

ARISTOTLE UNIVERSITY OF THESSALONIKI



PHYSICS DEPARTMENT
BACHELOR THESIS

Constraints on the Nuclear Equation of State induced by Binary Neutron Star Mergers

Author:

Anastasia Zachariadou

Supervisor:

Ch. C. Moustakidis

October 2023

Acknowledgments

I would like to express my heartfelt gratitude to my supervisor, Professor Ch. C. Moustakidis for his invaluable guidance, unwavering support, and profound expertise. His feedback and encouragement have been instrumental in shaping this thesis. I am also deeply indebted to my friend Panagiotis, whose advice, emotional support and insightful perspectives have been a constant source of inspiration and motivation. Their friendship and encouragement have made this academic journey not only bearable but also enjoyable.

Abstract

The aim of this thesis is to investigate Neutron Stars (NS) and their Equation of State (EOS) with a focus on Mass (M), Radius (R) and Tidal Deformability (λ) constraints using multimesenger observations from the GW170817 binary neutron star (BNS) inspiral event. The first chapter covers essential theoretical aspects of neutron stars, including their formation process, properties and structure, as well as a brief overview of Binary Neutron Stars with an emphasis on the significance of tidal deformability and the role of gravitational waves in their study. The subsequent chapter introduces the mathematical framework of TOV equations and then the system is solved for a sufficient number of different EOS. The result of this process is the determination of the Mass and Radius of a NS under various EOS models. Additionally, after the mathematical formalism of the tidal parameters is established, the tidal love number (k_2) and the tidal deformability (λ) are determined through the numerical integration of the system of equations. Our study then explores the relationship of these parameters with M , R and compactness parameter (β), presenting the results through graphs. Notably, an investigation into a possible systematic connection between the Star's radius at 1.4 solar masses ($1.4M_{\odot}$) and λ and k_2 is conducted. Wherever it was possible, our results were compared with data from LIGO, demonstrating that a majority of tested EOS models align with observed credible regions.

Περίληψη

Σκοπός αυτής της διατριβής είναι η διερεύνηση των Αστέρων Νετρονίων και της Καταστατικής Εξίσωσής τους (K.E.) με έμφαση στους περιορισμούς της Μάζας (M), της Ακτίνας (R) και της Παλιρροιακής Παραμόρφωσης (λ) χρησιμοποιώντας παρατηρήσεις από το σήμα GW170817 συγχώνευσης Δυαδικού συστήματος Αστέρων Νετρονίων. Το πρώτο κεφάλαιο καλύπτει ουσιώδεις θεωρητικές πτυχές σχετικά με τους αστέρες νετρονίων, συμπεριλαμβανομένης της διαδικασίας σχηματισμού τους, των ιδιοτήτων τους και της δομής τους, καθώς και μια σύντομη επισκόπηση των Δυαδικών συστημάτων με έμφαση στη σημασία της παλιρροιακής παραμόρφωσης και τον ρόλο των βαρυτικών κυμάτων στη μελέτη τους. Το επόμενο κεφάλαιο παρουσιάζει το μαθηματικό πλαίσιο των εξισώσεων TOV και στη συνέχεια το σύστημα επιλύεται αριθμητικά για έναν επαρκή αριθμό διαφορετικών καταστατικών εξισώσεων. Το αποτέλεσμα αυτής της διαδικασίας είναι η καθορισμός της Μάζας και της Ακτίνας ενός αστέρα για διάφορα μοντέλα. Επιπλέον, μετά την καθιέρωση των τύπων και του συστημάτος εξισώσεων που οδηγούν στην εύρεση των παλιρροιακών παραμέτρων, ο παλιρροιακός αριθμός L_{ove} k_2 και η παλιρροιακή παραμόρφωση λ καθορίζονται μέσω της αριθμητικής ολοκλήρωσης του συστήματος αυτού. Στη συνέχεια, εξετάζεται η σχέση αυτών των παλιρροιακών παραμέτρων με τα μεγέθη M , R και παράγοντα συμπίεσης β . Τα αποτελέσματα παρουσιάζονται μέσω διαγραμμάτων. Τέλος, διεξάγεται στατιστική ανάλυση με στόχο την εύρεση πιθανής συστηματικής σύνδεσης μεταξύ της ακτίνας του Αστέρα στις 1.4 ηλιακές μάζες ($R(1.4(M_\odot))$) και των λ και k_2 . Όπου ήταν δυνατόν, τα αποτελέσματά μας συγκρίθηκαν με δεδομένα από το LIGO, επιδεικνύοντας ότι τα περισσότερα από τα δοκιμασμένα μοντέλα K.E. έρχονται σε συμφωνία με τις περιοχές αξιοπιστίας.

Contents

1	Introduction: A brief overview of Neutron Stars	8
1.1	Star Formation	8
1.1.1	Fate of Stars of Different Masses	8
1.1.2	Neutron Star Formation	10
1.2	Neutron Star Properties and Structure	10
1.2.1	Characteristics	10
1.2.2	Structure	11
1.2.3	Types of Neutron Stars	12
1.3	Scientific History of Neutron Stars	13
1.4	Binary Neutron Stars and Tidal Deformability	14
1.4.1	Binary Neutron Star Mergers	14
1.4.2	Gravitational Waves and Tidal Deformability	16
2	Mathematical Framework	17
2.1	TOV Equations	17
2.2	Equations of State	18
2.3	Tidal Parameters	21
3	Computational Analysis and Results	24
3.1	Mass-Radius Relation	24
3.2	Tidal Effects	26
3.2.1	Tidal Love number k_2	27
3.2.2	Tidal Deformability λ	29
3.2.3	y_R	31
3.3	Statistical Analysis	33
3.3.1	Correlation Analysis method	34
3.3.2	Linear Regression method	35
3.4	Comparison to Observational Data	38
3.4.1	The BNS GW170817 event	38
3.4.2	Results	39
4	Concluding Remarks	41
5	Appendix: Python Codes	42
5.1	Eos Library	42
5.2	TOV and Tidal effects: numerical integration	47
5.3	Graphs and Statistical Analysis	49

Εκτεταμένη Περίληψη

Εισαγωγή

Το θέμα της παρούσας πτυχιακής εργασίας αφορά τους Αστέρες Νετρονίων, επομένως όταν εύλογο πρώτα να παρουσιάσουμε το βασικό θεωρητικό υπόβαθρο σχετικά με αυτούς. Οι Αστέρες Νετρονίων είναι από τα πιο αινιγματικά αστρικά αντικείμενα στο σύμπαν. Γεννιούνται ως αποτέλεσμα εκρήξεων αστεριών μεγάλης μάζας ($> 8M_{\odot}$) όταν αυτά φτάνουν στο τέλος της ζωής τους. Κύριο χαρακτηριστικό τους αποτελεί το ότι είναι εξαιρετικά πυκνά αντικείμενα, με σχετικά μεγάλες τυπικές τιμές μάζας: $1.1 - 2M_{\odot}$ και αντίστοιχα πολύ μικρές τιμές ακτίνας: $10 - 13 km$. Αποτελούνται από πυρηνική ύλη, κυρίως νετρόνια, χαρακτηρίζονται συγκέντρως από πολύ ισχυρά μαγνητικά πεδία ($B = 10^{12} - 10^{15} G$) και πολύ υψηλές περιστροφικές ταχύτητες.

Αναφορικά με τη δομή τους, οι Αστέρες Νετρονίων αποτελούνται από 5 στρώματα: την ατμόσφαιρα, την εξωτερική και εσωτερική χρούστα και τον εξωτερικό και εσωτερικό πυρήνα. Τα βασικά χαρακτηριστικά κάθε επιπέδου δίνονται παρακάτω.

- **Ατμόσφαιρα:** Είναι το εξώτερο στρώμα του αστέρα και αποτελείται από μια υπέρλεπτη στρώση πλάσματος. Καθορίζει τις ακτινοβολιακές ιδιότητες του αστέρα (η/μ φάσμα, θερμικά χαρακτηριστικά) και παρέχει πληροφορίες για την επιφάνεια (θερμοκρασία, βαρύτητα, σύνθεση, μαγνητικό πεδίο). Το πάχος της δεν είναι συγκεκριμένο, αλλά διαφοροποιείται ανάλογα με τη θερμοκρασία.
- **Εξωτερική Κρούστα:** Βρίσκεται ακριβώς κάτω από την ατμόσφαιρα και αποτελείται από ιόντα και e. Όσο αυξάνεται το βάθος και η πυκνότητα, τα e γίνονται ρελατιβιστικά και οδηγούμαστε σε πλήρη ιονισμό των ατόμων και στον σχηματισμό ενός ισχυρά συσχετισμένου συστήματος Coulomb το οποίο μπορεί να στερεοποιηθεί. Στην κρίσιμη τιμή πυκνότητας $4 \times 10^{11} g \times cm^{-3}$ τα νετρόνια αρχίζουν να 'δραπετεύουν', δημιουργώντας τελικά ένα ελεύθερο νετρονικό αέριο (neutron drip).
- **Εσωτερική Κρούστα:** Έχει περίπου $1 km$ έκταση και περιλαμβάνει e, ελεύθερα n και πυρήνες ατόμων πλούσιους σε n. Με αύξηση της πυκνότητας αυξάνεται και το ποσοστό των ελεύθερων n το οποίο προκαλεί "softening" της K.E. Πιο χαμηλά, δημιουργείται 'μανδύας' και στα όρια με τον πυρήνα μένουν μόνο n και νουκλεόνια σε υπέρρευστη κατάσταση.
- **Εξωτερικός Πυρήνας:** Αυτό το επίπεδο έχει εύρος πυκνοτήτων $0.5\rho_0 - 2\rho_0$, ($\rho_0 = 2.8 \times 10^{14} g \times cm^{-3}$) και έκταση αρκετών km. Έχει τη λεγόμενη premu σύσταση. Κι εδώ μπορεί να υπάρχουν υπέρρευστες καταστάσεις.
- **Εσωτερικός Πυρήνας:** Φτάνει πυκνότητες $\rho > 2\rho_0$ (ως και $10 - 15\rho_0$). Η σύνθεση και η K.E είναι ακόμη υπό έρευνα. Διάφορες υποθέσεις: εξωτικά σωματίδια (υπερόνια Σ,Λ), πιονικά/καονικά συμπυκνώματα, ελεύθερη ύλη κουάρκ, μικτή ύλη. Αυτές οι υποθέσεις μπορεί να προκαλέσουν μεταβατικές φάσεις, επηρεάζοντας την K.E και τη συμπεριφορά του αστέρα.

Είναι σύνηθες φαινόμενο οι αστέρες νετρονίων να εμφανίζονται σε Δυαδικά συστήματα. Αυτά τα συστήματα αποτελούνται από δύο αστέρια σε τροχιά γύρω από κοινό κέντρο μάζας. Μοίρα των αστέρων σε τέτοια συστήματα είναι η συγχώνευση, η οποία εκτελείται σε τρία στάδια: Κατά το πρώτο, το λεγόμενο inspiral, οι αστέρες περιστρέφονται ο ένας γύρω από τον άλλον με την απόστασή τους σταδιακά να μειώνεται και την ταχύτητα να αυξάνεται. Κατά τη φάση αυτή παράγονται βαρυτικά κύματα με μέγιστο συχνότητας ακριβώς πριν τη σύγκρουση. Ακολουθεί η συγχώνευση (Merger). Στο στάδιο αυτό, οι παλιρροιακές παράμετροι tidal Love number k_2 και tidal deformability λ γίνονται σημαντικές και μπορούν να μετρηθούν άμεσα μέσω των βαρυτικών κυμάτων και να μας παρέχουν πληροφορίες για τη δομή, το μέγεθος και την καταστατική εξίσωση των αστέρων. Το αποτέλεσμα της συγχώνευσης εξαρτάται από το είδος και τις μάζες των αρχικών αστέρων του συστήματος. Στη μελέτη μας χρησιμοποιήθηκαν παρατηρησιακά δεδομένα του LIGO από το σήμα GW170817 (αποτέλεσμα τέτοιας συγχώνευσης), με σκοπό τη σύγκριση με τα δικά μας αποτελέσματα.

Μαθηματικό Υπόβαθρο

Στο ακόλουθο μέρος δίνονται τα βασικότερα σημεία του μαθηματικού υποβάθρου που χρησιμοποιήσαμε στη μελέτη μας. Οι Εξισώσεις TOV (με κατάλληλο scaling):

Υδροαστατική Ισορροπία

$$\begin{aligned} \frac{d\bar{P}(r)}{dr} &= -1.474 \frac{\bar{\mathcal{E}}(r)\bar{M}(r)}{r^2} \left(1 + \frac{\bar{P}(r)}{\bar{\mathcal{E}}(r)}\right) \left(1 + 11.2 \times 10^{-6} r^3 \frac{\bar{P}(r)}{\bar{M}(r)}\right) \\ &\quad \times \left(1 - 2.948 \frac{\bar{M}(r)}{r}\right)^{-1} \end{aligned}$$

Σχέση Μάζας-Ακτίνας

$$\frac{d\bar{M}(r)}{dr} = 11.2 \times 10^{-6} r^2 \bar{\mathcal{E}}(r)$$

Από την επίλυση του συστήματος των εξισώσεων για καταστατικές εξισώσεις της μορφής $\mathcal{E}(P)$ (P σε μονάδες $MeV \cdot fm^{-3}$) προκύπτουν οι τιμές για μάζα M και ακτίνα R . Σημειώνεται πως λόγω της ιδιαίτερης δομής της χρούστας, έγιναν κάποιες αλλαγές στη μορφή των K.E. στο κομμάτι αυτό, σε ανάλογα διαστήματα τιμών της πίεσης.

Για τον προσδιορισμό των παλιρροιακών παραμέτρων απαιτείται η προσθήκη ακόμη μιας διαφορικής εξίσωσης στο σύστημα, η οποία λύνεται ταυτόχονα με τις TOV:

$$r \frac{dy(r)}{dr} + y^2(r) + y(r)F(r) + r^2Q(r) = 0$$

Τα F, Q είναι συναρτησοειδή των $M(r), P(r), \mathcal{E}(r)$

Οι παλιρροιακές παράμετροι που μας ενδιαφέρουν δίνονται από τις σχέσεις:

Tidal Love number:

$$\begin{aligned} k_2(\beta, y_R) &= \frac{8\beta^5}{5} (1 - 2\beta)^2 [2 - y_R + 2\beta(y_R - 1)] \\ &\quad \times [2\beta(6 - 3y_R + 3\beta(5y_R - 8))] \\ &\quad + 4\beta^3 (13 - 11y_R + \beta(3y_R - 2) + 2\beta^2(1 + y_R)) \\ &\quad + 3(1 - 2\beta)^2 [2 - y_R + 2\beta(y_R - 1)] \ln(1 - 2\beta)]^{-1}, \end{aligned}$$

$\beta = 1.474M/R$ παράμετρος συμπίεσης.

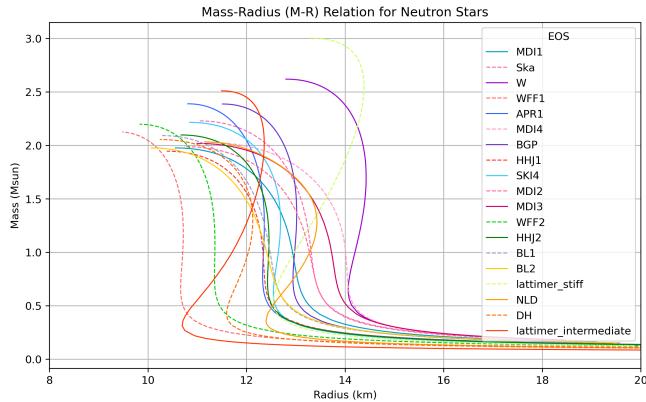
Παλιρροιακή παραμόρφωση:

$$\lambda = \frac{2}{3}k_2 R^5$$

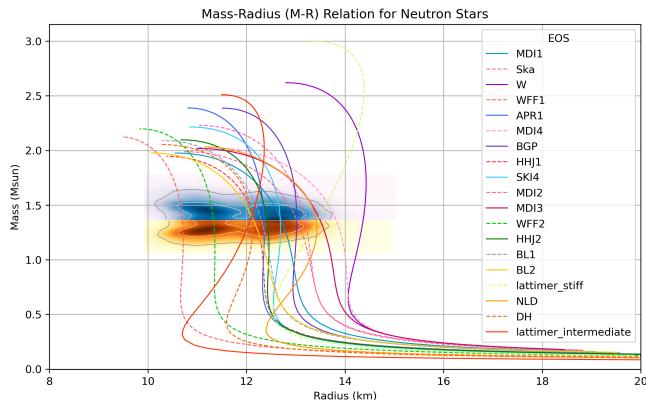
Η διαδικασία που ακολουθήθηκε, εν συντομίᾳ είναι η εξής: Μέσω κώδικα σε Python , για κάθε καταστατική εξίσωση έγινε τατυόχρονη αριθμητική επίλυση των TOV, από τις οποίες ποέκυψαν τα M και R και της διαφορικής, η οποία έδωσε το y_R . Ουσιαστικά διεξάχθη αριθμητική ολοκλήρωση από το κέντρο ($r = 0$) ως τα όρια του αστέρα ($r = R$) με οριακές συνθήκες $P(0) = P_c$, $M(0) = 0$, $y(0) = 2$ και $P(R) = M$, $y(R) = y_R$ αντίστοιχα. Αντί για 0 χρησιμοποιήθηκαν πολύ μικρές τιμές και για κεντρική πίεση επιλέχθηκε εύρος $P_c = 1 - 1200 MeV \cdot fm^{-3}$. Με τα M , R γνωστά υπολογίστηκε το β και έχοντας και το y_R προσδιορίστηκαν και οι παλιρροιακές παράμετροι k_2 και λ για κάθε καταστατική εξίσωση. Έγιναν τα αντίστοιχα διαγράμματα για τη μελέτη συμπεριφοράς των διαφόρων μεγεθών και ακολούθησε μια σατιστική ανάλυση.

Αποτελέσματα

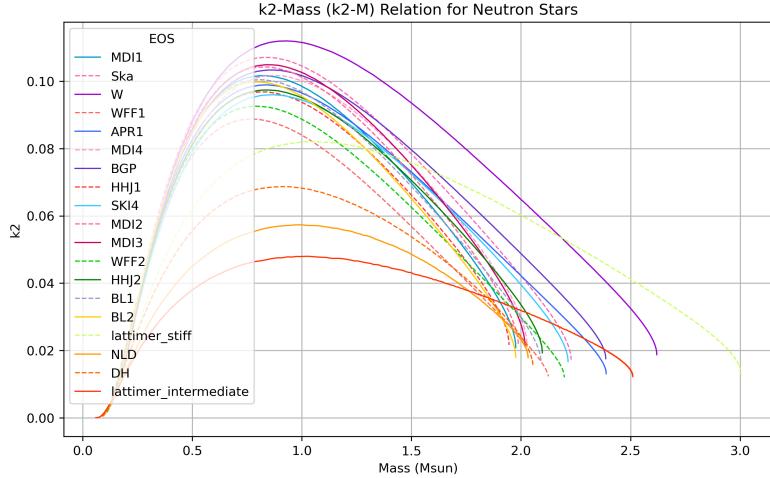
Παρουσιάζονται ενδεικτικά κάποια από τα γραφήματα που δημιουργήσαμε βάσει των αποτελεσμάτων της διαδικασίας που περιγράφηκε παραπάνω.



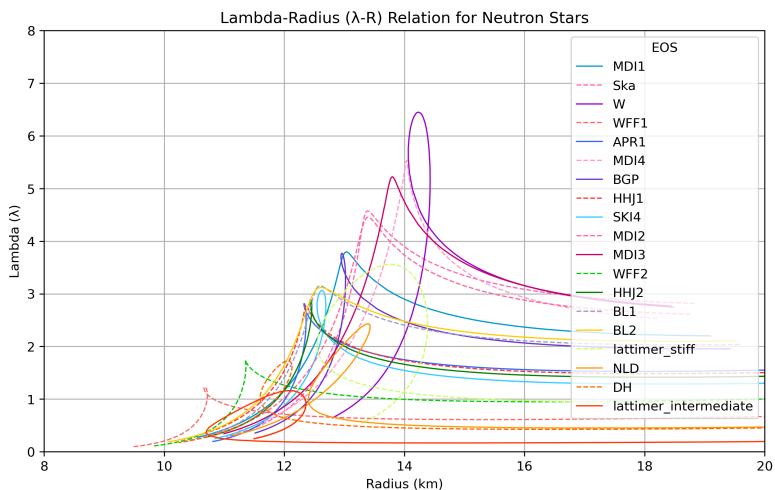
Διάγραμμα $M - R$



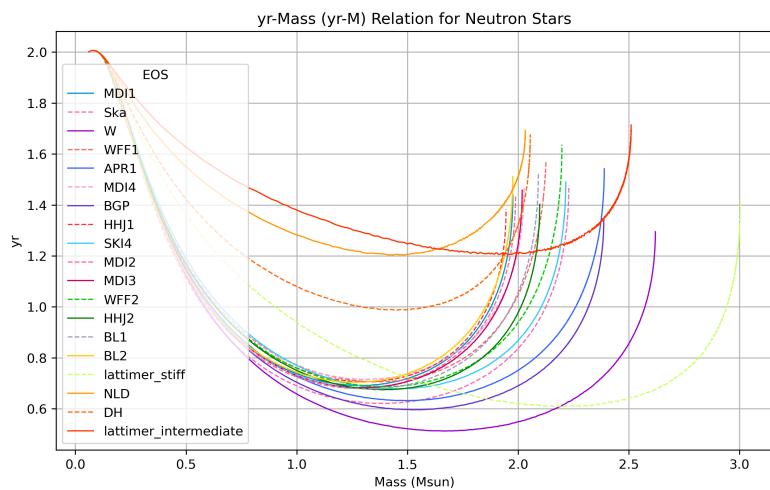
Διάγραμμα $M - R$: Σύγκριση με δεδομένα LIGO



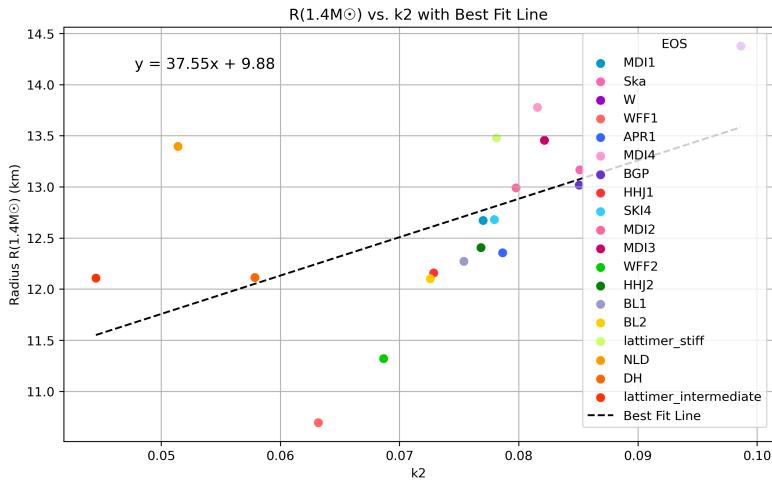
Διάγραμμα $k_2 - M$



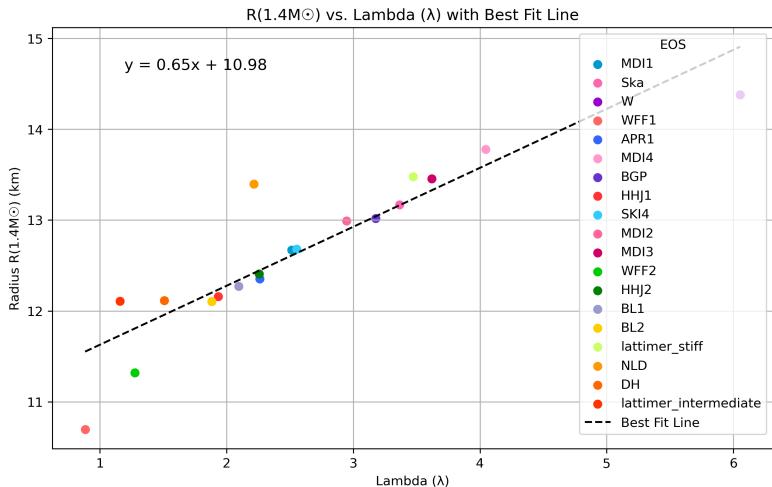
Διάγραμμα $\lambda - R$



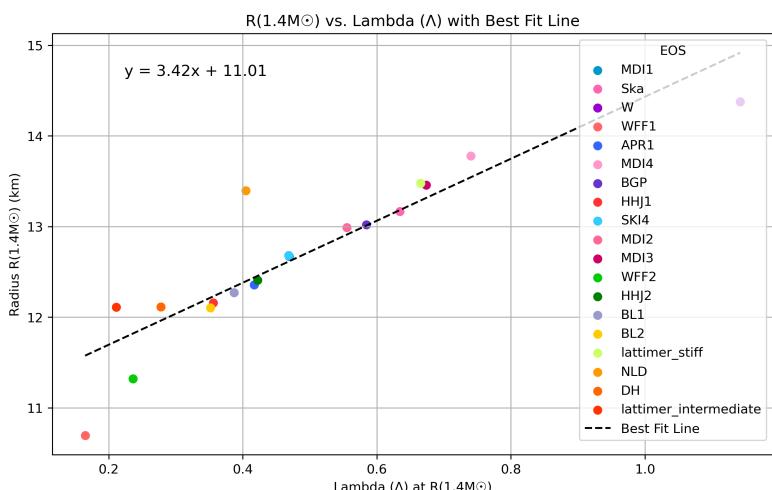
Διάγραμμα $y_R - R$



Στατιστική Ανάλυση: Καλύτερη ευθεία $R(1.4M_{\odot}) - k_2$



Στατιστική Ανάλυση: Καλύτερη ευθεία $R(1.4M_{\odot}) - \lambda$



Στατιστική Ανάλυση: Καλύτερη ευθεία $R(1.4M_{\odot}) - \Lambda$

Συμπεράσματα

Τα βασικά συμπεράσματα που προέκυψαν από τη μελέτη μας:

- Μέσω των $M - R$ διαγράμματων παρατηρήθηκε ότι οι πιο stiff K.E. προβλέπουν εν γένει μεγαλύτερες τιμές M και R .
- Οι K.E. που χρησιμοποιήθηκαν αναπαράγουν μέγιστη μάζα τουλάχιστον $2M_{\odot}$, επομένως υπάρχει συνέπεια και συμφωνία με τις μέχρι τώρα παρατηρήσεις.
- Από τη μελέτη του $M - R$ διαγράμματος σε συνδυασμό με παρατηρήσεις από το LIGO, μόνο 2 K.E. από τις 19 που χρησιμοποιήθηκαν βρέθηκαν εκτός περιοχών αξιοπιστίας. Αυτό ίσως αποτελεί ένδειξη ότι πρέπει να αποκλειστούν ως μοντέλα.
- Από τα διαγράμματα των παλιρροιαξών παραμέτρων προέκυψε ότι οι stiffer K.E. προβλέπουν μεγαλύτερες τιμές k_2 και λ , αντίθετα με τις πιο soft.
- Το λ παρουσιάζει μεγαλύτερο εύρος τιμών σε σχέση με το k_2 και επίσης επιβεβαιώθηκε η έντονη εξάρτησή του από την ακτίνα R .
- Από τη στατιστική ανάλυση, έγινε εμφανής η ύπαρξη σχέσης γραμμικής αύξησης μεταξύ της ακτίνας και της παλιρροιακής παραμόρφωσης.

Chapter 1

Introduction: A brief overview of Neutron Stars

The subject of this thesis concerns Neutron Stars, so it would be reasonable in this first chapter to give some theoretical background and information of significant importance about them. Neutron stars, some of the most enigmatic objects in the universe, are born from the explosive aftermath of massive stars (mass greater than $6 - 8M_{\odot}$). Their existence hinges on the remarkable phenomenon of supernova explosions. These celestial objects are exceptionally dense, composed mostly of nuclear matter -primarily neutrons- and characterized by intriguing physical properties. More specifically, they are stars with small radii, typically 10-13km and comparatively large masses, typically $1.1 - 2M_{\odot}$.

1.1 Star Formation

The formation of neutron stars is a fascinating process that originates from the life cycles of massive stars. To understand how a massive star transforms into a neutron star, we must delve into the complex interplay of stellar evolution, nuclear physics, and gravitational collapse. This process is particularly influenced by a star's mass, which determines its ultimate fate [4].

1.1.1 Fate of Stars of Different Masses

The evolution of a star is intricately linked to the nuclear fusion processes occurring within it. As hydrogen fusion in the core halts the star's contraction, the gradual depletion of hydrogen reserves eventually reaches a critical point where the star can no longer sustain itself against gravitational collapse. This pivotal moment marks the transition to various evolutionary scenarios, which are outlined below based on mass categories.

$M < 0.8M_{\odot}$: For stars with a mass less than 0.8 times that of the Sun (M_{\odot}), their evolution unfolds through a series of stages. It begins with a transformation into a red subgiant, followed by becoming a red giant. Subsequently, the star progresses through phases, including helium white dwarf, red supergiant, another helium white dwarf phase, and finally, the formation of a planetary nebula. It's important to note that the term "planetary nebula" in this context actually refers to a shell of gas surrounding the helium white dwarf, containing remnants of the earlier red supergiant phase.

$0.8M_{\odot} < M < 3M_{\odot}$: Stars within this mass range experience a distinctive evolution compared to their lower-mass counterparts. After becoming red giants, these stars possess sufficiently high temperatures to facilitate the consumption of helium. The evolutionary path includes stages such as red subgiant, red giant, horizontal branch, red supergiant, and ultimately culminates in the formation of a carbon and oxygen white dwarf, enveloped by a planetary nebula. During the red giant phase, helium undergoes uncontrollable fusion, leading to a phenomenon known as the Helium Flash, which results in a rise in luminosity. Once the star exhausts its helium reserves, it enters the asymptotic giant branch (AGB) before evolving into a red supergiant. The planetary nebula contains the mass that was shed by the star during the AGB and supergiant phases.

$3M_{\odot} < M < 10M_{\odot}$: In this range, the precise mass of the star plays a crucial role in determining its fate. After the red giant phase, the remaining mass may exceed the Chandrasekhar limit ($\sim 1.4M_{\odot}$). If the mass falls within the interval $[3M_{\odot}, 5M_{\odot}]$, carbon does not undergo fusion, leading to the formation of a white dwarf composed of carbon and oxygen. For stars with higher masses, the temperature is sufficiently high to trigger a Carbon Flash, a violent burning of carbon, making it challenging to predict the exact final state. The outcome may involve the formation of a neutron star or the catastrophic collapse of the star leading to a black hole, often associated with a type II supernova explosion.

$M > 10M_{\odot}$: For stars exceeding 10 times the mass of the Sun, a dormant core forms, sustained by ongoing nuclear fusion of its elements. The stability of this core is maintained as long as nuclear fusion persists, preventing gravitational collapse. However, once the core reaches either maximum stability, composed mainly of iron (Fe56), or when the temperature is insufficient to sustain fusion, it undergoes a sudden collapse, resulting in a type II supernova explosion. The ultimate fate hinges on the initial mass: stars with masses not exceeding $\sim 20M_{\odot}$ may give rise to neutron stars, while those with masses surpassing $20 - 25M_{\odot}$ tend to evolve into black holes.

In summary, the specific outcomes of a star's evolution depend intricately on its initial mass. Stars in the mass range of $\sim 10M_{\odot}$ to $\sim 20M_{\odot}$ typically evolve into neutron stars, while those in the $\sim 5M_{\odot}$ to $\sim 10M_{\odot}$ range also have the potential to become neutron stars. These scenarios are instrumental in understanding the conditions leading to the birth of neutron stars.

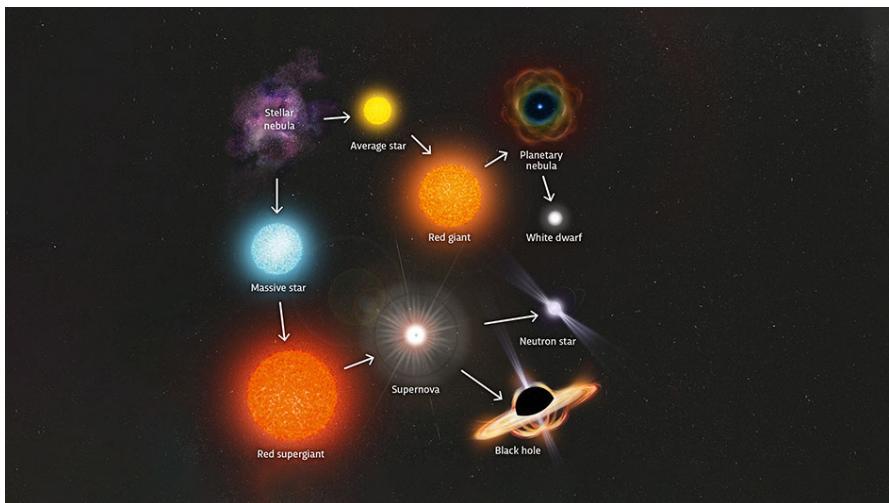


Fig 1.1: Star Evolution Outcomes

1.1.2 Neutron Star Formation

Having talked about the evolution of stars of different masses, we can now focus more on Neutron Stars specifically [4, 7].

Massive stars, typically those with initial masses exceeding about 8 times that of our Sun ($8M_{\odot}$), follow a unique path in the H-R diagram during their lifetimes. These stars start as hot, luminous objects in the upper-left region of the diagram, known as the Main Sequence (MS). Here, they fuse hydrogen into helium in their cores, emitting immense energy. As the hydrogen in the core depletes, these stars leave the MS and move toward the upper-right part of the H-R diagram. Here, they undergo successive stages of nuclear burning, forming heavier elements like carbon, oxygen, silicon, and iron in concentric shells around the core.

When a massive star exhausts its nuclear fuel, the core cannot support itself against gravitational collapse. The core rapidly contracts, leading to an increase in temperature and density. As the core reaches temperatures of about $100 \cdot 10^9$ Kelvin, iron nuclei start capturing electrons and turn into neutrons via electron capture. This process consumes energy and disrupts electron degeneracy pressure, allowing gravitational collapse to proceed unhindered. The core collapse occurs explosively, and within fractions of a second, the star's inner regions collapse to nuclear densities. The outer layers of the star, now devoid of support, rebound off the dense core in a tremendous explosion known as a supernova. This explosion releases an astonishing amount of energy, often briefly outshining an entire galaxy.

The core that remains after a supernova explosion is incredibly dense and composed almost entirely of neutrons, hence the term "neutron star". These neutron stars typically have masses between about $1.4M_{\odot} - 2.1M_{\odot}$, but are incredibly compact, with radii on the order of 10km.

1.2 Neutron Star Properties and Structure

1.2.1 Characteristics

As mentioned before, representative values for neutron star mass and radius are approximately $M \approx 1.4M_{\odot}$ and $R \approx 10\text{km}$. To put this into perspective, our sun, a typical main-sequence star, boasts a radius on the order of 10^5km . Thus, a neutron star's relatively normal mass is confined within a tiny radius, signifying its exceptional density and compactness. In fact, neutron stars are so dense that their mean density is approximately $\rho \approx 7 \cdot 10^{14}\text{gcm}^{-3}$, with even higher densities toward the central core. This makes them incredibly dense and compact objects, with regions in the core reaching densities approximately 10 to 20 times that of ρ_0 , where ρ_0 represents the mean density of heavy nuclear matter, approximately $2.8 \cdot 10^{14}\text{gcm}^{-3}$.

The intense gravitational fields of neutron stars are governed by Newton's Law of Universal Gravitation ($g = GM/R^2$) [7]. The gravitational energy (E_{grav}) associated with a neutron star is a substantial quantity, defined as $E_{grav} = (3GM^2)/(5R)$. Substituting the typical values presented above, we find $E_{grav} \approx 5 \cdot 10^{53}\text{erg}$, which is approximately equivalent to 0.2 times the star's mass-energy equivalence ($E = mc^2$). This energy is a testament to the immense gravitational potential held by neutron stars.

Neutron stars are also characterized by incredibly strong magnetic fields, often reaching values of $B \approx 10^{12}$ to 10^{15}G , which are billions to trillions of times stronger than Earth's magnetic field. The origin of such powerful magnetic fields is still a subject of scientific investigation. These magnetic fields play a significant role in various astrophysical processes, including the emission of intense radiation and the formation of pulsars.

These stars can also exhibit rapid rotation, with some spinning hundreds of times per second. This high rotation rate is a consequence of the conservation of angular momentum during the star's collapse. As a result, neutron stars are observed as pulsars when their beams of radiation intersect with Earth, providing valuable insights into their properties and internal structures.

1.2.2 Structure

Neutron stars exhibit a complex, multi-layered structure, with each layer playing a vital role in shaping the star's observable characteristics. These layers can be divided into five distinct regions, each with its own unique properties and behaviors: the atmosphere, the outer crust, the inner crust, the outer core, and the inner core [13, 7].

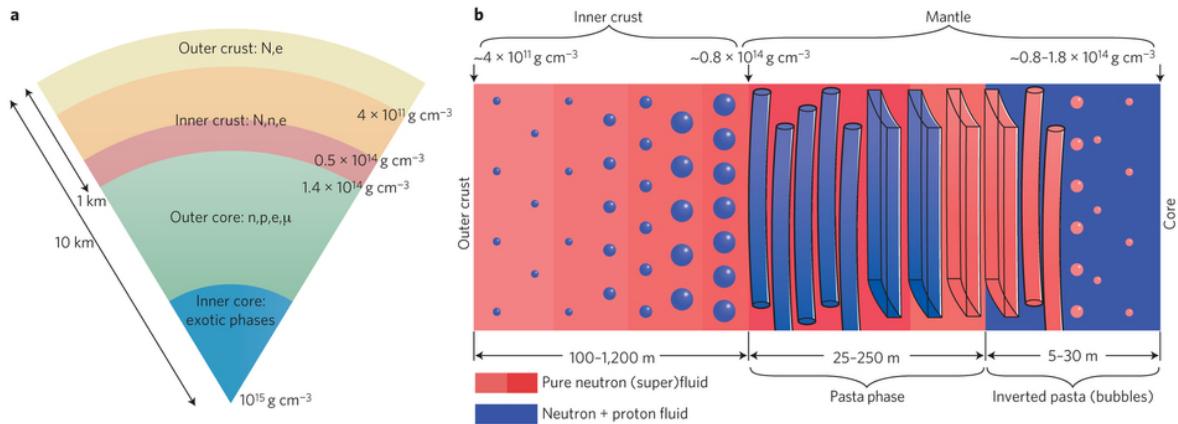


Fig 1.2: Neutron Star Structure

Atmosphere: The outermost layer, the atmosphere, represents an ultra-thin plasma region surrounding the neutron star. Despite its negligible mass, it plays a pivotal role in defining the star's radiation. Here, the thermal electromagnetic radiation spectrum begins to take shape, giving rise to important characteristics such as beaming, polarization, and thermal properties. This spectrum carries valuable information about the neutron star's surface properties, including its effective surface temperature, surface gravity, chemical composition, and magnetic field strength and configuration. Moreover, it offers insights into macroscopic attributes such as the star's mass and radius. The thickness of the atmosphere is highly variable and contingent upon the neutron star's temperature. For hot neutron stars, with surface temperatures around $3 \cdot 10^6 K$, this layer measures approximately 10 cm. In contrast, colder neutron stars, with surface temperatures of roughly $3 \cdot 10^5 K$, have atmospheres merely a millimeter thick. Under extraordinary circumstances, as seen in very cold or intensely magnetized neutron stars, the surface may even manifest as a solid or liquid.

Outer Crust: Just beneath the atmosphere lies the outer crust, which extends to depths where the density reaches approximately $\rho = \rho_{ND} \approx 4 \cdot 10^{11} g cm^{-3}$. This layer, spanning several hundred meters, is primarily composed of ions and electrons. As we venture deeper into this region, electrons transition into a highly degenerate state, approximating an ideal gas. At extreme densities, these electrons become ultra-relativistic. Electron pressure dominates and leads to

complete ionization of atoms when the density surpasses approximately 10^4 g cm^{-3} . In these dense conditions, ions may form a strongly coupled Coulomb system, oscillating between liquid and solid states. Notably, a significant portion of the envelope can solidify, hence its colloquial name, the "crust". As density increases, the electron Fermi energy escalates, inducing beta captures in atomic nuclei and an increase in neutron content. At the base of the outer crust, neutrons begin to escape from nuclei, forming a free neutron gas (hence the term "neutron drip", denoted as "ND")

Inner Crust: Moving inward from the outer crust, we enter the inner crust, with a thickness of roughly one kilometer. The density here varies from ρ_{ND} at its upper boundary to approximately $0.5\rho_0$ at the base. In this layer, the constituents include electrons, free neutrons (n), and neutron-rich atomic nuclei. As density increases, the fraction of free neutrons grows. Notably, neutronization at around $\rho \approx \rho_{ND}$ significantly softens the equation of state (EOS). At the bottom layers of the inner crust, which range from roughly $1/3\rho_0$ to approximately $1/2\rho_0$, atomic nuclei may become non-spherical and form a "mantle." However, the specifics of this structure can vary depending on the model. As we reach the crust-core interface, nuclei cease to exist, and free neutrons and nucleons remain, possibly in a superfluid state.

Outer Core: Progressing further toward the center, we encounter the outer core, which occupies the density range between $0.5\rho_0$ and $2\rho_0$ and spans several kilometers. The matter in this region is primarily composed of neutrons, with a small admixture of protons (p), electrons, and potentially muons (μ), known as the "npe μ composition". This composition adheres to electric neutrality and beta equilibrium, governed by the intricate dynamics of many-body nucleon interactions. Here, electrons and muons behave as nearly ideal Fermi gases, while neutrons and protons, interacting via the strong nuclear force, constitute a strongly interactive Fermi liquid. This layer can also host superfluid states among its components.

Inner Core: At the heart of the neutron star lies the inner core, characterized by a density exceeding $2\rho_0$. The central density can reach staggering values of up to $10 - 15\rho_0$. The exact composition and equation of state in the inner core remain the subject of intense scientific scrutiny and model-dependent speculations. Various hypotheses have emerged, postulating the presence of exotic particles such as hyperons (e.g., Σ and Λ), pion condensates, kaon condensates, or even deconfined quark matter. These scenarios can trigger a range of phase transitions, leading to the softening of the EOS. The existence of mixed phases of dense matter has not been ruled out, further underscoring the complexity of neutron star interiors.

1.2.3 Types of Neutron Stars

Based on data from observations, theoretical physics and computational modeling, neutron stars can be categorized as non-hypothetical (supported by observational evidence) and hypothetical (speculative constructs)[13, 7]. This thesis is primarily concerned with the first type, but for the sake of completeness, the subcategories and characteristics of both types are briefly analyzed in the following section.

Non-Hypothetical Neutron Stars: The most commonly observed and well-understood neutron stars are Standard Neutron Stars (SNS), also known as canonical neutron stars. They typically possess masses around 1.4 times that of the Sun ($1.4M_\odot$) and radii of approximately 10 kilometers. Their interior structures consist of the previously discussed regions: the outer crust, inner crust, outer core, and inner core. These neutron stars are primarily composed of

neutrons, protons, electrons, and possibly muons, with matter in highly degenerate states. SNSs can exhibit various surface temperatures, magnetic field strengths, and rotation rates, resulting in diverse observational characteristics.

A subset of standard neutron stars, Pulsars, are distinguished by their emission of regular, highly directional beams of radiation. These beams sweep across the line of sight of Earth observers, resulting in periodic pulses of radiation. Pulsars are often observed in radio, X-ray, and gamma-ray wavelengths. Their precise rotational periods make them invaluable tools for astrophysical studies. Another class of standard neutron stars are Magnetars. They are characterized by extraordinarily strong magnetic fields, reaching strengths of up to 10^{15} Gauss (for reference, the Earth's magnetic field is about 1 Gauss). These intense magnetic fields give rise to dramatic and sporadic X-ray and gamma-ray outbursts, making magnetars crucial for the study of extreme magnetic field effects.

Hypothetical Neutron Stars: The three most major hypothetical cases of neutron stars are Hybrid Stars, Quark Stars and Strange Stars. Hybrid neutron stars are hypothetical configurations that may contain exotic matter, such as hyperons or Bose condensates, within their inner cores. These components are currently the subject of theoretical speculation and are not yet confirmed to exist. If present, they could significantly affect the equation of state and the overall structure of the neutron star.

The scenario of Quark stars suggests that the inner core of a neutron star may transition into quark matter under extreme conditions. These exotic states of matter are composed of deconfined quarks and are purely theoretical at this point. Quark stars, if they exist, would challenge our understanding of nuclear physics and compact objects.

Finally, in the case of Strange Stars, strange quark matter is the absolute ground state of matter. These stars, if they exist, would be composed almost entirely of strange quark matter and could exhibit unique observational features.

1.3 Scientific History of Neutron Stars

The scientific journey to uncover the mysteries of neutron stars has been marked by theoretical innovations and serendipitous discoveries. The inception of this remarkable narrative can be traced back to the early 1930s when the theoretical groundwork was laid [13].

In 1932, Landau's seminal work on the maximum mass of compact stars set the stage for discussions on dense stellar objects. This was the same year that Chadwick discovered the neutron, laying the foundation for Landau's concept of "eerie stars," which would eventually evolve into the notion of neutron stars. The term "neutron stars" made its debut in 1933 through an abstract by Baade and Zwicky, who speculated that supernovae could give birth to these exotic entities. This concept motivated Oppenheimer and Volkoff to tackle the problem of neutron stars within the framework of general relativity. Their pioneering work culminated in the Tolman-Oppenheimer-Volkoff (TOV) equations, which predicted a maximum neutron star mass of 0.7 solar masses, a prescient result that remains relevant to this day.

Despite early theoretical endeavors, it was not until 1967 that the neutron star's existence was serendipitously revealed through the discovery of pulsars by Jocelyn Bell. These pulsating sources of radiation, later identified as rapidly rotating neutron stars, captured the attention of the scientific community and unleashed a torrent of research.

In the ensuing years, scientists such as Hewish, Harrison, Wheeler, and Cameron made significant strides in understanding neutron star properties. The development of a unified equation of state (EOS) connecting white dwarfs to neutron star cores, spearheaded by Harrison and Wheeler, was instrumental in our comprehension of these dense celestial bodies. Cameron's incorporation of nuclear interactions resulted in a maximum mass prediction of 2.0 solar masses at a radius of 8.1 km, heralding an era of heightened interest in neutron star physics.

Further investigations by Salpeter, Ambartsumyan, and Saakyan highlighted the potential role of hyperons in dense neutron star matter, shedding light on the complex composition of these stellar remnants. Today, our modern understanding of neutron stars, from their outer crusts to the high-density core, can be traced back to these early endeavors. Today, however, we possess a new tool aiding us in neutron star related research. In recent years, a groundbreaking chapter in the scientific history of neutron stars has unfolded with the advent of gravitational wave astronomy. This revolutionary field has allowed researchers to observe the coalescence of compact binary systems, which include neutron stars. Notably, the landmark detection of the binary neutron star merger GW170817 in 2017, by the Laser Interferometer Gravitational-Wave Observatory (LIGO) and Virgo collaborations, marked a watershed moment. The cataclysmic collision not only emitted gravitational waves but also generated a wealth of multi-messenger data, including electromagnetic signals such as gamma-ray bursts and kilonovae. This historic event offered a multifaceted view into the nature of neutron stars, confirming their existence as well as providing crucial insights into their masses, radii, and the properties of dense nuclear matter. These findings have reinvigorated neutron star research, igniting a new era of exploration where gravitational waves serve as a powerful tool for unraveling the secrets of these enigmatic cosmic entities.

1.4 Binary Neutron Stars and Tidal Deformability

It is possible for neutron stars to exist in binaries (BNS systems). These binaries consist of two neutron stars orbiting around their common center of mass. The scientific exploration of neutron star binaries has become increasingly significant due to the information they provide about neutron star properties, the equation of state (EOS) of ultra-dense matter, and the broader field of astrophysics. These binaries can consist of a variety of different star pairs, like Double Neutron Star Binaries (DNS), Neutron Star-White Dwarf Binaries(NS-WD), Low or High-Mass X-ray Binaries etc. Since this thesis is concerned with such systems, the following section is dedicated to presenting some basic yet substantial information about them.

1.4.1 Binary Neutron Star Mergers

Neutron star binaries are formed through various mechanisms. One common formation channel is the evolution of massive binary star systems. When one of the stars in such a binary undergoes a supernova explosion, it can leave behind a neutron star. If the binary system survives the supernova event, it can evolve into a neutron star binary. The dynamical behavior of neutron star binaries is characterized by several stages [12]:

I. Inspiral Phase (Before the Merger): This is the initial stage of a BNS system. The two neutron stars slowly spiral towards each other due to gravitational forces. During this phase, the inspiral motion is relatively slow and can last for millions of years. Non-linear effects are

generally not significant in this stage, making it easier to model. In the case of unequal mass BNS systems, the more massive neutron star tries to absorb its companion's mass, leading to mass transfer and eventually merger.

II. Merger: As the neutron stars approach each other closely, tidal effects become important. The gravitational forces between the stars cause them to merge, resulting in the formation of a single, more massive object. The merger stage is characterized by intense physical processes, including the Kelvin-Helmholtz instability, which can lead to complex structures in the matter around the merging stars. The result of this phase is practically a collision between the two stars.

III. Post-Merger Stages: Depending on the masses of the stars and other factors, several outcomes are possible after the merger. Some of the most notable ones are the following:

IIIa. Hypermassive Neutron Star: If the combined mass of the binary system is high but not high enough to immediately form a black hole, the merger product becomes a hypermassive neutron star. This stage is temporary and may last up to around 10 milliseconds. The support against gravitational collapse in this case is the neutron star's rapid rotation. This is typically the result of an equal mass BNS system.

IIIb. Black Hole Formation: Depending on various parameters and the evolution of the hypermassive neutron star, it can either collapse into a black hole due to gravitational instability or lose angular momentum over time and become a non-rotating neutron star. This leads to the formation of either a black hole-torus system or a non-rotating neutron star.

IIIc. Black Hole-Torus System: After the merger, when a black hole forms (typically for unequal high-mass systems), there is often matter scattered around it. This matter orbits the black hole in a stable toroidal shape, known as an accretion torus, which eventually gets absorbed into the black hole. The presence and characteristics of this torus can vary based on the system's parameters.

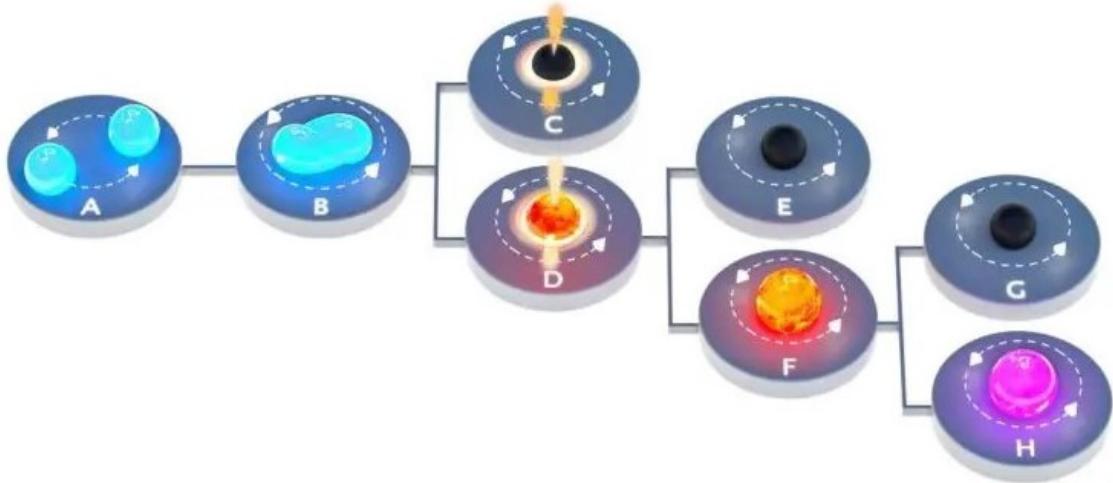


Fig 1.3: BNS Stages

1.4.2 Gravitational Waves and Tidal Deformability

When neutron stars orbit each other in a binary system and eventually merge, as explained above in more detail, they emit gravitational waves [13, 7]. These waves carry away energy and angular momentum from the binary, causing the neutron stars to spiral closer together. In recent years, the detection of gravitational waves from BNS mergers has provided valuable data for understanding the behavior of matter at extreme densities. These detections have contributed significantly to our knowledge of neutron star properties and have the potential to refine our understanding of the EOS of nuclear matter under extreme conditions.

Tidal deformability, a key concept in the study of neutron star binaries, arises from the tidal interaction between the two neutron stars as they get closer during the inspiral phase. As the neutron stars approach each other, their gravitational fields induce tidal forces, leading to a deformation in the shape of each star. The effect of tidal deformability on the gravitational wave signal is quantified using a parameter known as the tidal Love number (k_2). This dimensionless constant contains critical information about the internal structure and EOS of neutron stars. It has been shown that the tidal coupling between neutron stars is accessible to measurement by gravitational-wave detectors like LIGO and Virgo.

Tidal deformability provides several key pieces of information. The most significant insight gained is related to the equation of state (EOS) of neutron stars. The EOS describes the relationship between the pressure, density, and composition of the matter within a neutron star. By measuring the tidal deformability, constraints can be imposed upon the EOS, revealing whether neutron star matter is primarily composed of ordinary atomic nuclei, exotic quark matter, or other forms of dense nuclear matter. This knowledge is essential for understanding the fundamental properties of neutron stars.

Tidal deformability also offers insights into the internal structure of neutron stars. Different EOS models predict distinct levels of deformability, reflecting variations in the stiffness of the neutron star's core. Measuring the tidal deformability allows us to probe the composition and behavior of matter under extreme conditions, deep within the neutron star.

The degree of tidal deformability is moreover directly related to the neutron star's radius. By quantifying tidal deformability, neutron stars' sizes estimations can be made, providing crucial information for astrophysical models and comparisons with observational data.

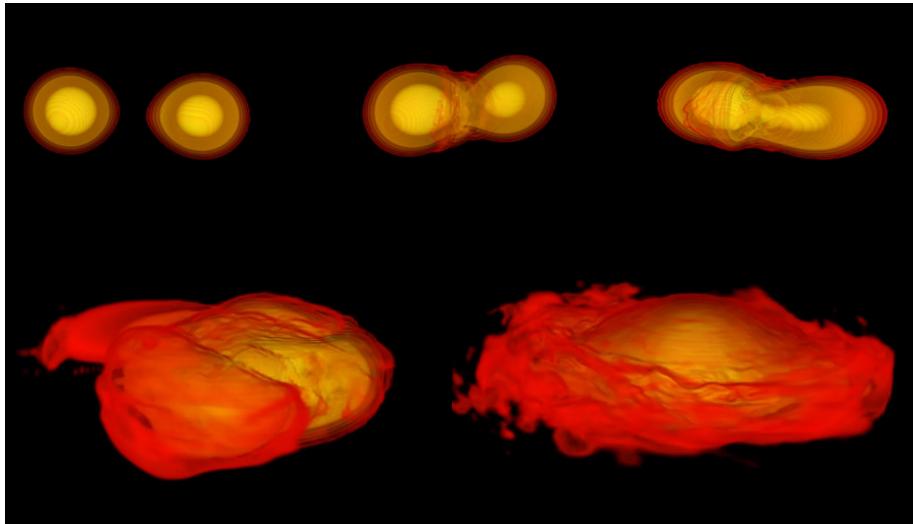


Fig 1.4: BNS merger simulation

Chapter 2

Mathematical Framework

2.1 TOV Equations

To unveil the fundamental characteristics of a neutron star, we start by assuming it possesses a spherically symmetric mass distribution in a state of hydrostatic equilibrium, while being in an extremely cold condition ($T = 0$) [5]. This analysis intentionally omits the influence of rotation and magnetic fields. The equilibrium configurations of neutron stars are derived by solving the Tolman–Oppenheimer–Volkoff (TOV) equations, which represent the gravitational equilibrium of neutron star matter under the principles of general relativity. These equations, extend the concepts of Newtonian gravity into the realm of general relativity and read [5, 11]:

$$\begin{aligned} \frac{dP(r)}{dr} &= -\frac{GM(r)\rho(r)}{r^2} \left(1 + \frac{P(r)}{c^2\rho(r)}\right) \left(1 + \frac{4\pi r^3 P(r)}{c^2 M(r)}\right) \\ &\quad \times \left(1 - \frac{2GM(r)^{-1}}{c^2 r}\right) \\ \frac{dM(r)}{dr} &= 4\pi r^2 \rho(r), \end{aligned} \tag{2.1}$$

where:

$$\rho(r) = \frac{\mathcal{E}(r)}{c^2}. \tag{2.2}$$

The first of the two is the Hydrostatic Equilibrium equation. In essence, it describes the balance between gravitational attraction and pressure within the star. The second one links the mass of the star within a given radius to the radius itself.

Solving the set of equations 2.1, for $P(r)$ and $M(r)$, entails numerical integration. This integration proceeds outward from the star's center ($r = 0$) to the surface, where pressure is equal to zero, delineating the star's outer boundary at $r = R$. The choice of the initial central pressure, denoted as $P_c = P(r = 0)$ is critical and the resultant values of R and the total mass of the neutron star, designated as $M \equiv M(R)$ depend on it. For this integration process to proceed effectively, knowledge of the energy density $\mathcal{E}(r)$ in terms of the pressure $P(r)$ is essential. This correlation defines the equation of state (EOS) for the matter constituting the neutron star.

Eqs.2.1 can be modified appropriately for numerical integration as follows [5]:

$$\begin{aligned}\frac{d\bar{P}(r)}{dr} &= -1.474 \frac{\bar{\mathcal{E}}(r)\bar{M}(r)}{r^2} \left(1 + \frac{\bar{P}(r)}{\bar{\mathcal{E}}(r)}\right) \left(1 + 11.2 \times 10^{-6} r^3 \frac{\bar{P}(r)}{\bar{M}(r)}\right) \\ &\quad \times \left(1 - 2.948 \frac{\bar{M}(r)}{r}\right)^{-1} \\ \frac{d\bar{M}(r)}{dr} &= 11.2 \times 10^{-6} r^2 \bar{\mathcal{E}}(r),\end{aligned}\tag{2.3}$$

where:

$$\begin{aligned}M(r) &= \bar{M}(r)M_{\odot}, \\ \mathcal{E}(r) &= \bar{\mathcal{E}}(r)\mathcal{E}_0, \\ P(r) &= \bar{P}(r)\mathcal{E}_0 \\ \mathcal{E}_0 &= 1 \text{MeV} \cdot \text{fm}^{-3}, \\ \frac{GM_{\odot}}{c^2} &= 1.474 \text{Km}, \\ \frac{4\pi}{M_{\odot}c^2} &= 0.7 \cdot 10^{-40} \frac{\text{s}^2}{\text{Kg} \cdot \text{Km}^2}.\end{aligned}\tag{2.4}$$

The result of this process gives us the mass of the star as [5]:

$$\bar{M}(R) = 11.2 \times 10^{-6} \int_0^R r^2 \bar{\mathcal{E}}(r) dr = b_0 \int \bar{\mathcal{E}}(r) dr\tag{2.5}$$

2.2 Equations of State

The equation of state (EOS) serves as the cornerstone in constructing models of neutron stars. It fundamentally delineates the relationship between pressure (P) and mass density (ρ) or equivalently, energy density (\mathcal{E}), along with the temperature of the stellar matter. In the context of neutron stars, which predominantly consist of strongly degenerate fermions, the temperature's influence is often negligible, allowing for the EOS to be predominantly assessed at an ultracold temperature of $T = 0K$. Neutron star model EOSs can be broadly categorized into two distinct types:

Stiff EOS: These models exhibit a significant increase in pressure as density rises. The matter described by such an EOS is characterized by low compressibility, endowing it with heightened resistance against gravitational collapse.

Soft EOS: In contrast, soft EOS models showcase a more gradual increase in pressure with increasing density, contrasting with the rapid ascent witnessed in stiff EOS cases. Consequently, matter governed by a soft EOS possesses a higher degree of compressibility.

The stiffness affects the M-R relation and the maximum possible mass of the neutron star that it describes. Specifically, for a given mass, a soft equation of state implies smaller radius and a lower value of maximum mass than those that a stiff equation of state suggests. Regarding the form of the equation, typically, the EOS is written as a sum of two distinct contributions:

(a) one for symmetric matter, that is having equal number of neutrons and protons, and (b) one for the symmetry energy to account for deviation from the symmetric limit.

It was explained that the EOS we need for the process we are following has to be a function of pressure, and more specifically in the form of:

$$P = P(\mathcal{E}) \quad (2.6)$$

which is called a barotropic equation of state. Because of the specific conditions and the purpose of our study, it is permissible to use equations of state in the form of:

$$\mathcal{E} = \mathcal{E}(P) \quad (2.7)$$

Pressure (P) is in units of $MeV \cdot (fm)^{-3}$. The chosen equations, all in form of 2.7 are listed below [14]:

$$E(P) = 4.1844P^{0.81449} + 95.00135P^{0.31736} \quad (\text{MDI-1})$$

$$E(P) = 5.97365P^{0.77374} + 89.24P^{0.30993} \quad (\text{MDI-2})$$

$$E(P) = 15.55P^{0.666} + 76.71P^{0.247} \quad (\text{MDI-3})$$

$$E(P) = 25.99587P^{0.61209} + 65.62193P^{0.15512} \quad (\text{MDI-4})$$

$$E(P) = 119.05736 + 304.80445(1 - e^{-P^{48.61465}}) + 33722.34448(1 - e^{-P^{17499.47411}}) \quad (\text{NLD})$$

$$E(P) = 1.78429P^{0.93761} + 106.93652P^{0.31715} \quad (\text{HHJ-1})$$

$$E(P) = 1.18961P^{0.96539} + 108.40302P^{0.31264} \quad (\text{HHJ-2})$$

$$E(P) = 0.53928P^{1.01394} + 94.31452P^{0.35135} \quad (\text{Ska})$$

$$E(P) = 4.75668P^{0.76537} + 105.722P^{0.2745} \quad (\text{SkI4})$$

$$E(P) = 172.858(1 - e^{-P^{22.8644}}) + 2777.75(1 - e^{-P^{1909.97}}) + 161.553 \quad (\text{HLPS-2})$$

$$E(P) = 131.811(1 - e^{-P^{4.41577}}) + 924.143(1 - e^{-P^{523.736}}) + 81.5682 \quad (\text{HLPS-3})$$

$$E(P) = 0.371414P^{1.08004} + 109.258P^{0.351019} \quad (\text{SCVBB})$$

$$E(P) = 0.00127717P^{1.69617} + 135.233P^{0.331471} \quad (\text{WFF-1})$$

$$E(P) = 0.00244523P^{1.62692} + 122.076P^{0.340401} \quad (\text{WFF-2})$$

$$E(P) = 0.261822P^{1.16851} + 92.4893P^{0.307728} \quad (\text{W})$$

$$E(P) = 0.0112475P^{1.59689} + 102.302P^{0.335526} \quad (\text{BGP})$$

$$E(P) = 0.488686P^{1.01457} + 102.26P^{0.355095} \quad (\text{BL-1})$$

$$E(P) = 1.34241P^{0.910079} + 100.756P^{0.354129} \quad (\text{BL-2})$$

$$E(P) = 39.5021P^{0.541485} + 96.0528P^{0.00401285} \quad (\text{DH})$$

$$E(P) = 0.000719964P^{1.85898} + 108.975P^{0.340074} \quad (\text{APR-1})$$

We also have to take into consideration the existence of the crust. As it was explained in the introduction section,(1.2), the neutron star's crust affects the EOS because it introduces complex variations in composition, density and pressure due to the presence of heavy nuclei, degenerate electron gas, and the transition region to the inner crust. It is crucial that all these effects are taken into account in order for the modeling to be accurate and for the equations of state to correctly predict the M-R relationship and other NS properties. In our study, this intricacy has been dealt with by partially modifying each applied EOS accordingly for four consecutive pressure intervals in the crust region. The crust-core limit was set at $P = 0.696\text{MeV} \cdot (\text{fm})^{-3}$. The system of equations for this region reads [14]:

$$\begin{aligned} P &\in (9.34375 \cdot 10^{-5}, 0.184] : \\ \mathcal{E}(P) &= 0.00873 + 103.17338 \left(1 - e^{\frac{-P}{0.38527}}\right) + 7.34979 \left(1 - e^{\frac{-P}{0.01211}}\right) \end{aligned} \quad (2.8)$$

$$\begin{aligned} P &\in (4.1725 \cdot 10^{-8}, 9.34375 \cdot 10^{-5}] : \\ \mathcal{E}(P) &= 0.00015 + 0.00203 \left(1 - e^{-P \cdot 344827.5}\right) + 0.10851 \left(1 - e^{-P \cdot 7692.3076}\right) \end{aligned} \quad (2.9)$$

$$\begin{aligned} P &\in (1.44875 \cdot 10^{-11}, 4.1725 \cdot 10^{-8}] : \\ \mathcal{E}(P) &= 0.0000051 \left(1 - e^{-P \cdot 0.2373 \cdot 10^{10}}\right) + 0.00014 \left(1 - e^{-P \cdot 0.4020 \cdot 10^8}\right) \end{aligned} \quad (2.10)$$

$$P < 1.44875 \cdot 10^{-11} : \\ \mathcal{E}(P) = 10^{c_0 + c_1 + c_2 + c_3 + c_4 + c_5} \quad (2.11)$$

with:

$$c_0 = 31.93753, \quad c_1 = 10.82611 \cdot \log_{10}(P), \quad c_2 = 1.29312 \cdot \log_{10}(P)^2, \\ c_3 = 0.08014 \cdot \log_{10}(P)^3, \quad c_4 = 0.00242 \cdot \log_{10}(P)^4, \quad c_5 = 0.000028 \cdot \log_{10}(P)^5$$

2.3 Tidal Parameters

As it was explained in more detail in 1.4.2, gravitational waves can be utilised to measure the tidal parameters of a BNS during the inspiral phase [8, 6], especially in the moments leading to the merger stage. At these moments, as the frequency increases, the tidal parameters become important. Both bodies of the system are affected, acquiring a tidal deformation in their shape which is linked to the structure and properties of the star. So, these directly measurable quantities (through GW measurements) give us access to information about the stars' compactness and their EOS. In this section, we lay the basic mathematical framework required for the understanding and studying of these tidal effects. We note, that as a convention we have set $G = c = 1$

For a static, spherically symmetric star of mass M , in a static external quadrupolar tidal field \mathcal{E}_{ij} we get the following description [6, 9, 8]; The tidal field induces a quadrupole moment Q_{ij} upon the stars. This response is quantified by the tidal parameters k_2 , known as the tidal Love number and λ which is called tidal deformability or polarizability [9]. To linear order, we define the tidal deformability relating the star's induced quadrupole moment to the external tidal field as:

$$Q_{ij} = -\lambda \mathcal{E}_{ij} \quad (2.12)$$

Tidal deformability λ is related to the $l = 2$ tidal Love number k_2 by

$$k_2 = \frac{3}{2} G \lambda R^{-5} \quad (2.13)$$

It should be noted that the tidal Love number k_2 is a dimensionless quantity.

The quadrupole moment Q_{ij} of the star and the external tidal field \mathcal{E}_{ij} are characterized as coefficients in a far-field expansion of the total metric around the star at large distances r [6, 8]. In this expansion, the metric component g_{tt} in asymptotically Cartesian, mass-centered coordinates involves the standard gravitational potential m/r , along with two dominant terms resulting from the perturbation; one representing an external tidal field that increases with r^2 and another representing the tidal distortion that decreases with r^{-3} :

$$\frac{(1 - g_{tt})}{2} = -\frac{M}{r} - \frac{3Q_{ij}}{2r^3} \left(n^i n^j - \frac{1}{3} \delta^{ij} \right) + O \left(\frac{1}{r^3} \right) + \frac{1}{2} \mathcal{E}_{ij} x^i x^j + O(r^3) \quad (2.14)$$

where $n^i = x^i/r$. Q_{ij} and \mathcal{E}_{ij} are symmetric and traceless.

Through the process described in [6, 8, 10], we are led to the following expression for the calculation of the tidal Love number k_2 :

$$\begin{aligned} k_2(\beta, y_R) = & \frac{8\beta^5}{5}(1-2\beta)^2[2-y_R+2\beta(y_R-1)] \\ & \times [2\beta(6-3y_R+3\beta(5y_R-8)) \\ & + 4\beta^3(13-11y_R+\beta(3y_R-2)+2\beta^2(1+y_R)) \\ & + 3(1-2\beta)^2[2-y_R+2\beta(y_R-1)]\ln(1-2\beta)]^{-1} \end{aligned} \quad (2.15)$$

As shown in equation 2.15, k_2 is dependant upon two other parameters. The first one is the compactness parameter β . Essentially, it is a measure of how densely packed the mass of the star is within its volume. It is defined as:

$$\beta = GM/Rc^2 \quad (2.16)$$

The other one is y_R which is a parameter related to the tidal effects and used for their calculation. It is determined by solving the following differential equation [6, 8, 10]:

$$r \frac{dy(r)}{dr} + y^2(r) + y(r)F(r) + r^2Q(r) = 0 \quad (2.17)$$

with boundary conditions:

$$y(0) = 2, \quad y_R \equiv y(R) \quad (2.18)$$

$F(r)$ and $Q(r)$ are functionals of energy density $\mathcal{E}(r)$, pressure $P(r)$ and mass $M(r)$ given by [6, 8, 10]:

$$F(r) = \left[1 - \frac{4\pi r^2 G}{c^4} (\mathcal{E}(r) - P(r)) \right] \left(1 - \frac{2M(r)G}{rc^2} \right)^{-1} \quad (2.19)$$

and

$$\begin{aligned} r^2Q(r) = & \frac{4\pi r^2 G}{c^4} \left[5\mathcal{E}(r) + 9P(r) + \frac{\mathcal{E}(r) + P(r)}{\partial P(r)/\partial \mathcal{E}(r)} \right] \left(1 - \frac{2M(r)G}{rc^2} \right)^{-1} \\ & - 6 \left(1 - \frac{2M(r)G}{rc^2} \right)^{-1} \\ & - \frac{4M^2(r)G^2}{r^2 c^4} \left(1 + \frac{4\pi r^3 P(r)}{M(r)c^2} \right)^2 \left(1 - \frac{2M(r)G}{rc^2} \right)^{-2} \end{aligned} \quad (2.20)$$

The derivative $dp/d\rho \equiv \partial P(r)/\partial \mathcal{E}(r)$ is the squared sound of speed $c_s^2(r)$ [sources]

From the way we defined the expressions for the tidal parameters (2.13, 2.15) the path that leads to their estimation becomes clear. Equation 2.17 must be integrated self-consistently and at the same time as the TOV equations (2.3) with boundary conditions:

$$y(0) = 2, P(0) = P_c$$

and

$$M(0) = 0$$

Mass M and radius R will be provided by the solution of the TOV system and the quantity y_R will be given by solving the differential equation 2.17. With M , R and y_R known, compactness β can also be calculated from 2.16 and consequently the tidal parameters k_2 and λ as well from 2.15 and 2.13 respectively.

For the numerical solution of the system of differential equations mentioned above, proper scaling needs to take place. We need to perform a mathematical transformations of these relations, as in section 2.2:

$$\begin{aligned}\bar{M}(r) &= \frac{M(r)}{M_\odot} \\ \bar{\mathcal{E}} &= \frac{\mathcal{E}}{\mathcal{E}_0} \\ \bar{P} &= \frac{P}{\mathcal{E}_0}\end{aligned}\tag{2.21}$$

From now on, energy density \mathcal{E} and pressure P are defined in $MeV \cdot fm^{-3}$ units, mass M in M_\odot (solar masses) and radius R in km. \mathcal{E}_0 is equal to $1MeV \cdot fm^{-3}$. Other quantities used include:

$$\begin{aligned}\frac{GM_\odot}{c^2} &= 1.474 \text{ km} \\ \frac{4\pi}{M_\odot c^2} &= 0.7 \cdot 10^{-40} \frac{s^2}{kg \cdot km^2} \\ \frac{4\pi\mathcal{E}_0}{M_\odot c^2} &= 11.2 \cdot 10^{-6} \frac{1}{km^3} \\ R_s &= \frac{2GM}{c^2} = \frac{2GM_\odot}{c^2} \bar{M} = 2.948 \bar{M} (\text{ km})\end{aligned}\tag{2.22}$$

Now, eqs. 2.19 and 2.20 can be rewritten as:

$$F(r) = \left[1 - 1.474 \cdot 11.2 \cdot 10^{-6} \cdot r^2 (\epsilon(r) - P(r)) \right] \left(1 - 2.948 \frac{M(r)}{r} \right)^{-1}\tag{2.23}$$

and:

$$\begin{aligned}r^2 Q(r) &= 1.474 \cdot 11.2 \cdot 10^{-6} \cdot r^2 \left[5\epsilon(r) + 9P(r) + \frac{\epsilon(r) + P(r)}{\partial P(r)/\partial \epsilon(r)} \right] \left(1 - 2.948 \frac{M(r)}{r} \right)^{-1} \\ &\quad - 6 \left(1 - 2.948 \frac{M(r)}{r} \right)^{-1} \\ &\quad - \frac{4 \cdot 1.474^2 M^2(r)}{r^2} \left(1 + 11.2 \cdot 10^{-6} r^3 \frac{P(r)}{M(r)} \right)^2 \left(1 - 2.948 \frac{M(r)}{r} \right)^{-2}\end{aligned}\tag{2.24}$$

Lastly, compactness (2.16) is transformed as:

$$\beta = \frac{GM(r)}{Rc^2} = 1.474 \frac{M}{R}\tag{2.25}$$

Chapter 3

Computational Analysis and Results

Having laid out the necessary theoretical and mathematical framework regarding, the next step of our study was computational analysis. Code we wrote in python was used for every step in our process. First, for every EOS in our set (from [MDI-1](#) to [APR-1](#)), the modified TOV system of equations [2.3](#) and the differential equation [2.18](#) were simultaneously numerically solved. In essence, numerical integration was performed begining from the center of the star at $r = 0$ moving outwards till $r = R$. Boundary conditions were set as: $P(0) = P_c$, $M(0) = 0$ and $y(0) = 2$ for the center of the star and: $P(R) = 0$, $M(R) = M$, $y(R) = y_R$ for the end of the surface. We note that we used extremely small values instead of 0, to avoid division by zero. The range of values for central pressure was $P_c = 1 - 1200 \text{ MeV} \cdot \text{fm}^{-3}$ and the crust was taken into account, as described in [2.3](#) by modifying parts of the EOS ([2.8-2.11](#)). Values from mass M and radius R are obtained from the solution of the TOV equations and subsequently, compactness parameter β can be calculated from [2.25](#). Substituting y_R , given by [2.18](#), and β in equation [2.15](#) gives the values for tidal Love number k_2 . As a result, tidal deformability λ can also be determined from [2.13](#). At the end of this process, the whole set of parameters needed for our analysis have been calculated. With this information, graphs were constructed to study the behavior of these parameters and statistical analysis was performed in search of a possible systematic relation.

3.1 Mass-Radius Relation

As previously explained, the modified system of TOV equations [2.2](#) is solved numerically for EOS [MDI-1](#) through [APR-1](#). More specifically, the integration starts from central pressure $P_c = 1 \text{ MeV} \cdot \text{fm}^{-3}$ up to $P_c = 1200 \text{ MeV} \cdot \text{fm}^{-3}$ (for other boundary conditions, values very close, but not equal to zero are used). Each EOS generates a set of M-R values for the star as a result that are represented as curves in the following graph [\(3.1\)](#). Mass is measured in solar masses (M_\odot) and radius in km [\(3.1\)](#).

As a first note, we see that the equations of state employed align seamlessly with known expected constraints regarding the mass of neutron stars. Each EOS predicts a maximum mass for the star and we can see that these equations consistently yield neutron star masses in the vicinity of approximately $2M_\odot$, reinforcing their compatibility with empirical limitations. However, values corresponding to the area beyond the maximum mass were produced as well. In [3.2](#) those values were cut in order to have a more accurate representation of the area that interests us.

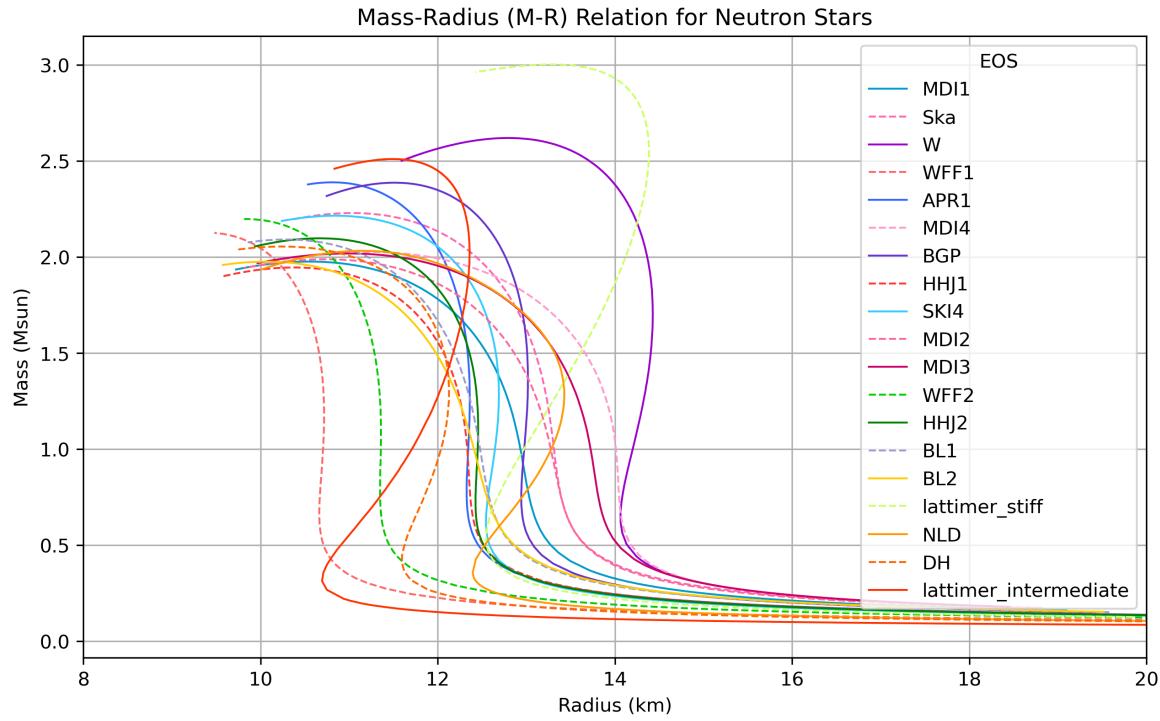


Fig. 3.1: M-R diagram

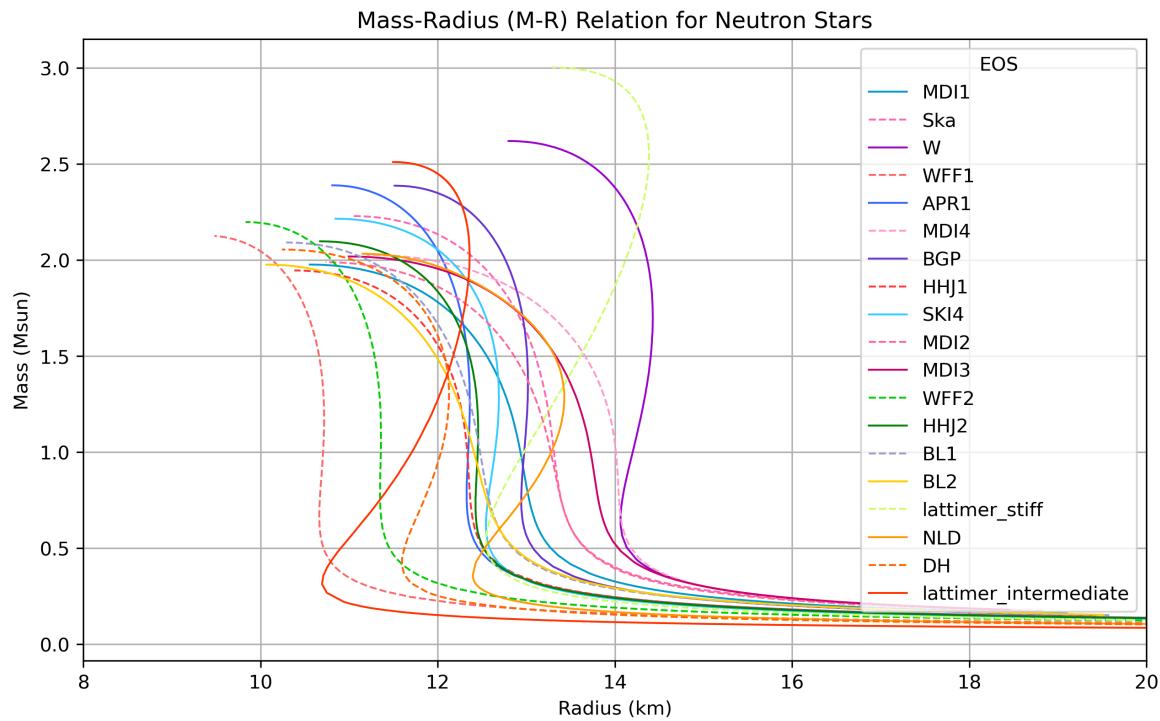


Fig. 3.2: M-R diagram

It is also clear, that for large values of radii R (approximately $R > 15\text{km}$), the mass M of the star tends to 0, which is reasonable taking into account the characteristics of Neutron Stars talked about in chapter 1. The most important observation to make is in regard to the different results of M and R values that stiff and soft equations of state yield. Stiffer EOS predict generally larger values both for the mass M and the radius R of the star. In contrast, softer EOS predict in general smaller values for these quantities. This behavior becomes clear when we take a look at the most extreme cases in graph 3.2. The stiffest of the equations of state that we used, [HLPS-3](#) (also called lattimer-stiff) and even [W](#) predict maximum masses of $3M_{\odot}$ and $2.6M_{\odot}$ respectively. On the other hand, the much softer [DH](#), reproduces a maximum mass of less than $2M_{\odot}$.

A table (3.3) was also constructed showcasing the values for the maximum mass, M_{max} , the radius corresponding to the maximum mass, $R(M_{max})$ and the radius corresponding to $1.4M_{\odot}$, which is considered to be the most probable value for a NS mass, according to many models.

EOS	$M_{max} (M_{\odot})$	$R(M_{max})$	$R(1.4M_{\odot}) (\text{km})$
MDI1	1.976194	10.556958	12.671
Ska	2.228849	11.035806	13.167
W	2.61987	12.8	14.378
WFF1	2.12574	9.479508	10.694
APR1	2.388921	10.810529	12.355
MDI4	2.026944	11.202	13.778
BGP	2.387047	11.513942	13.018
HHJ1	1.945446	10.381881	12.159
SKI4	2.214984	10.847	12.681
MDI2	1.988299	10.735	12.99
MDI3	2.018318	10.996	13.456
WFF2	2.197732	9.831483	11.32
HHJ2	2.097361	10.672908	12.407
BL1	2.091228	10.290629	12.272
BL2	1.975839	10.067	12.103
lattimer_stiff (HLPS-3)	3.003023	13.291725	13.479
NLD	2.032309	11.156	13.396
DH	2.054601	10.246879	12.115
lattimer_intermediate (HLPS-2)	2.510308	11.500784	12.109

Fig. 3.3: M_{max} , $R(M_{max})$, $R(1.4M_{\odot})$ table for each EOS

3.2 Tidal Effects

Continuing the process described in the previous section and simultaneously solving the system for the tidal parameters, we present the results for the Love number k_2 , the tidal deformability λ and y_r . As before, the EOS used were [MDI-1](#) through [APR-1](#) and the crust of the star was taken into account by modifying the equations accordingly as explained in section 2.3

3.2.1 Tidal Love number k_2

The first parameter presented is the tidal Love number k_2 . In order to study its behavior, plots were constructed to demonstrate the quantity's relation to the Mass M , (Fig. 3.4), Radius R (Fig. 3.5) and Compactness parameter β (Fig. 3.6) of the Neutron Star for the whole range of EOS mentioned. Like before, mass is measured in M_\odot , radius in km, β in M_\odot/km and k_2 is dimensionless.

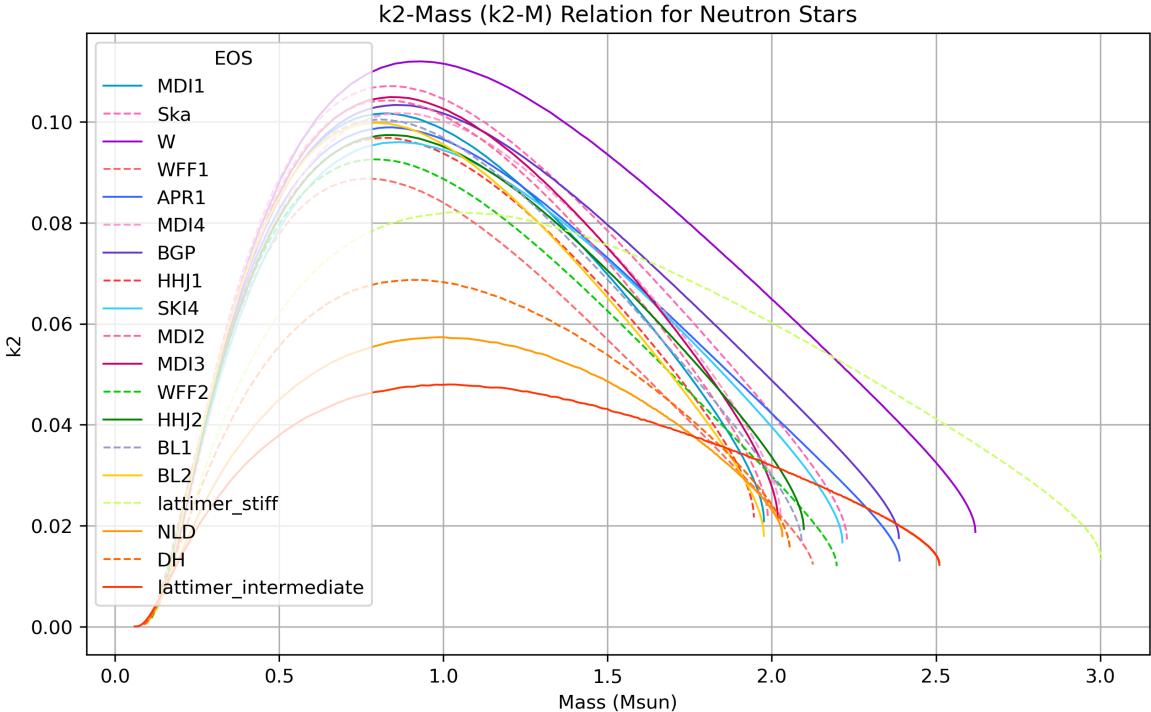


Fig. 3.4: $k_2 - M$ diagram

In Fig. 3.4, k_2 is given as a function of M . The Love number serves as a parameter that characterizes a star's response to tidal forces. Essentially, it quantifies how readily the star's mass distribution deforms when subjected to tidal effects. In simpler terms, a higher k_2 value indicates a greater degree of deformation or stretching in the star caused by tidal forces. By observing the plot, again a distinction in behavior can be made based on the stiffness of the EOS. Stiffer equations (e.g. **W**) produce higher values for k_2 , while softer equations (e.g. **HLPS-2** (lattimer-intermediate)) produce lower values.

In Fig. 3.5, we showcase the relation between k_2 and the radius R of a neutron star. We observe a consistent pattern: Neutron stars described by stiffer equations of state tend to have larger radii, while softer equations of state result in smaller radii (e.g. stiff **W** in comparison with softer **HLPS-2**). Although the range of k_2 values doesn't exhibit much variation across all EOS models, the extent of quadrupole response λ varies significantly as will be shown in the next subsection 3.2.2 .

Lastly, in Fig. 3.6, the plot of k_2 against the compactness parameter β , grants similar results to Fig. 3.4. We observe a decrease in value for k_2 with the increase of β . As compactness takes smaller values, tending to $\beta = 0.5$ (limit for black holes) [3] k_2 tends to 0, as expected from 2.15

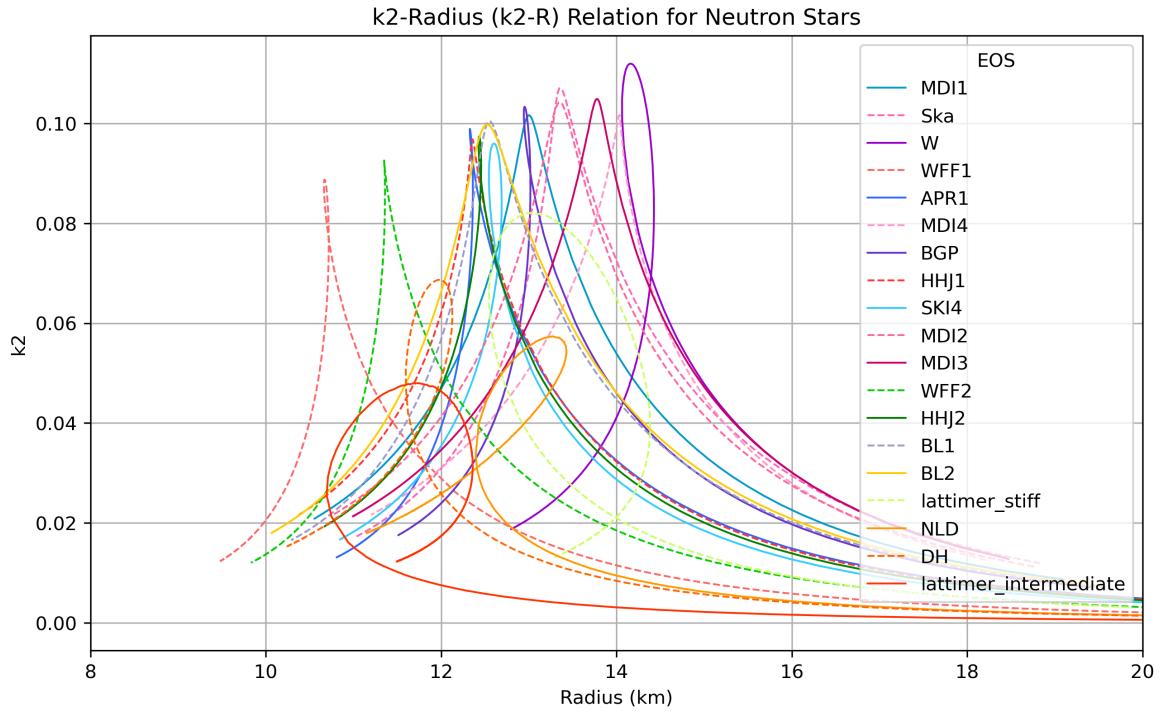


Fig. 3.5: $k_2 - R$ diagram

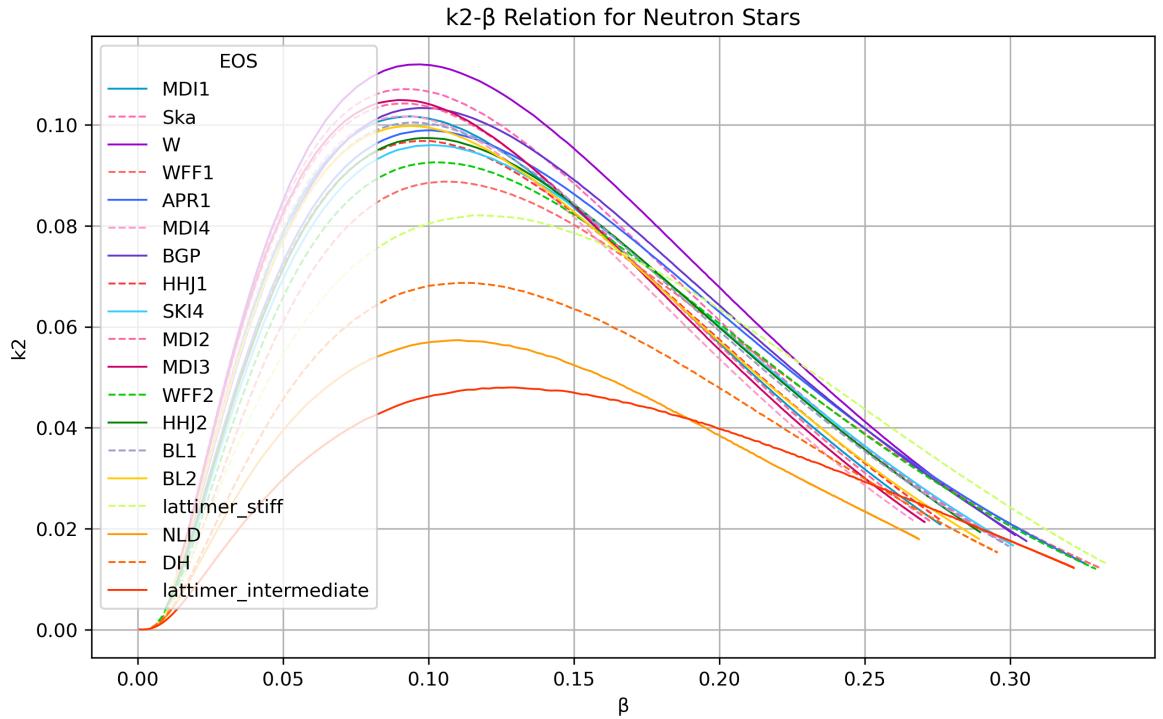


Fig. 3.6: $k_2 - \beta$ diagram

3.2.2 Tidal Deformability λ

Tidal deformability λ , as discussed in 1.4.2 is the result of tidal forces induced by the neutron stars getting closer and closer to each other during the inspiral phase [8, 13]. It quantifies the deformation in the shape of the stars, but also offers insights related to the EOS, structure and size of the stars. The quantity's importance is also amplified by the fact that it can directly be measured through the observation of gravitational waves emitted during the inspiral of a BNS. Such detections provide a means to accurately determine λ . Below, λ is plotted against mass M , radius R and compactness parameter β for the same variety of EOS (MDI-1 through APR-1) in order to establish its relation to each of these quantities.

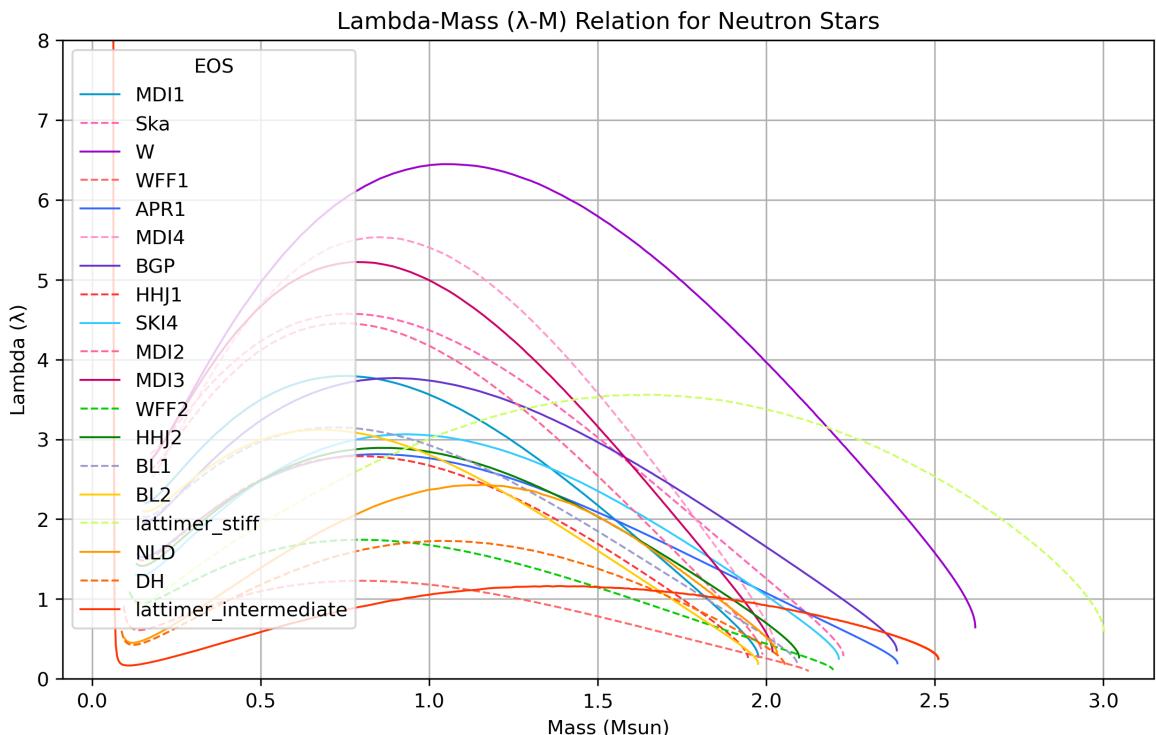


Fig. 3.7: $\lambda - M$ diagram

First, in Fig. 3.7, the $\lambda - M$ relation is showcased. It is evident that the λ parameter exhibits a significantly broader range of values across different Equation of State models (from $\lambda = 0 - 6$) [8, 10]. This variation arises from the fact that λ is directly proportional to the product of k_2 and R^5 (2.13). More specifically, EOS models that predict a greater maximum mass (so, stiffer equations, like **W** or **MDI-4**) for the neutron star, also generally result in larger values for the star's radius. This, in turn, leads to higher values for the tidal deformability.

This strong dependence of λ to radius R is more evident in Fig. 3.8. As expected, from equation 2.13 for an increase in radius R , the result is an even sharper increase in the value of tidal deformability λ .

Finally, λ is presented as a function of β in Fig. 3.9. The graph produced is very similar to the $\lambda - M$ plot, which is expected, given that the compactness parameter is given by $\beta = M/R$. Higher values of compactness align with higher values of tidal deformability.

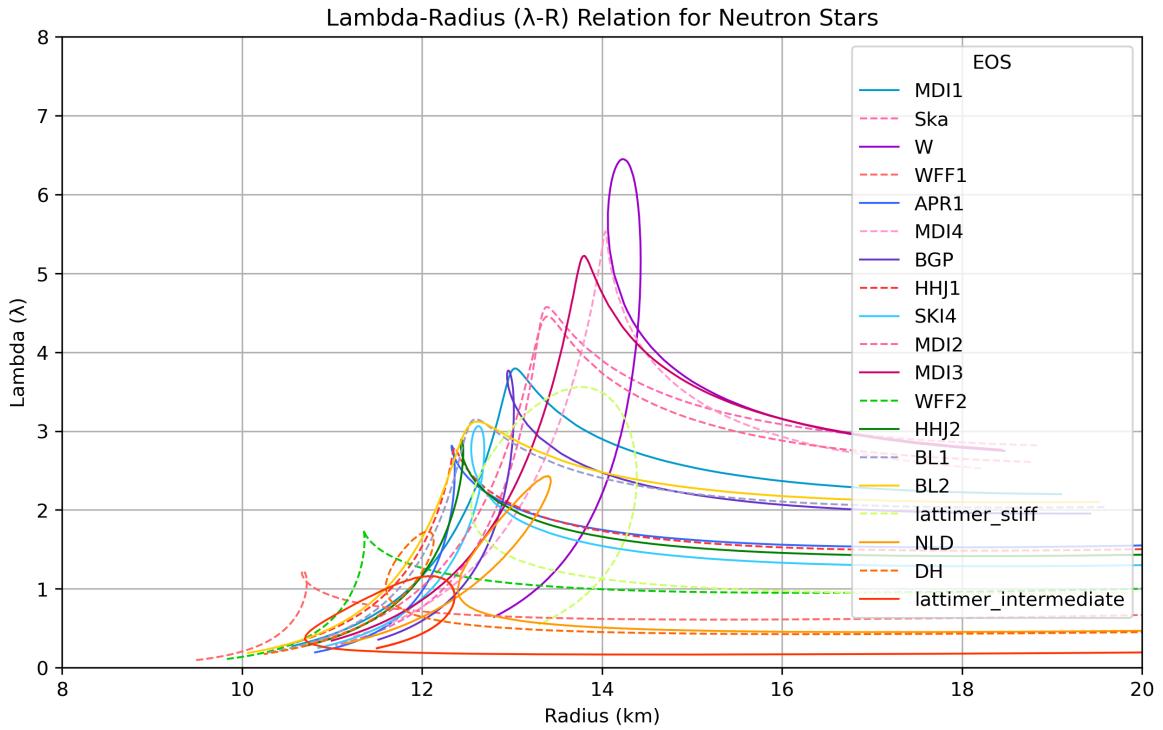


Fig. 3.8: $\lambda - R$ diagram

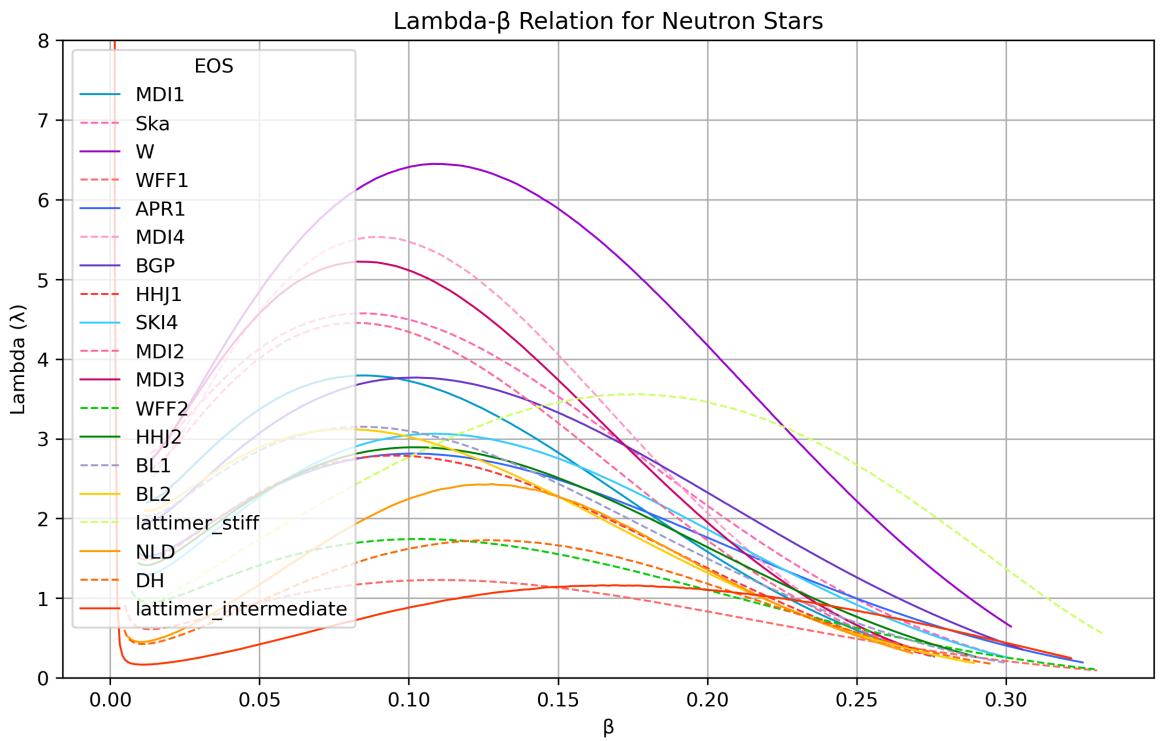


Fig. 3.9: $\lambda - \beta$ diagram

3.2.3 y_R

The last tidal parameter we studied is y_R . The results for the parameter's behavior with respect to mass M , radius R and compactness parameter β of the neutron star are presented in the following graphs. As always, a variety of EOS was tested ([MDI-1](#) through [APR-1](#)).

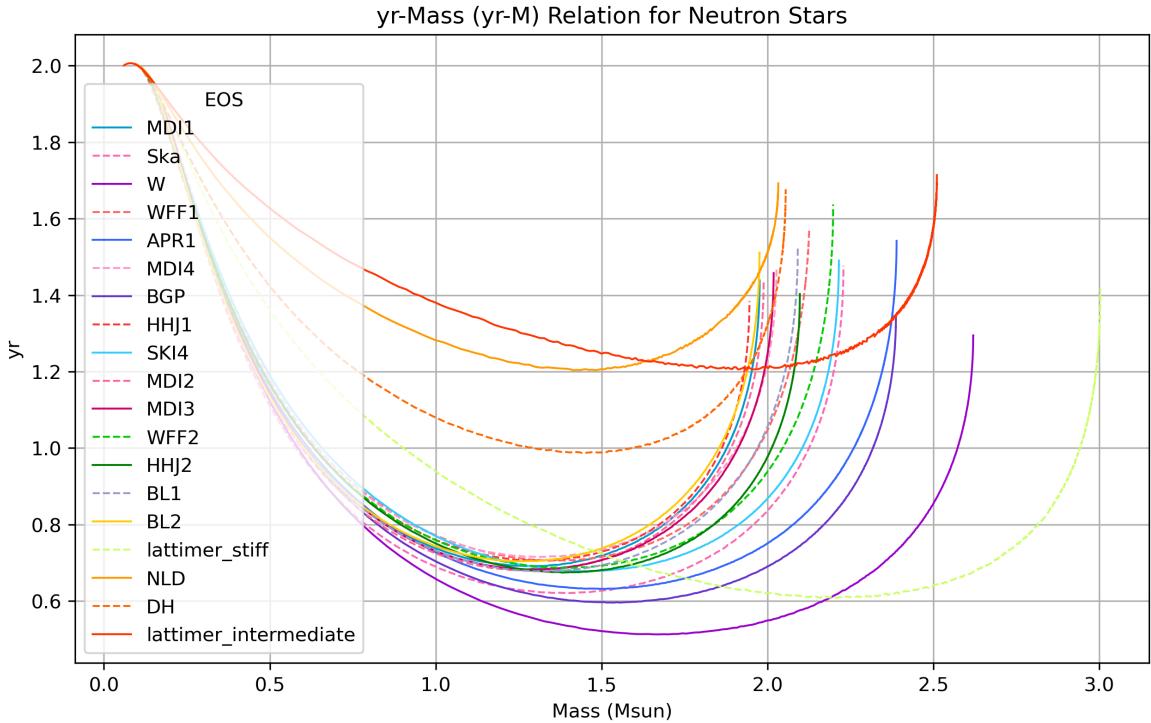


Fig. 3.10: $y_R - M$ diagram

The first thing we can note observing Fig. 3.10 is in regard to the stiffness of the EOS. We can deduce that stiffer equations of state, that generally predict larger mass values, (like [W](#), [BGP](#)) are linked to lower values of the y_R parameter (lowest at $y_R \approx 0.4 - 0.6$), while the softer equations (like [HLPS-2](#), [NLD](#)) correspond to much higher values ($\approx 1.2 - 1.3$ for the value of $1.4M_\odot$). Additionally, all EOSs exhibit a shared point $y_R = 2$ at $M = 0$. This is in line with the boundary condition set in section 2.4

In Fig. 3.11, a similar pattern is observed. Larger radii R correspond to smaller values of y_R and vice versa. This makes sense if we take into account all of our previous observations, since larger radii are generally predicted by stiffer models (as well as bigger maximum masses).

Expectedly, the same behavior is exhibited in Fig. 3.12, in which y_R is presented as a function of β .

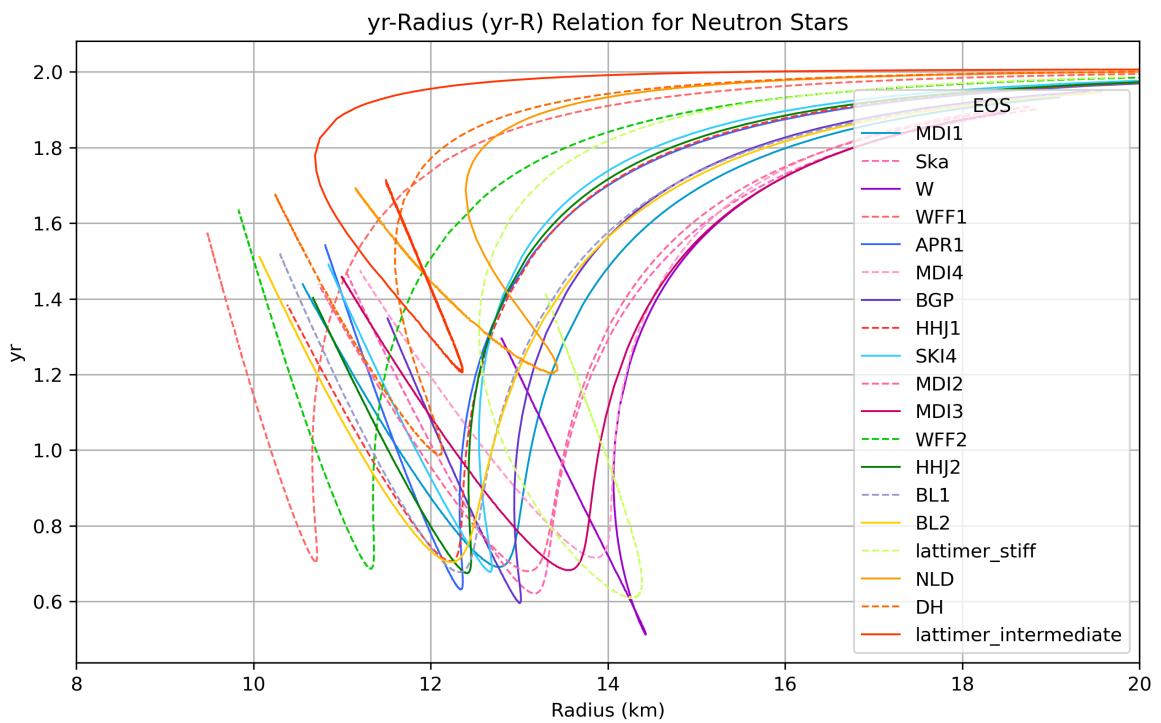


Fig. 3.11: $y_R - R$ diagram

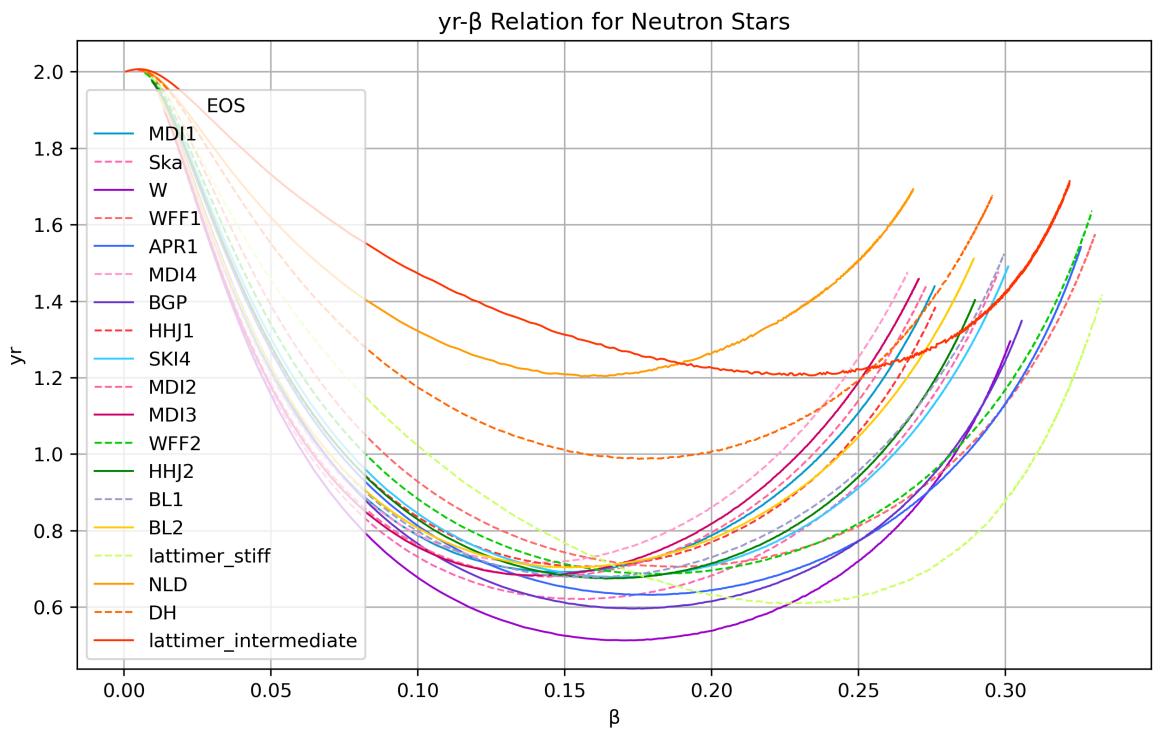


Fig. 3.12: $y_R - M$ diagram

3.3 Statistical Analysis

As a last step in the computational part of this study, we conducted a statistical analysis concerning the quantities of tidal deformability λ , dimensionless tidal deformability Λ , the tidal Love number k_2 and their relation to the Neutron Star's Radius R corresponding to a Mass value of $1.4M_\odot$ ($R(1.4M_\odot)$). We note that dimensionless tidal deformability is defined as: $\Lambda = \lambda/M^5$. As a reminder, the value of 1.4 solar masses is considered the most probable and common mass for a Neutron Star. Using our python code, before anything else, we computed $R(1.4M_\odot)$, as well as the corresponding k_2 , λ and Λ values for each equation of state (from [MDI-1](#) through [APR-1](#)). The results are demonstrated in Fig. 3.13.

EOS	$R(1.4M_\odot)$ (km)	k_2	λ	Λ
MDI1	12.671	0.077011	2.512634	0.469412
Ska	13.167	0.085126	3.365247	0.634381
W	14.378	0.098633	6.05392	1.141928
WFF1	10.694	0.063174	0.882599	0.164862
APR1	12.355	0.078651	2.261741	0.417152
MDI4	13.778	0.08157	4.04562	0.739985
BGP	13.018	0.085065	3.176832	0.584304
HHJ1	12.159	0.072847	1.933852	0.356211
SKI4	12.681	0.077939	2.552968	0.468251
MDI2	12.99	0.079779	2.947513	0.555037
MDI3	13.456	0.082147	3.619888	0.673405
WFF2	11.32	0.068652	1.274697	0.236154
HHJ2	12.407	0.076816	2.255832	0.42192
BL1	12.272	0.075385	2.095955	0.387039
BL2	12.103	0.072563	1.882353	0.351743
lattimer_stiff (HLPS-3)	13.479	0.078131	3.472447	0.665218
NLD	13.396	0.051405	2.215155	0.404503
DH	12.115	0.057858	1.50834	0.277762
lattimer_intermediate (HLPS-2)	12.109	0.044523	1.157831	0.211274

Fig. 3.13: Table for $R(1.4M_\odot)$, k_2 and λ for each EOS

Having obtained this data, we tested to see whether the pairs $R(1.4M_\odot)$ - k_2 , $R(1.4M_\odot)$ - λ and $R(1.4M_\odot)$ - Λ exhibit a systematic relation between them. The two methods used were Correlation Analysis and Linear Regression.

3.3.1 Correlation Analysis method

Correlation analysis involves calculating correlation coefficients to measure the strength and direction of the relationship between two variables. It provides a single number (correlation coefficient) that quantifies the degree of association between the variables. This analysis indicates whether there is a linear or monotonic relationship between the variables and the strength of that relationship. In this study, Pearson's correlation coefficient, Pearson's p value, Spearman's rank correlation coefficient and Spearman's p value were calculated for $R(1.4M_{\odot})-k_2$, $R(1.4M_{\odot})-\lambda$ and $R(1.4M_{\odot})-\Lambda$ by our program in order to establish if there is such a relation. For context, the conditions set are given in the table in 3.14. Alpha is the significance level, which was set to 0.05. Based on everything explained above, the results provided by our code are presented in figures 3.15 and 3.16.

Pearson's correlation coefficient	Pearson's p value	Linear Correlation
≥ 0.7	$< \alpha$	Strong & Positive
≥ 0.7	$\geq \alpha$	Not statistically Significant
$[0.3, 0.7)$	$< \alpha$	Moderate & Positive
$[0.3, 0.7)$	$\geq \alpha$	Not statistically Significant
< 0.3		Weak or None

Fig. 3.14: Intervals for Correlation Analysis interpretation

```

Pearson's correlation coefficient (R(1.4M⊕)-k2): 0.544204
Pearson's p-value (R(1.4M⊕)-k2): 0.016002
Spearman's rank correlation coefficient (R(1.4M⊕)-k2): 0.731579
Spearman's p-value (R(1.4M⊕)-k2): 0.000371
Pearson's correlation coefficient (R(1.4M⊕)-λ): 0.902636
Pearson's p-value (R(1.4M⊕)-λ): 0.000000
Spearman's rank correlation coefficient (R(1.4M⊕)-λ): 0.938596
Spearman's p-value (R(1.4M⊕)-λ): 0.000000
Interpretation for R(1.4M⊕)-k2: Moderate positive linear correlation
Interpretation for R(1.4M⊕)-λ: Strong positive linear correlation

```

Fig. 3.15: Correlation Analysis interpretation by python (k_2, λ)

```

Pearson's correlation coefficient (R(1.4M⊕)-Λ): 0.898276
Pearson's p-value (R(1.4M⊕)-Λ): 0.000000
Spearman's rank correlation coefficient (R(1.4M⊕)-Λ): 0.938596
Spearman's p-value (R(1.4M⊕)-Λ): 0.000000
Interpretation for R(1.4M⊕)-Λ: Strong positive linear correlation

```

Fig. 3.16: Correlation Analysis interpretation by python (Λ)

As shown above, the Correlation Analysis method predicts that there is a Moderate positive linear correlation between $R(1.4M_{\odot})$ and k_2 and a Strong positive linear correlation both between $R(1.4M_{\odot})$ and λ and between $R(1.4M_{\odot})$ and Λ . These predictions will be further investigated in the next subsection, using the Linear regression method.

3.3.2 Linear Regression method

A best-fit line, also known as a regression line or linear regression, is a line that represents the "best" linear relationship between two continuous variables. It is used to predict one variable (dependent variable) based on the other variable (independent variable). In simple linear regression, the goal is to find the equation of a straight line that minimizes the sum of the squared differences between the observed data points and the predicted values on that line. This line is represented by the equation: $y = mx + b$, where "y" is the dependent variable, "x" is the independent variable, "m" is the slope of the line, and "b" is the y-intercept. With the use of our python code, three plots were constructed with the data from Fig. 3.13 consisting of the pairs of $R(1.4M_{\odot})$ - k_2 , $R(1.4M_{\odot})$ - λ and $R(1.4M_{\odot})$ - Λ coordinates for each EOS. Following that, the linear regression method was applied and the best line was determined for each pair of parameters. The equation of the best line is displayed on top of the plots in Fig. 3.17, Fig. 3.18 and Fig. 3.19.

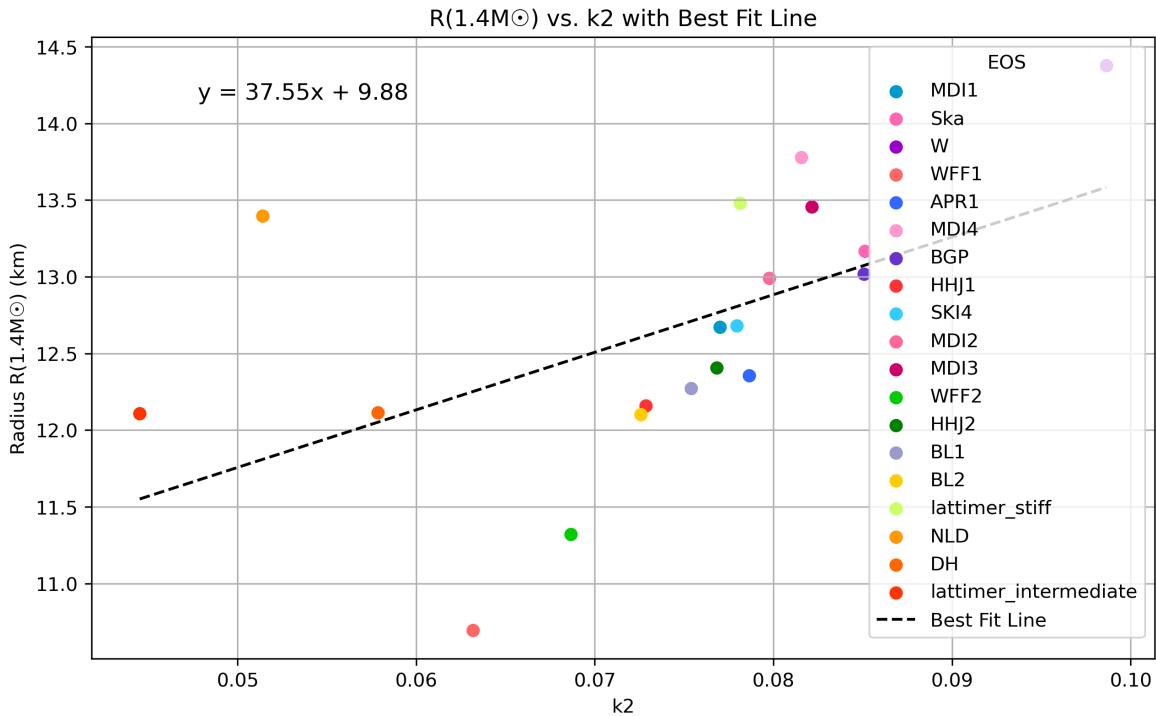


Fig. 3.17: $R(1.4M_{\odot})$) – k_2 plot with best line

The best lines predicted are $y = 37.55x + 9.88$, $y = 0.65x + 10.98$ and $y = 3.42x + 11.01$ for each pair respectively. We can observe a systematic relationship between the parameters, with the linear increase in $R(1.4M_{\odot})$ – λ and $R(1.4M_{\odot})$ – Λ being especially apparent. The results align with our findings in 3.3.1.

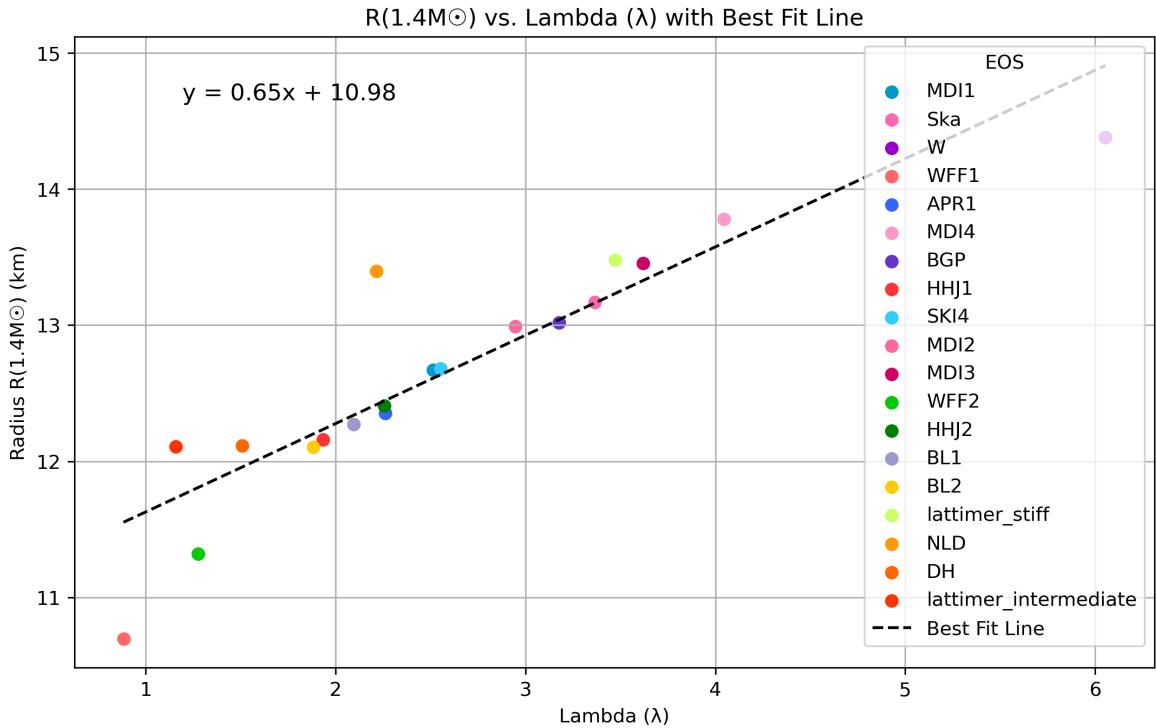


Fig. 3.18: $R(1.4M_{\odot})$ – λ plot with best line

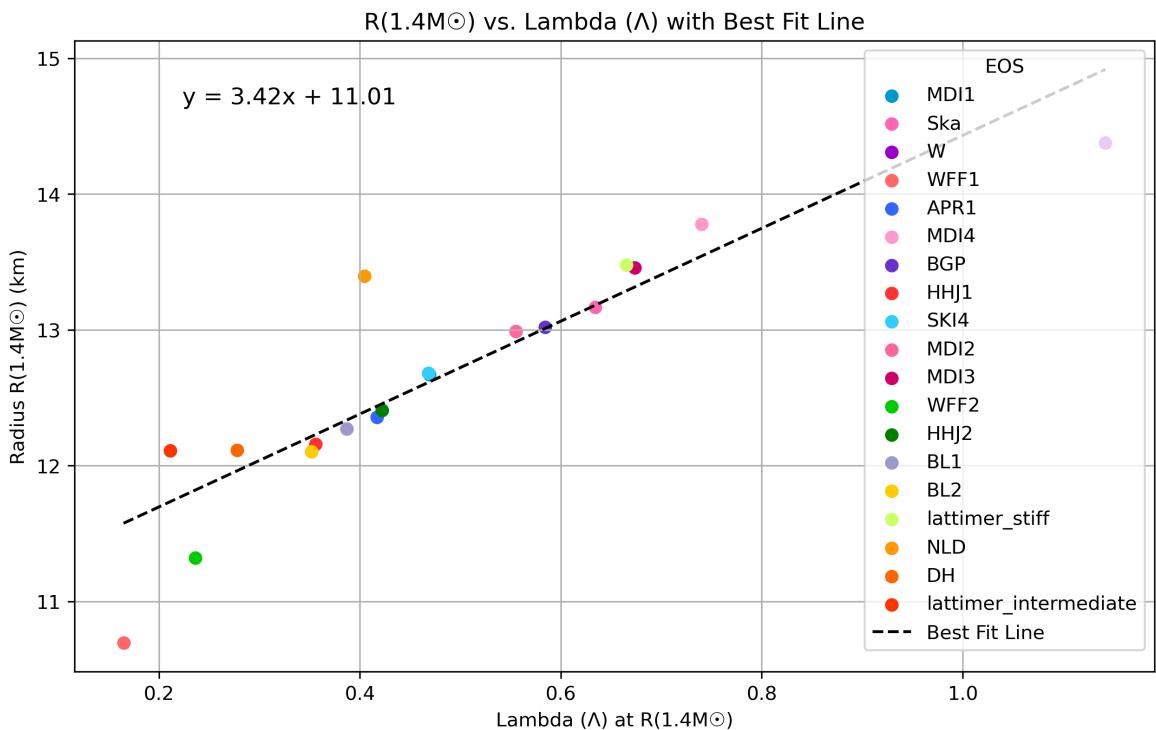


Fig. 3.19: $R(1.4M_{\odot})$ – Λ plot with best line

An interesting remark we could make is that the points corresponding to the softest EOS are the ones diverging the most from linearity. The same goes for the points corresponding to the stiffest EOS. This is just an observation regarding the behavior of the most extreme cases and does not imply anything about the validity of these models. For the sake of completeness, the best line with the softest and stiffest EOS removed is presented.

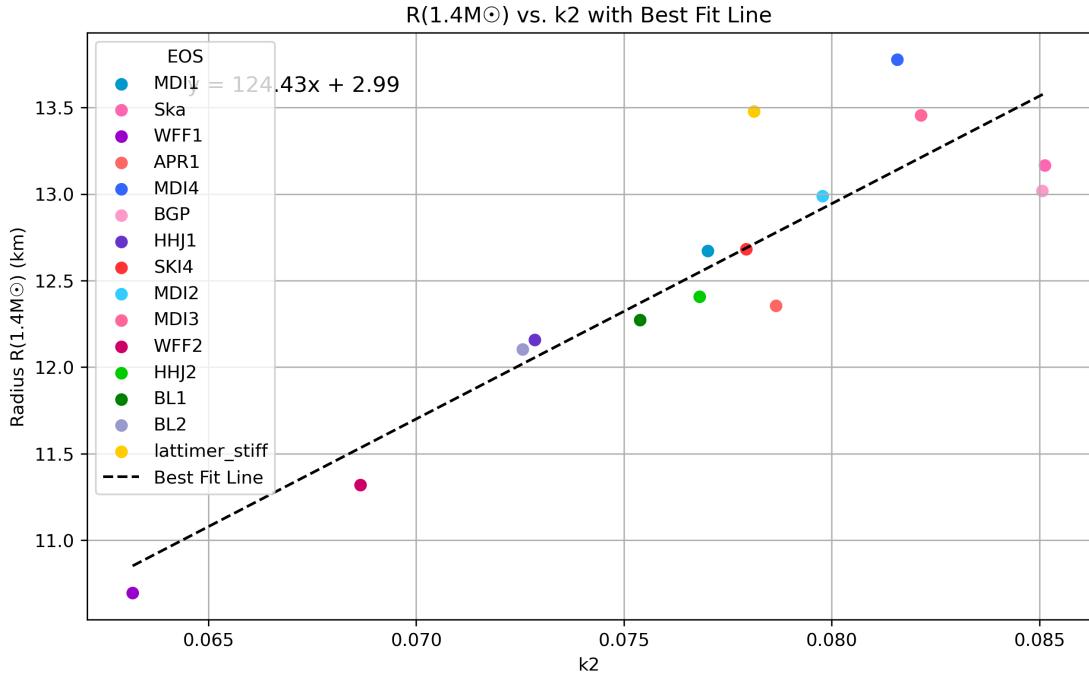


Fig. 3.20: $R(1.4M_\odot)) - k_2$ plot with best line without NLD, DH, HLPS-2 and W EOS

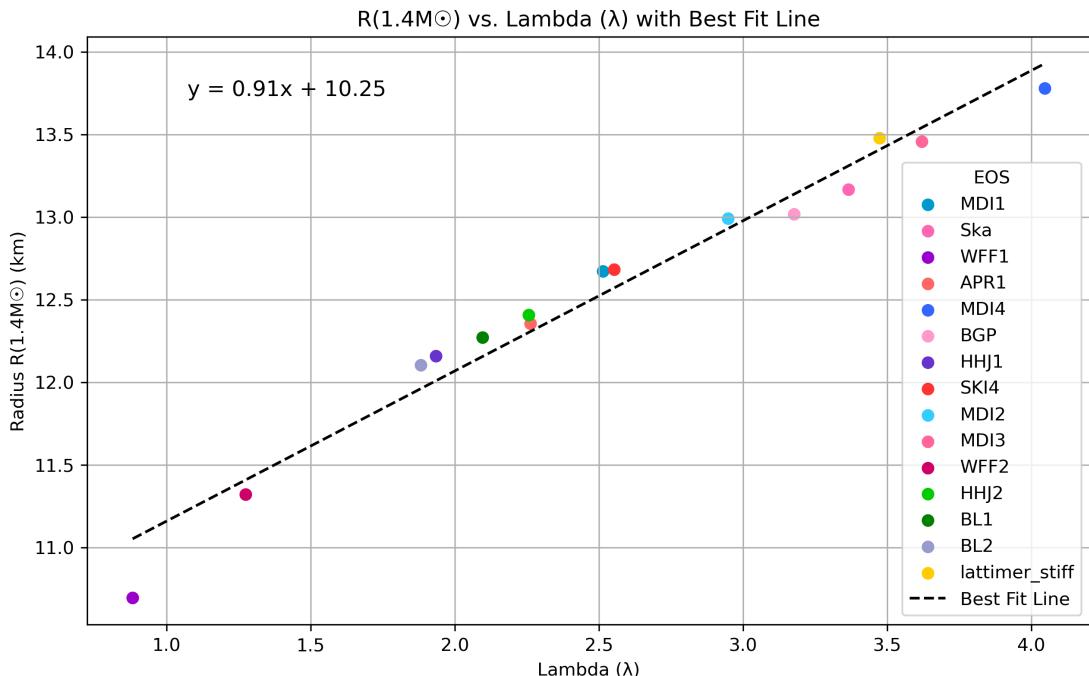


Fig. 3.21: $R(1.4M_\odot)) - \lambda$ plot with best line without NLD, DH, HLPS-2 and W EOS

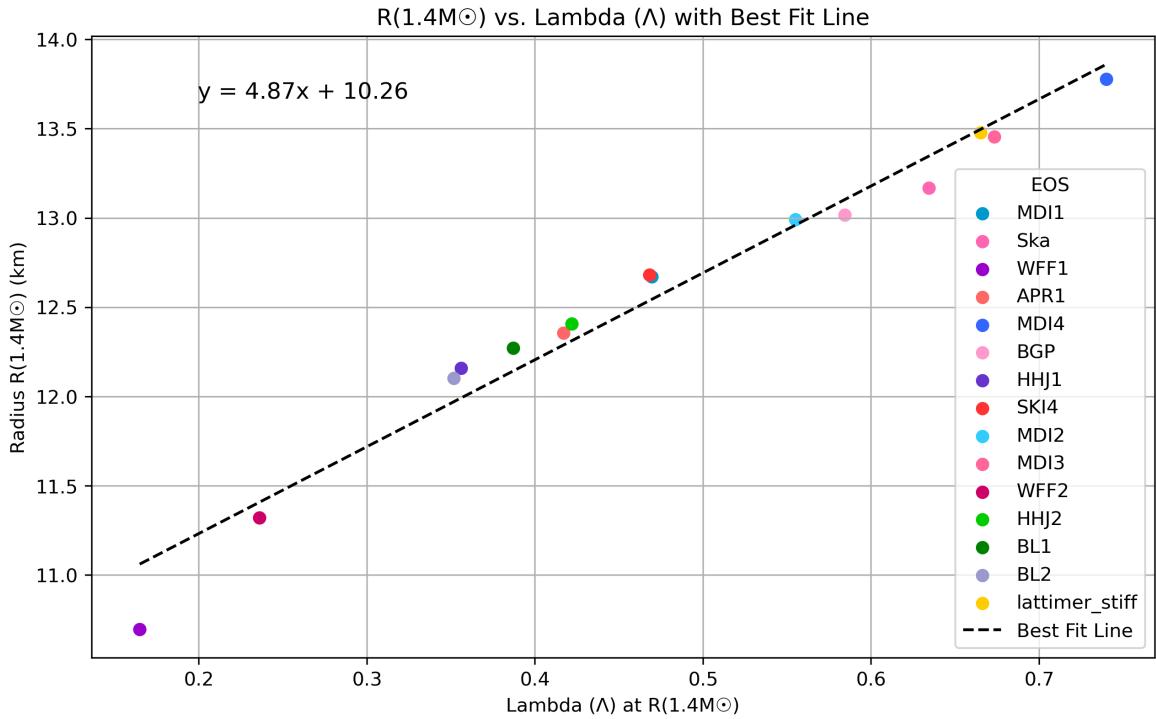


Fig. 3.22: $R(1.4M_\odot)$) – Λ plot with best line without NLD, DH, HLPS-2 and W EOS

3.4 Comparison to Observational Data

3.4.1 The BNS GW170817 event

On August 17, 2017, the joint network of gravitational wave detectors comprising LIGO and VIRGO made a significant observation [2, 1]. This event involved the detection of gravitational waves (GW) emitted during the inspiral of two compact objects, unequivocally identified as a binary neutron star system (BNS). The inclusion of VIRGO in this network played a crucial role in precisely determining the source’s location in the sky, employing triangulation-based techniques. By combining data from both LIGO and Virgo detectors, scientists were able to pinpoint the source’s sky position with remarkable accuracy, narrowing it down to an area spanning just 28 square degrees.

This newfound knowledge opened up a robust search for the optical and electromagnetic (EM) counterpart of this extraordinary celestial event in the days and weeks following its detection. Researchers identified a specific region of the sky that aligned with the one inferred from the gravitational wave signal. The process of detecting gravitational waves relies on a technique known as matched filtering, in which the raw data received from the detectors, taking into account the presence of noise, is compared to a preconceived theoretical waveform denoted as $h(t, \theta)$. This waveform depends on both time, denoted as t , and various parameters that describe the properties of the source, collectively referred to as θ .

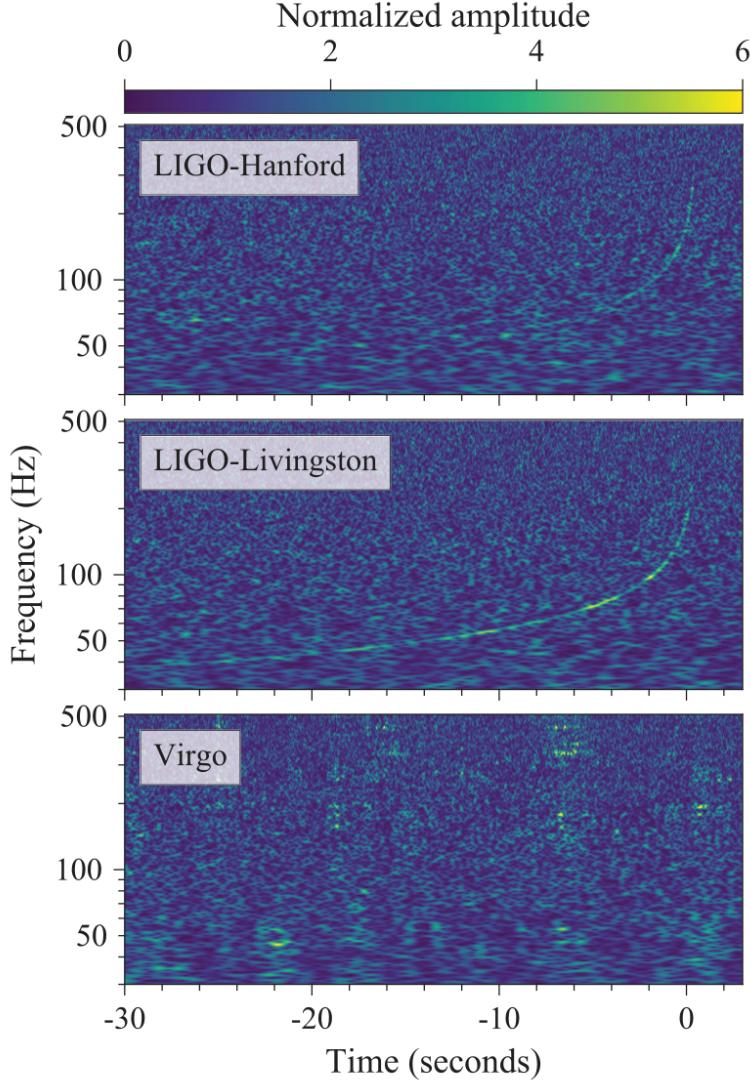


Fig. 3.23: The GW170817 signal as measured by the LIGO and Virgo gravitational wave detectors.

3.4.2 Results

Initially, our aim is to delve into the characteristics exhibited by the equations of state employed in our analysis. We are particularly interested in examining the estimated mass (M) and radius (R) within the marginalized contours, as outlined in [1]. We focused on the second method employed, namely the parameterized method. We projected the data provided from [1] to the graph (Fig. 3.2) produced by our computational analysis (3.1) for EOS MDI-1 through APR-1. The results are demonstrated in Fig. 3.24.

As explained above, this graph corresponds to the parameterized method with the constraint that the EOS should support neutron stars with masses larger than $1.97M_{\odot}$. The shaded area represents the posterior probability densities for both the mass (M) and radius (R) of each individual star in the binary neutron star system. The blue shading corresponds to the more massive component, while the orange shading represents the lighter one. The solid gray line delineates the 90% credible interval, while the dashed white line indicates the 50% credible

interval. The plotted curves on the diagram correspond to the equations of state utilized in our analysis.

This approach filters out excessively small radii R and predicts slightly larger radii. We observe that only two out of the complete set of 19 EOS applied, namely **W** and **MDI-4**, do not intersect with the marginalized contours. Notably, **MDI-4** just barely misses the contoured region. These two equations correspond to generally stiffer EOS, with **W** being the stiffest applied, since it predicts the largest M_{max} and R values. **MDI-4** might not be as stiff, but it still generates large R values. The rest of the 17 EOS fall into the credibility regions.

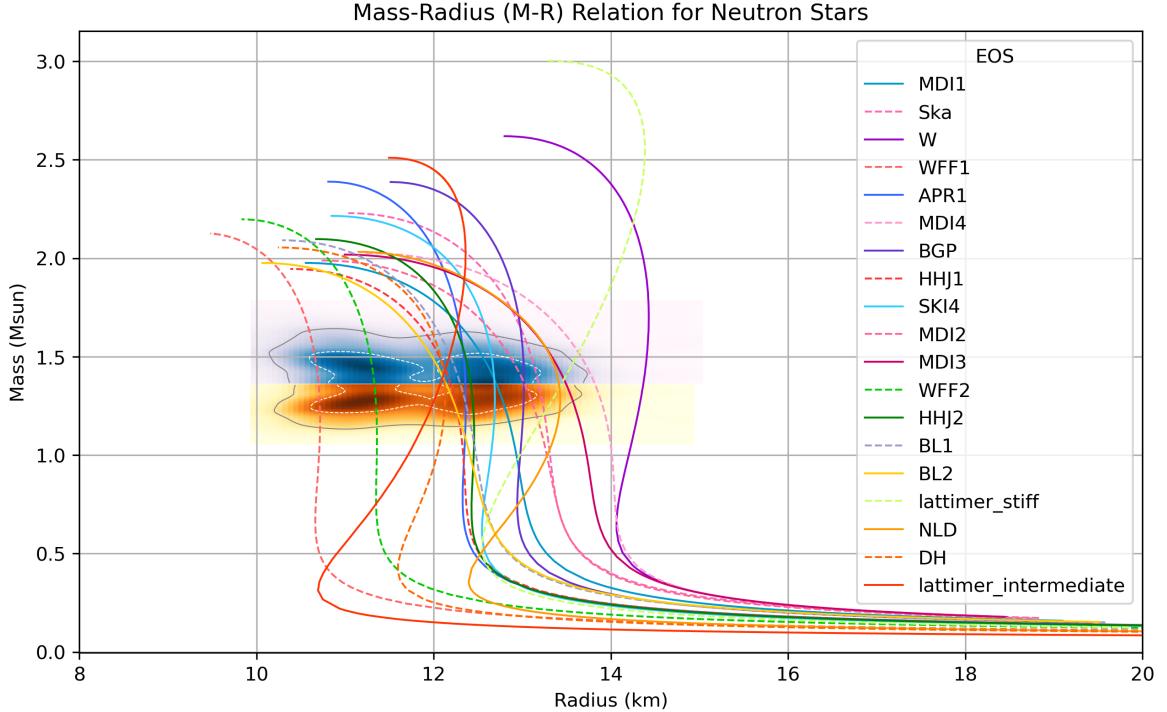


Fig. 3.24: M-R graph for the parameterized method (Maximum Mass constraint: $1.97M_\odot$)

Chapter 4

Concluding Remarks

The set of equations of state (EOS) utilized in our analysis are grounded in reality and align with the observational constraints placed on the upper limit of a neutron star's mass (M_{max}), as they successfully replicate neutron stars with a minimum mass of $\sim 2M_\odot$. Using code we wrote in python, through the numerical integration of the TOV equations, we derived the mass (M) and radius (R) of individual neutron stars for each EoS. Notably, our analysis of the M-R diagram revealed that the stiffer EoS generally produce larger values for both the mass (M) and the radius (R), unlike the softer ones

Furthermore, our investigation of the M-R diagram, in conjunction with posterior contours from LIGO estimations, using a parameterized method with a maximum mass restriction (at least $1.97M_\odot$) indicated that 2 of the stiffest EoS (out of the 19 applied), do not intersect the contours. This suggests the potential exclusion of these EoS.

Additionally, a comprehensive system of equations to compute the tidal parameters was established, extracting crucial insights into their behavior. More specifically, the dependence of the tidal Love number k_2 and tidal deformability λ to the mass M , radius R and compactness parameter β was examined. The stiffer EoSs were observed to yield larger values for the tidal parameters in contrast to the softer ones. Across all the EOS, λ exhibited a much larger range of values compared to k_2 . Moreover, the pronounced dependence of λ to R was confirmed.

Lastly, statistical analysis was performed utilizing the methods of Correlation Analysis and Linear Regression. Both methods showcased strong signs of a systematic increase relationship between both tidal deformability λ and dimensionless tidal deformability Λ with the value of radius at 1.4 solar masses ($R(1.4M_\odot)$). Such indication could prove of great importance, since, if further confirmed, this connection could be utilized as a way to make inferences about the properties, internal structure and equation of state of neutron stars based on directly observable and measurable quantities through GW, such as λ .

The era of multimessenger astronomy, though just starting, has already proven to be extremely enlightening in the study of Neutron Stars, their properties and equations of state. With focus on improvement of detectors, detection methods and theoretical models, the future looks bright for our understanding of these enigmatic celestial objects.

Chapter 5

Appendix: Python Codes

5.1 Eos Library

```
import numpy as np
from sympy import *

def lattimersoft (P):
    return - 4743.2711 * (P** (- 0.03313) )+ 4885.28224 * (P** (- 0.01027) )- 0.00535 * (P** 2.19938) + 0.01101 * (P** 2.09858)

def lattimerstiff (P):
    return 81.56823 + 131.81073 * (1-np. exp (-P/ 4.41577) )+ 924.14278 * (1-np. exp (-P/523.73573) )

def lattimerintermediate (P):
    return 161.55325 + 2777.74571 * (1-np. exp (-P/ 1909.97137) )+ 172.85805 * (1-np.exp (-P/ 22.86436) )

def nld(P):
    return 119.05736 + 304.80445 * (1-np. exp (-P/ 48.61465) )+ 33722.34448 * (1-np. exp(-P/ 17499.47411) )

def L80(P):
    return 5.97365 * (P** 0.77374) + 89.24 * (P** 0.30993)

def L95(P):
    return 15.54878 * (P** 0.66635) + 76.70887 * (P** 0.24734)

def sk14 (P):
    return 105.72158 * (P** 0.2745) + 4.75668 * (P** 0.76537)

def ska(P):
    return 0.53928 * (P** 1.01394) + 94.31452 * (P** 0.35135)

def wff1 (P):
    return 0.00127717*(P**1.69617)+135.233*(P**0.331471)

def wff2 (P):
    return 0.00244523 * (P** 1.62692) + 122.076 * (P** 0.340401)

def apr1 (P):
    return 0.000719964 * (P** 1.85898) + 108.975 * (P** 0.340074)

def b11(P):
    return 0.488686 * (P** 1.01457) + 102.26 * (P** 0.355095)

def b12(P):
    return 1.34241 * (P** 0.910079) + 100.756 * (P** 0.354129)

def dh(P):
    return 39.5021 * (P** 0.541485) + 96.0528 * (P** 0.00401285)

def bgp(P):
    return 0.0112475 * (P** 1.59689) + 102.302 * (P** 0.335526)

def w(P):
    return 0.261822 * (P** 1.16851) + 92.4893 * (P** 0.307728)

def mdi1 (P):
    return 4.1844 * (P** 0.81449) + 95.00134 * (P** 0.31736)

def mdi2 (P):
    return 5.97365 * (P** 0.77374) + 89.24 * (P** 0.30993)
```

```

def mdi3 (P):
    return 15.55 * (P** 0.666) + 76.71 * (P** 0.247)

def mdi4 (P):
    return 25.99587 * (P** 0.61209) + 65.62193 * (P** 0.15512)

def hhj1 (P):
    return 1.78429 * (P** 0.93761) + 106.93652 * (P** 0.31715)

def hhj2 (P):
    return 1.18961 * (P** 0.96539) + 108.40302 * (P** 0.31264)

def ps(P):
    return 9805.95 * (1-np. exp (0.000193624 * (-P)))+ 212.072 * (1-np. exp (0.401508 * (-P)))+ 1.69483

def lattimersoftsym ():
    pp = Symbol('pp')
    eosfunc = - 4743.2711 * (pp** (- 0.03313) )+ 4885.28224 * (pp** (- 0.01027) )-0.00535 * (pp** 2.19938) + 0.01101 * (pp** 2.09858)
    return eosfunc

def lattimerstiffsym ():
    pp = Symbol ('pp')
    eosfunc = 81.56823 + 131.81073 * (1-exp (-pp/ 4.41577) )+ 924.14278 * (1-exp (-pp/523.73573) )
    return eosfunc

def nldsym ():
    pp = Symbol ('pp')
    eosfunc = 119.05736 + 304.80445 * (1-exp (-pp/ 48.61465) )+ 33722.34448 * (1-exp (-pp/ 17499.47411) )
    return eosfunc

def L80sym ():
    pp = Symbol ('pp')
    eosfunc = 5.97365 * (pp** 0.77374) + 89.24 * (pp** 0.30993)
    return eosfunc

def L95sym ():
    pp = Symbol ('pp')
    eosfunc = 15.54878 * (pp** 0.66635) + 76.70887 * (pp** 0.24734)
    return eosfunc

def lattimerintermediatesym ():
    pp = Symbol ('pp')
    eosfunc = 161.55325 + 2777.74571 * (1-exp (-pp/ 1909.97137) )+ 172.85805 * (1-exp (-pp/ 22.86436) )
    return eosfunc

def sk4sym ():
    pp = Symbol ('pp')
    eosfunc = 105.72158 * (pp** 0.2745) + 4.75668 * (pp** 0.76537)
    return eosfunc

def skasym ():
    pp = Symbol ('pp')
    eosfunc = 0.53928 * (pp** 1.01394) + 94.31452 * (pp** 0.35135)

```

```

eosfunc = 0.53928 * (pp** 1.01394) + 94.31452 * (pp** 0.35135)
return eosfunc

def wff1sym():
    pp = Symbol ('pp')
    eosfunc = 0.00127717 * (pp** 1.69617) + 135.233 * (pp** 0.331471)
    return eosfunc

def wff2sym():
    pp = Symbol ('pp')
    eosfunc = 0.00244523 * (pp** 1.62692) + 122.076 * (pp** 0.340401)
    return eosfunc

def apr1sym():
    pp = Symbol ('pp')
    eosfunc = 0.000719964 * (pp** 1.85898) + 108.975 * (pp** 0.340074)
    return eosfunc

def bl1sym():
    pp = Symbol ('pp')
    eosfunc = 0.488686 * (pp** 1.01457) + 102.26 * (pp** 0.355095)
    return eosfunc

def bl2sym():
    pp = Symbol ('pp')
    eosfunc = 1.34241 * (pp** 0.910079) + 100.756 * (pp** 0.354129)
    return eosfunc

def dhsym():
    pp = Symbol ('pp')
    eosfunc = 39.5021 * (pp** 0.541485) + 96.0528 * (pp** 0.00401285)
    return eosfunc

def bgpsym():
    pp = Symbol ('pp')
    eosfunc = 0.0112475 * (pp** 1.59689) + 102.302 * (pp** 0.335526)
    return eosfunc

def wsym():
    pp = Symbol ('pp')
    eosfunc = 0.261822 * (pp** 1.16851) + 92.4893 * (pp** 0.307728)
    return eosfunc

def mdi1sym():
    pp = Symbol ('pp')
    eosfunc = 4.1844 * (pp** 0.81449) + 95.00134 * (pp** 0.31736)
    return eosfunc

def mdi2sym():
    pp = Symbol ('pp')
    eosfunc = 5.97365 * (pp** 0.77374) + 89.24 * (pp** 0.30993)
    return eosfunc

def mdi3sym():
    pp = Symbol ('pp')

```

```

eosfunc = 15.55 * (pp** 0.666) + 76.71 * (pp** 0.247)
return eosfunc

def mdi4sym():
    pp = Symbol ('pp')
    eosfunc = 25.99587 * (pp** 0.61209) + 65.62193 * (pp** 0.15512)
    return eosfunc

def hhj1sym():
    pp = Symbol ('pp')
    eosfunc = 1.78429 * (pp** 0.93761) + 106.93652 * (pp** 0.31715)
    return eosfunc

def hhj2sym():
    pp = Symbol ('pp')
    eosfunc = 1.18961 * (pp** 0.96539) + 108.40302 * (pp** 0.31264)
    return eosfunc

def pssym():
    pp = Symbol ('pp')
    eosfunc = 9805.95 * (1- exp (0.000193624 * (-pp)))+ 212.072 * (1- exp (0.401508 * (-pp)))+ 1.69483
    return eosfunc

# crust equations
def eosc (P):
    return 0.00873 + 103.17338 * (1-np. exp (-P/ 0.38527) )+ 7.34979 * (1-np. exp (-P/0.01211) )

def eosc2 (P):
    return 0.00015 + 0.00203 * (1-np. exp (-P* 344827.5) )+ 0.10851 * (1-np. exp (-P*7692.3076) )

def eosc3 (P):
    return 0.000051 * (1-np. exp (-P* 0.2373 * (10** 10) ))+ 0.00014 * (1-np. exp (-P*0.4021 * (10** 8) ))

def eosc4 (P):
    return 10** (31.93753 + 10.82611 * np. log10 (P)+ 1.29312 * (np. log10 (P)** 2)+0.08014 * (np. log10 (P)** 3)+ 0.00242 * (np. log10 (P)** 4)+ 0.000028 * (np. log10 (P)** 5))

def eoscsym():
    pp = Symbol ('pp')
    eosfunc = 0.00873 + 103.17338 * (1-exp (-pp/ 0.38527) )+ 7.34979 * (1- exp (-pp/0.01211) )
    return eosfunc

def eoscsym2():
    pp = Symbol ('pp')
    eosfunc = 0.00015 + 0.00203 * (1-exp (-pp* 344827.5) )+ 0.10851 * (1- exp (-pp*7692.3076) )
    return eosfunc

def eoscsym3():
    pp = Symbol ('pp')
    eosfunc = 0.000051 * (1- exp (-pp* 0.2373 * (10** 10) ))+ 0.00014 * (1- exp (-pp*0.4021 * (10** 8) ))
    return eosfunc

def eoscsym4():
    pp = Symbol ('pp')
    eosfunc = 10** (31.93753 + 10.82611 * log (pp ,10) + 1.29312 * ( log (pp ,10) ** 2)+0.08014 * ( log (pp ,10) ** 3)+ 0.00242 * ( log (pp ,10) ** 4)+ 0.000028 * ( log (pp ,10) ** 5))
    return eosfunc

eoslib = {

```

```

eoslib = {
    'lattimer_soft': lattimersoft ,
    'lattimer_stiff': lattimerstiff ,
    'lattimer_intermediate': lattimerintermediate ,
    'NLD': nld ,
    '195': L95 ,
    '180': L80 ,
    'SKI4': ski4 ,
    'Ska': ska ,
    'WFF1': wff1 ,
    'WFF2': wff2 ,
    'APR1': apr1 ,
    'BL1': bl1 ,
    'BL2': bl2 ,
    'DH': dh ,
    'BGP': bgp ,
    'W': w,
    'MDI1': mdi1 ,
    'MDI2': mdi2 ,
    'MDI3': mdi3 ,
    'MDI4': mdi4 ,
    'HHJ1': hhj1 ,
    'HHJ2': hhj2 ,
    'PS': ps ,
    'eoscore': eosc ,
    'eoscore2': eosc2 ,
    'eoscore3': eosc3 ,
    'eoscore4': eosc4
};

eossym = {
    'lattimer_soft': lattimersoftsym ,
    'lattimer_stiff': lattimerstiffsym ,
    'lattimer_intermediate': lattimerintermediatesym ,
    'NLD': nldsym ,
    '195': L95sym ,
    '180': L80sym ,
    'SKI4': ski4sym ,
    'Ska': skasym ,
    'WFF1': wff1sym ,
    'WFF2': wff2sym ,
    'APR1': apr1sym ,
    'BL1': bl1sym ,
    'BL2': bl2sym ,
    'DH': dhsym ,
    'BGP': bgpsym ,
    'W': wsym ,
    'MDI1': mdil1sym ,
    'MDI2': mdi2sym ,
    'MDI3': mdi3sym ,
    'MDI4': mdi4sym ,
    'HHJ1': hhj1sym ,
    'HHJ2': hhj2sym ,
    'PS': pssym ,
    'eoscore': eoscsym ,
    'eoscore2': eoscsym2 ,
    'eoscore3': eoscsym3 ,
    'eoscore4': eoscsym4
}

```

5.2 TOV and Tidal effects: numerical integration

```

import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp
import os, json
import myeoslibs
from sympy import *

# Define right - hand system of equations
def tovrhs(r,z):
    M = z[0] # mass coordinate as part of array z
    P = z[1] # pressure coordinate as part of array z
    y = z[2] #yr parameter
    if P0 >= 0.184:
        e = myeoslibs.eoslib[eostype](P) #Equation of state
        dd= deos(P)
    elif P0 >= 9.34375e-5:
        e= myeoslibs.eoslib['eoscore'](P) #1st crust equation
        dd= deoscore(P)
    elif P0 >= 4.1725e-8:
        e= myeoslibs.eoslib['eoscore2'](P) #2nd crust equation
        dd= deoscore2(P)
    elif P0 >= 1.44875e-11:
        e= myeoslibs.eoslib['eoscore3'](P) #3rd crust equation
        dd= deoscore3(P)
    else :
        e= myeoslibs.eoslib['eoscore4'](P) #4th crust equation
        dd= deoscore4(P)
    dMdt = 11.2 * (10 ** (-6))* (r** 2)* e #1st order derivative of Mass
    dPdt = - 1.474 * (e* M/( r** 2 ))* (1+ 11.2 * (10 ** (-6)))* (r** 3)* P/M )* ((1 -
    2.948 * M/r )** (-1) ) #1st order derivative of Pressure
    F = (1- 1.474 * 11.2 * (10 ** (-6)))* (r** 2)* (e-P))** ((1 - 2.948 * M/r )** (-1) ) #F part
    J = 1.474 * 11.2 * (10 ** (-6)))* (r** 2)* (5* e+9* P+(( e+P )/(1/dd ))))** ((1 - 2.948 * M/r )** (-1) )-6* ((1 - 2.948 * M/r )** (-1) )-4* ((1.474 ** 2)* (M** 2)/(r** 2) )** ((1 + 11.2 *
    (10 ** (-6)))* (r** 3)* (P/M))** 2)* ((1 - 2.948 * M/r )** (-2) )
    dyrdt = (-y* y-y* F-J )/r
    dzdt = [dMdt ,dPdt , dyrdt ] # array of derivatives
    return dzdt # return the array of derivatives

# initial conditions
Step =0.0001;
G = 6.674*10**(- 33) ;
n =100;
ic1 = np.arange(1.5 ,5 ,0.1)
ic2 = np.arange(5 ,1201 ,1)
ic=np.concatenate(( ic1 , ic2 ),axis = None )

# insert EOS and crust EOS
eostype = 'lattimer_soft'
pp = Symbol('pp')
eosfuncprime = myeoslibs.eossym[eostype]().diff(pp)
deos = lambdify (pp , eosfuncprime , 'numpy')

eosfuncprimecore = myeoslibs.eossym['eoscore']().diff( pp )
deoscore = lambdify( pp , eosfuncprimecore , 'numpy')
eosfuncprimecore2 = myeoslibs.eossym['eoscore2']().diff( pp )
deoscore2 = lambdify( pp , eosfuncprimecore2 , 'numpy')
eosfuncprimecore3 = myeoslibs.eossym['eoscore3']().diff( pp )
deoscore3 = lambdify( pp , eosfuncprimecore3 , 'numpy')
eosfuncprimecore4 = myeoslibs.eossym['eoscore4']().diff( pp )
deoscore4 = lambdify( pp , eosfuncprimecore4 , 'numpy')

# setting up working directory
directory = os.path.join( os.getcwd() , eostype )
if not os.path.exists( directory ):
    print(" path doesn't exist.trying to make ")
    os.makedirs( directory )

```

```

minmax = np.zeros (( len( ic ) , 6) );
j = 0;

for i in ic :
    z0 =[0.0000000001 , i ,2.]
    P0 = 1.0;
    rmin =0.0000001;
    rmax =.01;
    Mf = np.array ([] )
    Pf = np.array ([] )
    R = np.array ([] )
    yr = np.array ([] )
    z0_old = np.array ([] )
    print(i)
    while ( P0>1e-12 ) :
        res = solve_ivp ( tovrhs ,( rmin , rmax ) ,z0 , method ='LSODA',
        atol =10 ** -26 , rtol=10 ** -8)
        # print(res.y[0][-1])
        z0_old = z0 [1]
        z0 [1]= res.y [1][~np.isnan ( res.y [1]) ][-1]
        z0 [0]= res.y [0][~np.isnan ( res.y [0]) ][-1]
        z0 [2]= res.y [2][~np.isnan ( res.y [2]) ][-1]
        if z0 [0]<0:
            break
        if ( z0_old == z0 [1]) :
            break
        rmin = res.t[~np.isnan ( res.y [2]) ][-1]
        rmax = rmin + 0.001
        P0 = z0 [1]
        Mf = np.append (Mf , res.y [0][~np.isnan ( res.y [0]) ])
        Pf = np.append (Pf , res.y [1][~np.isnan ( res.y [1]) ])
        yr = np.append (yr , res.y [2][~np.isnan ( res.y [2]) ])
        R = np.append (R , res.t[~np.isnan ( res.y [2]) ])
    if Pf [-1]<0:
        idx = np.argwhere ( Pf<0) [0 ,0]
        Pf = np.delete (Pf , np.s_ [ idx ::] , 0)
        Mf = np.delete (Mf , np.s_ [ idx ::] , 0)
        yr = np.delete (yr , np.s_ [ idx ::] , 0)
        R = np.delete (R , np.s_ [ idx ::] , 0)
        minmax [ j] = R[-1] , max ( Mf ) ,min ( Pf ) , yr [-1] ,0 ,0
    j = j + 1;
    beta = 1.474 * minmax [ j- 1][1] / minmax [j- 1][0]
    k2 = (8 * ( beta ** 5)/5) * ((1 -2* beta )** 2)* (2-yr [-1]+2* beta * ( yr [-1]-1 ))* (2*
    beta * (6-3* yr [-1]+3* beta * (5* yr [-1]-8) )+4* ( beta ** 3)* (13 -11 * yr [-1]+ beta * (3*
    yr [-1]-2)+2* ( beta ** 2)*(1+yr [-1]))+3* ((1 -2* beta )** 2)* (2-yr [-1]+2* beta * ( yr
    [-1]-1))* np. log (1-2* beta ))** (-1)
    l = 2/3* (( R[-1] ** 5)/G)* k2* 10** (- 36)
    minmax [ j- 1][4] = k2
    minmax [ j- 1][5] = l

with open ( os . path . join ( directory , 'data_'+ eostype +'.txt') , mode ='w') as f1 :
    namefile =---- data for EoS:+ eostype +'\r\n'
    f1 . write ( namefile )
with open( os.path.join( directory , 'data_'+ eostype +'.txt') , mode ='ab') as f1 :
    np . savetxt ( f1 , minmax , delimiter =",", header ='radius R (km)\t\t mass M (Msun )\t\t pressure P \t yr \t\t k2 \t\t lambda '
    , fmt ="%16f", comments ='', newline ='\r\n')
with open ( os . path . join ( directory , 'data_'+ eostype +'.json') , 'w') as outfile :
    json . dump ( minmax . tolist () , outfile )

```

5.3 Graphs and Statistical Analysis

```
import pickle
import numpy as np
import matplotlib.pyplot as plt
import json
import os
from scipy.stats import pearsonr, spearmanr
from scipy.stats import linregress

# Define a list of EOSs
eos_list = ['MDI1', 'Ska', 'W', 'WFF1', 'APR1', 'MDI4', 'BGP', 'HHJ1', 'SKI4', 'MDI2',
            'MDI3', 'WFF2', 'HHJ2', 'BL1', 'BL2', 'lattimer_stiff', 'NLD', 'DH',
            'lattimer_intermediate'] # Add your EOS names here
# Define a custom color palette with shades of blue, pink, purple, and red
custom_colors = [
    '#0099cc', '#ff66b2', '#9900cc', '#ff6666', '#3366ff', '#ff99cc', '#6633cc',
    '#ff3333', '#33ccff', '#ff6699', '#cc0066', '#00cc00', 'g', '#9999ccfb',
    '#ffcc00', '#ccff66', '#ff9900', '#ff6600', '#ff3300', '#13cfef', '4444ccfb'
]

# Define a list of linestyles for each EOS
linestyles = ['-', '--', '-.', '---', '-.', '--', '-.', '-.', '-.', '-.-', '-.-',
              '-.-', '-.-', '-.-', '-.-', '-.-', '-.-', '-.-', '-.-']

# Initialize lists to store data for each EOS
r_1_4_msun = []
lambda_1_4_msun = []
k2_1_4_msun = []
masses_list = []
radii_list = []
k2_values_list = []
lambda_values_list = []
yr_values_list = []
beta_values = [] # New list to store θ values
lambda_at_1_4_msun_list = [] # New list to store Λ at R(1.4M⊙) values
# Loop through each EOS
for i, eos in enumerate(eos_list):
    # Load data from the JSON file for the current EOS
    with open(os.getcwd() + f"\data_{eos}.json", 'r') as json_file:
        minmax_data = json.load(json_file)

    minmax = np.array(minmax_data)
    bound = np.where(minmax[:,1]==max(minmax[:,1]))[0][0]

    # Extract data for plotting
    radii = minmax[:bound, 0] # Radius in km
    masses = minmax[:bound, 1] # Mass in Msun
    k2_values = minmax[:bound, 4] # k2 value
    lambda_values = minmax[:bound, 5] # Lambda (λ) values
    yr_values = minmax[:bound, 3] # YR values (update if necessary)

    # Find the radius corresponding to 1.4M⊙ for the current EOS
    index_1_4_msun = np.argmin(np.abs(masses - 1.4))
    r_1_4_msun.append(radii[index_1_4_msun])
    lambda_1_4_msun.append(lambda_values[index_1_4_msun])
    k2_1_4_msun.append(k2_values[index_1_4_msun])
```

```

# Calculate θ for the current EOS
beta = 1.474 * masses / radii # θ = (1.474 * M) / R
beta_values.append(beta.tolist()) # Store θ values

# Calculate Λ at R(1.4 solar masses) for each EOS
mass_1_4_msun = 1.4 # Solar masses
radius_1_4_msun = r_1_4_msun[i] # Radius at 1.4 solar masses
index_1_4_msun = np.argmin(np.abs(masses - mass_1_4_msun))
lambda_at_1_4_msun = lambda_values[index_1_4_msun] / (masses[index_1_4_msun]**5)
lambda_at_1_4_msun_list.append(lambda_at_1_4_msun)

# Store data for each EOS
masses_list.append(masses)
radii_list.append(radii)
k2_values_list.append(k2_values)
lambda_values_list.append(lambda_values)
yr_values_list.append(yr_values)

# Create a dictionary to store θ values for each EOS
beta_data = {eos_list[i]: beta_values[i] for i in range(len(eos_list))}

# Save θ values to a JSON file
with open("beta_values.json", "w") as beta_file:
    json.dump(beta_data, beta_file)

# Create a text file for Λ at R(1.4 solar masses) with UTF-8 encoding
with open("lambda_at_1_4_msun.txt", "w", encoding="utf-8") as lambda_file:
    lambda_file.write("EOS, Lambda at R(1.4M $\odot$ )\n")
    for i, eos in enumerate(eos_list):
        coordinates = f"{eos}, {lambda_at_1_4_msun_list[i]:.6f}\n"
        lambda_file.write(coordinates)

# Adjust the line thickness for all the plots
line_thickness = 1.0

# Plot k2-θ for each EOS
plt.figure(figsize=(10, 6), dpi=300)
for i, eos in enumerate(eos_list):
    plt.plot(beta_values[i], k2_values_list[i], label=eos, color=custom_colors[i],
              linestyle=linestyles[i], linewidth=line_thickness)

plt.xlabel('β')
plt.ylabel('k2')
plt.title('k2-β Relation for Neutron Stars')
plt.grid()
plt.legend(title='EOS')
plt.savefig('plots/k2-beta_relation.png', dpi=300) # Save the graph
plt.show()

# Plot λ-θ for each EOS
plt.figure(figsize=(10, 6), dpi=300)
for i, eos in enumerate(eos_list):
    plt.plot(beta_values[i], lambda_values_list[i], label=eos, color=custom_colors[i],
              linestyle=linestyles[i], linewidth=line_thickness)

```

```

plt.ylim([0,8])
plt.xlabel('β')
plt.ylabel('Lambda (λ)')
plt.title('Lambda-β Relation for Neutron Stars')
plt.grid()
plt.legend(title='EOS')
plt.savefig('plots/lambda-beta_relation.png', dpi=300) # Save the graph
plt.show()

# Plot M-R for each EOS
plt.figure(figsize=(10, 6), dpi=300)
for i, eos in enumerate(eos_list):
    plt.plot(radial_list[i], masses_list[i], label=eos, color=custom_colors[i],
              linestyle=linestyles[i], linewidth=line_thickness)

plt.xlim([8,20])
plt.xlabel('Radius (km)')
plt.ylabel('Mass (Msun)')
plt.title('Mass-Radius (M-R) Relation for Neutron Stars')
plt.grid()
plt.legend(title='EOS')
plt.savefig('plots/M-R_relation.png', dpi=300) # Save the graph
plt.show()

# Plot M-R for each EOS
plt.figure(figsize=(10, 6), dpi=300)
with open("GW170817_MR1_maxmass_contour_data.pkl","rb") as f:
    ligo1 = pickle.load(f)
with open("GW170817_MR2_maxmass_contour_data.pkl","rb") as f:
    ligo2 = pickle.load(f)
ligo1['z_grid']=np.where(ligo1['z_grid']!=0.,ligo1['z_grid'],np.nan)
ligo2['z_grid']=np.where(ligo2['z_grid']!=0.,ligo2['z_grid'],np.nan)
#trimming invalid values so sorting is quicker
X1 = list(filter(lambda x: (not np.isnan(x)), ligo1['z_grid'].flatten()))
X2 = list(filter(lambda x: (not np.isnan(x)), ligo2['z_grid'].flatten()))
#function for getting the value of probability density where the confidence % cuts-off
def getProbCutoff(lst, confidence):
    lst.sort(reverse=True)
    S = sum(lst)
    csum = 0
    out = 0
    for i in lst:
        csum += i
        #print("sum:",csum, " percentage:",csum/S, " i:", i)
        if csum>confidence*S:
            out = i
            break
    return out
#plots probability density
plt.pcolormesh(ligo1['xx'], ligo1['yy'], ligo1['z_grid'], shading='auto', cmap=plt.cm.PuBu)
plt.pcolormesh(ligo2['xx'], ligo2['yy'], ligo2['z_grid'], shading='auto', cmap=plt.cm.YlOrBr)
#plots 90% confidence region boundary
plt.contour(ligo1['xx'], ligo1['yy'], ligo1['z_grid'], [getProbCutoff(X1, 0.5)],
            linestyles = "dashed", linewidths = .5, colors=["w"])
plt.contour(ligo2['xx'], ligo2['yy'], ligo2['z_grid'], [getProbCutoff(X2, 0.5)],
            linestyles = "dashed", linewidths = .5, colors=["w"])

```

```

#plots 50% confidence region boundary
plt.contour(ligo1['xx'], ligo1['yy'], ligo1['z_grid'], [getProbCutoff(X1, 0.9)],
            linewidths = .5, colors=["gray"])
plt.contour(ligo2['xx'], ligo2['yy'], ligo2['z_grid'], [getProbCutoff(X2, 0.9)],
            linewidths = .5, colors=["gray"])
for i, eos in enumerate(eos_list):
    plt.plot(radii_list[i], masses_list[i], label=eos, color=custom_colors[i],
              linestyle=linestyles[i], linewidth=line_thickness)

plt.xlim([8,20])
plt.xlabel('Radius (km)')
plt.ylabel('Mass (Msun)')
plt.title('Mass-Radius (M-R) Relation for Neutron Stars')
plt.grid()
plt.legend(title='EOS')
plt.savefig('plots/M-R_relation.png', dpi=300) # Save the graph
plt.show()

# Plot  $\lambda$ -M for each EOS
plt.figure(figsize=(10, 6), dpi=300)
for i, eos in enumerate(eos_list):
    plt.plot(masses_list[i], lambda_values_list[i], label=eos, color=custom_colors[i],
              linestyle=linestyles[i], linewidth=line_thickness)

plt.ylim([0,8])
plt.xlabel('Mass (Msun)')
plt.ylabel('Lambda ( $\lambda$ )')
plt.title('Lambda-Mass ( $\lambda$ -M) Relation for Neutron Stars')
plt.grid()
plt.legend(title='EOS')
plt.savefig('plots/lambda-M_relation.png', dpi=300) # Save the graph
plt.show()

# Plot k2-M for each EOS
plt.figure(figsize=(10, 6), dpi=300)
for i, eos in enumerate(eos_list):
    plt.plot(masses_list[i], k2_values_list[i], label=eos, color=custom_colors[i],
              linestyle=linestyles[i], linewidth=line_thickness)

plt.xlabel('Mass (Msun)')
plt.ylabel('k2')
plt.title('k2-Mass (k2-M) Relation for Neutron Stars')
plt.grid()
plt.legend(title='EOS')
plt.savefig('plots/k2-M_relation.png', dpi=300) # Save the graph
plt.show()

# Plot  $\lambda$ -R for each EOS
plt.figure(figsize=(10, 6), dpi=300)
for i, eos in enumerate(eos_list):
    plt.plot(radii_list[i], lambda_values_list[i], label=eos, color=custom_colors[i],
              linestyle=linestyles[i], linewidth=line_thickness)

plt.xlim([8,20])
plt.ylim([0,8])
plt.xlabel('Radius (km)')
plt.ylabel('Lambda ( $\lambda$ )')

```

```

plt.title('Lambda-Radius ( $\lambda$ -R) Relation for Neutron Stars')
plt.grid()
plt.legend(title='EOS')
plt.savefig('plots/lambda-R_relation.png', dpi=300) # Save the graph
plt.show()

# Plot k2-R for each EOS
plt.figure(figsize=(10, 6), dpi=300)
for i, eos in enumerate(eos_list):
    plt.plot(radii_list[i], k2_values_list[i], label=eos, color=custom_colors[i],
              linestyle=linestyles[i], linewidth=line_thickness)

plt.xlim([8,20])
plt.xlabel('Radius (km)')
plt.ylabel('k2')
plt.title('k2-Radius (k2-R) Relation for Neutron Stars')
plt.grid()
plt.legend(title='EOS')
plt.savefig('plots/k2-R_relation.png', dpi=300) # Save the graph
plt.show()

# Plot yr-M for each EOS
plt.figure(figsize=(10, 6), dpi=300)
for i, eos in enumerate(eos_list):
    plt.plot(masses_list[i], yr_values_list[i], label=eos, color=custom_colors[i],
              linestyle=linestyles[i], linewidth=line_thickness)

plt.xlabel('Mass (Msun)')
plt.ylabel('yr')
plt.title('yr-Mass (yr-M) Relation for Neutron Stars')
plt.grid()
plt.legend(title='EOS')
plt.savefig('plots/yr-M_relation.png', dpi=300) # Save the graph
plt.show()

# Plot yr-R for each EOS
plt.figure(figsize=(10, 6), dpi=300)
for i, eos in enumerate(eos_list):
    plt.plot(radii_list[i], yr_values_list[i], label=eos, color=custom_colors[i],
              linestyle=linestyles[i], linewidth=line_thickness)

plt.xlim([8,20])
plt.xlabel('Radius (km)')
plt.ylabel('yr')
plt.title('yr-Radius (yr-R) Relation for Neutron Stars')
plt.grid()
plt.legend(title='EOS')
plt.savefig('plots/yr-R_relation.png', dpi=300) # Save the graph
plt.show()

# Plot yr-B for each EOS
plt.figure(figsize=(10, 6), dpi=300)
for i, eos in enumerate(eos_list):
    plt.plot(beta_values[i], yr_values_list[i], label=eos, color=custom_colors[i],
              linestyle=linestyles[i], linewidth=line_thickness)

plt.xlabel('B')

```

```

plt.ylabel('yr')
plt.title('yr- $\beta$  Relation for Neutron Stars')
plt.grid()
plt.legend(title='EOS')
plt.savefig('plots/yr-beta_relation.png', dpi=300) # Save the graph
plt.show()

# Plot  $R(1.4M\odot)$ - $\lambda$  for each EOS with custom colors and point labels
plt.figure(figsize=(10, 6), dpi=300)
for i, eos in enumerate(eos_list):
    plt.scatter(lambda_1_4_msun[i], r_1_4_msun[i], label=eos, color=custom_colors[i])

plt.xlabel('Lambda ( $\lambda$ )')
plt.ylabel('Radius  $R(1.4M\odot)$  (km)')
plt.title('R( $1.4M\odot$ ) vs. Lambda ( $\lambda$ ) for Neutron Stars')
plt.grid()
plt.legend(title='EOS')
plt.savefig('plots/R_1_4_msun_vs_lambda.png', dpi=300) # Save the graph
plt.show()

# Plot  $R(1.4M\odot)$ - $k_2$  for each EOS with custom colors and point labels
plt.figure(figsize=(10, 6), dpi=300)
for i, eos in enumerate(eos_list):
    plt.scatter(k2_1_4_msun[i], r_1_4_msun[i], label=eos, color=custom_colors[i])

plt.xlabel('k2')
plt.ylabel('Radius  $R(1.4M\odot)$  (km)')
plt.title('R( $1.4M\odot$ ) vs. k2 for Neutron Stars')
plt.grid()
plt.legend(title='EOS')
plt.savefig('plots/R_1_4_msun_vs_k2.png', dpi=300) # Save the graph
plt.show()

# Create a text file for R( $1.4M\odot$ )- $\lambda$  coordinates with UTF-8 encoding
with open("r_lambda_coordinates.txt", "w", encoding="utf-8") as r_lambda_file:
    r_lambda_file.write("EOS, Lambda ( $\lambda$ ), Radius  $R(1.4M\odot)$  (km)\n")
    for i, eos in enumerate(eos_list):
        coordinates = f"{eos}, {lambda_1_4_msun[i]:.6f}, {r_1_4_msun[i]:.6f}\n"
        r_lambda_file.write(coordinates)

# Create a text file for R( $1.4M\odot$ )- $k_2$  coordinates with UTF-8 encoding
with open("r_k2_coordinates.txt", "w", encoding="utf-8") as r_k2_file:
    r_k2_file.write("EOS, k2, Radius  $R(1.4M\odot)$  (km)\n")
    for i, eos in enumerate(eos_list):
        coordinates = f"{eos}, {k2_1_4_msun[i]:.6f}, {r_1_4_msun[i]:.6f}\n"
        r_k2_file.write(coordinates)

# Calculate Pearson's correlation coefficient and p-value for R( $1.4M\odot$ )- $k_2$ 
pearson_corr_k2, pearson_p_value_k2 = pearsonr(r_1_4_msun, k2_1_4_msun)

```

```

# Calculate Spearman's rank correlation coefficient and p-value for R(1.4M $\odot$ )-k2
spearman_corr_k2, spearman_p_value_k2 = spearmanr(r_1_4_msun, k2_1_4_msun)

# Calculate Pearson's correlation coefficient and p-value for R(1.4M $\odot$ )- $\lambda$ 
pearson_corr_lambda, pearson_p_value_lambda = pearsonr(r_1_4_msun, lambda_1_4_msun)

# Calculate Spearman's rank correlation coefficient and p-value for R(1.4M $\odot$ )- $\lambda$ 
spearman_corr_lambda, spearman_p_value_lambda = spearmanr(r_1_4_msun, lambda_1_4_msun)

# Format the results with higher accuracy
accuracy = 6 # Adjust the number of decimal places as needed

print(f"Pearson's correlation coefficient (R(1.4M $\odot$ )-k2): {pearson_corr_k2:.{accuracy}f}")
print(f"Pearson's p-value (R(1.4M $\odot$ )-k2): {pearson_p_value_k2:.{accuracy}f}")
print(f"Spearman's rank correlation coefficient (R(1.4M $\odot$ )-k2): {spearman_corr_k2:.{accuracy}f}")
print(f"Spearman's p-value (R(1.4M $\odot$ )-k2): {spearman_p_value_k2:.{accuracy}f}")

print(f"Pearson's correlation coefficient (R(1.4M $\odot$ )- $\lambda$ ): {pearson_corr_lambda:.{accuracy}f}")
print(f"Pearson's p-value (R(1.4M $\odot$ )- $\lambda$ ): {pearson_p_value_lambda:.{accuracy}f}")
print(f"Spearman's rank correlation coefficient (R(1.4M $\odot$ )- $\lambda$ ): {spearman_corr_lambda:.{accuracy}f}")
print(f"Spearman's p-value (R(1.4M $\odot$ )- $\lambda$ ): {spearman_p_value_lambda:.{accuracy}f}")

# Define a significance level (alpha)
alpha = 0.05

# Interpretation for R(1.4M $\odot$ )-k2
if abs(pearson_corr_k2) >= 0.7 and pearson_p_value_k2 < alpha:
    interpretation_k2 = "Strong positive linear correlation"
elif abs(pearson_corr_k2) >= 0.7 and pearson_p_value_k2 >= alpha:
    interpretation_k2 = "No statistically significant linear correlation"
elif 0.3 <= abs(pearson_corr_k2) < 0.7 and pearson_p_value_k2 < alpha:
    interpretation_k2 = "Moderate positive linear correlation"
elif 0.3 <= abs(pearson_corr_k2) < 0.7 and pearson_p_value_k2 >= alpha:
    interpretation_k2 = "No statistically significant linear correlation"
elif abs(pearson_corr_k2) < 0.3:
    interpretation_k2 = "Weak or no linear correlation"

# Interpretation for R(1.4M $\odot$ )- $\lambda$ 
if abs(pearson_corr_lambda) >= 0.7 and pearson_p_value_lambda < alpha:
    interpretation_lambda = "Strong positive linear correlation"
elif abs(pearson_corr_lambda) >= 0.7 and pearson_p_value_lambda >= alpha:
    interpretation_lambda = "No statistically significant linear correlation"
elif 0.3 <= abs(pearson_corr_lambda) < 0.7 and pearson_p_value_lambda < alpha:
    interpretation_lambda = "Moderate positive linear correlation"
elif 0.3 <= abs(pearson_corr_lambda) < 0.7 and pearson_p_value_lambda >= alpha:
    interpretation_lambda = "No statistically significant linear correlation"
elif abs(pearson_corr_lambda) < 0.3:
    interpretation_lambda = "Weak or no linear correlation"

# Print interpretations
print("Interpretation for R(1.4M $\odot$ )-k2:", interpretation_k2)
print("Interpretation for R(1.4M $\odot$ )- $\lambda$ :", interpretation_lambda)

# Fit a linear regression model for R(1.4M $\odot$ )- $\lambda$ 
slope_lambda, intercept_lambda, r_value_lambda, p_value_lambda, std_err_lambda = linregress(lambda_1_4_msun, r_1_4_msun)
best_fit_lambda = lambda x: slope_lambda * x + intercept_lambda

```

```

# Fit a linear regression model for R(1.4M $\odot$ )-k2
slope_k2, intercept_k2, r_value_k2, p_value_k2, std_err_k2 = linregress(k2_1_4_msun, r_1_4_msun)
best_fit_k2 = lambda x: slope_k2 * x + intercept_k2

# Plot R(1.4M $\odot$ )-λ with best fit line
plt.figure(figsize=(10, 6), dpi=300)
for i, eos in enumerate(eos_list):
    plt.scatter(lambda_1_4_msun[i], r_1_4_msun[i], label=eos, color=custom_colors[i], linewidth=line_thickness)

# Add the best fit line to the plot
x_lambda = np.linspace(min(lambda_1_4_msun), max(lambda_1_4_msun), 100)
plt.plot(x_lambda, best_fit_lambda(x_lambda), color='black', linestyle='--', label='Best Fit Line')

# Display the equation for the best fit line
equation_lambda = f'y = {slope_lambda:.2f}x + {intercept_lambda:.2f}'
plt.text(0.1, 0.9, equation_lambda, transform=plt.gca().transAxes, fontsize=12, color='black')

plt.xlabel('Lambda (λ)')
plt.ylabel('Radius R(1.4M $\odot$ ) (km)')
plt.title('R(1.4M $\odot$ ) vs. Lambda (λ) with Best Fit Line')
plt.grid()
plt.legend(title='EOS')
plt.savefig('plots/R_1_4_msun_vs_lambda_with_best_fit.png', dpi=300) # Save the graph
plt.show()

# Plot R(1.4M $\odot$ )-k2 with best fit line
plt.figure(figsize=(10, 6), dpi=300)
for i, eos in enumerate(eos_list):
    plt.scatter(k2_1_4_msun[i], r_1_4_msun[i], label=eos, color=custom_colors[i], linewidth=line_thickness)

# Add the best fit line to the plot
x_k2 = np.linspace(min(k2_1_4_msun), max(k2_1_4_msun), 100)
plt.plot(x_k2, best_fit_k2(x_k2), color='black', linestyle='--', label='Best Fit Line')

# Display the equation for the best fit line
equation_k2 = f'y = {slope_k2:.2f}x + {intercept_k2:.2f}'
plt.text(0.1, 0.9, equation_k2, transform=plt.gca().transAxes, fontsize=12, color='black')

plt.xlabel('k2')
plt.ylabel('Radius R(1.4M $\odot$ ) (km)')
plt.title('R(1.4M $\odot$ ) vs. k2 with Best Fit Line')
plt.grid()
plt.legend(title='EOS')
plt.savefig('plots/R_1_4_msun_vs_k2_with_best_fit.png', dpi=300) # Save the graph
plt.show()

# Initialize lists to store M_max, R(M_max), and R(1.4) for each EOS
M_max_list = []
R_M_max_list = []
R_1_4_list = []

# Loop through each EOS
for i, eos in enumerate(eos_list):
    # Find the index corresponding to the maximum mass (M_max) for the current EOS
    index_M_max = np.argmax(masses_list[i])

```

```

# Calculate R(M_max) and R(1.4) for the current EOS
R_M_max = radii_list[i][index_M_max]
R_1_4 = r_1_4_msun[i]

# Append the values to the respective lists
M_max_list.append(masses_list[i][index_M_max])
R_M_max_list.append(R_M_max)
R_1_4_list.append(R_1_4)

# Create a text file to save the values
with open("eos_properties.txt", "w") as eos_file:
    eos_file.write("EOS, M_max (Msun), R(M_max) (km), R(1.4) (km)\n")
    for i, eos in enumerate(eos_list):
        properties = f"{eos}, {M_max_list[i]:.6f}, {R_M_max_list[i]:.6f}, {R_1_4_list[i]:.6f}\n"
        eos_file.write(properties)

# Print the calculated values for each EOS
for i, eos in enumerate(eos_list):
    print(f"EOS: {eos}")
    print(f"M_max (Msun): {M_max_list[i]:.6f}")
    print(f"R(M_max) (km): {R_M_max_list[i]:.6f}")
    print(f"R(1.4) (km): {R_1_4_list[i]:.6f}\n")

# Load Λ values from the previously saved text file
lambda_at_1_4_msun = []

with open("lambda_at_1_4_msun.txt", "r", encoding="utf-8") as lambda_file:
    next(lambda_file) # Skip the header line
    for line in lambda_file:
        eos, lambda_value = line.strip().split(", ")
        lambda_at_1_4_msun.append((eos, float(lambda_value)))

# Separate EOS names and Λ values
eos_names, lambda_values_R_1_4_msun = zip(*lambda_at_1_4_msun)

# Fit a linear regression model for R(1.4M⊙)-Λ
slope_lambda_R, intercept_lambda_R, r_value_lambda_R, p_value_lambda_R, std_err_lambda_R = linregress(
    lambda_values_R_1_4_msun, r_1_4_msun)
best_fit_lambda_R = lambda x: slope_lambda_R * x + intercept_lambda_R

# Plot R(1.4M⊙) vs. Λ for each EOS with custom colors and point labels
plt.figure(figsize=(10, 6), dpi=300)
for i, eos in enumerate(eos_names):
    plt.scatter(lambda_values_R_1_4_msun[i], r_1_4_msun[i], label=eos,
               color=custom_colors[i], linewidth=line_thickness)

# Add the best fit line to the plot
x_lambda_R = np.linspace(min(lambda_values_R_1_4_msun), max(lambda_values_R_1_4_msun), 100)
plt.plot(x_lambda_R, best_fit_lambda_R(x_lambda_R), color='black', linestyle='--', label='Best Fit Line')

# Display the equation for the best fit line
equation_lambda_R = f'y = {slope_lambda_R:.2f}x + {intercept_lambda_R:.2f}'

```

```

# Display the equation for the best fit line
equation_lambda_R = f'y = {slope_lambda_R:.2f}x + {intercept_lambda_R:.2f}'
plt.text(0.1, 0.9, equation_lambda_R, transform=plt.gca().transAxes, fontsize=12, color='black')

plt.xlabel('Lambda ( $\Lambda$ ) at R( $1.4M\odot$ )')
plt.ylabel('Radius R( $1.4M\odot$ ) (km)')
plt.title('R( $1.4M\odot$ ) vs. Lambda ( $\Lambda$ ) with Best Fit Line')
plt.grid()
plt.legend(title='EOS')
plt.savefig('plots/R_1_4_msun_vs_lambda_R_with_best_fit.png', dpi=300) # Save the graph
plt.show()

# Calculate Pearson's correlation coefficient and p-value for R( $1.4M\odot$ )- $\Lambda$ 
pearson_corr_lambda_R, pearson_p_value_lambda_R = pearsonr(r_1_4_msun, lambda_values_R_1_4_msun)

# Calculate Spearman's rank correlation coefficient and p-value for R( $1.4M\odot$ )- $\Lambda$ 
spearman_corr_lambda_R, spearman_p_value_lambda_R = spearmanr(r_1_4_msun, lambda_values_R_1_4_msun)

# Print correlation analysis results
print(f"Pearson's correlation coefficient (R( $1.4M\odot$ )- $\Lambda$ ): {pearson_corr_lambda_R:.{accuracy}f}")
print(f"Pearson's p-value (R( $1.4M\odot$ )- $\Lambda$ ): {pearson_p_value_lambda_R:.{accuracy}f}")
print(f"Spearman's rank correlation coefficient (R( $1.4M\odot$ )- $\Lambda$ ): {spearman_corr_lambda_R:.{accuracy}f}")
print(f"Spearman's p-value (R( $1.4M\odot$ )- $\Lambda$ ): {spearman_p_value_lambda_R:.{accuracy}f}")

# Interpretation for R( $1.4M\odot$ )- $\Lambda$ 
if abs(pearson_corr_lambda_R) >= 0.7 and pearson_p_value_lambda_R < alpha:
    interpretation_lambda_R = "Strong positive linear correlation"
elif abs(pearson_corr_lambda_R) >= 0.7 and pearson_p_value_lambda_R >= alpha:
    interpretation_lambda_R = "No statistically significant linear correlation"
elif 0.3 <= abs(pearson_corr_lambda_R) < 0.7 and pearson_p_value_lambda_R < alpha:
    interpretation_lambda_R = "Moderate positive linear correlation"
elif 0.3 <= abs(pearson_corr_lambda_R) < 0.7 and pearson_p_value_lambda_R >= alpha:
    interpretation_lambda_R = "No statistically significant linear correlation"
elif abs(pearson_corr_lambda_R) < 0.3:
    interpretation_lambda_R = "Weak or no linear correlation"

# Print interpretation for R( $1.4M\odot$ )- $\Lambda$ 
print("Interpretation for R( $1.4M\odot$ )- $\Lambda$ : ", interpretation_lambda_R)

```

Bibliography

- [1] Benjamin P Abbott et al. “GW170817: Measurements of neutron star radii and equation of state”. In: *Physical review letters* 121.16 (2018), p. 161101.
- [2] Benjamin P Abbott et al. “GW170817: observation of gravitational waves from a binary neutron star inspiral”. In: *Physical review letters* 119.16 (2017), p. 161101.
- [3] Artur Alho et al. “Compactness bounds in general relativity”. In: *Physical Review D* 106.4 (2022), p. L041502.
- [4] Χαράλαμπος Βάρβογλης and Γιάννης Σειραδάκης. “Εισαγωγή στη σύγχρονη αστρονομία”. In: *Γαρταγάνης, Θεσσαλονίκη* (1994).
- [5] K Ch Chatzisavvas et al. “Complexity and neutron star structure”. In: *Physics Letters A* 373.43 (2009), pp. 3901–3909.
- [6] Eanna E Flanagan and Tanja Hinderer. “Constraining neutron-star tidal Love numbers with gravitational-wave detectors”. In: *Physical Review D* 77.2 (2008), p. 021502.
- [7] Paweł Haensel, Aleksander Yu Potekhin, and Dmitry G Yakovlev. *Neutron Stars 1*. Springer, 2007.
- [8] Tanja Hinderer. “Tidal Love numbers of neutron stars”. In: *The Astrophysical Journal* 677.2 (2008), p. 1216.
- [9] Tanja Hinderer et al. “Tidal deformability of neutron stars with realistic equations of state and their gravitational wave signatures in binary inspiral”. In: *Physical Review D* 81.12 (2010), p. 123016.
- [10] Ch C Moustakidis et al. “Bounds on the speed of sound in dense matter, and neutron star structure”. In: *Physical Review C* 95.4 (2017), p. 045801.
- [11] J Piekarewicz. “Nuclear astrophysics in the multimessenger Era: a partnership made in Heaven”. In: *arXiv preprint arXiv:1812.04438* (2018).
- [12] Nikhil Sarin and Paul D Lasky. “The evolution of binary neutron star post-merger remnants: a review”. In: *General Relativity and Gravitation* 53.6 (2021), p. 59.
- [13] Jürgen Schaffner-Bielich. *Compact star physics*. Cambridge University Press, 2020.
- [14] ά μώ ά-έ. “Constraints on neutron stars equation of state using tidal deformability of the GW170817 system”. In: (2019).