BETTER ESTIMATING THE CORE MASS OF JUPITER

BY

SHAMUS R. TOBIN

A thesis submitted to the

Department of Physics

Mount Allison University

in partial fulfillment of the requirements for the

Bachelor of Science degree

with Honours

April, 2022

ABSTRACT

The mass of Jupiter's core is currently estimated to be between 5 and $20E_{\oplus}$, which leaves the core mass as a significant unknown quantity when examining the solar system. The two primary theories that address the formation of giant planets such as Jupiter, are Core Accretion (CA) and Disc Instability (DI). CA predicts that the core mass should be between $15 - 20E_{\oplus}$, and DI predicts that the core mass should be between $5 - 10E_{\oplus}$. Since both predictions overlap with the current estimates, neither method can be confirmed as the one used to form Jupiter. Through Modules of Experiments in Stellar Astrophysics (MESA), a series of models of Jupiter was created with changes in their core mass in $0.5E_{\oplus}$ intervals within the current estimate. Using GYRE, a stellar oscillation code, the model's p-modes and g-modes were analyzed. Upon examination of p-modes, it was found that changes in the core had no significant effect on their oscillations per theoretical predictions. Several models exhibited irregular behaviours in their g-modes, showing higher than average percentage of frequencies with positive mode work and normalized growth rates (η). The physical characteristics of these irregular models were examined and found that the sound speed varied in regions less than 0.40 Jupiter radii. Additionally, period spacings of these models found the spacings to be regular and dips in the spacing, likely being caused by changes in the local molecular weight. Variations in regions of g-mode oscillations with core mass were found between 0.2 and 0.5 Jupiter radii, which further indicates that the core mass is affecting the g-mode oscillations and could be causing physical changes in the models. This research was not able to fully examine the exact differences in the interior structures that might be causing these irregular behaviours to occur but should be examined in the future. The results of

this research remain inconclusive in determining better estimations for the core mass of Jupiter but can act as a foundation for future researchers to continue.

ACKNOWLEDGMENTS

I would first like to thank Dr. Catherine Lovekin for supervising the preparation of this thesis and guiding the research that went into it. Without her amazing and unwavering support, this thesis would be but a shell of what it is now.

Thank you to Dr. Robert Sorba for always being there to hear my troubles over the last year and always being understanding and offering help to push me through.

A special thanks Abigail Flann for her absolute belief in me throughout the whole preparation of this thesis and its research, and for acting as my crutch for when the challenges of this thesis seemed overwhelming. I could not have done it without her.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Overview of Jupiter

Jupiter is the fifth and largest planet in our solar system and holds a mass of 1.898×10^{27}kg or $317.83 M_{\oplus}$. The planet is known as a Gas Giant Planet due to its thick atmosphere, composed mainly of Hydrogen (H) and Helium (He), that contributes to the planets relatively large volumetric mean radius of 6.9911×10^{9}cm, equivalent to $10.973 R_{\oplus}$ [18].

The structure of Jupiter can be divided into 4 main layers consisting of gaseous elements making up its atmosphere, liquid hydrogen directly below the atmosphere, then metallic hydrogen, and then finally its solid, rocky core at the center of the planet. Jupiter is dominated by hydrogen, which accounts for at least 70% of the planet's mass regardless of its form. The Galileo probe mission measured the composition of Jupiter to be approximately 24% Helium in its upper atmosphere, contradicting predictions that it should be consistent 28%, the composition of the original solar nebulae. This discrepancy is thought to be the result of some He, which had previously been dissolved into fluid hydrogen, has precipitated out of the solution and proceeded to sink down towards the planet's core while the planet has cooled over its lifespan [9].

Gaseous hydrogen dominates the upper layers of Jupiter until a depth of about 1000km, where a layer of liquid hydrogen extends to a depth of 20,000km, formed as a result of the high pressures produced by the upper layer's of Jupiter's atmosphere [8].

Then, at approximately a quarter of the distance down towards Jupiter's center, a layer of metallic hydrogen begins when molecular hydrogen transitions to a different set of electrical properties where a hydrogen atom's electron is no longer bound to their nuclei. This property gives the hydrogen the conductivity of a metal [9].

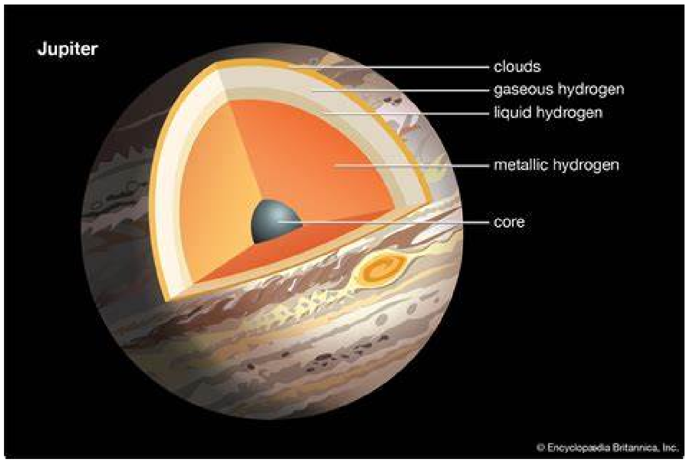

Figure 1.1: Structure of Jupiter displaying layers such as its core, metallic hydrogen, liquid hydrogen, and gaseous layers[5].

The thick gaseous atmosphere of Jupiter prevents visual observations of its core. This has resulted in the size and composition of the core remaining, largely, unknown quantities. There exists no direct way of observing the core via traditional ground-based telescopes or space based probes.

1.2 Core Accretion

There currently exist two prevailing theories for the method of Jovian planetary formation, the first being core accretion. This method is currently the most widely accepted theory for how giant planets are formed.

During a star's pre-main sequence where the star continues to contract and increase in rotation, the material left over from the parent nebulae, consisting mainly of silica dust grains and molecular hydrogen, becomes compressed into a flat disc around the young star. These discs are known as circumstellar discs or protoplanetary discs, as they are widely considered to be the origin of most planets. Initially, the dust grains that compose these discs are usually only millimeters in size or smaller. By collisional coagulation, these grains begin to clump together until they become sufficiently massive to gravitationally attract additional grains. Once a mass reaches this point, it is called a planetesimal [11].

As time progresses these planetesimals continue to attract left over grains which contribute further to its mass, continuously carving out an orbit within the protoplanetary disk. Eventually, planetesimals begin to grow in mass through the result of thousands of low-velocity collisions with other planetesimals to form protoplanetary cores. The protoplanetary core will then have sufficient mass to accrete gaseous materials from the protoplanetary disk to form an atmosphere. For Giant planets, such as Jupiter, the protoplanetary cores will have to reach a mass of at least several Earth masses to generate a sufficient gravitational force to capture the amount gaseous materials observed in the atmospheres of such planets [11].

Concerns regarding core accretion arise when attempting to use the theory to explain Jupiter's formation. Most notably, the time it would take to form a Jupiter sized planet via core accretion is approximately 10 Myrs. Recent simulations predict that most of the mass of a circumstellar disk will have dissipated or used to form other planets within 4-6 Myrs [7]. However, the process of the formation of planetesimals is not yet fully understood which may leave room for Jupiter to still have formed this way. Additionally, we most commonly see exoplanets like Jupiter at orbital distances around the host stars that would favor core accretion as opposed to the next theory. Given this uncertainty, another formation theory for giant planets should be explored that may be able to explain, with more certainty, the formation of Jupiter. Core

Accretion theory predicts that the core mass of a giant planet the size of Jupiter to be between 15-20 Earth Masses.

1.3 Disk Instability

The second prevailing theory in giant planet formation is known as Disk Instability. This theory suggest that giant planets form as result of localized gravitational instabilities in a protoplanetary disk. Regions in a protoplanetary disks can become unstable if they become sufficiently cool and dense Typically these regions are located at larger orbits where there is less heating from the host star. The condition for gravitational instability to occur is determined by the value of the Toomre Parameter Q [2]. The Toomre parameter Q is defined as,

$$Q = \frac{\kappa\sigma_0}{\pi G\Sigma} \tag{1.1}$$

Where κ is the epicyclic frequency or the frequency of radial motions resulting form small perturbations in the orbit of a particle, σ_0 is the sound speed, and Σ is the gas surface density. When $Q < 1$, the protoplanetary disk will be unstable [1].

Such an instability will result in the creation of a gravitationally bound subcondensation region (clump). Should the cooling time in this region of the protoplanetary disk be shorter than the dynamical time, then the "clump" will contract into a protoplanetary core from the gas of the circustellar disc, eventually evolving into a giant planet [2].

As with core accretion theory, concerns arise when attempting to apply this theory to explain Jupiter's formation. The main concern is that given Jupiter's current orbit at approximately 5.2AU, a clump at that orbital distance would not hold enough mass to form a Jupiter sized planet.

However, a possible solution to this concern is the possibility that Jupiter originally began to form further out of in the circumstellar disk where more massive clumps are able to form, and migrated Sun wards to its current orbit. This process would have likely taken place during the early evolution of the disc where the disc's structure was irregular and was still accreting mass from the envelope. Therefore, local regions that increase in surface density and cool sufficiently fast enough can become unstable and fragment into a protoplanetary core large enough to form a Jupiter sized planet [16]. Disk Instability theory predicts that the core mass of the resulting giant planet the size of Jupiter will be between $5 - 10E_{\oplus}$.

1.4 Importance of the Core Mass

The mass of Jupiter's core is an important piece in examining the two suspected formation methods for the giant planet. Core accretion predicts the core mass to be within the range of $15 - 20E_{\oplus}$ while Disk Instability predicts a core mass in the range of $5 - 10E_{\oplus}$. NASA's Juno mission conducted the first experimental study of Jupiter's core, and was able to determine a core exists. This disproved the theory that Jupiter had no core at all, and formed as a result of the contraction and cooling of a large cloud of gas and dust that surrounded the Sun after its formation [19]. This discovery has allowed for constraints to be placed on the core mass of being between $5 - 20E_{\oplus}$.

Better estimating the core mass of Jupiter will help to confirm which of the formation methods, described above, Jupiter underwent in the early solar system. Determining the formation method that Jupiter undertook is important because it help to determine how the other planets of the solar system formed, as they all came from the same circustellar disk. Determining the core mass of Jupiter will also help in determining other physical characteristics about the core of the planet, provide insight into the planetary formation in our early solar system as well as other Jupiter-like exoplanets.

Although Jupiter's thick atmosphere prevents direct measurements of its core's mass, there exists a method of theoretically determining the core mass of Jupiter indirectly; examining the Giant Planet's pulsation frequencies outlined in Section 1.5. Since it is currently not possible to directly measure the pulsation frequencies of Jupiter at this time, computational models of the Giant Planet are used to collect simulated data to form a theoretical determination of the nature of the pulsation frequencies present inside Jupiter's atmosphere for this project. These results can be used to inform future missions to focus for these pulsation signals.

1.5 Stellar Oscillations

Important characteristics of stars and giant planets are their pressure modes (p-modes) and gravity modes (g-modes). These modes are oscillations that occur in the interiors of stars and atmospheres of giant planets and are influenced by the physical characteristics of the star or giant planet.

The p-modes occur near the surface of a star or giant planet and are determined primarily from variations in the sound speed defined by the Lamb frequency, with pressure as a restoring force, thus denoting these oscillations as pressure modes. The Lamb frequency governs the p-mode oscillations in the upper layers of the star and atmosphere of giant planets, it is defined by

$$S_l^2 = \frac{l(l+1)c_s^2}{r^2} \tag{1.2}$$

where ℓ is the angular degree, c_s is the sound speed, and r is the radial order.

The g-mode oscillations are trapped gravity waves in the radiative interior of stars and giant planets and have higher displacement amplitude variations than p-modes. These modes are dominated by the variations in the Brunt-Vaisala frequency (N) [6].

The Brunt Vaisala frequency governs the g-mode oscillations in the radiative interior and is defined by,

$$N^2 = g(\frac{1}{\Gamma_1 P}\frac{dP}{dr} - \frac{1}{\rho}\frac{d\rho}{dr}) \tag{1.3}$$

where g is the local acceleration due to gravity, and P is the pressure. Γ is the adiabatic index or the heat capacity ratio, and is defined as the ratio of heat capacity at a constant pressure to heat capacity at a constant volume.

Since the Lamb and Brunt-Vaisala frequencies govern the p-modes and g-modes respectively, plotting the curve of their values with respect to radius, the region where the modes are expected to oscillate can be determined. The region that lies above both frequencies is where p-modes are expected, between the two frequencies is an evanescent zone, and the region below both frequency curves is where g-modes are expected. When the evanescent region is small modes are able to cross the evanescent zone, and mixed modes can occur.

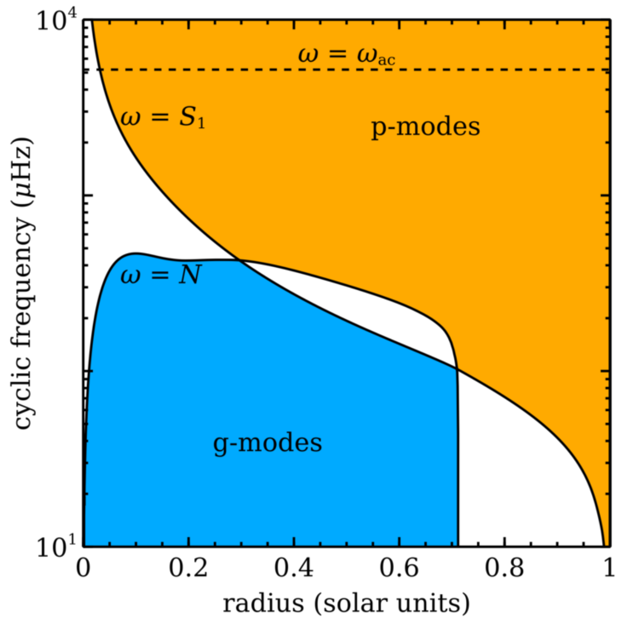

Figure 1.2: A propagation diagram for a solar model showing where g-modes oscillate shown in blue and where p-modes oscillate shown in orange; outlined by the Brunt-Vaisala & Lamb Frequencies, respectively [17].

For the p-mode and g-mode oscillations, eigenfrequencies and eigenfunctions can be calculated and the pulsational eigenfrequencies can be explored for their dependence on physical parameters such as total mass, core mass, radius, and the heavy metals fraction. †

The g-modes, in particular, are influenced greatly by the mass of the core due to their oscillation region existing deep inside the interior closer to the core. Thus, changes in these deep interior regions by change in the core mass are likely to cause changes in the behaviour of the g-modes. When examining the eigenfrequencies and eigenfunctions of the g-mode oscillations it is expected that they will vary with change in the core mass. Therefore, changes in the behaviour of the g-modes provide a

†The heavy metals fraction is the fraction of the mass that belongs to elements that are not either hydrogen or helium.

means of determining the core mass of a star or giant planet that would otherwise be obscured.

Chapter 2

MODELS

2.1 MESA

The models analysed in this research were created using Modules for Experimental Stellar Astrophysics (MESA). MESA has been developed over the last decade with the goal of providing an open source tool to perform stellar evolution calculations in areas such as asteroseismology, stellar hydrodynamics, stellar activity and more via one-dimensional stellar evolution code [10].

MESA develops a desired model based off of parameters defined in a file called an "inlist". An inlist contains specifications for the controls and options that are to be applied during the evolution, as well specifications for the input model or parameters for an initial to be created by should one need to be generated.

This process generates a stellar model at various points throughout the total given time of evolution, either when a phase of its evolution is completed, or after sufficient time has elapsed over the course of a phase. By editing the "inlist" files, MESA can be instructed to calculate various characteristics of a model throughout its evolution. These qualities are recorded in logs or profiles. A log records the models evolutionary properties over time such as age, current radius and mass, and a variety of other quantities. A profile records a model's properties at a specified timestep at each zone of the model from the center to the surface [10].

MESA provides numerous example cases for a variety of different kinds of models that a user may use to act as a basis when starting a new project in the test suite library. This research used one such example case called "make planets" to base the modelling procedure on.

2.1.1 Giant Planets

MESA was originally developed to model stars. The developers behind MESA have more recently developed capabilities in the software to model Giant Planets, allowing for detailed modelling of such planets to be done in similar fashion to stellar models within the same MESA interface and software.

Giant planets possess similar elemental composition to their host star(s), coming from the same nebular cloud. The elemental composition of stellar bodies, such as giant planets and stars, is characterized by X, Y, and Z fractions, the fraction of hydrogen, helium, and heavy metals, respectively, that compose a stellar body.

These giant planets have failed to become stars themselves due to the lack of sufficient temperature and pressures needed for their composite hydrogen atoms to fuse together. The lack of nuclear reactions allows giant planets to form thick gaseous atmospheres of primarily hydrogen and helium around a core, likely composed of rock, ice and heavy metals.

The absence of fusion occurring inside Giant Planets is one of the key differences that separates giant planets from stars. Distinguishing between a nuclear reactive core and one that is not is critical when creating a giant planet model in MESA. Therefore, MESA requires that an inert core, one that has no nuclear reactions taking place, be defined with the density of the core.

2.1.2 Constructing Models

In this research, the MESA test suite "make planets" was used as a base to model Giant Planets resembling Jupiter. This test suite consisted of three inlist files each corresponding to a specific phase of the model, which would be called in sequence.

The first inlist file that would be called, “inlist create”, instructs MESA to make an initial model. For this research, the initial radius was set higher than a Jupiter radii at $7.922 \times 10^9 cm$ due to significant contraction that takes place as the model evolves, and the initial mass was set $1.89819 \times 10^{30} g$ or 1 Jupiter mass which remained constant throughout evolution. Additionally, the initial Y and Z fractions were set to be 0.24 and 0.02, respectively, with the initial hydrogen fraction being automatically set to the remaining difference.

The second inlist file called, “inlist core”, instructs MESA to insert a core into the initial model created in “inlist create”. For this research, the core was defined to have an average density of $15 g/cm^3$ and a small average total energy generation rate of 1.0e-10 erg/g/s, essentially making the core inert. MESA also requires that “relax core” be set to true, which formally defines an inert core for the model. The core mass was varied between $5 - 20 M_\oplus$ in steps of $0.5 M_\oplus$.

The third and final inlist file called, “inlist evolve”, takes the now created initial model with the inserted core, and evolves the model. For this research, the main addition to this inlist included the defining of rotation parameters. Irradiation was neglected when constructing the models as the effects of it negligible for Jupiter.

Each inlist also contains instructions that define the starting age of the model and the maximum age that the model should follow the corresponding inlist instructions. For “inlist create” the initial age was set to 0 with a max age of 100 years, “inlist core” continued from 100 years until a max of 200 years, and “inlist evolve” continues from 200 years until a max age for the models of $4.503 \times 10^9 yrs$, the current age of Jupiter.

By default, MESA generates a non-rotating model unless specifically instructed to include rotation. For this project, rotating models were explored in order to examine the effects of including such a characteristic would have on the p-modes and g-modes.

Rotation was included by defining in “inlist evolve” and setting the angular velocity of the model to be equal to $1.75704 \times 10^{-4} rad/s$, the current average angular velocity of Jupiter.

2.2 GYRE

GYRE is a stellar oscillation code which, given a stellar model input, calculates the eigenfrequencies and eigenfunctions of the normal oscillation modes of the model [13]. The code is based on a “Magnus Multiple Shooting” (MMS) scheme developed by the GYRE development team. The MMS scheme solves boundary value problems (BVPs) that are defined by a system of linear, homogeneous, first order ordinary differential equations,

$$\frac{d\mathbf{y}}{dx} = \mathbf{A}(x)\mathbf{y} \tag{2.1}$$

defined on the interval $x^a \leq x \leq x^b$, along with boundary conditions and applied at each end of the interval [14],

$$\begin{aligned} \mathbf{B}^a \mathbf{y}(x^a) &= \mathbf{0} \\ \mathbf{B}^b \mathbf{y}(x^b) &= \mathbf{0} \end{aligned} \tag{2.2}$$

The data generated by GYRE can be used to compare theoretical data to observed oscillation frequencies of a star or planet. This allows for constraints to be established on the star’s or planet’s fundamental and internal properties.

2.2.1 Non-Adiabatic Calculations

While using GYRE to perform the necessary calculations, its non-adiabatic calculation functionality was used over its default adiabatic calculations.

Adiabatic calculations assume that when there is a change in a system that results in a transfer of energy into or out of the system exclusive in the form of work, no heat is transferred. For example, any process contained within a good thermal insulator is considered to be adiabatic as no heat is able to escape the system contained by the insulator. Non-adiabatic systems, however, are the opposite case of adiabatic systems where a change in the system that results in a transfer of energy into or out of the system involves the transfer of heat. Most naturally occurring systems, such as a gas giant, are examples of non-adiabatic systems.

Therefore, using non-adiabatic calculations provides a more accurate results when examining a non-adiabatic system that is Jupiter. These calculations provide linear growth or damping rates to accompany the usual frequencies of oscillation modes, which allow for predicting which mode(s) will be self-excited [13].

When including non-adiabatic effects, GYRE takes the linearized mass, momentum, and Poisson equations and augments them with the linearized heat and radiative diffusion equations, these equations can be found in the Linearized Equations section of the GYRE documents [**?**]. This augmentation causes the equations and their solutions to become complex quantities. The assumed time dependence for perturbations is $e^{(-i\sigma t)}$, indicating that the real and imaginary part of the eigenfrequency are related to the mode period Π and the growth e-folding time τ, respectively,

$$\Pi = \frac{2\pi}{\sigma_R} \tag{2.3}$$

$$\tau = \frac{1}{\sigma_I} \tag{2.4}$$

Where σ_R and σ_I represent the real and imaginary parts of the eigenfrequency, respectively.

GYRE solves the non-adiabatic equations by searching for the roots of a discriminant function $D(\omega)$. Since there is no way easy way to bracket roots in the complex plane, GYRE generates initial roots from adiabatic calculations and proceeds to refine them iteratively. GYRE provides three different methods for establishing these trial roots [13]. The method used here, the Adiabatic method, consists of adopting the real roots found via the adiabatic calculations as the initial trial roots for the non-adiabatic calculations. This method works well as long as the adiabatic and non-adiabatic roots lie close to one another in the complex plane, or when the oscillation modes are weakly non-adiabatic [13], where

$$|\frac{\sigma_I}{\sigma_R}| << 1 \tag{2.5}$$

In the case that the oscillations were strongly non-adiabatic, where $|\frac{\sigma_I}{\sigma_R}| \geq 1$, the Contour method should be used for obtaining the initial trial roots over the Adiabatic method. This is because if the imaginary part of the frequency increases in magnitude, the real part of the frequency typically shifts away from the adiabatic frequency. Consequently, when adiabatic roots are used for initial trial frequencies, the root solver could converge to non-adiabatic roots of neighbouring modes, missing other modes in the process [4]. This method was tested initially along with the Adiabatic method and the results found negligible differences between the two, and the Adiabatic method was then chosen due to its substantially shorter computation time.

2.2.2 Including Rotation

By default, GYRE assumes a non-rotating star/planet when determining its differential equations and boundary conditions [13]. This assumption can cause calculations for p-modes by GYRE is be less accurate than a rotating star/planet, as p-modes oscillate close to the surface which is influenced by the rotational effects of the star/planet. These rotational effects causes the planet to be stretched outwards about its equator making the shape of the planet to be less than a perfect sphere. Rotational effects also create jet streams that separates Jupiter's belts and zones in its upper atmosphere. Including rotational effects in the GYRE calculations allows for a more accurate determination of p-modes, and typically does not affect g-mode oscillations as they act deeper within the star/planet insulated from rotational effects.

Solving the full equations for a rotating star/planet is difficult, and currently lies outside the capabilities of GYRE. However, GYRE does provide two modifications to its calculations that arise from rotation including Doppler Shift, and the Coriolis Force.

The lowest-order effect of rotation appears in the Doppler shift that begins when transforming between inertial and local co-rotating reference frame. To account for this effect in the equations for oscillations, all points of the inertial-frame frequency σ are replaced by the co-rotating frequency,

$$\sigma_c = \sigma - m\Omega \tag{2.6}$$

where Ω is the rotating angular frequency, and m is the azimuthal order of the mode [13].

The Coriolis Force gives rise to the higher order effects of rotation, appearing in the linearized momentum equation to account for the non-inertial nature of the co-rotating frame of reference. GYRE integrates an approximate treatment of the Coriolis force based on the *Traditional Approximation of Rotation* (TAR)[13].

2.3 The Models

By using parameters outlined in section 2.1.2 and building from the "make planets" MESA test suite, a series of models were created with the core mass being the only difference among the inputted parameters. The core mass was varied between $5 - 20M_{\oplus}$ in steps of $0.5M_{\oplus}$, resulting in 31 distinct models of a Jupiter-like planet. Ensuring that the models were created successfully without significant errors occurring, several key physical characteristics of the models can be plotted to determine that they follow the expected behaviours.

The first of these characteristics is the radius as a function of time. In figure 2.1, the models with lower core mass have a higher initial and final radius than those with a larger core mass. This trend is expected due to heavier cores creating greater gravitional force to further contract the planet's large atmosphere. Figure 2.1 also demonstrates consistent changes in each of the model's radius throughout time indicating no errors occurred in during their evolution.

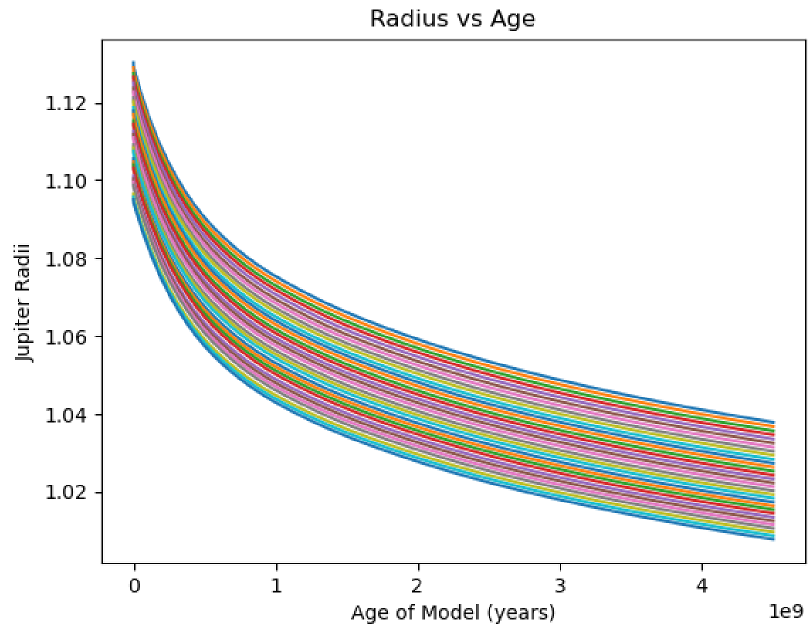

Figure 2.1: Radius as a function time of the each model created, where lower core mass models have a larger initial and final radius, and the larger core mass models having a smaller radius.

Tracking the density and pressures as a function of radius for each of the models shows whether these important and fundamental features are following the expected trends as well and that the model's interior do not exhibit unexpected behaviours. Examining pressure as a function of radius in Figure 2.2, it is shown that pressure decreases as radius increases up until the surface and increases closer to the core, as expected.

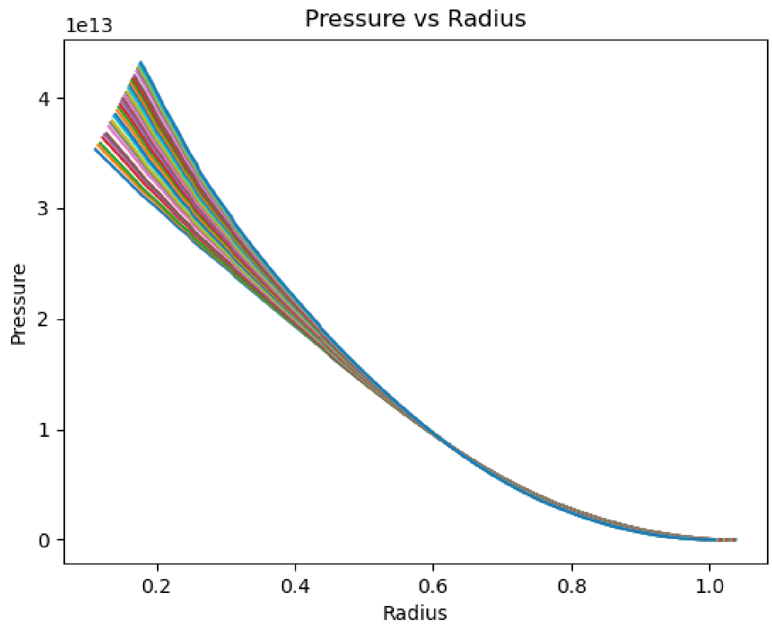

Figure 2.2: Pressure (g/cm^3s^2) as a function of Jupiter radii for all models, with those of a lower core mass having the lower pressure.

Likewise, tracking the density as a function of radius for each of the models in Figure 2.3 shows that the density increases closer to the core. This behaviour is expected as the increase in pressure shown in Figure 2.2, compresses the gases in the atmosphere as outlined in Section 1.1.

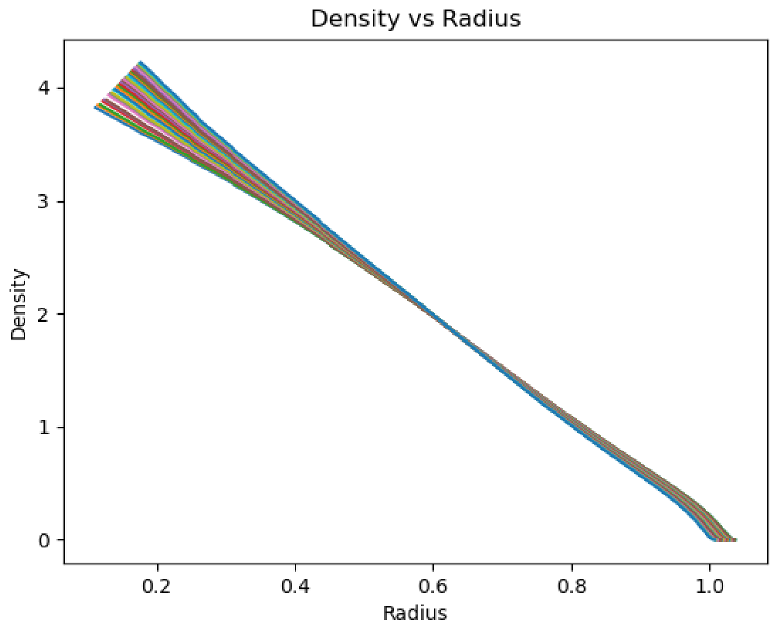

Figure 2.3: Density(cm/g^3) as a function of Jupiter radii for all models, with those of a lower core mass having the lower density.

From Figures 2.1 and 2.2, we see that the tracking along the radius axis stops at low radii at different points depending on the model. This is due to the radius of the core increasing with mass since density is set to be constant. The radius of the core is not an explicitly defined parameter when using "inlist core", but is determined based on the inputted values of the core mass and density. Since density is defined to be constant at $15.0g/cm^3$, the core radius increases as the inputted core mass increases.

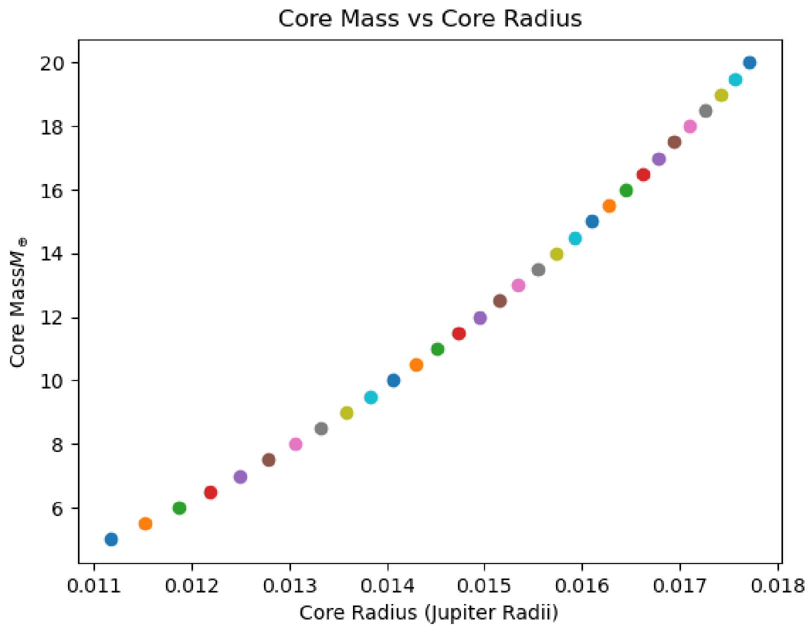

Figure 2.4: Core Mass as a function of core radius in Jupiter radii.

From all of these physical parameters, the models created by MESA are accurate to what is expected.

Chapter 3

RESULTS

Following the creation of the MESA models for a Jupiter-like planet with a core mass varying between $5-20E_{\oplus}$ in $0.5E_{\oplus}$ intervals, the models were inputted through GYRE to preform non-adiabatic calculations in the frequency range of $0.002-0.01(cyc/day)$. This revealed a host of information regarding the p-modes and g-modes of the models.

3.1 Intriguing GYRE findings

Whilst examining the non-adiabatic g-mode results generated by GYRE for the series of models, several of these models stood out from the rest as they exhibited significantly different behaviour in their real frequencies. These models included those that were generated with a $10.5E_{\oplus}$, $12.0E_{\oplus}$ $15.5E_{\oplus}$, and $17.0.0E_{\oplus}$ core mass. These models presented significant differences from the rest in two distinct aspects relating to the behaviour of their real frequencies, their mode work (W) and their normalized growth rate ($\eta^{'}$). Where $\eta^{'}$ is defined as,

$$\eta^{'} = \frac{\eta}{|\eta|} \tag{3.1}$$

η represents the fractional change in kinetic energy over period, otherwise known as the normal growth rate. When $\eta^{'} > 0$ the the corresponding frequencies of the p or g-modes are being driven, and when $\eta^{'} < 0$ the frequencies are being dissipated [12]. For frequencies that are being driven, they are at a higher likelihood of being detected then those that being dissipated.

The models in question, for g-modes, exhibited a higher than average percentage of frequencies, over 25%, that possessed positive mode work and positive normalized growth rate. The average percentage is 11%.

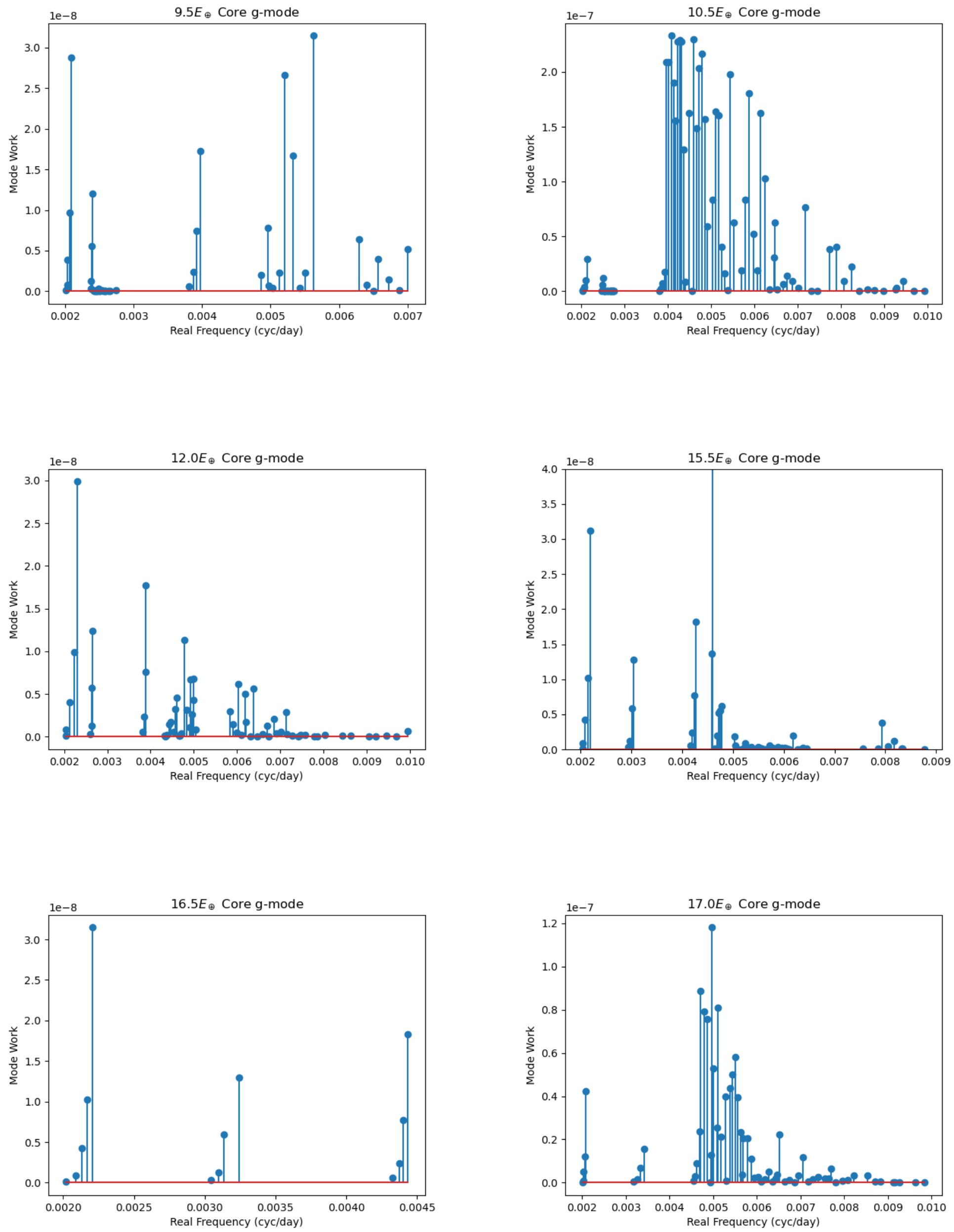

Figure 3.1: Mode Work vs. Real Frequencies (cyc/day) for g-modes of the models of interest along with $9.5E_{\oplus}$, $16.5E_{\oplus}$ core models as examples for the average results at 11% of frequencies with positive work.

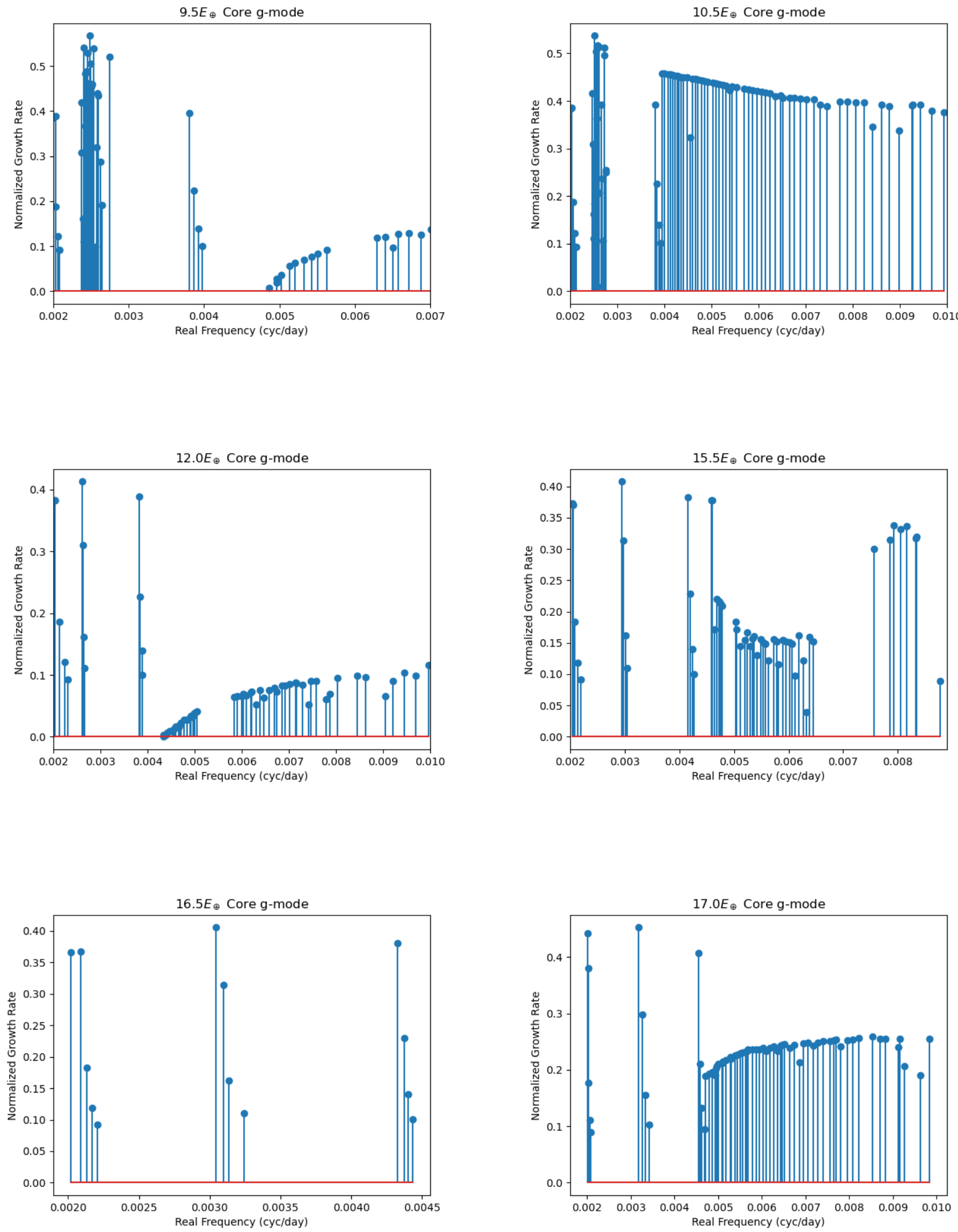

Figure 3.2: Normalized Growth Rate η' vs. Real Frequencies (cyc/day) for g-modes of the models of interest along with $9.5E_\oplus$, $16.5E_\oplus$ core models as examples for the average results at 11% of frequencies with positive normalized growth rate.

The $17.0E_\oplus$ core model had the highest percentage of positive mode work and η

frequencies at approximately 38% of the found frequencies possessing positive mode work and a positive normalized growth rate.

3.2 Examination of p-mode Results

As discussed in the previous chapter, p-modes occur near the regions near the surface of a giant planet and are influenced by the sound speed in those regions. Rotation is a characteristic of a giant planet that can influence these regions to affect the p-modes due to rotational forces causing a stretch along the equator due to centrifugal force. Core mass is not expected to be a characteristic that would influence the p-modes due to the relatively large distance from the core to the regions of oscillations of p-modes. However, it is important confirm that this is the case when examining the models.

Should it be the case that changes in the core mass are affecting the p-modes in our models, there are likely unknown behaviours taking place in the models that would require further investigation. As p-modes are expected to oscillate at higher frequencies than g-modes, a frequency range of $28.8 - 288.0 (cyc/day)$ was employed.

Figures 3.3 and 3.4 show the behaviours for the mode work and normalized growth rate of the frequencies found for p-modes for the same models examined for g-modes.

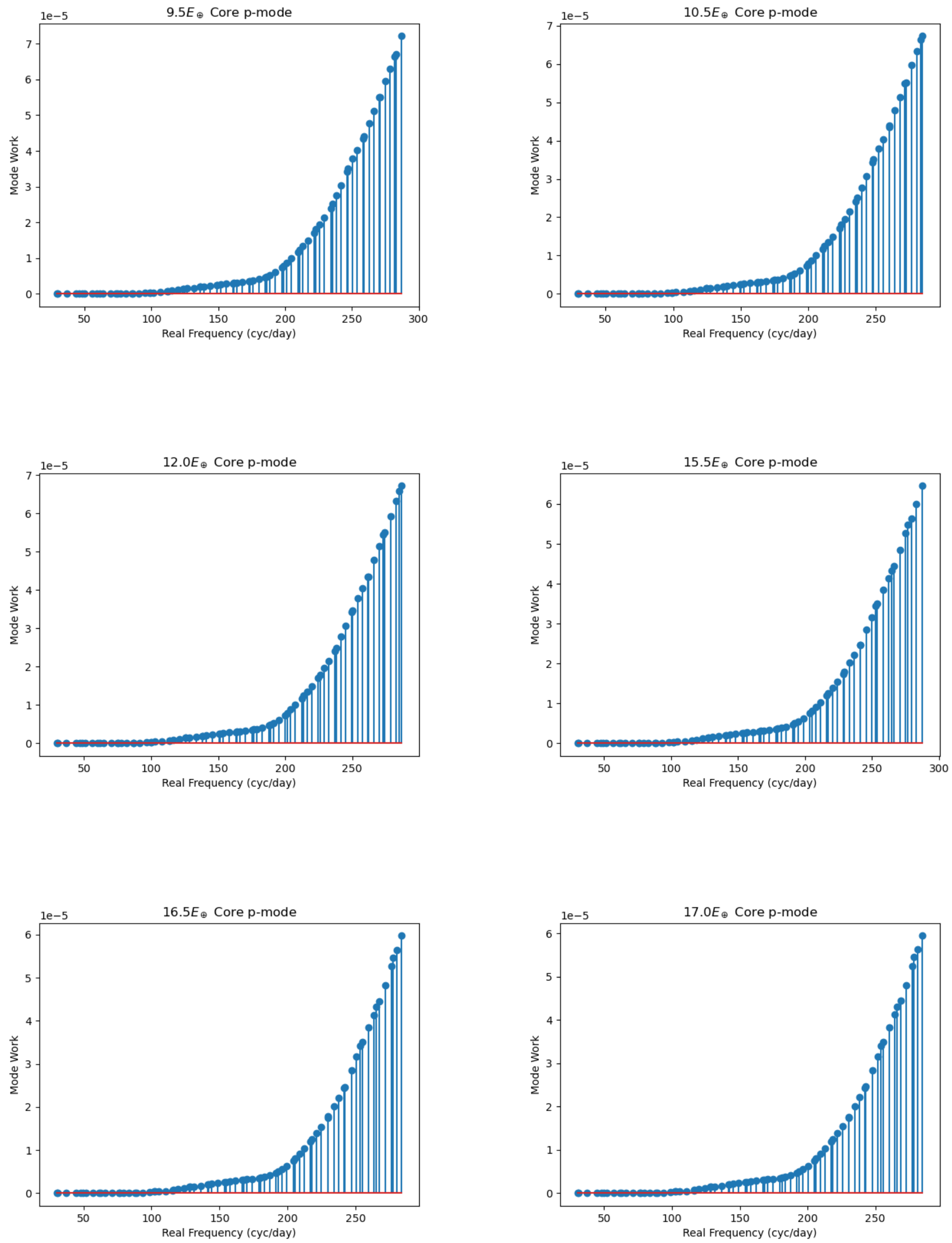

Figure 3.3: Mode Work vs. Real Frequencies (cyc/day) for p-modes of the models of interest along with $9.5E_{\oplus}$ and $16.5E_{\oplus}$ core models for comparison to g-mode results.

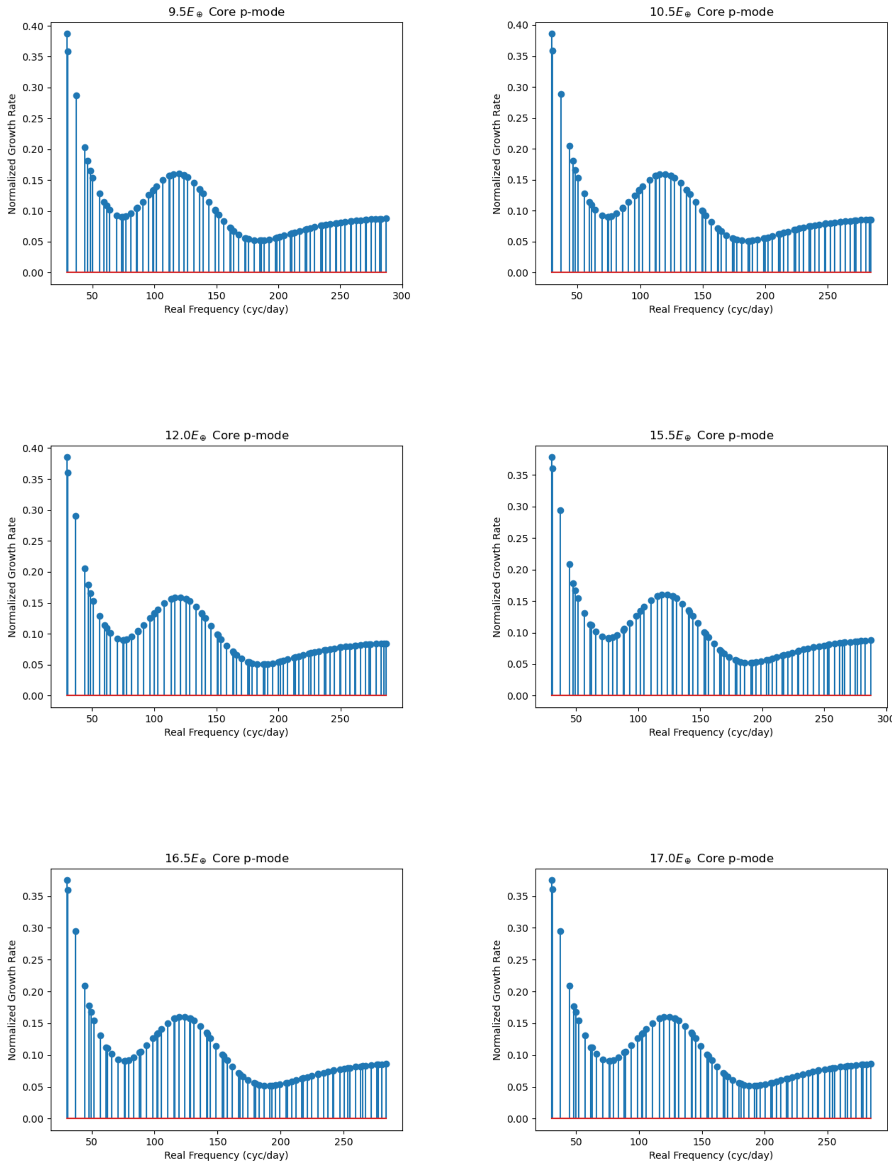

Figure 3.4: Normalized Growth Rate η' vs. Real Frequencies (cyc/day) for p-modes of the models of interest along with $9.5E_{\oplus}$ and $16.5E_{\oplus}$ core models for comparison to g-mode results.

Echelle diagrams, such as in Figure 3.5, are useful in demonstrating important information of the underlying structure of a star. The frequency separations between the frequencies of different harmonic degrees depend on the gradient of the sound speed in the deep interior of a star [3]. Figure 3.5 shows the echelle diagrams with the expected frequency separations between different harmonic degrees.

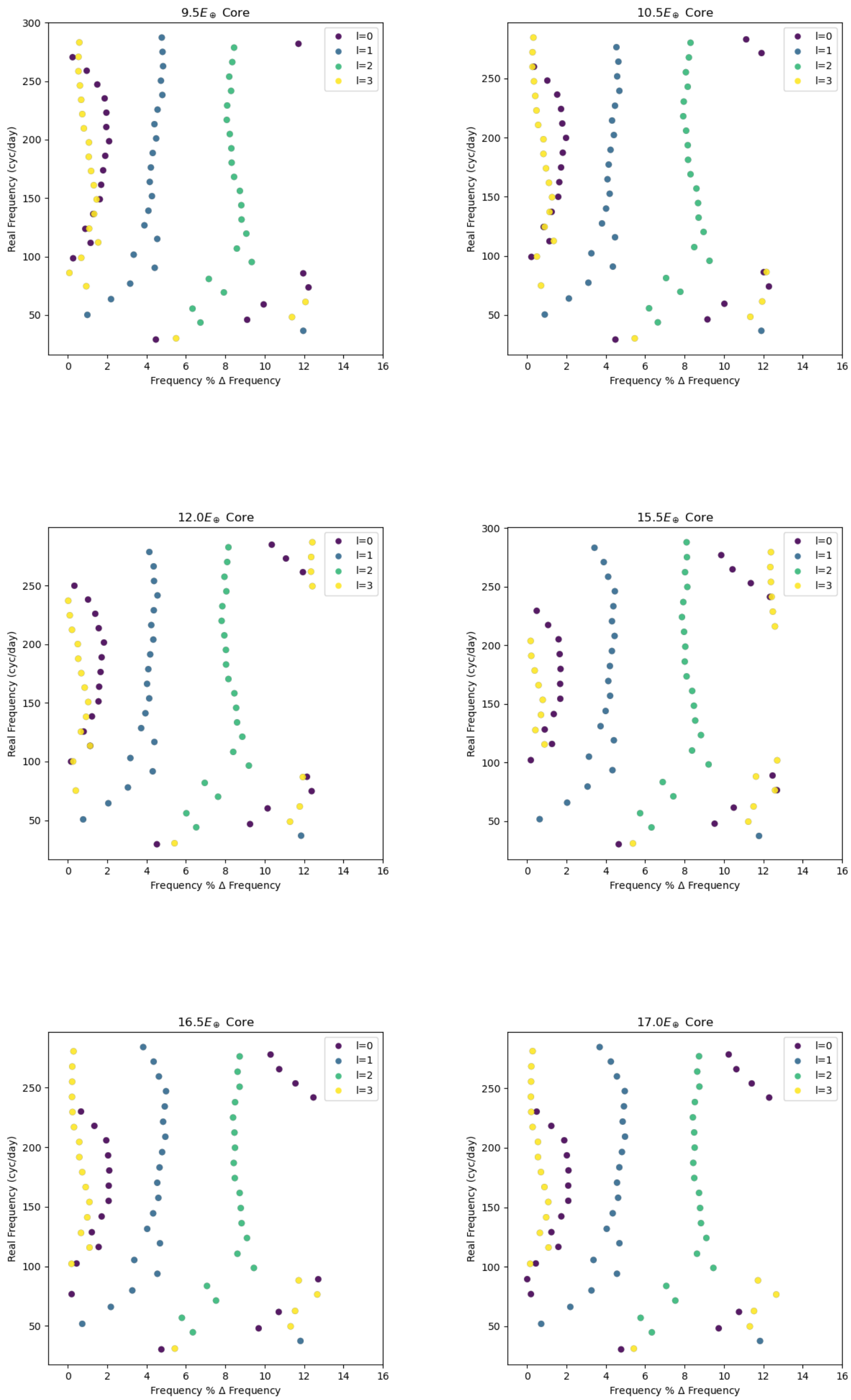

Figure 3.5: Echelle diagrams for l=0, 1, 2, and 3.

Examining Figures 3.3 and 3.4, no significant changes occur as core mass is changed, unlike with g-mode results. The consistency between the model's p-mode frequencies indicates that a change in the core mass is not affecting the p-modes, confirming expectations of p-modes and providing an additional check on the validity of the models. Therefore, these results confirm that g-modes should be examined instead p-modes when examining the affects of varying the core mass.

3.3 Examination of Model's Structure

These higher than average percentages of both positive mode work and normalized growth rate values prompted an investigation into what possible differences there may be between the structure these models and others that were generated. This allowed for a comparison to be made between the models in order to better understand which characteristics of the structure cause the differences in their mode work and normalized growth rate values.

Examining the fluctuations in the sound speed throughout the radius of the models found that the sound speed between the models remained consistent with each other except in the region between 0.13 and 0.40 Jupiter radii, where the sound speed saw deviations based on the core mass. At this region, the deviations found between 0.13 and 0.35 Jupiter radii change consistently with core mass and should more models be plotted, the noticeable gaps in Figure 3.7 would be filled. However, the changes in the sound speed shown in Figure 3.7 at 0.40 radii may be able to offset the frequencies enough to explain some differences between the models.

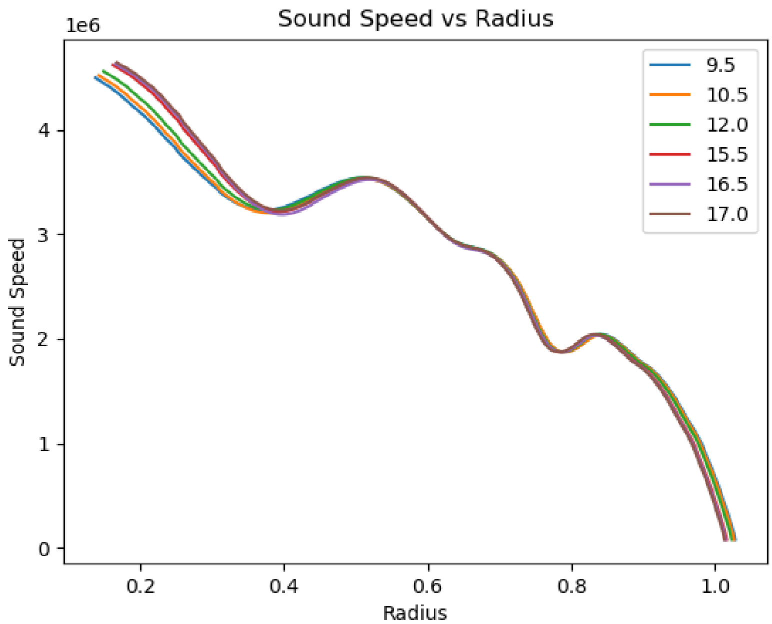

Figure 3.6: Sound speed (cm/s) as a function of Jupiter radii.

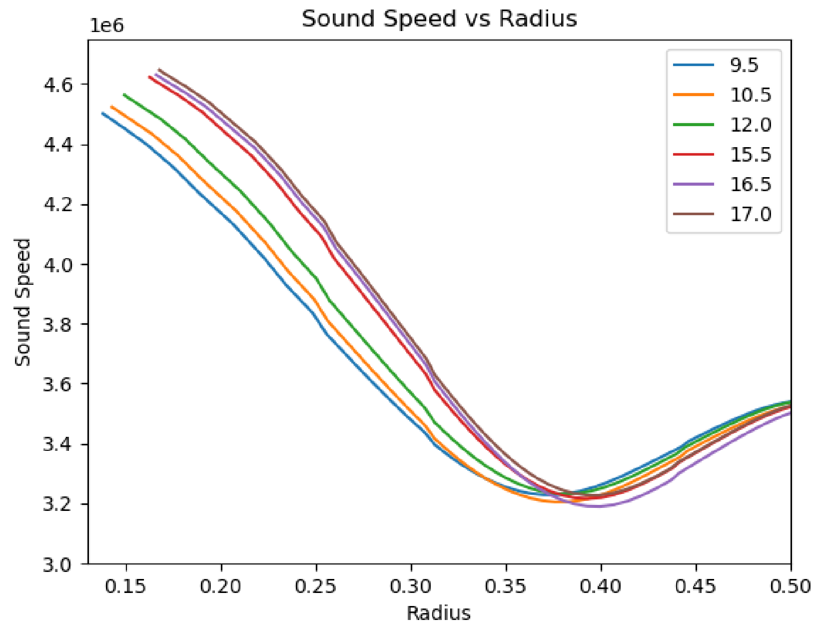

Figure 3.7: Sound speed (cm/s) as a function of Jupiter radii, focused on the region between 0.14 and 0.45 Jupiter radii where variations in the sound speed occur between the different models.

Viewing for difference in the convective velocity as a function of radius, there was found to be one region in which the convective velocity of the models varied slightly from each other that was within a radii that could be affected by the core. This region is between 0.16 and 0.22 Jupiter radii, shown in Figure 3.9. These deviations are also consistent with the deviations found for sound speed and the gaps between the models would be filled in should other models that were excluded be plotted. These deviations that are regular with core mass at low radii may be indication that core mass is having significant influence on the interior layers of the models.

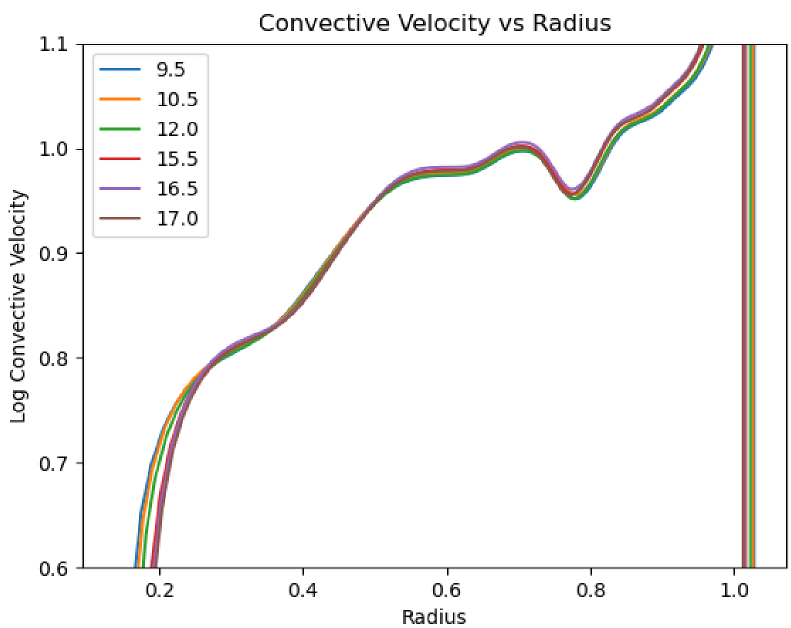

Figure 3.8: Convective Velocity (cm/s) as a function of Jupiter radii.

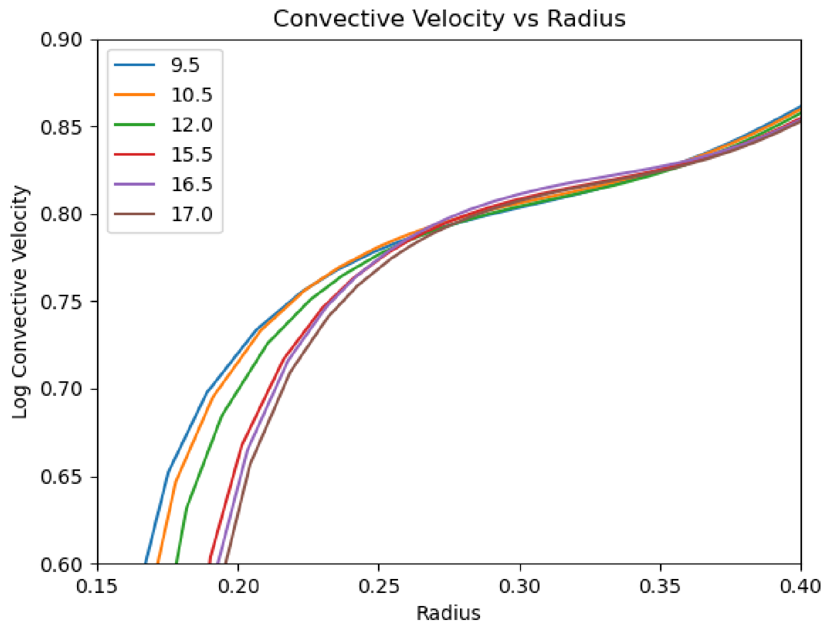

Figure 3.9: Convective Velocity (cm/s) as a function of Jupiter radii, the region of deviation at between 0.16 and 0.22 Jupiter radii.

Next, the Brunt-Vaisala and Lamb frequencies were calculated in order to determine variations in the regions of g-mode oscillation. From Figure 3.10, it is found that there are differences in the g-mode's region of oscillation particularly between the region of 0.2 to 0.5 Jupiter radii. This would indicate the core is contributing to changes in the structure of the models in that region, and also contributing to changes in the g-mode oscillations in the region.

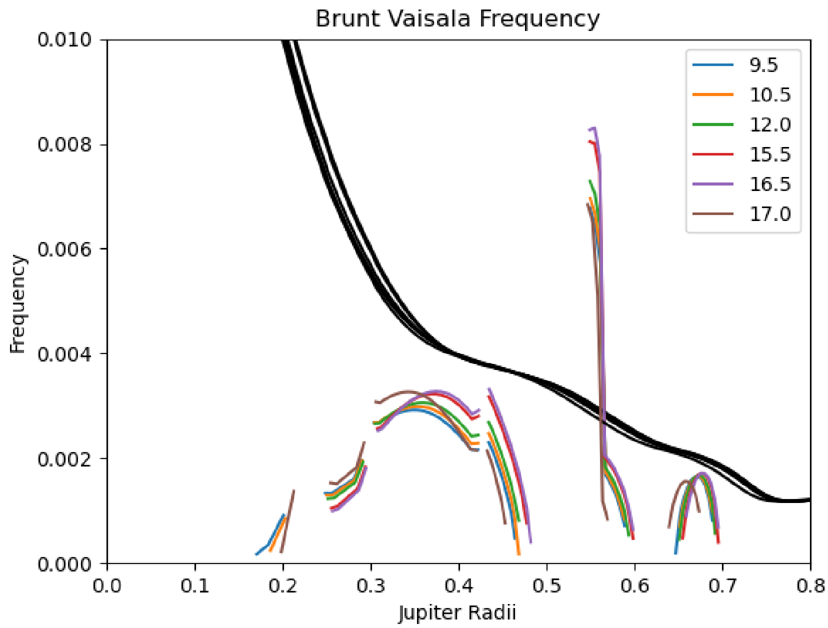

Figure 3.10: Brunt Vaisala Frequency (N) in colour and Lamb frequency (black) as a function of Jupiter Radii.

Examining the real frequency that has the maximum normalized growth rate for each model finds that for the majority of the models, these frequencies are relatively small being between $0.0020 - 0.0040 (cyc/day)$, at the low end of our frequency search range. Interestingly, the low end frequencies appear to increase linearly with core mass. Several frequencies of models such as $10.0E_{\oplus}$ were found to be close to 0.010, near the high end of our frequency search range.

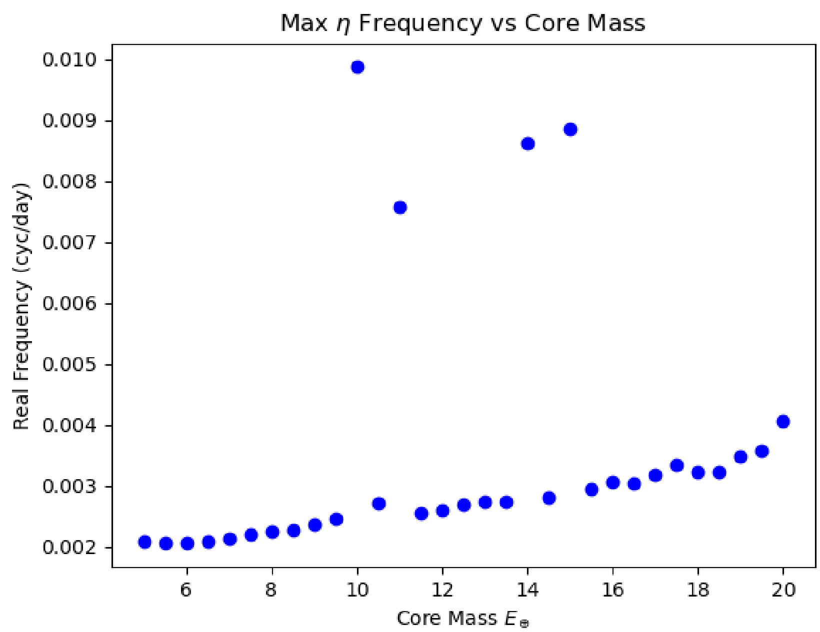

Figure 3.11: Real Frequencies (cyc/day) corresponding to the highest normalized growth rate as a function of core mass.

Period spacings of g-modes for when $n >> 1$, where n is the radial order, are expected to be equally spaced. Chemical gradients induced by the changing size of stellar core during a star's evolution, as well as rotation and mixing processes, can cause the uniform period spacings to characteristically deviate. It has been shown that changes in the local molecular weight leads to variations in the g-mode oscillation region, which translates into dips in the period spacing patterns. The amplitude of these dips indicates the steepness of the chemical gradient, and the periodicity indicates its location [15]. The models featured in Figure 3.12 have average $n > 600$, and shows the period spacings to be uniform and dips in the pattern which are most easily seen for $l = 1$ near ΔPeriod= 8 for the $10.5E_{\oplus}$ model.

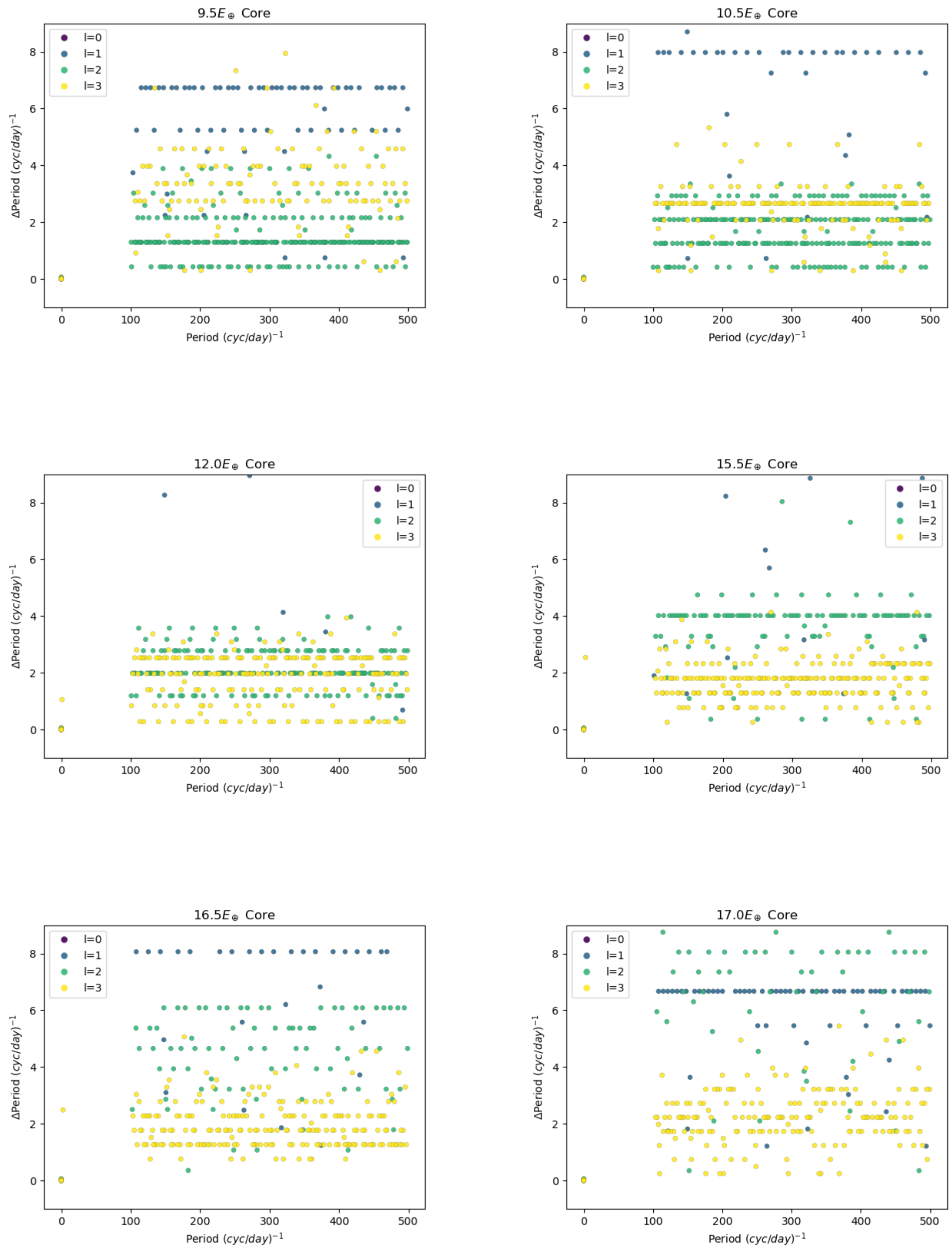

Figure 3.12: Absolute value of the difference in period as a function of period for frequencies for l=0, 1, 2, and 3.

Chapter 4

Conclusion

The goal of this research was to determine better estimations for the core mass of Jupiter using computational models. Models of Jupiter were generated using Modules for Experiments in Stellar Astrophysics (MESA) with varying core masses between $5 - 20E_{\oplus}$ in $0.5E_{\oplus}$ intervals. A stellar oscillation code, GYRE, was used to analyze the oscillations of p-modes and g-modes of the models for characteristics such as their frequency, harmonic degree (l), mode work, and normalized growth rate (η), which also allowed for period spacing and Echelle diagrams to be made. The range in frequency of oscillations that were analyzed with GYRE for p-modes was between $28.8 - 288.0(cyc/day)$ and g-modes was between $0.001 - 0.1(cyc/day)$. However, it was found that the initial frequency range for g-modes was too large and was resulting in missed frequencies and inconsistent results. Therefore, the frequency search range for g-modes was shortened to $0.002 - 0.02(cyc/day)$, which provided more consistent results. Values for various physical characteristics from the models such as sound speed, convective velocity, and the inputs to calculate the Brunt Vaisala frequency were collected.

Analysis of the p-modes found that their behaviour remained consistent between all models. This consistency is shown in the p-mode frequency's mode work and normalized growth rate values lacking any significant changes as the core mass is varied. The lack of significant changes confirms theoretical expectations that the p-modes oscillate in regions that are not affected by changes in the core. Examination of the g-modes found that their frequency behaviour was dynamic with changes in the core mass. Four models exhibited a greatly higher than average percentage, $> 25\%$, of frequencies with positive mode work and η. These models included those with a core mass of $10.5E_{\oplus}$, $12.0E_{\oplus}$, $15.5E_{\oplus}$, and $17.0E_{\oplus}$.

The irregular changes in the positive mode work frequency percentages go against expectations that these percentages should change gradually with core mass. Results found in the period spacings for g-modes and the Echelle diagrams for p-modes matched with predictions. The period spacings exhibit uniform spacing and showed dips in the period spacing patterns, likely caused by changes in the local molecular weight, i.e. changes from molecular to atomic hydrogen. Changes in the local molecular weight can lead to variations in the g-mode oscillation region.

Deviations in the sound speed found below 0.40 Jupiter radii should be investigated further to determine whether the causes of these changes could be contributing to the differences found in the g-modes. Additionally, the exact causes leading to non-uniform spacings and dips should be examined in greater detail to better detail the differences in the interior structure of these models and what effect are have on g-mode oscillations.

Changes in the regions of g-mode oscillations were found by calculating the Brunt Vaisala and Lamb frequencies. These frequencies showed that the region between 0.2 and 0.5 Jupiter radii vary significantly with changes in the core mass. These changes would further indicate that the core mass is affecting the interior structure of the models, and contributing to changes g-mode oscillations.

In conclusion, the data gathered from the models are inconclusive in determining better estimations for the core mass of Jupiter. However, the interior structure affecting the sound speed and period spacings should be examined in greater detail to understand what may be causing the differences in the oscillations of the several models with a higher than average percentage of frequencies with positive mode work and normalized growth rate η.

References

[1] Alessandro B. Romeo, Andreas Burkert, Oscar Agertz, A Toomre-like stability criterion for the clumpy and turbulent interstellar medium, Monthly Notices of the Royal Astronomical Society, Volume 407, Issue 2, September 2010, Pages 1223–1230, https://doi.org/10.1111/j.1365-2966.2010.16975.x

[2] Alibert Y., Helled R. (2015) Disk Instability, Model for Giant Planet Formation. In: Gargaud M. et al. (eds) Encyclopedia of Astrobiology. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-44185-55109

[3] *Asteroseismic data analysis ... - princeton university.* (n.d.). Retrieved March 20, 2022, from http://assets.press.princeton.edu/chapters/s11170.pdf

[4] Goldstein, J., &; Townsend, R. H. (2020). *The contour method: A new approach to finding modes of nonadiabatic Stellar Pulsations.* The Astrophysical Journal, 899(2), 116. https://doi.org/10.3847/1538-4357/aba748

[5] Gregerson, E. (n.d.). 5 mysteries of Jupiter that Juno might solve. Encyclopædia Britannica. Retrieved March 2, 2022, from https://www.britannica.com/list/5-mysteries-of-jupiter-that-juno-might-solve

[6] Jorgen Christensen-Dalsgaard , Lecture Notes on Stellar Oscillations, Institut for Fysik og Astronomi, Aarhus Universitet

[7] Manangath, P. P. (n.d.). Theory of Planet Formation. Talk slides. Retrieved March 4, 2022, from https://astrowin.uwo.ca/talks.html

[8] NASA. (n.d.). *A freaky fluid inside Jupiter?* NASA. Retrieved March 28, 2022, from https://science.nasa.gov/science-news/science-at-nasa/2011/09augjuno3:̃:text=(1)%20Jupiter%20is%20about%2070,a%20depth %20of%2020%2C000%20kilometers.

[9] Owen, T. C. (n.d.). The interior of Jupiter. Encyclopædia Britannica. Retrieved March 7, 2022, from https://www.britannica.com/place/Jupiter-planet/The-interior

[10] Paxton, B., Bildsten, L., Dotter, A., Herwig, F., Lesaffre, P., amp; Timmes, F. (2010). Modules for experiments in Stellar astrophysics (MESA). The Astrophysical Journal Supplement Series, 192(1), 3. https://doi.org/10.1088/0067-0049/192/1/3

[11] Raymond S. (2011) Core Accretion (Model for Giant Planet Formation). In: Gargaud M. et al. (eds) Encyclopedia of Astrobiology. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-11274-4353

[12] Stellingwerf, R. F. (1978). Helium ionization driving in beta Cephei stars. *The Astronomical Journal*, 83, 1184. https://doi.org/10.1086/112308

[13] Townsend, R. The GYRE Stellar Oscillation Code, https://gyre.readthedocs.io

[14] Townsend, R. H., &; Teitler, S. A. (2013). Gyre: An open-source stellar oscillation code based on a new Magnus multiple shooting scheme. Monthly Notices of the Royal Astronomical Society, 435(4), 3406–3418. https://doi.org/10.1093/mnras/stt1533

[15] Van Reeth, T., Tkachenko, A., Aerts, C., Pápics, P. I., Degroote, P., Debosscher, J., Zwintz, K., Bloemen, S., De Smedt, K., Hrudkova, M., Raskin, G., &; Van Winckel, H. (2015). Detecting non-uniform period spacings in the kepler photometry of γ doradus stars: Methodology and case studies. *Astronomy ; Astrophysics*, 574. https://doi.org/10.1051/0004-6361/201424585

[16] Vorobyov, E. I., &; Basu, S. (2010). *The burst mode of accretion and disk fragmentation in the early embedded stages of Star Formation.* The Astrophysical Journal, 719(2), 1896–1911. https://doi.org/10.1088/0004-637x/719/2/1896

[17] Warrickball. (2022, January 9). Asteroseismology. Wikipedia. Retrieved March 7, 2022, from https://en.wikipedia.org/wiki/Asteroseismology

[18] Williams, D.R., *Jupiter Fact Sheet*, National Aeronautics and Space Administration, Maryland, 2021.

[19] What's in Jupiter's core? Mission Juno. (n.d.). Retrieved March 29, 2022, from https://www.missionjuno.swri.edu/origin?show=hsoriginstorywhats-in-jupiters-core

Appendix A
AN APPENDIX

A.1 Python Code

A.1.1 Python Script to generate and analyze models in sequence

```
import numpy as np
import os
import mesa_reader as ms
import subprocess
from subprocess import call
import shutil
import matplotlib.pyplot as plt
import multiprocessing as mp

#Mass in MSun units
#mass = np.array([5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 8.5, 9.0, 9.5, 10.0, 10.5,
11.0, 11.5, 12.0, 12.5, 13.0, 13.5, 14.0, 14.5, 15.0, 15.5, 16.0, 16.5, 17.0,
17.5, 18.0, 18.5, 19.0, 19.5, 20.0])
mass = np.array([16.5])
#mass = np.array([9.0, 9.5, 13.5, 14.0, 16.5, 17.0])
#masses of sun and earth in kg
MEarth = 5.972e+24
MSun = 1.989e+30
core_mass = mass*MEarth/MSun

for m in core_mass:
    c = m*MSun/MEarth
    saveName = '"planet_core_M_'+str(c)+'.mod"'
    template = open('rn', 'r+')
    data = template.readlines()
    template.close()
    data[10] = 'do_one inlist_core '+saveName+'\n'
    rn = open('rn', 'r+')
    rn.writelines(data)
    rn.close()

    template = open('inlist_core', 'r+')
    data = template.readlines()
```

```
template.close()
data[12] = '      save_model_filename = '+saveName+'\n'
data[25] = '      new_core_mass = '+str(m)+'\n'
inlist = open('inlist_core', 'r+')
inlist.writelines(data)
inlist.close()

template = open('inlist_evolve', 'r+')
data = template.readlines()
template.close()
data[10] = '      saved_model_name = '+saveName+'\n'
inlist = open('inlist_evolve', 'r+')
inlist.writelines(data)
inlist.close()

#os.system("./mk")
os.system("./rn")

direc = 'LOGS/M_'+str(c)+'_EM_core'
if not os.path.exists(direc):
    os.makedirs(direc)
for fname in os.listdir('LOGS'):
        if os.path.isfile(os.path.join('LOGS', fname)):
            os.rename(os.path.join('LOGS', fname),
                      os.path.join(direc, fname))

direct = '/Users/stobin/desktop/Research/mesa-r15140/my_models
/jupiter_planets_g/gyre/work/LOGS/Adiabatic/M_'+str(c)+'_EM_core'
if not os.path.exists(direct):
    os.makedirs(direct)
for fname in os.listdir('/Users/stobin/desktop/Research/mesa-r15140/
my_models/jupiter_planets_g/gyre/work/LOGS'):
        if os.path.isfile(os.path.join('/Users/stobin/desktop/Research/
        mesa-r15140/my_models/jupiter_planets_g/gyre/work/LOGS', fname)):
            os.rename(os.path.join('/Users/stobin/desktop/Research/
            mesa-r15140/my_models/jupiter_planets_g/gyre/work/LOGS', fname),
                      os.path.join(direct, fname))

profile6 = '/Users/stobin/desktop/Research/mesa-r15140/my_models/
jupiter_planets_g/LOGS/M_'+str(c)+'_EM_core/profile6.data.GYRE'
if os.path.exists(profile6):
    direct = '/Users/stobin/desktop/Research/mesa-r15140/my_models/
    jupiter_planets_g/gyre/work/gyre.in'
```

```
    template = open(direct, 'r+')
    data = template.readlines()
    template.close()
    data[5] = "  file = '/Users/stobin/desktop/Research/mesa-r15140/my_models/
    jupiter_planets_g/LOGS/M_"+str(c)+"_EM_core/profile6.data.GYRE'\n"
    gyre = open(direct, 'r+')
    gyre.writelines(data)
    gyre.close()

    if not os.path.exists(profile6):
        direct = '/Users/stobin/desktop/Research/mesa-r15140/my_models/
        jupiter_planets_g/gyre/work/gyre.in'
        template = open(direct, 'r+')
        data = template.readlines()
        template.close()
        data[5] = "  file = '/Users/stobin/desktop/Research/mesa-r15140/my_mod
        jupiter_planets_g/LOGS/M_"+str(c)+"_EM_core/profile5.data.GYRE'\n"
        gyre = open(direct, 'r+')
        gyre.writelines(data)
        gyre.close()

direct = '/Users/stobin/desktop/Research/mesa-r15140/my_models/
jupiter_planets_g/gyre/work/gyre.in'
template = open(direct, 'r+')
data = template.readlines()
template.close()
data[86] = "  summary_file = '/Users/stobin/desktop/Research/mesa-r15140/my_mo
jupiter_planets_g/gyre/work/LOGS/Adiabatic/M_"+str(c)+"_EM_core/
summary_ad_file.txt'\n"
data[96] = "  summary_file = '/Users/stobin/desktop/Research/mesa-r15140/my_mo
jupiter_planets_g/gyre/work/LOGS/Adiabatic/M_"+str(c)+"_EM_core/
summary_nad_file.txt'\n"
gyre = open(direct, 'r+')
gyre.writelines(data)
gyre.close()

profile6 = '/Users/stobin/desktop/Research/mesa-r15140/my_models/
jupiter_planets_g/LOGS/M_'+str(c)+'_EM_core/profile6.data.GYRE'
if os.path.exists(profile6):
    star = ms.MesaData('/Users/stobin/desktop/Research/mesa-r15140/
    my_models/jupiter_planets_g/LOGS/M_'+str(c)+'_EM_core/profile6.data')
    radius = 10**(star.data('logR')[0])
    mass = star.data('mass')[0]
```

```
    scalefactor = np.sqrt(6.672e-8 * mass * 1.989e33 /
    (radius * 6.959e10)**3) / (2 * np.pi)
    direct = '/Users/stobin/desktop/Research/mesa-r15140/my_models/
    jupiter_planets_g/gyre/work/gyre.in'
    template = open(direct, 'r+')
    data = template.readlines()
    template.close()
    freqmin = 1.0e-3 / (scalefactor * 86400)
    freqmax = 0.1 / (scalefactor * 86400)
    data[63] = "  freq_min = "+str(freqmin)+"\n"
    data[64] = "  freq_max = "+str(freqmax)+"\n"
    #data[58] = "  freq_min = "+str(freqmin)+"\n"
    #data[59] = "  freq_max = "+str(freqmax)+"\n"
    gyre = open(direct, 'r+')
    gyre.writelines(data)
    gyre.close()
    if not os.path.exists(profile6):
        star = ms.MesaData('/Users/stobin/desktop/Research/mesa-r15140/
        my_models/jupiter_planets_g/LOGS/M_'+str(c)+'_EM_core/profile5.data')
        radius = 10**(star.data('logR')[0])
        mass = star.data('mass')[0]
        scalefactor = np.sqrt(6.672e-8 * mass * 1.989e33 /
        (radius * 6.959e10)**3) / (2 * np.pi)
        direct = '/Users/stobin/desktop/Research/mesa-r15140/my_models/
        jupiter_planets_g/gyre/work/gyre.in'
        template = open(direct, 'r+')
        data = template.readlines()
        template.close()
        freqmin = 1.0e-3 / (scalefactor * 86400)
        freqmax = 0.1 / (scalefactor * 86400)
        data[63] = "  freq_min = "+str(freqmin)+"\n"
        data[64] = "  freq_max = "+str(freqmax)+"\n"
        #data[58] = "  freq_min = "+str(freqmin)+"\n"
        #data[59] = "  freq_max = "+str(freqmax)+"\n"
        gyre = open(direct, 'r+')
        gyre.writelines(data)
        gyre.close()

direct = '/Users/stobin/desktop/Research/mesa-r15140/my_models/
jupiter_planets_g/gyre/work/gyre.in'
template = open(direct, 'r+')
data = template.readlines()
template.close()
```

```
    data[62] = "  grid_type = 'INVERSE'\n"
    gyre = open(direct, 'r+')
    gyre.writelines(data)
    gyre.close()

    os.chdir("/Users/stobin/desktop/Research/mesa-r15140/my_models/
    jupiter_planets_g/gyre/work")
    os.system("$GYRE_DIR/bin/gyre gyre.in")
    os.chdir("/Users/stobin/desktop/Research/mesa-r15140/my_models/
    jupiter_planets_g")
```

A.1.2 Python Script to calculate Brunt Vaisala and Lamb Frequencies

```
import numpy as np
import os
import mesa_reader as ms
import pandas as pd
import matplotlib.pyplot as plt
from subprocess import call
import shutil
import pygyre as pg
from numpy import argmax
import math
from math import log
from numpy import diff
from mpl_toolkits.mplot3d import Axes3D

mass = np.array([5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 8.5, 9.0, 9.5, 10.5, 11.5,
12.0, 12.5, 13.0, 13.5, 14.0, 15.5, 16.0, 16.5, 17.0, 17.5, 18.5, 19.0, 19.5, 20.0
#mass = np.array([9.0, 10.0, 10.5, 13.0, 17.0])
#masses of sun and earth in kg
MEarth = 5.972e+24
MSun = 1.989e+30
# Mass converted into MSun units
core_mass = mass*MEarth/MSun



for m in core_mass:
    cs=m*MSun/MEarth
    profile6 = 'LOGS/M_'+str(cs)+'_EM_core/profile6.data'
    path = os.path.exists(profile6)
```

```
if path == True:
    direc = 'LOGS/M_'+str(cs)+'_EM_core/profile6.data'
else:
    direc = 'LOGS/M_'+str(cs)+'_EM_core/profile5.data'

star = ms.MesaData(file_name=direc)
dir = pg.read_output('gyre/work/LOGS/Adiabatic
/M_'+str(m*MSun/MEarth)+'_EM_core/summary_nad_file.txt')
c = np.array(star.data("csound"))
r = np.array(star.data("radius"))
M = np.array(star.data("mass"))
gamma = np.array(star.data("gamma1"))
logP = np.array(star.data("logP"))
logrho = np.array(star.data("logRho"))
rho = 10**logrho
lnP = np.empty(shape=(1, len(logP)))
lnrho = np.empty(shape=(1, len(logrho)))
l = 3
radius = np.array(star.data("radius"))
radius = radius[:-1]
mass = np.array(star.data("mass"))
mass = mass[:-1]
logPressure = np.array(star.data("logP"))
logPressure = logPressure[:-1]
logdensity = np.array(star.data("logRho"))
logdensity = logdensity[:-1]
G = 6.67430e-8 #cgs units for gravitational constant
R = np.empty(shape=(1, len(c)))
S = np.empty(shape=(1, len(c)))
N2 = np.empty(shape=(1, len(c)))
g = np.empty(shape=(1, len(c)))
for core in core_mass:
    lnP = np.log((10**logP))
    lnrho = np.log((10**logrho))
    for i in range(len(r)-1):

        g[0, i] = (G*(M[i]*1.989e+33))/(((r[i])*6.9570e+10)**2)
        S[0, i] = math.sqrt((l*(1+l)*c[i]**2)/( (r[i]*6.9570e+10 )**2))
        R[0, i] = (r[i]*6.9570e+10)/(6.9911e+9)


        #Gamma = diff(lnP)/diff(lnrho)
        dP_dr = diff(10**logP)/diff(r*6.9570e+10)
```

```
        drho_dr = diff(10**logrho)/diff(r*6.9570e+10)
        dRho_dr = np.gradient(10**logrho, r*6.9570e+10)
        dp_dr = np.gradient(10**logP, r*6.9570e+10)

        N2[0, i] = g[0, i] * ( ( ( 1 / ( gamma[i]* ( 10**logP[i])) * dp_dr[i])
        - ( ( 1 / ( 10**logrho[i] ) )  * dRho_dr[i] ) )


        #print(dp_dr[i], dRho_dr[i])
    #N2 = np.array(star.data('brunt_N2'))
    zero_array = np.zeros(N2.shape, dtype=N2.dtype)
    #print(N2)
    N2_pos =  np.maximum(N2, zero_array)

    #print(len(N2_pos[0]), len(N2[0]), len(R[0]))

    N = np.empty(shape=(1, len(c)))

    for i in range(0, len(r)):
        N[0, i] = math.sqrt(N2_pos[0,i])

    #print(N)
    #print(Gamma)
#plt.scatter(r*6.9570e+10/6.9911e+9, dp_dr)
# Catherine's suggestion:

g = G * M * 1.989e33 / (r * 6.957e10)**2
R = r/r[0]

N2 = g * (1/ (gamma * 10**logP ) * dp_dr - 1/(10**logrho) * dRho_dr)


c2 = round(cs, 3)
#plt.scatter(R, S, s=3.0, color = 'k')
#plt.plot(R, S, color = 'k')
#plt.scatter(R, N, s=3.0, color = 'g')
plt.plot(R, np.sqrt(N2),label=c2)
plt.legend()
#s = plt.scatter(R, S)
#s = plt.scatter(R, N)

#text_labels = ['mass=9.0', 'mass=9.5', 'mass=13.5', 'mass=14.0',
'mass=16.5', 'mass=17.0']
```

```
#plt.legend(s.legend_elements()[0], text_labels)

#plt.legend(*s.legend_elements())

plt.title('Regions of g-mode Oscillations')
plt.xlabel("Jupiter Radii")
plt.ylabel("Frequency")
#plt.legend(loc="upper left")
plt.ylim(0, 0.5e-3)
plt.xlim(0, 1.05)
plt.savefig('Graphs/N&S_Frequency')
#plt.show()
```

A.1.3 Python Code to find frequencies corresponding to maximum normalized growth rate

```
    import numpy as np
import os
import mesa_reader as ms
import pandas as pd
import matplotlib.pyplot as plt
from subprocess import call
import shutil
import pygyre as pg
#from numpy import argma


#Mass in MSun units
mass = np.array([5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 8.5, 9.0, 9.5, 10.0, 10.5,
11.0, 11.5, 12.0, 12.5, 13.0, 13.5, 14.0, 14.5, 15.0, 15.5, 16.0,
16.5, 17.0, 17.5, 18.0, 18.5, 19.0, 19.5, 20.0])
#mass = np.array([10.0, 11.0, 14.5, 19.5, 20.0 ])
#masses of sun and earth in kg
MEarth = 5.972e+24
MSun = 1.989e+30
core_mass = mass*MEarth/MSun

for m in core_mass:

    s = pg.read_output('gyre/work/LOGS/Adiabatic/
    M_'+str(m*MSun/MEarth)+'_EM_core/summary_nad_file.txt')
    y = np.array(s['eta'])
    h = np.array(s['freq'].real)
```

```
    #print(h)
    x = np.array([z for z in h if 0.002< z <= 0.01])
    #print(x)
    max_y = max(y)  # Find the maximum y value
    #print(max_y)
    indices_eta = np.where(y == max_y)  # Find the indices of the max eta value
    #print(indices_eta)
    max_x = x[indices_eta]
    #print(max_x)
    core = np.ones_like(max_y)
    #print(core)
    #print(max_x, max_y)
    x = core*(m*MSun/MEarth)
    plt.scatter(x, max_x, c='blue')

    plt.title('Max $\eta$ Frequecy vs Core Mass')
    plt.ylabel('Real Frequency (cyc/day)')
    plt.xlabel('Core Mass $E_\oplus$')

    #plt.ylim(21.5, 23.5)
    plt.savefig('Graphs/maxeta_freq_vs_model')
    #plt.show()
```

A.1.4 Python Code to read and plot MESA profile and history files

```
    import numpy as np
import os
import mesa_reader as ms
import pandas as pd
import matplotlib.pyplot as plt
from subprocess import call
import shutil
import pygyre as pg


#Mass in MSun units
#mass = np.array([5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 8.5, 9.0, 9.5, 10.0,
10.5, 11.0,11.5, 12.0, 12.5, 13.0, 13.5, 14.0, 14.5, 15.0, 15.5,
16.0, 16.5, 17.0, 17.5,18.0, 18.5, 19.0, 19.5, 20.0])
#mass = np.array([6.0, 10.0, 14.0, 17.0, 17.5, 18.0, 18.5])
mass = np.array([9.0, 10.0, 10.5, 13.0, 16.5, 17.0])
#masses of sun and earth in kg
MEarth = 5.972e+24
```

```
MSun = 1.989e+30
core_mass = mass*MEarth/MSun

for m in core_mass:
    c = m*MSun/MEarth
    profile6 = '/Users/stobin/desktop/Research/mesa-r15140/my_models
    /jupiter_planets_g/LOGS/M_'+str(c)+'_EM_core/profile6.data'
    profile5 = '/Users/stobin/desktop/Research/mesa-r15140/my_models
    /jupiter_planets_g/LOGS/M_'+str(c)+'_EM_core/profile5.data'
    if os.path.exists(profile6):
        n = 6
    if os.path.exists(profile5):
        n = 5

    p = ms.MesaData('LOGS/M_'+str(m*MSun/MEarth)+'_EM_core/
    profile'+(str(n))+'.data')
    h = ms.MesaData('LOGS/M_'+str(m*MSun/MEarth)+'_EM_core/history.data')
    log = 10**(p.logRho)
    R = log*69.634e+9/(6.9911e+9)
    r = (p.radius)*(6.957e+10)/(6.9911e+9)

    c2 = round(c, 3)
    plt.plot(r, p.log_conv_vel, label = c2)
    plt.legend()
    plt.title('Convective Velocity vs Radius')
    plt.xlabel('Radius')
    plt.ylabel('Log Convective Velocity')

    plt.ylim(0.6, 0.8)
    plt.xlim(0.14, 0.22)

    plt.savefig('Graphs/log_conv_velvsradius(select)')
```

A.1.5 Python Code to read and plot GYRE summary files

```
    import numpy as np
import os
import mesa_reader as ms
import pandas as pd
import matplotlib.pyplot as plt
from subprocess import call
import shutil
import pygyre as pg
```

```
#Mass in MSun units
#mass = np.array([5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 8.5, 9.0, 9.5, 10.0, 10.5,
11.0, 11.5, 12.0, 12.5, 13.0, 13.5, 14.0, 14.5, 15.0, 15.5, 16.0,
16.5, 17.0, 17.5, 18.0, 18.5, 19.0, 19.5, 20.0])
#mass = np.array([9.0, 9.5, 13.5, 14.0, 16.5, 17.0])
mass = np.array([16.5])
#masses of sun and earth in kg
MEarth = 5.972e+24
MSun = 1.989e+30
core_mass = mass*MEarth/MSun

for m in core_mass:
    c = m*MSun/MEarth
    s = pg.read_output('gyre/work/LOGS/Adiabatic/M_'+str(m*MSun/MEarth)+'_EM_core
    /summary_nad_file.txt')
    #d = pg.read_output('gyre/work/detail.l%l.n%n.h5')



    plt.stem(s['freq'].real, s['eta'])
    #plt.figure()
    plt.xlabel('Real Frequency (cyc/day)')
    plt.ylabel('Normalized Growth Rate')
    plt.xlim(0.002, 0.005)
    #plt.ylim(-0.1,)
    plt.title(str(m*MSun/MEarth)+'$E_\oplus$ Core g-mode')
    plt.savefig('gyre/work/LOGS/Adiabatic/M_'+str(m*MSun/MEarth)+'_EM_core
    /eta_vs_real_frequency')
    #plt.show()
```

A.1.6 Python Code to plot period spacings

```
import numpy as np
import os
import mesa_reader as ms
import pandas as pd
import matplotlib.pyplot as plt
from subprocess import call
import shutil
import pygyre as pg
from numpy import argmax
```

```
import math
from math import log
from numpy import diff

#mass = np.array([5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 8.5, 9.0, 9.5, 10.0, 10.5,
11.0, 11.5, 12.0, 12.5, 13.0, 13.5, 14.0, 15.0, 15.5, 15.5, 16.0, 16.5,
17.0, 17.5, 18.5, 19.0, 19.5, 20.0])
mass = np.array([17.0])
#masses of sun and earth in kg
MEarth = 5.972e+24
MSun = 1.989e+30
# Mass converted into MSun units
core_mass = mass*MEarth/MSun



for m in core_mass:
    h=m*MSun/MEarth
    profile6 = 'LOGS/M_'+str(h)+'_EM_core/profile6.data'
    path = os.path.exists(profile6)
    if path == True:
        direc = 'LOGS/M_'+str(h)+'_EM_core/profile6.data'
    else:
        direc = 'LOGS/M_'+str(h)+'_EM_core/profile5.data'

    star = ms.MesaData(file_name=direc)
    dir = pg.read_output('gyre/work/LOGS/Adiabatic
    /M_'+str(m*MSun/MEarth)+'_EM_core/summary_nad_file.txt')

    work = np.array(dir['W'])
    w = np.where(work > 0)
    v  = np.where(work < 0)
    #frequency = np.array(dir['freq'].real)
    freq = np.array(dir['freq'].real)
    #freq = frequency[w]
    period = 1/freq
    delta_period = np.empty(shape=(len(freq)))
    #l = dir['l'][w]
    l = dir['l']

    #delta_period = np.diff(period)
    for i in range(len(freq)-1):
        delta_period[i] = np.array((period[i+1]-period[i]))
```

```
    #print(len(delta_period), len(period))
    c2 = round(h, 3)
    plt.title(str(c2)+"$E_\oplus$ Core")
    plt.ylabel("$\Delta$Period $(cyc/day)^{-1}$")
    plt.xlabel("Period $(cyc/day)^{-1}$")
    #print(len(n_g), len(delta_g))
    #plt.scatter(period, delta_period)
    s = plt.scatter(period, abs(delta_period),linewidths=0.1, alpha=.9,
    edgecolor='k', s = 20, c=l)
    plt.ylim(-1, 9)
    #plt.xlim(0, 0.015)
    text_labels = ['l=0', 'l=1', 'l=2', 'l=3']
    plt.legend(s.legend_elements()[0], text_labels)
    plt.savefig('gyre/work/LOGS/Adiabatic/M_'+str(m*MSun/MEarth)+'_EM_core
    /Delta_PeriodvsPeriod')
    #plt.show()
```

A.1.7 Python Code to plot Echelle diagrams

```
    import numpy as np
import os
import mesa_reader as ms
import pandas as pd
import matplotlib.pyplot as plt
from subprocess import call
import shutil
import pygyre as pg
from numpy import argmax
import math
from math import log
from numpy import diff
from mpl_toolkits.mplot3d import Axes3D

#mass = np.array([5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 8.5, 9.0, 9.5, 10.5,
11.5, 12.0, 12.5, 13.0, 13.5, 14.0, 15.5, 16.0, 16.5,
17.0, 17.5, 18.5, 19.0, 19.5, 20.0])
mass = np.array([17.5])
#masses of sun and earth in kg
MEarth = 5.972e+24
MSun = 1.989e+30
# Mass converted into MSun units
```

```
core_mass = mass*MEarth/MSun



for m in core_mass:
    c=m*MSun/MEarth
    profile6 = 'LOGS/M_'+str(c)+'_EM_core/profile6.data'
    path = os.path.exists(profile6)
    if path == True:
        direc = 'LOGS/M_'+str(c)+'_EM_core/profile6.data'
    else:
        direc = 'LOGS/M_'+str(c)+'_EM_core/profile5.data'

    star = ms.MesaData(file_name=direc)
    dir = pg.read_output('gyre/work/LOGS/Adiabatic
    /M_'+str(m*MSun/MEarth)+'_EM_core/summary_nad_file.txt')

    work = np.array(dir['W'])
    w = np.where(work > 0)
    v = np.where(work < 0)
    freq = np.array(dir['freq'].real)
    #frequency = np.array(dir['freq'].real)
    #freq = frequency[w]
    delta_p = np.array(dir['Delta_p'])
    l = dir['l']


    delta_freq = np.mean(np.diff(freq[l==2][-10:]))
    freqmod = (freq)%(delta_freq)
    #print(len(freq[w]), len(freq))

    fig = plt.figure(figsize=(6, 6))

    ax = fig.add_subplot()
    s = plt.scatter(freqmod[w], freq[w],linewidths=0.1, alpha=.9,
    edgecolor='k', s = 40, c=l[w], marker = 'o')


    plt.xlim(-1,16)
    #plt.ylim(0, 350)

    text_labels = ['l=0', 'l=1', 'l=2', 'l=3']
    plt.legend(s.legend_elements()[0], text_labels)
```

```
#plt.legend(h.legend_elements()[0], text_labels)
#plt.legend(*s.legend_elements())

c2 = round(c, 3)
plt.title(str(c2)+"$E_\oplus$ Core")
plt.xlabel('Frequency % $\Delta$ Frequency')
plt.ylabel('Real Frequency (cyc/day)')
plt.savefig('gyre/work/LOGS/Adiabatic/M_'+str(m*MSun/MEarth)+'_EM_core
/freqvsf_mod_deltafreq')
#plt.show()
```

A.2 MESA Inlists

A.2.1 inlist create

```
    ! inlist_create

&star_job

      mesa_dir = '../..'

      show_log_description_at_start = .true.

      save_model_when_terminate = .true.
      save_model_filename = "planet_create_3.0_ST.mod"
      required_termination_code_string = 'max_age'

      ! new adiabatic, contracting initial model
      create_initial_model = .true.
      radius_in_cm_for_create_initial_model = 7.922d9 ! in cm
      mass_in_gm_for_create_initial_model = 1.89819d30 ! in grams
      initial_model_relax_num_steps = 5

      set_initial_age = .true.
      initial_age = 0.d0 ! in years
      set_initial_dt = .true.
      years_for_initial_dt = 1.d-5 ! in years

      initial_zfracs = 6

      pre_ms_d_log10_P = 0

      !relax_opacity_max = .false.
```

```
      !relax_initial_opacity_max = .false.
      !new_opacity_max = 0
      !opacity_max_multiplier = 0

      !pgstar_flag = .true.

/ !end of star_job


&eos
      eosPT_file_prefix = 'mesa'

/ ! end of eos namelist


&kap
   Zbase = 0.02d0

   kap_file_prefix = 'a09'
   kap_lowT_prefix = 'lowT_Freedman11'

/ ! end of kap namelist

&controls

   use_gold2_tolerances = .true. ! wait a few steps
   steps_before_use_gold2_tolerances = 50

   warning_limit_for_max_residual = 1d99 ! off until evolve
   warn_when_large_virial_thm_rel_err = 1d99 ! off until evolve

   limit_for_rel_error_in_energy_conservation = 1d-5
   hard_limit_for_rel_error_in_energy_conservation = 1d-3

   convergence_ignore_equL_residuals = .true.

   max_model_number = 300
   max_number_retries = 100

! for create_initial_model
initial_Z = 0.02d0
initial_Y = 0.24d0
```

```
! when to stop
   max_age = 1.d3

! knobs
T_mix_limit = 0d0

   max_resid_jump_limit = 1d18
   max_corr_jump_limit = 1d18

   min_timestep_limit = 0

! output to files and terminal
max_num_profile_models=1000

photo_interval = 50
profile_interval = 50
history_interval = 1
terminal_interval = 10
write_header_frequency = 10

   ! surface heating
   !column_depth_for_irradiation = 300.d0 ! 3.d2 cm^2/g
   !irradiation_flux = 50260.00d0


   !atm_option = 'irradiated_grey'
   !atm_irradiated_opacity = 'fixed'

   ! interior heating
inject_uniform_extra_heat = 0.0

/


&pgstar

      ! top level controls

         !pause = .true.
            ! if true, the code waits for user to enter a RETURN on the command li
```

```
/ ! end of pgstar namelist
```

A.2.2 inlist core

```
    ! inlist_core

&star_job

      mesa_dir = '../..'

      show_log_description_at_start = .true.

      load_saved_model = .true.
      saved_model_name = "planet_create_3.0_ST.mod"

      save_model_when_terminate = .true.
      save_model_filename = "planet_core_M_15.000000000000002.mod"
      required_termination_code_string = 'max_age'

      !set_initial_age = .true.
      !initial_age = 0.0 ! in years
      !set_initial_dt = .true.
      !years_for_initial_dt = 1.d-5 ! in years

      set_initial_model_number = .true.
      initial_model_number = 0

      ! controls for adding core of a certain density
      relax_core = .true.
      new_core_mass = 4.5037707390648576e-05
      core_avg_rho = 15.d0 ! g/cm^3
      core_avg_eps = 1.d-10 ! ergs/g/sec
      dlg_core_mass_per_step = 0.05d0     ! decrease if convergence problems
      relax_core_years_for_dt = 0.1d0     ! increase(?) if convergence problems

      initial_zfracs = 6

      !relax_R_center = .true.
      !relax_initial_R_center = .true.
      !new_R_center = 1.9911d9
      !dlgR_per_step = 3d-3
      !relax_R_center_dt = 3.1558149984d1
```

```
      pre_ms_d_log10_P = 0

      !relax_opacity_max = .false.
      !relax_initial_opacity_max = .false.
      !new_opacity_max = 0
      !opacity_max_multiplier = 0

      !pgstar_flag = .true.

/ !end of star_job


&eos
      eosPT_file_prefix = 'mesa'
/ ! end of eos namelist


&kap
         Zbase = 0.02d0

      kap_file_prefix = 'a09'
      kap_lowT_prefix = 'lowT_Freedman11'

/ ! end of kap namelist

&controls

      use_gold2_tolerances = .true. ! wait a few steps
      steps_before_use_gold2_tolerances = 50

         warn_when_large_rel_run_E_err = 99.d0
         warning_limit_for_max_residual = 1d99 ! off until evolve
         warn_when_large_virial_thm_rel_err = 2d-2

      ! limit max_model_number as part of test_suite
         max_model_number = 1000

      ! surface heating
      !column_depth_for_irradiation = 300.d0 ! 3.d2 cm^2/g
      !irradiation_flux = 50260.00d0
```

```
      !atm_option = 'irradiated_grey'
      !atm_irradiated_opacity = 'fixed'

! when to stop
max_age = 2.d3

! knobs
!use_lnE_for_eps_grav = .false.
T_mix_limit = 0

      max_model_number = 1000
      max_number_retries = 30

! output to files and terminal
photo_interval = 50
profile_interval = 100
max_num_profile_models=1000
history_interval = 2
terminal_interval = 10
write_header_frequency = 10

      ! interior heating
inject_uniform_extra_heat = 0.0

/


&pgstar

      ! top level controls

         !pause = .true.
            ! if true, the code waits for user to enter a RETURN
            ! on the command line


/ ! end of pgstar namelist
```

A.2.3 inlist evolve

```
    ! inlist_evolve

&star_job
```

```
      mesa_dir = '../..'

      show_log_description_at_start = .true.

      load_saved_model = .true.

      saved_model_name = "planet_core_M_15.000000000000002.mod"
      save_model_when_terminate = .true.
      save_model_filename = "planet_evolve_3.0_ST.mod"
      required_termination_code_string = 'max_age'

      set_initial_cumulative_energy_error = .true.
      new_cumulative_energy_error = 0d0

      initial_zfracs = 6

      !relax_R_center = .true.
      !relax_initial_R_center = .true.
      !new_R_center = 6.3811d9
      !dlgR_per_step = 5.5d-3
      !relax_R_center_dt = 1.1558149984d1

      pre_ms_d_log10_P = 0

      !relax_opacity_max = .false.
      !relax_initial_opacity_max = .false.
      !new_opacity_max = 0
      !opacity_max_multiplier = 0

      !pgstar_flag = .true.

      !To enable rotation at set anular velocity
      !new_rotation_flag = .true.
      !rotation_flag = .true.
      !change_rotation_flag = .true.
      !set_omega = .true.
      !new_omega = 1.75704e-4


/ !end of star_job
```

```
&eos
      eosPT_file_prefix = 'mesa'
/ ! end of eos namelist


&kap
      Zbase = 0.02d0

      kap_file_prefix = 'a09'
      kap_lowT_prefix = 'lowT_Freedman11'

/ ! end of kap namelist

&controls

      use_gold2_tolerances = .true. ! wait a few steps
      steps_before_use_gold2_tolerances = 50

      warn_when_large_rel_run_E_err = 99.d0
      warn_when_large_virial_thm_rel_err = 5d-2

      use_dedt_form_of_energy_eqn = .true.

      num_trace_history_values = 2
      trace_history_value_name(1) = 'rel_E_err'
      trace_history_value_name(2) = 'log_rel_run_E_err'


      max_model_number = 1600
      max_number_retries = 200

! surface heating

      !column_depth_for_irradiation = 300.d0 ! 3.d2 cm^2/g
      !irradiation_flux = 50260.00d0

      !Can have one or the other

      !atm_option = 'irradiated_grey'
      !atm_irradiated_opacity = 'fixed'

! interior heating
inject_uniform_extra_heat = 0.0
```

```
! when to stop
max_age = 4.503d9
Teff_lower_limit = 20.d0

      min_timestep_limit = 0

! knobs
T_mix_limit = 0

! output to files and terminal
photo_interval = 50
profile_interval = 50
max_num_profile_models=1000
history_interval = 2
terminal_interval = 10
write_header_frequency = 10

      write_pulse_data_with_profile = .true.
      pulse_data_format = 'GYRE'

/


&pgstar

      ! top level controls

        !pause = .true.
          ! if true, the code waits for user to enter a RETURN on the command
          ! line

      ! show HR diagram
      ! this plots the history of L,Teff over many timesteps
      HR_win_flag = .true.

      ! set static plot bounds
       HR_logT_min = 3.5
       HR_logT_max = 4.6
       HR_logL_min = 2.0
       HR_logL_max = 6.0

      ! set window size (aspect_ratio = height/width)
```

```
       HR_win_width = 6
       HR_win_aspect_ratio = 1.0


      ! show temperature/density profile
      ! this plots the internal structure at single timestep
       TRho_Profile_win_flag = .true.

      ! add legend explaining colors
       show_TRho_Profile_legend = .true.

      ! display numerical info about the star
        show_TRho_Profile_text_info = .true.

      ! set window size (aspect_ratio = height/width)
       TRho_Profile_win_width = 8
       TRho_Profile_win_aspect_ratio = 0.75

/
```

A.3 GYYE Inlist

```
    &constants
/

&model
  model_type = 'EVOL'
  file = '/Users/stobin/desktop/Research/mesa-r15140/my_models/jupiter_planets_g/
  LOGS/M_14.5_EM_core/profile6.data.GYRE'
  file_format = 'MESA'
/

&mode
  l = 0
  m = 0
/

&mode
  l = 1
  m = 0
/

&mode
```

```
  l = 2
  m = 0
/

&mode
  l = 3
  m = 0
/

!&mode
  !l = 4
  !m = 0
!/

!&mode
  !l = 5
  !m = 0
!/

!&mode
  !l = 6
  !m = 0
!/

&osc
  adiabatic = .TRUE.
  nonadiabatic = .TRUE.
  outer_bound = 'VACUUM'
/

&rot
/

&num
  diff_scheme = 'MAGNUS_GL2'
  restrict_roots = .FALSE.
  nad_search = 'AD'
  ad_search = 'BRACKET'
  n_iter_max = 50
/

&scan
  grid_type = 'INVERSE'
```

```
  freq_min = 0.0001226187189467836
  freq_max = 0.01226187189467836
  n_freq = 300
  !axis = 'REAL'
  !freq_max_units = 'CYC_PER_DAY'
  !freq_min_units = 'CYC_PER_DAY'
/

!&scan
  !grid_type = 'LINEAR'
  !freq_min = 3.6277368806689076
  !freq_max = 36.277368806689076
  !n_freq = 300
  !axis = 'IMAG'
!/

&grid
  w_osc = 10
  w_exp = 2
  w_ctr = 5
/

&ad_output
  summary_file = '/Users/stobin/desktop/Research/mesa-r15140/my_models/
  jupiter_planets_g/gyre/work/LOGS/Adiabatic/
  M_15.000000000000002_EM_core/summary_ad_file.txt'
  summary_item_list = 'M_star,R_star,L_star,l,m,n_g,n_p,omega,E_norm,freq,
  freq_units,freq_frame,j,f_T,f_g,eta,W,Delta_p,Delta_g'
  summary_file_format = 'TXT'
  detail_item_list = 'omega,n,freq,freq_units,freq_frame,j,l,m,omega_int,
  eta,W,dW_dx,R_star,M_star,P,rho,T,L_star'
  detail_template = 'detail.l%l.n%n.h5'
  detail_file_format = 'TXT'
  freq_units = 'CYC_PER_DAY'
/

&nad_output
  summary_file = '/Users/stobin/desktop/Research/mesa-r15140/my_models/
  jupiter_planets_g/gyre/work/LOGS/Adiabatic/
  M_15.000000000000002_EM_core/summary_nad_file.txt'
  summary_item_list = 'M_star,R_star,L_star,l,m,n_g,n_p,omega,E_norm,freq,
  freq_units,freq_frame,j,f_T,f_g,eta,W,Delta_g,Delta_p'
  summary_file_format = 'TXT'
```

```
  detail_item_list = 'omega,n,freq,freq_units,freq_frame,j,l,m,omega_int,
  eta,W,dW_dx,R_star,M_star,P,rho,T,L_star'
  detail_template = 'detail.l%l.n%n.h5'
  detail_file_format = 'TXT'
  freq_units = 'CYC_PER_DAY'
/
```