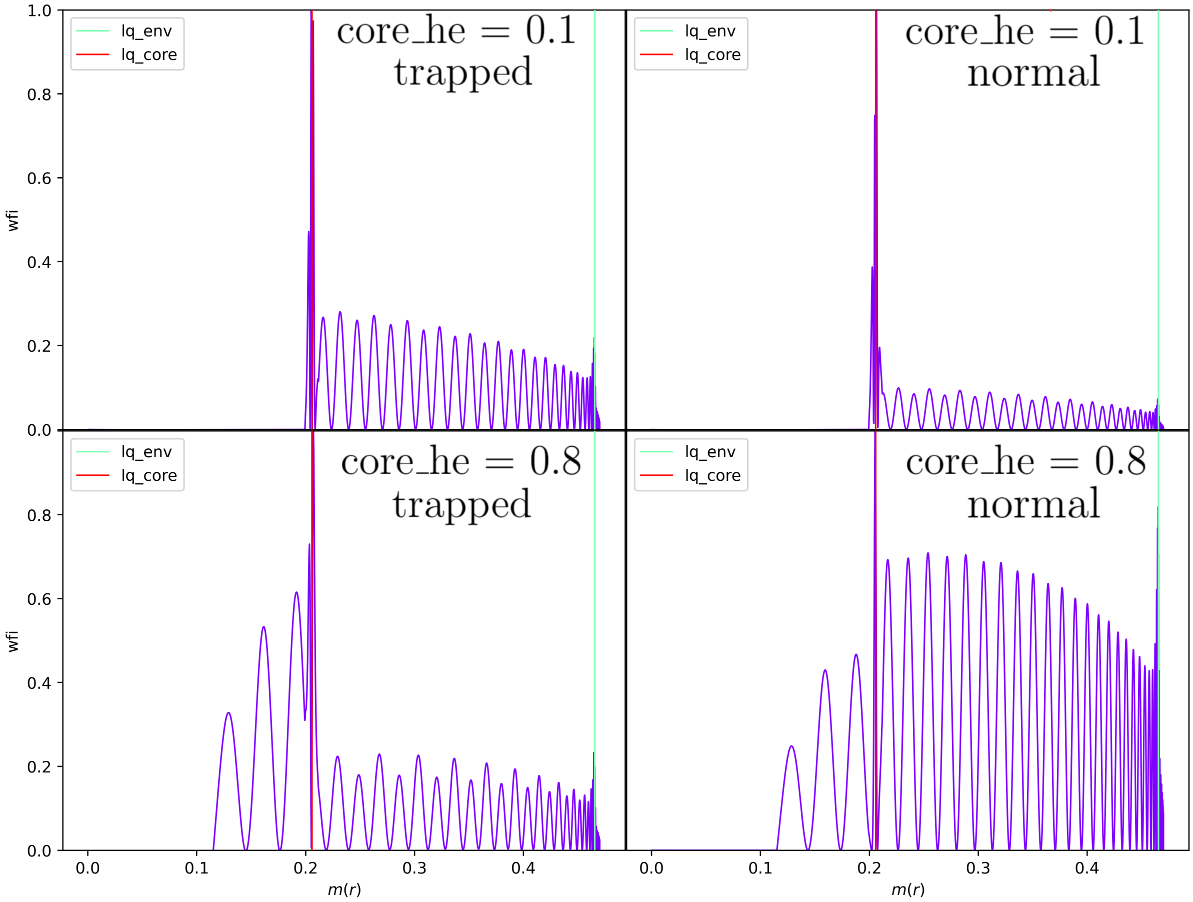


Fig. 3.17: Weight function of modes. Top left : a trapped mode at core_he = 0.1 ; top right : a normal mode at core_he = 0.1 ; bottom left : a trapped mode at core_he = 0.8 ; bottom right : a normal mode at core_he = 0.8. Model parameters : mass = 0.47, lq_env = -2, lq_core = -0.25.

to the usual trapped modes at high core_he, might be due to the C-O/He transition increasing in strength the lower the core helium abundance is, although this calls for another set of experiences out of the scope of this master thesis.

It is notable that the intensity of the local maxima of the kinetic energy spectra for core_he ≥ 0.6 shown in Fig. 3.16 are a direct consequence of our trapped modes having a strong weight function in the small radiative region very deep in the star, just below lq_core = -0.25. Indeed, while all g-modes of our pulsation spectrum are confined in the radiative mantle, only the trapped modes for such core helium abundances have a non-negligible amplitude of the weight function in the small radiative region below the C-O/He transition. Because it is in the deeper regions of the star, we thus have trapped modes of higher kinetic energy than normal ones.

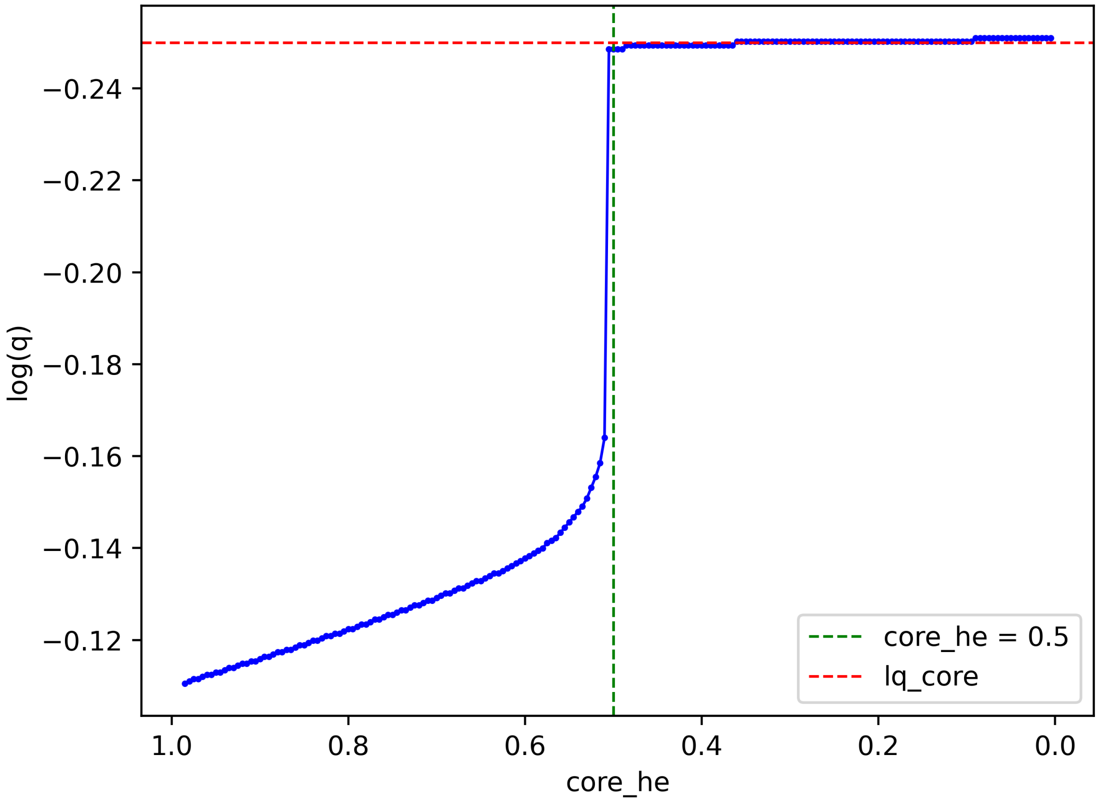


Fig. 3.18: Upper limit of the convective core in function of core_he. The green dashed line represents when core_he = 0.5, while the red dash line represents the C-O/He transition at lq_core = -0.25. We show in blue the growth of the convective core, and the blue dots are each representative of a single model.

3.3.2 Influence of the main parameters on pulsation spectra

In this subsection, we explore the influence of the three main parameters of the static models, the mass of the star, of the envelope (aka the lq_env parameter), and of the core region (aka the lq_core parameter), on the resulting pulsation spectra, and on the presence of trapped modes in particular.

Mass of the star

In Fig. 3.19, we give the pulsation spectra of different mass parameters with mass = 0.40, 0.47 and 0.50 $M_{\odot}$, for a fixed He/H+He transition at lq_env = -2, and for two different core helium abundances, core_he = 0.9 and core_he = 0.1. Changing the mass of our models does not fundamentally change the overall aspect of the pulsation spectra we observe, and has no influence on the presence of trapped modes. Instead, the pulsation spectra translates upwards and to the right with increasing mass. It is expected that our pulsations remain of the same nature since we do not affect the chemical transitions positions by changing the mass of the star. An upwards translation means here that the period spacing between our modes is increasing with mass, while the translation to the right indicates that our modes have higher periods with increasing mass.

This is well explained when considering what increasing the mass of the star actually means in the context of our static models. In our graphs, we often use the $\log(q) = \log\left(1 - \frac{m(r)}{M_*}\right)$ scale, which is unable to tell us the actual size of the different zones we look at. In Fig. 3.20,

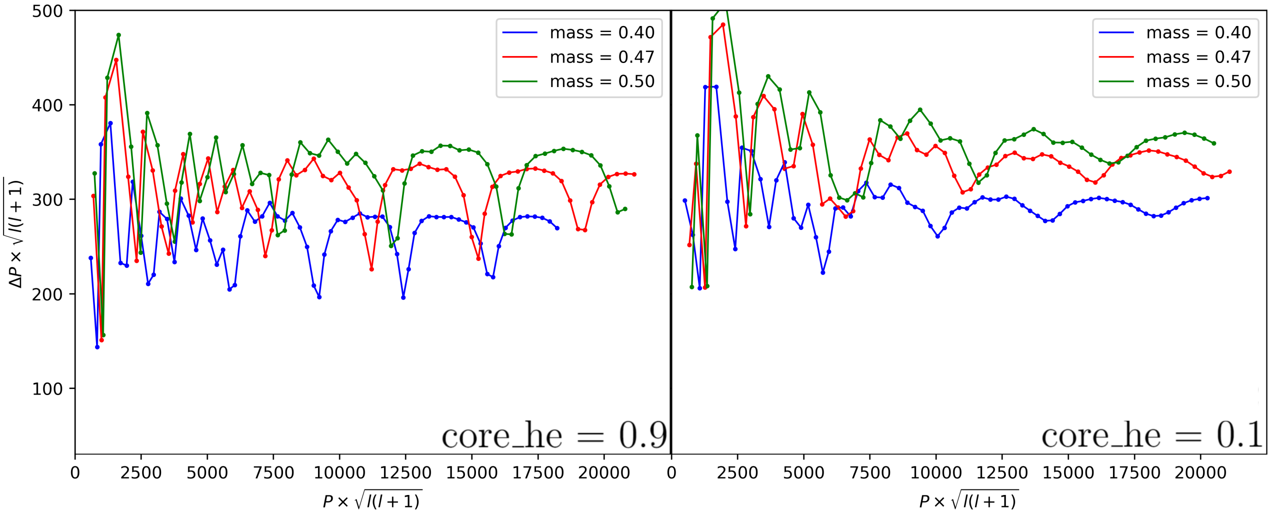


Fig. 3.19: Influence of the mass parameter on pulsation spectra, shown for three different masses, in green $0.50 M_\odot$, in red $0.47 M_\odot$ and in blue $0.40 M_\odot$ (all models have lq_env = -2 and lq_core = -0.25). We show in the left colum core_he = 0.9, and in the right column core_he = 0.1.

we show the impact of increasing the mass on the actual size of the zones we are considering, for lq_env = -2 and at core_he = 0.9. What actually happens when the mass is increasing, is that we mainly grow the size of the radiative mantle and the envelope. Because our modes propagate until Lamb's frequency, this rise in size will directly mean that our modes will have to travel a longer distance, causing them to have a longer period in average. As for the increase in reduced period spacing, we will view in subsection 4.3 that the period spacing of our modes directly depends on the size of the region they propagate in, and a region of higher size will indeed result in a higher period spacing.

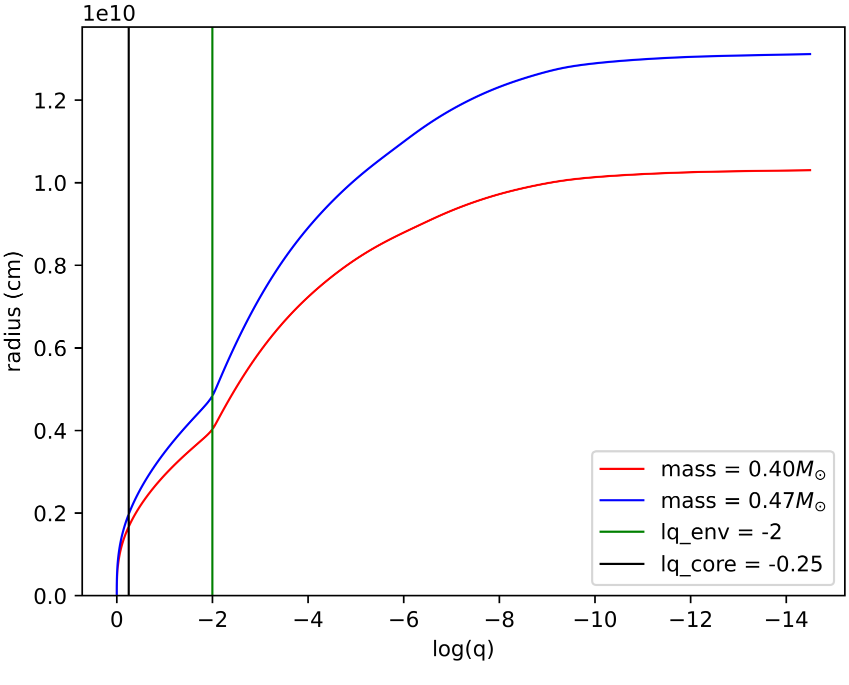


Fig. 3.20: Impact of changing the mass on the radius of the different regions of the star. Parameters : core_he = 0.9, lq_env = -2 ; lq_core = -0.25.

Mass of the envelope

The influence of the mass of the envelope, that is, of the parameter lq_env, is different in so far that it seems to clearly modify the behavior of our modes. In Fig. 3.21, we represent the difference between the modes from changing the position of the He/H+He transition lq_env while keeping the total mass constant. Compared to the lq_env = -2 pulsation spectra analysed until now, the lq_env = -4 spectra seems more "chaotic", in the sense that we see variations of period spacing ΔP from modes to modes (i.e, between adjacent modes), while the general trend, and in particular the presence of minima of trapped modes at core_he = 0.9, are conserved.

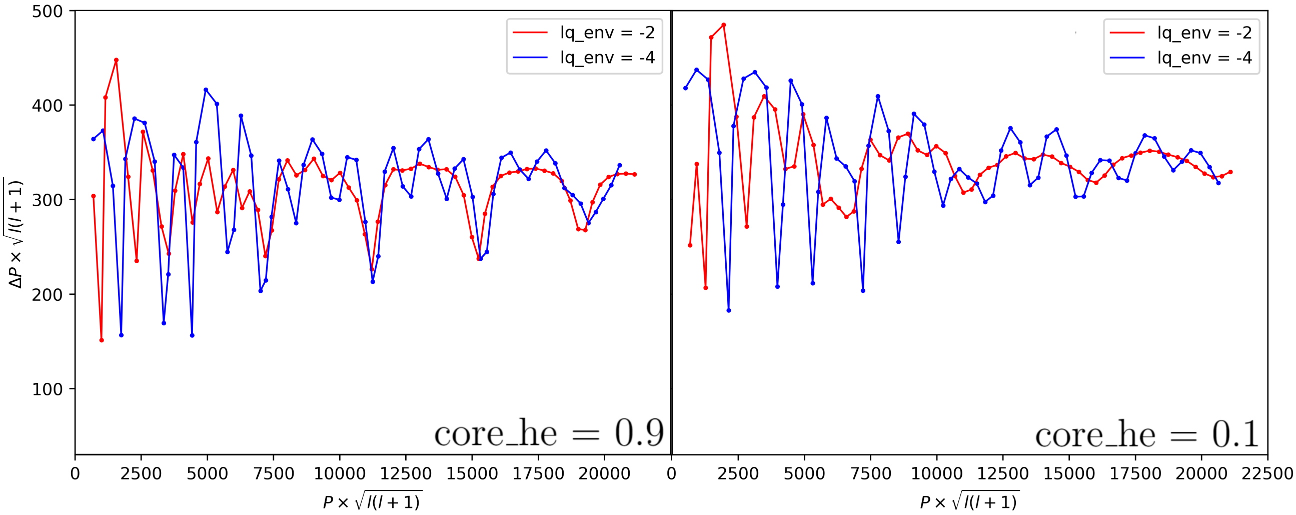


Fig. 3.21: Influence of the envelope's mass on different pulsation spectra, with in red lq_env = -2 and in blue lq_env = -4. Left column is for core_he = 0.9, right column for core_he = 0.1. Parameters of the model : mass = $0.47 M_{\odot}$; lq_core = -0.25.

A first insight into why this is the case is gained by looking at the kinetic energy of those modes. In Fig. 3.22, we display a pulsation spectrum of lq_env = -4 at core_he = 0.9 with its associated kinetic energy, as well as the propagation diagram corresponding to this model. It is directly noticable that instead of a near constant kinetic energy with local maxima at high k order (as was the case for lq_env = -2, see Fig. 3.16), we now find a kinetic energy which follows the oscillations of the reduced period spectrum. The kinetic energy now presents local minima which correspond to local minima of the pulsation spectrum, and now display an oscillating pattern. We anticipate the envelope to have a stronger influence at lq_env = -4 than before at lq_env = -2 from this change in behavior. On Fig. 3.22, the propagation diagram clearly shows that we still have a node-pinching around the He/H+He transition at lq_env = -4, although less strong than for lq_env = -2, and that most nodes are still contained in the radiative mantle. Even though the Lamb frequency is shifted toward the surface due to lq_env being higher, the number of nodes above the radiative mantle cannot justify the seemingly stronger impact the envelope would have on our modes.

The expression of the weight function, as well as the expression of the kinetic energy, are dependent on ρr^2, as was seen in equations (2.17) and (2.18). In Fig. 3.23, we give this term's value for both lq_env = -2 and -4. It is directly seen that term is very similar for both lq_env. As expected, the C-O/He transition at lq_core is the point of strongest ρr^2, which supports that the local maxima in kinetic energy associated to modes trapped in the small region under lq_core, are indeed directly related to those particular modes propagating in the radiative region of high density located under lq_core. For any model, we find a node-pinching phenomenon around the He/H+He transition at lq_env, and the proportion of nodes above and under this

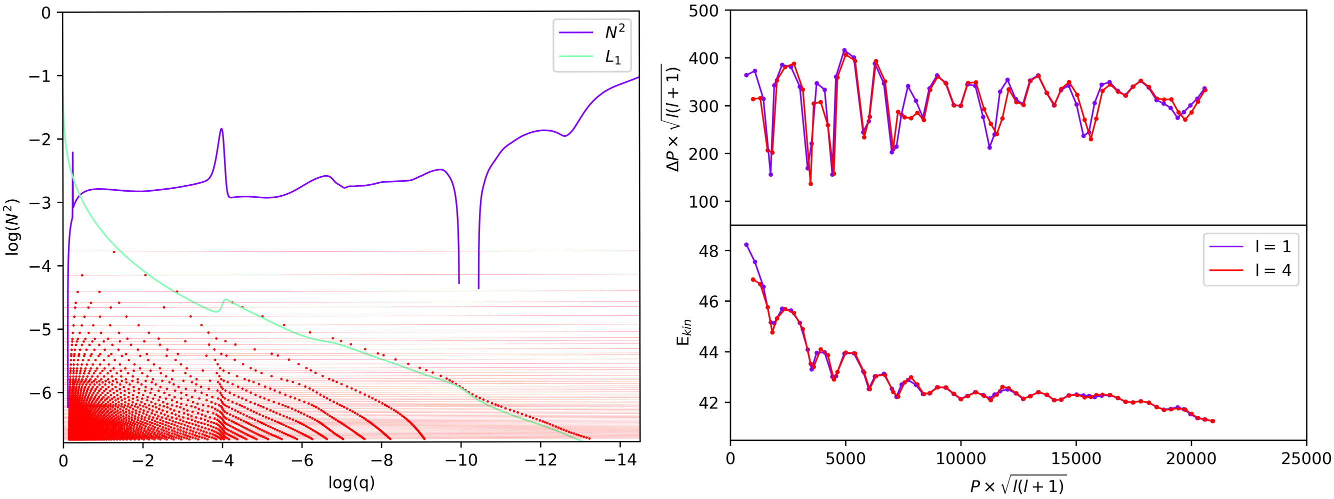


Fig. 3.22: Left : propagation diagram of a lq_env = -4 model. Right : Subplot of the reduced period spacing spectra and kinetic energy associated to the same model. Parameters : mass = 0.47 $M_\odot$, lq_env = -4, core_he = 0.9, lq_core = 0.9.

transition is the same across all of our different lq_env values. However, this means that for lq_env = -2, most nodes of high order k modes are in regions of higher ρr^2 compared to the case of lq_env = -4. In this latter case, in proportion, a higher part of the number of nodes of a mode lies in a region of lower ρr^2, giving rise to a lower kinetic energy modes.

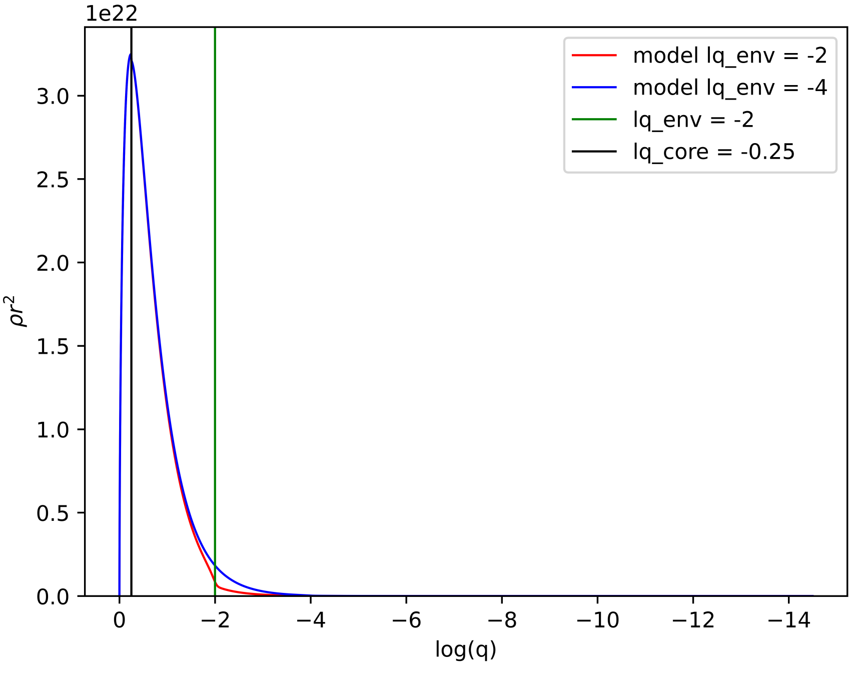


Fig. 3.23: ρr^2 term in function of log(q), for lq_env = -2 and -4. We display vertical lines for the C-O/He transition and He/H+He transition of a lq_env = -2 model.

The oscillating pattern seen in pulsation spectra could be an indication of a form of weak trapping from the He/H+He transition, as the influence of this chemical transition on our modes is stronger from the fact that more nodes are now located in regions of lower density. We proceed with an experiment and remove the He/H+He transition (that is taking the Ledoux term B(He/H+He) to zero) to confirm this. The result of this experiment is shown in Fig. 3.24 (middle panel), where indeed, removing the He/H+He transition strongly dampens the oscillating pattern we observe for the unmodified model (top panel).

However, some residual oscillations are still observable (albeit at a much lower level). We extend the experiment to remove the upper chemical transition from He+H envelope to pure H envelope (controlled by the lq_diff parameter, here called H+He/H transition), which could have an influence on the residual oscillating pattern. The result of this experiment is shown in the bottom panel of Fig. 3.24, and shows that this higher transition zone has no impact on the pulsation spectrum.

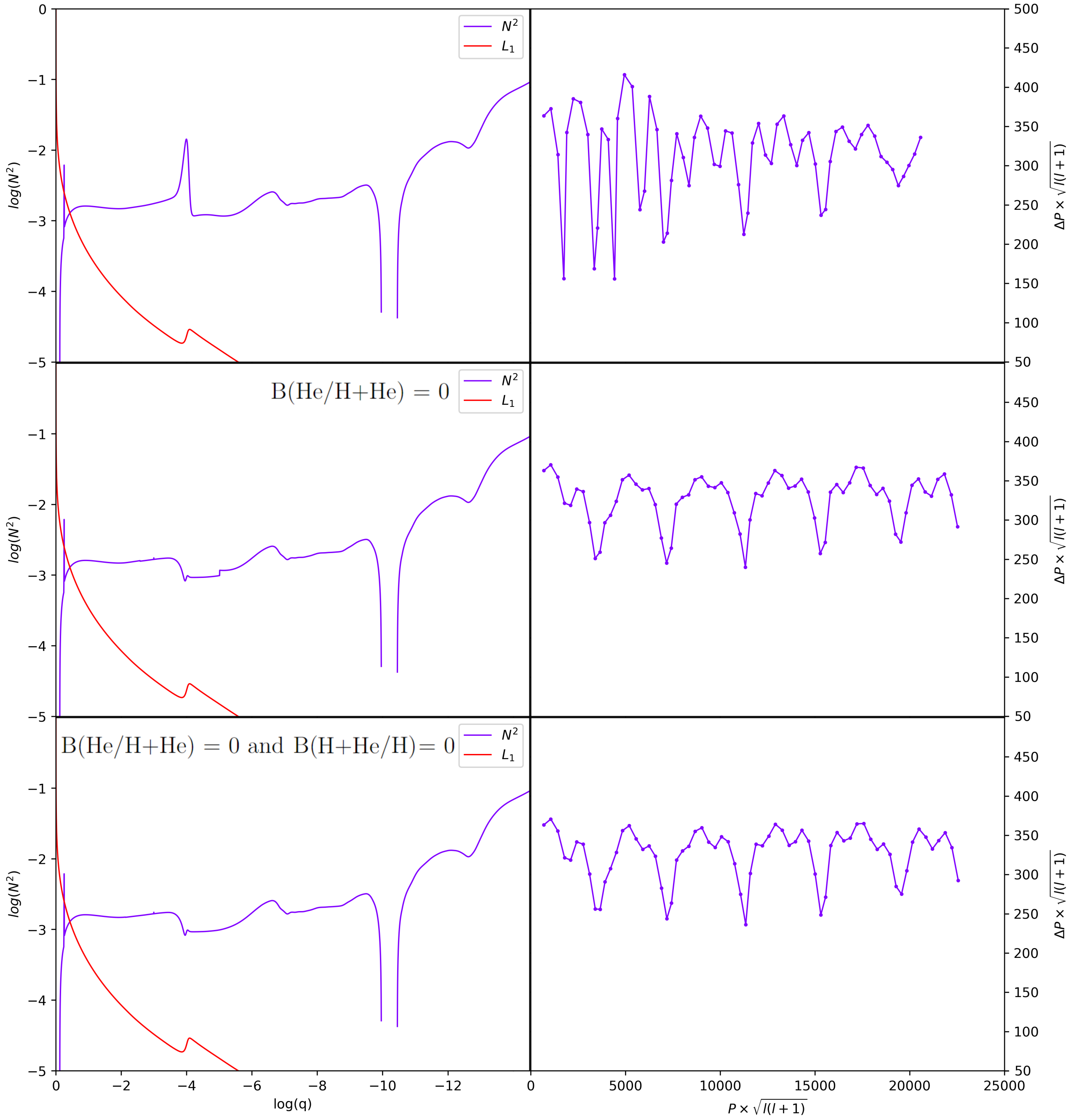


Fig. 3.24: Brunt-Väisälä frequencies and their associated reduced period spacing spectra. From top to bottom, we find : the unmodified model ; a model where the He/H+He transition is removed ; a model where the He/H+He transition and the He+H/H transition are both removed. Parameters are mass = $0.47 M_\odot$, lq_core = -0.25 and lq_env = -4, at core_he = 0.9.

Mass of the core

In this master thesis, so far, the C-O/He transition was always fixed at lq_core = -0.25. In this subsection, we aim to vary this parameter in order to see its influence on pulsation spectra. We show the results for the C-O/He transition lq_core = -0.10, -0.25, -0.40 in Fig. 3.25, and in this figure, the left column corresponds to core helium abundance core_he = 0.9 while the right column to core_he = 0.1. Then, in descending order, we find the pulsation spectra and associated kinetic energies for C-O/He transitions at lq_core = -0.40, -0.25 and -0.10. In all of these models, the He/H+He transition is fixed at lq_env = -2. Note that in this master thesis, we will use "high" or "low" lq_core in view of its absolute value, hence, lq_core = -0.40 is high while lq_core = -0.10 is low. This is to be aligned with the associated mass of the core, since then a high lq_core corresponds to a high core mass with respect to a low lq_core, which then corresponds to a low core mass. The same ideology is applied to the mass of the envelope given by lq_env.

Let us first consider the lq_core = -0.10 spectra (bottom panels of Fig. 3.25). As we previously saw, the convective core for lq_core = -0.25 progressively grows in size when the core helium abundance decreases, until it reaches the C-O/He transition at about core_he = 0.5, at which point we have a completely convective zone under the C-O/He transition. In the case of lq_core = -0.10, the convective core does not need to grow, as it already reaches the C-O/He transition right from the very beginning, that is at core_he = 0.985. As a consequence, the pulsation spectra we anticipate to observe are ones which ressemble low core_he at lq_core = -0.25. For easier comparison, a pulsation spectra at lq_core = -0.25 and core_he = 0.1 is given in the middle right panel of Fig. 3.25. At high core_he for lq_core = 0.10, we observe weak trapping of isolated modes, which is is in direct contrast with what is found for high core_he at lq_core = -0.25. This is expected, as for lq_core = -0.10, the radiative zone under the C-O/He transition which is at the origin of the mode trapping for lq_core = -0.25 with high core helium abundance, does not exist, and thus no modes can be trapped in it. At low core helium abundance core_he, the same behavior is found for both lq_core, however for lq_core = -0.10, the spectrum shows modes trapped slightly more deeply. The weight functions of two trapped modes, at lq_core = -0.10, for both core_he = 0.1 and 0.9, are displayed in Fig. 3.26. Comparing those weight functions to the ones of trapped modes at lq_core = -0.25 and core_he = 0.1 in Fig. 3.17, we understand that the trapped modes found for lq_core = -0.10 at low or high core_he are actually of the same nature as trapped modes found for lq_core = -0.25 and low core_he. Finally, for lq_core = -0.10, finding deeper minima at low core helium abundance than at high core helium abundance corroborates the role of the C-O/He transition in mode trapping, as we have established earlier that this chemical transition becomes stronger as core helium abundance decreases in the convective core, thus providing stronger trapping.

We now consider the lq_core = -0.40 spectra (top panels of Fig. 3.25). First, we compare the lq_core = -0.40 and lq_core = -0.25 spectra at core_he = 0.9. In those spectra, we have not only a lot more trapped modes for lq_core = -0.40, but they are also deeper, which is expected. Indeed, at this stage of core helium abundance, both models have a radiative zone under the C-O/He transition, which we demonstrated was the culprit for mode trapping at those high core_he, and for lq_core = -0.40, the size of this radiative zone is bigger than for smaller cores. In Fig 3.27, with the comparison of the propagation diagrams for both lq_core, we show, in $m(r)$ scale, the increased size (understand mass fraction of the total star) of the radiative zone under the C-O/He transition (which happens at $m(r) \approx 0.2M_\odot$ at lq_core = -0.25 and $m(r) \approx 0.28M_\odot$ at lq_core = -0.40). In the same figure, we see that modes have a greater amount of nodes below the C-O/He transition for lq_core = -0.40, which directly implies that those modes are trapped deeper due to the modes having more nodes in regions of higher density. Now looking at the lq_core = -0.40 and core_he = 0.1 spectrum, we find strongly trapped modes, in opposition to

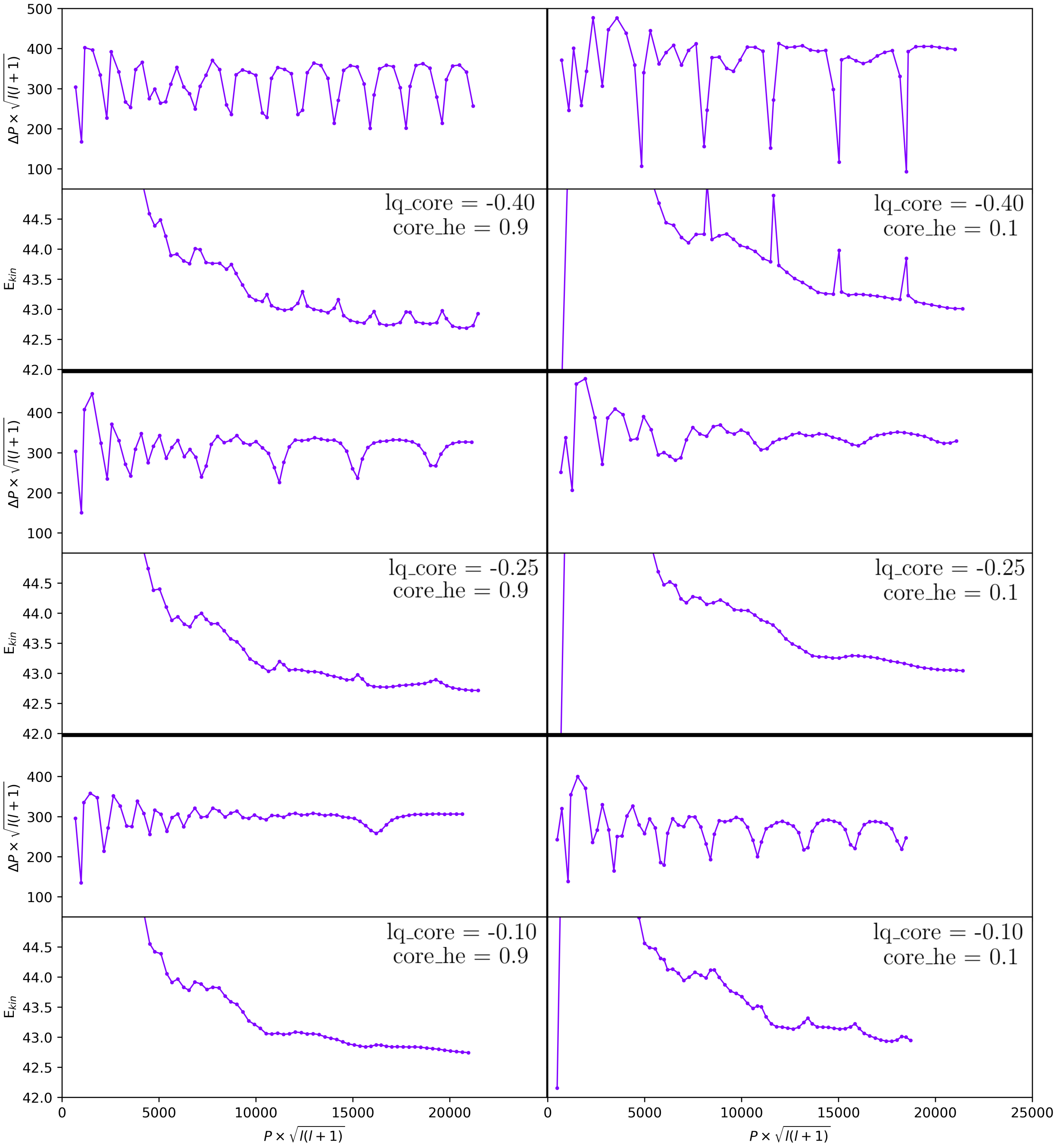


Fig. 3.25: Period spacing spectrum and associated kinetic energies. From top to bottom, we find lq_core = -0.10, -0.25 and -0.40. From left to right, we find core_he = 0.9 and 0.1. Parameters are mass = $0.47 M_{\odot}$ and lq_env = -4

spectra for other core masses at this low core helium abundance. This behavior is understood by looking at the Brunt-Väisälä frequency, which is given in Fig. 3.28, for lq_core = -0.25 with core_he = 0.6 on the left, and lq_core = -0.40 with core_he = 0.40 on the right. We see what I call a “radiative arch” under the C-O/He transition, and separated from said transition by a convective zone. This radiative arch is only temporary for models of lq_core = -0.25 (it persists from core_he = 0.6 to core_he = 0.5), and in fact causes the jump we saw in convective core size for those models due to its sudden disappearence (in practice, it srinks from core_he = 0.6 to core_he = 0.5, then disappears once sufficiently small). However, for models of lq_core = -0.40, the arch persists from core_he = 0.4 to core_he = 0.05 included. By studying the weight function of a trapped mode from lq_core = -0.40 with core_he = 0.4, we find in Fig. 3.29 that

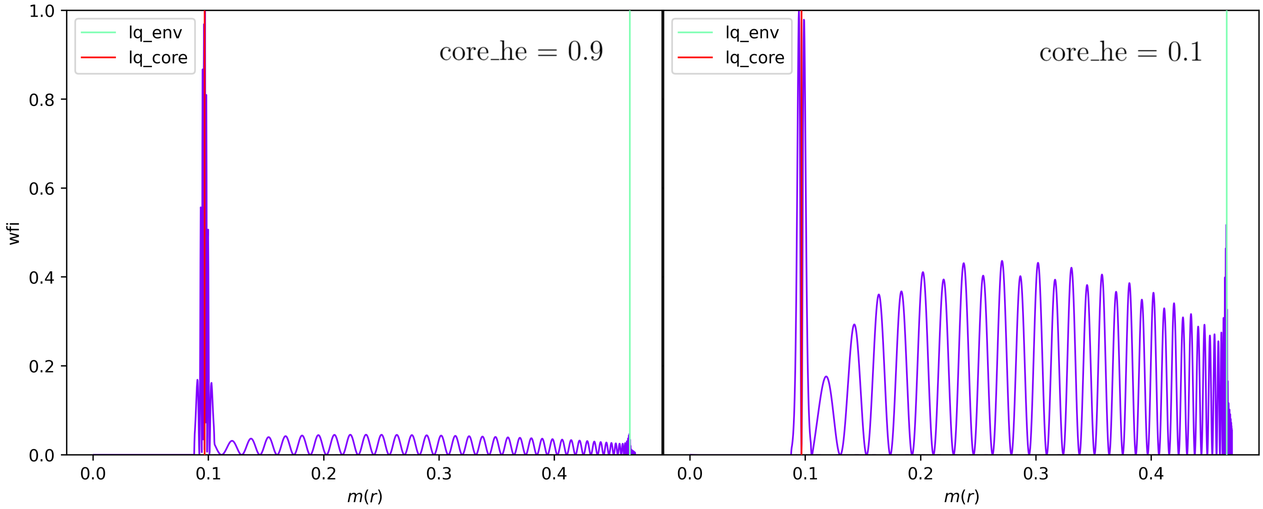


Fig. 3.26: Weight functions of trapped modes at lq_core = -0.10. The left figure shows a trapped mode for high core_he = 0.9 and the right figure for a low core_he = 0.1. Parameters are mass = $0.47 M_{\odot}$ and lq_env = -2.

those modes are trapped in this radiative arch.

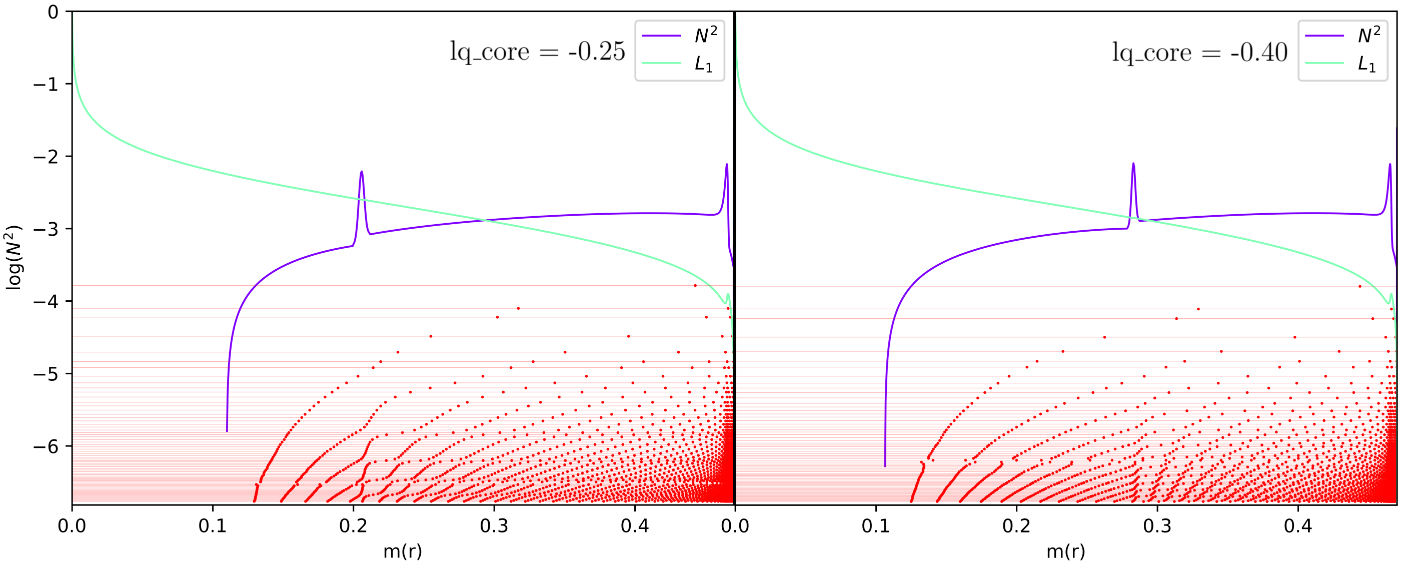


Fig. 3.27: Propagation diagrams for different lq_core, with on the left lq_core = -0.25 and on the right lq_core = -0.4. Parameters are mass = $0.47 M_{\odot}$, lq_env = -2, at core_he = 0.9.

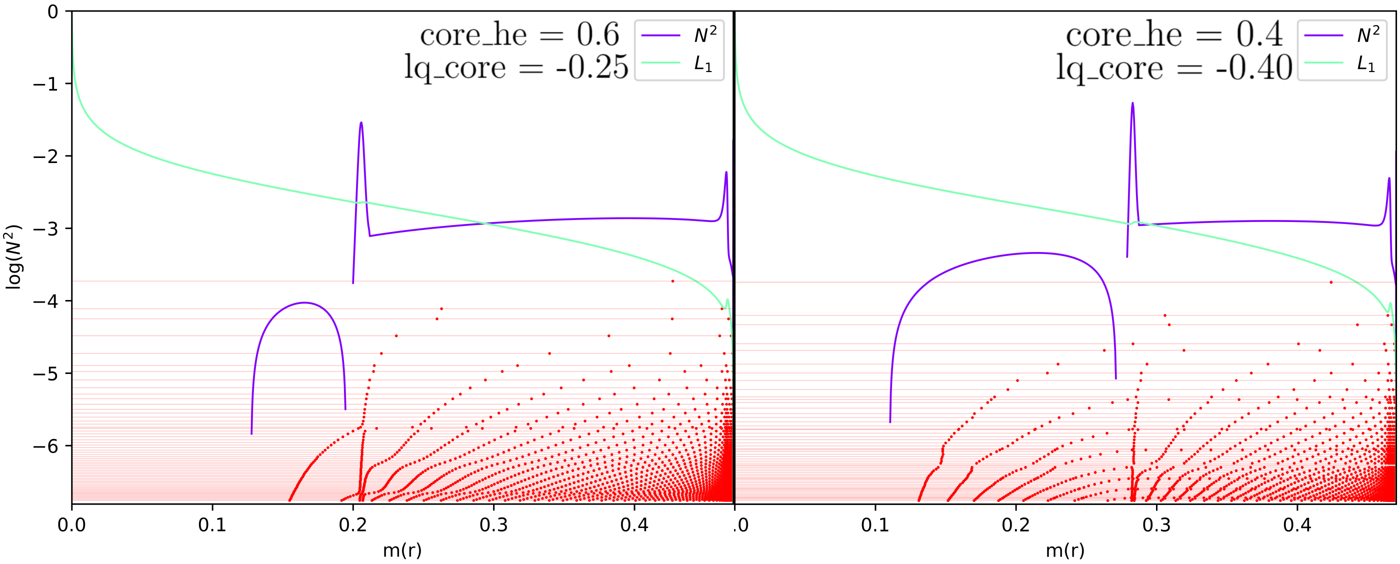


Fig. 3.28: A display of the "radiative arch" appearing at low core_he. We find two propagation diagrams, on the left a model using lq_core = -0.25 and core_he = 0.6, on the right a model using lq_core = -0.4 and core_he = 0.4. Parameters are mass = $0.47 M_{\odot}$ and lq_env = -2.

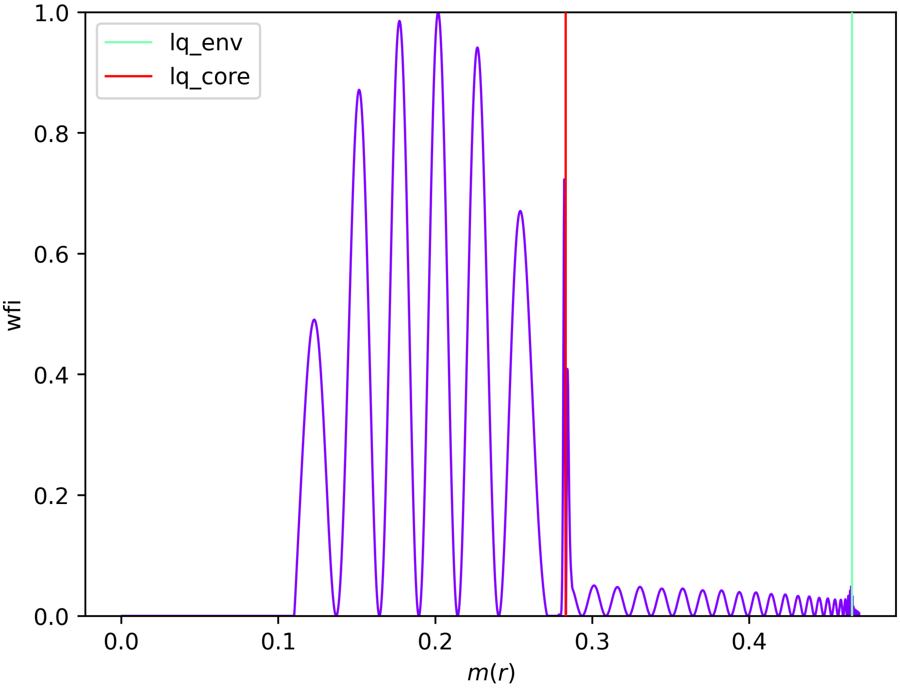


Fig. 3.29: Weight function of a mode trapped in the radiative arch region. Parameters are mass = $0.47 M_{\odot}$, lq_core = -0.4 and lq_env = -2, at core_he = 0.40.

The apparition of the radiative arch, as well as the convective zone right under lq_core with it, is a direct consequence of the prescription taken for the real temperature gradient ∇_T in 4G static models. Fig 3.30 shows the radiative temperature gradient ∇_{rad} with the green dashed line, the adiabatic temperature gradient ∇_{ad} with the light blue dashed line, and the real temperature gradient ∇_T in the full blue line. The real temperature gradient is determined by the constitutive physics of the models (opacities, equation of state). So far, we have talked about radiative and convective zones often using the Brunt-Väisälä frequency, however, in a code such as STELUM, one needs a criterion to know where a zone is convective, and where it is radiative, that is, defining the thermal structure of the star. In stellar structure litterature, we find two criteria. The Schwarzschild criterion gives that a convective zone is when $\nabla_{ad} \leq \nabla_{rad}$. The Ledoux criterion $\nabla_{rad} > \nabla_{ad} + \frac{\phi}{\delta}\nabla_\mu$ corresponds to a convective zone, where $\phi = \frac{\partial \ln \rho}{\partial \ln T}_{|P,\mu}$, $\delta = \frac{\partial \ln \rho}{\partial \ln T}_{|P,T}$ and $\nabla_\mu = \frac{d \ln \mu}{d \ln P}$. We find in fact ∇_μ in the expression of B (equation 1.33), and in zones where we have no chemical gradients, we find back the Schwarzschild criterion from the Ledoux criterion.

In static models, below the C-O/He transition at lq_core, the temperature gradient simply follow the Schwarzschild criterion : $\nabla_T = \nabla_{ad}$ where $\nabla_{rad} \geq \nabla_{ad}$ (convective zones), and $\nabla_T = \nabla_{rad}$ where $\nabla_{rad} < \nabla_{ad}$. As is well known for CHeB stars, issues can arise due to the dependency of the radiative gradient ∇_{rad} with the opacity of the star. Indeed, one can find that $\nabla_{rad} \propto \kappa$, with κ the opacity, and because the opacity of elements such as carbon and oxygen is higher than the opacity of helium, ∇_{rad} lowers as more helium is burned in the core. Due to the shape of the radiative gradient under lq_core, notably because of its local minimum, this leads to $\nabla_{rad} > \nabla_{ad}$ just under lq_core, which creates a convective zone. This is precisely this phenomenon which is shown in Fig 3.30. From the center of the star and towards the surface, we first find a convective zone with $\nabla_{rad} > \nabla_{ad}$, then a radiative zone which corresponds to the radiative arch where $\nabla_{rad} < \nabla_{ad}$, and then a small convective zone again with $\nabla_{rad} > \nabla_{ad}$. This second convective zone might not be physical, because it could lead (if powerful enough) to the injection of fresh helium in the core. As a consequence, the trapped modes we have found at high core size (section 3.3.2) might be non-physical. This is a limitation that should be kept in mind when modeling observed stars with 4G models. Other prescription for the actual temperature gradient could be implemented in 4G models, such as forcing a fully convective core below the C-O/He transition (by $\nabla_T = \nabla_{ad}$ below lq_core). The issue of defining convective and radiative zones, and more generally on the usage of the Schwarzschild and Ledoux criteria, is an old and well-known problem. We will discuss this in more details in chapter 5.

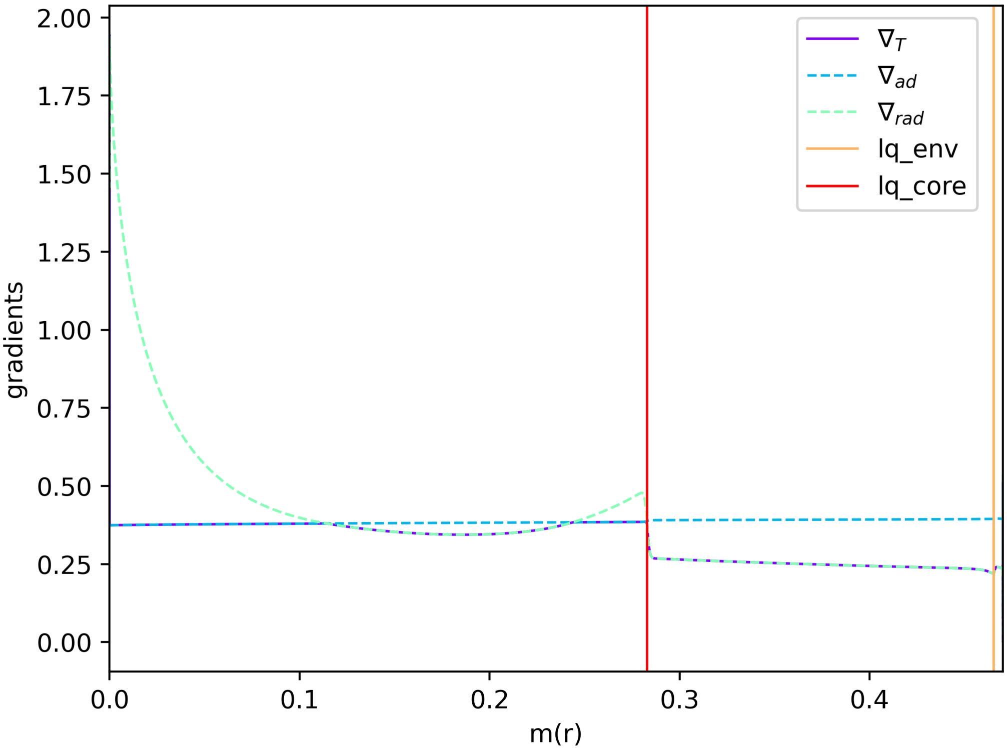


Fig. 3.30: Gradients associated to a Brunt-Väisälä frequency displaying a radiative arch. Parameters are mass = $0.47 M_\odot$, lq_core = -0.4 and lq_env = -2, at core_he = 0.05.

Chapter 4

Evolutionary models and semi-convection

Evolutionary models are in essence distinct from the static models we just studied on chapter 3. By definition, a given evolutionary model is dependent on previous ones, making the thermal (gradients) and chemical structures co-dependent. Additional convective phenomena such as overshooting and semi-convection are now taken in account (through diffusion coefficients, see section 2.1), and will be studied further. Studying the behavior of pulsation spectra in the context of evolutionary models allows unveiling general trends associated to them, and then to compare those trends to observationnal data, inferring on the physical relevance of our models. Furthermore, comparing evolutionary and static models pulsation spectra, which is an aim of this chapter, leads to the development of more accurate static models, in turn resulting in better and more flexible seismic probing of observed stars.

Because of the computing time of an evolutionary sequence[1], we directly used models that were computed by Stephane Charpinet prior to this master thesis. We have in total 6 different evolutionary tracks with similar total mass at about $0.47M_{\odot}$. Actually, the only difference in the evolutionary models we study is the position of the He/H+He transition lq_env, which takes the values lq_env = -2, -2.5, -3, -4, -4.5 and -5 in $\log(q)$ scale.

In this chapter, we highlight first the differences found in stellar structure between evolutionary and 4G models, in particular in the description of the core of the star, and the presence of overshooting and semi-convection zones in evolutionary models. In addition, we will pinpoint when the onset of semi-convection occurs, and its influence on the stellar structure of the star, as well as the thermal prescriptions associated to such a phenomenon. Then, we will study our pulsation spectra and their dependencies with respect to core helium abundance (core_he) and the position of the He/H+He transition (lq_env). Finally, we will compute the asymptotic period spacings obtained from the asymptotic theory of Tassoul (1980), and show they correspond well to the observed period spacings in evolutionary models.

4.1 Stellar structure

4.1.1 Overshooting and semi-convection

In static models, the thermal and chemical structure are not correlated. As we consider the core to be fully mixed before the C-O/He transition, we do not need to introduce an

[1] 1-3 months for an evolutionary sequence up to white dwarfs.

overshooting phenomenon, instead fixing the thermal and chemical structures directly. This is not the case in evolutionary models, for which the small convective zone before the C-O/He transition seen in Fig. 3.30 is then an issue, which is adressed by adding both the overshooting and semi-convection phenomena. Figure 4.1 shows two different chemical structures for an evolutionary model with the He/H+He transition at lq_env = -2, at core helium abundances core_he = 0.8 (top panels) and core_he = 0.2 (bottom panels), in $\log(q)$ (right panels, showing the envelope structure) and $m(r)$ scales (left panels, showing the core structure).

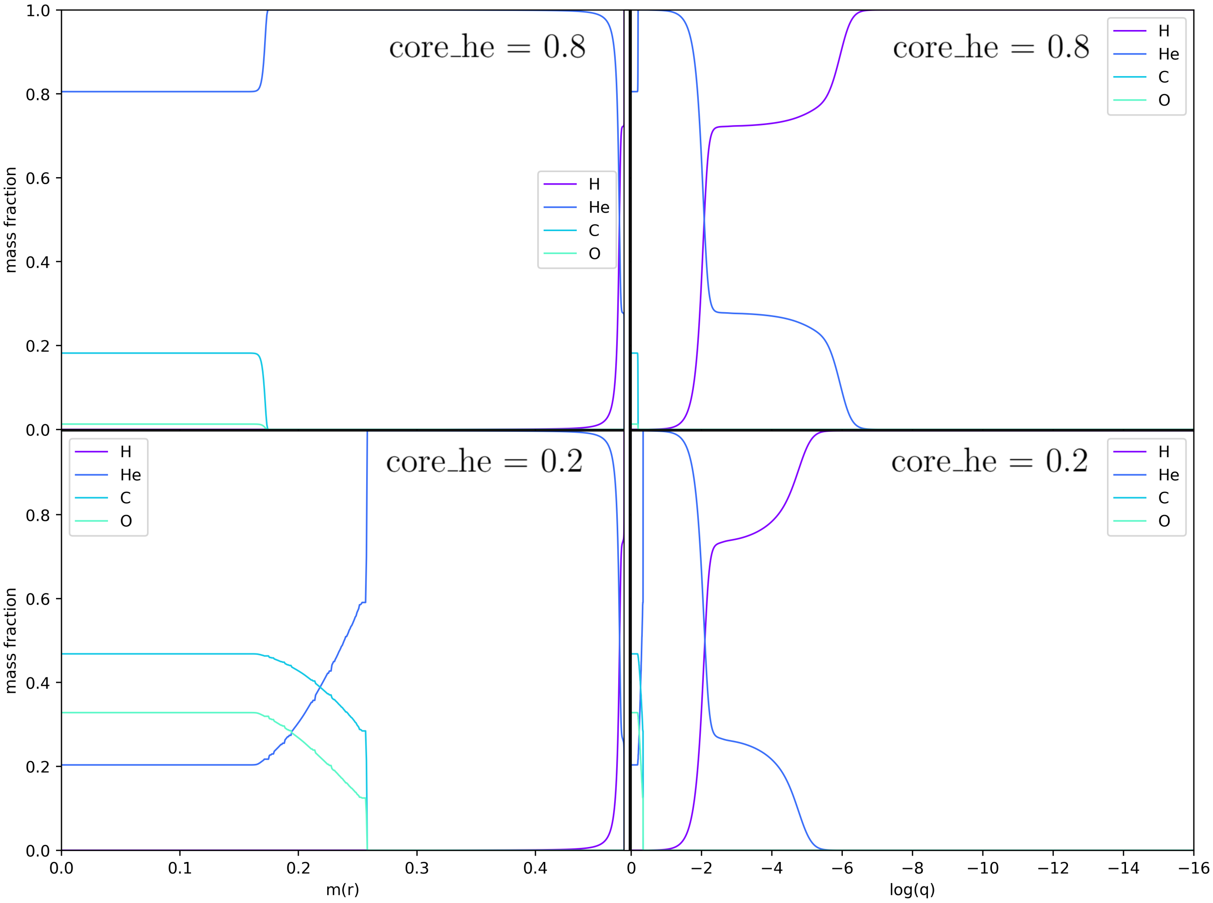


Fig. 4.1: Chemical abundances for a model without semi-convection on the top row (core_he = 0.8), and with semi-convection on the bottom row (core_he = 0.2). We give on the right column the $\log(q)$ scale, and on the left the $m(r)$ scale. Parameters are lq_env = -2.

On the top part of the figure at high core_he, that is at the start of the evolution of our star, we find a chemical structure which is similar to 4G static models. The convective core is now containing oxygen in addition to carbon and helium (in static models, only carbon and helium), and we do not find the helium flash pollution after the C-O/He transition from convective core to radiative mantle, the latter located around $\log(q) \approx -0.20$ for core_he = 0.8. In evolutionary models, there is no fixed position for the C-O/He transition, as this transition actually shifts during the evolution of an evolutionary model with core_he, which is shown in the next subsection. In $m(r)$ scale, the behavior of chemical abundances near the C-O/He transition appears to be completely different between high and low core_he. What we see here at core_he = 0.2 is a consequence of the management of the semi-convection problem in evolutionary models, which is treated through a zone of partial mixing of elements.

As indicated at the end of the last chapter, the radiative gradient lowers in value as the star core helium abundance diminishes. Initially, in static models, we had a convective core with a radiatve zone followed by the C-O/He transition. Due to the evolution and shape of the radiative gradient with core_he, the thermal structure eventually becomes a convective core, a radiative zone, and another non-physical convective zone before the C-O/He transition (see Fig. 3.30). In evolutionary models (STELUM and others such as CLES), there is an explicit treatment of this problem, called the "semi-convection treatement", for which various ad-hoc prescriptions exists. STELUM implements convection/diffusion interactions, through diffusion coefficients mentionned in section 2.1. The temperature gradients are not fixed or prescribed, and convective zones are defined according to the Schwarzschild criterion. This treatment allows finding a partial mixing zone which is akin to regular semi-convection treatments, albeit strongly dynamic and slightly varying between one model and the next. The semi-convection zone is visible in Fig. 4.1 (bottom left) as the zone of slowly varying chemical gradient from $m(r) \approx 0.18$ to $m(r) \approx 0.26 M_\odot$. Some "micro-structures" can be seen, as a consequence of this interactive treatment.

Figure 4.2 shows on the left the gradients in $m(r)$ scale associated with a low core_he model, and on the right, a zoom on those gradients in the semi-convection zone. We highlight three different zones. First, we find on the right and left figures the convective core, where $\nabla_{rad} > \nabla_{ad}$, and $\nabla_T = \nabla_{ad}$. Then, on the right figure, we find the overshooting zone from $m(r) \approx 0.13$ to $m(r) \approx 0.17 M_\odot$, where $\nabla_{rad} < \nabla_{ad}$, and with $\nabla_T = \nabla_{rad}$. Finally, above it from $m(r) \approx 0.17$ to $m(r) \approx 0.26 M_\odot$, we have the semi-convection zone itself where gradients adopt a much more complicated behavior, with sometimes radiative zones and sometimes convective zones, according to the Schwarzschild criterion. This "chaotic" behavior of gradients in the semi-convection zone is directly due to the implementation of convection/diffusion interactions to address the semi-convection problem.

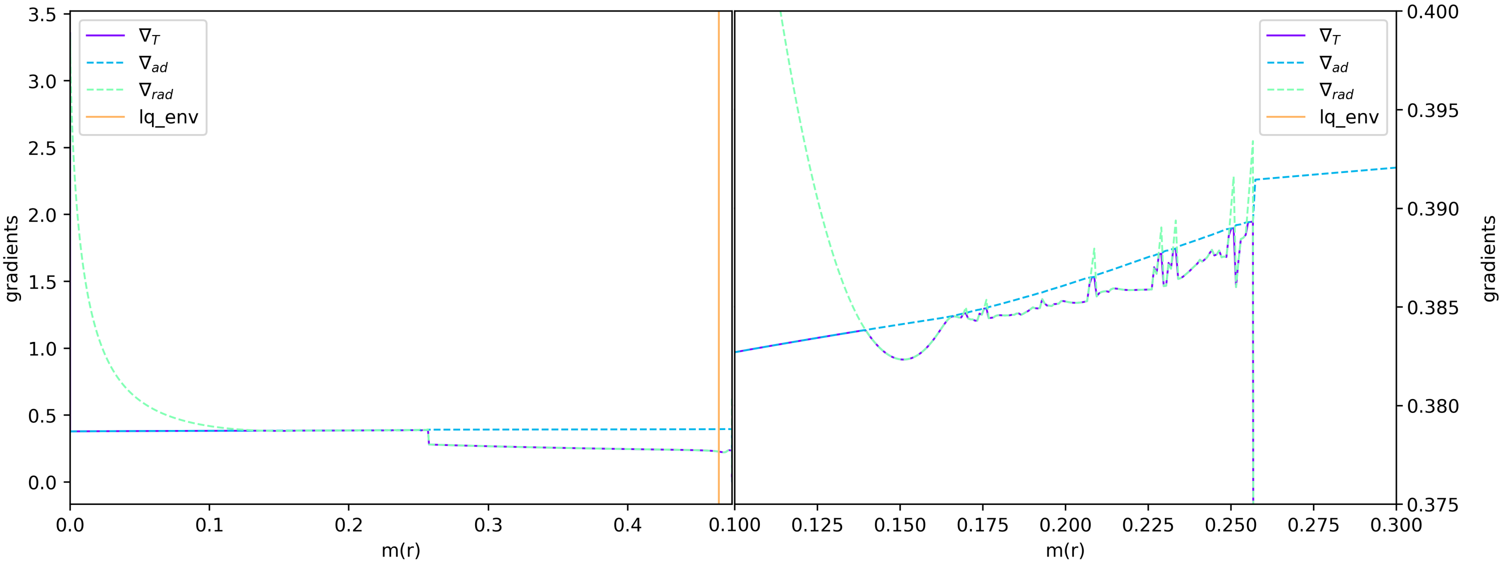


Fig. 4.2: Radiative, adiabatic and real gradients for a model with semi-convection. On the left are the gradients over the whole $m(r)$ scale, on the right zoomed on the semi-convection and overshooting zones. Parameters are lq_env = -2, core_he = 0.2.

4.1.2 Onset of semi-convection

The semi-convection phenomenon is not always present in evolutionary models, in particular, it appears only once the core helium abundance is low enough. In this subsection, we investigate the size of the convective core, overshooting zone, and semi-convection zone, as a function of the core helium abundance. As this information is not readily available in STELUM output files, I apply the Schwarzschild criterion such that the first layer for which $\nabla_{rad} < \nabla_{ad}$ is the upper limit of the convective core, while the lower boundary of it is at the center of the star. However, due to the behavior of the gradients, the Schwarzschild criterion is not suited to find the boundaries of the semi-convection zone. For the lower boundary of the semi-convection zone, I use the chemical gradient of carbon abundance. If the difference of carbon abunbance between two consecutive layers is high enough, I consider the lower boundary of the partial mixing zone found (see Fig. 4.1 bottom left, slowly decreasing carbon abundance), and so is the lower boundary of the semi-convection zone. I then take the top boundary of the semi-convection zone when the carbon abundance is almost null (since it is the case in the radiative mantle, after the C-O/He transition). Because the overshooting zone is bounded by the convective core and the semi-convection regions, its lower boundary is the upper boundary of the convective core, while its upper boundary is the lower boundary of the semi-convection zone. In Fig 4.3, a representation of the different boundaries found with the aforementioned critera is given, and we follow in good approximation the visible boundaries of each zone. Do note that the carbon criterion used to find the start of the semi-convection zone is ill-suited for high core_he values, before the onset of semi-convection. In these cases, the size of the semi-convection zone found just corresponds to the (small) size of the C-O/He transition (see later Fig. 4.4).

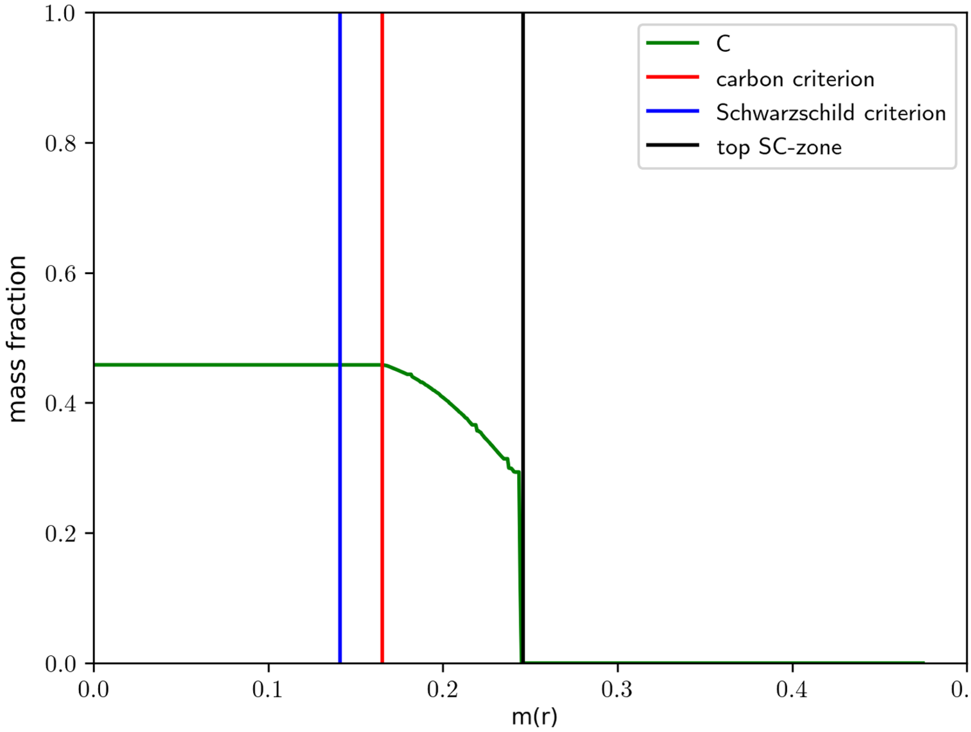


Fig. 4.3: Criteria used to find the size of the convective core, overshooting, and semi-convection zones. In green is the abundance of carbon, in red the carbon criterion defining the bottom of the semi-convective zone (“SC”), in blue the Schwarzschild criterion defining the bottom of the overshooting zone, and in black the top of the semi-convection zone, using a carbon abundance criterion. Parameters are lq_env = -2, core_he = 0.2.

In Fig. 4.4 the masses $\Delta m(r)$ contained within the convective core, the overshooting zone, and the semi-convection zone, are given as a function of the core helium abundance core_he. We show these core regions masses for the 6 evolutionary sequences, which vary only from the mass contained in their envelope. First, we note that the general trends for any of the curves are the same for all evolutionary sequences we consider, no matter the mass of the H-rich envelope (lq_env parameter). The convective core follows the same behavior as static models : it steadily grows with diminishing core helium abundance. However, this behavior only lasts until core_he ≈ 0.7, as the onset of semi-convection occurs and halts the convective core growth. The semi-convection zone then grows steadily with diminishing core helium abundance, while the convective core shows a small decrease in mass. The overshooting zone is of relatively constant size throughout the different evolutionary sequences, showing only a slowly decreasing mass until the onset of semi-convection, at which point the overshooting zone stays of near constant mass for the remaining values of core_he.

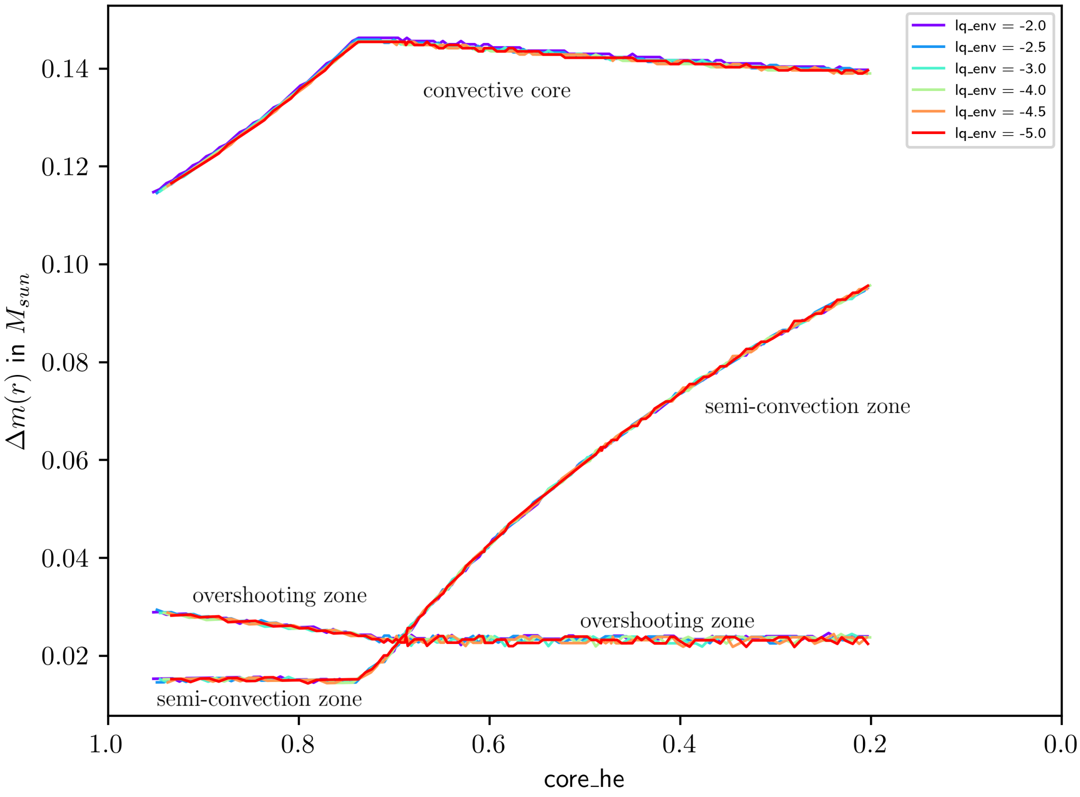


Fig. 4.4: Mass contained in the convective core, semi-convection zone, and overshooting zone for lq_env = -5, -4.5, -4.0, -3.0, -2.5 and -1.

4.2 Pulsation spectrum

In this section, we aim to study the behavior of pulsation spectra in evolutionary sequences in function of their main parameters, as we previously did for 4G models. In the case of evolutionary models however, since the different parameters cannot be changed directly, we only study the dependency of pulsation spectra against the helium core abundance and the mass of the envelope. This section highlights the discrepancies and similarities found between pulsation spectra of evolutionary and static models.

4.2.1 Mass of the envelope

We start by analysing the impact of changing the He/H+He transition lq_env in our models, that is, the mass of the envelope. Fig. 4.5 shows pulsation spectra found at core_he ≈ 0.8 and core_he ≈ 0.2. The top figures contain every lq_env tested, while the bottom figures only display the extremes lq_env = -2 and -5, for clarity. We do not give the kinetic energies associated for each lq_env for readability and because we will view those, as well as the thorough study of the modes themselves, in the next subsection. Trapped modes and normal modes for both core helium abundances are clearly present with the arch-like patterns at low core_he, and weak trapped modes as minima of the pulsation spectra at high core_he. It is readily apparent that for both core_he pulsation spectra, changing the mass of the envelope does not impact the behavior of high order modes, contrary to what we observed for static models (Fig. 3.21). Similarly to 4G models, low order modes are strongly affected by changes in lq_env, and their behavior stays chaotic from the nature of their node positionning, which are not confined in the radiative mantle due to not being as impacted by the node pinching phenomenon around the He/H+He transition, unlike high order modes (let us recall that observed g-modes in sdB stars correspond to high order modes).

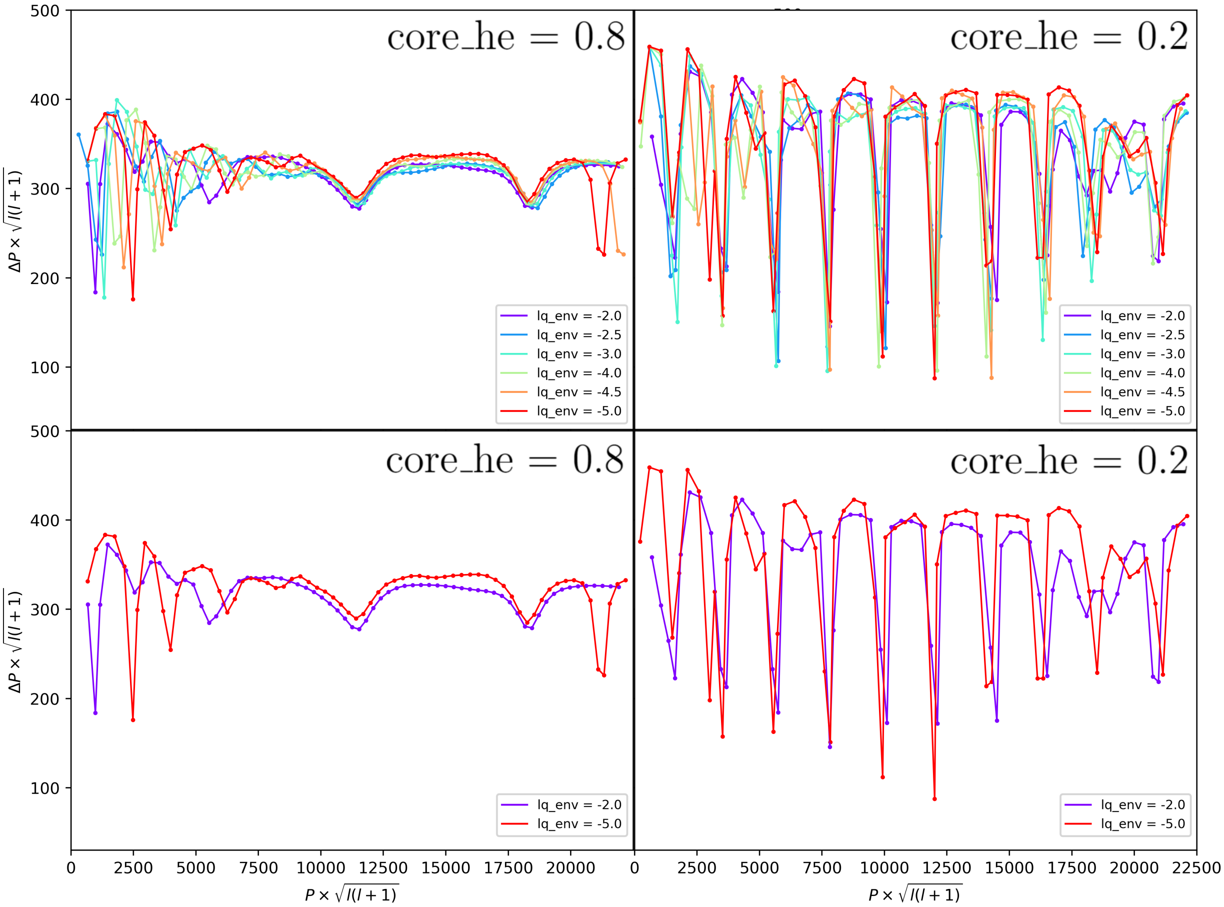


Fig. 4.5: Left to right : core_he = 0.2 ; core_he = 0.8. Top : pulsation spectra for every lq_env. Bottom : pulsation spectra for both extreme lq_env, for clarity.

4.2.2 Core helium abundance

We now investigate the different pulsation spectra during the evolution of the star, that is as a function of core helium abundance. We will use, as we did for 4G models, the He/H+He transition at lq_env = -2 as our reference point for the study of the influence of core_he. Figure. 4.6 shows pulsation spectra and their associated kinetic energy at core_he = 0.9, 0.8, 0.7, 0.6, 0.4, and 0.2, so that it is readily comparable with the study done for 4G models, in particular the pulsation spectra found in Fig. 3.16 which features the same core helium abundances[2].

We proceed in order of the evolution of the star, and start by studying high core_he pulsation spectra, before the onset of semi-convection. For both core_he = 0.9 and 0.8, we find a few weakly trapped modes at high order. The weight function of one weakly trapped mode as well

[2] Except for core_he = 0.1 as the evolutionary model for lq_env = -2 stops at core_he = 0.16.

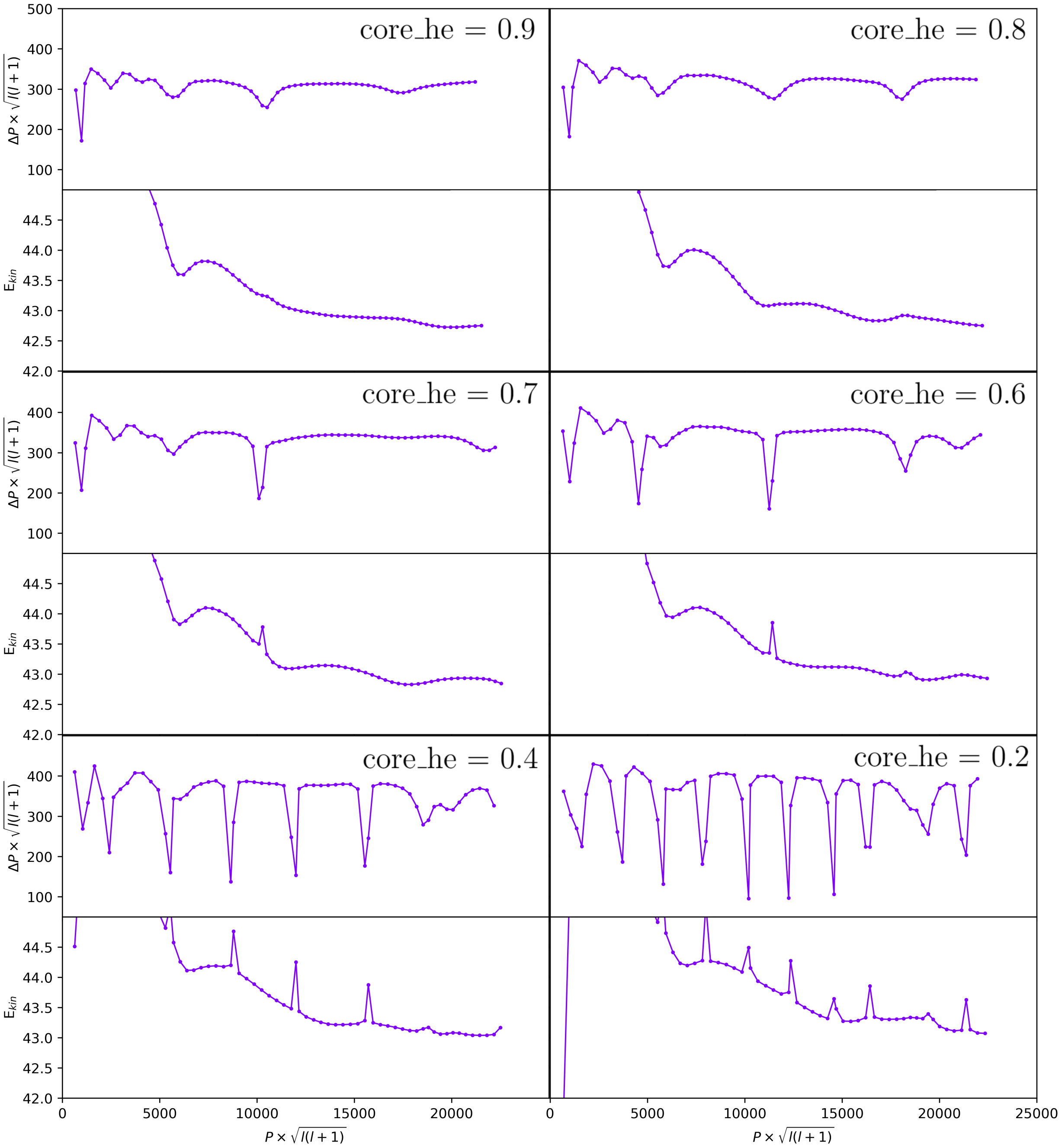


Fig. 4.6: Pulsation spectra for different core_he, indicated on the figure. Parameters : lq_env = -2.

as a normal mode at core_he = 0.9 is displayed in Fig. 4.7. The weight functions are maximal at the C-O/He transition, both for trapped and normal modes. The difference between both modes is the same as in static models : for a trapped mode, we find a stronger weight function in amplitude below the C-O/He transition compared to after it, and the opposite for a normal mode. As for static models, trapped modes in high core_he pulsation spectra of evolutionary models are trapped in a radiative region below the C-O/He transition, however in this case, the radiative region is the overshooting zone showed previously on Fig. 4.2. While both the radiative region of static models and the overshooting zone here are alike, in evolutionary models, the modes trapped in the overshooting region are weakly trapped, while in static models, the modes were more strongly trapped. This is a direct consequence of the size of both zones : the overshooting region is small ($\Delta m(r) \approx 0.03 M_{\odot}$, meaning $\Delta \log(q) = \log\left(1 - \frac{0.03}{0.47}\right) = 0.028$, Fig. 4.4) and contains at most two nodes for a given mode, while the radiative region of static models is larger ($\Delta\log(q) \approx 0.1$ (low core_he) to 0.3 (high core_he), infered from Fig. 3.18), and contains 6 to 8 nodes, thus having a greater influence on the modes. Note that we cannot find any static model which has a radiative zone as small as the evolutionary models overshooting zone, which would allow to compare both. Indeed, the radiative zone of static models becomes smaller as long as the core_he diminishes, however, a radiative arch (Fig. 3.28) appears before the radiative zone could potentially become comparable in size to the overshooting one. This is a first limitation of static models : because of their prescription on gradients and their fixed C-O/He transition, they cannot properly reproduce the overshooting phenomenon found in evolutionary models.

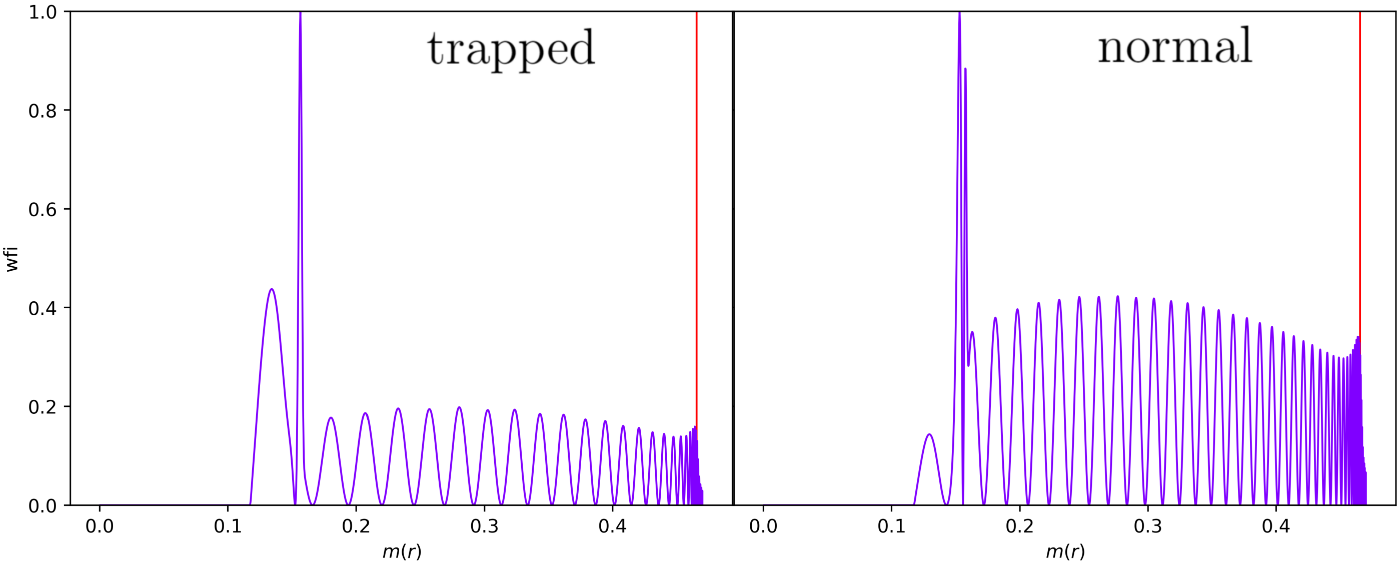


Fig. 4.7: Weight function for a trapped mode (left), and a normal mode (right), for an evolutive model at core_he = 0.9. Parameters : lq_env = -2.

Let us now analyse pulsation spectra after the onset of semi-convection, that is from core_he = 0.7. Strongly trapped modes are directly apparent, and each trapped modes we see here, would they be rather weakly trapped as on the right of the pulsation spectrum at core_he = 0.6, or more strongly trapped, as the modes for core_he = 0.2 and 0.4, are modes trapped in the semi-convection zone. In Fig. 4.8, we show the weight function of such a trapped mode, against a normal mode. We find that trapped modes have a strong weight function amplitude in the semi-convection zone, compared the radiative mantle and envelope, and the opposite for the normal mode. It is now readily apparent that as a general trend, every trapped modes, for static or evolutionary models, are characterized by weight functions of high amplitude in their trapping regions compared to outside of it, providing another reliable way to distinguish those modes from others (in addition of pulsation spectra and their kinetic energy).

An insight on why we find trapped modes in the semi-convective zone is given by the propagation diagram on Fig. 4.9. We directly see that the wave like pattern of the nodes we once saw only near the C-O/He transition in static models (Fig. 3.12 and Fig. 3.13), and that was associated with mode trapping, is now present at multiple positions in the semi-convection zone. This provides not only modes which are strongly trapped, but also a greater number of them in general. The effects on the Brunt-Väisälä frequency of the semi-convection treatment by diffusion coefficients is also clearly seen as the “noisy” part between $m(r) \approx 0.17$ and $0.25 M_{\odot}$. While the overshooting zone did have an effect for high core_he modes by providing a form of weak trapping (Fig. 4.7), it seems to be completely overpowered once semi-convection starts. In particular, we cannot find modes which are trapped in the overshooting zone while not being

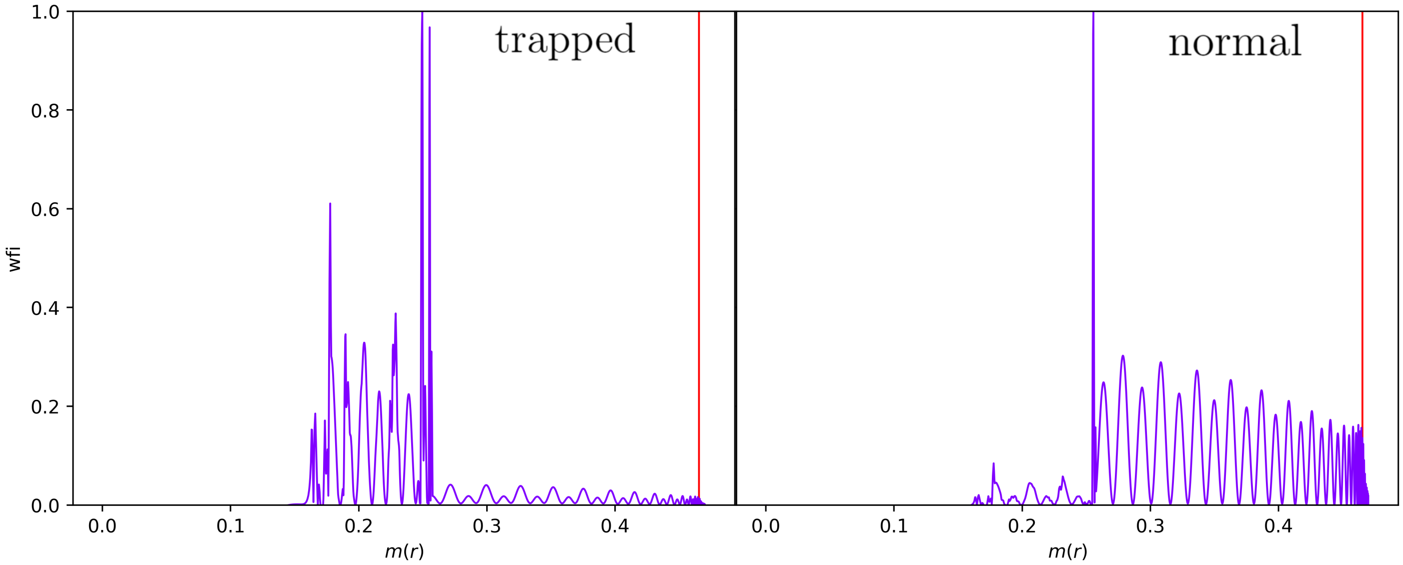


Fig. 4.8: Weight function for a trapped mode (left), and a normal mode (right), for an evolutive model at core_he = 0.2 and lq_env = -2.

trapped in the semi-convection zone. Note that pulsation spectra dominated by trapping in the semi-convection zone for evolutionary models are actually quite close to pulsation spectra at high lq_core with low core_he for 4G models (Fig. 3.25), that is, pulsation spectra associated with the radiative arch seen on Fig. 3.28 appearing at those low core_he. We will discuss this further in the next chapter, and for now leave it as a noticable feature.

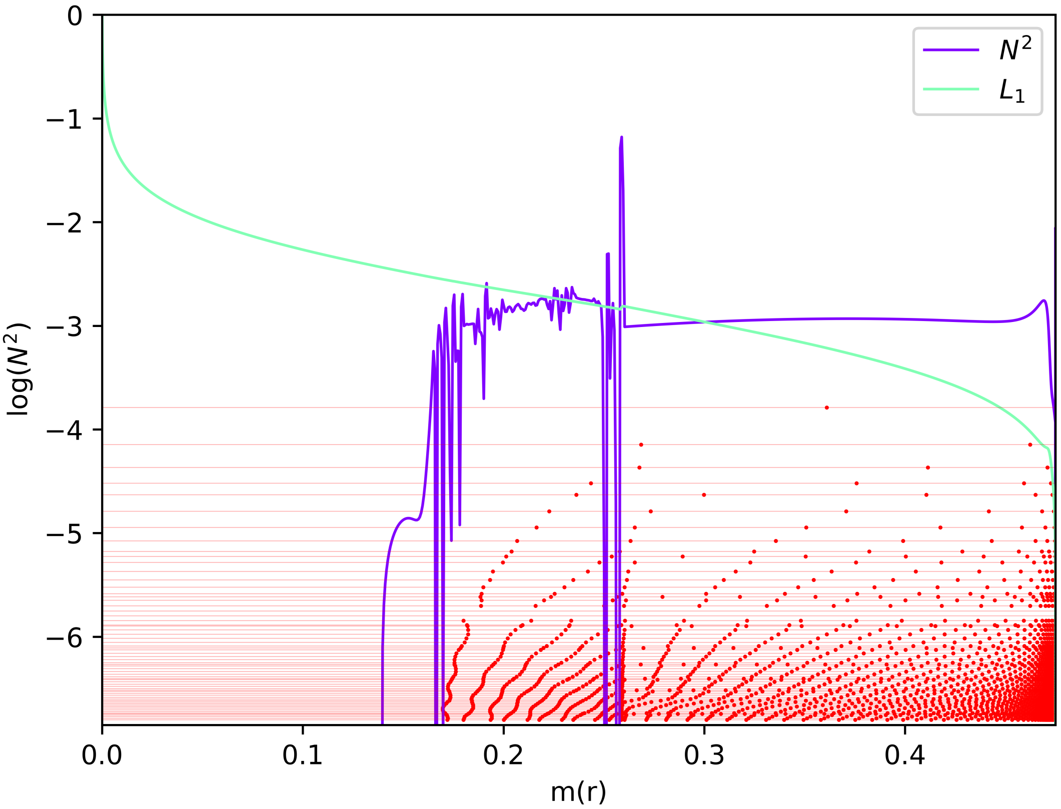


Fig. 4.9: Propagation diagram for an evolutionary model in $m(r)$ scale, for core_he = 0.2 and lq_env = -2.

4.3 Asymptotic period spacing

M. Tassoul developed an asymptotic theory for non-radial adiabatic pulsations of high order in the context of the Cowling approximation, that is, from the system (1.39) - (1.40), in Tassoul (1980). Following this work, in Brassard et al. (1992b), the theory is expended upon to take in account the discontinuities of the Brunt-Väisälä frequency due to chemical transitions for a white dwarf, and also valid in a stratified star such as an sdB. The asymptotic theory allows predicting the mean period spacing between two consecutive modes of radial indices k and $k+1$:

$$\overline{P_{k+1} - P_k} = \Pi_{0,l} \tag{4.1}$$

$$\text{with} : \Pi_{0,l} = \frac{2\pi^2}{\sqrt{l(l+1)}} \left(\int_{r_c}^{R} \frac{|N|}{r} dr \right)^{-1} \tag{4.2}$$

The integration in equation (4.2) occurs from r_c, which is the convective core upper boundary, to R, the surface of the star, and gives the theoretical mean period spacing between two consecutives modes, as we integrate over the whole propagation region of any g-modes. In the same way, we can find the theoretical period spacing between two trapped modes in the semi-convection region $\Pi_{T,l}$ by integrating over the boundaries of the semi-convection zone. We will then compare those two theoretical spacings to observed ones from the pulsation spectra of our evolutionary models. With this experiment, we not only validate the asymptotic theory against our models, but also further ensure that it is indeed the semi-convection zone that has a strong influence on our trapped modes. The integral is computed using a classic trapezoidal numerical integration scheme, and we limit ourselves to core_he = 0.2 for the minimal amount of core helium abundance. Below this core helium abundance, we find a phenomenon called "breathing pulses" which brings back helium into the core, as well as changing the pulsation spectra heavily. This phenomenon is not yet well understood, and since it is unsure if it is physical, we consider its study outside the scope of this master thesis.

In Fig. 4.10 we give the theoretical mean reduced period spacing for all lq_env, $\Pi_{0,l}\sqrt{l(l+1)}$, as a function of the core helium abundance core_he. The difference in behavior observed after the onset of semi-convection, around core_he = 0.7, is directly due to the variations in the Brunt-Väisälä frequency associated with semi-convection treatment (Fig. 4.9). This is found for all lq_env, in alignement with the non-dependency of pulsation spectra with lq_env shown in Fig. 4.5. The theoretical mean period spacing steadily increases with core_he diminishing, and shows a maximum around core_he = 0.4, decreasing afterward.

Figure 4.11 shows the theoretical reduced period spacing between two modes trapped in the semi-convection zone $\Pi_{T,l}\sqrt{l(l+1)}$, the behavior of which is also independent on lq_env. The onset of semi-convection is clear, with a sharp rise around core_he = 0.7, and actually the part before this onset should be discarded during the comparison between theoretical and observed spacings. Indeed, there can be no mode trapped in the semi-convection zone if it has not appeared yet, and the value we find below core_he = 0.7 is only a consequence of the criterion used to define the semi-convection zone in section 4.1.2, which gives a non-zero semi-convection zone size for high core_he, as seen on Fig 4.4. The downward trend of $\Pi_{T,l}\sqrt{l(l+1)}$ after the onset of semi-convection is explained directly from the size of the semi-convection zone increasing with diminishing core helium abundances, as seen Fig 4.4. This size growth widens the integration boundaries, increasing the value of the integral as a consequence, and thus decreasing the value of $\Pi_{T,l}\sqrt{l(l+1)}$ due to the negative exponent on the integral in equation (4.2).

Since the trends we find for the two theoretical period spacings are the same for any lq_env,

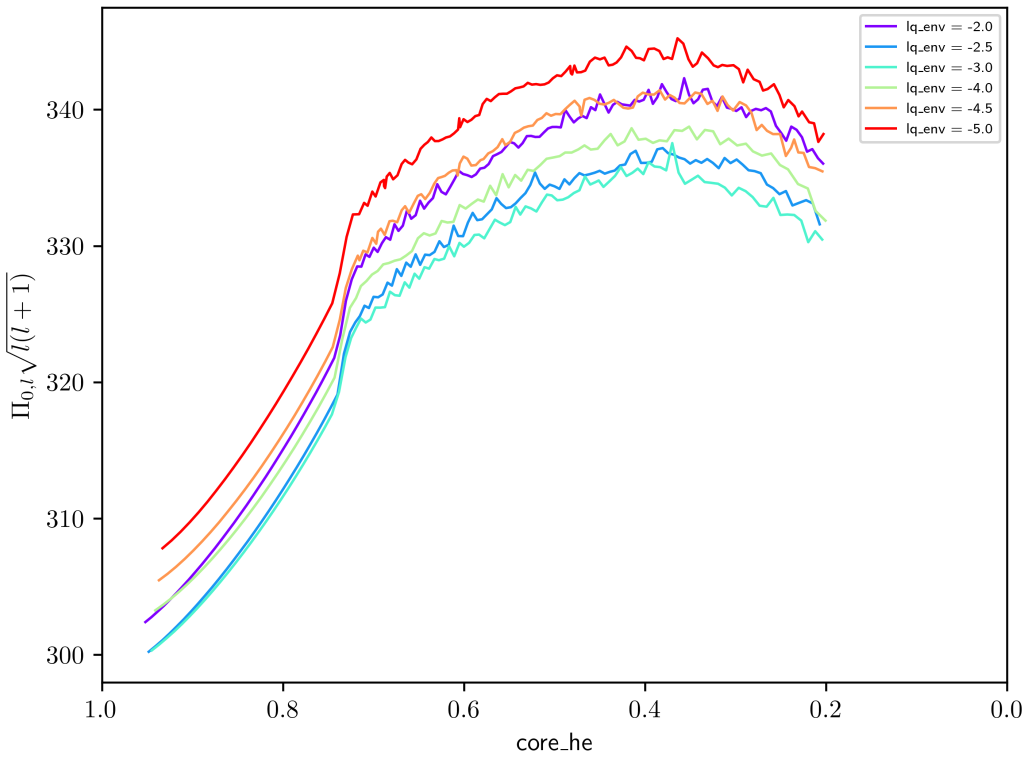


Fig. 4.10: Mean reduced period spacing between two consecutive modes of order k and $k+1$, for each lq_env in function of core_he.

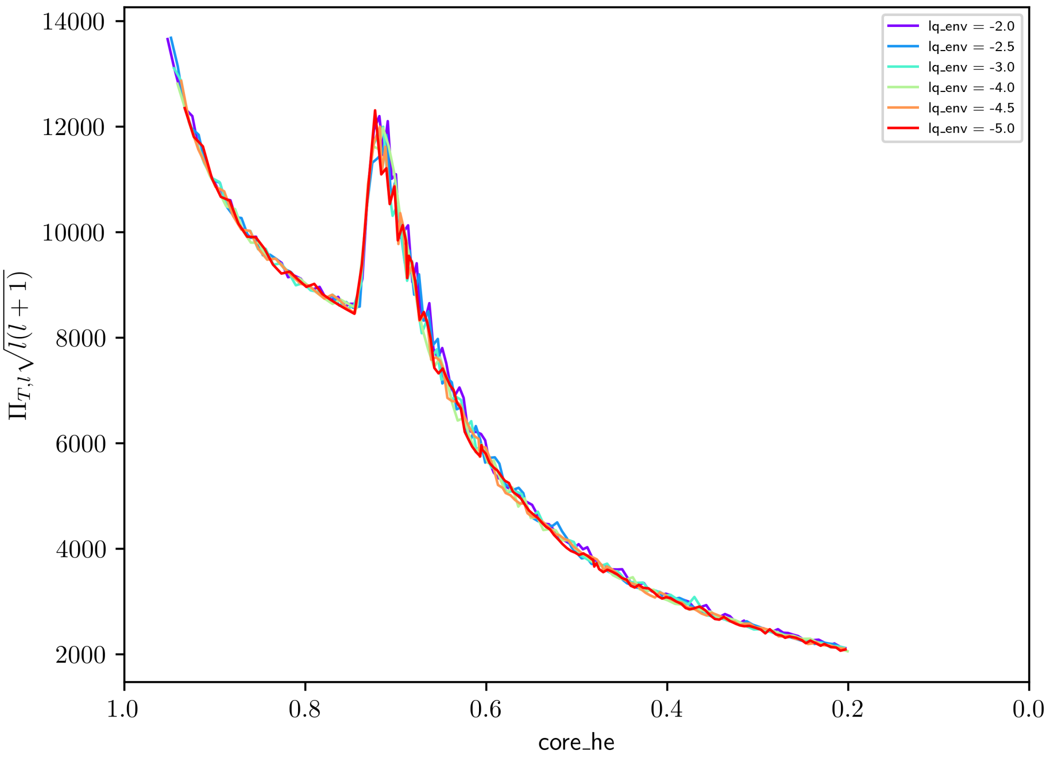


Fig. 4.11: Mean reduced period spacing between two trapped modes in the semi-convective zone, for each lq_env in function of core_he.

we now focus on the lq_env = -2 model. We limit our study of the mean period spacing in those models at low enough core_he so that we have not only the presence of the semi-convection zone, but also at least two trapped modes, since it is not possible to compute a mean period spacing for consecutively trapped modes otherwise. In Fig. 4.12 we give on the top panel the theoretical mean period spacing in blue, and in red the observed period spacing. We observe a very good agreement between model and theory, and the relative difference (bottom panel) between the two is of the order of 1%, and always less than 2%.

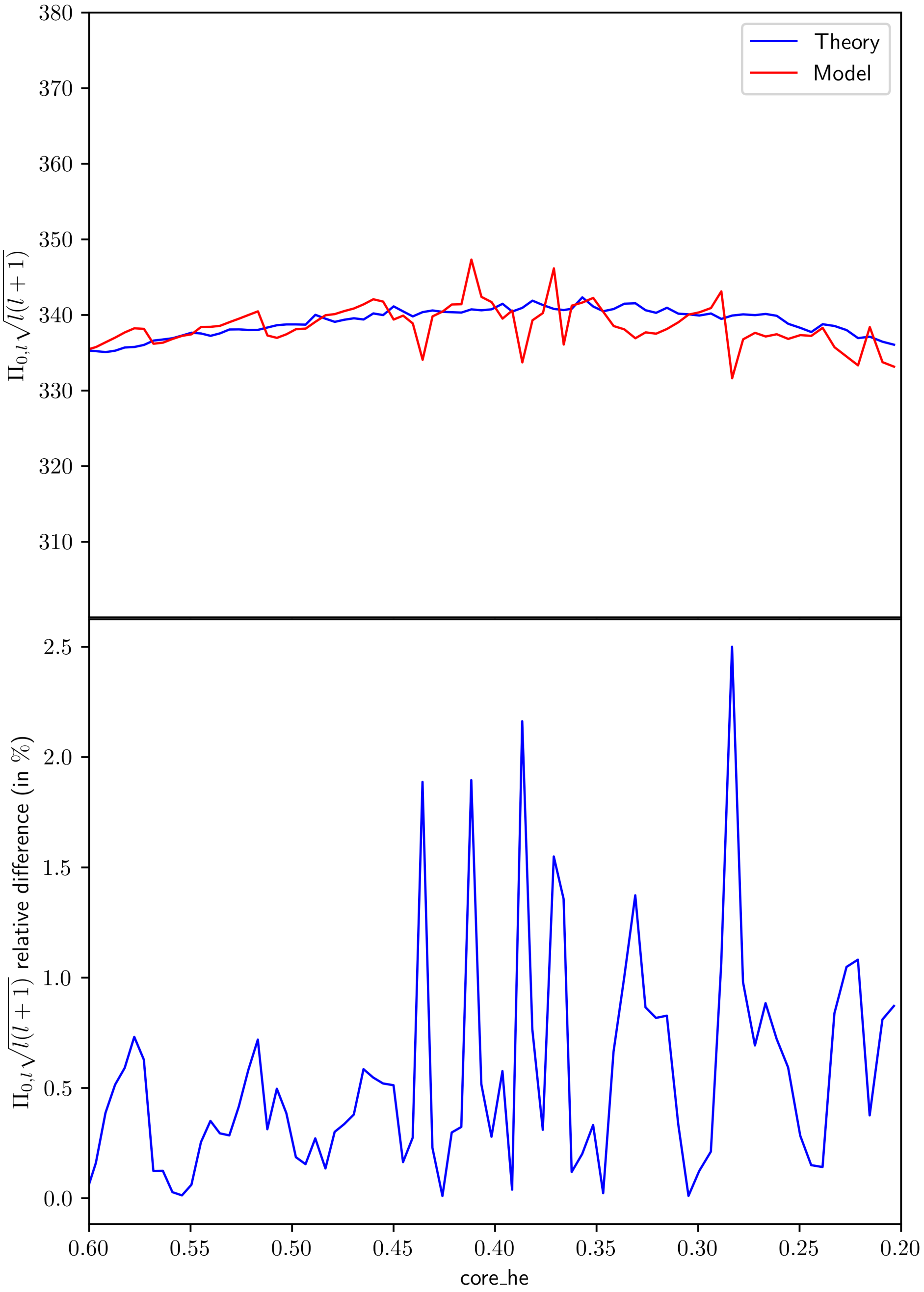


Fig. 4.12: On top : mean reduced period spacing between two consecutive modes of order k and $k+1$, in blue the theoretical value, in red the model computed value, for lq_env = -2. On the bottom : relative difference between the theoretical value and model computed one, in %.

In Fig. 4.13 we give on the top panel, the theoretical and observed mean period spacings between two trapped modes in a semi-convection zone, as a function of the core helium abundance. The observed value is in red, and we compute the theoretical value with integral boundaries over only the semi-convection zone in blue, and over both the overshooting and semi-convection zone in green. It is direct that taking in account only the semi-convection zone gives a better agreement between period spacings from model and theory. The impact of the semi-convection zone compared to the overshooting zone increases with diminishing core_he, as the semi-convection zone grows in size, and thus we see that the theoretical spacing counting both zones (green) converges to the one counting only semi-convection (blue) with decreasing core_he. This further highlights the low influence of the overshooting region on the trapped modes at low core helium abundances. On the bottom panel of Fig. 4.13, the relative difference between the theoretical spacing $\Pi_{T,l}\sqrt{l(l+1)}$ counting only the semi-convection zone, and the observed spacing, is between 5 and 10%, and never more than 16%. The relative difference between spacings decreases with core_he since we find an increased number of trapped modes at low core_he, as seen on Fig. 4.6, which increases the accuracy of the observed spacing as we take a mean of all observed spacings between consecutive trapped modes in our computations, for each core_he. Additionally, the number of layers in our models associated with the semi-convection zone is initially low, reflecting its small size in $\log(q)$ scale, and this number of layers increases as the semi-convection zone grows. Each of those layers represents a value point for the physical quantities of our models, and a higher number of points, which then increases with diminishing core_he, is associated with a more accurate numerical integration.

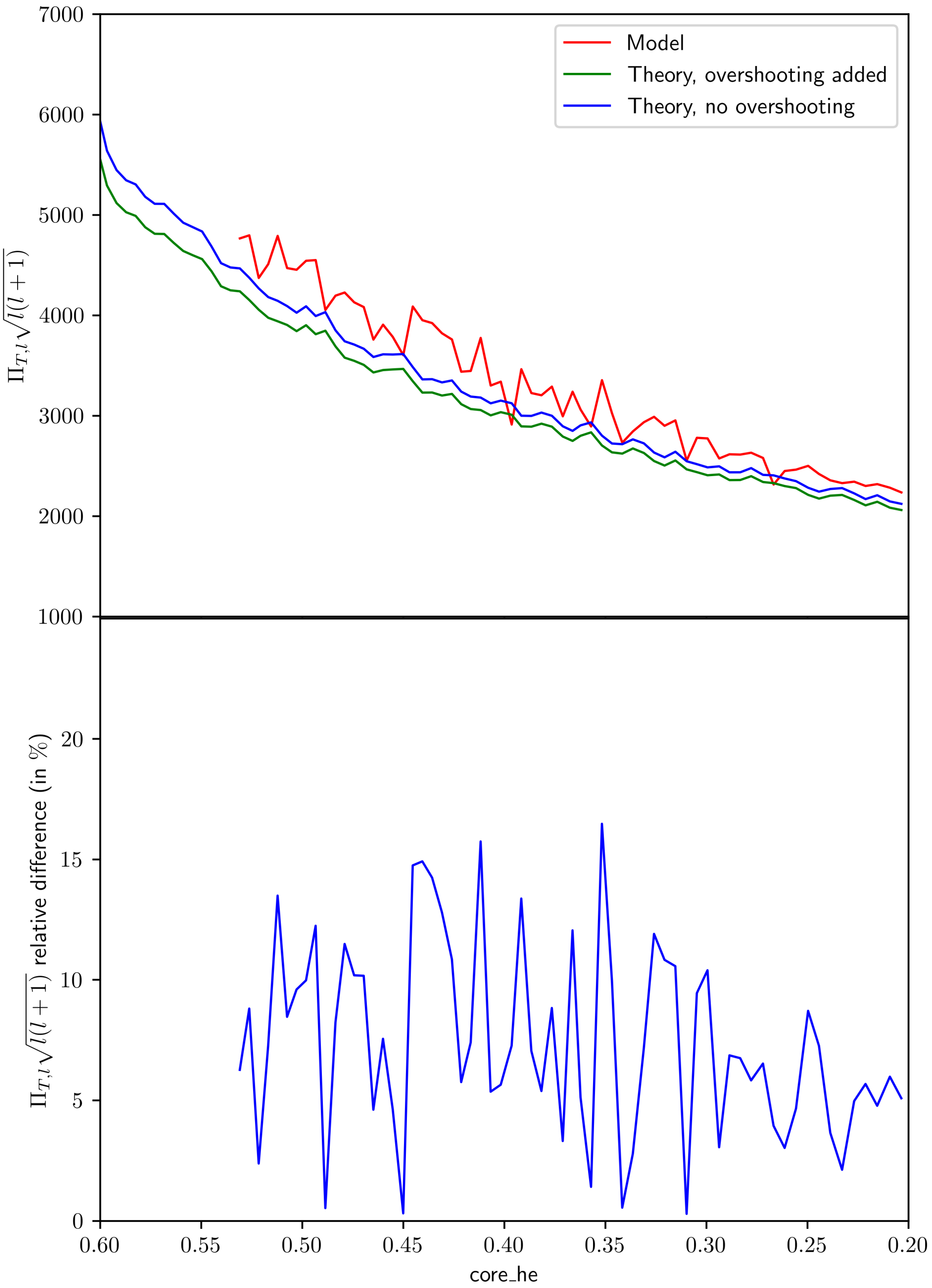


Fig. 4.13: On the top : mean reduced period spacing between two consecutive trapped modes in the semi-convection zone, in blue the theoretical value with no overshooting zone, in green the theoretical value including the overshooting zone, and in red the model computed value, for lq_env = -2. On the bottom : relative difference between the theoretical value and model computed one, in %.

Chapter 5

Conclusion and Perspectives

In this chapter, I summarize the results found in this master thesis, and I discuss their implications. First, we have observed notable differences for the evolution of the pulsation spectra as long as core helium is burnt in the core between static and evolutionary models. For static models, high core helium abundance core_he spectra show numerous trapped modes (Fig. 3.16, top panels) in a radiative region below the C-O/He transition (Fig. 3.17). The transition's position in $\log(q)$ scale lq_core determines the behavior of the pulsation spectra at low core_he. If lq_core is low (lower than ≈ -0.20), the region below the C-O/He transition is fully convective at low core_he, and we find weakly trapped modes (Fig. 3.16, for core_he $\leq$ 0.4, Fig. 3.25, bottom and middle panels). If lq_core is high enough (greater than ≈ -0.20), the convective core, which grows with decreasing core helium abundance (Fig. 3.18), is unable to reach the C-O/He transition at low core_he, and we find strongly trapped modes (Fig. 3.25, top right panel), which are trapped in a radiative arch below the C-O/He transition (Fig. 3.28, Fig.3.29). Concerning the evolutionary models, pulsation spectra evolution is linked to the onset of semi-convection (Fig. 4.4). Before the onset of semi-convection, at high core helium abundance (core_he $\gtrsim$ 0.7), we find weakly trapped modes, which are trapped in an overshooting region below the C-O/He transition (Fig. 4.6 top panels and Fig. 4.7). After the onset (core_he $\lesssim$ 0.7), we find strongly trapped modes in the semi-convection zone instead (Fig. 4.6 bottom panels and Fig. 4.8). Both static and evolutionary models highlight the importance of the presence of a radiative zone under the C-O/He transition, as it is the main actor of mode trapping for high core helium abundance in both cases.

The 4G static models do not appear to be able to correctly represent observed trapped modes found in KIC 10553698A (Østensen et al. 2014), KIC 10001893 (Uzundag et al. 2017) and EPIC 211779126 (Baran et al. 2017). On the other hand, not all sdB stars present such trapped modes, showing on the contrary a more or less constant period spacing for high order modes. It should be noted that the only way for 4G static models to reproduce a pulsation spectrum akin to those of the aforementioned stars, is through modes trapped in the radiative arch for high lq_core and low core_he, however, it is unsure if this radiative arch is physical or not as of now. Seimic probing using static models has been done for (but not only) KPD 1943+4058 (Van Grootel et al. 2010a), KPD 0629-0016 (Van Grootel et al. 2010b) and EC 21494-7018 (Charpinet et al. 2019), giving respectively lq_core = -0.37 ± 0.01, -0.27 ± 0.01 and -0.295 ± 0.013 at core_he = 0.739 ± 0.008, 0.59 ± 0.01 and $0.575^{+0.063}_{-0.027}$. It is noticable that in these three stars, the C-O/He transition position lq_core is found to be systematically higher than predicted from evolutionary models, while the he_core abundance is found to be relatively high.

I wish also to highlight the differences in thermal structures of static and evolutionary models, in particular in the core, with different prescriptions for the temperature gradients. In static models, the Schwarzschild criterion is used to define convective zones with $\nabla_{ad} \leq \nabla_{rad}$ and $\nabla_T = \nabla_{ad}$ fixed, and radiative zones with $\nabla_{ad} > \nabla_{rad}$, in those fixing $\nabla_T = \nabla_{rad}$ instead. This prescription leads to the so called "semi-convection problem" (Fig. 3.30), where a convective zone appears right under the C-O/He transition, which might be non-physical. In STELUM evolutionary models, this problem is addressed with a semi-convection zone obtained through diffusion/convection interactions, leading to a convective core followed by an overshooting zone, itself followed by the semi-convection zone (Fig. 4.2, right panel). Actually, we do not know the behavior of the temperature gradient in actual stars. In a very recent development, Blouin et al. (2024) found that, from 3D hydrodynamical simulations, a semi-convection zone might not be able to last for long below the C-O/He transition. Instead, simulations show that the zone below this transition becomes rapidly fully mixed thanks to efficient internal gravity waves transport, implying that CHeB stars do not harbour a semi-convection zone. If this result holds true, evolutionary models implementing semi-convection (whatever the prescription) do not provide a better representation of the reality than 4G static models.

In that case, static models with a fully mixed (that is already the case) and convective core (by imposing $\nabla_T = \nabla_{ad}$) might be a better description than evolutionary models with a semi-convection zone. Note that an ongoing development of static models (called the 5th generation) do allow for a partial mixing zone above the convective core and below the C-O/He transition. This is at the price of several more parameters. It remains to be seen if these 5G models could be used in practice to model observed g-modes in sdB stars. For evolutionary models, a fully mixed core up to the C-O/He transition could also be implemented, by controlling the overshooting parameter to allow the temperature gradient to be equal to the adiabatic gradient.

To conclude, our study shows that static and evolutionary models need to be used in conjunction to infer if a semi-convection zone might or not be located in the core of CHeB stars. It is only by comparing the result of both model types with observed pulsation spectra that this uncertainty will be lifted, leading to an accurate description of thermal and chemical structures of core-helium burning stars on the horizontal and extreme horizontal branches.

Bibliography

C. Aerts, J. Christensen-Dalsgaard, and D.W. Kurtz. *Asteroseismology.* 2010. Springer. doi: 10.1007/978-1-4020-5803-5.

A. S. Baran, M. D. Reed, R. H. Østensen, et al. *EPIC 211779126: a rare hybrid pulsating subdwarf B star richly pulsating in both pressure and gravity modes.* 2017. A&A, Vol. 597, A95. doi: 10.1051/0004-6361/201629651.

T. Battich, M. M. Miller Bertolami, A. H. Córsico, and L. G. Althaus. *ϵ-mechanism driven pulsations in hot subdwarf stars with mixed H-He atmospheres.* 2017. Open Astronomy, Vol. 26 (N°1), 152–159. doi: 10.1515/astro-2017-0431.

T. Battich, M. M. Miller Bertolami, A. H. Córsico, and L. G. Althaus. *Pulsational instabilities driven by the ϵ-mechanism in hot pre-horizontal branch stars.* 2018. A&A, Vol. 614, A136. doi: 10.1051/0004-6361/201731463.

S. Blouin, F. Herwig, H. Mao, et al. *3D hydrodynamics simulations of a 3 $M_\odot$ core helium burning star.* 2024. MNRAS, Vol. 527 (N°3), 4847–4862. doi: 10.1093/mnras/stad3518.

P. Brassard. *Propriétés adiabatiques des naines blanches puisantes de type ZZ Ceti.* 1991. PhD thesis, Université de Montréal.

P. Brassard and S. Charpinet. *PULSE: a finite element code for solving adiabatic nonradial pulsation equations.* 2008. Astrophysics and Space Science, Vol. 316 (N°1), 107–112. doi: 10.1007/S10509-007-9733-Z.

P. Brassard and G. Fontaine. *Computations of static white dwarf models: A must for asteroseismological studies.* 1994. IAU Colloq. 147: The Equation of State in Astrophysics, page 560.

P. Brassard, G. Fontaine, F. Wesemael, et al. *Adiabatic Properties of Pulsating DA White Dwarfs. I. The Treatment of the Brunt-Väisälä Frequency and the Region of Period Formation.* 1991a. ApJ, Vol. 367, 601. doi: 10.1086/169655.

P. Brassard, G. Fontaine, F. Wesemael, and C. J. Hansen. *Adiabatic Properties of Pulsating DA White Dwarfs. II. Mode Trapping in Compositionally Stratified Models.* 1992b. ApJS, Vol. 80, 369. doi: 10.1086/191668.

P. Brassard, C. Pelletier, G. Fontaine, and F. Wesemael. *Adiabatic Properties of Pulsating DA White Dwarfs. III. A Finite-Element Code for Solving Nonradial Pulsation Equations.* 1992c. ApJS, Vol. 80, 725. doi: 10.1086/191679.

A. Bédard, P. Brassard, P. Bergeron, and S. Blouin. *On the Spectral Evolution of Hot White Dwarf Stars. II. Time-dependent Simulations of Element Transport in Evolving White Dwarfs with STELUM.* 2022. ApJ, Vol. 927 (N°1), 128. doi: 10.3847/1538-4357/ac4497.

V. Castellani, Giannone, P., and Renzini, A. *Overshooting of convective cores in helium-burning horizontal-branch stars.* 1971a. Astrophysics and Space Science, Vol. 10 (N°2), 340–349. doi: 10.1007/BF00704092.

S. Chandrasekhar. *The Dynamical Instability of Gaseous Masses Approaching the Schwarzschild Limit in General Relativity.* 1964. ApJ, Vol. 140, 417. doi: 10.1086/147938.

S. Charpinet. *Le potentiel de l'astéroséismologie pour les étoiles sous-naines de type B.* 1999. PhD thesis, Université de Montréal.

S. Charpinet, G. Fontaine, P. Brassard, and B. Dorman. *The Potential of Asteroseismology for Hot, Subdwarf B Stars: A New Class of Pulsating Stars?* 1996. ApJ, Vol. 471 (N°2), L103. doi: 10.1086/310335.

S. Charpinet, G. Fontaine, P. Brassard, et al. *A Driving Mechanism for the Newly Discovered Class of Pulsating Subdwarf B Stars.* 1997. ApJ, Vol. 483 (N°2), L123. doi: 10.1086/310741.

S. Charpinet, G. Fontaine, P. Brassard, and B. Dorman. *Adiabatic Survey of Subdwarf B Star Oscillations. II. Effects of Model Parameters on Pulsation Modes.* 2002a. ApJS, Vol. 139 (N°2), 487. doi: 10.1086/338822.

S. Charpinet, G. Fontaine, P. Brassard, et al. *Structural parameters of the hot pulsating B subdwarf Feige 48 from asteroseismology.* 2005a. A&A, Vol. 443 (N°1), 251–269. doi: 10.1051/0004-6361:20053294.

S. Charpinet, G. Fontaine, P. Brassard, et al. *Structural parameters of the hot pulsating B subdwarf PG 1219+534 from asteroseismology.* 2005b. A&A, Vol. 437 (N°2), 575–597. doi: 10.1051/0004-6361:20052709.

S. Charpinet, V. Van Grootel, D. Reese, et al. *Testing the forward modeling approach in asteroseismology - II. Structure and internal dynamics of the hot B subdwarf component in the close eclipsing binary system PG 1336-018.* 2008. A&A, Vol. 489 (N°1), 377–394. doi: 10.1051/0004-6361:200809907.

S. Charpinet, P. Brassard, G. Fontaine, et al. *TESS first look at evolved compact pulsators: Discovery and asteroseismic probing of the g-mode hot B subdwarf pulsator EC 21494-7018.* 2019. A&A, Vol. 632, A90. doi: 10.1051/0004-6361/201935395.

T. Constantino, S. W. Campbell, J. Christensen-Dalsgaard, et al. *The treatment of mixing in core helium burning models - I. Implications for asteroseismology.* 2015. MNRAS, Vol. 452 (N°1), 123–145. doi: 10.1093/mnras/stv1264.

J. P. Cox. *Theory of stellar pulsations.* 1980. Princeton University press.

B. Dorman, R. T. Rood, and R. W. O'Connell. *Ultraviolet Radiation from Evolved Stellar Populations. I. Models.* 1993. ApJ, Vol. 419, 596. doi: 10.1086/173511.

G. Fontaine, P. Brassard, and P. Bergeron. *The Potential of White Dwarf Cosmochronology.* 2001. PASP, Vol. 113 (N°782), 409. doi: 10.1086/319535.

G. Fontaine, P. Brassard, S. Charpinet, et al. *A Driving Mechanism for the Newly Discovered Long-Period Pulsating Subdwarf B Stars.* 2003. ApJ, Vol. 597 (N°1), 518–534. doi: 10.1086/378270.

N. Giammichele, S. Charpinet, and P. Brassard. *Seismic Cartography of White-Dwarf Interiors From the Toulouse-Montréal Optimal-Design Approach.* 2022. Frontiers in Astronomy and Space Sciences, Vol. 9, 879045. doi: 10.3389/fspas.2022.879045.

E. M. Green, G. Fontaine, M. D. Reed, et al. *Discovery of A New Class of Pulsating Stars: Gravity-Mode Pulsators among Subdwarf B Stars.* 2003. ApJ, Vol. 583 (N°1), L31. doi: 10.1086/367929.

U. Heber. *The atmosphere of subluminous B stars. II. Analysis of 10 helium poor subdwarfs and the birthrate of sdB stars.* 1986. A&A, Vol. 155, 33–45.

M. L. Humason and F. Zwicky. *A Search for Faint Blue Stars.* 1947. ApJ, Vol. 105, 85. doi: 10.1086/144884.

D. Kilkenny, C. Koen, D. O'Donoghue, and R.S Stobie. *A new class of rapidly pulsating star — I. EC 14026-2647, the class prototype.* 1997. MNRAS, Vol. 285 (N°3), 640–644. doi: 10.1093/mnras/285.3.640.

P.-O. Quirion, G. Fontaine, and P. Brassard. *Wind competing against settling: a coherent model of the GW Virginis instability domain.* aug 2012. ApJ, Vol. 755 (N°2), 128. doi: 10.1088/0004-637X/755/2/128.

M. Tassoul. *Asymptotic approximations for stellar nonradial pulsations.* 1980. ApJS, Vol. 43, 469–490. doi: 10.1086/190678.

W. Unno, Y. Osaki, H. Ando, et al. *Nonradial oscillations of stars.* 1989. University of Tokyo Press.

M. Uzundag, A. S. Baran, R. H. Østensen, et al. *KIC 10001893: a pulsating sdB star with multiple trapped modes.* 2017. MNRAS, Vol. 472 (N°1), 700–707. doi: 10.1093/mnras/stx2011.

V. Van Grootel. *Étude des étoiles de la branche horizontale extrême par l'astérosismologie.* 2008. PhD thesis, Université Paul Sabatier.

V. Van Grootel, S. Charpinet, G. Fontaine, et al. *Early Asteroseismic Results from Kepler: Structural and Core Parameters of the Hot B Subdwarf KPD 1943+4058 as Inferred from g-mode Oscillations.* 2010a. ApJ, Vol. 718 (N°2), L97–L101. doi: 10.1088/2041-8205/718/2/L97.

V. Van Grootel, S. Charpinet, G. Fontaine, et al. *Structural and core parameters of the hot B subdwarf KPD 0629-0016 from CoRoT g-mode asteroseismology.* 2010b. A&A, Vol. 524, A63. doi: 10.1051/0004-6361/201015437.

V. Van Grootel, S. Charpinet, P. Brassard, et al. *Third generation stellar models for asteroseismology of hot B subdwarf stars: A test of accuracy with the pulsating eclipsing binary PG 1336-018.* 2013. A&A, Vol. 553, A97. doi: 10.1051/0004-6361/201220896.

R. H. Østensen, J. H. Telting, M. D. Reed, et al. *Asteroseismology revealing trapped modes in KIC 10553698A.* 2014. A&A, Vol. 569, A15. doi: 10.1051/0004-6361/201423611.

Appendix A

Codes

In this appendix, I present the codes I developed to produce the different graphs of this master thesis. To interpret evolutionary and static models files, I created a local package in python called Asteroseismology for STELUM and PULSE (ASP). This package is composed of three main types of programs. The *interpreters* are charged of extracting data from various files and computing values from them, the *plotters* produce graphs from those data, and the *main* programs make the link between plotters and interpreters program files. There exists three interpreters, one for STELUM stellar structure data, another for PULSE files which notably gathers the period and kinetic energy of the modes, and a last one which gathers the weight functions and eigenfunctions of our modes from so called “EIG files” (files with the .eig extension). Each of those interpreters is associated with a plotter respectively, which are given later on. Additionally, we find a few special plotter programs which compute the cancelling factor for Fig. 1.4, the propagation diagrams, and the spherical harmonics on Fig. 1.3. Finally, we find a standalone file in charge of computing and plotting (for evolutive models only) the size of the convective core, overshooting zone, and semi-convection zone, as well as the theoretical and experimental period spacings for all modes (mean period spacing), and for modes trapped in the semi-convective region.

This package was designed by necessity of automating effectively the obtention of several graphs to do systematic studies of parameters in our models. Special attention was given to having adaptable functions for both static and evolutionary models in order to gain clarity as well as computing efficiency. In total, I computed 63 static models with various parameters in this master thesis, associated with close to 90000 graphs. I was also given 6 different evolutionary models, for which I computed around 16000 graphs.

A.1 Interpreters

The stelum interpreter first extracts the data from a typical STELUM stellar structure file. In this master thesis, we work with the so-called “5m format”, which means that we are given 5 lines of different data for each layers of our stellar structure (generally, we have 4800 layers for a given core helium abundance). We give a more detailed description of what each line is composed of below, with the functions LINE_ZERO() to LINE_FOUR(). Those where implemented in the code for readability.

```
def line_zero():

    """
    n      - layer number
    r      - radius
    mr     - mass inside a radius r
    rho    - density
    p      - pressure
    t      - temperature
    chir   - partial ln P / partial ln rho |T
    chit   - partial ln P / partial ln T |rho
    del    - gradient
    delad  - adiabatic gradient
    y      - Helium mass fraction
    b      - Ledoux term
    lq     - log(1-mr/Mtot)
    mode   - 1:Y=0,2:Y is rising,3:Y=1,4:Y is diminishing,5:Y=0
    fl     - 0:radiative, 1:convective
    fx     - 0:liquid, 1:solid"""
```

```
def line_one():

    """
    lum    - luminosity
    eta    - d ln L / d ln r
    etar   - d ln L_rad / d ln r
    kappa  - rosseland opacity
    kappar - partial ln k / partial ln rho |T
    kappat - partial ln k / partial ln T |rho
    delra  - radiative gradient
    zeta   - d ln delad / d ln r
    eps    - nuclear epsilon & epsilon neutrino
    epsr   - partial ln eps / partial ln rho |T
    epst   - partial ln eps / partial ln T |rho"""
```

```
def line_two():

    """
    tau    - log(optical depth)
    w      - delrad/delrad_diffusion
    wg     - d ln W / d ln g |Teff,tau
    wtef   - d ln W / d ln Teff |g,tau
    wtau   - d ln W / d ln tau |g,Teff
    delp   - partial ln del / partial ln P
    delt   - partial ln del / ln T
    deltau - partial ln del / ln tau
    delg   - partial ln del / ln g
    delgb  - partial ln del / partial ln g_ (?)
    deltf  - partial ln del / ln Teff
    deldel - partial ln del / partial ln delrad
    deladp - partial ln delad / ln P |T
    deladt - partial ln delad / partial ln T |P
    dadmd  - delad-del"""

```

```
def line_three():

"""
```

```
u       - internal energy
dudt    - partial ln U / partial ln T |rho,X_i
dudrho  - partial ln U / partial ln rho |T,X_i
cp      - specific heat at constant pressure
cv      - specific heat at constant density
eta_e   - degeneracy parameter
z       - average ionization state
gamma   - ionic correlation parameter"""
```

```
def line_four():

"""
h       - hydrogen mass fraction
he      - helium mass fraction
c       - carbon mass fraction
o       - oxygen mass fraction"""

```

We extract those lines and associated data with the EXTRACTING() function below. Since the static and evolutionary models come with the same stellar structure format, we can use this function for both without issues. Inputing a given stellar structure file, we gather afterward the data in a three dimentionnal array, which contains the five lines of data for each layers of the file.

```
def extracting(file):

    """Extracts data from format 5m stelum files, format is as follows :

    Extracted data is in an array read as [layer,line,data_index]
    Below is "line" - "data_index":"data"
    {xx} is log10 value of xx

    0 - 0:<n>,1:r,2:mr,3:rho,4:p,5:t,6:chir,7:chit,8:grad,9:delad,10:Y,11:b,12:lq,13:<mode>,14:<fl>,15:<fx>
    1 - 0:<lum>,1:eta,2:etar,3:kappa,4:kappar,5:kappat,6:delrad,7:zeta,8:eps,9:epsr,10:epst
    2 - 0:{tau},1:w,2:wtau,3:delr,4:delp,5:deltau,6:dellum,7:dp,8:dt,9:dadmd
    3 - 0:<ui>,1:duidp,2:cp,3:cv,4:eta_e,5:zmoy,6:gamma
    4 - 0:<H>,1:He,2:C,3:O,4:pg

    Additional information can be found with help(line_number)
    For example help(line_zero) gives the zero line quantity details."""

    if file[-3:] == ".gz":
        f = opening_gz(file)
    else:
        f = opening_non_gz(file)

    main_count = 0 #counts the line we are at in the file
    secondary_count = 0
    #stelum data comes in periods of 5 lines for a given layer
    #we thus use a secondary count which will inform us
    #on which of the five lines we are.

    for line in f:

        splitted = np.array(line.split()).astype(str)

        if not line:
            # just checks if we are at the end of the file
            break

        if main_count == 1:

            # init stelum data set
            n_couches = int(splitted[0])
            n_lines = 5 #this is the 5 lines mentionned earlier (called 5m format)
            n_rows_max = 16 #max number of datas in a given row in 5m format (some may
            ↪  have less)

            stelum_data = np.zeros(n_couches*n_lines*n_rows_max).reshape(n_couches,n_line⌋
            ↪  s,n_rows_max)
            #init of the array containing all stelum files data

        elif main_count > 1:

            l_splitted = splitted.shape[0]
            n_layer = (main_count-2)//5 #finding the layer we are currently in

            stelum_data[n_layer,secondary_count,:l_splitted] =
            ↪  np.array(line.split()).astype(float)
            #gathering the data !

            secondary_count += 1
            if secondary_count == 5:
                secondary_count = 0#changing the line we're at and resetting to zero if we
                ↪  went above 5

        main_count += 1

return(stelum_data)
```

We then can use those data to compute the Brunt-Väisälä and Lamb frequencies with the following function COMPUTE_BV_LAMB().

```
def compute_bv_lamb(stelum_data,lamb):

    """Computes Brunt-Vaisala and Lamb frequencies

    Takes [stelum_data] and [lamb] = np.array([1,2,3,4..])

    Returns :   b_v for log(N^2), lamb_l[l] for log10(L_l**2)
                lq, mr
    """

    grav_constant = grav_const = 6.6743e-8 #cgs units

    r = stelum_data[:,0,1] #radius
    mr = stelum_data[:,0,2] #mass contained in said radius
    rho = stelum_data[:,0,3] #density
    p = stelum_data[:,0,4] #pressure
    chir,chit = stelum_data[:,0,6],stelum_data[:,0,7] #see "line zero()"
    grad,grad_ad,b = stelum_data[:,0,8],stelum_data[:,0,9],stelum_data[:,0,11]
    #real gradient, adiabatic gradient, ledoux compositionnal gradient
    lq = stelum_data[:,0,12] #log(q)


    #below, we compute all terms needed for the Brunt Vaisala frequency
    #and Lamb frequency
    g = grav_const*mr/(r**2)
    cs_2 = chir*p/rho
    b_v = np.log10((g**2)*(rho/p)*(chit/chir)*(grad_ad - grad + b))

    #iteration over degree "l" for the Lamb frequency
    lamb_l = np.zeros(lamb.shape[0]*r.shape[0]).reshape(lamb.shape[0],r.shape[0])
    for i,l in enumerate(lamb):
        lamb_l[i] = np.log10(l*(l+1)*cs_2/(r**2))

    mr_to_msun = mr/(1.9884e+33)
    #converting the mass radius in msuns, for plots

return(b_v,lamb_l,lq,mr_to_msun)
```

The PULSE file interpreter works in the same way, except that evolutionary and static pulse files do not have the same amounts of data. Thus, in the following function EXTRACTING(), we must know which one we are working with, which is done with the extension of the files. Static pulse files have a ”.gz” extension, while evolutionary pulse files do not, which we use to identify which is which. Note that the EXTRACTING() function in this case does not gather information, it simply distinguishes betwen different models and calls the functions which will then extract data.

```
def extracting(file: str,l_max = 4,k_max = 70,mode_spacing=False):

    """
    Input :
        -- file_path:str
        -- l_max:int default = 4
        -- k_max:int default = 70
        -- mode_spacing:bool default = False ; if True, returns period spacing aswell

    Returns :
        -- pulse_data:array[l,k,data]
        if Static : data: 0:ModeID,1:L,2:K,3:M,4:Pad,5:Pnad,6:Sigma,7:log
        ↪  Ekin,8:Ckl,9:Kp,10:Kg,11:Kp+Kg
        if Evol   : data: 0:L,1:K,2:Per,3:Var,4:Log(Ene),5;C,6:Kp,7:Kg,8:Kp+Kg,9:Ko
        -- period_spacing:array[l,spacing for mode index+2-index+1] ; period spacing of
        ↪  the modes
    """

    if file[-3:] == ".gz":
        #.gz files are always static files in our case
        #we use this to our advantage !
        opened_file = opening_gz(file)
        static = True
        pulse_data,adaptive_counter = data_extraction(opened_file,l_max,k_max,static)

    else:
        #if the file isn't .gz, it is thus an evolutive model
        opened_file = opening_non_gz(file)
        static = False
        pulse_data,adaptive_counter = data_extraction(opened_file,l_max,k_max,static)

    for i in range(adaptive_counter.shape[0]):
        #this makes sure we're not having zeros for modes that are just not there
        #at the same time, it ensures we do not count the anormal value in period spacing
        pulse_data[i,adaptive_counter[i]-1:,:] = np.nan

    if mode_spacing == True:
        #computing takes time, we use argument such mode_spacing = True
        #to choose what we want, and when we want it, avoiding useless computing
        period_spacing = period_spacing_computing(pulse_data,static,adaptive_counter)
        return(pulse_data,period_spacing)

return(pulse_data)
```

The EXTRACTING() function calls DATA_EXTRACTION(), which is the one extracting data from the PULSE files. In particular, it identifies g-modes, their degree and radial order, as well as their period and kinetic energy.

```
def data_extraction(f,l_max,k_max,static):

    """Extracts the data from .pulse files"""

    main_count = 0 #count the line we are at in the file
    start = 1e3 #to define the start of datas (avoiding non-data lines)

    adaptive_counter = np.zeros(l_max,dtype=int)
    #each modes of degree l do not have the same amount of radial orders computed (sometimes)
    #this makes sure we track the correct number of radial orders for each

    for line in f:

        main_count +=1

        #here, part for the "é" accents in evolutive files...
        try:
            splitted = np.array(line.split()).astype(str)
        except UnicodeDecodeError:
            #spots the "é" accent and skips the data line
            splitted = []

        if "K" in splitted and "L" in splitted:
            #for pulse files, data gathering starts 2 lines after the "K" and "L" letter line
            start = main_count + 2

        if start == main_count:
            #we initialize the array for data gathering here
            #this cannot be done before as we use an adaptive number of data
            #this is to account for both static and evolutives pulse files, which are different
            data_number = splitted.shape[0]
            pulse_data = np.zeros(l_max*k_max*data_number).reshape(l_max,k_max,data_number)

        if start <= main_count:
            splitted = splitted.astype(float)
            if static == True:
                #gathering l and k values for static models
                l,k = int(splitted[1]),int(splitted[2])
            else:
                #gathering l and k values for evolutive models
                l,k = int(splitted[0]),int(splitted[1])
            l = l-1 #for l to follow python indexing...
            if k < 0 and adaptive_counter[l] < k_max:
                #and here we iterate on the adaptive counter
                #we'll use this counter later on
                pulse_data[l,adaptive_counter[l],:] = np.array(line.split()).astype(float)
                adaptive_counter[l] += 1

    return(pulse_data,adaptive_counter)
```

If we wish to compute the period spacing between modes as well, then we use MODE_SPACING = TRUE in the EXTRACTING() function, which calls PERIOD_SPACING_COMPUTING :

```
def period_spacing_computing(pulse_data,static,adaptive_counter):

    """
    Computes the period spacing for each modes"""

    if static == True:
        index = 4
    else:
        index = 2
    #above, we simply account for periods not being at the same location
    #in evolutive and static models

    periods = pulse_data[:,:,index]
    rolled_periods = np.roll(periods,axis=1,shift=-1)
    period_spacing = rolled_periods - periods
    #2-1,3-2....,70-69,1-70
    #we compute the period spacing above

    return(period_spacing)
```

Finally, for the interpreters, we find the EIG file interpreter. EIG files are large data files containing in particular the weight function and the y_1 and y_2 eigenfunctions values. Those files contain, for each, a mode of single degree l (they can have multiple degrees, but then become extremely heavy), but multiple radial orders. In our case, generally, the EIG files then contain $l = 1$ modes with up to 70 radial orders. We use a function called EXTRACTING(), to extract those radial orders as well as the associated weight function, and the eigenfunctions y_1 and y_2.

```
def extracting(file,k_max=70,data_number=12,average_n = 16000):

    """Extracts data from format eig files

    Returns : eig_data[k,line(n),data]

    data : 0:xi,1:y1,2:y2,3:y3,4:y4,5:ri/rt,6:xir/rt,7:xih/rt,8:dT/T,9:wfi/wfmax,10:lqi,11:rki/rkmax"""

    main_count = 0 #counts which line we are at in the file
    mode_count = 1e3 #needed to skip non-data lines
    k_finder = 1e3 #needed to skip non-data lines
    real_k = 0 #needed against ill-counted modes
    k = 0

    eig_data = np.zeros(k_max*data_number*average_n).reshape(k_max,average_n,data_number)

    with open(file,'r') as f:
        for line in f:

            main_count += 1

            splitted = np.array(line.split()).astype(str)[1:] #array of data
            #[-1:] to account for the first collum which put some random "****" after n = 9999.

            if "K" in splitted: #spots the starts of mode of radial order k
                mode_count = main_count + 5 #accounts for non data lines
                k_finder = main_count + 1 #accounts for non data lines
            if k_finder == main_count: #indicates we can start gathering data
                k = int(splitted[0]) #retains the radial order of the mode
            if k < 0 and mode_count <= main_count and real_k < k_max:
                #k < 0 to account for g-modes only
                #real_k < k_max to avoid overflow from badly numeroted modes
                try:
                    eig_data[real_k,main_count - mode_count,:] = splitted #actually gathers data
                except ValueError:
                    real_k += 1 #counts the "real k", which can be different from the k given in the file
                    n = main_count - mode_count #find the real "n" (finds the last line of data for a given mode k)

    eig_data = eig_data[:,:n,:] #slicing at the real "n"
    x = int(np.sum(eig_data[:,-1,5])) #finds the real number of modes < k_max
    eig_data = eig_data[:x,:,:] #slice at the real number of modes

    return(eig_data)
```

In order to draw propagation diagrams, we also need the zeros of y_1 (and y_2 if we wish to see them). To find them, we use the ZERO_FINDER() function given below :

```
def zero_finder(eig_data):

    """
    Returns : zeros[y_{index+1},k,zeros]
    Modes with anormal zeros are a line of full np.nan."""

    k_max = eig_data.shape[0] #find the k_max to define the size of next arrays

    zeros = np.zeros(k_max**2).reshape(k_max,k_max)
    zeros = np.array([zeros,zeros])
    #this array contains the zeros of the y1 and y2 eigenfunctions for each k

    y = np.array([eig_data[:,:,1],eig_data[:,:,2]]) #array containing y1 and y2 values

    for i in range(y.shape[0]):
        for k in range(k_max):
            before_zeros = np.where(np.diff(np.sign(y[i,k,:])))[0]
            #we find the index at which we switch sign
            #we gather both the index before and after that sign switch
            after_zeros = before_zeros + 1
            try:
                zeros[i,k,:k+1] = ((eig_data[k,before_zeros,10] +
                ↪   eig_data[k,after_zeros,10])/2)
                #find the half index (for more precision)
            except ValueError as error:
                continue
    zeros[zeros == 0] = np.nan #to not have issues in plots
    return(zeros)
```

A.2 Plotters

The plotters are used to "plot" (understand display) the graphs we have seen in this master thesis. We first find three main plotters associated to the three interpreters. The STELUM plotter provides a way to plot the Brunt-Väisälä frequency with PLOT_BV_LAMB(), the chemical abundances with PLOT_CHEMICAL_FRACTIONS(), the different gradients (∇_T, ∇_{ad}, ∇_{rad}) with PLOT_GRADIENTS() and the ledoux term B with PLOT_LEDOUX_TERM(). We give those function in that same order below.

```
def plot_BV_Lamb(stelum_data,image_path,file_name,l=np.array([1,4]),mr_axes =
↪   False,set_xlim=["min","max"],set_ylim=["min","max"]):

    """Plotting of the Brunt-Vaisala and Lamb frequency"""

    b_v,lamb_l,lq,mr = stelum.compute_bv_lamb(stelum_data,l)
    #Gathers the Brunt-Vaisala frequency, the Lamb frequency, the log(q) and m(r) scales

    colors = cm.rainbow(np.linspace(0, 1, 1 + l.shape[0])) #color scheme

    fig,ax = plt.subplots()

    if mr_axes == True:
        #here, if we want axes in m(r) scale
        lq = mr
        ax.set_xlabel(r"m(r)")
        ax.set_ylim(-15,0)
        ax.set_xlim(xmin = np.nanmax(lq),xmax = np.nanmin(lq))
    else:
        #and here if we want axes in log(q) scale
        ax.set_xlabel(r"log(q)")
        ax.set_ylim(-5,0)
        ax.set_xlim(np.nanmin(lq),np.nanmax(lq))

    if set_ylim != ["min","max"]:
        ax.set_ylim(ymin=set_ylim[0],ymax=set_ylim[1])

    if mr_axes == False and set_xlim != ["min","max"]:
        #m(r) scale axes are from 0 to positive value
        ax.set_xlim(xmin=set_xlim[0],xmax=set_xlim[1])
    elif mr_axes == True and set_xlim != ["min","max"]:
        #log(q) scale axes are from 0 to negative value
        ax.set_xlim(xmin=set_xlim[1],xmax=set_xlim[0])


    ax.plot(lq,b_v,color = colors[0],label = r"$N^2$",linewidth = 1)
    #plots the Brunt-Vaisala frequency

    for i,l in enumerate(l):
        #plots the Lamb frequency for each degree "l" we want
        ax.plot(lq,lamb_l[i],color = colors[i+1], label =
        ↪  r"$L_{{{}}}$".format(l),linewidth = 1)

    ax.invert_xaxis()
    ax.set_ylabel(r"$log(N^{2})$")
    ax.legend()
    plt.savefig(image_path + file_name + ".png", dpi = 500, format =
    ↪  "png",bbox_inches='tight')
    plt.close()
```

```
def plot_chemical_fractions(stelum_data,image_path,file_name,mr_axes =
↪  False,set_xlim=["min","max"]):

    """Plotting of the chemical abundances"""

    h,he,c,o = stelum_data[:,4,0],stelum_data[:,4,1],stelum_data[:,4,2],stelum_data[:,4,3]
    #gathers the hydrogen, helium, carbon, oxygen abundances
    lq,mr = stelum_data[:,0,12],stelum_data[:,0,2]/(1.9884e+33) #converting to Msuns
    #gathers the log(q) and m(r) scales, changes the m(r) scale to Msun units

    colors = cm.rainbow(np.linspace(0, 1, 4)) #colors scheme

    #colors = cm.rainbow(np.linspace(0, 1, 4+4)) #color scheme Fig.~3.3

    fig,ax = plt.subplots()

    if mr_axes == True:
        #here if we want m(r) axes
        lq = mr
        ax.set_xlabel(r"m(r)")
        ax.set_xlim(xmin = np.nanmax(lq),xmax = np.nanmin(lq))
    else:
        #here if we want log(q) axes
        ax.set_xlabel(r"log(q)")
        ax.set_xlim(np.nanmin(lq),np.nanmax(lq))

    if mr_axes == False and set_xlim != ["min","max"]:
        #m(r) scale axes are from 0 to positive value
        ax.set_xlim(xmin=set_xlim[0],xmax=set_xlim[1])
    elif mr_axes == True and set_xlim != ["min","max"]:
        #log(q) scale axes are from 0 to negative value
        ax.set_xlim(xmin=set_xlim[1],xmax=set_xlim[0])
    chem_fractions = np.array([h,he,c,o])
    labels = ["H","He","C","O"]

    for i,chem in enumerate(chem_fractions):
        #plots each chemical abundances
        ax.plot(lq,chem,color = colors[i],label = labels[i],linewidth = 1)

    #below are special Fig.~3.3 plots
    #ax.axvline(x = -2,color = colors[4],linewidth = 1,label =
    ↪  "lq_env",linestyle="dashed")
    #ax.axvline(x = -4.5,color = colors[5],linewidth = 1,linestyle="dashed",label =
    ↪  "lq_diff")
    #ax.axvline(x = -0.25,color = colors[6],linewidth =
    ↪  1,linestyle="dashed",label="lq_core")
    #ax.hlines(y = 0.05,xmin = 0, xmax= -0.23 , color = colors[7],linewidth =
    ↪  1,linestyle="dashed",label="flash_c")

    ax.set_ylim(0,1)
    ax.set_ylabel("mass fraction")
    ax.invert_xaxis()
    ax.legend()
    plt.savefig(image_path + file_name + ".png", dpi = 500, format =
    ↪  "png",bbox_inches='tight')
    plt.close()
```

```
def plot_gradients(stelum_data,image_path,file_name,lq_env = -2, lq_core = -0.25,mtot =
↪  0.47,mr_axes = False,set_xlim=["min","max"],set_ylim=["min","max"],static = True):

    """Plotting of the real, adiabatic and radiative gradients"""

    grad_T = stelum_data[:,0,8] #real gradient
    grad_ad = stelum_data[:,0,9] #adiabatic gradient
    grad_rad = stelum_data[:,1,6] #radiative gradient
    lq,mr = stelum_data[:,0,12],stelum_data[:,0,2]/(1.9884e+33)
    #gathering log(q) and m(r) scales, converting m(r) scale in Msun

    colors = cm.rainbow(np.linspace(0, 1, 5)) #color scheme

    fig,ax = plt.subplots()

    if mr_axes == True:
        #here if we want m(r) axes
        lq = mr
        ax.set_xlabel(r"m(r)")
        ax.set_xlim(xmin = np.nanmax(lq),xmax = np.nanmin(lq))
        lq_env = (1-10**(lq_env))*mtot
        lq_core = (1-10**(lq_core))*mtot
        #converting transitions lq_env and lq_core log(q) values in m(r) ones
    else:
        #here if we want log(q) axes
        ax.set_xlabel(r"log(q)")
        ax.set_xlim(np.nanmin(lq),np.nanmax(lq))

    if mr_axes == False and set_xlim != ["min","max"]:
        #m(r) scale axes are from 0 to positive value
        ax.set_xlim(xmin=set_xlim[0],xmax=set_xlim[1])
    elif mr_axes == True and set_xlim != ["min","max"]:
        #log(q) scale axes are from 0 to negative value
        ax.set_xlim(xmin=set_xlim[1],xmax=set_xlim[0])

    if set_ylim != ["min","max"]:
        ax.set_ylim(ymin=set_ylim[0],ymax=set_ylim[1])

    #we plot all three gradients below
    ax.plot(lq,grad_T,color = colors[0],label = r"$\nabla_T$",linewidth = 1)
    ax.plot(lq,grad_ad,color = colors[1],label = r"$\nabla_{ad}$",linewidth = 1,linestyle
    ↪  = "dashed")
    ax.plot(lq,grad_rad,color = colors[2],label = r"$\nabla_{rad}$",linewidth =
    ↪  1,linestyle = "dashed")

    #we plot lq_env and lq_core transitions below
    ax.axvline(x = lq_env,color = colors[3],linewidth = 1,label = "lq_env")
    if static == True:
        ax.axvline(x = lq_core,color = colors[4],linewidth = 1,label = "lq_core")

    ax.set_ylabel("gradients")
    ax.invert_xaxis()
    ax.legend()
    plt.savefig(image_path + file_name + ".png", dpi = 500, format =
    ↪  "png",bbox_inches='tight')
    plt.close()
```

```
def plot_ledoux_term(stelum_data,image_path,file_name,mr_axes =
↪  False,set_xlim=["min","max"],set_ylim=["min","max"]):

    """Ledoux term plotting"""

    b = stelum_data[:,0,11] #gathering the Ledoux term
    lq,mr = stelum_data[:,0,12],stelum_data[:,0,2]/(1.9884e+33)
    #gathers log(q) and m(r) scale, converting m(r) scale to Msun

    colors = cm.rainbow(np.linspace(0, 1, 1)) #4 as in 4 chem_fractions to look at here

    fig,ax = plt.subplots()

    if mr_axes == True:
        #if we want m(r) scale
        lq = mr
        ax.set_xlabel(r"m(r)")
        ax.set_xlim(xmin = np.nanmax(lq),xmax = np.nanmin(lq))
    else:
        #if we want log(q) scale
        ax.set_xlabel(r"log(q)")
        ax.set_xlim(np.nanmin(lq),np.nanmax(lq))


    if mr_axes == False and set_xlim != ["min","max"]:
        #m(r) scale axes are from 0 to positive value
        ax.set_xlim(xmin=set_xlim[0],xmax=set_xlim[1])
    elif mr_axes == True and set_xlim != ["min","max"]:
        #log(q) scale axes are from 0 to negative value
        ax.set_xlim(xmin=set_xlim[1],xmax=set_xlim[0])

    if set_ylim != ["min","max"]:
        ax.set_ylim(ymin=set_ylim[0],ymax=set_ylim[1])

    ax.plot(lq,b,color = colors,label = "B",linewidth = 1)
    #plots the Ledoux term

    ax.set_ylabel("mass fraction")
    ax.invert_xaxis()
    ax.legend()
    plt.savefig(image_path + file_name + ".png", dpi = 500, format =
    ↪  "png",bbox_inches='tight')
    plt.close()
```

The PULSE plotter allows plotting pulsation spectra with PLOT_DELTAP() and the kinetic energy with PLOT_ECIN(). We can merge those two graphs with SUBPLOT_DELTAP_ECIN (actually, we do not really "merge" the graphs, rather, we compute them again, but to form a subplot). Those functions are given below in order.

```python
def plot_deltaP(pulse_data,mode_spacing,image_path,file_name,l=np.array([1,4])):

    """Plotting of the reduced period spacing pulsation spectra"""


    colors = cm.rainbow(np.linspace(0, 1, l.shape[0])) #color scheme

    fig,ax = plt.subplots()

    for i,l in enumerate(l):
        #iterates on "l", would we want multiple degrees to show

        if pulse_data[l-1,0,:].shape[0] == 12:
            #static models pulse data arrays have a given number of datas
            #we use this to find if we plot a static or evolutive model
            reduced_periods = pulse_data[l-1,:,4]*np.sqrt(l*(l+1))
            reduced_period_spacing = mode_spacing[l-1,:]*np.sqrt(l*(l+1))
            #we also compute reduced periods and period spacing at the same time

        else:
            reduced_periods = pulse_data[l-1,:,2]*np.sqrt(l*(l+1))
            reduced_period_spacing = mode_spacing[l-1,:]*np.sqrt(l*(l+1))
            #computes reduced periods and period spacing

        ax.plot(reduced_periods,reduced_period_spacing,color=colors[i],label = "l = {}".format(l), linewidth = 1)
        #plots the reduced periods spacing pulsation spectra
        ax.scatter(reduced_periods,reduced_period_spacing,color=colors[i], s = 3)
        #plots a dot for each radial order

    ax.set_ylabel(r"$\Delta P\times\sqrt{l(l+1)}$")
    ax.set_xlabel(r"$P\times\sqrt{l(l+1)}$")

    ax.legend()
    ax.set_ylim(50,500)
    ax.set_xlim(0,25000)

    plt.savefig(image_path + file_name + ".png", dpi = 500, format = "png",bbox_inches='tight')
    plt.close()
```

```python
def plot_Ecin(pulse_data,image_path,file_name,l=np.array([1,4]),set_ylim=np.array([40.5,49])):

    """Plotting of the kinetic energy"""

    colors = cm.rainbow(np.linspace(0, 1, l.shape[0]))

    fig,ax = plt.subplots()

    for i,l in enumerate(l):

        if pulse_data[l-1,0,:].shape[0] == 12:
            #static models pulse data arrays have a given number of datas
            #we use this to find if we plot a static or evolutive model
            reduced_periods = pulse_data[l-1,:,4]*np.sqrt(l*(l+1))
            reduced_periods[reduced_periods==0] = np.nan
            Ecin = pulse_data[l-1,:,7]
            #Computes the reduced periods and gather Ecin data
        else:
            #Computes the reduced periods and gather Ecin data
            reduced_periods = pulse_data[l-1,:,2]*np.sqrt(l*(l+1))
            reduced_periods[reduced_periods==0] = np.nan
            Ecin = pulse_data[l-1,:,4]


        ax.plot(reduced_periods,Ecin,color=colors[i],label = "l = {}".format(l), linewidth = 1)
        #plots the kinetic energy against the reduced periods
        ax.scatter(reduced_periods,Ecin,color=colors[i], s = 3)
        #plots a dot for each radial order

    ax.set_xlabel(r"$P\times\sqrt{l(l+1)}$")
    ax.set_ylabel(r"E$_{kin}$")

    ax.legend()
    ax.set_xlim(0,25000)
    ax.set_ylim(set_ylim[0],set_ylim[1])

    plt.savefig(image_path + file_name + ".png", dpi = 500, format = "png",bbox_inches='tight')
    plt.close()
```

```
def subplot_deltaP_Ecin(pulse_data,mode_spacing,image_path,file_name,l=np.array([1,4]),set_ylim_ekin=np.array([40.5,49])):

    """Plotting of a double plot of pulsation spectra and kinetic energy"""

    fig,ax = plt.subplots(2,1,sharex=True)
    fig.subplots_adjust(hspace=0)

    colors = cm.rainbow(np.linspace(0, 1, l.shape[0])) #color scheme


    for i,l in enumerate(l):

        if pulse_data[l-1,0,:].shape[0] == 12:
            #static models pulse data arrays have a given number of datas
            #we use this to find if we plot a static or evolutive model
            reduced_periods = pulse_data[l-1,:,4]*np.sqrt(l*(l+1))
            reduced_period_spacing = mode_spacing[l-1,:]*np.sqrt(l*(l+1))
            reduced_periods[reduced_periods==0] = np.nan
            Ecin = pulse_data[l-1,:,7]
            #Computes the reduced periods, reduced period spacings and gather Ecin data
        else:
            #Computes the reduced periods, reduced period spacings and gather Ecin data
            reduced_periods = pulse_data[l-1,:,2]*np.sqrt(l*(l+1))
            reduced_periods[reduced_periods==0] = np.nan
            Ecin = pulse_data[l-1,:,4]
            reduced_period_spacing = mode_spacing[l-1,:]*np.sqrt(l*(l+1))

        ax[0].plot(reduced_periods,reduced_period_spacing,color=colors[i], linewidth = 1)
        #plots the pulsation spectra on the top part
        ax[0].scatter(reduced_periods,reduced_period_spacing,color=colors[i], s = 3)
        #plots a dot for each radial order

        ax[1].plot(reduced_periods,Ecin,color=colors[i],label = "l = {}".format(l),
        ↪  linewidth = 1)
        #plots the kinetic energy on the bottom part
        ax[1].scatter(reduced_periods,Ecin,color=colors[i], s = 3)
        #plots a dot for each radial order

    ax[0].set_ylabel(r"$\Delta P\times\sqrt{l(l+1)}$")
    ax[1].set_ylabel(r"E$_{kin}$")

    ax[1].set_xlabel(r"$P\times\sqrt{l(l+1)}$")

    ax[1].legend()
    ax[0].set_ylim(50,500)
    ax[1].set_ylim(set_ylim_ekin[0],set_ylim_ekin[1])

    ax[1].set_xlim(0,25000)

    plt.savefig(image_path + file_name + ".png", dpi = 500, format =
    ↪  "png",bbox_inches='tight')
    plt.close()
```

The last of the main plotters is the EIG plotter, which can plot the weight function with PLOT_WFI().

```
def plot_wfi(eig_data,image_path,file_name,k,mr_axes = False,lq_env = -2,lq_core =
↪  -0.25,mtot = 0.47,set_xlim=["min","max"], static = False):

    wfi = eig_data[k-1,:,9] #weight functions
    lq = eig_data[k-1,:,10] #log(q) scale

    colors = cm.rainbow(np.linspace(0, 1, 3)) #color scheme

    fig,ax = plt.subplots()

    if mr_axes == True:
        #here, if we want axes in m(r) scale
        lq = (1-10**(lq))*mtot #changes the lq data to m(r) scale
        lq_env = (1-10**(lq_env))*mtot #for the transition lq_env aswell
        plt.axvline(x = lq_env,color = colors[1],linewidth = 1,label = "lq_env",zorder =
        ↪  5)
        #plots lq_env line
        if static == True:
            #below, same for lq_core
            lq_core = (1-10**(lq_core))*mtot
            plt.axvline(x = lq_core,color = colors[2],linewidth = 1,label =
            ↪  "lq_core",zorder = 10)
        ax.set_xlabel(r"$m(r)$")
    else:
        #here if we want axes in log(q)
        plt.axvline(x = lq_env,color = colors[1],linewidth = 1,label = "lq_env",zorder =
        ↪  5)
        if static == True:
            plt.axvline(x = lq_core,color = colors[2],linewidth = 1,label =
            ↪  "lq_core",zorder = 10)
        ax.set_xlabel(r"$log(q)$")
        ax.invert_xaxis()

    if mr_axes == False and set_xlim != ["min","max"]:
        #m(r) scale axes are from 0 to positive value
        ax.set_xlim(xmin=set_xlim[0],xmax=set_xlim[1])
        ax.invert_xaxis()
    elif mr_axes == True and set_xlim != ["min","max"]:
        #log(q) scale axes are from 0 to negative value
        ax.set_xlim(xmin=set_xlim[0],xmax=set_xlim[1])

    ax.plot(lq,wfi,color = colors[0],linewidth = 1,zorder = 0)
    #weight function against m(r) or log(q) axes

    ax.set_ylabel(r"wfi")
    ax.legend()
    ax.set_ylim(0,1)
    plt.savefig(image_path + file_name + ".png", dpi = 500, format = "png",bbox_inches='tight')
    plt.close()
```

We then find the plotter for the propagation diagrams, which is composed of a single function PLOT_PROPAGATION_DIAGRAM(), which using, EIG, PULSE and STELUM file data at the same time, which is why it is apart from other three more “classic” plotters.

```
def plot_propagation_diagram(eig_data,stelum_data,pulse_data,image_path,file_name,l=np.array([1]),mr_axes=False,set_xlim=["min","max"],y=[1]):

    b_v,lamb_l,lq,mr = stelum.compute_bv_lamb(stelum_data,l)
    #gathers the brunt vaisala frequency, lamb frequency, log(q) scale, m(r) scale

    colors = cm.rainbow(np.linspace(0, 1, 1 + l.shape[0] + len(y))) #color scheme

    fig,ax = plt.subplots()

    if mr_axes == True:
        #if we want m(r) axes, we convert log(q) values in m(r) ones below
        lq = mr
        ax.set_xlabel(r"m(r)")
        ax.set_ylim(-15,0)
        ax.set_xlim(xmin = np.nanmax(lq),xmax = np.nanmin(lq))
    else:
        #else, we keep it the way it is
        ax.set_xlabel(r"log(q)")
        ax.set_ylim(-5,0)
        ax.set_xlim(np.nanmin(lq),np.nanmax(lq))

    if mr_axes == False and set_xlim != ["min","max"]:
        #m(r) scale axes are from 0 to positive value
        ax.set_xlim(xmin=set_xlim[0],xmax=set_xlim[1])
    elif mr_axes == True and set_xlim != ["min","max"]:
        #log(q) scale axes are from 0 to negative value
        ax.set_xlim(xmin=set_xlim[1],xmax=set_xlim[0])

    ax.invert_xaxis()
    ax.set_ylabel("log" + r'$(N^{2})$')

    ax.plot(lq,b_v,color = colors[0],label = r"$N^2$",linewidth = 1) #plots brunt vaisala frequency

    for i,l in enumerate(l):
        #plots each Lamb frequency (those we want)
        ax.plot(lq,lamb_l[i],color = colors[i+1], label =
        ↪  r"$L_{{{}}}$".format(l),linewidth = 1)

    if pulse_data[l-1,0,:].shape[0] == 12:
        #static models pulse data arrays have a given number of datas
        #we use this to find if we plot a static or evolutive model
        periods = pulse_data[l-1,:,4]
    else:
        periods = pulse_data[l-1,:,2]

    zeros = eig.zero_finder(eig_data) #finds the zero for y1, y2 eigenfunctions
    sigmas = np.log10((2*np.pi/periods)**2) #finds the log(sigma^2) horizontal lines for
    ↪  each modes

    colors = ["red","green","black","blue"] #for y1,y2,y3,y4 if ever needed, because color
    ↪  scheme is not nice for y2 otherwise

    if mr_axes == True:
        #need to displace each zeros if we don't take the log(q) scale
        #because .eig files give log(q) scale data
        zeros = (1-10**(zeros))*mr[-1]

    for j in range(len(y)):
        for k in range(zeros.shape[1]):
            plt.axhline(sigmas[k],linewidth = 0.3,alpha = 0.3, color = "red") #plots each
            ↪  log(sigma^2)
            ax.scatter(zeros[j,k,:],np.array([sigmas[k]]*(zeros.shape[1])),alpha = 1,color
            ↪  = colors[j],s=0.25)
            #plots each red dots for the zeros of y1 or y2 or both (as we want)

    ax.set_ylim(np.nanmin(sigmas) - 0.05,0)
    #off set the bottom limit of the graph otherwise we do not see nicely k = 65-70 modes
    ax.legend()

    plt.savefig(image_path + file_name + ".png", format="png", dpi = 1000,
    ↪  bbox_inches='tight')
```

Finally, two isolated plotting programs compute and plot the cancelling factor and spherical harmonics. We give the cancelling factor program first :

```
import matplotlib.pyplot as plt
from matplotlib import cm, colors
from mpl_toolkits.mplot3d import Axes3D
from scipy.special import legendre
import numpy as np

l = np.array([0,1,2,3,4,5,6,7,8,9,10])
#defining the degree "l" over which we want the cancelling factor

theta = np.linspace(0, np.pi/2, 200)#defines the theta in spherical symmetry
c_l0 = np.sqrt((2*l + 1)/(4*np.pi)) #normalisation factor

integral = np.zeros(l.shape[0]) #init array containing the integrals

for i in range(l.shape[0]):

    poly = legendre(l[i])
    #array containing legendre polynomials of degree "l"
    for j in range(theta.shape[0]-1):

        #computing spherical harmonics
        f_a = poly(np.cos(theta[j]))*np.sin(theta[j])*np.cos(theta[j])
        f_b = poly(np.cos(theta[j+1]))*np.sin(theta[j+1])*np.cos(theta[j+1])

        dr = theta[j+1] - theta[j] #definining the minimal displacement (\pi/400)

        integral[i] += (1/2)*(f_a + f_b)*dr #trapezoidal integral

    integral[i] = integral[i]*c_l0[i] #accounts the normalizing factor in


fig, ax = plt.subplots()
ax.plot(l,7.0875*integral, linewidth=1)
ax.set_xlim(0,8)
ax.axhline(y=0,linewidth = 1)
ax.set_xlabel("azimuthal order")
ax.set_ylabel("partial cancelling")
plt.savefig("C:/Master_Thesis/Articles_and_Books/Notes_Draft/Images_overleaf/cancelling.png", dpi = 500, format = "png",bbox_inches='tight')
```

And then the speriical harmonics one :

```
import matplotlib.pyplot as plt
from matplotlib import cm, colors
from mpl_toolkits.mplot3d import Axes3D
import numpy as np
from scipy.special import sph_harm


def set_axes_equal(ax: plt.Axes):

    """Credits : https://stackoverflow.com/questions/13685386/how-to-set-the-equal-aspect-ratio-for-all-axes-x-y-z
       Answer of : AndrewCox and edited by Trenton McKinney

    Set 3D plot axes to equal scale.
    This is needed since

    Make axes of 3D plot have equal scale so that spheres appear as
    spheres and cubes as cubes.  Required since `ax.axis('equal')`
    and `ax.set_aspect('equal')` don't work on 3D.
    """

    limits = np.array([
        ax.get_xlim3d(),
        ax.get_ylim3d(),
        ax.get_zlim3d(),
    ])
    origin = np.mean(limits, axis=1)
    radius = 0.5 * np.max(np.abs(limits[:, 1] - limits[:, 0]))
    _set_axes_radius(ax, origin, radius)

```

```
def _set_axes_radius(ax, origin, radius):

    """Credits : https://stackoverflow.com/questions/13685386/how-to-set-the-equal-aspect
    ↪  -ratio-for-all-axes-x-y-z
       Answer of : AndrewCox and edited by Trenton McKinney

    Called in the previous function"""

    x, y, z = origin
    ax.set_xlim3d([x - radius, x + radius])
    ax.set_ylim3d([y - radius, y + radius])
    ax.set_zlim3d([z - radius, z + radius])

#below, we define spherical coordinates phi and theta
phi = np.linspace(0, np.pi, 200)
theta = np.linspace(0, 2*np.pi, 200)
phi, theta = np.meshgrid(phi, theta)

#we gather the points located on a sphere
x = np.sin(phi) * np.cos(theta)
y = np.sin(phi) * np.sin(theta)
z = np.cos(phi)

m, l = 2,3 #azimuthal order and degree

#normalized spherical harmonics are computed below
coloring = sph_harm(m, l, theta, phi).real
max, min = coloring.max(), coloring.min()
coloring = (coloring - min)/(max - min)

fig = plt.figure(figsize=(6,6))
ax = fig.add_subplot(111, aspect='auto', projection='3d')
ax.plot_surface(x, y, z,  rstride=1, cstride=1, facecolors=cm.seismic(coloring))
#plotting of the spherical harmonics on the sphere
ax.view_init(elev=0, azim=180)
ax.set_axis_off()
ax.set_box_aspect([1,1,1])
set_axes_equal(ax)
plt.savefig("D:/Master_Thesis/Articles_and_Books/Notes_Draft/Images_overleaf/spherical_ha
↪  rmonic_save/Y_3_2_test.png", dpi = 500, format =
↪  "png",bbox_inches='tight')
plt.close()
```

A.3 Mains

The *main* programs link the plotters and interpreters together. Those main programs should be looked at adapative ones, where we choose which graphs we want to compute, or not compute. We give first the main program for static models, which loops on a static model to get the graphs for each different core helium abundances organized into a same folder.

```
import numpy as np
import re
import os
import matplotlib
from matplotlib import pyplot as plt
import math
```

```
from matplotlib import cm

matplotlib.use("Agg")

from interpreters import eig_interp as eig
from interpreters import pulse_interp as pulse
from interpreters import stelum_interp as stelum

from plotters import eig_plots as eig_plt
from plotters import pulse_plots as pulse_plt
from plotters import stelum_plots as stelum_plt
from plotters import propagation_diagram_plot as propadiag_plt

def main_plotting():

    if os.path.isdir(image_pulse_main) == False:
        os.mkdir(image_pulse_main)

    if os.path.isdir(image_stelum_main) == False:
        os.mkdir(image_stelum_main)

    if os.path.isdir(image_eig_main) == False:
        os.mkdir(image_eig_main)

    Files_pulse = os.listdir(main_pulse_string)

    for i,file in enumerate(Files_pulse):

        print(run, file)

        file_pulse = Files_pulse[i]
        image_main = "_" + "md" + str(i+1).zfill(4)
        file_stelum = "model" + Files_pulse[i][5:]
        file_eig = run + "_" + str(i+1) + ".eig"


        pulse_data,mode_spacing = pulse.extracting(main_pulse_string +
        ↪   file_pulse,mode_spacing=True)
        stelum_data = stelum.extracting(main_stelum_string + file_stelum)

        He_core = stelum_data[0,4,1]

        file_name_main = "md" + str(i+1).zfill(4) + "-{}".format(He_core)

        pulse_image = image_pulse_main + file_name_main + "/"
        if os.path.isdir(pulse_image) == False:
            os.mkdir(pulse_image)

        stelum_image = image_stelum_main + file_name_main + "/"
        if os.path.isdir(stelum_image) == False:
            os.mkdir(stelum_image)

        if no_eig == False:
            try:
                eig_data = eig.extracting(main_eig_string + file_eig)
                image_eig = image_eig_main + file_name_main + "/"
                if os.path.isdir(image_eig) == False:
                    os.mkdir(image_eig)
                for k in range(eig_data.shape[0]):
                    mode_path = image_eig + str(k+1) + "/"
                    if os.path.isdir(mode_path) == False:
                        os.mkdir(mode_path)
                    wfi_name = "wfi_logq_" + str(k+1) + image_main
                    eig_plt.plot_wfi(eig_data,mode_path,wfi_name,k=k+1,lq_env = lq_env,lq_core = lq_core,static=True)
                    wfi_name = "wfi_mr_" + str(k+1) + image_main
                    eig_plt.plot_wfi(eig_data,mode_path,wfi_name,k=k+1,lq_env = lq_env,lq_core = lq_core,mr_axes=True,static=True)
                    wfi_name = "wfi_logq_h_he" + str(k+1) + image_main
                    eig_plt.plot_wfi(eig_data,mode_path,wfi_name,k=k+1,lq_env = lq_env,lq_core = lq_core,set_xlim=[lq_env-1,lq_env+1],static=True)

                propagation_path = image_eig + "propagation_diagram/"
                if os.path.isdir(propagation_path) == False:
                    os.mkdir(propagation_path)
                propa_name = "propagation_diagram_logq" + image_main
                propadiag_plt.plot_propagation_diagram(eig_data,stelum_data,pulse_data,propagation_path,propa_name)
                propa_name = "propagation_diagram_mr" + image_main
                propadiag_plt.plot_propagation_diagram(eig_data,stelum_data,pulse_data,propagation_path,propa_name,mr_axes=True)
                propa_name = "propagation_diagram_logq_h_he" + image_main
                propadiag_plt.plot_propagation_diagram(eig_data,stelum_data,pulse_data,propagation_path,propa_name,set_xlim=[lq_env-1,lq_env+1])

            except:
                excepted = True

        if no_pulse == False:
            deltaP_name = "deltaP" + image_main
            pulse_plt.plot_deltaP(pulse_data,mode_spacing,pulse_image,deltaP_name,l=larray)
            Ekin = "Ekin" + image_main
            pulse_plt.plot_Ecin(pulse_data,pulse_image,Ekin,l=larray)
            Ekin = "Ekin_zoomed" + image_main
            pulse_plt.plot_Ecin(pulse_data,pulse_image,Ekin,l=larray,set_ylim=np.array([42,45]))
            subplot = "subplot_deltaP_Ekin" + image_main
            pulse_plt.subplot_deltaP_Ecin(pulse_data,mode_spacing,pulse_image,subplot,l=larray)
            subplot = "subplot_deltaP_Ekin_zoom" + image_main
            pulse_plt.subplot_deltaP_Ecin(pulse_data,mode_spacing,pulse_image,subplot,l=larray,set_ylim_ekin=np.array([42,45]))

        if no_stelum == False:
            bv = "brunt_vaisala_lamb_logq" + image_main
            stelum_plt.plot_BV_Lamb(stelum_data,stelum_image,bv)
            bv = "brunt_vaisala_lamb_mr" + image_main
            stelum_plt.plot_BV_Lamb(stelum_data,stelum_image,bv,mr_axes=True)
            bv = "brunt_vaisala_lamb_logq_h_he" + image_main
            stelum_plt.plot_BV_Lamb(stelum_data,stelum_image,bv,set_xlim=[lq_env-1,lq_env+1])

            chem = "chemical_fraction_logq" + image_main
            stelum_plt.plot_chemical_fractions(stelum_data,stelum_image,chem)
            chem = "chemical_fraction_mr" + image_main
            stelum_plt.plot_chemical_fractions(stelum_data,stelum_image,chem,mr_axes=True)
            chem = "chemical_fraction_logq_h_he" + image_main
            stelum_plt.plot_chemical_fractions(stelum_data,stelum_image,chem,set_xlim=[lq_env-1,lq_env+1])
```

```python

            ledoux = "ledoux_term_logq" + image_main
            stelum_plt.plot_ledoux_term(stelum_data,stelum_image,ledoux)
            ledoux = "ledoux_term_mr" + image_main
            stelum_plt.plot_ledoux_term(stelum_data,stelum_image,ledoux,mr_axes=True)
            ledoux = "ledoux_term_logq_h_he" + image_main
            stelum_plt.plot_ledoux_term(stelum_data,stelum_image,ledoux,set_xlim=[lq_env-1,lq_env+1])

            gradient = "gradient_term_logq" + image_main
            stelum_plt.plot_gradients(stelum_data,stelum_image,gradient,lq_core=lq_core)
            gradient = "gradient_term_mr" + image_main
            stelum_plt.plot_gradients(stelum_data,stelum_image,gradient,lq_core=lq_core,mr_axes=True)
            gradient = "gradient_term_logq_h_he" + image_main
            stelum_plt.plot_gradients(stelum_data,stelum_image,gradient,lq_core=lq_core,set_xlim=[lq_env-1,lq_env+1])

run_array = ["R0005_lqcore_010"]
ref_run = ["R0005"]
end_sequence = ["L1_ONLY_lqcore__010"]
lq_env_array = [-2]
lq_core_array = [-0.10]

for i,run in enumerate(run_array):

    main_pulse_string = "C:/Master_Thesis/Data/Runs/" + run + "/PULSE-R51-SDB/"
    main_stelum_string = "C:/Master_Thesis/Data/Runs/" + run + "/STELUM_R51_SDB_EVO/"
    main_eig_string = "C:/Master_Thesis/Data/Runs/" + run + "/EIG/"

    no_stelum = True
    no_pulse = False
    no_eig = True

    larray = l=np.array([1])

    if os.path.isdir("C:/Master_Thesis/Data/Image_new/Static/" + ref_run[i] + "/") == False:
        os.mkdir("C:/Master_Thesis/Data/Image_new/Static/" + ref_run[i] + "/")

    image_new = "C:/Master_Thesis/Data/Image_new/Static/" + ref_run[i] + "/" + end_sequence[i] + "/"

    if os.path.isdir(image_new) == False:
        os.mkdir(image_new)

    image_pulse_main = image_new + "PULSE/"
    image_stelum_main= image_new + "STELUM/"
    image_eig_main = image_new + "EIG/"

    lq_env = lq_env_array[i]
    lq_core = lq_core_array[i]

    main_plotting()
```

Similarly, the main program for evolutionary models :

```python
import numpy as np
import re
import os
import matplotlib
from matplotlib import pyplot as plt
import math
from matplotlib import cm

matplotlib.use("Agg")

from interpreters import eig_interp as eig
from interpreters import pulse_interp as pulse
from interpreters import stelum_interp as stelum

from plotters import eig_plots as eig_plt
from plotters import pulse_plots as pulse_plt
from plotters import stelum_plots as stelum_plt
from plotters import propagation_diagram_plot as propadiag_plt

def main_plotting():

    if os.path.isdir(image_pulse_main) == False:
        os.mkdir(image_pulse_main)

    if os.path.isdir(image_stelum_main) == False:
        os.mkdir(image_stelum_main)

    if os.path.isdir(image_eig_main) == False:
        os.mkdir(image_eig_main)

    Files_pulse = os.listdir(main_pulse_string)

    for i,file in enumerate(Files_pulse):

        print(dh,file)

        file_pulse = Files_pulse[i]
        image_main = "_" + Files_pulse[i][:6]
        file_stelum = Files_pulse[i][:6] + "_modified.txt"
        file_eig = Files_pulse[i][:6] + ".eig"

        pulse_data,mode_spacing = pulse.extracting(main_pulse_string +
        ↪  file_pulse,mode_spacing=True)
        stelum_data = stelum.extracting(main_stelum_string + file_stelum)

        He_core = stelum_data[0,4,1]

        file_name_main = "md" + str(i+1).zfill(4) + "-{}".format(He_core)

        pulse_image = image_pulse_main + file_name_main + "/"
        if os.path.isdir(pulse_image) == False:
            os.mkdir(pulse_image)

        stelum_image = image_stelum_main + file_name_main + "/"
        if os.path.isdir(stelum_image) == False:
            os.mkdir(stelum_image)

        if no_eig == False:
            eig_data = eig.extracting(main_eig_string + file_eig)
            image_eig = image_eig_main + file_name_main + "/"
            if os.path.isdir(image_eig) == False:
                os.mkdir(image_eig)
            for k in range(eig_data.shape[0]):
                mode_path = image_eig + str(k+1) + "/"
                if os.path.isdir(mode_path) == False:
                    os.mkdir(mode_path)
                wfi_name = "wfi_logq_" + str(k+1) + image_main
                eig_plt.plot_wfi(eig_data,mode_path,wfi_name,k=k+1,lq_env =
                ↪  (-int(dh)*(1e-2)))
                wfi_name = "wfi_mr_" + str(k+1) + image_main
                eig_plt.plot_wfi(eig_data,mode_path,wfi_name,k=k+1,lq_env =
                ↪  (-int(dh)*(1e-2)),mr_axes=True)
                wfi_name = "wfi_mr_sc_" + str(k+1) + image_main
                eig_plt.plot_wfi(eig_data,mode_path,wfi_name,k=k+1,lq_env =
                ↪  (-int(dh)*(1e-2)),mr_axes=True,set_xlim=[0.10,0.30])

            propagation_path = image_eig + "propagation_diagram/"
            if os.path.isdir(propagation_path) == False:
                os.mkdir(propagation_path)
            propa_name = "propagation_diagram_logq" + image_main
            propadiag_plt.plot_propagation_diagram(eig_data,stelum_data,pulse_data,propag⌋
            ↪  ation_path,propa_name)
            propa_name = "propagation_diagram_mr" + image_main
            propadiag_plt.plot_propagation_diagram(eig_data,stelum_data,pulse_data,propag⌋
            ↪  ation_path,propa_name,mr_axes=True)
            propa_name = "propagation_diagram_mr_sc" + image_main
            propadiag_plt.plot_propagation_diagram(eig_data,stelum_data,pulse_data,propag⌋
            ↪  ation_path,propa_name,mr_axes=True,set_xlim=[0.10,0.30])


        if no_pulse == False:
            deltaP_name = "deltaP" + image_main
            pulse_plt.plot_deltaP(pulse_data,mode_spacing,pulse_image,deltaP_name)
            Ekin = "Ekin" + image_main
            pulse_plt.plot_Ecin(pulse_data,pulse_image,Ekin)
            subplot = "subplot_deltaP_Ekin" + image_main
            pulse_plt.subplot_deltaP_Ecin(pulse_data,mode_spacing,pulse_image,subplot)

        if no_stelum == False:
            bv = "brunt_vaisala_lamb_logq" + image_main
            stelum_plt.plot_BV_Lamb(stelum_data,stelum_image,bv)
            bv = "brunt_vaisala_lamb_mr" + image_main
            stelum_plt.plot_BV_Lamb(stelum_data,stelum_image,bv,mr_axes=True)
            bv = "brunt_vaisala_lamb_mr_sc" + image_main
            stelum_plt.plot_BV_Lamb(stelum_data,stelum_image,bv,mr_axes=True,set_xlim=[0.⌋
            ↪  10,0.30])

            chem = "chemical_fraction_logq" + image_main
            stelum_plt.plot_chemical_fractions(stelum_data,stelum_image,chem)
            chem = "chemical_fraction_mr" + image_main
            stelum_plt.plot_chemical_fractions(stelum_data,stelum_image,chem,mr_axes=True)
            chem = "chemical_fraction_mr_sc" + image_main
            stelum_plt.plot_chemical_fractions(stelum_data,stelum_image,chem,mr_axes=True⌋
            ↪  ,set_xlim=[0.10,0.30])

            ledoux = "ledoux_term_logq" + image_main
            stelum_plt.plot_ledoux_term(stelum_data,stelum_image,ledoux)
            ledoux = "ledoux_term_mr" + image_main
            stelum_plt.plot_ledoux_term(stelum_data,stelum_image,ledoux,mr_axes=True)
            ledoux = "ledoux_term_mr_sc" + image_main
            stelum_plt.plot_ledoux_term(stelum_data,stelum_image,ledoux,set_xlim=[0.10,0.30],mr_axes = True)

            gradient = "gradient_term_logq" + image_main
            stelum_plt.plot_gradients(stelum_data,stelum_image,gradient,lq_env = (-int(dh)*(1e-2)),static = False)
            gradient = "gradient_term_mr" + image_main
            stelum_plt.plot_gradients(stelum_data,stelum_image,gradient,lq_env = (-int(dh)*(1e-2)),mr_axes=True,static = False)
            gradient = "gradient_term_mr_sc" + image_main
            stelum_plt.plot_gradients(stelum_data,stelum_image,gradient,lq_env = (-int(dh)*(1e-2)),set_xlim=[0.10,0.30],set_ylim=[0.375,0.4],mr_axes = True,static = False)

dh = ["200"]

for i,dh in enumerate(dh):

    main_pulse_string = "C:/Master_Thesis/Data/Runs/Evol_models_pulseR51/SDB-ZVAR-M04699530-DH"+ dh +"-C3600/PULSE/"
    main_stelum_string = "C:/Master_Thesis/Data/Runs/Evol_models/Evol_models_modified/SDB-ZVAR-M04699530-DH"+ dh +"-C3600/"
    main_eig_string = "C:/Master_Thesis/Data/Runs/Evol_models_pulseR51/SDB-ZVAR-M04699530-DH"+ dh +"-C3600/EIG/"

    no_stelum = False
    no_pulse = True
    no_eig = True

    if os.path.isdir("C:/Master_Thesis/Data/Image_new/Evol/DH" + dh + "/") == False:
        os.mkdir("C:/Master_Thesis/Data/Image_new/Evol/DH" + dh + "/")

    image_new = "C:/Master_Thesis/Data/Image_new/Evol/DH" + dh + "/" + "Pure/"

    if os.path.isdir(image_new) == False:
        os.mkdir(image_new)

    image_pulse_main = image_new + "PULSE/"
    image_stelum_main= image_new + "STELUM/"
    image_eig_main = image_new + "EIG/"


    main_plotting()
```

A.4 Asymptotic computations

Finally, we find a program to compute the asymtotical period spacings, both in our models and in theory, as well as the size of the convective core, the overshooting zone and the semi-convective zone. This program is only used for static models, and we separate it in three different parts. First, we find the function which is used to compute the size of our zones, REGION_SIZES(). It uses the indexes which delimit the different regions, found with SEARCH_INDEX().

```python
def search_index(stelum_data):

    """Finds the bottom and top of :

    - convective core
    - overshooting zone
    - semi-convective zone"""

    #Init conditions to stop searching given indexes
    carbon = stelum_data[:,4,2]
    sch = stelum_data[:,0,14]

    indexes = np.zeros(3*2).reshape(2,3)

    for i in range(carbon.shape[0]):

        if sch[i] == 0 and indexes[1,0] == 0:
            indexes[0,0],indexes[1,0] = i,1
            #top of convective core
        if np.absolute(carbon[i] - carbon[i+1]) > 1e-4 and indexes[1,1] == 0:
            indexes[0,1],indexes[1,1] = i,1
            #top of overshooting zone
        if carbon[i] < 1e-8 and indexes[1,2] == 0:
            indexes[0,2],indexes[1,2] = i,1
            #top of semi-convective zone
        if np.sum(indexes[1,:]) == 3:
            return(indexes[0,:])
```

```python
def region_sizes(stelum_data,indexes):

    indexes = indexes.astype(int)

    mr = stelum_data[:,0,2]/(1.9884e+33)
    #m(r) in Msun

    #we find convective core size, overshooting size, SC zone size
    convective_core_size = mr[indexes[0]] - mr[0]
    overshooting_size = mr[indexes[1]] - mr[indexes[0]]
    SC_size = mr[indexes[2]] - mr[indexes[1]]

    sizes = np.array([convective_core_size,overshooting_size,SC_size])

    return(sizes)
```

We then plot those using PLOT_SIZES().

```python
def plot_sizes(sizes,he_core,img_string,list):

### for different zone sizes ###

colors = cm.rainbow(np.linspace(0, 1, len(list)))

fig,ax = plt.subplots()
for i,dh in enumerate(list):
    lq_env = (-int(dh)*(1e-2))
    ax.plot(he_core[i,:],sizes[i,:,0], label = "lq_env = {}".format(lq_env), color =
    ↪  colors[i], linewidth = 1)
    ax.plot(he_core[i,:],sizes[i,:,1], color = colors[i], linewidth = 1)
    ax.plot(he_core[i,:],sizes[i,:,2], color = colors[i], linewidth = 1)

ax.set_xlabel("core_he")
ax.set_ylabel(r"$\Delta m(r)$ in $ M_{sun}$")
ax.set_xlim(0,1)
ax.invert_xaxis()
ax.legend(fontsize = "xx-small")
plt.savefig(img_string + "sizes.png", dpi = 500, format = "png",bbox_inches = 'tight')
plt.close()
```

We can also compute the theoretical asymptotic spacings (for modes trapped in a given zone, and the mean asymptotic spacing), with THEORETICAL_PERIOD_SPACINGS() which uses TRAPEZOIDAL_INTEGRAL() to do the integrals necessary for the theoretical period spacing computations.

```
def trapezoidal_integral(f,bottom,top,variable):

    integral = 0

    for i in range(bottom,top):

        if (math.isnan(f[i]) == False) and (math.isnan(f[i+1]) == False):
            #this notably prevents issues with
            #the partial ionization zone of HeII/HeIII at log(q) = -10ish

            f_a,f_b = f[i],f[i+1] #trapezoidal integral
            dr = variable[i+1] - variable[i]

            integral += (1/2)*(f_a + f_b)*dr

    return(integral)
```

```
def theoretical_period_spacings(stelum_data,indexes):

    """Computes the theoretical reduced period spacings"""

    indexes = indexes.astype(int)
    reduced = 2*(np.pi)**2

    log10N2, lamb, lq, mr = stelum.compute_bv_lamb(stelum_data,lamb=np.array([1]))
    #getting the Brunt Vaisala frequency
    r = stelum_data[:,0,1]

    abs_N = np.absolute(np.sqrt(10**(log10N2)))
    #getting |N|
    f = abs_N/r #this is the term in the integral

    surface_index = r.shape[0]-1 #not taking the surface point, otherwise index error

    trapped_sc_carbon = trapezoidal_integral(f,indexes[1],indexes[2],r)
    #spacing trapped in semi-convective zone
    trapped_sc_sch = trapezoidal_integral(f,indexes[0],indexes[2],r)
    #spacing trapped in semi-convective + overshooting zone
    asympt_spacing = trapezoidal_integral(f,indexes[0],surface_index,r)
    #mean spacing (global)

    trapped_sc_carbon = reduced/(trapped_sc_carbon)
    trapped_sc_sch = reduced/(trapped_sc_sch)
    asympt_spacing = reduced/(asympt_spacing)
    #to get the *actual* spacing

    integrals = np.array([trapped_sc_sch,trapped_sc_carbon,asympt_spacing])

    return(integrals)
```

Which we plot using PLOT_INTEGRALS() :

```
def plot_integrals(integrals,he_core,img_string):

    fig,ax = plt.subplots()
    ### trapped modes integrals ###
    ax.plot(he_core,integrals[:,0], label = "overshooting + SC", color = "blue", linewidth
    ↪  = 1)
    ax.plot(he_core,integrals[:,1], label = "SC", color = "red", linewidth = 1)

    ax.set_xlabel("core_he")
    ax.set_ylabel(r"$\Pi_{T,l}\sqrt{l(l+1)}$")
    ax.set_xlim(0,1)
    ax.invert_xaxis()
    ax.legend(fontsize = "x-small")
    plt.savefig(img_string + "reduced_DP_trapped.png", dpi = 500, format =
    ↪  "png",bbox_inches = 'tight')
    plt.close()

    fig,ax = plt.subplots()
    ### mean spacing ###
    ax.plot(he_core,integrals[:,2], color = "blue", linewidth = 1)

    ax.set_xlabel("core_he")
    ax.set_ylabel(r"$\Pi_{0,l}\sqrt{l(l+1)}$")
    ax.set_xlim(0,1)
    ax.invert_xaxis()
    ax.legend(fontsize = "x-small")
    plt.savefig(img_string + "reduced_DP.png", dpi = 500, format = "png",bbox_inches =
    ↪  'tight')
    plt.close()
```

Finally, we can compute the observed trapped mode spacings (that is, the experimental ones, as opposed to theoretical). We use TRAPPED_MODE_FINDER() and EKIN_TRAPPED() to compute the mean asymptotic period spacing and the trapped mode period spacing at the same time.

```
def trapped_mode_finder(pulse_data,mode_spacing):

    no_nan_spacing = mode_spacing[np.isnan(mode_spacing) == False]*np.sqrt(2)
    full_spacing = np.sum(no_nan_spacing)/no_nan_spacing.shape[0]
    #computes the observed mean spacing

    mode_spacing = mode_spacing[20:]*np.sqrt(2) #we skip the low order modes
    periods = pulse_data[20:,2]*np.sqrt(2) #same thing here
    ekin = pulse_data[20:,4]
```

```python

    length = mode_spacing.shape[0]
    trapped_mode = np.array([])

    average_spacing = np.nansum(mode_spacing)/length
    #above this average : automatically a normal mode
    #below : could be a trapped mode (giving if condition after)

    for i in range(2,length-1):
        if (mode_spacing[i] < average_spacing): #this condition there
            if (mode_spacing[i]) == min(mode_spacing[i-2:i+3]):
                #finds that there is a trapped mode in a range
                #one center mode + 2 nearest neighbours
                index = ekin_trapped(ekin,i)#finds which is actually trapped
                trapped_mode = np.append(trapped_mode,index)


    trapped_mode = trapped_mode.astype(int)
    periods = np.diff(periods[trapped_mode])

    periods = periods[periods < 5000] #accounts for occasional too little counts
    periods = periods[periods > 1500] #accounts for occasional too much counts

    if periods.size == 0:
        periods = np.array([np.nan])
        #in case we do not find 2 or more trapped modes (for plotting)

    trapped_spacing = np.sum(periods)/periods.shape[0]
    #and here is the spacing for trapped modes in SC zone

    obs_spacings = np.array([trapped_spacing,full_spacing])
    print(obs_spacings)
    return(obs_spacings)
```

```python
def ekin_trapped(ekin,i):

    #finds the "real" index of our trapped mode
    maximum = max(ekin[i-1:i+2])
    index = np.where(ekin == maximum)
    return(index[0])
```

We plot against observed versus theoretical spacings in PLOT_TH_OBS(), a function that also plot the associated relative spacing between experimentation and theory.

```python
def plot_th_obs(obs_spacings,th_spacings,he_core,img_string):

    fig,ax = plt.subplots()
     ### trapped spacing ###
    ax.plot(he_core,obs_spacings[:,0],label = "Model", color = "red", linewidth = 1)
    ax.plot(he_core,th_spacings[:,0],label = "Theory, overshooting added", color = "green", linewidth = 1)
    ax.plot(he_core,th_spacings[:,1],label = "Theory, no overshooting", color = "blue", linewidth = 1)

    ax.set_xlabel("core_he")
    ax.set_ylabel(r"$\Pi_{T,l}\sqrt{l(l+1)}$")
    ax.set_xlim(0.2,0.6)
    ax.set_ylim(2000,6000)
    ax.invert_xaxis()
    ax.legend(fontsize = "x-small")
    plt.savefig(img_string + "obs_th_trapped.png", dpi = 500, format = "png",bbox_inches = 'tight')
    plt.close()

    fig,ax = plt.subplots()
    ### classic spacing ###
    ax.plot(he_core,th_spacings[:,2], color = "blue", label= "Theory", linewidth = 1)
    ax.plot(he_core,obs_spacings[:,1], color = "red", label = "Model", linewidth = 1)

    ax.set_xlabel("core_he")
    ax.set_ylabel(r"$\Pi_{0,l}\sqrt{l(l+1)}$")
    ax.set_xlim(0.2,0.6)
    ax.set_ylim(330,350)
    ax.invert_xaxis()
    ax.legend(fontsize = "x-small")
    plt.savefig(img_string + "obs_th.png", dpi = 500, format = "png",bbox_inches = 'tight')
    plt.close()

    fig,ax = plt.subplots()
    ### trapped spacing relative spacing ###
    error_trapped = (np.absolute(obs_spacings[:,0] - th_spacings[:,1])/obs_spacings[:,0])*100
    ax.plot(he_core,error_trapped, color = "blue", linewidth = 1)
    ax.set_xlabel("core_he")
    ax.set_ylabel(r"$\Pi_{T,l}\sqrt{l(l+1)}$" + " relative spacing")
    ax.set_xlim(0.2,0.6)
    ax.set_ylim(0,25)
    ax.invert_xaxis()
    plt.savefig(img_string + "error_trapped.png", dpi = 500, format = "png",bbox_inches = 'tight')
    plt.close()

    fig,ax = plt.subplots()
    ### mean spacing relative spacing ###
    error_mean = (np.absolute(obs_spacings[:,1] - th_spacings[:,2])/obs_spacings[:,1])*100
    ax.plot(he_core,error_mean, color = "blue", linewidth = 1)
    ax.set_xlabel("core_he")
    ax.set_ylabel(r"$\Pi_{0,l}\sqrt{l(l+1)}$" + " relative spacing")
    ax.set_xlim(0.2,0.6)
    ax.invert_xaxis()
    plt.savefig(img_string + "error_mean.png", dpi = 500, format = "png",bbox_inches = 'tight')
    plt.close()
```