Ph.D. Thesis

Structure of rotating self-gravitating figures of equilibrium in Newtonian gravity and general relativity with an emphasis on neutron stars

Thesis Advisors:
Prof. Simonetta Filippi
Prof. Jorge Armando Rueda Hernandez

Ph.D. student:
Federico Cipolletta

Academic Year 2013/2016

Contents

Introduction **1**

1 Uniform Rotation in Newtonian Gravity **3**
1.1 Hystorical Overview . . . 3
1.1.1 Observations . . . 3
1.1.2 Figures of equilibrium . . . 4
1.2 The problem of equilibrium . . . 4
1.2.1 EOS . . . 6
1.2.2 Rotation Law . . . 7
1.2.3 Numerical treatment . . . 7
1.2.4 Newton-Raphson's iteration scheme . . . 10
1.2.5 Initial guess solution . . . 11
1.2.6 Stopping Condition . . . 12
1.2.7 Physical Properties of a Configuration . . . 14
1.3 Results of Computationsand Comparisons with the Literature . . . 16
1.3.1 Grid Dependence . . . 17

2 Differential Rotation in Newtonian Gravity **21**
2.1 Introduction . . . 21
2.2 Preliminaries . . . 22
2.3 Already Known Differential Rotation Laws . . . 22
2.4 Implementation . . . 23
2.4.1 Stability . . . 23
2.4.2 Stopping Condition . . . 24
2.5 Known Reluts . . . 24
2.5.1 Comparison . . . 25
2.6 Multi-parametric Differential Rotation Law . . . 27
2.6.1 Note on Centrifugal Potential . . . 28
2.6.2 Note on the Stability Analysis . . . 30
2.6.3 Study of the parameters space . . . 33

3 Relativistic Figures of Equilibrium: Neutron Stars **43**
3.1 Introduction . . . 43
3.1.1 Numerical Methods . . . 44
3.2 The problem of Equilibrium in GR . . . 45
3.2.1 Structure Equations for a rotating star . . . 45

3.2.2 Note on Rotation Law . . . 48
3.2.3 EOS . . . 48
3.3 Equilibrium Configurations of NS in full GR . . . 51
3.3.1 Stability . . . 51
3.3.2 Constant Frequency sequences . . . 53
3.3.3 Observational Constraints . . . 57
3.3.4 Maximal Frequency and "Kerr Parameter" . . . 60
3.3.5 Some fitting formulas . . . 61
3.3.6 Eccentricity and Moment of Inertia . . . 62
3.3.7 Quadrupole Moment . . . 64

4 The Orbits of Particles around Rapidly Rotating NSs 73
4.1 Introduction . . . 73
4.2 Innermost Stable Circular Orbits . . . 73
4.2.1 BH Case . . . 74
4.2.2 Numerical Results for NS . . . 75
4.2.3 Existence of an exterior mostly bound circular orbit . . . 79

General Conclusions and Future Perspectives 85

Appendix A "CLASS_ROT" code 89
A.1 How the code works . . . 89
A.2 C code . . . 89

Introduction

Rotation plays a fundamental role in the study and understanding of the physics of stars. In effect, it affects many physical properties of the star, both in a classical framework (considering only Newtonian gravity) and relativistic one. To comprehend this statement, it only suffices to foresee that all rotating physical bodies in Universe being in gravitational equilibrium cannot show a spherical shape. Take the Earth as the simplest example: the centrifugal force pulls the matter at the equator against the gravitational force directed towards the centre and this results in an oblate configuration of equilibrium (the Earth is not a sphere, but in a first approximation it can be defined as a rotational ellipsoid flattened at poles). Spherical models of stars are the simplest to compute and understand, but they cannot be more than a limit to which realistic models of stars should tend in a non-rotational regime.

Rotation and the equilibrium between gravitational and centrifugal forces, allow a star to maintain more mass with respect to the non-rotating counterpart and this produces relevant physical effects, both in a Newtonian and relativistic regime.

Despite the interest that rotation could stimulate this, on the other hand, causes many difficulties in the production of models. But the numerical analysis helps when analytical methods are not available or not pragmatic. Many numerical methods were developed in past years to produce models of rotating stars, and they are still growing in efficiency and precision as technology pulls the computer science forward.

In the first chapter, we will start introducing the problem of equilibrium of rotating stars from a classical point of view. We will introduce the numerical method developed by Eriguchi and Muller in [24] to solve this problem in the case of polytropic EOS (the simplest one). In particular, we will present results of computations performed with our code based on this method in the case of uniform rotation, comparing them with the already known results in the literature (see paper by James [40], where an expansion method is used).

In the second chapter, we will introduce the problem of differential rotation, applying it to the method presented in the first chapter. We will compare results from our code with the ones presented in the paper by Eriguchi and Muller [24]. Then we will skip to a multi-parametric differential rotation law, where the angular velocity is a function of the distance from the axes of rotation (cylindrical radius) and of two free parameters. We will see how such a rotation law modifies the shape and structure of configurations and we will perform an analysis of the parameters space, to understand where stable equilibrium configurations are allowed to exist. This is a new way to study the problem and besides the physical relevance of such a rotation law, it should be interesting to have a two parameters space to investigate,

to understand where equilibrium configurations are admitted to exist and where not.

In the third chapter, we will focus on the same equilibrium problem in the case of Neutron Stars (NSs) in a relativistic regime. After a brief summary of known methods to approach this problem, we will present the method by Stergioulas and Friedmann. We will also report results for rapidly rotating NSs, from computations performed with the public code RNS by Stergioulas and their implications on both the internal structure of stars and exterior spacetime. Concerning the internal structure, we will define the limits of the stability region, namely the static limit, the keplerian limit and the secular instability limit. We will be able to study physical properties of rotating stars and also introduce some fits for physical quantities of NS (namely the binding energy, that is the relation between the baryonic mass and the gravitational mass). Concerning the exterior gravitational field, we will describe how to obtain information about the mass quadrupole moment of a rapidly rotating NS.

In the last chapter, we will study stable orbits of particles around the NS and we will also give some fits for binding energy and angular momentum of particles in these orbits (in particular for the mostly bound ones). We will additionally define a subregion in the aforementioned stability region, in which the mostly bound orbit of a NS reside outside the surface.

In the conclusions, we will summarize all novel results obtained in this work, and additionally we will provide an example of an astrophysical application of results in chapter three and four, where a collapsing scenario of a binary with a CO core and a NS companion is studied. The fits obtained in the aforementioned chapter will play a fundamental role in understanding the NS behaviour during this hypothetical situation.

In the appendix we will finally report our C code to build numerical models of rotating polytropic stars in Newtonian gravity, taking into account a two parameters differential rotation law.

Chapter 1

Uniform Rotation in Newtonian Gravity

1.1 Hystorical Overview

1.1.1 Observations

The first proof of the beginning of the study of stellar rotation dates back to 1610, when sunspots were firstly observed through a refracting telescope (see e. g. [63]). From these observations Galileo Galilei started his work, in which for the first time he noticed in 1613 that the motion of these spots, which turned out to be much slower when near the solar limb than when near the center, was an effect of the proximity to the stellar surface, thus they could not be small planets orbiting around the sun as Jesuit Father Christoph Scheiner suggested before. This was the first public declaration of Galileo of his adherence to the heliocentric theory of the solar system (and the beginning of his persecution by the Roman Inquisition).

Then for about two hundred years the problem of solar rotation remained ignored until 1850, when with a long series of observation Carrington and Spörer confirmed independently the differential rotation of the Sun. After that in 1871, Vogel in Potsdam showed how to determine the solar rotation from the relative Doppler shift of the spectral lines at opposite edges of the solar disc (one approaching and the other receding). In this direction in 1877 moved also Sir William de Wiveleslie Abney, who suggested that the Doppler effect could explain the great widths of certain stellar absorption lines.

The first convincing evidence of stellar axial rotation were instead made by Schlesinger in 1909, observing the eclipsing binary δ Librae, and after one year he reached the same conclusion also for the eclipsing system λ Tauri. In 1924, Rossiter observed the same effect in β Lyrae, giving the complete curve of residuals in the velocity during the eclipse, while McLaughing investigated this effect in β Persei. These last two were the first accurate measurements of axial rotation of stars. Also other approaches to the determination of stellar rotation were provided more or less simultaneously, but still for binary system. We have to wait 1930-1934 for a systematic study of stellar rotation performed on single stars by Sturve, Elvey and Westgate, in Chicago.

1.1.2 Figures of equilibrium

From a theoretical point of view, researches on how the rotation affects the internal structure and evolution of stars were begun by Sir Isaac Newton, who firstly realized the importance of a theory of gravitation to determine the figure of equilibrium of celestial bodies. From his work, many were the signs of progress during history, mainly due to Huygens, Descartes, Clairaut, Mclaurin, Legendre, Laplace, Poisson, Jacobi, Poincaré and Jeans. Their works are summarized in the theory of equilibrium and stability of ellipsoidal forms. The basic concepts of the theory of rotating stars were developed in the period from 1919 to 1941, with works by Milne and Chandrasekhar, who modelled slowly rotating stars, slightly deviating from spherical symmetry. Later on, with the advent of numerical computations, many results were reached, due to different methods. Also, the work of Jeans of 1928 (see [41]) plays an important role, with his study of polytropic stars both with uniform and differential rotation, with the hypothesis that the latter could lead to fission, thus to the formation of two stars.

One of the first methods is the one by James (see [40]), in which equilibrium sequences of uniformly rotating polytropic stars and completely degenerate white dwarfs were obtained. After this, Stoeckly in [61], proposed to treat differentially rotating stars with another method, afterwards extended by Butterworth and Ipser to treat also relativistic stars. However the choice to solve equations in their differential form, with the consequent necessity to treat boundary conditions, makes this method unstable and incapable of producing solutions of figures of equilibrium in general. In 1968, Ostriker and Mark (see [49]) introduced the Self Consistent Field (SFC) method, where the gravitational potential was treated in integral form, using Poisson's equation and this makes boundary conditions automatically and easily included. On the other hand, the gravitational potential and the hydrostatic equilibrium equation were solved iteratively, and this turns out to no certainty in the convergence, even if equilibria exist.

In 1985 Eriguchi and Müller (see [24]) published a new general method in which both Poisson's equation and hydrostationary equilibrium equation were treated in integral form and solved simultaneously to obtain the figure of equilibrium of rotating polytropic stars. It is worth to mention that this method has no limitations for rotation law and polytropic index. In the next sections, we will review this method and summarize the results of calculations performed taking into account uniform rotation.

1.2 The problem of equilibrium

Let's consider a configuration of rotating and self-gravitating gas for which the equation of hydrostationary equilibrium, in its differential form, reads

$$-v(\nabla v) = \frac{1}{\rho}\nabla P + \nabla\Phi_g, \tag{1.1}$$

being v, ρ, P and Φ_g respectively the fluid's velocity, density, pressure and gravitational potential. The last quantity should satisfy the Poisson's equation, which

for a general configuration reads

$$\Delta\Phi_g = \begin{cases} 4\pi G\rho, & \text{inside,} \\ 0, & \text{outside,} \end{cases} \tag{1.2}$$

being G the constant of gravitation. Note that left-hand side of Eq. (1.1) can be written as

$$-v(\nabla v) = -\frac{1}{2}\nabla v^2 - (\nabla \times v) \times v, \tag{1.3}$$

thus one obtains as the integrability condition for Eq. (1.1)

$$\nabla \times \left\{ \frac{1}{\rho}\nabla P + (\nabla \times v) \times v \right\} = 0. \tag{1.4}$$

Writing explicitly the fluid's velocity of a rotating gas in hydrostationary equilibrium in cylindrical coordinates (ϖ, z, ϕ) as

$$v = \varpi\Omega(\varpi, z)e_\phi, \tag{1.5}$$

one obtains that Equations (1.1) and (1.4) are respectively equivalent to

$$\frac{1}{\rho}\nabla P = -\nabla\Phi_g + \varpi\Omega^2(\varpi, z)e_\varpi, \tag{1.6}$$

$$2\varpi\Omega(\varpi, z)\frac{\partial\Omega(\varpi, z)}{\partial z}e_\phi = \nabla\frac{1}{\rho} \times \nabla P. \tag{1.7}$$

From Eq. (1.7), if one assume a barotropic Equation Of State (EOS), $P = P(\rho)$, one automatically obtains that

$$\frac{\partial\Omega}{\partial z} = 0, \tag{1.8}$$

and it is well known that this is a sufficient condition (see [63]) for isopycnic (constant density) and isobaric (constant pressure) surfaces to coincide. In addition, we also obtain that the centrifugal term in Eq. (1.6) comes out from a potential which can be defined as

$$\Phi_c = -\int_0^\varpi \varpi'\Omega^2(\varpi')d\varpi'. \tag{1.9}$$

Another important implication of Eq. (1.8), is the fulfillment of the Høiland criterion for stability (ref. [63]).

One can in principle treat Equations (1.2) and (1.6) in their respective presented form (differential), but this give rise to problems in treating the boundary conditions to impose, which are the finiteness of Φ_g and P at the center, the vanishing of Φ_g at infinity and the definition of the surface where P vanishes. On the other, treating the integral form of these equations, allows one to incorporate the boundary condition in an easier way. To do so, we have to note that Φ_g at a point $\vec{x}$, due to the presence of mass in the volume V, can be written as (see e.g. [39])

$$\Phi_g(\vec{x}) = -G \int_V \frac{\rho(\vec{x}')}{|\vec{x} - \vec{x}'|} dV, \tag{1.10}$$

which using spherical coordinates (r, θ, ϕ), axial and equatorial symmetry, is equal to

$$\begin{aligned} \Phi_g(r,\theta) = -4\pi G \int_0^{\frac{\pi}{2}} \sin(\theta') d\theta' \int_0^{r_{\text{surf}}(\theta)} r'^2 dr' \\ \sum_{n=0}^{\infty} f_{2n}(r,r') P_{2n}(\cos(\theta)) P_{2n}(\cos(\theta')) \rho(r',\theta'), \end{aligned} \tag{1.11}$$

where with $P_{2n}(\cos(\theta))$ we indicate the Legendre's polynomial of order $2n$ computed in $\cos(\theta)$ and f_{2n} are the Green's functions (of even orders), defined by

$$f_{2n}(r,r') = \begin{cases} \frac{r'^{2n}}{r^{2n+1}} & \text{for } r \geq r' \\ \frac{r^{2n}}{r'^{2n+1}} & \text{for } r < r' \end{cases} \tag{1.12}$$

At this point, it is possible to treat Eq. (1.1) in its integral form, which can be written as

$$\int \frac{1}{\rho} dP + \Phi_g + \Phi_c = C \text{ (const.)} \ . \tag{1.13}$$

The system to be solved is defined via Eq. (1.13) coupled to Eq. (1.11). But to close the circle one still has to insert the EOS, the boundary conditions to define the surface of the figure of equilibrium, namely

$$\rho(r_{\text{surf}}) = 0 \tag{1.14}$$

and a rotation law, that is a relation to express Ω as a function of the used coordinates, which will be used in the centrifugal potential term, Φ_c.

1.2.1 EOS

The choice of the EOS relation is one the most delicate points in approaching the problem of equilibrium of rotating gases, from the physical point of view. In fact, different EOSs can result in very different configurations of equilibrium of rotating stars.

For sake of simplicity we suppose that the gas we are modelling is a perfect gas, in the sense that it is composed of non-interacting particles (i.e. the effects of interacting particles is negligible, thus we can neglect viscosity implicated by energy dissipation during motion due to the interaction between particles) and it is non-degenerate (see e. g. [39]). An EOS to model this kind of physical properties is the polytropic one, thus

$$P = K\rho^{1+\frac{1}{n}}, \tag{1.15}$$

where K is the polytropic constant and n is the polytropic index.

1.2.2 Rotation Law

Another physically delicate point is the choice of the rotation law. A first condition which is worth noticing is the fact that the isopycnic (constant ρ) and isobaric (constant P) surfaces coincide if and only if $\frac{\partial \Omega}{\partial z} = 0$, that is the angular velocity does not depend on the cylindrical z component, thus it is constant over cylinders centred at the axis of rotation (see [39] and discussion in Sec. 1.2 about Eq. (1.8)). In particular, this condition is automatically satisfied for barotropes (thus also for polytropes) in a state of permanent rotation.

Knowing this one possibility is to assume a uniform rotation law, in which the angular velocity is constant in the entire configuration. This leads to a model of rigidly rotating body which can be considered as a first step in treating the problem of equilibrium of rotating stars. Intuitively, a more realistic motion of a body composed of gas should take into account the fact that following the density in a decreasing direction, the angular velocity should also decrease. This is what is called a "differential rotation law". However, as an approaching case, we initially assume uniform rotation, thus

$$\Omega = \text{ const.}, \tag{1.16}$$

letting the differentially rotating case to be treated afterwards.

1.2.3 Numerical treatment

To solve the system of equations defined by Equations (1.13), (1.14) and (1.11), in [24] a numerical method is presented in which the interior of the star is discretized in a mesh of points. Considering spherical coordinates, a grid is built dividing the domain into N_T points along the $\theta-$direction, and in N_R points along each $r-$direction. Grid points are defined as follows:

$$\theta_i = \frac{\pi}{2} \frac{i-1}{N_T - 1}, \ i = 1, \ldots, N_T, \tag{1.17}$$

$$r_{ij} = r_j(\theta_i) = r_{\text{surf}}(\theta_i) \frac{j}{N_R}, \ i = 1, \ldots, N_T, j = 1, \ldots, N_R. \tag{1.18}$$

In particular, this grid does not consider the centre of configurations because it has to be treated separately from other points. Note in addition that the angular values vary between 0 and $\frac{\pi}{2}$, as equatorial symmetry is taken into account, while there is no dependence on the φ coordinate as expected by axial symmetry. For sake of example refer to Fig. (1.1), where a simple grid of 4×4 points is depicted for a spherical symmetric case.

Now everything is ready to discretize the system of equations, but before doing so, let's introduce dimensionless variables, in order to make computations easier and to obtain an increased order of convergence (see [24]). Using EOS given by Eq. (1.15), allows us to introduce dimensionless variables like the ones used in [15], [61] or [63]. In particular, we adopt the following definitions for dimensionless variables:

$$\rho = \rho_c \sigma^n, \tag{1.19a}$$

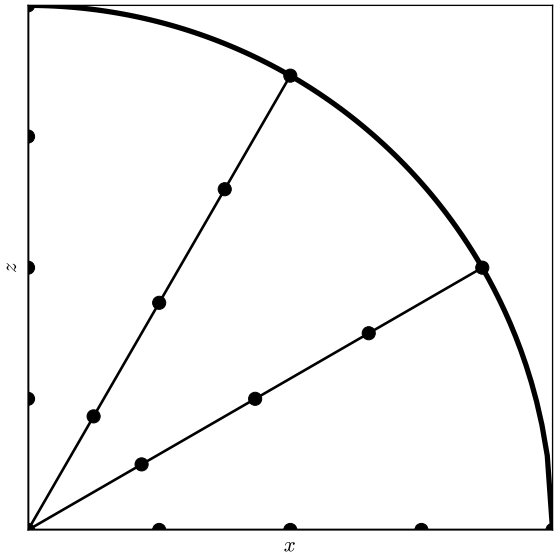

Figure 1.1: Example of a numerical grid defined by Equations (1.17) and (1.18), considering $N_T \times N_R = 4 \times 4$ and $r_{i,4} = r_{\rm surf}(\theta_i) =$ const. for every i.

$$r = \left[\frac{K(n+1)}{4\pi G}\rho_c^{\frac{1}{n}-1}\right]^{\frac{1}{2}} \xi, \tag{1.19b}$$

$$\Phi_g = \left[K(n+1)\rho_c^{\frac{1}{n}}\right] \Psi, \tag{1.19c}$$

$$\Omega^2 = [4\pi G\rho_c]\, \nu, \tag{1.19d}$$

where ρ_c is the central density of the model, σ, ξ, Ψ and ν are the dimensionless density, radius and gravitational potential and squared angular velocity, respectively. In particular, regarding definition in Eq. (1.19d), it is worth noticing that there are different possibilities, but we adopted the one used in [61], and keep in mind that to compare the results with ones presented in the literature we should always check the definitions of dimensionless variables.

Explicitly writing the system of equation for the fixed EOS and rotation law, one can note that a model is completely determined by the prescription of polytropic constant K, central density ρ_c and angular velocity Ω (this last holds in the case of uniform rotation; in case of differential rotation one should instead fix the central value of angular velocity only, knowing that the values in all other points are given by the chosen rotation law). With definitions given by Equations (1.19a - 1.19d) thus, a configuration in obtained only fixing the value of dimensionless squared angular velocity ν. In the method by Eriguchi and Mueller [24], it is stressed that fixing the angular velocity is not the best way to solve the system of equation. Instead a better choice for numerical calculation is to fix the axis ratio, namely $\frac{r_{\rm eq}}{r_{\rm pol}} = \frac{r_{N_R}(\frac{\pi}{2})}{r_{N_R}(0)} = \frac{\xi_{\rm eq}}{\xi_{\rm pol}}$, the last equality obtained by the introduction of the dimensionless radial coordinate. This last condition to fix gives one last equation to write which will make the system

solvable (see discussion on the number of equations and variables at the end of the present section).

Now we can write the discretized system of equations, which will read as follows:

(i) in the centre of configuration:

$$1 + \Psi_{\text{centre}} = C, \tag{1.20}$$

where

$$\Psi_{\text{centre}} = -\sum_{p=1}^{N_T} \sin(\theta_p)\Theta_p \sum_{q=1}^{N_R} \xi_{pq} R_{pq} \sigma_{pq}^n; \tag{1.21}$$

(ii) in all the grid points:

$$\sigma_{ij} + \Psi_{ij} - \frac{1}{2}\nu\xi_{ij}^2 \sin(\theta_i)^2 = C, \tag{1.22}$$

where

$$\begin{aligned} \Psi_{ij} = &-\sum_{p=1}^{N_T} \sin(\theta_p)\Theta_p \sum_{q=1}^{N_R} \xi_{pq}^2 R_{pq} \sigma_{pq}^n \\ &\sum_{m=0}^{\infty} f_{2m}(\xi_{ij}, \xi_{pq}) P_{2n}(\cos(\theta_i)) P_{2n}(\cos(\theta_p)) \sigma_{pq}^n; \end{aligned} \tag{1.23}$$

(iii) boundary conditions to define the surface:

$$\sigma_{i,N_R} = 0, \text{ for } i = 1, \ldots, N_T; \tag{1.24}$$

(iv) a last equation for the axis ratio:

$$\frac{\xi_{N_T,N_R}}{\xi_{1,N_R}} = \lambda. \tag{1.25}$$

Observe that in Equations (1.21) and (1.23), the integral over the volume is discretized. In particular, the terms Θ_p and R_{pq} denote respectively angular and radial grid spacings multiplied by weight factors which depend on the numerical integration scheme which in radial direction is the Simpson rule while in angular direction is the trapezoidal rule. Explicitly for the implementation, the values used for Θ_p and R_{pq} are:

$$\Theta_p = \begin{cases} \frac{\pi}{4(N_T-1)} & \text{for } p = 1, N_T \\ \frac{\pi}{2(N_T-1)} & \text{for every other } p \end{cases} \tag{1.26}$$

$$R_{pq} = \begin{cases} \frac{\xi_{\text{surf}}(\theta_p)}{3N_R} & \text{for } q = 1, N_R \\ \frac{2\xi_{\text{surf}}(\theta_p)}{3N_R} & \text{for } q \text{ odd} \\ \frac{4\xi_{\text{surf}}(\theta_p)}{3N_R} & \text{for } q \text{ even} \end{cases} \tag{1.27}$$

Let's count the number of variables and of equations in the system of equations. Choosing as initial condition for one particular configuration the axis ratio λ, the unknowns are: N_T dimensionless surface radii, $(N_T \times N_R)$ dimensionless density values for each grid point, the dimensionless squared angular velocity of the configuration ν and the constant of integration C (which one can observe that it corresponds to the dimensionless gravitational potential at the pole of the system for the current configuration cf.[63]). The total number of unknowns is, therefore, $N_T \times (N_R+1)+2$. One can easily count the total number of Equations (1.20), (1.22), (1.24) and (1.25) to find that it is equal to the number of unknowns.

In the next section, we will describe the numerical method used to solve the aforementioned system of equations.

1.2.4 Newton-Raphson's iteration scheme

The system of equations presented in the last section can be solved using an iterative numerical method, such the one called the Newton-Raphson's. In general, for functions of one variable only, the Newton-Raphson iteration scheme is based on the Taylor expansion of the function. Thus, given a function $f(x)$, one writes its Taylor expansion at a point $x + \delta$, such that δ is small enough, as:

$$f(x + \delta) = f(x) + \delta f'(x) + \frac{\delta^2}{2} f''(x) + \dots \tag{1.28}$$

Assuming that $f'(x) \neq 0$, if we want to write the i-th iteration, taking $x = x_i$ and $x_i + \delta = x_{i+1}$, with x_i sufficiently near the root of $f(x)$ and searching for a x_{i+1} such that $f(x_{i+1}) \approx 0$, one can use (1.28) to write:

$$0 = f(x_i) + (x_{i+1} - x_i) f'(x_i),$$

with an error up to the second order, thus

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}. \tag{1.29}$$

The generalization of Eq. (1.28) to the case in which $\mathbf{F} : \mathbb{R}^m \to \mathbb{R}^m$ is:

$$\mathbf{F}(\mathbf{x} + \delta) = \mathbf{F}(\mathbf{x}) + \mathbf{J}(\mathbf{x})\delta + O(\delta^2), \tag{1.30}$$

where with $\mathbf{J}$ we denote the Jacobian matrix associated with the function $\mathbf{F}$, with $J_{ik} = \frac{\partial F_i}{\partial x_k}$. As before, one obtains

$$\mathbf{x_{i+1}} = \mathbf{x_i} - \mathbf{J}^{-1}(\mathbf{x_i})\mathbf{F}(\mathbf{x_i}), \tag{1.31}$$

where, $\mathbf{x_{i+1}} - \mathbf{x_i} = \delta = -\mathbf{J}^{-1}(\mathbf{x_i})\mathbf{F}(\mathbf{x_i})$, can be computed solving the linear system $\mathbf{J}(\mathbf{x_i})\delta = -\mathbf{F}(\mathbf{x_i})$ or in other words, finding the inverse matrix $\mathbf{J}^{-1}(\mathbf{x_i})$.

Now, applying the Newton-Raphson method to the problem of equilibrium described in the previous section, the vector function $\mathbf{F}$ can be the one defined in Equations (1.20), (1.22), (1.24) and (1.25). We only have to define the vector of unknowns $\mathbf{x}$, and we put

$$\begin{aligned} &x_i = r_{\rm surf}(\theta_i), \text{ for } i = 1, \ldots, N_T; \\ &x_{N_T+(i-1)\times N_R+j} = \sigma_{i,j}, \text{ for } i = 1, \ldots, N_T \text{ and } j = 1, \ldots, N_R; \\ &x_{N_T\times(N_R+1)+1} = \nu; \\ &x_{N_T\times(N_R+1)+2} = C. \end{aligned} \tag{1.32}$$

To obtain the solution for the i-th iteration, namely $\mathbf{x}_{i+1}$, one as to compute the Jacobian matrix of the system with respect to the vector of unknowns given by Eq. (1.32), and to solve the linear system analogous to the one in (1.31). The iterations are performed until the relative change in each variable is less than a prescribed limit.

To implement the Newton-Raphson method in our C code (see App. A), we decided to insert some routines taken from the book [55]. In these routines, the aforementioned limit for the change in each variable is taken as TOLX $= 10^{-4}$, and they allow to solve a general user defined system with a globally convergent method. The only missing thing up to now is the initial guess, namely $\mathbf{x}_0$, and in the next section, we will see how it can be computed.

1.2.5 Initial guess solution

The method in [24] start from a spherical (non-rotating) configuration and gradually increases the axis ratio of the model by a constant factor near 1 (for example 1.05 or less) to obtain a new model via the Newton-Raphson method and then repeats the procedure until a certain stopping condition is reached. The computation of a spherically symmetric model (meant as the density distribution and the surface radius) of a polytropic star of index n is a well-known problem in astrophysics and it was studied by many authors in the past (see e. g. [15]). The equation to solve, namely the Lane-Emden equation, reads

$$\frac{1}{\xi^2}\frac{\partial}{\partial\xi}\left(\xi^2\frac{\partial\sigma(\xi)}{\partial\xi}\right) + \sigma(\xi)^n = 0, \tag{1.33}$$

where we have adopted the same notation for dimensionless variables as in the Sec. 1.2.3. It is known in the literature that Eq. (1.33) admits analytical solutions for the cases $n = 0$, 1 and 5, if "geometrical" dimensions are three (the case of a sphere). In [39] one could see that the following are exact analytic solutions for these three cases:

(i) $n = 0$

$$\theta(\xi) = 1 - \frac{\xi^2}{6}, \tag{1.34}$$

(ii) $n = 1$

$$\theta(\xi) = \frac{\sin \xi}{\xi}, \tag{1.35}$$

(iii) $n = 5$

$$\theta(\xi) = \left(1 + \frac{\xi^2}{3}\right)^{-\frac{1}{2}}. \tag{1.36}$$

For different n, also singular solutions exist, but this is not relevant in our context. Excluded the cases of Equations (1.34), (1.35) and (1.36), the numerical approach only would work in solving Lane-Emden equation.

One can solve Eq. (1.33) numerically, finding $\sigma(\xi)$, imposing as initial conditions

$$\sigma(0) = 1, \tag{1.37a}$$

$$\sigma'(0) = 0. \tag{1.37b}$$

Through a Newton-Raphson iteration scheme, one can also find the zero of function $\sigma(\xi)$, which represents the surface radius of the spherical solution. Finally, one can put $\nu_0 = 0$ and find C_0 with an expression analogous to Eq. (1.23).

It is worth to mention that in our C code presented in App. A the routine to build the initial guess solution is not reported, but a simple program to solve differential equations may be used to do it. At this point, all components of the initial guess vector $\mathbf{x}_0$ are defined, and this is the starting point for the computation of a sequence of configurations, to be performed by Newton-Raphson iteration scheme exposed in sec. 1.2.4. The sequence will be continued, increasing the value of the axis ratio $\frac{\xi_{\rm eq}}{\xi_{\rm pol}}$, until some condition will be satisfied and in the next section this will be explained.

1.2.6 Stopping Condition

In the previous sections, we saw how the method by [24] produces a sequence of equilibrium configurations with increasing axis ratio (thus increasing angular velocity). But when this sequence should be stopped? A good answer to this question can be found defining the concept of equilibrium of a model of a rotating star. As already presented in [40] or [63] one limit for equilibrium configurations is when the effective gravity of the surface at the equator become zero, which means that the gravitational force is perfectly balanced by the centrifugal force (and vice-versa). Actually, if the angular velocity would increase further after this condition is reached, a portion of the total mass should begin to shed from the star, because the centrifugal force would become greater than the gravitational one (the so-called "mass-shedding").

As for all other values considered in the computation, we would like to treat with dimensionless quantities, thus to understand the non-dimensionalization of the effective gravity we use the following definition, for the non-rotational case:

$$g_{\rm eff} = -\frac{\frac{dP(r)}{dr}}{\rho(r)}, \tag{1.38}$$

or in other words, the gradient of the pressure P. Using the EOS in Eq. (1.15) and the definitions given by Equations (1.19a) and (1.19b), one gets the following definition for dimensionless effective gravity

$$g_{\text{eff}} = -2\left[\pi G(n+1)K\rho_c^{\frac{1}{n}+1}\right]^{\frac{1}{2}}\frac{\partial\sigma}{\partial\xi}. \tag{1.39}$$

Moving to the uniformly rotating case let's compute the effective gravity and reporting its expression from [63], it reads:

$$g_\varpi = -\frac{\partial\Phi_g}{\partial\varpi} + \Omega^2\varpi, \tag{1.40a}$$

$$g_z = -\frac{\partial\Phi_g}{\partial z}, \tag{1.40b}$$

being (ϖ, z, ϕ) cylindrical coordinates. As we are interested in the *equatorial* effective gravity, observe that the only component to consider in our case is the one in Eq. (1.40a), where the cylindrical radius ϖ corresponds to the spherical radius r. Thus

$$g_{\text{eq eff}} = g_\varpi = g_r = -\frac{\partial\Phi_g}{\partial r} + \Omega^2 r. \tag{1.41}$$

Now we can perform a non-dimensionalization also for the effective gravity, using Equations (1.19b), (1.19c) and (1.19d). Explicitly

$$-\frac{\partial\Phi_g}{\partial r} + \Omega^2 r = -\frac{\left[(4\pi G)^{\frac{1}{2}}K(n+1)\rho_c^{\frac{1}{n}}\right]}{(K(n+1)\rho_c^{\frac{1}{n}-1})^{\frac{1}{2}}}\frac{\partial\Psi}{\partial\xi} + [4\pi G\rho_c]\left[\frac{K(n+1)}{4\pi G}\rho_c^{\frac{1}{n}-1}\right]^{\frac{1}{2}}\nu\xi =$$

$$= 2\left[\pi G(n+1)K\rho_c^{\frac{1}{n}+1}\right]^{\frac{1}{2}}\left(-\frac{\partial\Psi}{\partial\xi} + \nu\xi\right),$$

thus denoting with $\tilde{g}_{\text{eq eff}}$ the dimensionless equatorial effective gravity, we can write

$$\tilde{g}_{\text{eq eff}} = -\frac{\partial\Psi}{\partial\xi} + \nu\xi. \tag{1.42}$$

Considering the following general form of Eq (1.22) for the dimensionless gravitational potential

$$\Psi(\xi,\theta) = -\sigma(\xi,\theta) + \frac{1}{2}\nu\xi^2\sin(\theta)^2 + \Psi_0, \tag{1.43}$$

being Ψ_0 the dimensionless gravitational potential at the pole of the system, we then get:

$$\tilde{g}_{\text{eq eff}} = -\frac{\partial\Psi}{\partial\xi}(\xi,\theta) + \nu\xi = -\frac{\partial\left(-\sigma(\xi,\theta) + \frac{1}{2}\nu\xi^2\sin(\theta)^2 + \Psi_0\right)}{\partial\xi} + \nu\xi,$$

thus we can say

$$\tilde{g}_{\text{eq eff}} = \frac{\partial \sigma}{\partial \xi}(\xi_{\text{surf eq}}, \frac{\pi}{2}). \tag{1.44}$$

Concerning the implementation, given an equilibrium configuration one has all the numerical values of the dimensionless density σ in each grid-point and the numerical approximation of the derivative of σ with respect to the variable ξ at the point $(\xi_{\text{surf eq}}, \frac{\pi}{2})$ is needed to compute the value of equatorial effective gravity. To perform such an approximation we recall the *three points approximation*, which allows computing the numerical approximation of the derivative of a function calculated at a boundary point, knowing the function's values on a discrete set of points. Denoting with $[x_1, \ldots, x_k]$ this set of k points, for a given function f, with a fixed $h = x_{i+1} - x_i$ for each index $i = 1 \ldots k-1$, one can use the Taylor expansion to easily verify that:

$$f'(x_k) = \frac{1}{2h}\left[3f(x_k) - 4f(x_{k-1}) + f(x_{k-2})\right] + O(h^2). \tag{1.45}$$

In the case of function in m variables, the expression of the partial derivative with respect to one particular variable is the same of Eq. (1.45), keeping fixed all the other variables. Thus using Eq. (1.45) with $f \equiv \sigma$, $x_k \equiv \xi_{\text{surf eq}}$ and $h = \frac{\xi_{\text{surf eq}}}{N_R}$, we can compute the dimensionless equatorial effective gravity using Eq. (1.44). When the condition

$$\frac{\partial \sigma}{\partial \xi}(\xi_{\text{surf eq}}, \frac{\pi}{2}) > 0 \tag{1.46}$$

is reached the sequence should be stopped because it implies a positive equatorial effective gravity, which means that mass-shedding has occurred.

1.2.7 Physical Properties of a Configuration

Once the density distribution and surface radii of a configuration are computed, one could also compute physical properties of the model and this allows to give some physical interpretations. In the present section, we will review how to compute the most important physical properties of a rotating star in Classical (Newtonian) gravity. The reference for the subsequent expression can be found in [39] and [16].

Firstly, the rotational kinetic energy of a rotating configuration can be found with the following expression

$$T = \frac{1}{2}\int_V \rho \Omega^2 dV, \tag{1.47}$$

where V is the volume and ρ represents as usual the density.

The gravitational potential energy, on the other hand, can be computed as

$$W = -\frac{1}{2} G \int_V \rho \Phi_g dV, \tag{1.48}$$

being G the constant of gravitation and Φ_g the gravitational potential, which can be computed with Eq. (1.10).

The internal heat energy of the system reads as

$$U = \frac{1}{\gamma - 1} \int_V \rho dV, \tag{1.49}$$

where γ is the polytropic adiabatic exponent defined as $\gamma = 1 + \frac{1}{n}$.

Obviously the total mass of the configuration is simply

$$M = \int_V \rho dV, \tag{1.50}$$

while the total angular momentum is given by

$$J = \int_V \rho \Omega^2 \varpi^2 dV, \tag{1.51}$$

where with ϖ again the cylindrical radius is meant, thus $\varpi = r\sin(\theta)$ when spherical coordinates are considered.

Finally another quantity which is worth computing is the value of virial test, which in [24] (just as in [48]) is expressed in the following form

$$V.T. = \left| \frac{(2T + W + 3(\gamma - 1)U)}{W} \right|, \tag{1.52}$$

and it should be 0 in order to have an accurate model.

For a computation matter it is important to obtain non-dimensional quantities in a natural way, just from Equations (1.19a), (1.19b), (1.19c) and (1.19d). One can define each physical quantity in dimensionless form, adopting the following equation as a general rule:

$$F = k_F \widetilde{F}, \tag{1.53}$$

where F is the physical quantity taking into account its real dimensions, k_F is the constant for non-dimensionalization and $\widetilde{F}$ is the quantity in dimensionless form. In particular, as in [24] we adopt non-dimensionalization constant as reported in Tab. 1.1. In this table n is the polytropic index, K is the polytropic constant and G is the constant of gravitation. To make calculations of all the aforementioned quantities, numerical integration can be used, with same methods and same numerical weight factors described in Sec. 1.2.3, given by Equations (1.26) and (1.27).

Table 1.1: Constants adopted to obtain dimensionless quantities and names of these.

F	k_F	$\widetilde{F}$
ρ	ρ_c	σ^n
P	$K\rho_c^{1+\frac{1}{n}}$	σ^{n+1}
Ω^2	$[4\pi G\rho_c]$	ν
Φ_g	$K(n+1)\rho_c^{\frac{1}{n}}$	Ψ
r	$\left[\frac{K(n+1)}{4\pi G}\rho_c^{\frac{1}{n}-1}\right]^{\frac{1}{2}}$	ξ
T	$\frac{[K(n+1)]^{\frac{5}{2}}}{[4\pi]^{\frac{1}{2}}G^{\frac{3}{2}}}\rho_c^{\frac{5-n}{2n}}$	$\widetilde{T}$
W	$\frac{[K(n+1)]^{\frac{5}{2}}}{[4\pi]^{\frac{1}{2}}G^{\frac{3}{2}}}\rho_c^{\frac{5-n}{2n}}$	$\widetilde{W}$
U	$\frac{n}{(n+1)}\frac{[K(n+1)]^{\frac{5}{2}}}{[4\pi]^{\frac{1}{2}}G^{\frac{3}{2}}}\rho_c^{\frac{5-n}{2n}}$	$\widetilde{U}$
M	$4\pi\left[\frac{K(n+1)}{4\pi G}\right]^{\frac{3}{2}}\rho_c^{\frac{3-n}{2n}}$	$\widetilde{M}$
J	$\frac{[K(n+1)]^{\frac{5}{2}}}{4\pi G^2}\rho_c^{\frac{5-2n}{2n}}$	j

1.3 Results of Computations and Comparisons with the Literature

We implemented the method by Eriguchi and Mueller [24] with the C code reported in App. A, taking into account a uniform rotation law, to produce a sequence of configurations of rotating polytropic stars, starting from the solution of Lane-Emden equation (1.33), changing the axis ratio $\frac{r_{\rm eq}}{r_{\rm pol}}$ by a certain factor and implementing the Newton-Raphson method of Sec. 1.2.4 until mass-shedding occurs, that is when equatorial effective gravity, given by Eq. (1.44), becomes positive. In particular we will focus in the case $n = 1.5$. To have a good comparison of results, we decided to use the ones presented in the paper by James [40] and in Tab. 1.2 they are reported. A detail which is worth to mention is that the dimensionless angular velocity used for the present results have the following definition:

$$D = \frac{\Omega^2}{2\pi G\rho_c}, \tag{1.54}$$

thus $D = 2\nu$, if one consider the previous definition given in Eq. (1.19d).

Table 1.2: Results for numerical computations performed by James, taken by [40].

D	$\frac{\xi_{\text{eq}}}{\xi_{\text{pol}}}$	$\lvert g_{\text{eff eq}}\rvert$	M
0.	1.000000000	0.20330	2.7141
$2.00e-3$	1.011763736	0.19829	2.7297
$4.00e-3$	1.023931765	0.19315	2.7457
$6.00e-3$	1.036509301	0.18790	2.7622
$8.00e-3$	1.049586547	0.18251	2.7791
$1.00e-2$	1.063199105	0.17697	2.7966
$1.20e-2$	1.077381120	0.17128	2.8145
$1.40e-2$	1.092140157	0.16542	2.8331
$1.60e-2$	1.107631706	0.15936	2.8522
$1.80e-2$	1.123832840	0.15309	2.8719
$2.00e-2$	1.140928849	0.14659	2.8923
$2.20e-2$	1.158991008	0.13983	2.9134
$2.40e-2$	1.178178669	0.13277	2.9354
$2.60e-2$	1.198660132	0.12537	2.9581
$2.80e-2$	1.220619518	0.11756	2.9818
$3.00e-2$	1.244414512	0.10928	3.0056
$3.20e-2$	1.270458759	0.10040	3.0323
$3.40e-2$	1.299373041	0.09080	3.0593
$3.60e-2$	1.332114619	0.08021	3.0877
$3.80e-2$	1.370353771	0.06825	3.1177
$4.00e-2$	1.417457647	0.05410	3.1496
$4.16e-2$	1.467388681	0.03983	3.1767
$4.24e-2$	1.501133787	0.03063	3.1910
$4.32e-2$	1.551074210	0.01769	3.2057
$4.36e-2$	1.608026938	0.00400	3.2133

1.3.1 Grid Dependence

In our computations we tried different grids and in the subsequent figures the reader can see comparisons for dimensionless angular velocity D, modulus of equatorial effective gravity $|g_{\text{eff eq}}|$ and total mass M between these computations for grid $N_T \times N_R = 8 \times 22$ (green circles), 10×30 (blue triangles), 15×40 (red squares), 20×60 (purple stars) and results of table Tab. 1.1 (black diamonds). From one configuration to the sequent the factor used to increase the axis ratio is about 1.02.

From Fig.s 1.2, 1.3 and 1.4 the grid-dependence of configurations is evident, and this is also quite intuitive: using a rather coarse grid, like the one represented by green circles, the results seem to be quite inaccurate (at least for values of M), most of all for greater values of axis ratio, that is for most rapidly rotating configurations. On the other hand, when a more refined grid is considered, one can notice that results tend to the ones of James [40], which were obtained through analytical expansions'

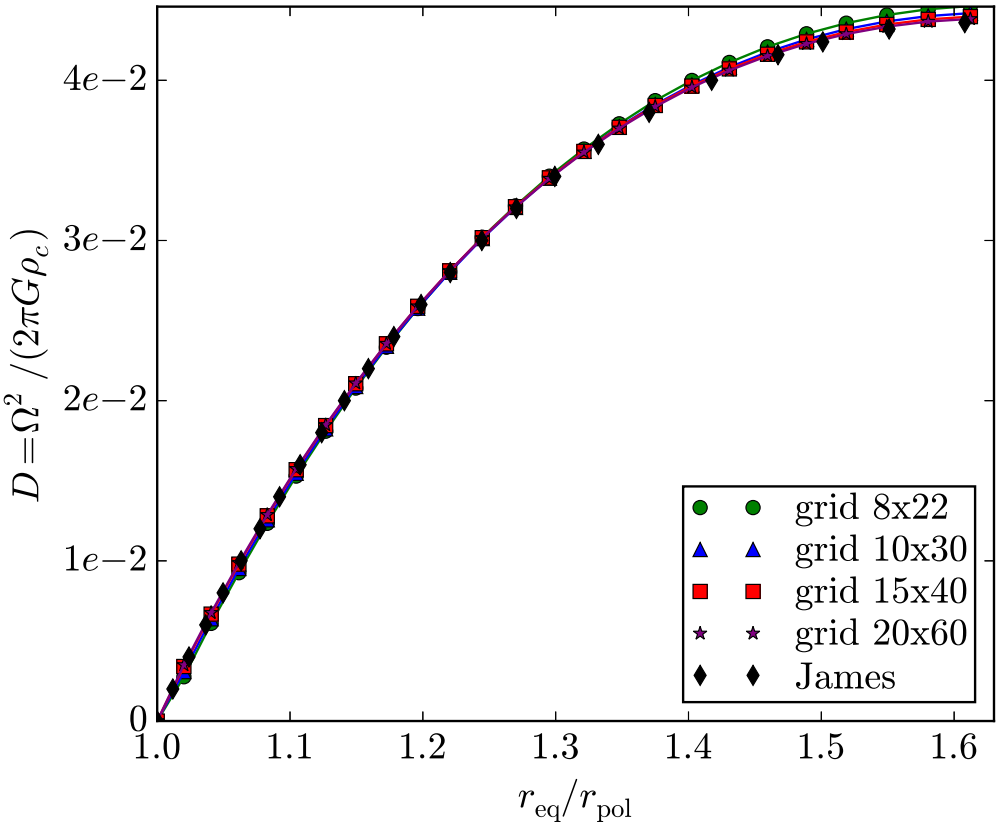

Figure 1.2: Comparison of results obtained with our implementation of method by Eriguchi and Mueller and results by James, where dimensionless angular velocity D is plotted as a function of axis ratio $\frac{r_{\mathrm{eq}}}{r_{\mathrm{pol}}} = \frac{\xi_{\mathrm{eq}}}{\xi_{\mathrm{pol}}}$.

approach.

A last remark which is worth to make is that considering smaller steps in the factor to increase the axis ratio from one configuration to another, for example a factor of 1.01, some numerical oscillations may occur in results, but again these fluctuations in numerical values, appear to recede with a direct dependence on the chosen grid for computations, to finally tend to the values provided by James in [40].

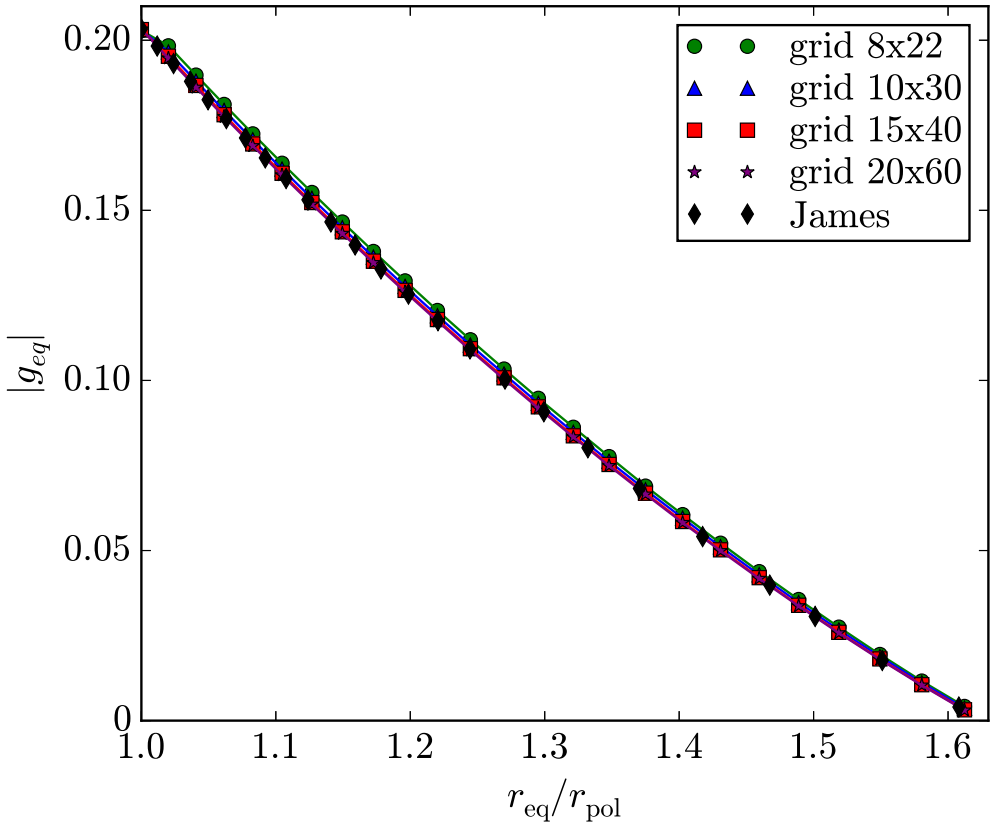

Figure 1.3: Same as Fig. 1.2 but for equatorial effective gravity $|g_{\rm eff\ eq}|$.

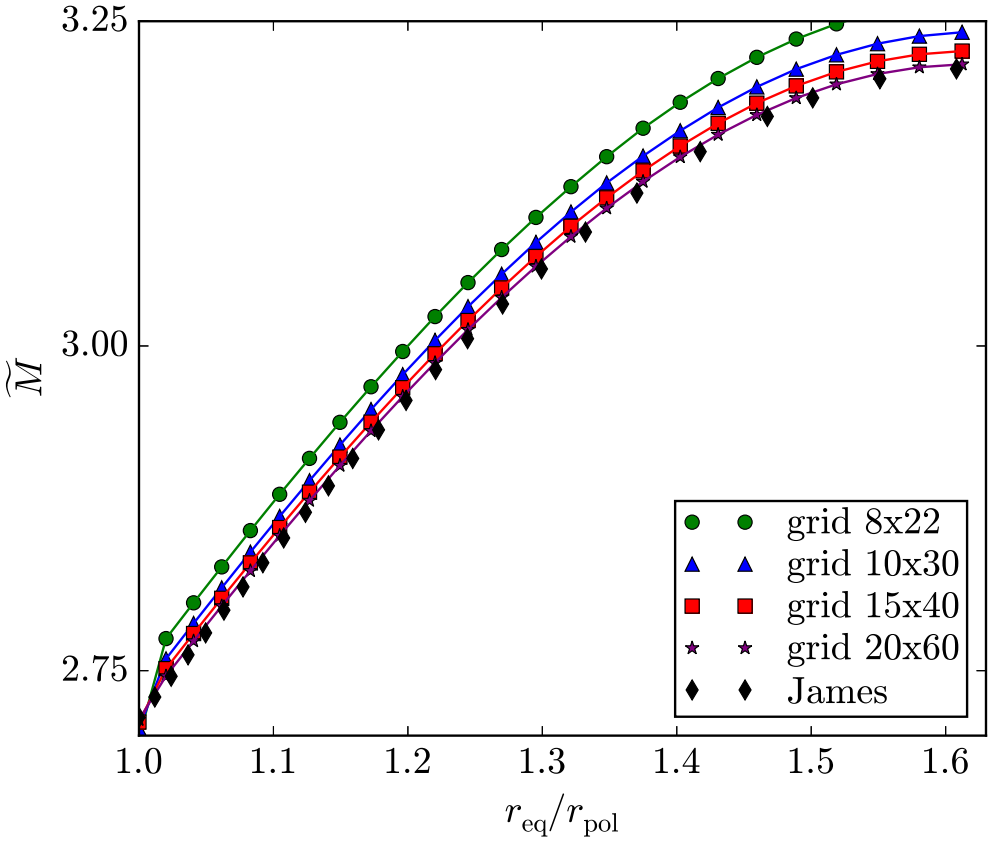

Figure 1.4: Same as Fig. 1.2 but for total mass of configurations M.

Chapter 2

Differential Rotation in Newtonian Gravity

2.1 Introduction

In the previous chapter, we had seen how the problem of equilibrium of rotating polytropes is formulated and solved in the case of uniform rotation, i.e. when the angular velocity is constant all over the entire configuration. Intuitively this case of rigid rotation is not very likely when one considers a rotating gas. On the other hand, a more realistic situation may be represented by the introduction of differential rotation, in which the angular velocity decreases together with the density going from the centre of configuration towards the surface. Introducing such a rotation law makes the problem more complicated. In effect to express the centrifugal potential term, given by Eq. (1.9), one should compute explicitly a non-trivial integral in cylindrical coordinates although the entire problem is stated in spherical ones. Of course, if an exact analytical integral exists, the procedure is easy to be done, with a straightforward substitution in the formula for centrifugal potential of this kind

$$\varpi = r\sin(\theta), \tag{2.1}$$

where ϖ is the cylindrical radius (distance from the rotation axis), r is the spherical radius (i.e. distance from the center) and θ is the angle between the axis of rotation and the radius vector (i.e. the polar angle), exactly with same notations of Chap. 1. However if an exact integral doesn't exist, the centrifugal potential term should be obtained numerically integrating over cylindrical radius and only then using substitution in Eq. (2.1). Probably because of this main reason, in literature one may find different approaches to differential rotation, but when one considers a numerical approach, usually the differential rotation law is such that Eq. (1.9) can be solved analytically (e. g. this is the case presented together with the method in [24]).

In the present chapter we will review some choices for several rotation laws, namely the ones in [61] and [24]. After that, we will compare results obtained through our C code (see App. A) with the ones of ref. [24]. Finally, a new two free parameters differential rotation law will be introduced in the code and a study will be

performed in order to understand where in the parameters space stable equilibrium configurations are admitted.

2.2 Preliminaries

The first study of differentially rotating polytropes was probably due to Stoeckly in [61], where, starting from the study by Jeans and James for uniformly rotating configuration, he reformulated the problem taking into account a non-uniform rotation law. In this paper, he outlined that an important condition which the chosen rotation law should satisfy is the stability condition against axisymmetric perturbation, provided by the Solberg-Høiland criterion,

$$\frac{\partial\left[\varpi^2\Omega(\varpi)\right]}{\partial\varpi} > 0, \tag{2.2}$$

where, as in Chap. 1, ϖ is the cylindrical radius, which states that the angular momentum per unit mass $\varpi^2\Omega$ must increase outwards (c.f. [39], [63] and [61]), i. e. going from the pole to the equator. Thus, when choosing a rotation law, one must always verify that Eq. (2.2) is satisfied in order to guarantee stability.

Returning now to our case of a rotating polytropic star, for sake of example, the differential rotation law in [61], is taken of the following form

$$\Omega(\varpi) = \Omega_{\rm c} e^{-\frac{c\varpi^2}{r_{\rm e}{}^2}}, \tag{2.3}$$

where $\Omega_{\rm c}$ is the central value of angular velocity, c is the free parameter for differential rotation and $r_{\rm e}$ is the equatorial radius. In order to satisfy Eq. (2.2), the parameter c should be chosen such that $\frac{c\varpi^2}{r_{\rm e}{}^2} < 1$. Once this condition is satisfied, the centrifugal potential in the interior of configuration can be written in the following way:

$$\Phi_c(\varpi) = \Phi_c(r,\theta) = \int_0^{r\sin(\theta)} \varpi'\Omega^2(\varpi')d\varpi' = \frac{\frac{1}{2}{\Omega_{\rm c}}^2}{2b}\left[1 - e^{-2br^2\sin^2(\theta)}\right], \tag{2.4}$$

where $b \equiv \frac{c}{r_{\rm e}{}^2}$ and (r,θ) are the spherical coordinates, such that $\varpi = r\sin(\theta)$.

In the next section we will review differential rotation laws presented in [24] and the formula for their respective centrifugal potential.

2.3 Already Known Differential Rotation Laws

Apart from the aforementioned differential rotation law presented by Stoeckly in [61], Eriguchi and Mueller in [24] adopted two kinds of differential rotation laws, where angular velocity decrease as $\frac{1}{r^2}$ or as $\frac{1}{r}$ going from the centre towards the surface of configuration. These two differential rotation laws may be written as:

$$\Omega(\varpi) = \frac{\Omega_{\rm c}}{\left(1 + \frac{\varpi^2}{A^2}\right)}, \tag{2.5}$$

$$\Omega(\varpi) = \frac{\Omega_c}{\left(1 + \frac{\varpi}{A}\right)}, \tag{2.6}$$

where same notation of Chap. 1 is used and A is a parameter for differential rotation, which can be written as

$$A = a r_s, \tag{2.7}$$

with a being a numerical parameter and r_s the radius of the corresponding spherical configuration, which as described in Sec. 1.2.5 depends only on the solution of Lane-Emden's equation. As one can notice from Equations (2.5) and (2.6) both laws tend to represent rigid rotation for large values of parameter A (i.e. precisely for values of $a = 2.0$ or greater), while for small values of A (i.e. for a of the order of 10^{-2}), the first approaches to rotation law with constant specific angular momentum and the second to one with constant rotational velocity (see ref. [24]).

From the choice of non-uniform rotation law, of course, the centrifugal potential terms differ from the ones used in Equations (1.20 – 1.23), and from Equations (2.5) and (2.6) the following expressions for centrifugal potential hold respectively:

$$\Phi_c = \frac{A^2 {\Omega_c}^2}{2\left(1 + \frac{\varpi^2}{A^2}\right)}, \tag{2.8}$$

$$\Phi_c = -A^2 {\Omega_c}^2 \left\{ \frac{1}{1 + \frac{\varpi}{A}} + \log(1 + \frac{\varpi}{A}) \right\}, \tag{2.9}$$

in fact the integral of Eq. (1.9) can be analytically done, thanks to these choices.

2.4 Implementation

In order to test our code, we decided to use the rotation law given by Eq. (2.5) to compute some sequences of equilibrium configurations, with the purpose of comparing our results with ones given in [24].

2.4.1 Stability

Rapidly performing the analysis described in Sec. 2.2 for rotation law given in Eq. (2.5), we get

$$\frac{\partial\left[\varpi^2 \Omega(\varpi)\right]}{\partial \varpi} = \frac{2\varpi A^4 \Omega_c}{\left(A^2 \varpi^2\right)^2}, \tag{2.10}$$

thus, the stability condition given in Eq. (2.2) is always fulfilled, provided that $\Omega_c > 0$.

2.4.2 Stopping Condition

Regarding the stopping condition to control during the construction of a sequence of configurations with increasing axis ratio, taking into account the aforementioned rotation law, it is worth noting that the one given in Eq. (1.46) only refers to the case in which at a certain point mass could shed from equator of the rotating star. In fact, as reported in [24], sequences of configurations with rotational parameter $A = 2.0r_{\rm s}$ (thus with more rigid rotation) behave just like the case with uniform rotation: the configuration distorts itself from spherical to spheroidal until gravitational force at the equator is balanced by centrifugal force. On the other hand for smaller values of A (i. e. $A = 0.2r_{\rm s}$ and $A = 0.02r_{\rm s}$), a cusp is firstly formed in the polar region, and then the centre of configuration doesn't coincide with the point of maximum density, which tells us that a ring-like structure appears. In the latter case, the sequence of configurations is computed up to a certain number of configurations (usually 100 configurations for a step in the axis ratio of 1.05).

2.5 Known Reluts

To test our results with ones from literature, we will refer to tables presented in [24], for the so-called *j-const* rotation law, i.e. the one of Eq. (2.5), for choices of parameters $A = 2.0r_{\rm s}$, $0.2r_{\rm s}$ and $0.02r_{\rm s}$, which are reported in the following Tables 2.1, 2.2 and 2.3. Precisely the definition of quantities reported in each table is the same given in Sec. 1.2.7, with the exception of j^2, the squared dimensionless angular momentum, which in paper [24] in defined as

$$j^2 = \frac{\widetilde{J}^2}{(4\pi)^{\frac{4}{3}}\,\widetilde{M}^{\frac{10}{3}}\sigma_{\max}{}^{\frac{n}{3}}}, \tag{2.11}$$

where dimensionless quantities are again defined in Sec. 1.2.7 and $\sigma_{\max}$ is the maximum value of the dimensionless density of configuration.

Table 2.1: Results for numerical computations taken from [24]
$A = 2.0r_{\rm s}$

$\frac{\xi_{\rm eq}}{\xi_{\rm pol}}$	j^2	$\frac{T}{\lvert W\rvert}$	$\frac{U}{\lvert W\rvert}$
1.050	$2.966e-4$	$9.183e-3$	0.4911
1.276	$1.622e-3$	$4.530e-2$	0.4549
1.551	$3.0228e-3$	$7.689e-2$	0.4233
1.886	$4.244e-3$	$9.991e-2$	0.4003
2.292	$4.766e-3$	$1.086e-1$	0.3916

Table 2.2: Results for numerical computations taken from [24]
$A = 0.2r_{\mathrm{s}}$

$\frac{\xi_{\mathrm{eq}}}{\xi_{\mathrm{pol}}}$	j^2	$\frac{T}{\lvert W\rvert}$	$\frac{U}{\lvert W\rvert}$
1.050	$1.521e-4$	$7.236e-3$	0.4931
1.340	$9.305e-4$	$4.323e-2$	0.4570
1.710	$1.789e-3$	$7.514e-2$	0.4251
2.183	$2.616e-3$	$1.022e-1$	0.3991
2.786	$3.446e-3$	$1.220e-1$	0.3783
3.386	$4.089e-3$	$1.356e-1$	0.3647
4.538	$5.037e-3$	$1.528e-1$	0.3475
5.792	$5.840e-3$	$1.651e-1$	0.3352
7.392	$6.639e-3$	$1.762e-1$	0.3240
9.434	$7.446e-3$	$1.864e-1$	0.3138
12.04	$8.290e-3$	$1.961e-1$	0.3042

Table 2.3: Results for numerical computations taken from [24]
$A = 0.02r_{\mathrm{s}}$

$\frac{\xi_{\mathrm{eq}}}{\xi_{\mathrm{pol}}}$	j^2	$\frac{T}{\lvert W\rvert}$	$\frac{U}{\lvert W\rvert}$
1.050	$3.393e-6$	$4.472e-4$	0.4998
1.710	$5.310e-5$	$6.282e-3$	0.4940
2.786	$1.253e-4$	$1.302e-2$	0.4872
3.920	$1.896e-4$	$1.808e-2$	0.4822
6.385	$3.098e-4$	$2.596e-2$	0.4743
9.434	$4.392e-4$	$3.331e-2$	0.4670
14.64	$6.332e-4$	$4.305e-2$	0.4572
25.03	$9.597e-4$	$5.705e-2$	0.4432
38.83	$1.315e-3$	$6.996e-2$	0.4303
60.24	$1.771e-3$	$8.428e-2$	0.4160
89.01	$2.277e-3$	$9.806e-2$	0.4022

2.5.1 Comparison

We will now turn to compare results of our computations with values given in Tables 2.1, 2.2 and 2.3, plotting each quantity as a function of axis ratio in the subsequent figures. Calculations as already explained in Sec. 1.3.1, have been performed using several numerical grid, to obtain different resolution in configurations . Namely, we considered $N_T \times N_R = 8 \times 22$ (green dots in figures), 12×34 (blue triangles), 15×40 (red squares) and 20×60 (purple stars) and in each plot results by Eriguchi and Mueller are represented by black diamonds. In addition, three different values of the rotational parameter have been considered, analogously to results

presented in Tables 2.1, 2.2 and 2.3. The step factor for axis ratio change has been taken as 1.05 for the construction of each sequence.

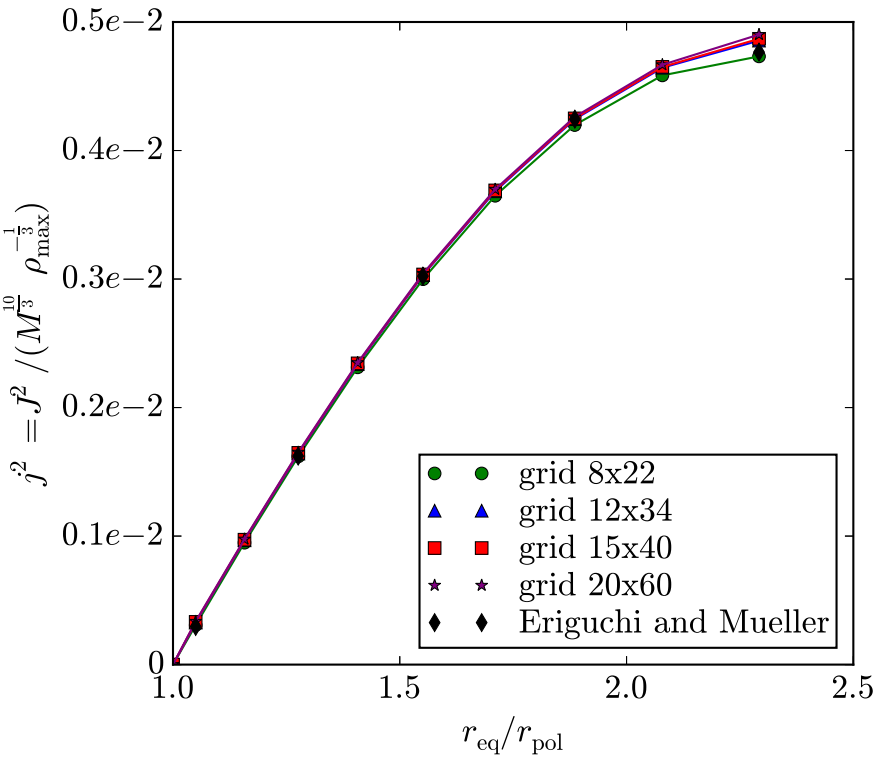

Figure 2.1: Comparison of results obtained with our own implementation of method by Eriguchi, Mueller and results from Tab. 2.1 (thus taking $A = 2.0 r_{\rm s}$), where dimensionless squared angular momentum j^2 is plotted as a function of axis ratio $\frac{r_{\rm eq}}{r_{\rm pol}}$.

Again a grid dependent behaviour arises from computation: in effect we could not obtain the entire sequences of configurations for all grids considered. For example, if one takes a grid of 10×30 points, the code reaches an error in the Newton-Raphson method. This instability of the code is evidently due to the chosen routine for the implementation of Newton-Raphson method (see discussion at the end of Sec. 1.2.4; for the routine refer to [55]). In addition, again considering a "too" small step factor for axis ratio from one configuration to the subsequent, some numerical oscillations occur, most of all in the first sequence of configurations , that is the one in which $A = 2.0 r_{\rm s}$ (or in other words the most rigid one).

On the other hand, from Fig.s 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8 and 2.9 the reader can easily notice that deviations from results of paper [24] are very small thus we may conclude again that the implementation works fine. Moreover, deviations between results obtained with grid 15×40 (red squares) and 20×60 (purple stars) could not be observed, thus, suggestion given in [24] to use a 15×40 grid to obtain accurate results is confirmed.

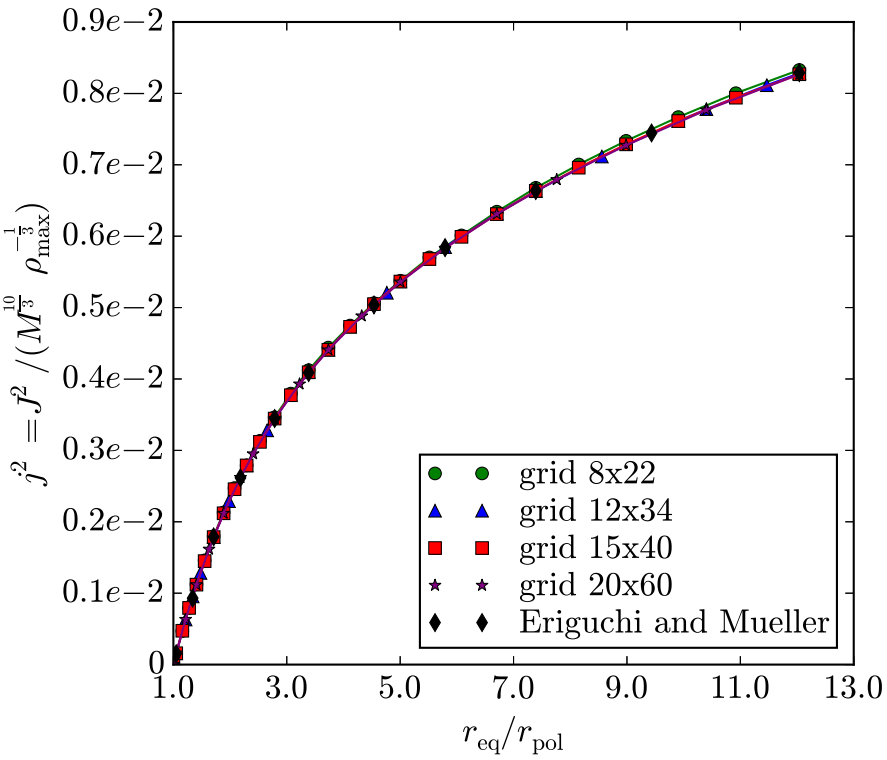

Figure 2.2: Same as Fig. 2.1 but in comparison with results from Tab. 2.2 (thus taking $A = 0.2 r_{\rm s}$).

2.6 Multi-parametric Differential Rotation Law

Having confirmed in previous sections that the code which has been written starting from the method in [24] is well calibrated, one could be interested in inserting a multi-parametric differential rotation law, which should allow treating several kind of rotations. For this reason, we will introduce a two-parameter law, taken by [17], which can be written in the following way:

$$\Omega(\varpi) = \Omega_{\rm c} \frac{e^{-B\varpi^2}}{1 + \left(\frac{\varpi}{A}\right)^2}, \tag{2.12}$$

where $\Omega_{\rm c}$ is the central value of angular velocity, ϖ is the distance from the axis of rotation, B and A free parameters. The dimensionless form of Eq. (2.12) is the same given in Eq. (1.19d), provided that parameters B and A has dimensions of $\frac{1}{r^2}$ and r respectively, thus it is useful to impose

$$B = \frac{b}{{r_{\rm e}}^2}, \tag{2.13}$$

$$A = a r_{\rm s}, \tag{2.14}$$

being $r_{\rm e}$ the equatorial spherical coordinate radius and $r_{\rm s}$ the radius of spherical configuration, just like in [61] (for parameter B) and [24] (for parameter A). Note

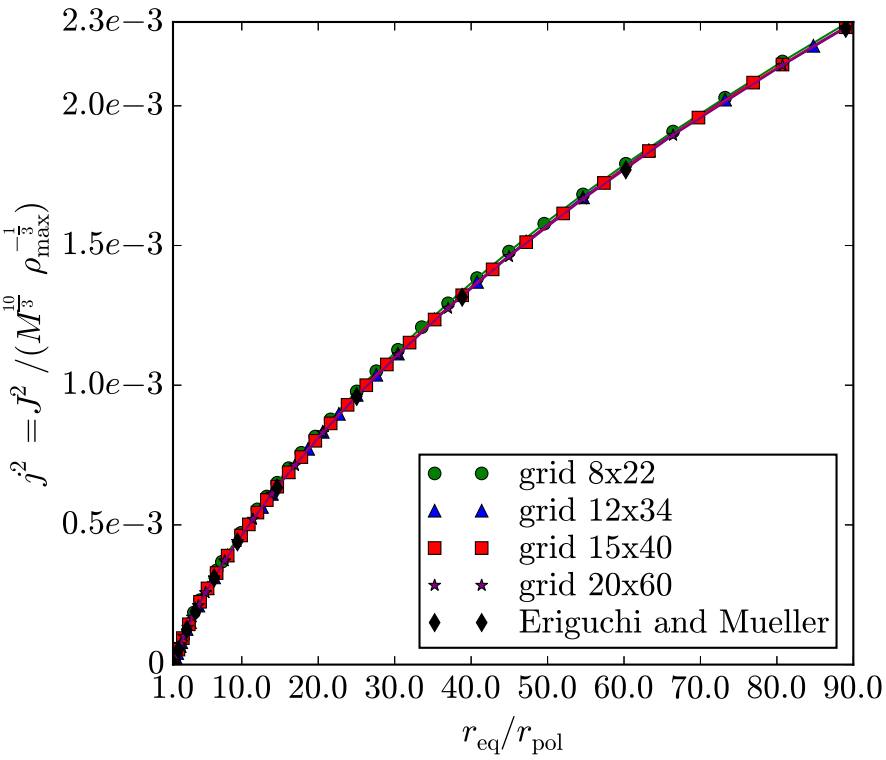

Figure 2.3: Same as Fig. 2.1 but in comparison with results from Tab. 2.3 (thus taking $A = 0.02 r_{\rm s}$).

that in case $b = 0$ the rotation law reduce simply to the one of Eq. (2.5) thus, we indeed have a method to test if implementation is performed correctly.

2.6.1 Note on Centrifugal Potential

The main difference in choosing differential rotation to be performed via law given by Eq. (2.12), is that the integral of Eq. (1.9) cannot be performed analytically. In general, we know that the primitive of a function is defined up to a constant in the integration variable, thus if we denote with $F(\varpi)$ a particular primitive for function $\varpi\Omega^2(\varpi)$, it holds

$$\Phi_c(\varpi) = F(\varpi) - F(0) + C, \tag{2.15}$$

being C this integration constant. How to find this constant? A straightforward condition could be that

$$\lim_{\varpi \to +\infty} \Phi_c(\varpi) = 0, \tag{2.16}$$

thus, if one could numerically perform integration of right-hand side of Eq. (1.9) up to infinity and indicates this value with Φ_∞, from Eq. (2.16) is clear that

$$C = -\Phi_\infty, \tag{2.17}$$

and in particular in the centre of configuration

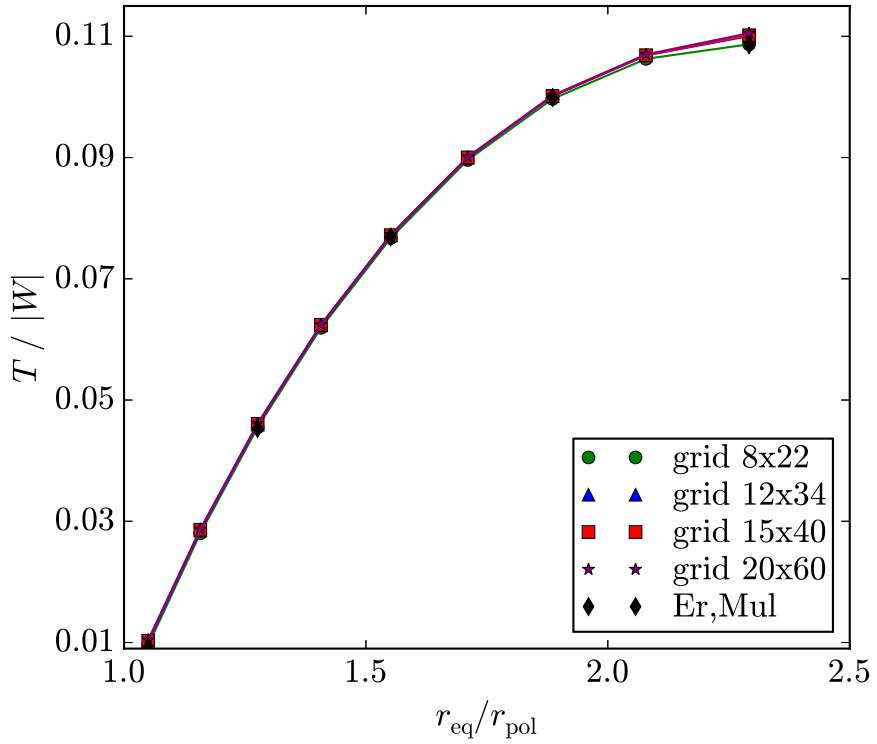

Figure 2.4: Comparison of results obtained with our own implementation of method by Eriguchi, Mueller and results form Tab. 2.1 (thus taking $A = 2.0 r_s$), where ratio between rotational kinetic energy and gravitational potential energy is plotted as a function of axis ratio $\frac{r_{\rm eq}}{r_{\rm pol}}$.

$$\Phi_0 = \Phi_c(0) = C. \tag{2.18}$$

For numerical implementation, we decided to define an artificial grid in the cylindrical coordinate radius direction and perform a numerical integration via the trapezoidal method to obtain values of centrifugal potential in each grid-point defined to solve the system of equation. Explicitly, for each grid-point, let's fix spherical dimensionless coordinates (ξ, θ). Thus, the maximum dimensionless cylindrical radius is clearly $\widetilde{\varpi}_{max} = \xi \sin(\theta)$ and the artificial grid and integration weights (just like in the case of angular grid defined in Sec. 1.2.3) can be defined as

$$\widetilde{\varpi}_k = \widetilde{\varpi}_{max} \frac{(k-1)}{(N_{cyl}-1)}, \text{ for } k = 1, \ldots, N_{cyl}, \tag{2.19}$$

$$\Delta\widetilde{\varpi}_k = \begin{cases} \frac{\widetilde{\varpi}_{max}}{2(N_{cyl}-1)} & \text{for } k = 1, \ldots, N_{cyl} \\ \frac{\widetilde{\varpi}_{max}}{(N_{cyl}-1)} & \text{for every other } k, \end{cases} \tag{2.20}$$

for a fixed large number of points (e.g. $N_{cyl} = 1000$). To find the integration constant (and thus the value of centrifugal potential in the centre of configuration) one should compute Φ_∞ and this can be done in an analogous way, considering ,for example, $\widetilde{\varpi}_{max} = \widetilde{\varpi}_\infty = 100.0 \approx \infty$ and $N_{centre} = 10000$, taking into account that dimensionless equatorial radius is usually less that 10.0 (indeed it has order of unity).

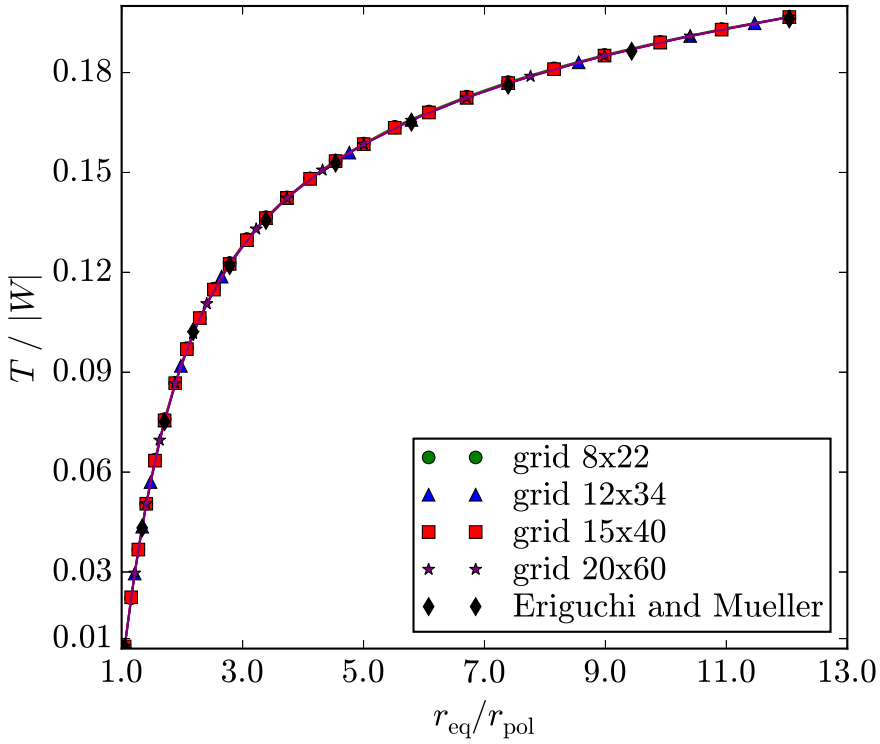

Figure 2.5: Same as Fig. 2.4 but in comparison with results from Tab. 2.2 (thus taking $A = 0.2 r_s$).

With such numerical grid in the cylindrical radial coordinate, the aforementioned numerical integration for the centrifugal potential is able to produce same results as the ones presented in Fig. 2.1–2.9, if one considers in Eq. (2.12) parameter $b = 0$ and same values of parameter a as indicated in the respective figures. This allows us to treat in a general numerical way every kind of differential rotation law, without assuming it to admit an analytical form for centrifugal potential.

The same care should be taken for the calculation of derivative of Eq. (2.2) to check if the Solberg-Høiland criterion for stability is satisfied and this is what we will point out in the next section.

2.6.2 Note on the Stability Analysis

Using the two-parameter rotation law of Eq. (2.12) in Eq. (2.2), it analytically gives the following condition:

$$-\frac{2\Omega_c \varpi e^{-B\varpi^2} A^2 \left(B\varpi^2 A^2 + B\varpi^4 - A^2\right)}{\left(A^2 + \varpi^2\right)^2} > 0, \tag{2.21}$$

thus clearly

$$\left(B\varpi^2 A^2 + B\varpi^4 - A^2\right) < 0, \tag{2.22}$$

provided $A > 0$. Recalling now Equations (2.13) and (2.14) it is clear that the values of B and A can be defined only after the numerical model has been obtained,

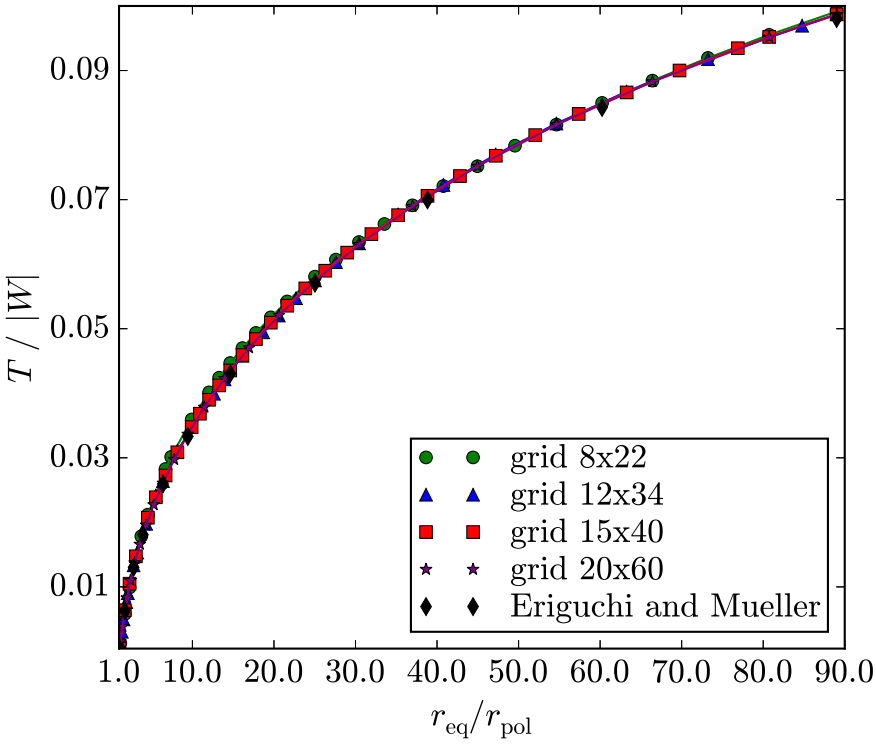

Figure 2.6: Same as Fig. 2.4 but in comparison with results from Tab. 2.3 (thus taking $A = 0.02 r_s$).

i.e. after convergence of N-R method. Thus to be sure that the configuration produced has physical meanings the first control to adopt is given by Eq. (2.22).

Now, in order to make the code the most portable one, it can be useful to compute the derivative in Eq. (2.2) numerically. This can be done using a three point approximation with forward (or backward) differences for numerical derivatives. In general, giving a function $f(x)$ to derive numerically, with points spaced of a certain value $h = x_{i+1} - x_i$ the three points approximation comes out almost straightforwardly from Taylor's expansion of function f computed in points x, $x+h$ and $x+2h$ (or respectively in $x-h$ and $x-2h$ in the backward case), and it reads

$$f'(x) = \frac{1}{2h}\left[-3f(x) + 4f(x+h) - f(x+2h)\right] + O(h^2), \tag{2.23}$$

in the forward case, or

$$f'(x) = \frac{1}{2h}\left[3f(x) - 4f(x-h) + f(x-2h)\right] + O(h^2), \tag{2.24}$$

in the backward one. In our case function $f \equiv \varpi^2\Omega$, integration variable $x \equiv \varpi$, and using the artificial grid introduced in Sec. 2.6.1 the spacing $h \equiv \frac{\varpi_{\max}}{(N_{cyl}-1)}$, where it is worth noting that $\varpi_{\max} = r_{\rm surf}\sin(\theta)$. However, instead of integrating function $\varpi^2\Omega$, we can also take advantage of a useful relation which can be found in Ref. [39]. In there, one can find that the Solberg-Høiland criterion of Eq. (2.2), in the

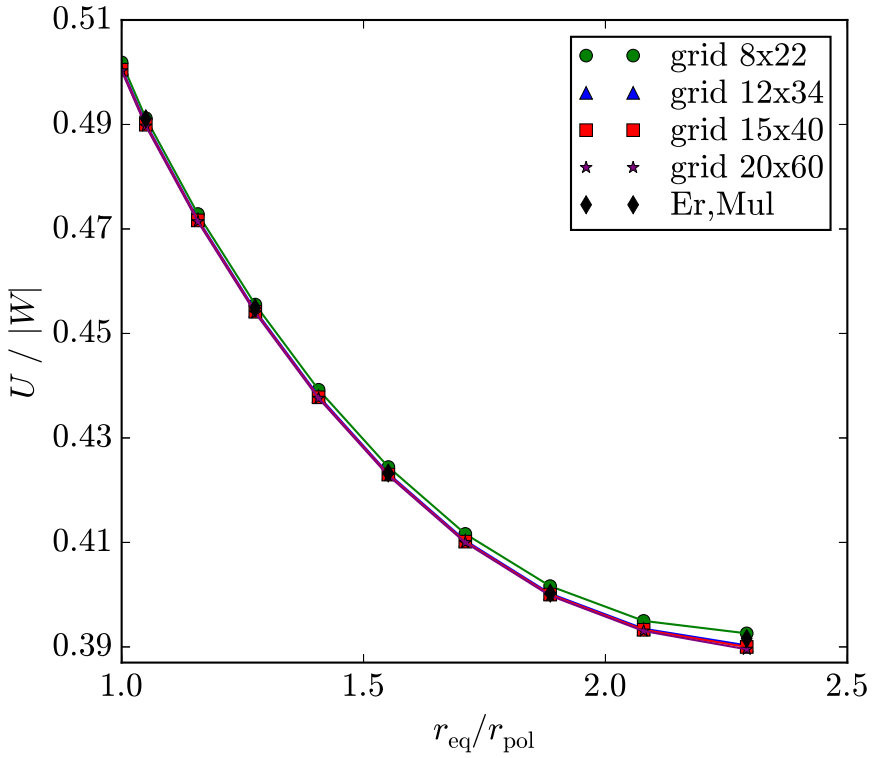

Figure 2.7: Comparison of results obtained with our own implementation of method by Eriguchi, Mueller and results form Tab. 2.1 (thus taking $A = 2.0r_{\rm s}$), where ratio between thermal energy and gravitational potential energy is plotted as a function of axis ratio $\frac{r_{\rm eq}}{r_{\rm pol}}$.

case of barotropes (thus also for polytropes), where $\Omega = \Omega(\varpi)$, taking into account surfaces of constant entropy, can be written as

$$2\Omega(\varpi)\left(2\Omega(\varpi) + \varpi\frac{\partial\Omega(\varpi)}{\partial\varpi}\right) > 0, \tag{2.25}$$

thus, we can numerically differentiate simply function Ω with Equations (2.23) and (2.24), checking then if the condition of Eq. (2.25) is satisfied in every grid-point.

At this point, it could be interesting to understand what is the physical meaning of the violation of the stability criterion given by Eq. (2.2). For a complete derivation of this criterion, the reader should refer to [45] but to have an intuitive idea of the approach, imagine that a fluid element with angular momentum per unit mass j_0 is displaced from his position p_0 in the direction perpendicular to the rotation axis (thus in the cylindrical radius direction) to an outer one p_1, where other fluid elements are characterized by angular momentum per unit mass j_1. Then if $j_0 < j_1$, the displaced element will be taken back to the equilibrium position, while in the opposite case it will be driven farther, thus, the configuration would not be stable. For this reason, it is important to see what happens to the angular momentum per unit mass distribution for different choices of rotation parameter. In Fig. 2.10 the angular momentum per unit mass distribution along the distance from rotation axis direction is plotted for three configurations , obtained with same differential rotation exponential parameter $b = 0.128$ and axis ratio $\frac{r_{\rm eq}}{r_{\rm pol}} = 1.05$ but with different choices

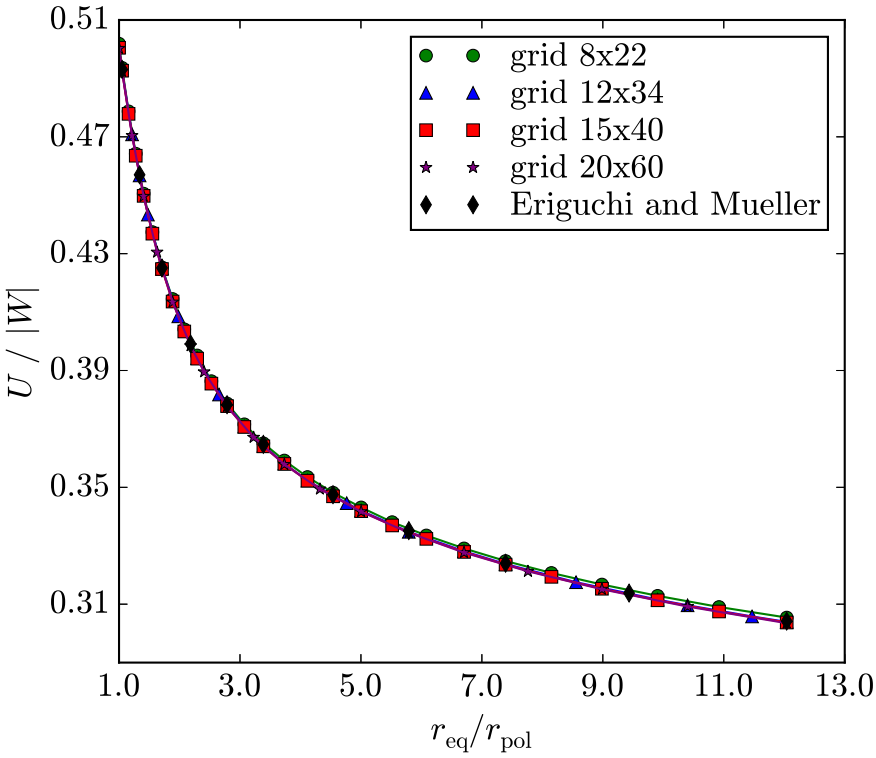

Figure 2.8: Same as Fig. 2.7 but in comparison with results from Tab. 2.2 (thus taking $A = 0.2r_s$).

of parameter a (namely $0.02, 1.12, 2.00$). It can be seen that for sequences represented by dashed and dash-dotted curves (respectively with $a = 1.12$ and $a = 2.00$) the condition of Eq. (2.2) is obviously satisfied, leading to stable configurations and allowing the entire sequence to be constructed (as the reader will see in Figures of next section), while for sequence represented by solid curve, obtained with $a = 0.02$, the angular momentum per unit mass distribution looks like a constant one. In reality what seems to be constant, is instead very slowly decreasing, and in fact, this is a case where the stability condition is violated, thus the sequence is stopped.

In Fig. 2.11 we show the surface plots of sample unstable configurations. Basically we let the code go further with axis ratio increasing beyond the violation of Solberg-Høiland criterion keeping fixed for the sake of example $a = 0.02$ and $b = 0.128$ and indicating the value of axis ratio in parenthesis over each plot. As the reader can see the configuration seems to tend to a ring-like one, thus the instability is not well observable from the shape of configuration, although from the aforementioned discussion on the angular momentum distribution it is well understood.

In the next section, we will perform an analysis of the free parameters space to understand where stable configurations are allowed to exist.

2.6.3 Study of the parameters space

Adopting a multi-parametric differential rotation law, such the one given in Eq. (2.12), gives a very detailed control on the way the star rotate and it could be

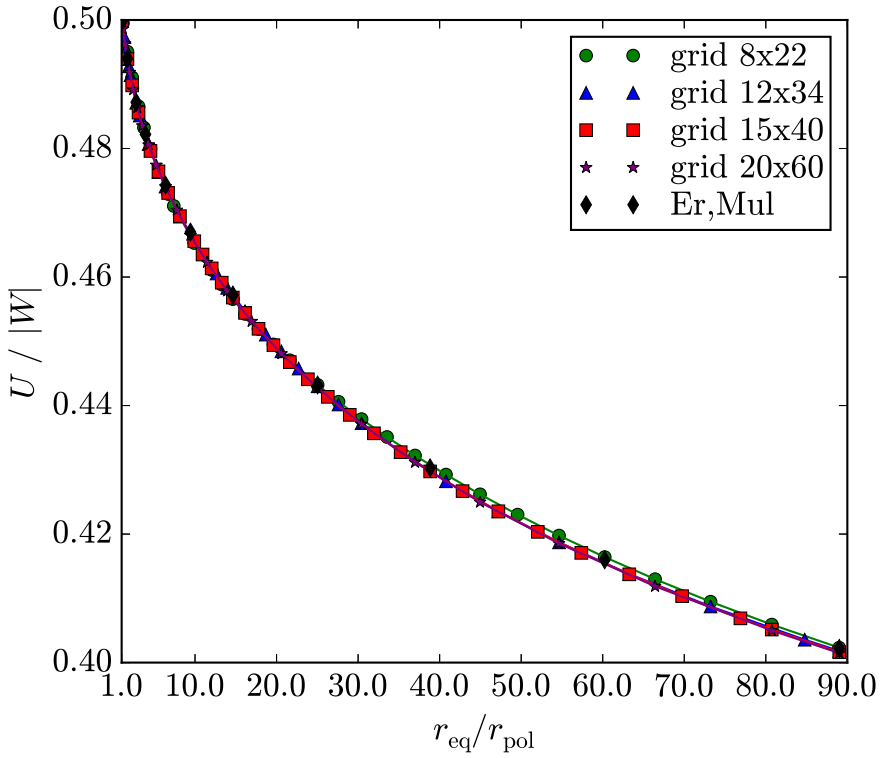

Figure 2.9: Same as Fig. 2.7 but in comparison with results from Tab. 2.3 (thus taking $A = 0.02 r_{\rm s}$).

interesting to understand the variety of figures of equilibrium which could result from different choices of these control parameters. To perform such an analysis, one could use our code to build several equilibrium sequences, each always given by a fixed change in the axis ratio from one configuration to another (see Sec. 1.3), taking into account mass-shedding condition given in Eq. (1.46) and Solberg-Høiland criterion in Eq. (2.25). Thus, we decided to fix initial extrema for parameters value, namely $b_{\rm min}$, $b_{\rm max}$, $a_{\rm min}$ and $a_{\rm max}$ and to compute respectively N_b and N_a linearly equally spaced parameter's value between these extremes. Explicitly, the parameters value are given by the following equation:

$$p_k = p_{\rm min} + \left((p_{\rm max} - p_{\rm min}) \times \frac{(k-1)}{(N_p - 1)} \right) \quad \text{for } k = 1 \dots N_p, \tag{2.26}$$

where with p the parameter's name is intended, namely b or a. After having computed each parameter's value we can start the sequences of configurations to be computed, always starting from the spherical configuration given by Lane-Emden equation's solution, until some kind of instability is reached. However if the configuration is a ring-like one, it is possible that no of the two instability condition could be reached, thus another condition is considered, in which the maximum number of configuration is fixed at 100 configurations (to limit the computation time).

In Fig. 2.12 the kind of instability reached (after the entire sequence is computed) for each choice of the two rotational parameters are represented, with a particular colours convention:

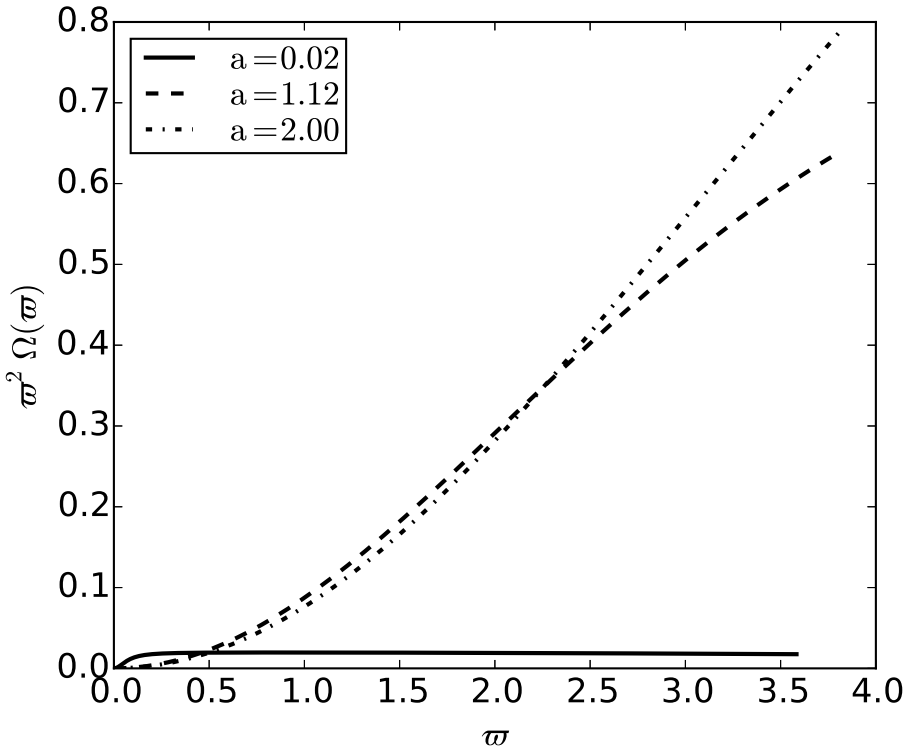

Figure 2.10: Angular momentum per unit mass distribution along distance from rotation axis for three sample sequences obtained with same choice of parameter $b = 0.128$ and axis ratio $\frac{r_{\rm eq}}{r_{\rm pol}} = 1.05$, but different choices of parameter $a = 0.02, 1.12, 2.00$.

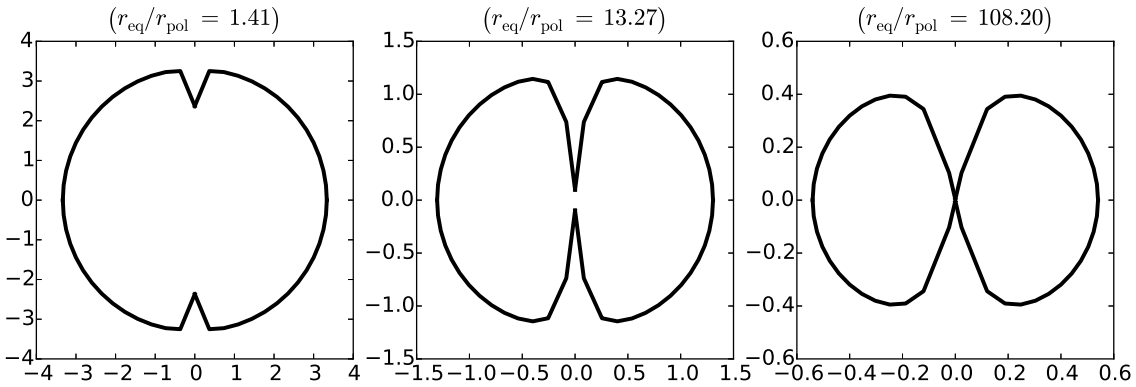

Figure 2.11: Configurations showing surfaces obtained setting $a = 0.02$ and $b = 0.128$ in the differential rotation law given in Equations (2.12), (2.13) and (2.14). However these configurations should be unstable we let the code go further with axis ratio increase, and values of axis ratio $\frac{r_{\rm eq}}{r_{\rm pol}}$ are given between parenthesis over each plot. Grid used for these configurations is 15×40.

- **Green** - mass-shedding instability, i. e. $g_{\text{eq. eff.}} > 0$;
- **Yellow** - violation of Solberg-Høiland criterion;
- **Pale-blue** - ring-like configuration, in which none of the two stability criteria is violated;

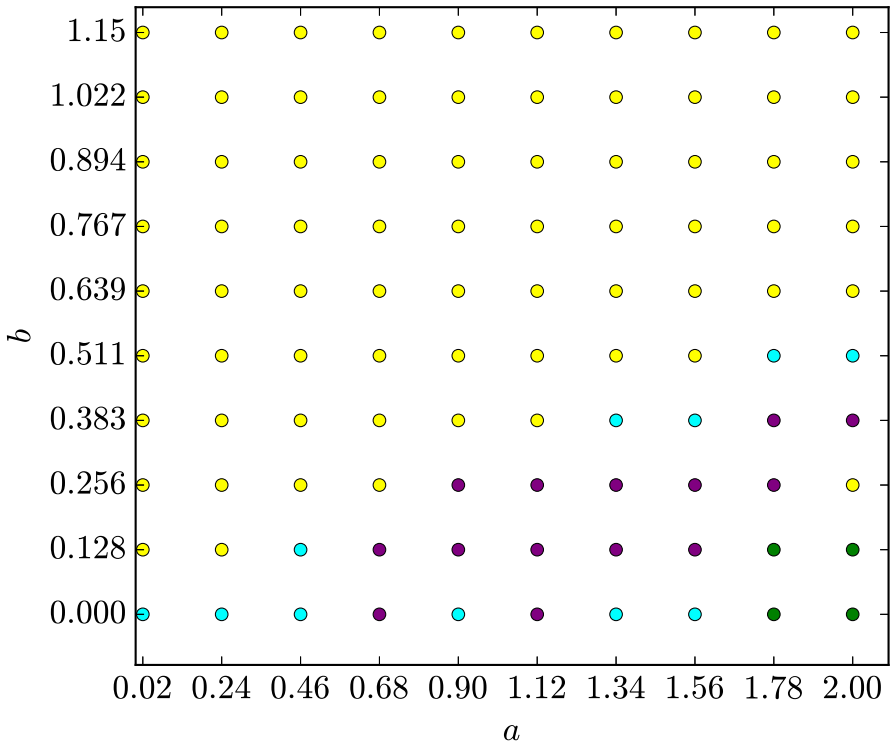

Figure 2.12: Resulting equilibrium configurations for different choices of rotational parameters b and a of Eq. (2.12). Colour convention is as follow: green is for mass-shedding ($g_{\text{eq. eff.}} > 0$), yellow is for Solberg-Høiland criterion violation, pale blue is for ring-like configurations (which after 100 configurations did not violate any stability criterion), purple is for ring-like configuration which reach mass-shedding at equator.

- **Purple** - ring-like configuration in which $g_{\text{eq. eff.}} > 0$.

From this figure, is also clear that, depending on the choice of rotational parameters, different kinds of configurations may be obtained. It could also be interesting to study the parameters' space in three dimensions, e.g. taking into account the maximum value of central dimensionless angular velocity reached in each sequence.

In Fig. 2.13 a three-dimensional histogram is presented. It has been obtained considering in $x-y$ plane the same as Fig. 2.12 and in the vertical axes the maximum dimensionless central angular velocity reached by configurations in each sequence of configurations just before the different kinds of instability occur. In particular, it is worth noting that the value of central angular velocity goes well beyond the value of 0.2 for each sequence in which a ring-like structure appear. This is due to the fact that effective gravity slightly increases in these sequences, although in some cases mass-shedding occurs before the limit of 100 configurations is reached.

In Fig. 2.14, 2.15, 2.16 and 2.17 density distribution of some configurations are plotted, revealing also the shape of configuration. In particular, Fig. 2.14 is obtained through the solution of Lane-Emden equation (1.33) for the construction of the spherical configuration. From this configuration, all the other configurations are obtained via increasing of axis ratio with the method described in Chap. 1 and previous Sections, fixing several values for differential rotation parameters. In

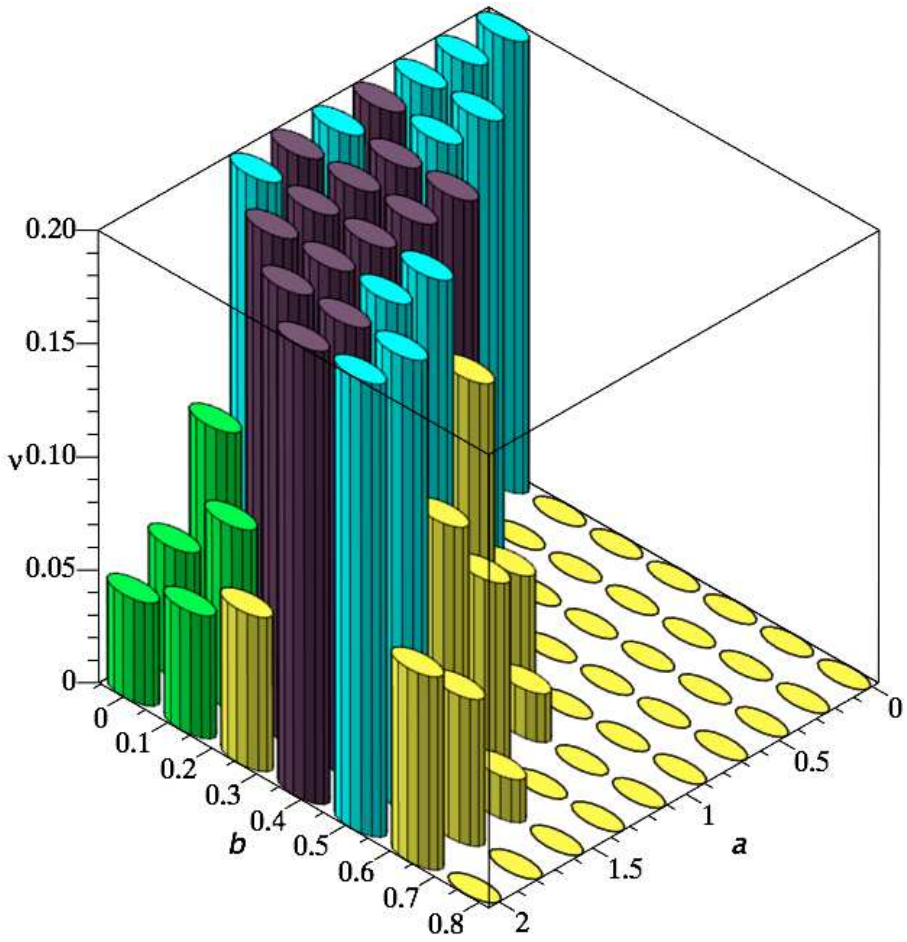

Figure 2.13: Same as Fig. 2.12 where in vertical axes is indicated the maximum value of dimensionless central angular velocity of the last stable configuration for each sequence of configurations computed.

Fig. 2.15, fixing $b = 0.0$ and $a = 2.0$ in Equations (2.13) and (2.14) and increasing the axis ratio value up to $\frac{r_{\rm eq}}{r_{\rm pol}} = 1.71$, produce a well evident oblate structure, in which the equatorial centrifugal force will increase until a mass-shedding instability is reached (see Fig. 2.12). Finally, Fig.s 2.16 and 2.17 belong to the same sequence of configurations , obtained fixing $b = 0.128$ and $a = 0.46$ in Equations (2.13) and (2.14), respectively with $\frac{r_{\rm eq}}{r_{\rm pol}} = 1.22$ and 7.04. In these two figures the formation of a ring-like structure is evident: as the axis ratio increase, mass density maximum transfers from the center of configuration to a certain grid point and this process, with further increasing in axis ratio, will produce a torus (although in present calculation this result can not be seen, as the method is not appropriate for the study of ring-like structures, see [25]).

In Fig. 2.18 we show a sample of possible figures of equilibrium which can be obtained using the differential rotation law of Equations (2.12), (2.13) and (2.14), with different choices of free parameters a and b. In particular, each plot corresponds to the last stable configuration which our code produced, just before the computation had been stopped due to aforementioned possible reasons (namely mass-shedding,

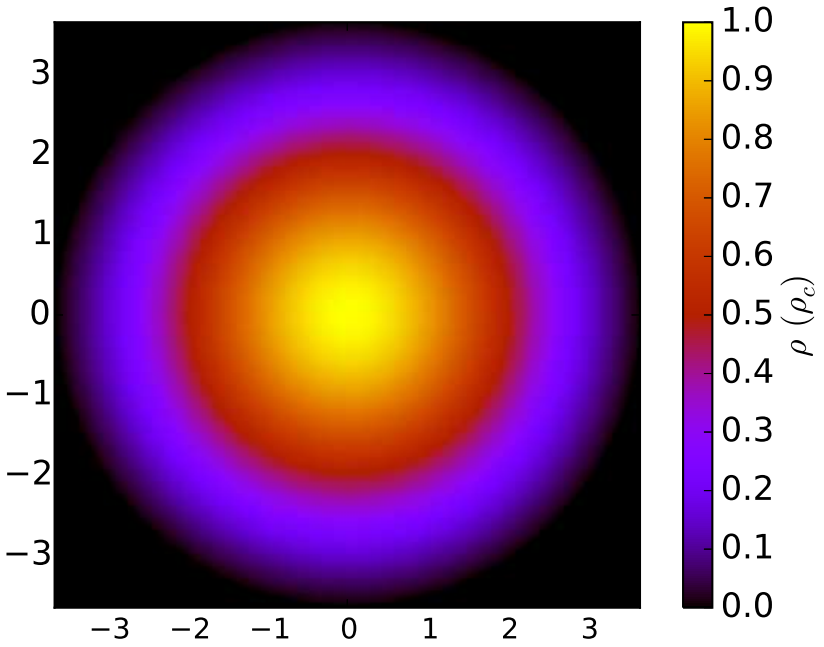

Figure 2.14: Density distribution for spherical configuration obtained via solution of Lane-Emden equation (1.33).

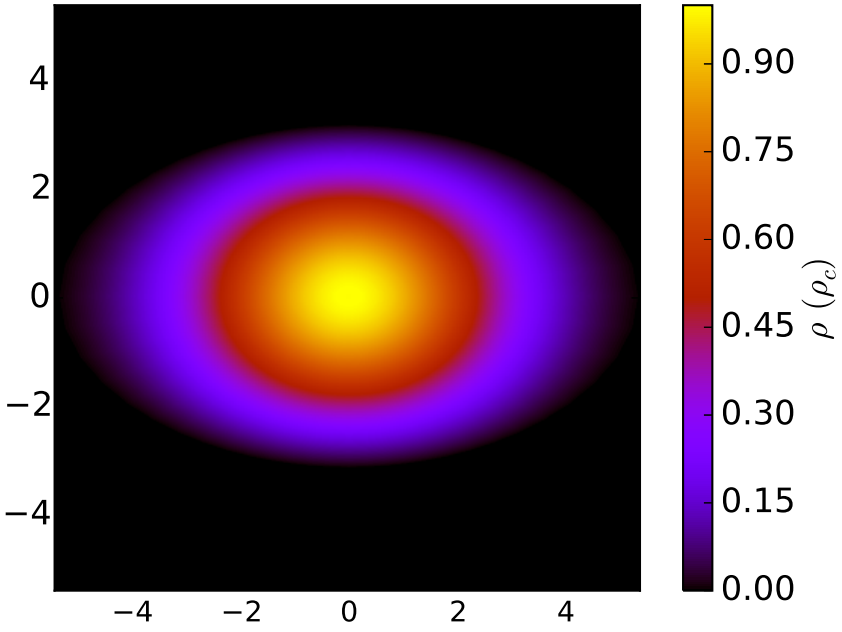

Figure 2.15: Density distribution for an oblate configuration, obtained fixing $b = 0.0$ and $a = 2.0$ in Equations (2.13) and (2.14). The axis ratio value here is $\frac{r_{\rm eq}}{r_{\rm pol}} = 1.71$, thus the oblateness is evident. This kind of configuration will undergo to mass-shedding instability, according to Fig. 2.12.

violation of stability criterion, exceeding in the number of configurations or Newton-

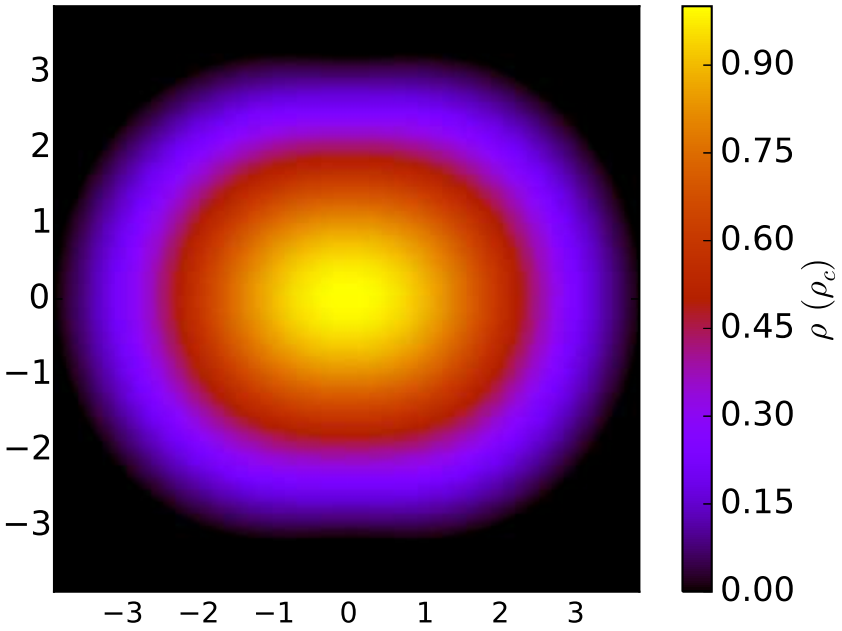

Figure 2.16: Density distribution for a configuration which is starting to be a ring-like one, obtained fixing $b = 0.128$ and $a = 0.46$ in Equations (2.13) and (2.14). The axis ratio value here is $\frac{r_{\rm eq}}{r_{\rm pol}} = 1.22$ and although it is nearly spherical an accurate observation shows that the differential rotation parameters' value begin to create a ring-like structure.

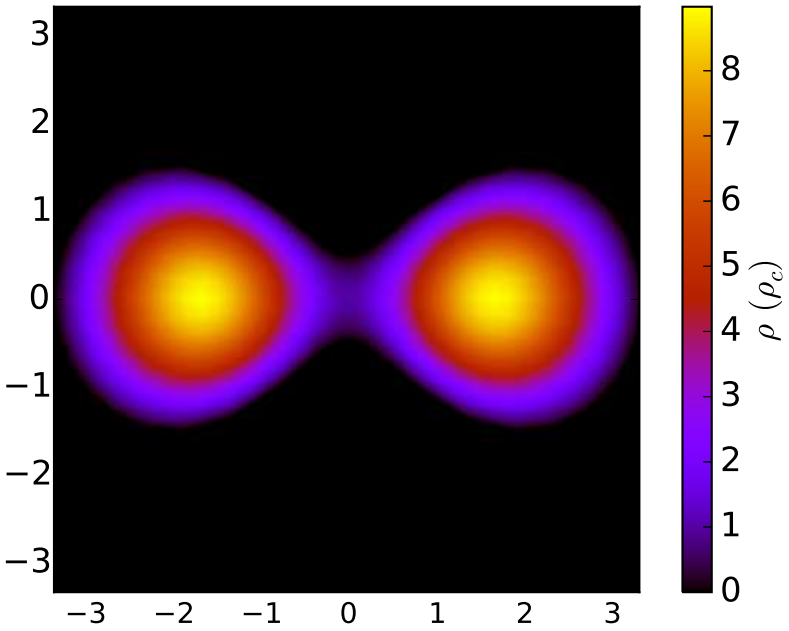

Figure 2.17: Density distribution for a configuration in which the ring-like structure appear evident, obtained fixing $b = 0.128$ and $a = 0.46$ in Equations (2.13) and (2.14) (as in Fig. 2.16). The axis ratio value here is $\frac{r_{\rm eq}}{r_{\rm pol}} = 7.04$ and here the ring like structure is evident.

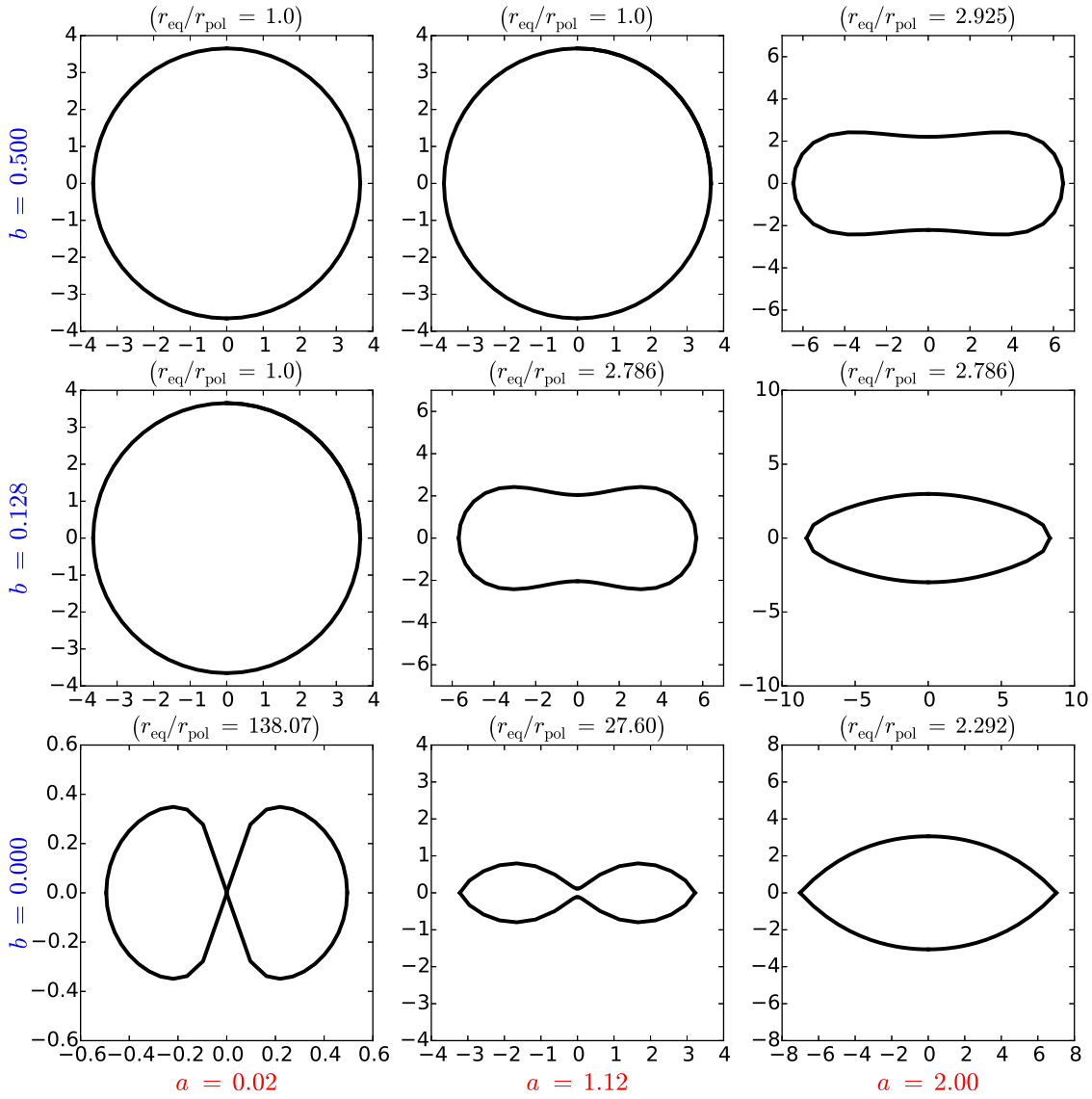

Figure 2.18: Sample of shape of configurations for different choices of parameters of differential rotation. Precisely the figures show the last stable configurations just before computation is stopped because of aforementioned conditions (namely mass-shedding, violation of Solberg-Høiland criterion or error in N-R method's routine). For each row and column of figures, values of parameters a and b of Eq. (2.13) and (2.14) are indicated respectively with red and blue colours.

Raphson's routine failure to converge). The axis ratio $\frac{r_{eq}}{r_{pol}}$ of each configuration is reported in parenthesis above each plot. A spherical figure means that the particular choice of rotational parameter leads immediately to some kind of instability, for which kind the reader should refer to Fig. 2.12. With this argumentation the kind of figure of equilibrium which can be obtained and the analysis of the free parameters space should be complete.

Before concluding this section, an important consideration must be done: remember that our code, as reported in Sec. 1.2.4 contains some routines from ref. [55]. We found that these routines are not stable, in fact, the N-R method so im-

plemented could lead to local minima of the system, instead to the absolute one. For this reason and because of the strong grid-dependence of Eriguchi and Mueller method, in some cases, the computation could be interrupted for some particular choices of rotational parameters before one of the stopping conditions occurs. How can be possible to recover these missing result? In this cases the grid-dependence comes to our help, in fact, one could perform some cross-checks between results obtained using several grids and this guarantees the accuracy of results presented. In particular, for results presented in this section, we compared results obtained with 8×22 and 15×40 grids.

To conclude we can also report in following tables, the physically relevant values of some of the sequences computed, exemplifying the different configurations of equilibrium obtained.

Table 2.4: Results of our numerical computations for an almost uniformly rotating sequence of equilibrium configurations obtained fixing $b = 0.0$ and $a = 2.0$.
This sequence will end in mass-shedding.

$\frac{\xi_{\rm eq}}{\xi_{\rm pol}}$	j^2	$\frac{T}{\lvert W\rvert}$	$\frac{U}{\lvert W\rvert}$	$\widetilde{M}$
1.000	0.000000	0.000000	0.500	2.710
1.158	$0.972e-3$	$0.2861e-1$	0.472	2.994
1.276	$0.165e-2$	$0.4607e-1$	0.454	3.156
1.407	$0.234e-2$	$0.6242e-1$	0.438	3.323
1.551	$0.304e-2$	$0.7720e-1$	0.423	3.488
1.710	$0.369e-2$	$0.9003e-1$	0.410	3.643
1.886	$0.425e-2$	0.1002000	0.400	3.774
2.078	$0.465e-2$	0.1069000	0.393	3.867
2.292	$0.487e-2$	0.1102000	0.390	3.934

Table 2.5: Results of our numerical computations for an almost uniformly rotating sequence of equilibrium configurations obtained fixing $b = 0.0$ and $a = 1.12$.
This sequence will end in mass-shedding although it will also tend to have a ring-like structure.

$\frac{\xi_{\rm eq}}{\xi_{\rm pol}}$	j^2	$\frac{T}{\lvert W\rvert}$	$\frac{U}{\lvert W\rvert}$	$\widetilde{M}$
1.000	0.000000	0.000000	0.500	2.710
1.477	$0.307e-2$	$0.8168e-1$	0.419	3.565
2.407	$0.853e-2$	0.1873000	0.313	6.146
3.920	$0.133e-1$	0.2583000	0.242	17.172
6.385	$0.166e-1$	0.2763000	0.224	47.965
10.401	$0.187e-1$	0.2867000	0.214	114.025
16.943	$0.201e-1$	0.2949000	0.206	246.237
27.598	$0.212e-1$	0.3016000	0.199	467.418

Table 2.6: Results of our numerical computations for an almost uniformly rotating sequence of equilibrium configurations obtained fixing $b = 0.383$ and $a = 1.34$.
This sequence will have a ring-like structure and it will not reach an instability although 100 configurations are computed.

$\frac{\xi_{\rm eq}}{\xi_{\rm pol}}$	j^2	$\frac{T}{\|W\|}$	$\frac{U}{\|W\|}$	$\widetilde{M}$
1.000	0.000000	0.000000	0.500	2.710
2.079	$0.698e-2$	0.1620000	0.338	5.299
3.733	$0.128e-1$	0.2527000	0.248	16.536
6.705	$0.162e-1$	0.2714000	0.229	54.776
12.041	$0.182e-1$	0.2813000	0.219	150.250
21.623	$0.193e-1$	0.2884000	0.212	378.059
38.833	$0.201e-1$	0.2936000	0.207	915.252
69.739	$0.207e-1$	0.2977000	0.207	2165.079

Chapter 3

Relativistic Figures of Equilibrium: Neutron Stars

3.1 Introduction

As already stated in Chap. 1, the theory of figures of equilibrium of rotating bodies in Newtonian theory of gravitation was the first step in understanding physics and shape of rotating stars and planets. The advent of General Relativity (GR) evolved the methods to approach the problem of equilibrium of rotating bodies: to solve this problem one should completely determine the geometry of the spacetime interior and surrounding the studied object. This kind of approach is motivated also by astrophysics: when treating Neutron Stars (hereafter NSs) or other compact objects, Einstein's theory must be used to understand their shape and physical properties. In particular understanding the physics of NSs is one of the most important objectives for modern astrophysics, as from one side it could be an excellent test for the general relativity and on the other because the strong forces acting and the high level of densities reached cannot be tested in any place in the earth.

Although our actual incapability to describe these astrophysical objects exactly, some observational limits have already been determined, via simplifying assumptions (such as a spherical configuration in the X-ray binaries observation). Other general considerations on the nature of NSs and pulsars are often extracted in the literature from the use of fiducial structure parameters: a canonical NS is characterized by a mass $M = 1.4\ M_{\odot}$, a radius $R = 10$ km, and a moment of inertia $I = 10^{45}$ g cm^2 (see, e.g., [13], and references therein). Based on these parameters together with the so inferred surface magnetic field from the classic point-like magneto-dipole rotating model [52, 32], NSs have been traditionally classified according to the thought nature of the energy source powering their observed emission. NSs are also thought to possibly participate in the most powerful explosions in the Universe, the gamma-ray bursts (GRBs), e.g. via NS mergers in the case of short GRBs (see, e.g, [47], and references therein) and hypercritical accretion processes leading to gravitational collapse to a black hole (BH) in the case of long GRBs associated to supernovae (see, e.g., [56, 28], and references therein).

Although these assumptions may remain valid for some general approaches to problems taking into account NSs, there are still, many open issues regarding the

above global picture of NSs both from the physics and astrophysics points of view (see, e.g., [34, 64]).

In the present chapter, after a brief description of different methods used to approach the problem of equilibrium of rotating bodies in GR, we will focus on one particular method, implemented by Stergioulas and Friedman (refer to [60]) in the public code RNS[1], including the matter of Equation Of States (EOSs) and describing the ones used in our calculations obtained through a Relativistic Mean-Field (RMF) theory approach. Then we will summarize the physically relevant aspects of rotating NSs with a realistic EOS, also comparing results with observational constraints.

3.1.1 Numerical Methods

There are many different numerical schemes to treat the problem of relativistic figures of equilibrium of rotating stars. Here we will only list and generically describe some of them, but the interested reader should refer to [59], to see also some important references. Note that all methods are numerical, as it doesn't exist any analytical self-consistent solution to this problem for both the interior and the exterior spacetime.

At first, we may report the Hartle's formalism for slow rotations (refer to series of paper beginning with [35] and references therein), in which to the order $O(\Omega^2)$, the structure of the star may change only by quadrupole terms and the equilibrium equations may be reduced to a set of ordinary differential equations. As it models slow rotating stars, it may be used for typical pulsar rotation periods, but it isn't appropriate to work with rapidly rotating relativistic stars, near the mass-shedding limit.

Another method is the one by Butterworth and Ipser (BI) (see [12]), which adopts a Newton-Raphson iterative procedure to solve the system of equations given by the four field equations and the hydrostationary equilibrium equation, beginning with a non-rotating spherical configuration, incrementing the angular velocity of the model by a small step and considering the new model as a linear perturbation of the preceding model. Space is truncated at a finite distance from the surface of the star and the boundary conditions are imposed via an expansion of the metric potentials in powers of $\frac{1}{r}$.

The method by Komatsu, Eriguchi and Hachisu (KEH) described in [42], solves the same set of equations as in the BI method but in here three elliptic-type field equations are converted into integral-type equations while the fourth is an ordinary first-order differential equation. This method is iterative and the parameters to fix are the maximum energy density (central) and the axis ratio (polar over equatorial) of each configuration, but the spatial region is again truncated at a finite distance from the star, as in BI. Cook, Shapiro and Teukolsky (CST) in [20] presented a modified KEH method in which they introduced a new radial variable, which maps an infinite spatial region. Stergiuolas and Friedman (SF) in [60] presented their own implementation of the CST scheme. Their code, named RNS, is available for the public[2].

[1] http://www.gravity.phys.uwm.edu/rns/

[2] http://www.gravity.phys.uwm.edu/rns/

Finally, another important method is the one by Ansorg *et al.* (AKM) presented in [46] which uses several domain inside the star and thus can treat a model which consider different phase transition regions.

3.2 The problem of Equilibrium in GR

In order to construct equilibrium models of rotating stars in GR, one should solve the Einstein's equations, which in general form may be written as

$$G_{\alpha}^{\beta} = \frac{8\pi G}{c^4} T_{\alpha}^{\beta}, \tag{3.1}$$

being G the universal constant of gravitation, c the speed of light, T_{α}^{β} the stress-energy tensor and G_{α}^{β} the Einstein's tensor, coupled to the equation of hydrostationary equilibrium. Eq. (3.1) connect a pure "physical" object on the right-hand side, expressed via the stress-energy tensor which contains information on the physical nature of matter composing the spacetime, with a pure"geometrical" one on the left-hand side, represented by the Einstein's tensor which indeed describe the geometrical properties of the spacetime. Solving Eq. (3.1), means finding an expression for the metric tensor, with which one could entirely define the spacetime inside and outside the equilibrium configuration.

In the following sections, we will see how explicit expressions may be written using an ad-hoc coordinate system and what kinds of methods exist to solve this problem.

3.2.1 Structure Equations for a rotating star

In the method adopted by the RNS public code, one considers the equilibrium equations for a self-gravitating, rapidly rotating NS, within a fully general relativistic framework. Let's start with the stationary axisymmetric spacetime metric (see, e.g., [59]):

$$ds^2 = -e^{2\nu}dt^2 + e^{2\psi}(d\phi - \omega dt)^2 + e^{2\lambda}(dr^2 + r^2 d\theta^2), \tag{3.2}$$

where ν, ψ, ω and λ depend only on variables r and θ.[3] It is useful to rename the quantity $e^{\psi} = r\sin(\theta)Be^{-\nu}$, being again $B = B(r,\theta)$. The above form of the metric is obtained under two assumptions: 1) the Killing vector fields are, one timelike (t^a), and one relative to the axial symmetry (ϕ^a); 2) the spacetime is asymptotically flat. Then, one can introduce *quasi-isotropic coordinates*, which in the non-rotating limit they tend to isotropic ones.

Turning to the physical matter content in the NS interior, if one neglects sources of non-isotropic stresses, viscosity, and heat transport, then the energy-momentum tensor becomes the one of a perfect fluid

$$T^{\alpha\beta} = (\varepsilon + P)u^{\alpha}u^{\beta} + Pg^{\alpha\beta}, \tag{3.3}$$

where ε and P denote the energy density and pressure of the fluid, and u^{α} is the fluid 4-velocity. In terms of the two Killing vectors,

$$u^{\alpha} = \frac{e^{-\nu}(t^{\alpha} + \Omega\phi^{\alpha})}{\sqrt{1-v^2}}, \tag{3.4}$$

where v is the fluid 3-velocity with respect to the local zero angular momentum observer (ZAMO),

$$v = (\Omega - \omega)e^{\psi-\nu} \tag{3.5}$$

being $\Omega \equiv u^{\phi}/u^{t} = d\phi/dt$ the angular velocity in the coordinate frame, equivalent to the one measured by an observer at rest at infinity.

Thus, with the metric given by Eq. (3.2) and the energy-momentum tensor given by Eq. (3.3), one can write the field equations as (analogously to Ref. [12] setting

[3]Computing eigenvalues of the matrix associated to this metric tensor it is easy to check that the signature here adopted is $(-,+,+,+)$(the non-rotating case is straightforward, imposing $\omega = 0$). It is worth remembering that fixed the metric $g_{\alpha\beta}$, there exist only one Levi-Civita connection ∇_{α}, for which it holds $\nabla_{\alpha}g_{\beta\gamma} = 0$ and which allows to computes Christoffel symbols as

$$\Gamma^{\alpha}_{\beta\gamma} = \frac{1}{2}g^{\alpha\delta}\left(g_{\gamma\delta,\beta} + g_{\beta\delta,\gamma} - g_{\beta\gamma,\delta}\right),$$

where the common Einstein's notation is used and a comma followed by a letter stands for differentiation with respect to coordinate indicated with this letter. With Christoffel symbols one is able to compute the Riemann tensor with the following formula

$$R^{\delta}_{\alpha\beta\gamma} = \Gamma^{\delta}_{\alpha\gamma,\beta} - \Gamma^{\delta}_{\beta\gamma,\alpha} + \Gamma^{\sigma}_{\alpha\gamma}\Gamma^{\delta}_{\sigma\beta} - \Gamma^{\sigma}_{\beta\gamma}\Gamma^{\delta}_{\sigma\alpha},$$

and subsequently the Ricci tensor, given by the contraction of one index of the Riemann tensor,

$$R_{\alpha\gamma} = R^{\beta}_{\alpha\beta\gamma}$$

and the Ricci scalar R via the trace of Ricci tensor. Finally the Einstein's tensor can be written as

$$G_{\alpha\beta} = R_{\alpha\beta} - \frac{1}{2}g_{\alpha\beta}R,$$

(see [66] for a complete treatment).

$\zeta = \lambda + \nu$):

$$\nabla\cdot(B\nabla\nu) = \frac{1}{2}r^2\sin^2\theta B^3 e^{-4\nu}\nabla\omega\cdot\nabla\omega + 4\pi B e^{2\zeta-2\nu}\left[\frac{(\varepsilon+P)(1+v^2)}{1-v^2}+2P\right], \quad (3.6a)$$

$$\nabla\cdot\left(r^2\sin^2\theta B^3 e^{-4\nu}\nabla\omega\right) = -16\pi r\sin\theta B^2\times e^{2\zeta-4\nu}\frac{(\varepsilon+P)v}{1-v^2}, \quad (3.6b)$$

$$\nabla\cdot(r\sin(\theta)\nabla B) = 16\pi r\sin\theta B e^{2\zeta-2\nu}P, \quad (3.6c)$$

$$\begin{aligned}
\zeta_{,\mu} = &-\left\{\left(1-\mu^2\right)\left(1+r\frac{B_{,r}}{B}\right)^2+\left[\mu-\left(1-\mu^2\right)\frac{B_{,r}}{B}\right]^2\right\}^{-1}\\
&\times\left[\frac{1}{2}B^{-1}\left\{r^2B_{,rr}-\left[\left(1-\mu^2\right)B_{,\mu}\right]_{,\mu}-2\mu B_{,\mu}\right\}\right.\\
&\times\left\{-\mu+\left(1-\mu^2\right)\frac{B_{,\mu}}{B}\right\}+r\frac{B_{,r}}{B}\left[\frac{1}{2}\mu+\mu r\frac{B_{,r}}{B}+\frac{1}{2}\left(1-\mu^2\right)\frac{B_{,\mu}}{B}\right]\\
&+\frac{3}{2}\frac{B_{,\mu}}{B}\left[-\mu^2+\mu\left(1-\mu^2\right)\frac{B_{,\mu}}{B}\right]-\left(1-\mu^2\right)r\frac{B_{,\mu r}}{B}\left(1+r\frac{B_{,r}}{B}\right)-\mu r^2(\nu_{,r})^2\\
&-2\left(1-\mu^2\right)r\nu_{,\mu}\nu_{,r}+\mu\left(1-\mu^2\right)(\nu_{,\mu})^2-2\left(1-\mu^2\right)r^2B^{-1}B_{,r}\nu_{,\mu}\nu_{,r}\\
&+\left(1-\mu^2\right)B^{-1}B_{,\mu}\left[r^2(\nu_{,r})^2-\left(1-\mu^2\right)(\nu_{,\mu})^2\right]+\left(1-\mu^2\right)B^2e^{-4\nu}\\
&\times\left\{\frac{1}{4}\mu r^4(\omega_{,r})^2+\frac{1}{2}\left(1-\mu^2\right)r^3\omega_{,\mu}\omega_{,r}-\frac{1}{4}\mu\left(1-\mu^2\right)r^2(\omega_{,\mu})^2\right.\\
&+\frac{1}{2}\left(1-\mu^2\right)r^4B^{-1}B_{,r}\omega_{,\mu}\omega_{,r}-\frac{1}{4}\left(1-\mu^2\right)r^2B^{-1}B_{,\mu}\\
&\left.\left.\times\left[r^2(\omega_{,r})^2-\left(\mu^2\right)(\omega_{,\mu})^2\right]\right\}\right],
\end{aligned} \quad (3.6d)$$

where, in the Eq. (3.6d) we introduced $\mu \equiv \cos(\theta)$.

The projection of the conservation of the energy-momentum tensor, normal to the 4-velocity, $\left(\delta^c_b + u^c u_b\right)\nabla_a T^{ab} = 0$, leads to the hydrostationary equilibrium equation:

$$P_{,i} + (\varepsilon+P)\left[\nu_{,i}+\frac{1}{1-v^2}\left(-vv_{,i}+v^2\frac{\Omega_{,i}}{\Omega-\omega}\right)\right] = 0, \quad (3.7)$$

where $i = 1, 2, 3$ and, as usual a comma followed by the index i means partial derivation with respect to the $i-$th coordinate.

For a barotropic equation of state (EOS), $P = P(\varepsilon)$, and in the case of uniform rotation the above hydrostationary equilibrium equation has a first integral that can be written as

$$\int_0^P \frac{dP}{\varepsilon+P} - \ln(u^a\nabla_a t) = \nu|_{\rm pole}, \quad (3.8)$$

where the constant of motion has been obtained, for instance, at the pole of the star (see, e.g.,Ref. [59]).

RNS code uses a user defined EOS to solve the system given by Equations (3.6a), (3.6b), (3.6c), (3.6d) and (3.8). The EOS (barotropic, i.e. of type $P = P(\varepsilon)$) supplied allows to relate the pressure P to the energy density ε, making the problem solvable. An interesting feature of the code is that there is a compactification of variables, via the definition of a new radial variable, namely

$$s(r) = \frac{r}{(r + r_e)}, \tag{3.9}$$

being r the spherical radial coordinate and r_e the spherical radius at the equator. With this definition the values of unknowns in entire spacetime, from the centre of configuration (corresponding to $s = r = 0$) up to radial infinity ($s = 1$), can be computed, thus, one can obtain information also on the gravitational field exterior to the surface (represented by $s = 0.5$). A numerical grid is then defined, similarly to the one in the case of Newtonian configurations described in Chap. 1 and the equations are numerically solved assuming uniform rotation.

3.2.2 Note on Rotation Law

It is worth noting that the only case of uniform rotation may be used to obtain equilibrium configurations of rotating NS. What about differential rotation? Regarding to [59] and references therein, one should consider that even if a newly born NS is expected to rotate in a differential way, it has already been studied that after a very brief period (namely of the order of days, roughly of one year) many physical mechanisms should act braking the differential rotation. After the first year of formation it has been shown that adopting uniform rotation instead of differential one comport roughly an error in the spacetime metric of the order of 10^{-12}, thus the bulk properties of a rotating NS may be well approached by a uniformly rotating model. Differential rotation and finite temperature affect relativistic NS's physical properties only during the first year since it has been formed. For this reason, one could also neglect temperature effects.

3.2.3 EOS

To obtain a solution to the field equations, the matter EOS must be supplied. In general, a NS is composed of two regions, namely the core and the crust. The core, with densities overcoming the nuclear saturation value, $\rho_{\rm nuc} \approx 3 \times 10^{14}$ g cm^{-3}, is composed of a degenerate gas of baryons (e.g. neutrons, protons, hyperons) and leptons (e.g. electrons and muons). The crust, in its outer region ($\rho \leq \rho_{\rm drip} \approx 4.3 \times 10^{11}$ g cm^{-3}), is composed of ions and electrons, while in its inner region ($\rho_{\rm drip} < \rho < \rho_{\rm nuc}$), there is an additional component of free neutrons dripped out from nuclei. For the crust, we adopt the Baym-Pethick-Sutherland (BPS) EOS [5]. For the core, we here adopt modern models based on relativistic mean-field (RMF) theory. Indeed, RMF models have become the most used ones in NS literature, being its success mainly owing to important properties such as Lorentz covariance, intrinsic inclusion of spin, a simple mechanism of saturation for nuclear matter, and

being consistently relativistic, they do not violate causality (see, e.g., Ref. [57]). We adopt, as now becoming traditional, an extension of the original formulation of Boguta and Bodmer [10] in which nucleons interact via massive meson mediators of different nature providing the attractive long range (scalar σ) and repulsive short range (vector ω) of the nuclear force, isospin and surface effects (vector ρ). Meson-meson interactions can be also present; for instance, in the version of Boguta and Bodmer [10] there is the presence of a self-interacting scalar field potential in the form of a quartic polynomial with adjustable coefficients. We consider here the possibility of including, in addition to such potential, vector-vector interactions of the ω meson. For a very recent and comprehensive analysis of the performance of several RMF models in the description of observed properties of ordinary nuclei, we refer the reader to Ref. [23], and for a brief historical and chronological reconstruction of the developments of the RMF models, Ref. [8].

Thus, we shall constrain ourselves to models in which the energy density and pressure are given by (in units with $\hbar = c = 1$) [23]:

$$\varepsilon = \frac{1}{2}m_\sigma^2\sigma^2 + \frac{g_{\sigma 2}}{3}\sigma^3 + \frac{g_{\sigma 3}}{4}\sigma^4 - \frac{1}{2}m_\omega^2\omega_0^2 - \frac{g_{\omega 3}}{4}(g_\omega^2\omega_0^2)^2 - \frac{1}{2}m_\rho^2\rho_0^2 + g_\omega\omega_0 n_B + \\ + \frac{g_\rho}{2}\rho_0 n_3 + \sum_{i=n,p,e,\mu}\varepsilon_i, \tag{3.10}$$

$$P = -\frac{1}{2}m_\sigma^2\sigma^2 - \frac{g_{\sigma 2}}{3}\sigma^3 - \frac{g_{\sigma 3}}{4}\sigma^4 + \frac{1}{2}m_\omega^2\omega_0^2 + \frac{g_{\omega 3}}{4}(g_\omega^2\omega_0^2)^2 + \frac{1}{2}m_\rho^2\rho_0^2 + \sum_{i=n,p,e,\mu}P_i, \tag{3.11}$$

where $m_{\sigma,\omega,\rho}$ are the masses of the scalar and vector mesons, $g_{\sigma 2,3}$, g_ω, $g_{\omega 3}$ are coupling constants, σ, ω_0, ρ_0 denote the scalar meson and the time-component of the ω and ρ vector mesons, respectively. The components ε_i and P_i for each kind of particle considered are

$$\varepsilon_{n,p} = \frac{2}{(2\pi)^3}\int_0^{k_{n,p}^F}\sqrt{k^2 + (m_{n,p}^*)^2}\,d^3k, \tag{3.12a}$$

$$\varepsilon_{e,\mu} = \frac{2}{(2\pi)^3}\int_0^{k_{e,\mu}^F}\sqrt{k^2 + (m_{e,\mu})^2}\,d^3k, \tag{3.12b}$$

$$P_{n,p} = \frac{1}{3}\frac{2}{(2\pi)^3}\int_0^{k_{n,p}^F}\frac{k^2}{\sqrt{k^2 + (m_{n,p}^*)^2}}d^3k, \tag{3.12c}$$

$$P_{e,\mu} = \frac{1}{3}\frac{2}{(2\pi)^3}\int_0^{k_{e,\mu}^F}\frac{k^2}{\sqrt{k^2 + (m_{e,\mu})^2}}d^3k, \tag{3.12d}$$

where with m_i^* is intended the effective mass of baryons.

The scalar, isospin, and baryon densities are given by, respectively,

$$n_s = \frac{2}{(2\pi)^3}\sum_{i=n,p}\int_0^{k_i^F}\frac{m_i^*}{\sqrt{k^2+(m_i^*)^2}}d^3k, \tag{3.13a}$$

$$n_3 = n_p - n_n, \tag{3.13b}$$

$$n_B = n_p + n_n. \tag{3.13c}$$

where, $n_i = (k_i^F)^3/(3\pi^2)$, are the particle number densities with k_i^F the particle Fermi momenta.

The equations of motion of the meson fields within the RMF approximation are:

$$m_\sigma^2\sigma = g_\sigma n_s - g_{\sigma 2}\sigma^2 - g_{\sigma 3}\sigma^3, \tag{3.14a}$$

$$m_\omega^2\omega_0 = g_\omega n_B - g_{\omega 3}g_\omega(g_\omega\omega_0)^3, \tag{3.14b}$$

$$m_\rho^2\rho_0 = \frac{g_\rho}{2}n_3. \tag{3.14c}$$

A barotropic EOS can be obtained if and only if additional closure relations are supplied. A first condition to be imposed is the request of the stability of matter against beta decay. The second closure equation that has been traditionally adopted is the condition of local charge neutrality of the system. The latter condition has been recently shown to be not fully consistent with the equilibrium equations in presence of multicomponent charged constituents such as protons and electrons (see [8], and references therein). Instead, only global charge neutrality must be requested. The new system of equations, referred to as Einstein-Maxwell-Thomas-Fermi (EMTF) equations, introduces self-consistently the Coulomb interactions in addition to the strong, weak, and gravitational interactions within a full GR framework. It is worth to notice that in this case no perfect-like form of the total energy-momentum tensor is obtained since the presence of electromagnetic fields breaks the pressure isotropy. Static NSs fulfilling the EMTF equations were constructed in Ref. [8], and uniformly rotating configurations in the second-order Hartle approximation can be found in Ref. [7]. To construct rotating NSs beyond the slow rotation regime, we take advantage of existing public numerical codes (e.g. the RNS code, see section 3.1.1) that solve the field equations without any limitation of the rotation rate of the star. However, an implementation of the equations and boundary conditions of the EMTF system within these codes is not yet available. Thus, as a first step, we adopt in here the condition of local charge neutrality, bearing in mind the necessity of a future implementation of the EMTF equations of equilibrium in fast rotation regime.

With both the beta equilibrium and the local charge neutrality conditions, a numerical relation between the energy density and the pressure can be obtained. We here adopt the nuclear parametrizations (for the specific values of the coupling constants, particle, and meson field masses) NL3 [43], TM1 [62], and GM1 [31, 53] (after having adapted these in units used by the code). In Fig. 3.1, we compare and contrast the three selected EOS used in our work in the nuclear and supranuclear regime, relevant for NS cores.

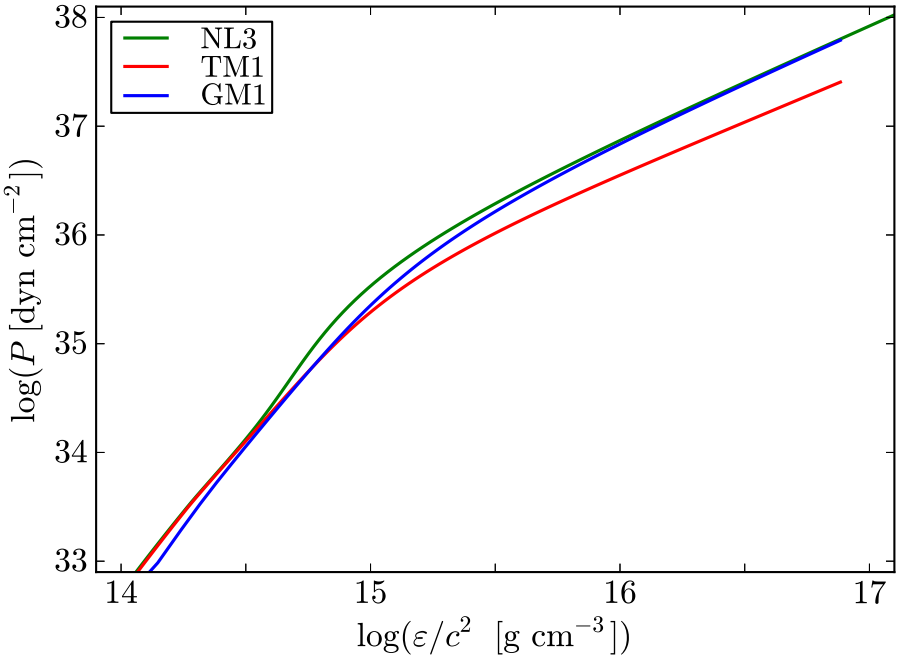

Figure 3.1: Pressure-energy density relation for the three EOS used.

3.3 Equilibrium Configurations of NS in full GR

Supplying tabulated EOSs described in Sec. 3.2.3, allows RNS code to produce models of rapidly rotating self-gravitating realistic NS, without making an assumption on the angular velocity, as instead is required by expansion methods, like the one in [35]. In the subsequent sections results of our calculations, already published in [18], will be reported.

3.3.1 Stability

To solve the system of field equations (3.6a–3.6d) and the hydrostationary equilibrium equation (3.8), one has to fix one (in the spherical static case) or two parameters (in the rotating case). The first quantity to fix is the central value of energy density, ε_c. For a rotating model, one can choose the second parameter amongst different possibilities: the axes ratio ($r_{\rm pol}/r_{\rm eq}$) of coordinate radii, angular velocity (Ω), dimensionless angular momentum (j), gravitational mass (M) or baryonic mass (M_b). Thus, it is always possible to construct a sequence of rotating models by fixing a value of the second parameter and letting the central energy density to vary in a given range which is constrained to stability limits which we now discuss.

The first limit to introduce is for sure given by the *static* (non-rotating) sequence of models, because if we imagine a M-ε_c plane, being M the mass of a configuration with central density ε_c, the curve described by non-rotating models gives a lower limit to the mass of a NS, given a value for its central energy density.

Another limit for uniformly rotating configurations which we take into account is given by the sequence of maximally rotating stars, also referred to as *Keplerian* or *mass-shedding* sequence. In all the stars belonging to such a sequence, the gravita-

tional force equals the centrifugal force at the star equator, in such a way that faster rotation rates would induce the expulsion of mass from the star. The RNS code calculates this sequence by decreasing the axis ratio (polar over equatorial, which correspond to an increase in angular velocity) until the angular velocity equals the one of a particle orbiting the star at its equator.

Another limit to the physically relevant models is determined by the called secular axisymmetric instability. For static configurations, the maximum stable mass (the critical mass) coincides with the first maximum of a sequence of configurations with increasing central density, namely the first point where $\partial M/\partial\varepsilon_c = 0$. At this point, the frequency of the radial perturbations vanishes. For higher densities, imaginary frequencies are obtained which lead to non-oscillatory perturbations, hence an instability. Thus, for static configurations, a *turning-point* of the M-ε_c relation locates the onset of unstable configurations. This instability proceeds on *secular* timescales, i.e. not dynamical, so that it proceeds on long times that allow the star to accommodate itself to the energy loss that occurs when going from one equilibrium point to another during the gravitational collapse (see, e.g., Ref. [27], and references therein). As shown by Friedman, Ipser and Sorkin in [27], the turning-point method leading to points of secular instability can be also used in uniformly rotating stars as follows. In a constant angular momentum sequence, the turning point of a sequence of configurations with increasing central density separates secularly stable from secularly unstable configurations. Namely, secular axisymmetric instability sets in at

$$\left.\frac{\partial M\left(\varepsilon_c, J\right)}{\partial\varepsilon_c}\right|_{J=\text{constant}} = 0, \tag{3.15}$$

and therefore the curve connecting all the maxima (turning-points) limits the stability region. The intersection of such a limiting curve with the Keplerian sequence gives the fastest possible configuration. It is important to mention that the numerical code adopted (described in the next section) builds sequences of constant dimensionless angular momentum, defined as

$$j \equiv \frac{cJ}{GM_\odot^2}, \tag{3.16}$$

which is the quantity we refer to in the sequel.

The angular momentum J is computed from the definition

$$J = \int_\Sigma T_{ab}\phi^a\hat{n}^b dV, \tag{3.17}$$

being Σ a spacelike 3-surface, $\hat{n}^a = \nabla_a t/|\nabla_b t\nabla^b t|$ the unit normal vector field to the t=constant spacelike hypersurfaces and $dV = \sqrt{|^3g|}d^3x$ the proper 3-volume element (with ${}^3g$ the determinant of the 3-metric). With this, Eq. (3.17) becomes [12]

$$J = \int B^2 e^{2\zeta-4\nu}\frac{(\varepsilon+P)v}{1-v^2}r^3\sin^2(\theta)drd\theta d\phi. \tag{3.18}$$

The major intuitive effect of rotation is to deform the figure of equilibrium with respect to the spherical static counterpart. This can be seen from many points of

view. For instance, we can compute sequences of constant angular velocity Ω. An important aspect should be taken into account from now on: the RNS code builds fast rotating models starting from a spherical (static) guess and decreasing the polar to equatorial radii ratio until the parameter fixed (e.g. the angular velocity) is reached with a prescribed accuracy. Thus, the axes ratio is a parameter used intrinsically by the numerical method, while other parameters (see the beginning of the previous section for a list) can be chosen, but are reached spanning decreasing values of axis ratio.

We show in Fig. 3.2 the total mass-central energy density plane for the selected EOS TM1, GM1 and NL3. We also show the stability limits discussed above and show explicitly some constant angular momentum sequences.

3.3.2 Constant Frequency sequences

As said in the previous section, one possible parameter to fix in the angular velocity Ω, and then, if one lets central energy density vary in a certain range (defined by the stability region) it is possible to obtain sequences of models. But, remember that the "preferred" parameter is always the axis ratio. In particular, as an example, the code does not converge for every value of fixed angular velocity in every range of central energy density, and the range of convergence gets reduced by decreasing the angular velocity. To be more precise, choosing fixed rotation frequencies below 300 Hz, the code fails to converge in the entire range in which equilibrium models should exist (thus between the Keplerian and secular instability limits), even adopting a very dense numerical grid (300 angular times 600 radial points), different accuracy and tolerance values (down to 10^{-16}) or values of a relaxation factor from 1 to 0.8 (see code's manual for information about these parameters). Effectively the code does converge for this kind of rotation frequencies but in very limited ranges of energy density. Thus, how can the slow rotation regime be recovered in all the stability region? As a technical advice, we would like to mention that one could compute first, sequences keeping various values of axes ratio constant (in the vicinity of unity), and then select in this set of models the ones with small values of a particular angular velocity. We construct with this simple method the sequences with low rotation frequency (e.g. models from 50 Hz to 200 Hz). For other values of rotation frequency and for all other parameters constant sequences, we achieved optimal convergence using 300 angular times 600 radial points numerical grid, and accuracy and tolerance of 10^{-8}, while the relaxation parameter was not necessary.

Fig. 3.3 shows the total mass-central energy density plane but in this case, we show explicitly some selected constant rotation frequency sequences ranging from 50 Hz all the way up to the rotation frequency of the fastest observed pulsar, PSR J1748 − 2446ad, with $f = \Omega/(2\pi) \approx 716$ Hz [38]. In figure 3.4 we plot the same $\Omega-$constant sequences to show the relation between M and the equatorial radius, $R_{\rm eq}$.

From Fig.s 3.3 and 3.4 we can clearly see that, as expected, the higher the frequency of rotation is, the higher the value of the mass at which start the departures from the non-rotating mass-radius relation results. We find that for rotation frequencies $\lesssim 200$ Hz (or rotation periods $\gtrsim 5$ ms), the non-rotating star becomes an

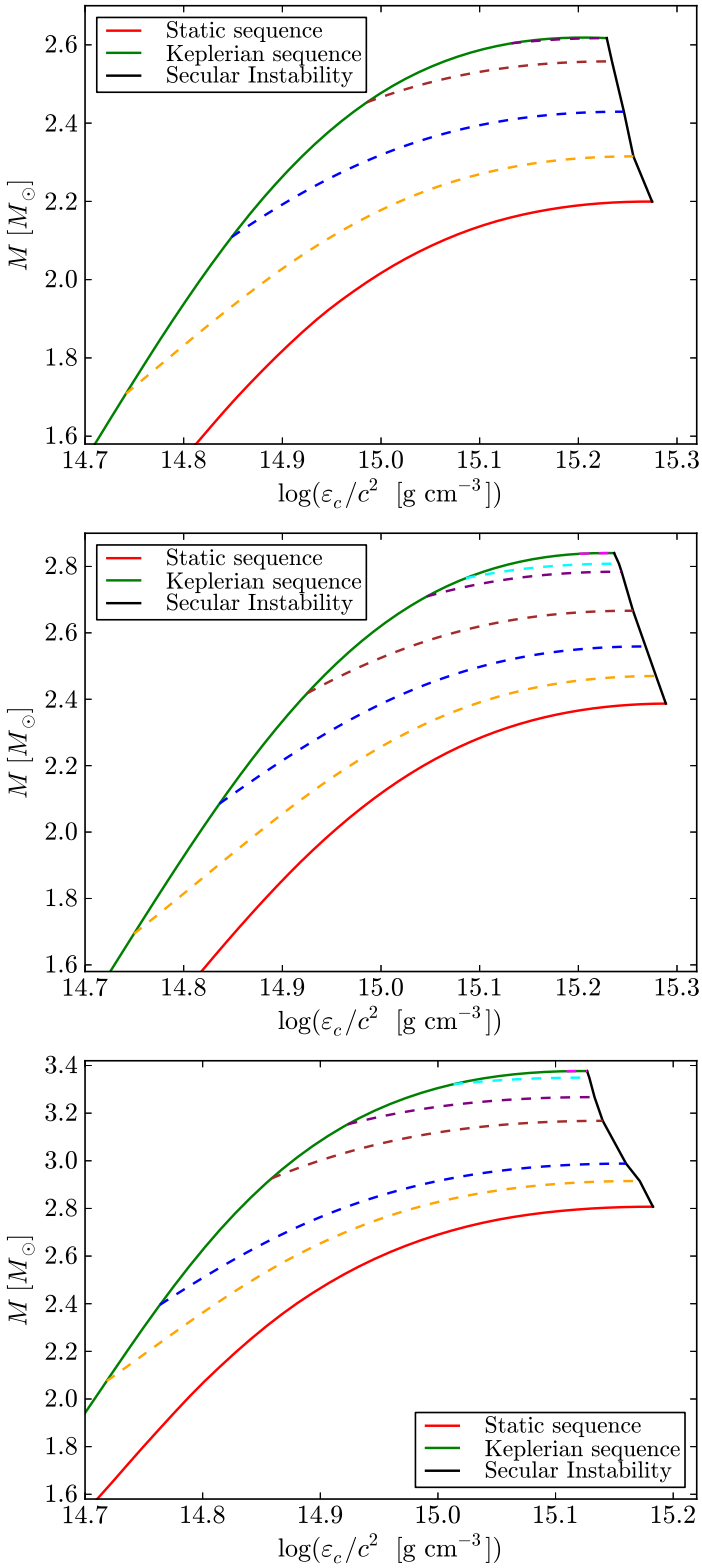

Figure 3.2: Gravitational mass is plotted against central energy density for j-constant sequences obtained with the EOS TM1, GM1 and NL3 (from top to bottom). In this plot and hereafter, the red, green and black curves represent respectively the static sequence, the Keplerian sequence, and the limit for secular stability. Here other colors stand for various j-constant sequences.

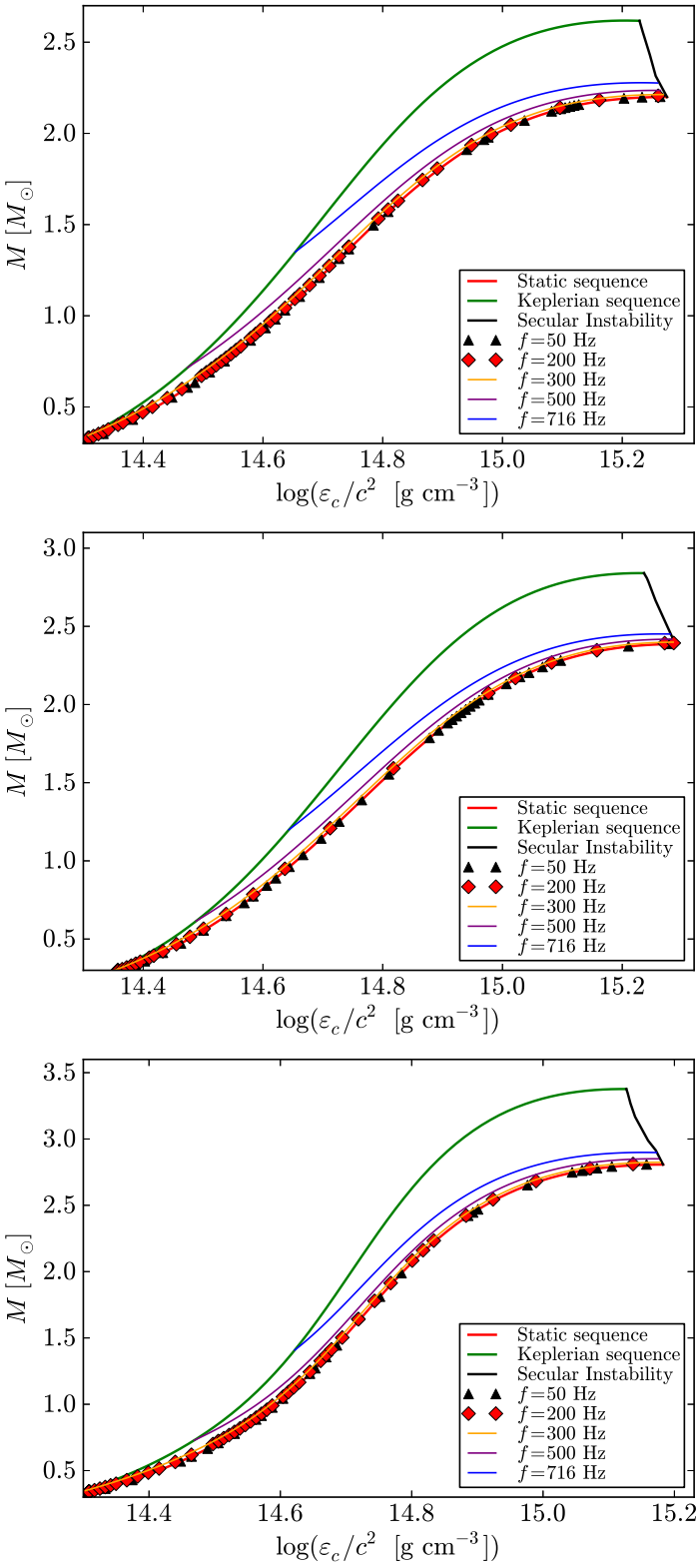

Figure 3.3: Mass is plotted against central energy density for sequences of constant angular velocity using the EOS TM1, GM1 and NL3 (from top to bottom) set of parameters. Red and green curves represent the static and Keplerian sequences. Other colors correspond to constant frequency sequences of value 716 Hz (fastest observed pulsar; blue), 500 Hz (purple), 300 Hz (orange), 200 Hz (red diamonds) and 50 Hz (black triangles)

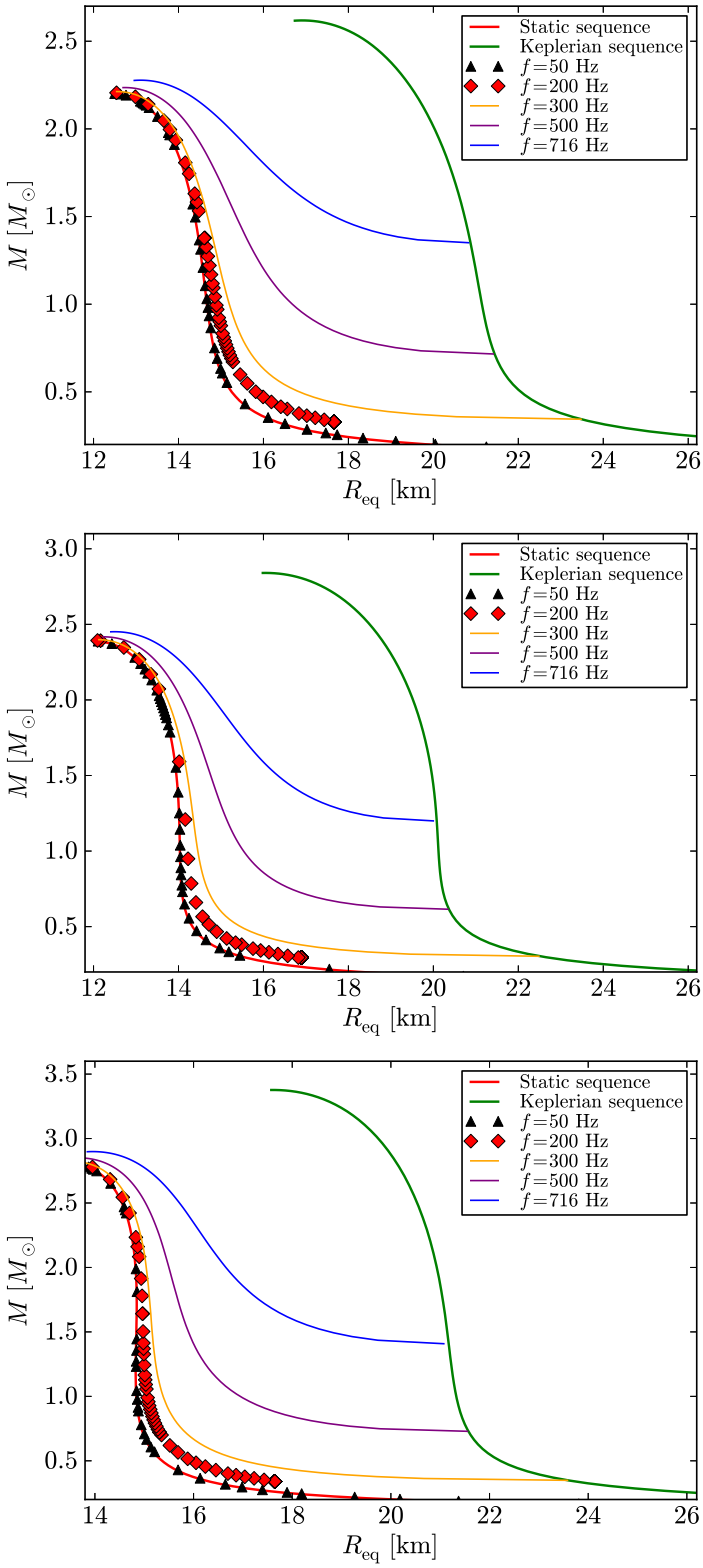

Figure 3.4: Same as Fig. 3.3 but for mass versus equatorial radius.

accurate representation of the object. This is in accordance with previous results, see e.g. figure 5 in Ref. [9], where it was shown that the moment of inertia of sequences computed with different EOS, starts to deviate considerably from the static and the slow-rotation Hartle's approximations for frequencies above ~ 0.2 kHz. As we show below, this is also the case for the moment of inertia in the same range of frequencies (thus, the moment of inertia of non-rotating configurations, can be safely approximated with the one of spinning configurations, with frequencies below the aforementioned limit, and vice-versa). For higher frequencies, full rotation effects are needed for an accurate description. This is especially important for objects with masses lower that the maximum value, where departures from a non-rotating or slow rotation approximation become more and more evident.

Following this reasoning, it is important to see how a constant frequency sequence imposes structure constraints to a pulsar. Particularly interesting becomes the case of the $f = 716$ Hz sequence (blue curve), which corresponds to the fastest observed pulsar, PSR J1748−2446ad. The constant frequency sequence intersects the stability region in two points: at the maximally rotating Keplerian sequence, defining a minimum mass for the pulsar, and at the secular axisymmetric instability limit, in the upper part, defining the maximum possible mass for the given frequency. Clearly these minimum and maximum mass values depend upon the EOS. For the EOS employed here, we can see that the mass of PSR J1748 − 2446ad has to be $\gtrsim [1.41, 1.35, 1.20]\ M_\odot$ for NL3, TM1, and GM1, respectively.

3.3.3 Observational Constraints

Why are these results called realistic?

In effect, there are in literature some observational constraints imposed on the NS structure, in several physical aspect. To make a list of the most relevant and updated constraints, one should consider:

- *Most massive NS observed.* The mass value of the most massive NS observed is the one of PSR J1614–2230 with $2.01 \pm 0.04 M_\odot$ [22]. The rotation frequency of this pulsar is 46 Hz; thus, the deviations from spherical symmetric are negligible. This implies that every mass-radius relation for non-rotating NSs must have a maximum stable mass larger than this value.

- *Fastest observed NS.* The highest rotation frequency observed in a pulsar is the one of PSR J1748 − 2446ad with $f = 716$ Hz [38]. The constant frequency sequence of this value for any mass-radius relation must have at least one stable configuration that supports such a rotation frequency; namely, the constant frequency sequence for this pulsar must lie within the region of stability. This is actually a very weak constraint since most of NS models allow much higher rotation frequencies. Interestingly, as we shall show below, the construction of the constant frequency sequence for PSR J1748 − 2446ad allows us to infer a lower mass for this pulsar.

- *Causality.* The speed of sound in the NS interior cannot exceed the speed of light. This implies an approximate lower limit to the NS radius, $R \gtrsim 3GM/c^2$.

This is also a weak constraint and in the case of the present EOS, it is satisfied by construction since the models are relativistic.

- *Constraints to the NS radius.* Since the surface temperatures of not-so-young NSs ($t > 10^3$–10^4 y) are of the order of million degrees (see, e.g., [21]), their thermal spectrum is expected to peak in the soft X-rays. Thus, the modelling of the NS X-ray emission appears to be, at the present, one of the most promising methods to obtain information on the NS radius. Systems which are currently used to this aim are isolated NSs, quiescent low-mass X-ray binaries (qLMXBs), NS bursters, and rotation-powered millisecond pulsars (see, Ref. [26], and references therein). From the modelling of the observed spectrum, the radius of the NS as measured by an observer at infinity, $R_\infty = R/\sqrt{1-2GM/(c^2R)}$, can be extracted[4]. The observation of a preferable radius at infinity clearly represents a constraint on the NS mass-radius relation since the above definition for R_∞ can be rewritten as $2GM/c^2 = R - R^3/(R_\infty^3)$. In Ref. [44] (see, also, Ref. [36]), the X-ray emission from the NSs in the qLMXBs M87, NGC 6397, M13, ω Cen, and NGC 6304 was revisited, and in Ref. [37] the one of the NS X7 in the Globular Cluster 47 Tucanae. From the extracted values of R_∞ consistent with these observational data at 90% confidence level, we can conclude that the current X-ray data constraints very weakly the mass-radius relation, allowing radii in the interval $R_\infty = [7.64, 18.86]$, where the lower limit is obtained for NGC 6304 and the upper one for X7. It is important to mention that X-ray measurements suffer from a variety of uncertainties which are the main reason for the very large spread in possible NS radii. The spectra modelling depends on the atmosphere composition, magnetic fields, accurate knowledge of the distance to the source, hence the extinction, and to some extent on the NS exterior geometry which could be affected by the rotation of the NS in the case of some LMXBs which could harbour NSs rotating with frequencies of a few hundreds of Hz (see, e.g., Ref. [4], for details). In these latter cases, a more reliable comparison between theory and the above data constraints, which assume spherical symmetry, could be obtained by plotting mass-radius relation using, instead of the equatorial radius, a mean or average spherical radius such as the authalic radius, $\langle R\rangle = (2R_{\rm eq} + R_{\rm pol})/3$. However, for the purposes of this work, it is sufficient to make a comparison with the mass-radius relation produced by the non-rotating configurations.

Taking into account the aforementioned observational constraint, we can now see the mass-radius relation in Fig. 3.5, noticing that our chosen set of EOSs are well in accordance with these.

[4]Actually, accurate spectra modelling leads to preferable values for both mass and radius; however, for a simpler comparison between different results from different methods and for a simple test of the mass-radius relation it is sufficient to plot the constraints obtained from the values of R_∞ consistent with the data [37].

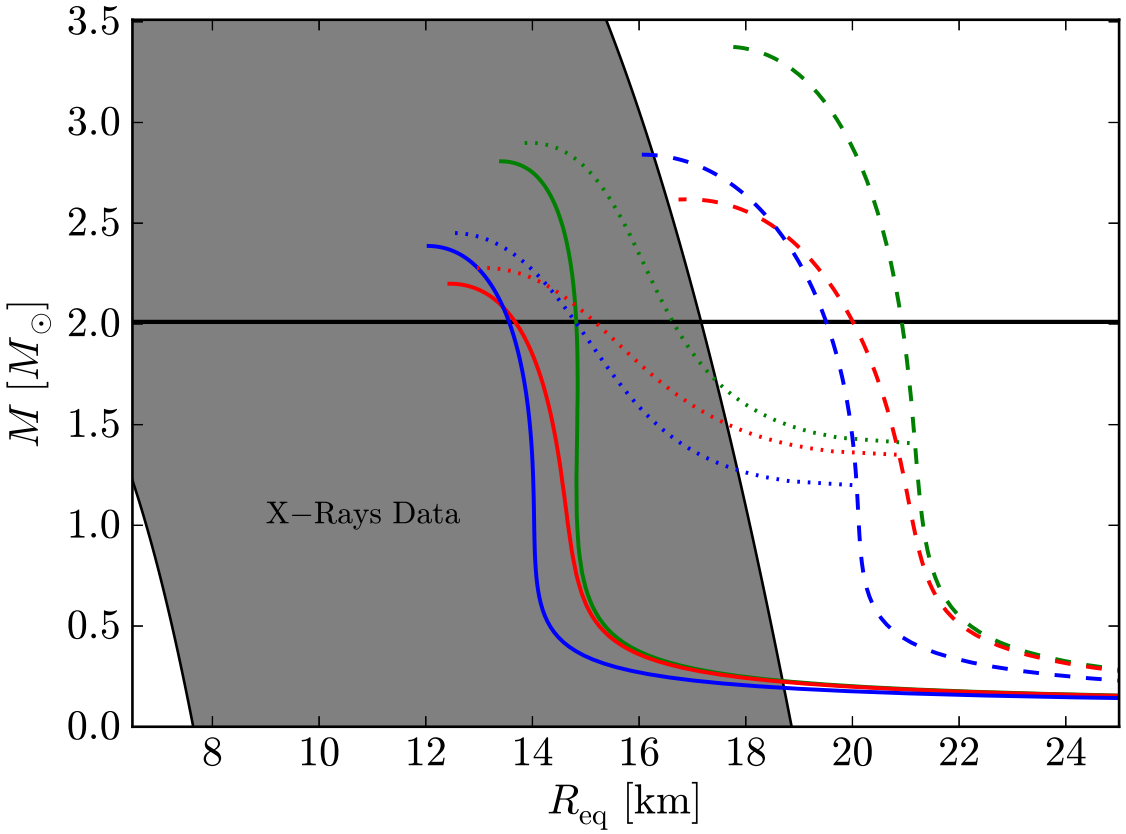

Figure 3.5: Mass versus equatorial radius relation using EOS TM1, GM1 and NL3 (respectively with colors red, blue and green) set of parameters, plotted together with latest observational constraints founded. Dashed color curves represent static NS configurations, dashed-dotted color curves represent the sequence of models rotating with spin frequency of the fastest observed pulsar (PSR J1748–2446ad), and solid color curves represent sequences of models rotating at the fastest (Keplerian) frequency. Black curves shows observational limits, as described in the text.

3.3.4 Maximal Frequency and "Kerr Parameter"

We now determine the maximum rotation frequency of NSs. The fastest configuration for a given EOS is the one that terminates at the Keplerian sequence, namely the configuration at the intersection between the Keplerian and the secular axisymmetric instability sequences. We show in Fig. 3.6 the rotation frequency of the maximally rotating configurations, i.e. the frequencies of NSs along the Keplerian sequence.

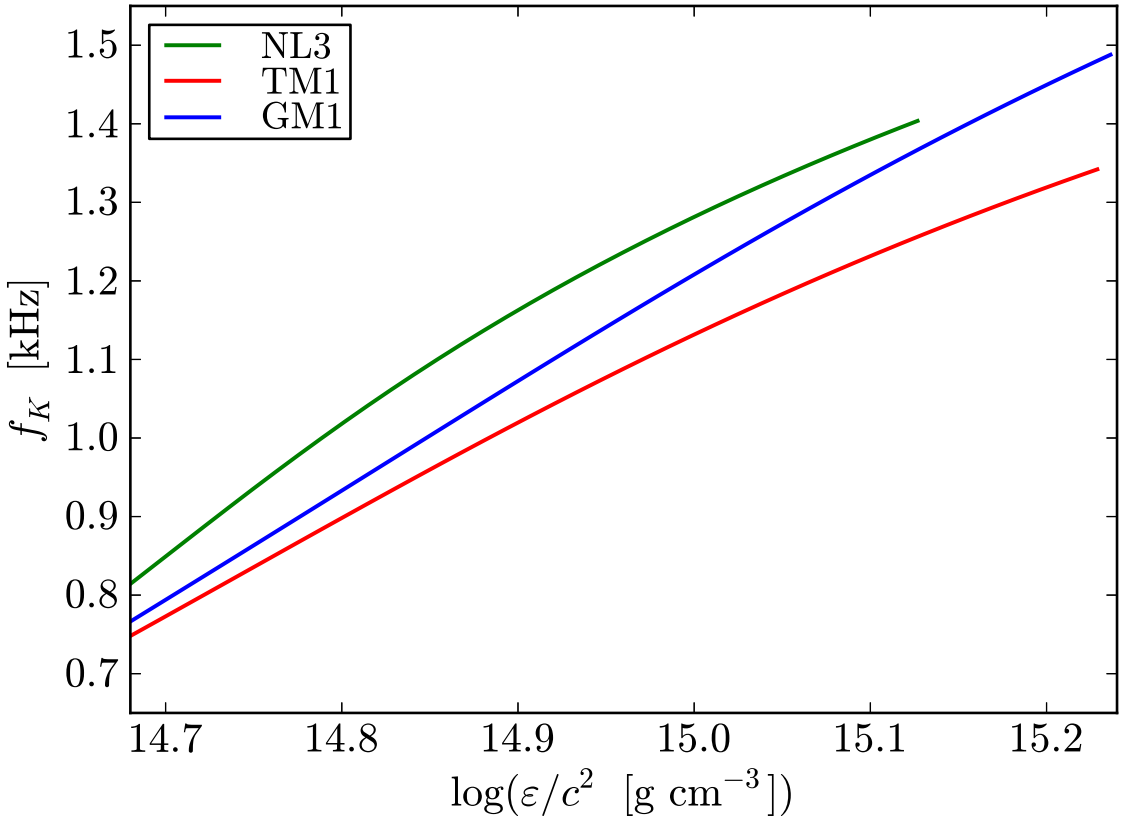

Figure 3.6: Frequency of the maximally rotating configurations (Keplerian sequence) as a function of the total mass, for the EOS NL3, GM1 and TM1. The curves end at the maximum frequency value, which is given by the intersection of the Keplerian sequence with the secular axisymmetric instability.

Another important quantity for this discussion is the dimensionless angular momentum ("Kerr parameter"), $a/M \equiv cJ/(GM^2)$, which we show in Fig. 3.7 as a function of the total mass for the maximally rotating configurations, namely the Keplerian sequence. It can be seen how the maximum value attained by the NS, $(a/M)_{\rm max} \approx 0.7$, holds for all the selected EOS. The maximum value is reached for the mass $[0.96, 1.05, 3.37]\ M_\odot$ for the TM1, GM1, and NL3 EOS, respectively. The existence of such a particular EOS-independent maximum value of a possibly implies the existence of universal limiting values of the NS compactness and the rotational to gravitational energy ratio. This is a conjecture which deserves further exploration. In the same plot, the same sequences obtained with other already known EOS (represented by differently dashed curves), obtained assuming widely

different kind of interactions and via different many-body-theories, are shown, and the reader can notice a general universal behaviour of the dimensionless angular momentum, even if for these other EOS the exact maximal values of this dimensionless parameter are slightly different. On the other hand, such a general behaviour of a parameter in not surprising, in fact, as it was already shown in [14], it can be chosen as a parameter to establish a universal I-Love-Q relation. Nevertheless, the important argument here is that, although different stiffness, our chosen set of EOS presents a common maximal dimensionless angular momentum a/M.

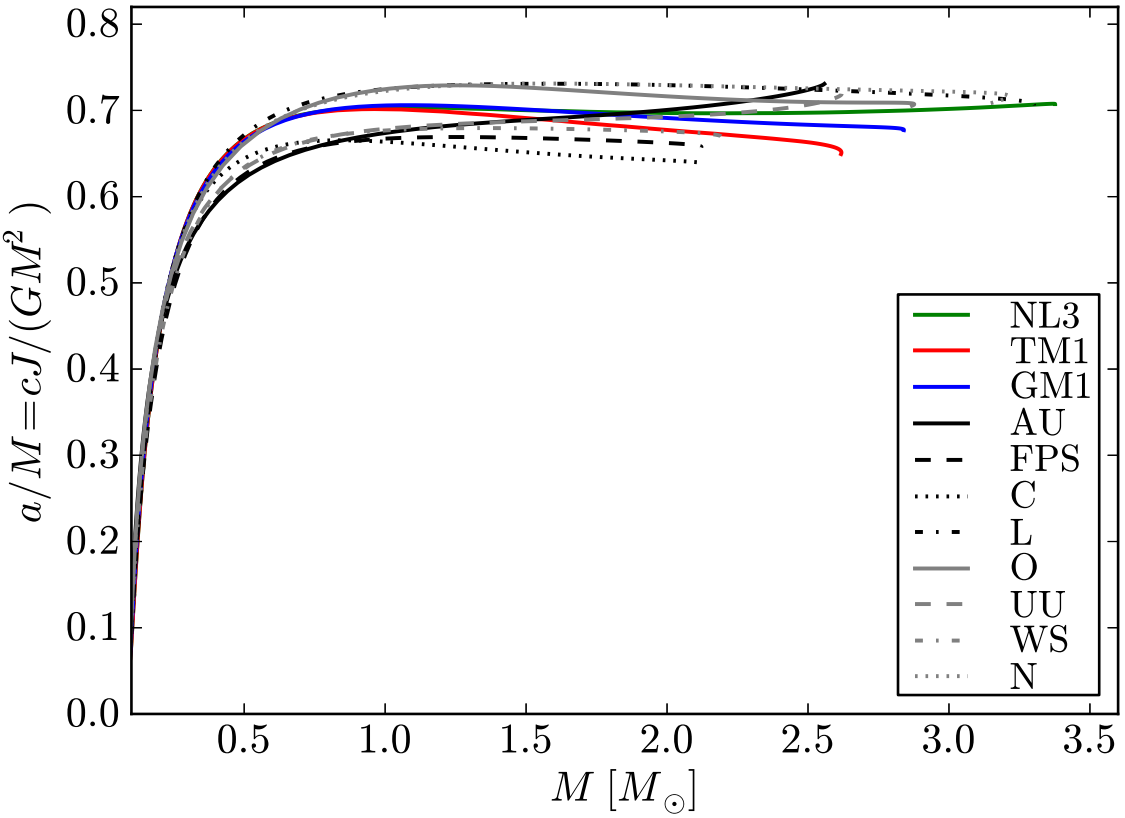

Figure 3.7: Dimensionless angular momentum ("Kerr parameter"), $a/M \equiv cJ/(GM^2)$, as a function of the total mass of the NS. Keplerian sequences obtained with our selection of EOS (colored curves) are plotted with same sequences with other already know EOS (black differently dashed curves), taken from a set supplied together with RNS code. For all these other EOS, the reader should refer to EOS.INDEX file (which can be downloaded on RNS web page), and references therein.

3.3.5 Some fitting formulas

In this section we would like to provide some relations between physically important quantities for rotating NS, which may be useful in astrophysical applications.

An often useful physical quantity to be computed is the binding energy of the configurations or the relation between the baryonic mass and the gravitational mass. For non-rotating NSs, we found that for the three analysed EOS, the following relations hold:

$$\begin{aligned} \frac{M_b}{M_\odot} &\approx \frac{M}{M_\odot} + \frac{13}{200}\left(\frac{M}{M_\odot}\right)^2, \\ \frac{M}{M_\odot} &\approx \frac{M_b}{M_\odot} - \frac{1}{20}\left(\frac{M_b}{M_\odot}\right)^2, \end{aligned} \tag{3.19}$$

where M_b is the baryonic mass, hold, and thus appear to be a universal property. The maximum relative errors obtained for non-rotating sequences of GM1, TM1 and NL3 are respectively 1.4%, 1.3% and 0.99%. For rotating configurations, $M = M(M_b, J)$ or $M_b = M_b(M, J)$, we find that for our set of EOS there is indeed a common relation given by:

$$\frac{M_b}{M_\odot} = \frac{M}{M_\odot} + \frac{13}{200}\left(\frac{M}{M_\odot}\right)^2 \left(1 - \frac{1}{130} j^{1.7}\right), \tag{3.20}$$

which duly generalizes Eq. (3.19), and it is accurate within an error of 2%.

Another important relation which we obtained is the one for the masses of the NSs lying along the secular axisymmetric instability line. Using the dimensionless angular momentum j, defined in equation (3.16) and related to the Kerr parameter by $j = (M/M_\odot)^2\, a$, we obtained

$$M = M_{\rm max}^{J=0}(1 + k j^l), \tag{3.21}$$

where the values of $M_{\rm max}^{J=0}$ are given in Table 3.1, $k = [0.017, 0.011, 0.0060]$ and $l = [1.61, 1.69, 1.68]$ for the EOS TM1, GM1, NL3, respectively, together with some physically relevant features of NSs. The maximum relative errors obtained for values of mass along the secular axisymmetric instability line with respect to fits for each EOS are respectively $[0.33\%, 0.44\%, 0.45\%]$.

Table 3.1: Some properties of NSs for the selected EOS: critical mass for non-rotating case, $M_{\rm max}^{J=0}$, maximum mass in uniform rotation, $M_{\rm max}^{J\neq 0}$, maximum rotation frequency, $f_{\rm max}$, and maximum dimensionless angular momentum ("Kerr parameter"), $(a/M)_{\rm max} \equiv [cJ/(GM^2)]_{\rm max}$.

EOS	$M_{\rm max}^{J=0}$ $[M_\odot]$	$M_{\rm max}^{J\neq 0}$ $[M_\odot]$	$f_{\rm max}$ [kHz]	$(a/M)_{\rm max}$
TM1	2.20	2.62	1.34	0.70
GM1	2.39	2.84	1.49	0.71
NL3	2.81	3.38	1.40	0.71

3.3.6 Eccentricity and Moment of Inertia

In order to see how a figure of equilibrium is deformed by rapid rotation, we compute the eccentricity

$$\epsilon = \sqrt{1 - \left(\frac{R_{\rm pol}}{R_{\rm eq}}\right)^2}, \tag{3.22}$$

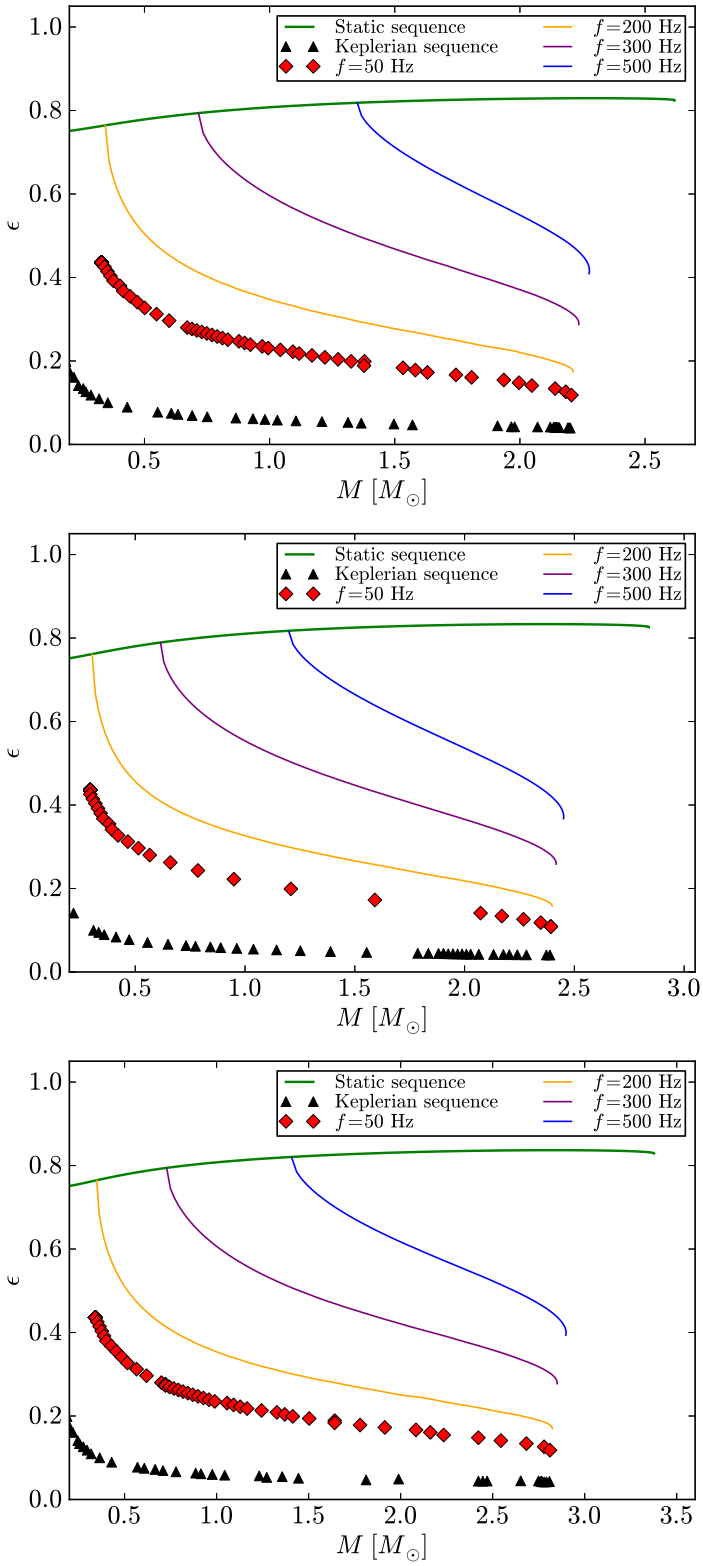

Figure 3.8: Eccentricity versus gravitational mass using EOS with TM1, GM1 and NL3 (from top to bottom) set of parameters for same sequences as in Fig. 3.2.

which in Fig. 3.8 we plot as a function of the mass, M, for the same constant Ω sequences of the previous figures.

It is also interesting to investigate the distribution of the energy density within the figure of equilibrium both the static and rotational case for the different EOS. In figure 3.9, we show the contours of constant energy density of a model with central value $\varepsilon_c = 10^{15}$ g cm^{-3} both in the static case and in the rotational one with dimensionless angular momentum $j = 4$, for the sake of example for the GM1 EOS.

We turn now to compute the moment of inertia of the star, which is one of the most relevant properties in pulsar analysis. The moment of inertia can be estimated as [59]

$$I = \frac{J}{\Omega}, \tag{3.23}$$

where J is the star angular momentum which is given by Eq. (3.17).

In figure 3.10 we plot the moment of inertia as a function of the mass for some Ω-constant sequences together with the Keplerian sequence, while in figure 3.11 we show the relations between I and the compactness, $GM_*/(c^2R_*)$, where M_* and R_* are the mass and the radius of the spherical configuration with the same central density as the rotating one, ε_c.

The above figures confirm that for rotation frequencies $\lesssim 200$ Hz, or rotation periods $\gtrsim 5$ ms, the deformation of the star is very little and, indeed, the non-rotating or the slow rotation regimes can be safely adopted as accurate approximations of the rotating NS.

3.3.7 Quadrupole Moment

The quadrupole moment in the RNS code is given by

$$M_2 = \frac{1}{2} {r_{eq}}^3 \int_0^1 \frac{{s'}^2 ds'}{(1-s')^4} \int_0^1 P_2(\mu') \tilde{S}_\rho(s', \mu') d\mu', \tag{3.24}$$

where r_{eq} is the value of the coordinate radius at equator, $\rho \equiv 2\nu - \ln(B)$, $s = r/(r + r_{eq}) \in [0, 1]$ is a compacted radial coordinate, $\mu = \cos(\theta)$, $P_2(\mu)$ is the Legendre polynomial of second order, and $\tilde{S}_\rho = r^2 S_\rho$, being S_ρ a source function defined as

$$\begin{aligned} S_\rho(r,\mu) = \quad & e^{\frac{\gamma}{2}} \left[8\pi e^{2\lambda} (\varepsilon + P) \frac{1+u^2}{1-u^2} + r^2 e^{-2\rho} \left[\omega_{,r}^2 + \frac{1}{r^2}(1-\mu^2)\omega_{,\mu}^2 \right] + \frac{1}{r}\gamma_{,r} - \frac{1}{r^2}\mu\gamma_{,\mu} \right. \\ + \quad & \left. \frac{\rho}{2} \left\{ 16\pi e^{2\lambda} - \gamma_{,r} \left(\frac{1}{2}\gamma_{,r} + \frac{1}{r} \right) \frac{1}{r^2}\gamma_{,\mu} \left[\frac{1}{2}\gamma_{,\mu}(1-\mu^2) - \mu \right] \right\} \right], \end{aligned} \tag{3.25}$$

with $\gamma = \ln(B)$. However, as shown in Ref. [54], Eq. (3.24) is not the actual quadrupole moment of the rotating source according to the Geroch-Hansen multipole moments [29, 30, 33]. Indeed, the quadrupole moment extracted via the Ryan's expansion method [58] is [54, 67]

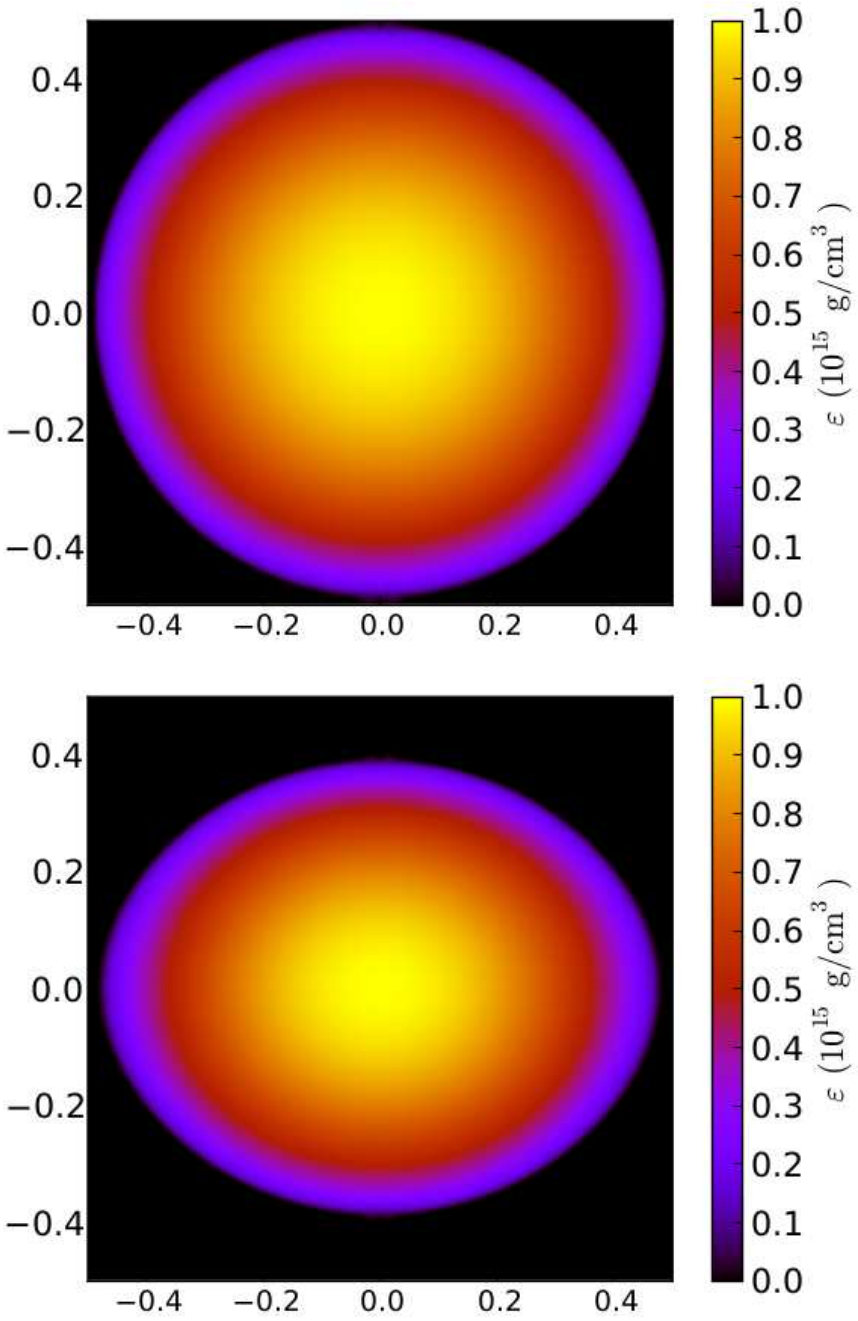

Figure 3.9: Contours of constant energy density of a model with central value $\varepsilon_c = 10^{15}$ g cm^{-3} both in the static case (top panel) and in the rotational one with dimensionless angular momentum $j = 4$ (bottom panel) for the GM1 EOS.

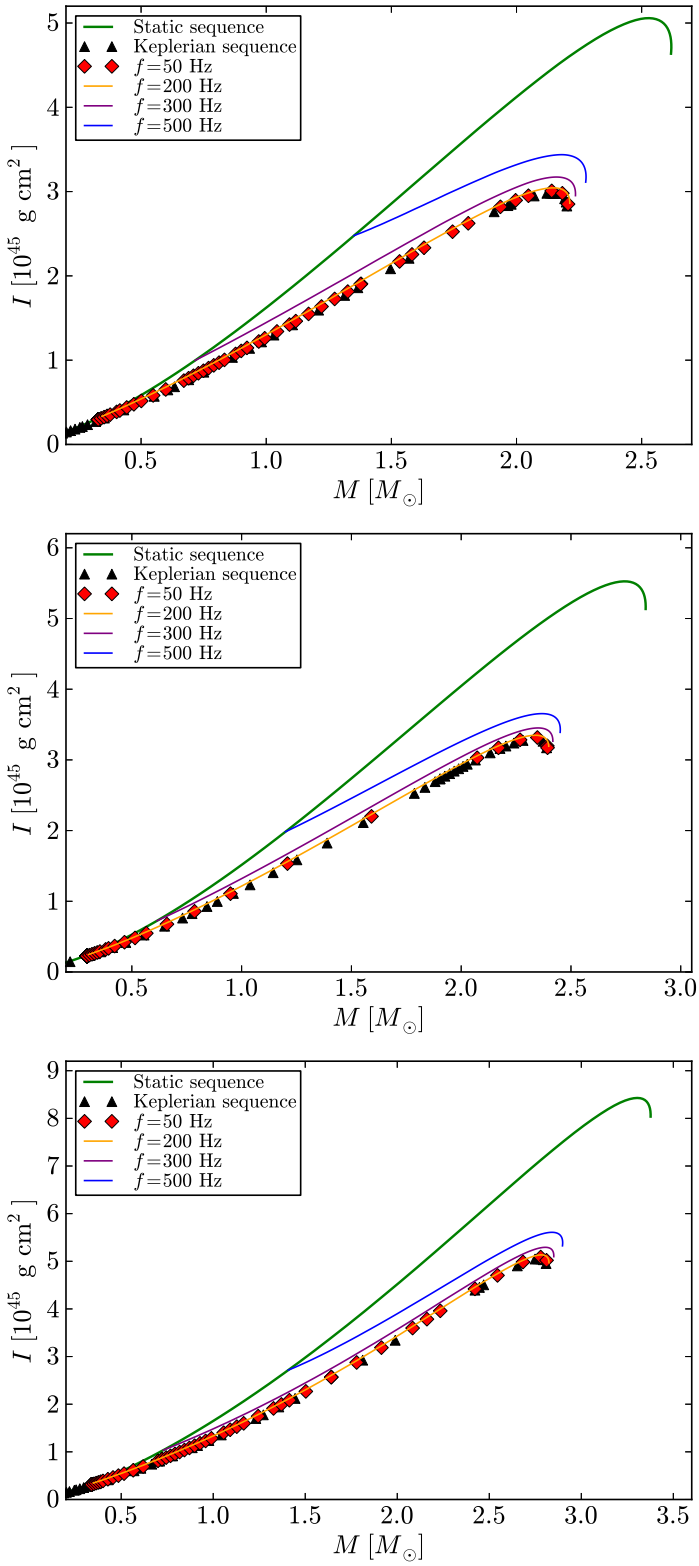

Figure 3.10: Results of moment of inertia versus mass for NS model obtained using EOS TM1, GM1, and NL3 (from top to bottom) set of parameters for same sequences as in figure 3.2

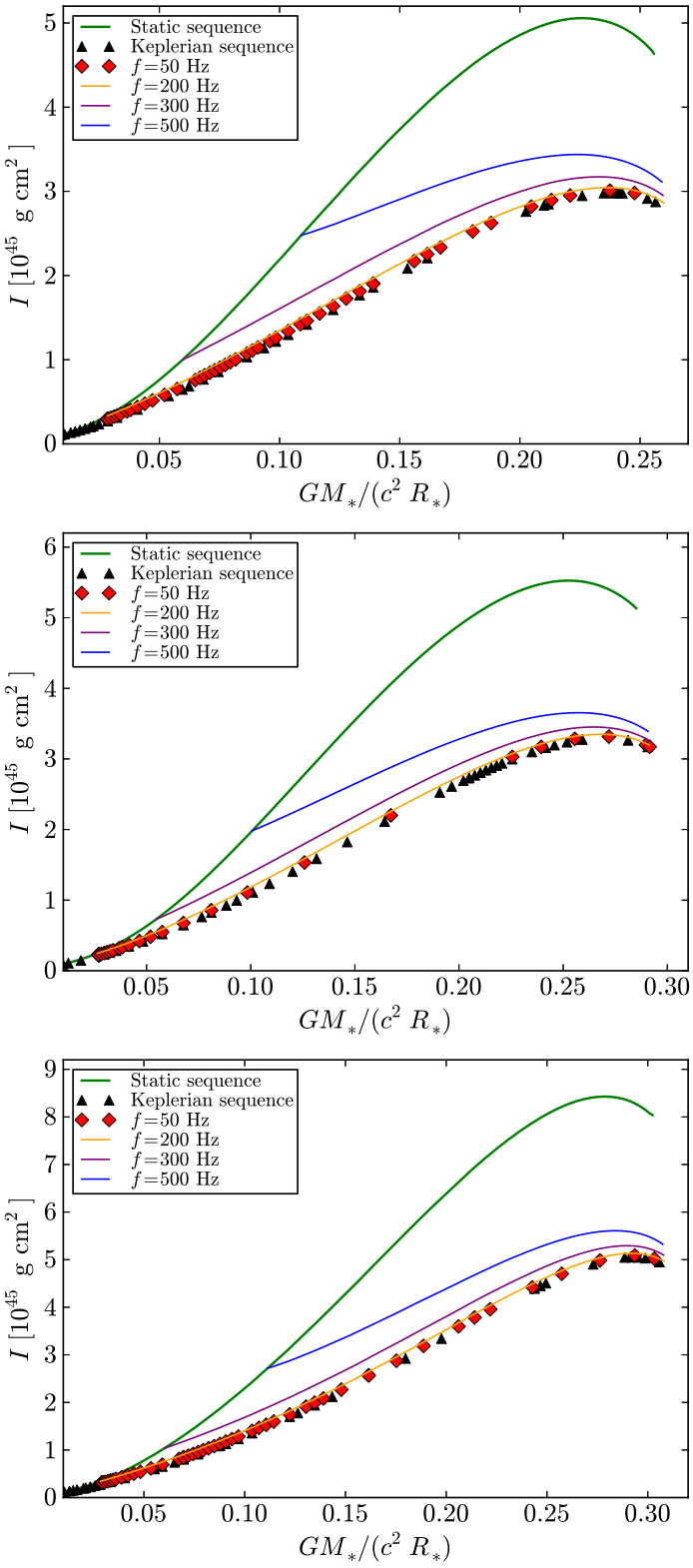

Figure 3.11: Same as Fig. 3.10 but for moment of intertia plotted against compactness.

$$M_2^{\rm corr} = M_2 - \frac{4}{3}\left(\frac{1}{4} + b_0\right) M^3, \tag{3.26}$$

$$\begin{aligned} b_0 &= -\frac{16\sqrt{2\pi}{r_{eq}}^4}{M^2}\int_0^{\frac{1}{2}} \frac{{s'}^3 ds'}{(1-s')^5} \\ &\times \int_0^1 d\mu' \sqrt{1-{\mu'}^2} P(s',\mu') e^{\gamma+2\lambda} T_0^{\frac{1}{2}}(\mu'), \end{aligned} \tag{3.27}$$

where M_2 is given by Eq. (3.24) and $T_0^{\frac{1}{2}}$ is the Gegenbauer polynomial of order 0 with normalization $T_0^{1/2} = \sqrt{2/\pi} C_0$, with C_0 the traditional 0th-order Gegenbauer polynomial.

Following Refs. [54, 67], we computed numerically the correcting factor b_0 given by Eq. (3.27), and then obtained the corrected quadrupole moment through Eq. (3.26). In figure 3.12 the modulus of $M_2^{\rm corr}$ is plotted in logarithmic scale against the gravitational mass for selected constant frequency sequences. Each sequence was stopped at the secular instability limit. We can see that the quadrupole moment is a decreasing function of the mass along a constant frequency sequence while it is an increasing function along the Keplerian sequence.

We turn to compare and contrast the above mass quadrupole moment with the one of the Kerr solution,

$$M_2^{\rm Kerr} = \frac{J^2}{M}. \tag{3.28}$$

The reason for this is twofold. First, we point out the long-time discussed question in astrophysics if the Kerr solution may describe the exterior gravitational field of a realistic astrophysical source besides a black hole, namely, is there any matter content which could generate a Kerr exterior field? (see, e.g., Refs. [50, 11], and references therein). Second, if the answer to the previous question is negative, then one can distinguish a NS from a black hole with the same mass and angular momentum from the knowledge of the quadrupole moment (see, e.g., Ref. [51], and references therein).

In Fig. 3.13 we show the ratio between the NS quadrupole moment, $M_2^{\rm corr}$ given by equation (3.26), and the Kerr solution quadrupole moment, $M_2^{\rm Kerr}$, for selected constant frequency sequences. We find that $M_2^{\rm corr}$ starts to approach $M_2^{\rm Kerr}$, as intuitively expected, for masses close to the maximum stable value. An interesting feature that we can see from Fig. 3.13 is that the stiffer the EOS the more the quadrupole moment approaches the Kerr value. This result is well in accordance with previous results that showed that the compactness of the star increases, also the moment of inertia, Love numbers and mass quadrupole approach the ones of a black hole though they will never coincide (see, e.g., Ref. [68]). Moreover, we confirm in the full rotation regime, the previous result obtained in the slow-rotation Hartle's approximation [65], that the ratio $M_2^{\rm corr}/M_2^{\rm Kerr}$ is a decreasing function of the NS mass, hence reaching its lowest value at the maximum mass configuration. Indeed, as we can see from Fig. 3.13, the largest the maximum mass attained by a

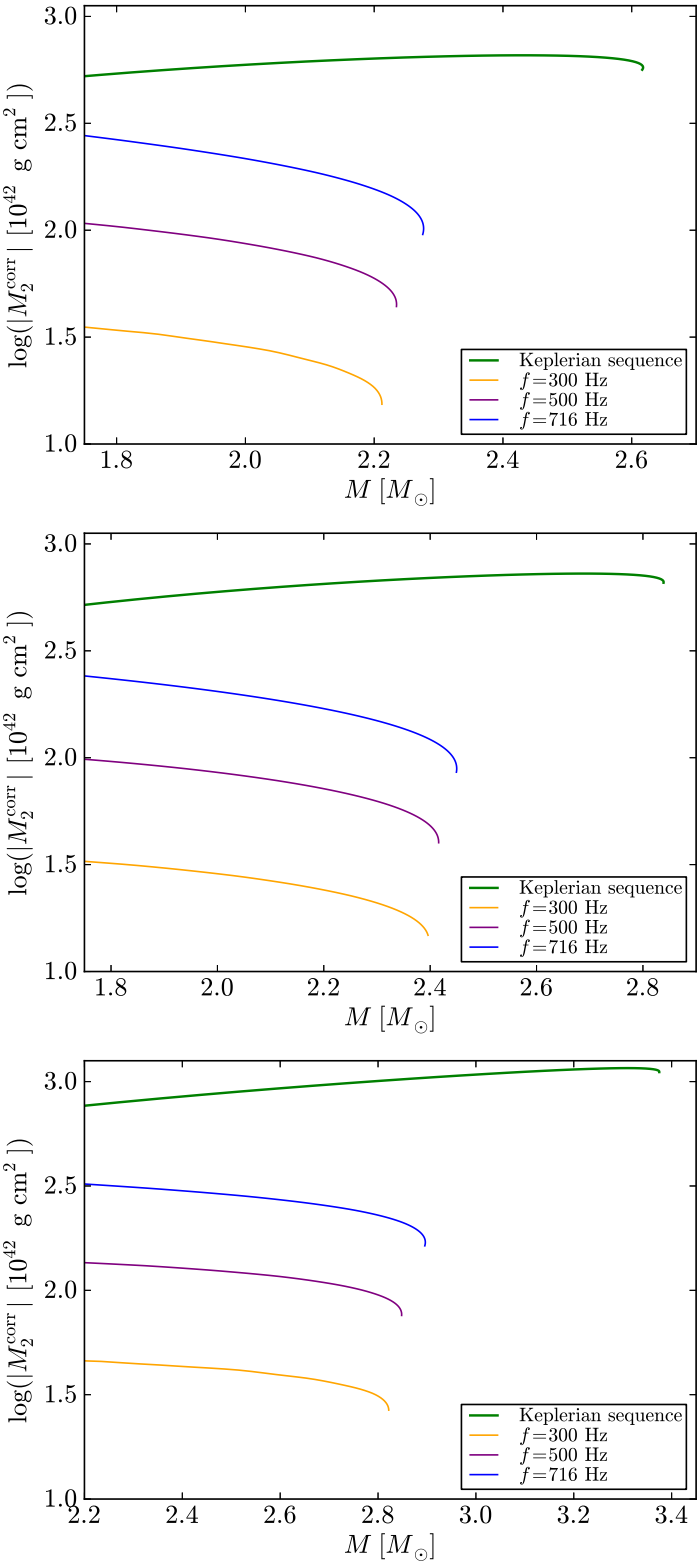

Figure 3.12: The modulus of corrected value for the mass quadrupole (in logarithmic scale) obtained via Eq. (3.26) is plotted against gravitational mass for the same constant frequency sequences of Fig. 3.3.

NS model, the closest its quadrupole moment approaches the Kerr solution value, reaching even values < 1.5 for stiff EOS such as the NL3 model.

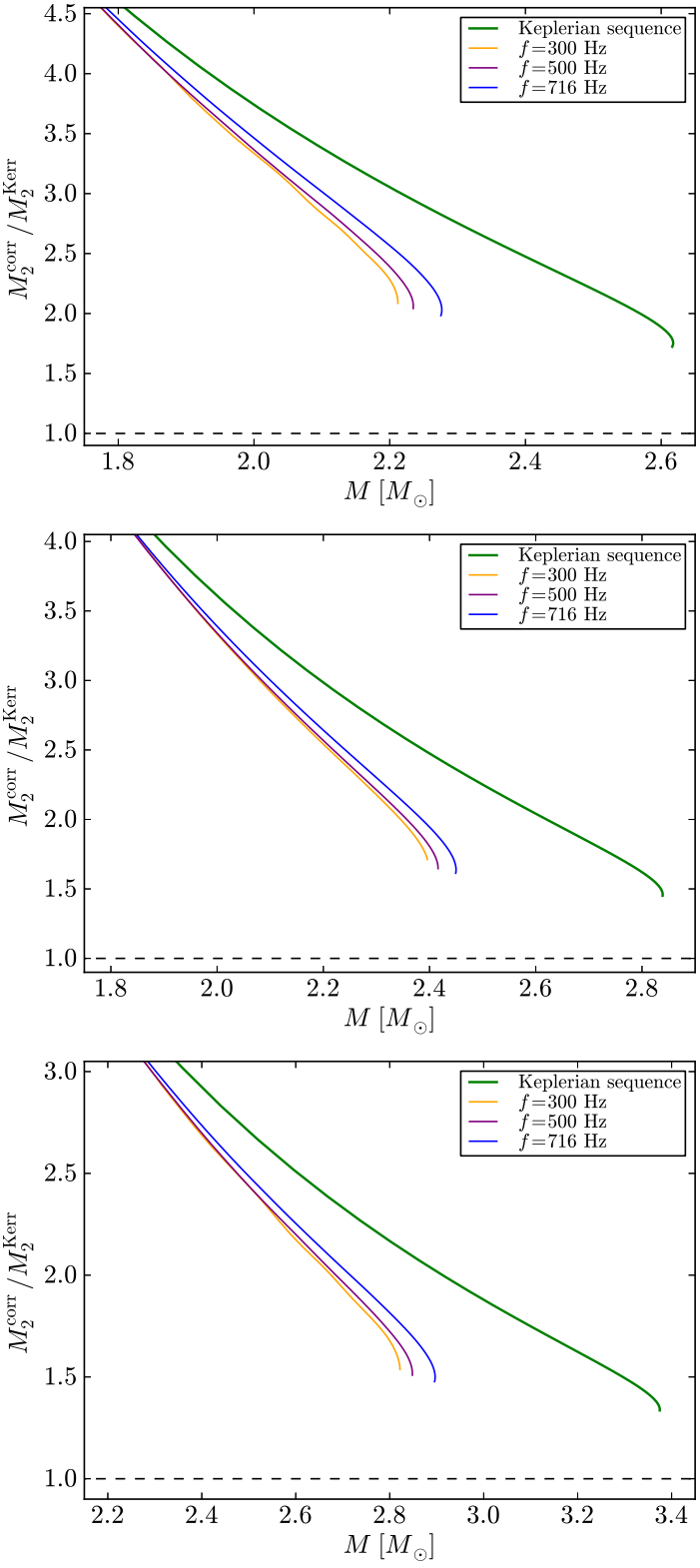

Figure 3.13: $M_2^{\rm corr}/M_2^{\rm Kerr}$ ratio for the same selected sequences of constant frequency of Fig. 3.12 and EOS TM1, GM1, and NL3 (from top to bottom). We show here only the region of large masses where $M_2^{\rm corr}$ starts to approach the Kerr value $M_2^{\rm Kerr}$.

Chapter 4

The Orbits of Particles around Rapidly Rotating NSs

4.1 Introduction

In the last chapter we had seen how numerical models of rapidly rotating NSs can be obtained, showing that these contain both the stars internal structure's information (e. g. density distribution and shape) and external spacetime's information (e. g. the quadrupole mass moment of the source).

Now let's move to a possible interesting application of the aforementioned results. In general one could study orbits on the equatorial plane of particles around a rotating source (a Black Hole or a Neutron Star), to understand what are the conditions that particles must satisfy in order to maintain a stable equilibrium when rotating in such orbits. In particular, we will treat the problem of Innermost Stable Circular Orbits (ISCO) around NSs, reaching some important relations for binding energy and angular momentum of particles which can be useful in some astrophysical applications (e.g. see [6]).

4.2 Innermost Stable Circular Orbits

In the present section, we are interested in analysing the stability of circular orbits of particles in the equatorial plane. A practical way to do such an analysis, locating the radii of marginally stable orbits, is given in [2], where all concern the study of the effective potential as function of radius r, energy per unit mass $\tilde{E}$ and angular momentum per unit mass $\tilde{L}$ of particles in such orbit, $V(r, \tilde{E}, \tilde{L})$.

As reported in [19], with metric given by Eq. (3.2) one can express the effective potential $V(r, \tilde{E}, \tilde{L})$, as follows:

$$V(r, \tilde{E}, \tilde{L}) = e^{2\lambda+\gamma}\left(\frac{dr}{d\tau}\right)^2 = e^{-\rho}\left(\tilde{E} - \omega\tilde{L}\right)^2 - e^{\gamma} - \frac{e^{\rho}}{r^2}\tilde{L}^2, \tag{4.1}$$

being τ the proper time. In order to obtain a circular orbit, one should impose the conditions

$$V = V_{,r} = 0, \tag{4.2}$$

and from Equations (4.1) and (4.2), one obtains:

$$\tilde{E} - \omega\tilde{L} = \frac{\tilde{v}e^{\frac{\gamma+\rho}{2}}}{(1-\tilde{v}^2)^{\frac{1}{2}}}, \tag{4.3}$$

$$\tilde{L} = \frac{\tilde{v}re^{\frac{\gamma-\rho}{2}}}{(1-\tilde{v}^2)^{\frac{1}{2}}}, \tag{4.4}$$

where $\tilde{v}$ is the velocity as measured by the ZAMO, for which the following relation holds:

$$\tilde{v} = \frac{1}{2 + r(\gamma_{,r} - \rho_{,r})}\left\{e^{-\rho}r^2\omega_{,r} \pm \left[e^{-2\rho}r^4{\omega_{,r}}^2 + 2r(\gamma_{,r} + \rho_{,r}) + r^2({\gamma_{,r}}^2 - {\rho_{,r}}^2)\right]^{\frac{1}{2}}\right\}, \tag{4.5}$$

in which the plus sign is for co-rotating particles, while minus sign is for counter-rotating particles.

It might happen that there is a minimum radius, external to the surface of the star, for which no stable circular orbit may exist in the region from the stellar surface up to this radius: we shall denote this radius to as $r_{\rm mbo}$, the *mostly bound orbit* (MBO), although in the literature it is usually indicated as *innermost stable circular orbit* (ISCO). Obviously the aforementioned situation is the case when $r_{\rm mbo} > r_{\rm eq}$, being $r_{\rm eq}$ the coordinate equatorial radius of the star. If instead $r_{\rm mbo} \leq r_{\rm eq}$, it means that each circular orbit down to the stellar surface is stable. Precisely the value of $r_{\rm mbo}$, is the one that minimizes the effective potential $V(r, \tilde{E}, \tilde{L})$, thus in principle one should check if $V_{,rr} \geq 0$, in which the equality is obtained for the value of $r_{\rm mbo}$, considering $\tilde{E}$ and $\tilde{L}$ as given by Equations (4.3) and (4.4).

4.2.1 BH Case

To better understand the physics of the spacetime external to a NS, it can be helpful to compare the binding energy and the angular momentum of the MBO of such compact objects with the corresponding of a Kerr black hole. The problem of the rotating Kerr black hole was treated in [3], where the effective potential in the equatorial plane in Boyer-Lindquist coordinates is given, in terms of mass M, Kerr parameter a and radius r, by

$$V(r, \tilde{E}, \tilde{L}) = \left(\tilde{E}(r^2 + a^2) - \tilde{L}a\right)^2 - \left(r^2 - 2Mr + a^2\right)\left(r^2 + (\tilde{L} - \tilde{E}a)^2\right), \tag{4.6}$$

and the respective of equations (4.3) and (4.4) for particle's energy and angular momentum per unit mass, are written as

$$\tilde{E} \;=\; \frac{r^{\frac{3}{2}} - 2Mr^{\frac{1}{2}} \pm aM^{\frac{1}{2}}}{r^{\frac{3}{4}}\left(r^{\frac{3}{2}} - 3Mr^{\frac{1}{2}} \pm 2aM^{\frac{1}{2}}\right)^{\frac{1}{2}}}, \tag{4.7}$$

$$\tilde{L} = \frac{\pm M^{\frac{1}{2}}\left(r^2 \mp 2aM^{\frac{1}{2}}r^{\frac{1}{2}} + a^2\right)}{r^{\frac{3}{4}}\left(r^{\frac{3}{2}} - 3Mr^{\frac{1}{2}} \pm 2aM^{\frac{1}{2}}\right)^{\frac{1}{2}}}. \quad (4.8)$$

In order to find $r_{\rm mbo}$ for a Kerr black hole, one has to solve $V_{,rr} = 0$, with $\tilde{E}$ and $\tilde{L}$ given by Equations (4.7) and (4.8). The so-called extreme solution of a Kerr black hole corresponds to the limit $a = M$, and the solutions are $r_{\rm mbo+} = M$ (for the co-rotating case) and $r_{\rm mbo-} = 9M$ (for the counter-rotating case). For a non-rotating Kerr Black Hole ($a = 0$) one obtains the already known Schwarzschild solution $r_{\rm mbo\pm} = 6M$.

In Table 4.1 we summarize known properties for the analytical relativistic solutions (namely the Schwarzschild and Kerr ones).

Table 4.1: Binding energy and dimensionless angular momentum of particles at the mostly bound circular orbit in the Schwarzschild and Kerr exact solutions for both co- (sign +) and counter-rotating (sign -) orbits.

	Schwarzschild ±	Extreme Kerr +	Extreme Kerr −
$\frac{r}{M}$	6.0	1.0	9.0
$1 - \frac{E_{bind}}{\mu}$	5.72 %	42.35 %	3.77 %
$\frac{L}{\mu M}$	$2\sqrt{3}$	$\frac{2}{\sqrt{3}}$	$-\frac{22}{3\sqrt{3}}$

4.2.2 Numerical Results for NS

We here present the numerical results obtained through integrations performed with RNS public code, for NS figures of equilibrium, considering mass-constant sequences that begin with the spherically symmetric case (non-rotating) up to the Keplerian one (just before mass-shedding occurs).

In Fig.s 4.1, 4.2 and 4.3 results of our computations are shown: in top panels we present the binding energy against "Kerr parameter" relations for the three EOS implemented for mass-constant sequences; in bottom panels we present the modulus of angular momentum of particles in the MBO against "Kerr parameter" relations for same EOS and same masses. Regarding the binding energy one can note that, as expected, each sequence start from the Schwarzschild solution but neither reaches the extreme Kerr values nor the Kerr Black Hole solution for values of the dimensionless angular momentum approaching the maximum one. Accordingly, concerning angular momentum of particles the situation presented confirms this result as the NS solution, although a gradual approach with the growth of Kerr parameter, is always different from the Kerr one.

As one can notice from Fig.s 4.1, 4.2 and 4.3 the binding energy and the angular momentum of rotating NS seem to be power-law functions of mass and "Kerr parameter". Indeed, we find that for the three EOS implemented the following relations are universal:

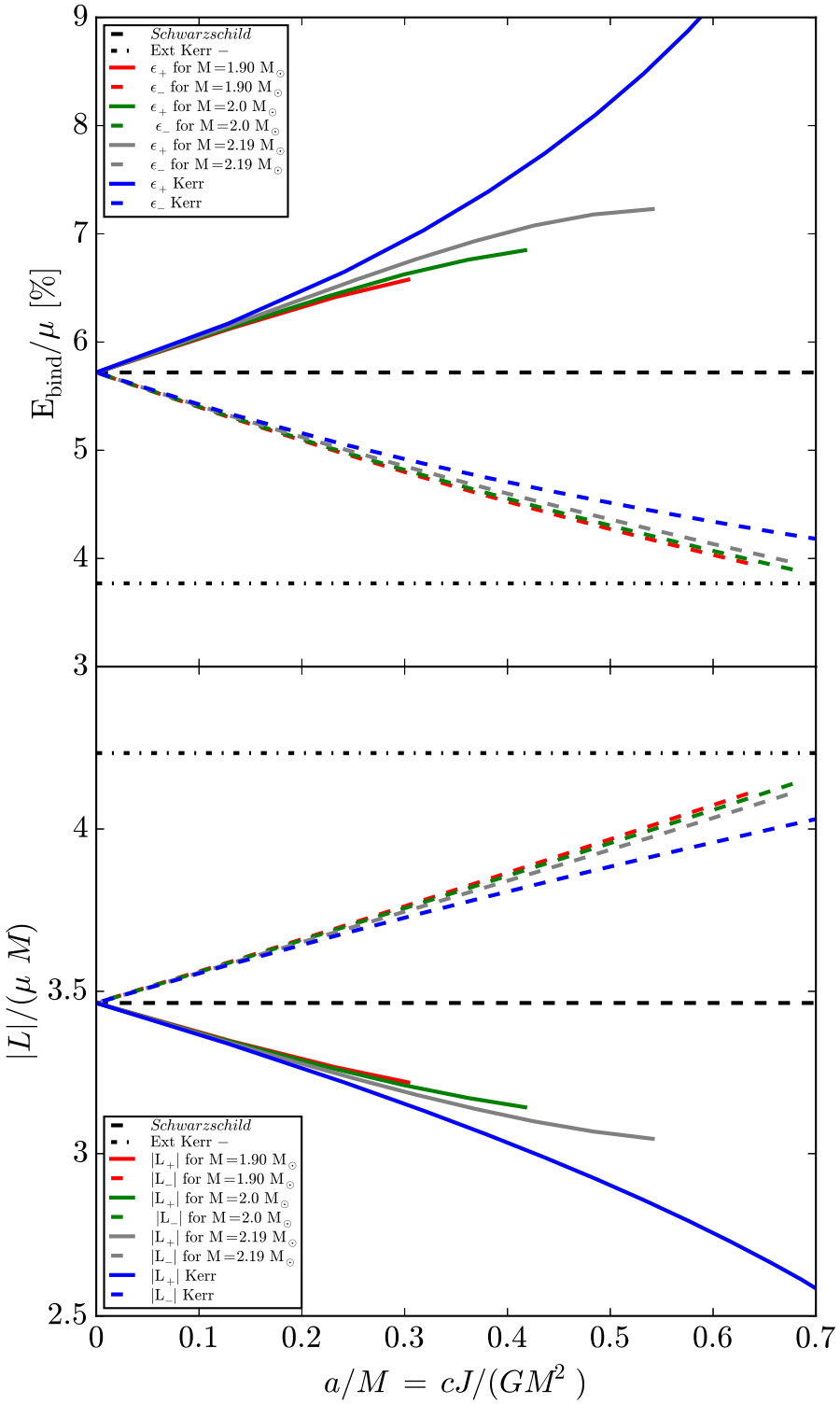

Figure 4.1: Binding energy and angular momentum of particles in the MBO for mass constant sequences of NS configurations plotted against "Kerr parameter", using TM1 EOS, compared to the Schwarzschild and Kerr solutions.

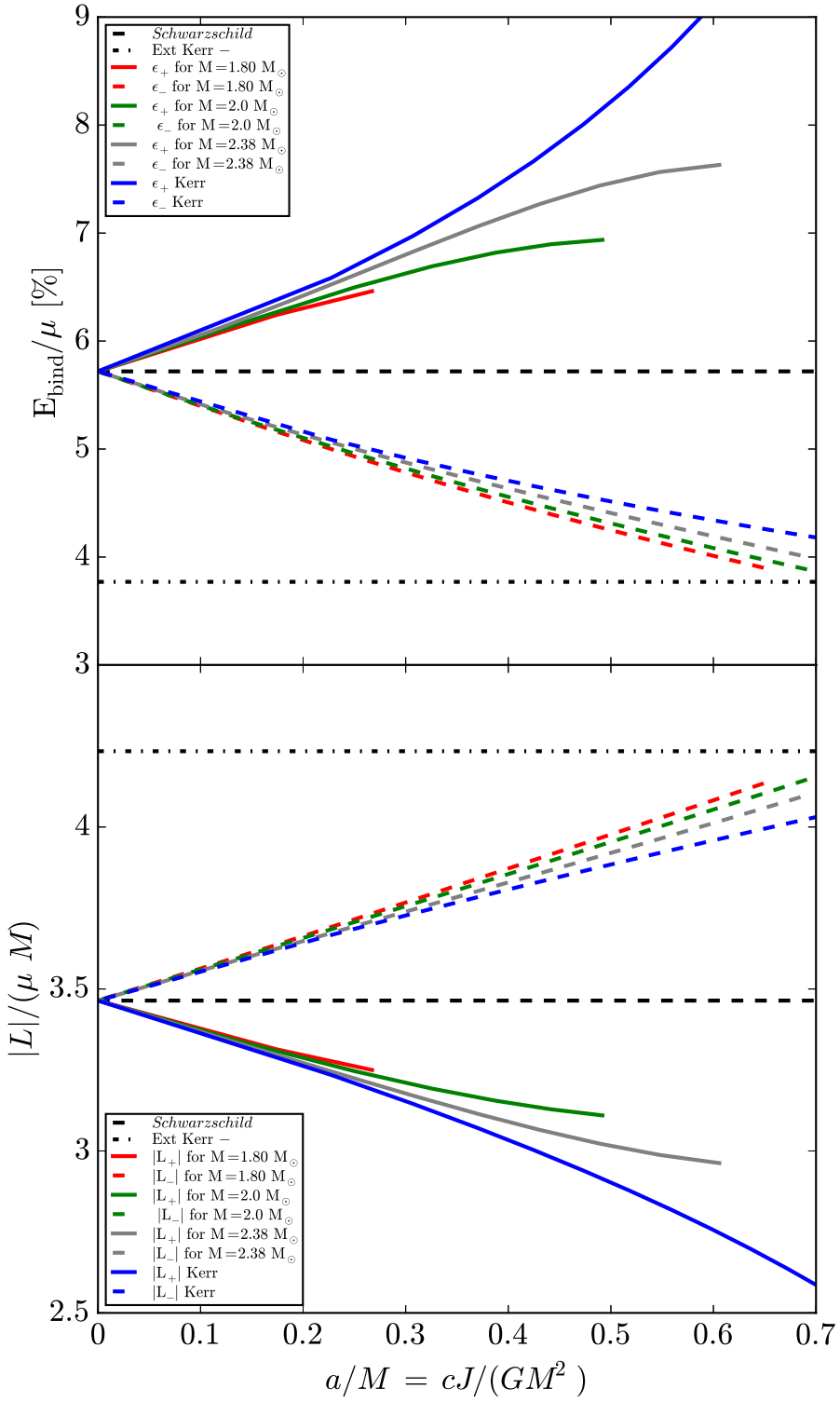

Figure 4.2: Same as Fig. 4.1 but for GM1 EOS.

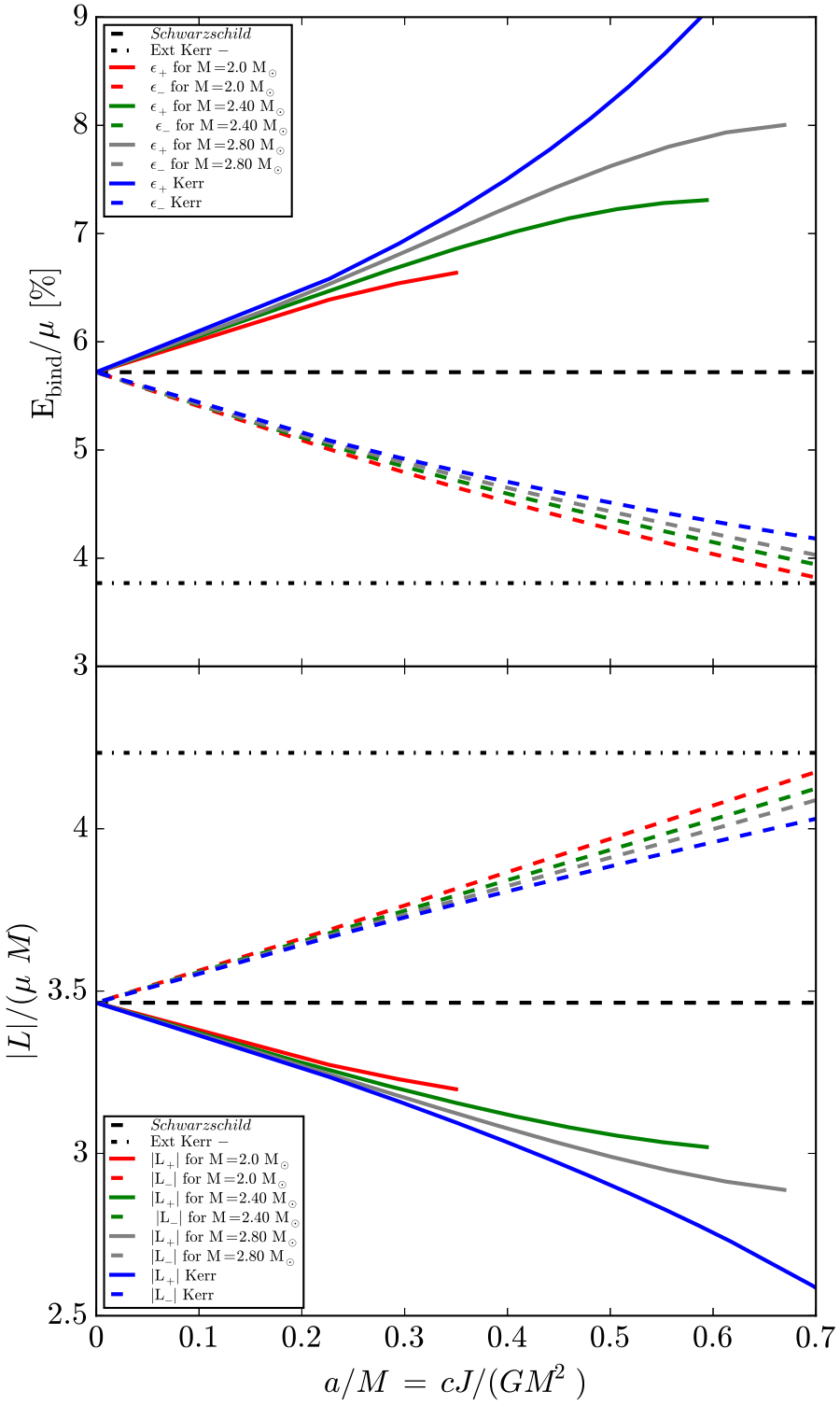

Figure 4.3: Same as Fig. 4.1 but for NL3 EOS.

$$\tilde{E} = 0.9428 - 0.0132 \left(\frac{j}{M/M_\odot} \right)^{0.85}, \tag{4.9}$$

$$\tilde{L} = 3.464 - 0.37 \left(\frac{j}{M/M_\odot} \right)^{0.85}, \tag{4.10}$$

where $j = cJ/(GM_\odot^2) = (a/M)(M/M_\odot)^2$, hence $[j/(M/M_\odot)] = [a/M \times M/M_\odot] = 1$. Note that in the non-rotational (static) case, one obtains the values of $\tilde{E}$ and $\tilde{L}$ for the Schwarzschild solution. These kinds of fit have been performed using constant mass sequences below the critical static mass, but the maximum relative error for each value of mass, even for supra-massive sequences, reaches 1% only near the maximum rotational mass, thus when the mass-shedding limit is approached. In Fig. 4.4, one can see the percent maximum relative error for the two fits, as a function of the mass. It is worth mentioning the fact that for supramassive sequences the maximum value for the error is obtained when the "Kerr parameter" (i.e. the dimensionless angular momentum) approaches to the maximum possible value, thus near the mass-shedding limit for a fixed mass sequence.

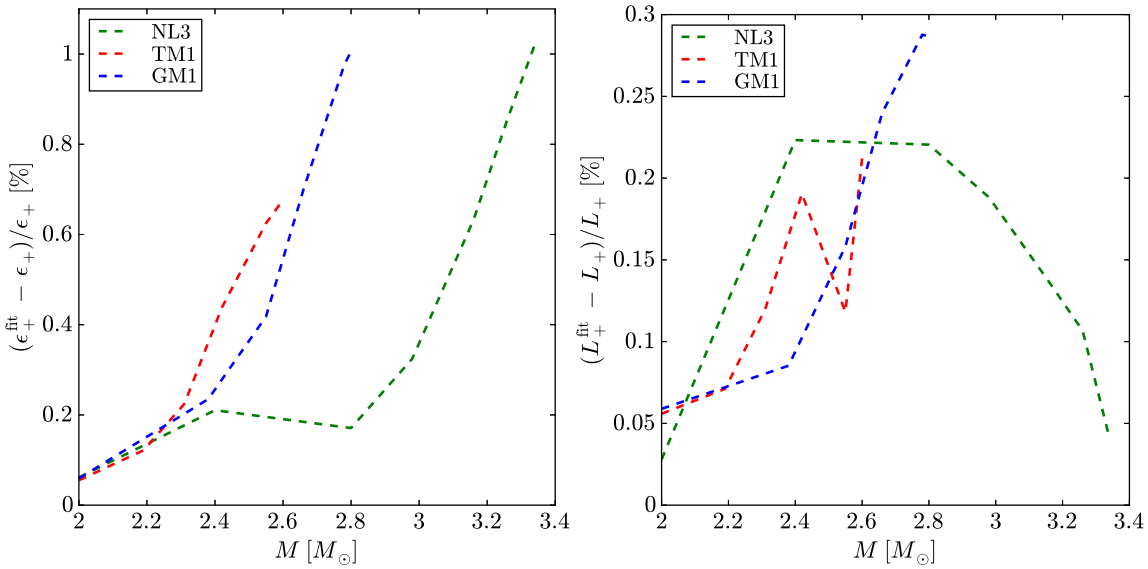

Figure 4.4: Maximum relative error of fits given by Eq. (4.9) (left panel) and Eq. (4.10) (right panel) as function of the mass.

4.2.3 Existence of an exterior mostly bound circular orbit

Having analysed the problem of circular orbits of test particles around rotating relativistic NSs and knowing how to find a minimum for the radius of such orbits, one may ask if they always reside outside the surface, or if there are particular physical limitation for this to happen. In the latter case, it would then be clear that the condition to have the MBO lying outside the NS, would establish a minimum mass for a given value of the angular momentum, or conversely, for a given mass there

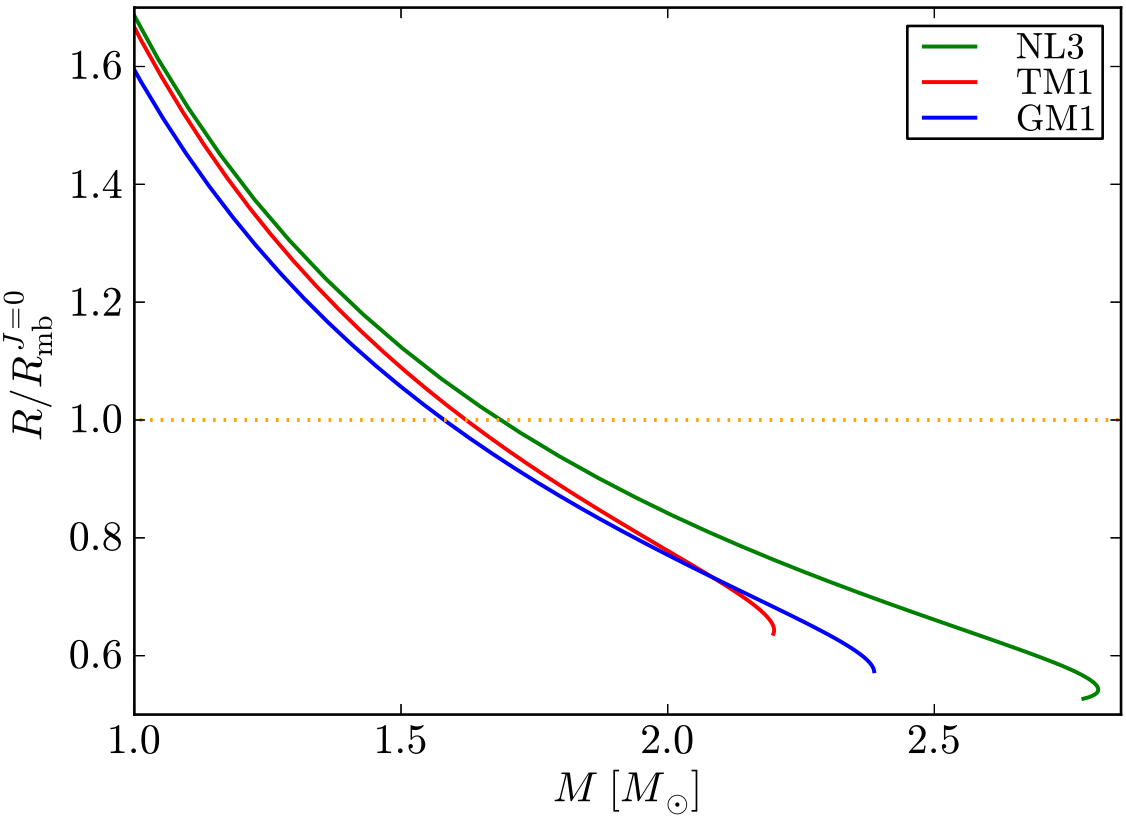

Figure 4.5: Ratio of the radius of non-rotating NSs, R, to the radius of the MBO of a static object, $R_{\rm mb}^{J=0} = 6GM/c^2$, as a function of the NS mass for the selected TM1, GM1 and NL3 EOS.

would be a maximum angular momentum. In the case of $J = 0$, namely for the static case, it is known that the MBO is located at $R_{\rm mb}^{J=0} = 6GM/c^2$, and therefore, the minimum mass to have this orbit outside the star is obtained for the configuration with radius $R = R_{\rm mb}^{J=0}$. For the TM1, GM1 and NL3 EOS, this minimum mass is $[1.78, 1.71, 1.67]\ M_\odot$, respectively (see Fig. 4.5).

Fig.s 4.6, 4.7 and 4.8 show the results in the rotating case. The stable NS models obtained using the TM1, GM1 and NL3 EOS reside in the interior region bounded by the red, green and black curves which are the static, Keplerian, and secular instability limits, respectively. The dashed black curve, instead, separates models with MBO residing outside (right to the dashed curve) and inside (left to dashed the curve).

We can obtain a fitting function from which computes the minimum values of a NS mass, $M_{\rm min}$, for a dimensionless angular momentum, $j = \frac{cJ}{GM_\odot^2}$, being J the NS total angular momentum. We found for the selected EOS the relation:

$$M_{\rm min} = M_{\rm min}^{j=0} + \frac{c_1}{j^{c_2}}, \tag{4.11}$$

where $M_{\rm min}^{j=0}$, c_1 and c_2 are constants that depend on the EOS. We report the values of these fitting parameters in Tab. 4.2 together the maximum percent relative errors and the values of NS mass for which these maxima are obtained.

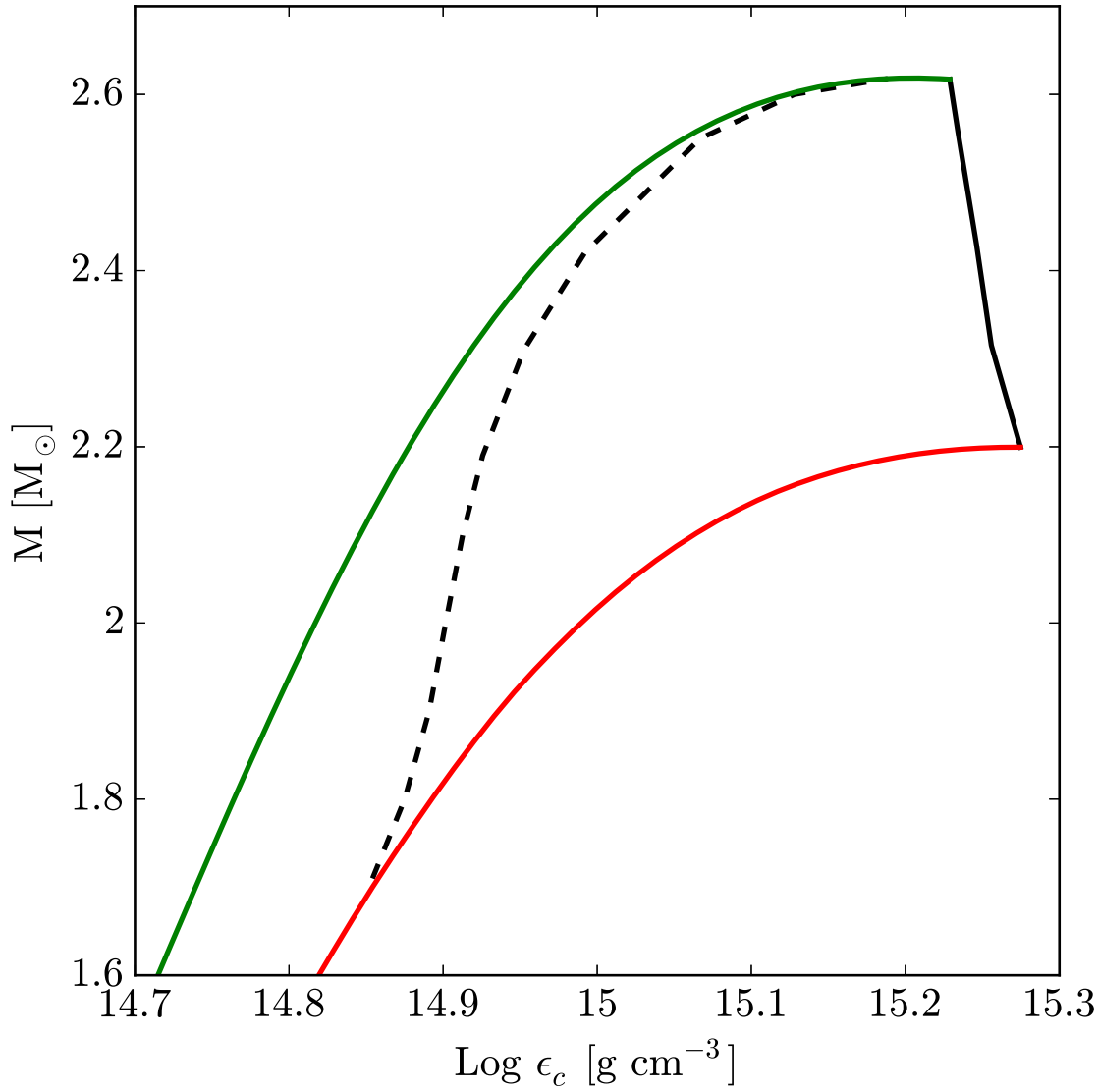

Figure 4.6: Stability region for numerical models of NS obtained using TM1 EOS. The red curve represents the static (non-rotating) sequence, the green one contains models rotating at the mass-shedding (Keplerian) limit and black continued one represents the limit of secular stability against infinitesimal radial perturbation. Dashed black curve is the limit of existence of external ISCO: models on the right side admit one, while on the left side do not.

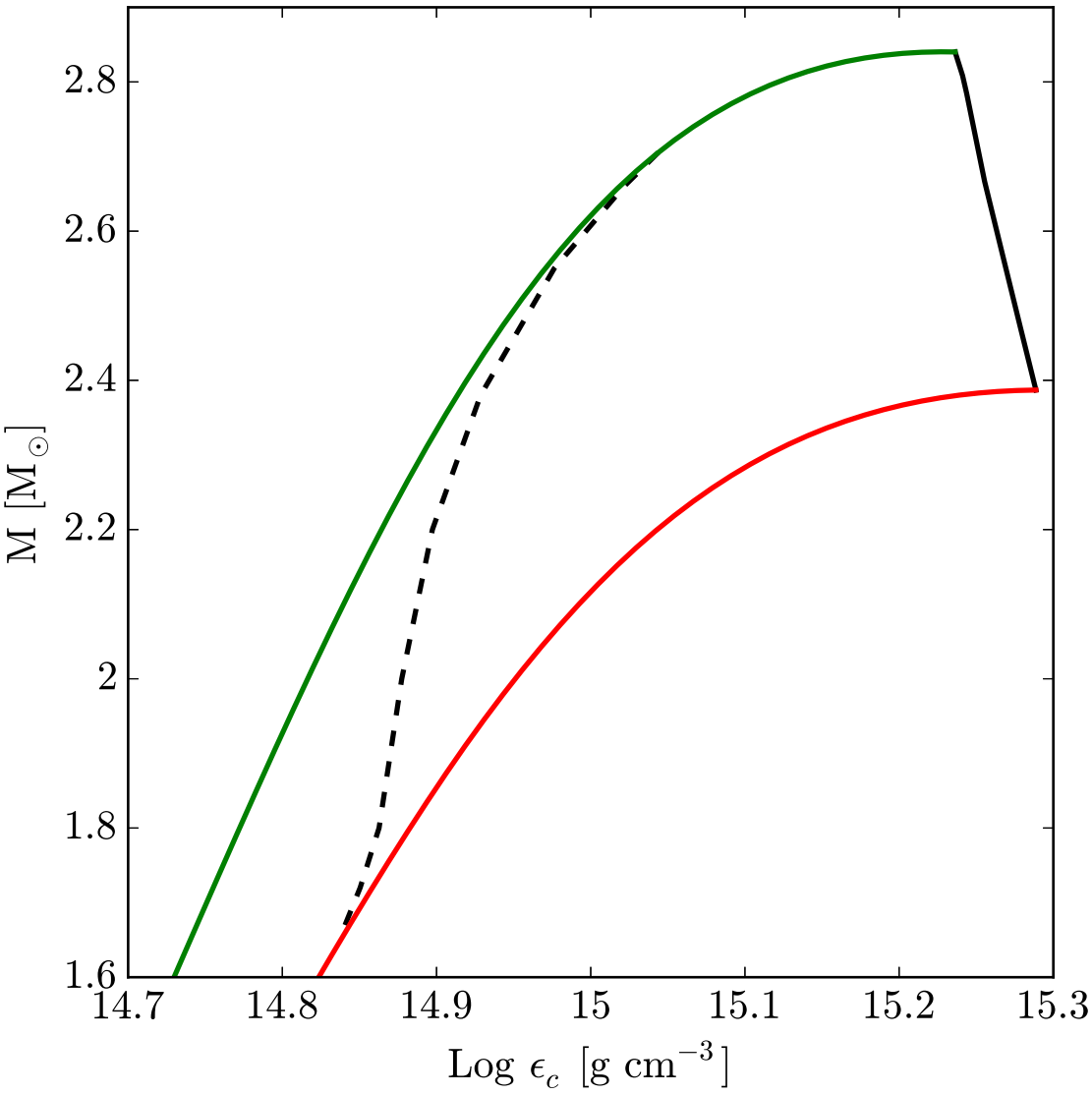

Figure 4.7: Same as Fig. 4.6 but for GM1 EOS.

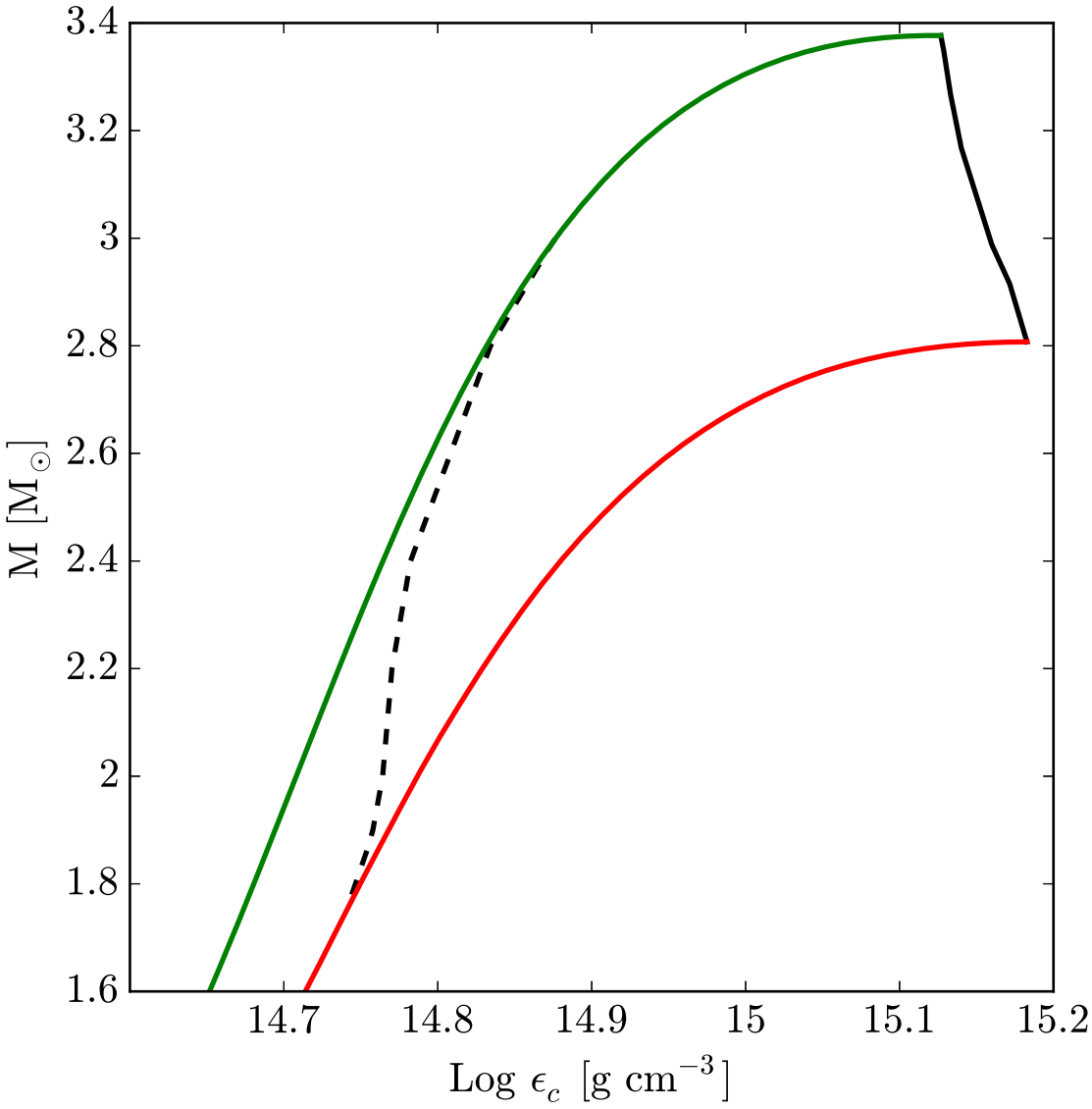

Figure 4.8: Same as Fig. 4.6 but for NL3 EOS.

Table 4.2: Values for constant parameters of fits given by equation 4.11 for the three EOSs used, together with maximum relative errors.

EOS	$M_{\rm min}^{j=0}$	c_1	c_2	Max rel err(%)	$M_{\rm Max\,rel\,err}$
NL3	1.78	0.125	1.235	0.97	2.00
TM1	1.71	0.130	1.30	0.65	1.90
GM1	1.67	0.130	1.30	0.71	1.80

General Conclusions and Future Perspectives

Conclusions

In all previous chapters we had treated the problem of equilibrium for a self-gravitating, rapidly rotating star, noting how numerical methods result to be powerful in approaching this kind of problems, both when working in Newtonian and general Relativistic regimes.

From the Newtonian point of view in Chapters 1 and 2 we had restricted our study to the case of a polytropic EOS, using an already known method (see [24]) to treat both uniform and differential rotation, proving also some results for the two-parameters differential rotation law of Eq. (2.12).

In particular, in Chap. 1, after a brief introduction to the problem of equilibrium of self-gravitating stars in Newtonian gravity, we had outlined the method by Eriguchi and Mueller from ref. [24], in which a numerical approach is adopted to solve this kind of problem with Newton-Raphson iterative method. We had tested the method with a particularly simple rotation law, namely the uniform one of Eq. (1.16) and compared our results with ones obtained by James in [40] with an analytical expansion approach. From this comparison, we understood that, even if the numerical approach is the fastest way to solve this kind of problems, attention must be paid to interpreting results. In effect in sec. 1.3.1 we had seen that the method implemented is strongly grid dependent, at least in the analysis of some particular physical quantities, like Mass of configuration (see Fig. 1.4).

However, we also concluded that the code works appropriately and thus we could treat a more complex situation in Chap 2, in which differential rotation of polytropic stars in a classical framework is taken into account. After showing that our code is able to reproduce known results, such the ones in [24], we also had shown how to modify the code which can be written starting from ref. [24] inserting a multi-parametric differential rotation law, in which angular velocity is given as a function of the distance from the axis of rotation, taking into account axisymmetric configurations. We presented results obtained with our C code with a two parameters differential rotation law, analysing the parameter space to identify for which values of the two parameters stable configurations are admitted, provided that stability conditions (namely mass-shedding limit and Solberg-Høiland criterion) are satisfied. We will provide to the reader our own C code in the subsequent App. A. This code has been written in a portable way, which allows everyone to change only the

expression for the differential rotation law (and, of course, adding some conditions to select appropriate free parameters) to treat different ones.

From the relativistic point of view, we instead had used a public domain code, namely RNS[1], to obtain numerical models of rapidly rotating realistic NSs in full GR. We had seen in Chapters 3 and 4 how this kind of configurations can be obtained and how a complete physical study of this results can be performed.

More precisely, in Chap. 3 models of uniformly rotating NSs for selected relativistic mean-field nuclear matter model EOS (TM1, GM1, and NL3) were shown. Specifically, we have calculated their gravitational mass, equatorial and polar radii, eccentricity, angular momentum, moment of inertia and quadrupole moment. We have established the region of stability against mass-shedding and the secular axisymmetric instability. We have provided plots of all these physical quantities e.g. as a function of the mass of the configurations. We have also constructed sequences of constant rotation frequency and determined approximately the rotation rate at which deviations of the structure parameters from the spherically symmetric (or slowly rotating) values start, obtaining $f \approx 200$ Hz, which is a value in agreement with previous works (see, e.g., Ref. [9]). From the astrophysical point of view, we have obtained a lower bound for the mass of the fastest observed pulsar, PSR J1748–2446ad with $f = 716$ Hz, by constructing its constant rotation frequency sequence and constraining it to be within the stability region: we obtained $M_{\rm min} = [1.2–1.4]\ M_\odot$, for the EOS we have used in this work, a prediction submitted for observational verification. We have obtained also a fitting formula relating the baryonic and gravitational mass of both non-rotating NSs and rotating ones, given respectively in Equations (3.19) and (3.20), independent on the EOS. We have computed a formula for the NSs' masses on the secular instability line as a function of their angular momentum, see Eq. (3.21). We studied the Kerr parameter (dimensionless angular momentum) of NSs and found that it reaches a maximum value $(a/M)_{\rm max} \approx 0.7$, independent on the EOS. This result brings us to the important conclusion that the gravitational collapse of a uniformly rotating NS, constrained to mass-energy and angular momentum conservation, cannot lead to a maximally rotating Kerr black hole, which by definition has $(a/M)_{\rm BH,max} = 1$. We have also shown that the quadrupole moment of realistic NSs does not reach the Kerr value (for the same values of mass and angular momentum), but this is closely approached from above at the maximum mass value, in physical agreement with the no-hair theorem. We have also found that the stiffer the EOS the closer the Kerr solution is approached. It is important to stress that the results which are shown in this work for some specific nuclear EOS likely will remain valid in the case of other different models, providing they are consistent with current observational constraints, especially the mass of PSR J0348+0432, $M = 2.01 \pm 0.04 M_\odot$ [1]. The existence of such a massive NS clearly favours stiff nuclear EOS as the ones obtained via RMF theory, which leads to a critical NS mass higher than this constraint.

Finally, in Chap. 4 we had analysed the problem of particles orbiting in the equatorial plane of a rotating NS. In particular results regarding the mostly bound orbits of rotating NSs were reported. We have computed binding energy and angular momentum of particles orbiting in the innermost stable circular orbit of a rotating

[1] http://www.gravity.phys.uwm.edu/rns/

NS, supposing that this orbit lies outside the surface. We have provided numerical universal fitting formulas for binding energy and angular momentum of particles orbiting in such orbits which depend on mass and dimensionless angular momentum of configuration (strictly within an error of 1%) and we also have provided a formula to relate the minimum mass of a rotating NS for such orbit to be external with the same mass in the non-rotating case and its dimensionless angular momentum.

To conclude the present work, now having a more or less complete view of the problem of equilibrium of rotating self-gravitating stars, we can understand how this field of astrophysics is abundant in physically relevant aspects. The application of numerical methods allows us to reach some results which, although subject to some kind of errors, cannot actually be obtained in other ways.

Perspectives

As final remarks, we would like to provide some examples of future perspectives which this kind of topics may offer, which make it easy to understand that considering this field of study as exhausted is very far from the truth.

Firstly, from a Newtonian point of view, it is worth noting that the code supplied in App. A in principle requires some adjustments to be considered as a general one. At first, it would be useful to insert a routine to compute the spherical initial guess internally, without the requirements of supplying it. Then, remember that the Newton-Raphson routine to solve the non-linear system of equations should be modified, in fact, the one provided in [55] which we used for implementation, can reach some local minima instead of the global one, returning an error and stopping in this way the production of models. Because of this behaviour of the numerical routine, it may be interesting to write a new routine to implement the Newton-Raphson method, in order to continue to follow the building of equilibrium configurations further in the increasing of axis ratio.

Another point which is worth to mention, is to insert a more general rotation law. The reason for using such a rotation law given by Eq. (2.12) was to treat the analysis of the two parameters space in term of stability of equilibrium configurations (see the discussion in Sec. 2.6.3), but we decided to write the code in a portable way, in order to allow future implementations of other rotation laws. Thus, one could in principle choose the desired law to implement, write it in the code and let the numerical routines do the job. For example, one interesting case could be the one in which the angular velocity in not a strictly decreasing function of the cylindrical radius as in the case of the coalescence of the two component of a binary system (for example of white dwarfs), during merging.

Finally, the last perspective of this work is the one of inserting the possibility to consider different kinds of EOS. In the present case, all previously presented results were obtained with a $n = 1.5$ polytropic EOS, but in principle, the method of Chapters 1 and 2 can be used for every kind of EOS, included the case of a numerical EOS, in which no analytical relation is known.

From a relativistic point of view, as already mentioned in Sec. 3.2.3, let's remind the importance of considering a global charge neutrality condition for the system of self-gravitating NS, instead of a local one, which needs a new and more complete

code to treat this kind of problems, including the case of fast rotating strange quark stars with crust, which show similar features in the core-crust transition.

In conclusion, all the fits provided in Chapters 3 and 4 are very useful and powerful instruments with which one can approach several astrophysical applications. An example is provided by [6], in which the accretion process from the Carbon-Oxygen (CO) core to the NS companion in a tight binary is modelled, taking into account rapid rotation with consequential deviations from spherical symmetry.

Bibliography produced by the Ph.D. candidate

Published

1. **Cipolletta, F.**; Cherubini, C. ; Filippi, S.; Rueda, J. A.; Ruffini, R., "Fast rotating neutron stars with realistic nuclear matter equation of state", 2015, Phys. Rev. D, 023007, 92;

2. Becerra, L.; **Cipolletta, F.**; Fryer, Chris L.; Rueda, Jorge A.; Ruffini, Remo, "Angular Momentum Role in the Hypercritical Accretion of Binary-driven Hypernovae", 2015, ApJ, 100, 812;

In Production

1. **Cipolletta, F.**; Cherubini, C. ; Filippi, S.; Rueda, J. A.; Ruffini, R., "Fast Rotating Neutron Stars with Realistic Nuclear Matter Equation of State", Proceeding of the second Cesar-Lattes metting 2014;

2. Becerra, L.; **Cipolletta, F.**; Fryer, Chris L.; Rueda, Jorge A.; Ruffini, Remo, "Angular Momentum Transfer During the Hypercritical Accretion in Binary Driven Hypernovae", Proceeding of the second Cesar-Lattes metting 2014;

3. **Cipolletta, F.**; Cherubini, C. ; Filippi, S.; Rueda, J. A.; Ruffini, R., "Structure And Stability For Realistic Rapidly Rotating NS: Full GR Treatment", Proceeding of XIV Marcel Grossman metting 2015;

In Preparation

1. **Cipolletta, F.**; Rueda, J. A.; Ruffini, R., "On the mostly bound circular orbit around rapidly rotating neutron stars";

2. Becerra, L. M.; Rodrigues, R. C.; **Cipolletta, F.**; Rueda, J. A.; Ruffini, R., "On the accuracy of the Hartle-Thorne approximation in realistic rapidly rotating neutron stars";

3. **Cipolletta, F:**; Filippi, S.; Cherubini, C., "On the equilibrium configuration for Newtonian polytropes with a two pa- rameters differential rotation law";

Appendix A

"CLASS_ROT" code

A.1 How the code works

The `CLASS_ROT` code is written in C language. It has a code file, called "`CLASS_ROT.c`", which must be compiled together with two other files, namely `nrutil.c` and `nrutil.h` which can be found with the ref. [55]. These last two files contain some declarations and functions used by routines in the main `.c` file to solve the system of equations through a Newton-Raphson approach, as described in Chap.s 1 and 2.

After the compilation is done, an initial guess solution describing the spherical model must be supplied to the code, through a terminal command flag of the form `-f filename.txt`. This `.txt` file should contain only one row vector, which element should be as described in the comments just before the routine `load_sphere()`, which will be reported in the following.

It is worth noting that the free parameters used in the rotation law reported in Sec. 2.6 are used with different names in this code. Exponential parameter b is in the code called with letter a, while the other parameter a is called m.

A.2 C code

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <stdbool.h>
#include "nrutil.h"

char version[30] = "1.0, September 2015";


#define PI 3.1415926535                 /* what else */
#define SQ(x) ((x)*(x))                 /* square macro */
#define enne 1.5                        /* polytropic index */
```

```
#define step 1.05                    /* step in axis ratio
                                        to build the
                                        sequence of models */

#define MODNUMMAX 100
#define MAXITS 200
#define NT 15                        /* grid angular
                                        dimension */
#define NR 40                        /* grid radial
                                        dimension */
#define N_CENTRE 10000               /* grid number for
                                        cylindrical radius
                                        to compute Phi_c
                                        at the centre */
#define inf_cil_rad 100.0            /* infinity in
                                        cylindrical radius */
#define NCIL 1000                    /* grid cylindrical
                                        radius dimension */
#define Na 10                         /* number of values
                                         of parameter a to
                                         compute for dif._
                                         ferential
                                         rotation */
#define Nm 10                         /* number of values
                                         of parameter emme to
                                         compute for dif._
                                         ferential
                                         rotation */
#define aa_max 1.15                   /* maximum value of
                                         parameter a */
#define aa_min 0.0                    /* minimum value of
                                         parameter a */
#define emmeemme_max 2.0              /* maximum value of
                                         parameter m */
#define emmeemme_min 0.02             /* minimum value of
                                         parameter m */
#define DNMAX 11                     /* number of terms in
                                        Legendre polynomial
                                        expansions */
#define TINY 1.0e-20                 /* A small number */
#define EPS 1.0e-4
#define TOLX 1.0e-4
#define TOLF 1.0e-7
#define ALF 1.0e-4                   /* Ensures sufficient
                                        decrease in function
                                        value. */
```

```
#define STPMX 100.0
#define TOLMIN 1.0e-6
#define TOLDENS 0.1                    /* Tolerance to define
                                          if a configuration is
                                          a ring-like one */

#define FREERETURN {free_dvector(fvec,1,n);
                    free_dvector(xold,1,n);
                    free_dvector(p,1,n);
                    free_dvector(g,1,n);
                    free_dmatrix(fjac,1,n,1,n);
                    free_ivector(indx,1,n);
                    return;}

/* Here MAXITS is the maximum number of iterations;
   TOLF sets the convergence criterion on function values;
   TOLMIN sets the criterion for deciding whether spurious
   convergence to a minimum of FNORM has occurred;
   TOLX is the convergence criterion on  x;
   STPMX is the scaled maximum step length allowed in line
   searches. */

void load_sphere( char sphere[]);
void make_grid(double y[]);
void usrfun(double x[],int n,double *fvec,double **fjac);
void newt(double x[], int n, int *check
          ,void (*vecfunc)(int, double [], double []));
void vecfunc(int n,double x[],double f[]);
void print_header(int n_a, int n_m);
void (*nrfuncv)(int n, double v[], double f[]);
double FNORM(double x[]);
double plgndr(int l, int m, double x);
void CENTR_POT_CENTRE(double x[]);
double DIFF_ROT_LAW(double xi_cil,double x[]);

char file_name[80] = "no_sphere_file_specified",
     eos_type[30] = "tab";

int ITERS,
    CKC=0,
    nn,
    a_check=0,        /* indicates if iteration diverged */
    print_option,     /* select print out */
    num_it,           /* counter of number of iterations in
                         N-R method */
```

```
    modnum=0,        /* model number */
    cntr = 0;

    /* control parameter
       for the model: - 0 by default
                      - 1 problems by Nwt-Raph routine
                      - 2 no Hoiland crituerium
                      - 3 eq eff grav >0 (mass-shedding)
                      - 4 no virial theorem
                      - 5 sequence exceeds MODNUMMAX models
                      - 6 OK
                      - 7 ring-like configuration
                      - 10 = 7+3 ring-like + mass-shedding
                      - 11 = 7+4 ring-like + ratio of kinetic
                             and potential energy higher than
                             one half
                      - 12 = 7+5 ring-like + sequence exceeds
                             MODNUMMAX models */


double obl;              /* initial value for axis ratio
                            defined as r_eq/r_pol */
double *SPH_val;         /* spherical model */
double *init;            /* variables for the guess model */
double *theta;           /* grid points in angular direction */
double *THETA;           /* coefficient for integration in
                            angular direction */
double **ERRE;           /* coefficient for integration in
                            radial direction */
double *LEG_POL_2c_i;    /* vector of Legendre Polynomials used
                            for gravitational potential */
double *fgr;             /* vector of Green's functions coef._
                            ficients */
double Fsys_x[NT*(NR+1)+3];   /* vector which contains value
                                 of system functions computed
                                 in a certain point x */
double a;                /* first parameter for rotation law in
                            system = aa*(1/r_s^2), where r_s is
                            the radius of sphere */
double *aa;              /* dimensionless first parameter for
                            differential rotation */
double emme;             /* second parameter for rotation law in
                            system = emmeemme*(1/r_s^2), where r_s
                            is the radius of sphere */
double *emmeemme;        /* dimensionless second parameter for
```

```
                                differential rotation */
double *fvec;
double CONST_INF;       /* minus dimensionless centrifugal pot._
                           ential at infinity */
double **OUT_FILE;      /* matrix to store values of parameters
                           and cntr for each sequence */

/*******************************/
/* Routine for LU decomposition */
/* from Numerical Recipies in C */
/*******************************/

void ludcmp(double **a, int n, int *indx, double *d)

{

int i,
    imax,
    j,
    k;

double big,
       dum,
       sum,
       temp;

double *vv;

vv = dvector(1,n);
*d=1.0;

for (i=1;i<=n;i++) {
    big=0.0;

    for (j=1;j<=n;j++) {
        if ((temp=fabs(a[i][j])) > big) big=temp;
      }

    if (big == 0.0) nrerror("Singular_matrix_in_routine
ludcmp");

    vv[i]=1.0/big;

}
```

```
for (j=1;j<=n;j++) {

 for (i=1;i<j;i++) {
       sum=a[i][j];

       for (k=1;k<i;k++) sum -= a[i][k]*a[k][j];

       a[i][j]=sum;
  }

 big=0.0;

 for (i=j;i<=n;i++) {
    sum=a[i][j];

  for (k=1;k<j;k++) sum -= a[i][k]*a[k][j];

  a[i][j]=sum;

  if ( (dum=vv[i]*fabs(sum)) >= big) {

    big=dum;
    imax=i;
  }

 }

 if (j != imax) {

  for (k=1;k<=n;k++) {
      dum=a[imax][k];
      a[imax][k]=a[j][k];
    a[j][k]=dum;
  }

  *d = -(*d);
  vv[imax]=vv[j];
 }

 indx[j]=imax;

 if (a[j][j] == 0.0) a[j][j]=TINY;

 if (j != n) {
```

```
    dum=1.0/(a[j][j]);

    for (i=j+1;i<=n;i++) a[i][j] *= dum;
   }

}

free_dvector(vv,1,n);

return;

}

/*********************************/
/* Routine to solve linear system */
/* from Numerical Recipies in C   */
/*********************************/

void lubksb(double **a, int n, int *indx, double b[])

{

 int i,
     ii=0,
     ip,
     j;

 double sum;

 for (i=1;i<=n;i++) {
    ip=indx[i];
    sum=b[ip];
    b[ip]=b[i];

    if (ii)
      for (j=ii;j<=i-1;j++) sum -= a[i][j]*b[j];
    else if (sum) ii=i;
     b[i]=sum;
  }

 for (i=n;i>=1;i--) {
      sum=b[i];

     for (j=i+1;j<=n;j++) sum -= a[i][j]*b[j];
     b[i]=sum/a[i][i];
```

```
  }

 return;
}

/*************************************************/
/* Routine for linear search used in Newton-Raphson */
/* method from Numerical Recipies in C           */
/*************************************************/

void lnsrch(int n, double xold[], double fold, double g[],
            double p[], double x[], double *f, double stpmax,
            int *check, double (*func)(double []))

{

int i;

double a,
       alam,
       alam2,
       alamin,
       b,
       disc,
       f2,
       rhs1,
       rhs2,
       slope,
       sum,
       temp,
       test,
       tmplam;

*check=0;

for (sum=0.0,i=1;i<=n;i++) sum += p[i]*p[i];

sum=sqrt(sum);

if (sum > stpmax)
  for (i=1;i<=n;i++) p[i] *= stpmax/sum;

for (slope=0.0,i=1;i<=n;i++)
    slope += g[i]*p[i];
```

```
if (slope >= 0.0) nrerror("Roundoff problem in lnsrch.");

test=0.0;

for (i=1;i<=n;i++) {
  temp=fabs(p[i])/DMAX(fabs(xold[i]),1.0);
  if (temp > test) test=temp;
}
alamin=TOLX/test;
alam=1.0;

for(;;) {
    for (i=1;i<=n;i++) x[i]=xold[i]+alam*p[i];
    *f=(*func)(x);
    if (alam < alamin) {
    for (i=1;i<=n;i++) x[i]=xold[i];
    *check=1;
    return;
    }
    else if (*f <= fold+ALF*alam*slope) return;
    else {
        if (alam == 1.0)
            tmplam = -slope/(2.0*(*f-fold-slope));
        else {
            rhs1 = *f-fold-alam*slope;
            rhs2=f2-fold-alam2*slope;
            a=(rhs1/(alam*alam)
              -rhs2/(alam2*alam2))/(alam-alam2);
            b=(-alam2*rhs1/(alam*alam)
              +alam*rhs2/(alam2*alam2))/(alam-alam2);
            if (a == 0.0) tmplam = -slope/(2.0*b);
            else {
                disc=b*b-3.0*a*slope;
                if (disc < 0.0) tmplam=0.5*alam;
                else if (b <= 0.0) {
                  tmplam=(-b+sqrt(disc))/(3.0*a);
                }
                else tmplam=-slope/(b+sqrt(disc));
                 }
            if (tmplam > 0.5*alam)
                tmplam=0.5*alam;
              }
         }
    alam2=alam;
    f2 = *f;
```

```
    alam=DMAX(tmplam,0.1*alam);

  }

  return;
}

/*********************************************/
/* Globally convergent Newton-Raphson routine */
/* from Numerical Recipies in C               */
/*********************************************/

void newt(double x[], int n, int *check,
          void (*vecfunc)(int, double [],
          double []))

{

void fdjac(int n, double x[], double fvec[], double **df,
     void (*vecfunc)(int, double [], double []));
double FNORM(double x[]);
void lnsrch(int n, double xold[], double fold, double g[],
            double p[], double x[],double *f, double stpmax,
            int *check, double (*func)(double []));
void lubksb(double **a, int n, int *indx, double b[]);
void ludcmp(double **a, int n, int *indx, double *d);

int i,
    its,
    j,
    *indx;

double d,
       den,
       f,
       fold,
       stpmax,
       sum,
       temp,
       test,
       **fjac,
       *g,
       *p,
       *xold;
```

```
indx=ivector(1,n);
fjac=dmatrix(1,n,1,n);
g=dvector(1,n);
p=dvector(1,n);
xold=dvector(1,n);
fvec=dvector(1,n);
nn=n;
nrfuncv=vecfunc;
f=FNORM(x);
test=0.0;

for (i=1;i<=n;i++)
  if (fabs(fvec[i]) > test) test=fabs(fvec[i]);

if (test < 0.01*TOLF) {
  *check=0;
  FREERETURN
}

for (sum=0.0,i=1;i<=n;i++) sum += SQ(x[i]);

stpmax=STPMX*DMAX(sqrt(sum),(double)n);

for (its=1;its<=MAXITS;its++) {

   ITERS = its;

   fdjac(n,x,fvec,fjac,vecfunc);

   for (i=1;i<=n;i++) {
       for (sum=0.0,j=1;j<=n;j++) sum += fjac[j][i]*fvec[j];
       g[i]=sum;
   }

   for (i=1;i<=n;i++) xold[i]=x[i];
   fold=f;
   for (i=1;i<=n;i++) p[i] = -fvec[i];

   ludcmp(fjac,n,indx,&d);
   lubksb(fjac,n,indx,p);
   lnsrch(n,xold,fold,g,p,x,&f,stpmax,check,FNORM);

   test=0.0;

   for (i=1;i<=n;i++)
```

```
        if (fabs(fvec[i]) > test) test=fabs(fvec[i]);

    if (test < TOLF) {
    *check=0;
    FREERETURN
    }

    if (*check) {
        test=0.0;
        den=DMAX(f,0.5*n);
        for (i=1;i<=n;i++) {
            temp=fabs(g[i])*DMAX(fabs(x[i]),1.0)/den;
            if (temp > test) test=temp;
          }
        *check=(test < TOLMIN ? 1 : 0);
        FREERETURN
      }

    test=0.0;
    for (i=1;i<=n;i++) {
        temp=(fabs(x[i]-xold[i]))/DMAX(fabs(x[i]),1.0);
        if (temp > test) test=temp;
      }

    if (test < TOLX) FREERETURN
  }
    nrerror("MAXITS exceeded in newt");

}

/*************************************************/
/* Routine to compute Jacobian of a vector function */
/* from Numerical Recipies in C                  */
/*************************************************/

void fdjac(int n, double x[], double fvec[], double **df,
           void (*vecfunc)(int, double [], double []))

 {

 int i,
     j;

 double h,
        temp,
```

```
        *f;

  f = dvector(1,n);

  for (j=1;j<=n;j++) {

      temp=x[j];
      h=EPS*fabs(temp);

      if (h == 0.0) h=EPS;

      x[j]=temp+h;
      h=x[j]-temp;

      (*vecfunc)(n,x,f);

      x[j]=temp;
      for (i=1;i<=n;i++) df[i][j]=(f[i]-fvec[i])/h;

   }

   free_dvector(f,1,n);
   return;
}

/*************************************/
/* Routine for norm of a vector function */
/* from Numerical Recipies in C          */
/*************************************/

double FNORM(double x[])

{
   int i;
   double sum;

   (*nrfuncv)(nn,x,fvec);
   for (sum=0.0,i=1;i<=nn;i++) sum += SQ(fvec[i]);

   return 0.5*sum;
}

/*****************************************************/
/* Create computational mesh for weight for integration. */
/* Points in the angular direction are stored in the     */
```

```
/* array theta[i]. Points in the r-direction are stored  */
/* in the array s_pg[j].                                 */
/*******************************************************/
void make_grid(double y[])
{
  int i,           /* counter theta */
      j;           /* counter r */

 for(i=1;i<=NT;i++) {
      if(i==1){
       THETA[i] = (PI/(4.0*(NT-1.0)));
       }
      else if(i==NT) {
       THETA[i] = (PI/(4.0*(NT-1.0)));
      }
      else {
       THETA[i] = (PI/(2.0*(NT-1.0)));
      }
 }

 /* Grid points in r directions are x[i]*j/NR, for i=1..NT and
    j=1..NR, x[i] = r_surf[i] */

 for(i=1;i<=NT;i++) {
  for(j=1;j<=NR;j++) {
    if(j==1) {
      ERRE[i][j] = (y[i])/(3.0*NR);
    }
    else if(j==(NR)) {
      ERRE[i][j] = (y[i])/(3.0*NR);
    }
    else if((j % 2)==0) {
      ERRE[i][j] = (4.0*y[i])/(3.0*NR);
    }
    else {
      ERRE[i][j] = (2.0*y[i])/(3.0*NR);
    }
  }
 }

 return;
}

/*******************************************************/
/* Routine to Legendre Polynomials of given theta.     */
```

```
/* The first "plgndr" is taken from Numerical Recipies  */
/******************************************************/

double plgndr(int l, int m, double x)
/* Computes the associated Legendre polynomial P^l_m(x). */
{

void nrerror(char error_text[]);

double fact,
       pll,
       pmm,
       pmmp1,
       somx2;

int i,
    ll;

if (m < 0 || m > l || fabs(x) > 1.0) nrerror("Bad arguments
in routine plgndr.\n");
pmm=1.0;

if (m > 0) {
   somx2=sqrt((1.0-x)*(1.0+x));
   fact=1.0;
   for (i=1;i<=m;i++) {
    pmm *= -fact*somx2;
    fact += 2.0;
    }
}

if (l == m)
    return pmm;

else {
    pmmp1=x*(2*m+1)*pmm;

    if (l == (m+1))
       return pmmp1;

    else {
       for (ll=m+2;ll<=l;ll++) {
            pll=(x*(2*ll-1)*pmmp1-(ll+m-1)*pmm)/(ll-m);
           pmm=pmmp1;
           pmmp1=pll;
```

```
            }
        return pll;
      }
     }

}


void leg_polyn(){

/* Order is LEG_POL_2c_i[i + c*NT] = P(2*c, cos(theta[i])) */

      double x;

      int c,
          i;

      for (i=1;i<=NT;i++){
       x=cos(theta[i]);
       for (c=0;c<DNMAX;c++){
         LEG_POL_2c_i [ i + c*NT ] = plgndr((2*c),0,x);
       }

      }
      return;
}

/*************************************************/
/* Load SPHERE file. Format:  NT*(NR+1)+10 vector  */
/*************************************************/

void load_sphere(char sphere[])
{
 int i,                          /* counter */
     n_var;                      /* number of variables */

 double y;       /* spherical values ordered as listed below */

 FILE *f_sphere;                 /* pointer to sphere */


    /* OPEN FILE TO READ */

    if((f_sphere=fopen(sphere,"r")) == NULL ) {
       printf("cannot open file:  %s\n",sphere);
```

```
        exit(0);
    }


    /* COMPUTE NUMBER OF VARIABLES AND PHYSICAL QUANTITIES */

    n_var = NT*(NR+1)+11;


    /* READ SPHERE INPUT */

    for(i=1;i<n_var;i++) {

        fscanf(f_sphere,"%lf ",&y) ;

/*  the order is
    the following: - NT surface radii
                   - NT*NR values of dimensionless density
                     following radii gridpoints from center
                     to surface, from pole to equator
                   - 1 dimensionless central angular velocity
                   - 1 dimensionless gravitational potential
                     at the pole
                   ------------------------------- NT*(NR+1)+2
                   - dimensionless kinetic energy
                   - dimensionless gravitational potential
                     energy
                   - dimensionless thermal energy
                   - dimensionless total mass
                   - dimensionless total angular momentum
                   - dimensionless surface effective gravity
                     at the equator
                   - cntr value to check if the sequence
                     should be stopped
                   - VT = | ( 2*T + W + (3/n)*U ) / W |
                   ------------------------------- NT*(NR+1)+10 */
        SPH_val[i] = y;
       }

    return;
}

/*******************************************************/
/* Routine for rotation law in cylindrical coordinates */
/*******************************************************/
```

```
/* NB: xi_cil = r * sin(theta), thus OMEGA(xi_cil) =
       OMEGA(r,theta) if one considers spherical coordinates

   NB2: note that parameters a and emme should have dimensions
        of 1/r^2!

   The routine returns the value of angular velocity at a
   certain distance from the axis */

double DIFF_ROT_LAW(double xi_cil,double x[])
{
  double OM;

  OM = sqrt(x[NT*(NR+1)+1])*((exp(-a*SQ(xi_cil)))
       /(1.0+(SQ(xi_cil/emme))));

  return OM;
}

/*****************************************/
/* Routine to compute Green's Functions  */
/*****************************************/

void FGREEN(double x[]){

  int i,
      j,
      p,
      q,
      c,
      v;

  for (i=1;i<=NT;i++){
    for (j=1;j<=NR;j++){
      for (p=1;p<=NT;p++){
        for (q=1;q<=NR;q++){
          for (c=0;c<DNMAX;c++){
            v = i + (j-1)*NT + (p-1)*NT*NR + (q-1)*NT*NT*NR
                + c*NR*NR*NT*NT;
            if ((x[p]*q/NR) > (x[i]*j/NR)) {
              fgr[ v ] = pow((x[i]*j/NR),(2.0*c))
               /pow((x[p]*q/NR),(2.0*c+1.0));
             }
            else {
```

```
                fgr[ v ] = pow((x[p]*q/NR),(2.0*c))
                                /pow((x[i]*j/NR),(2.0*c+1.0));
              }
            }
          }
        }
      }
    }

    return;
}

/*****************************************/
/* Routine to compute Centrifugal Potential */
/* at the centre of configuration          */
/*****************************************/

void CENTR_POT_CENTRE(double x[]){

    /* This routine compute the centrifugal potential at
       infinity by definition, called CONST_INF, then the
       centrifugal potential at the centre is posed as
       (- CONST_INF) and every centrifugal potential
       in each gridpoint is subtracted by CONST_INF */

    int i,
        k;

    double *r_cil,
           *weight_r_cil,
           temp_k;

    r_cil = dvector(1,N_CENTRE);
    weight_r_cil = dvector(1,N_CENTRE);

    for (i=1;i<=N_CENTRE;i++){

      r_cil[i] = (inf_cil_rad/(N_CENTRE-1.0))*(i-1.0);
                /* Definition of cylindrical grid for
                   integration */
      if (i==1) {
                /* Definition of weights for integration
                   considering trapezoidal method */
        weight_r_cil[i] = (inf_cil_rad/(2.0*(N_CENTRE-1.0)));
      }
```

```
    else if (i==N_CENTRE) {
      weight_r_cil[i] = (inf_cil_rad/(2.0*(N_CENTRE-1.0)));
    }
    else {
      weight_r_cil[i] = (inf_cil_rad/((N_CENTRE-1.0)));
    }

  }

  temp_k = 0.0;

  for (k=1;k<=N_CENTRE;k++){
    temp_k += SQ(DIFF_ROT_LAW(r_cil[k],x)) * r_cil[k]
              * weight_r_cil[k];
  }

  CONST_INF = temp_k;
     /* integration in radial coordinate up to infinity
        has been performed and the constant of integration
        which correspond to centrifugal potential at the
        centre has been defined (there is no minus sign
        because the constant of integration is the opposite
        of value of Phi_c at infinity */

  free_dvector(r_cil,1,N_CENTRE);
  free_dvector(weight_r_cil,1,N_CENTRE);

  return;
}

/*********************************/
/* Routine to compute Centrifugal */
/* Potential in fixed gridpoint   */
/*********************************/

double CENTR_POT_GRID(double r,double t,double x[]){

  int i,
      k;

  double *r_cil,
         *weight_r_cil,
         CP_val,
         temp_k;
```

```
  r_cil = dvector(1,NCIL);
  weight_r_cil = dvector(1,NCIL);

  for (i=1;i<=NCIL;i++){

    r_cil[i] = ((r*sin(t))/(NCIL-1.0))*(i-1.0);
        /* Definition of cylindrical grid for integration */

    if (i==1) {
        /* Definition of weights for integration
           considering trapezoidal method */
      weight_r_cil[i] = ((r*sin(t))/(2.0*(NCIL-1.0)));
    }
    else if (i==NCIL) {
      weight_r_cil[i] = ((r*sin(t))/(2.0*(NCIL-1.0)));
    }
    else {
      weight_r_cil[i] = ((r*sin(t))/((NCIL-1.0)));
    }

  }

  temp_k = 0.0;

  for (k=1;k<=NCIL;k++){
    temp_k += SQ(DIFF_ROT_LAW(r_cil[k],x)) * r_cil[k]
              * weight_r_cil[k];
  }

  CP_val = -temp_k + CONST_INF;

  free_dvector(r_cil,1,N_CENTRE);
  free_dvector(weight_r_cil,1,N_CENTRE);

  return CP_val;
}

/*************************************************/
/* Routine that returns the values of functions */
/* in the system of equations f[1,..,n],        */
/* calculated at a certain point x[1,..,n]      */
/*************************************************/

void vecfunc(int n,double x[],double f[]){
```

```
void FGREEN(double x[]);
void CENTR_POT_CENTRE(double x[]);
double CENTR_POT_GRID(double r,double t,double x[]);

int i,
    j,
    v,
    p,
    q,
    c;

double GPC,
       GPG[NT+1][NR+1],
       CPC,
       CPG[NT+1][NR+1];

double sum_pot_r = 0.0,      /* intermediate sum in r
                                direction */
       sum_pot_th = 0.0,     /* intermediate sum in theta
                                direction */
       sum_pot_c = 0.0;      /* intermediate sum in n */

double temp=0.0,
       temp_in=0.0;

FGREEN(x);
CENTR_POT_CENTRE(x);

  for (p=1;p<=NT;p++){
    temp_in = 0.0;
    for (q=1;q<=NR;q++){
      temp_in = temp_in + ( ((x[p]*q)/NR)* ERRE[p][q] *
                            pow(x[NT+(p-1)*NR+q],enne) );
    }
    temp = temp + (sin(theta[p]) * THETA[p] * temp_in);
  }

  GPC = - temp;

  for (i=1;i<=NT;i++)
  {
    for (j=1;j<=NR;j++)
    {
      sum_pot_th = 0.0;
      for(p=1;p<=NT;p++)
```

```
      {
        sum_pot_r = 0.0;
        for (q=1;q<=NR;q++)
        {
          sum_pot_c = 0.0;
          for (c=0;c<DNMAX;c++)
          {
              sum_pot_c +=  fgr[ i + (j-1)*NT + (p-1)*NT*NR
                                 + (q-1)*NT*NT*NR
                                 + c*NR*NR*NT*NT ]
                            * LEG_POL_2c_i[ i + c*NT ]
                            * LEG_POL_2c_i[ p + c*NT ]
                            * pow(x[ NT+((p-1)*NR)+q ],enne);
          }
          sum_pot_r += SQ((x[p]*q/NR))*ERRE[p][q]*sum_pot_c;
        }
        sum_pot_th += sin(theta[p])*THETA[p]*sum_pot_r;
      }
      GPG[i][j] = - sum_pot_th;
    }
  }

CPC = CONST_INF;

for (i=1;i<=NT;i++) {
  for (j=1;j<=NR;j++){
    CPG[i][j] = CENTR_POT_GRID((x[i]*j/NR),theta[i],x);
  }
}

f[1] = 1.0 + GPC + CPC - x[NT*(NR+1)+2];

v = 1;

for (i=1;i<=NT;i++){
  for(j=1;j<=NR;j++){
    v +=1;
    f[v] = x[NT+NR*(i-1)+j] + GPG[i][j]
         + CPG[i][j] - x[NT*(NR+1)+2];
  }
}

for (i=1;i<=NT;i++) f[1+NT*NR+i] = x[NT+(i-1)*NR+NR];

if(n==(NT*(NR+1)+2)) f[n] = x[NT]-(obl*x[1]);
```

```
else printf("ERROR in SYSTEM DIMENSION's definition\n");

return;
}

/*************************************************/
/* Routine to compute physical quantities stored in */
/* x[NT*(NR+1)+3,..,NT*(NR+1)+10]                   */
/*************************************************/

/* x[NT*(NR+1)+3] - dimensionless kinetic energy
   x[NT*(NR+1)+4] - dimensionless gravitational potential
                    energy
   x[NT*(NR+1)+5] - dimensionless thermal energy, defined as
                    n/(n+1) * int(sin(theta)*dtheta
                    *int(sigma^(n+1)*xi^2
                    ,xi=0..xi_surf(theta)),theta=0..PI/2),
                    here being xi the dimensionless spherical
                    radius
   x[NT*(NR+1)+6] - dimensionless total mass
   x[NT*(NR+1)+7] - dimensionless total angular momentum
   x[NT*(NR+1)+8] - dimensionless surface effective gravity
                    at the equator
   x[NT*(NR+1)+9] - cntr value for the model
   x[NT*(NR+1)+10]- V.T. */

void PHYS_QUANT(double x[], int modn){

  int i,
      j,
      p,
      q;

  double temp_p = 0.0,
         temp_q = 0.0,
         CPG_val[NT+1][NR+1];

  void CONTROL_MOD(double x[],int modn);
  double CENTR_POT_GRID(double r,double t,double x[]);

  for (i=1;i<=NT;i++) {
  for (j=1;j<=NR;j++){
    CPG_val[i][j] = CENTR_POT_GRID((x[i]*j/NR),theta[i],x);
    }
  }
```

```
for(p=1;p<=NT;p++){
  temp_q = 0.0;
  for(q=1;q<=NR;q++){
    temp_q += pow((x[p]*q/NR),4.0) * ERRE[p][q]
              * pow((x[NT+(p-1)*NR+q]),enne)
              * SQ(DIFF_ROT_LAW((x[p]*q/NR
                                 *sin(theta[p])),x));
  }
  temp_p += pow(sin(theta[p]),3.0)* THETA[p] * temp_q;
}

x[NT*(NR+1)+3] = 0.5 * temp_p;

temp_p = 0.0;
temp_q = 0.0;

for(p=1;p<=NT;p++){
  temp_q = 0.0;
  for(q=1;q<=NR;q++){
    temp_q += SQ((x[p]*q/NR)) * ERRE[p][q]
              * pow((x[NT+(p-1)*NR+q]),enne)
              * ( x[NT*(NR+1)+2] - x[NT+(p-1)*NR+q]
                  - CPG_val[p][q] );
  }
  temp_p += sin(theta[p]) * THETA[p] * temp_q;
}

x[NT*(NR+1)+4] = 0.5 * temp_p;

temp_p = 0.0;
temp_q = 0.0;

for(p=1;p<=NT;p++){
  temp_q = 0.0;
  for(q=1;q<=NR;q++){
    temp_q += SQ((x[p]*q/NR)) * ERRE[p][q]
              * pow((x[NT+(p-1)*NR+q]),enne+1.0);
  }
  temp_p += sin(theta[p]) * THETA[p] * temp_q;
}

x[NT*(NR+1)+5] = enne/(enne+1.0) * temp_p;

temp_p = 0.0;
```

```
temp_q = 0.0;

for(p=1;p<=NT;p++){
  temp_q = 0.0;
  for(q=1;q<=NR;q++){
    temp_q += SQ((x[p]*q/NR)) * ERRE[p][q]
              * pow((x[NT+(p-1)*NR+q]),enne);
  }
  temp_p += sin(theta[p]) * THETA[p] * temp_q;
}

x[NT*(NR+1)+6] = temp_p;

temp_p = 0.0;
temp_q = 0.0;

for(p=1;p<=NT;p++){
  temp_q = 0.0;
  for(q=1;q<=NR;q++){
    temp_q += pow((x[p]*q/NR),4.0) * ERRE[p][q]
              * pow((x[NT+(p-1)*NR+q]),enne)
              * DIFF_ROT_LAW((x[p]*q/NR*sin(theta[p])),x);
  }
  temp_p += pow(sin(theta[p]),3.0)* THETA[p] * temp_q;
}

x[NT*(NR+1)+7] = temp_p;

temp_p = 0.0;
temp_q = 0.0;

/* Using a three-points approximation for sigma(xi,theta)
   it turns out that ((d sigma)/(d xi))(xi_surf,Pi/2) =
   (1/2)*(NR/xi_surf)*( 3*sigma(xi_surf,Pi/2) +
  - 4*sigma(xi_surf*(NR-1)/NR,Pi/2) +
   sigma(xi_surf*(NR-2)/NR,Pi/2) ) */

x[NT*(NR+1)+8] = 0.5 * (NR/x[NT])
                 * ( 3.0*x[NT+(NT-1)*NR+NR]
                   - 4.0*x[NT+(NT-1)*NR+NR-1]
                   + x[NT+(NT-1)*NR+NR-2] );

CONTROL_MOD(x,modn);

x[NT*(NR+1)+9] = cntr;
```

```
  x[NT*(NR+1)+10] = fabs(( (2.0*x[NT*(NR+1)+3])
                          + x[NT*(NR+1)+4]
                         + ((3.0/enne)*x[NT*(NR+1)+5]) )
                        /(x[NT*(NR+1)+4]));

  return;
}

/*****************************************/
/* Routine to assign cntr value for model */
/*****************************************/

void CONTROL_MOD(double x[],int modn){

  int i,
      j,
      isnegative,
      k;

  double diff_ang_vel=0.0,
         DIFF_ANG_MOM=0.0,
         MAX_DENS_VAL = 1.0;

  for (k=NT+1;k<=NT*(NR+1);k++){
    if ((x[k] - MAX_DENS_VAL - TOLDENS)>=0.0) {
      MAX_DENS_VAL = x[k];
    }
  }

  isnegative = 0; /* default value */

  for (i=1;i<=NT;i++){
    for (j=1;j<=NR;j++){
      DIFF_ANG_MOM=0.0;
      diff_ang_vel=0.0;
      if(i<=2){
        diff_ang_vel = 0.5 * ((NCIL-1.0)
                              /(x[i]*sin(theta[i])))
        *(-3.0*(DIFF_ROT_LAW((x[i]*sin(theta[i])*(j/NR)),x))
        + 4.0*(DIFF_ROT_LAW(((x[i]*sin(theta[i])*(j/NR))
          +((x[i]*sin(theta[i]))/(NCIL-1.0))),x))
        - (DIFF_ROT_LAW(((x[i]*sin(theta[i])*(j/NR))
          +((2.0*x[i]*sin(theta[i]))/(NCIL-1.0))),x)) );
      }
```

```
      else{
        diff_ang_vel = 0.5 * ((NCIL-1.0)
                              /(x[i]*sin(theta[i])))
       *( 3.0*(DIFF_ROT_LAW((x[i]*sin(theta[i])*(j/NR)),x))
       - 4.0*(DIFF_ROT_LAW(((x[i]*sin(theta[i])*(j/NR))
        -((x[i]*sin(theta[i]))/(NCIL-1.0))),x))
       + (DIFF_ROT_LAW(((x[i]*sin(theta[i])*(j/NR))
        -((2.0*x[i]*sin(theta[i]))/(NCIL-1.0))),x)) );
      }

      /* NB: in diff_ang_vel the three points approx is used
             with forward substituion if i<=2 or backward
             substitution otherwise. Cylindrical grid spacing
             is h = x[i] sin(theta[i])/(NCIL-1.0) */

      DIFF_ANG_MOM = 2.0*DIFF_ROT_LAW((x[i]*sin(theta[i])
                                       *(j/NR)),x)
                     *((2.0*DIFF_ROT_LAW((x[i]
                                          *sin(theta[i])
                                          *(j/NR)),x))
                       + ((x[i]*sin(theta[i])*(j/NR))
                          *diff_ang_vel));

      /* DIFF_ANG_MOM = 2*OMEGA(r_cil)*(2*OMEGA(r_cil) + r_cil
                        * (d[OMEGA(r_cil)]/dr_cil)) */

      /* NB: each derivative is computed in each grid point,
             which has a cylinrical radius r_cil = (x[i]
             *sin(theta[i])*(j/NR)), i = 1..NT, j=1..NR */

      if(DIFF_ANG_MOM<=0.0) isnegative += 1;

      /* if DIFF_ANG_MOM is negative the control is
         incremented */

    }
  }


  if(ITERS>=MAXITS){
    cntr = 1;          /* too much iterations in N-R routine */
    return;
  }

  else if(isnegative > 0) {
```

```
    cntr = 2;          /* violation of Hoiland's criterium for
                          stability */
    return;
  }

  else if((x[NT*(NR+1)+8])>0) {
    if ((MAX_DENS_VAL - 1.0)>0.0) cntr = 10;
                       /* ring-like + mass-shedding */
    else cntr = 3;
                       /* mass-shedding */
    return;
  }

  else if(((x[NT*(NR+1)+3]/fabs(x[NT*(NR+1)+4]))-0.5)>0.0){
    if ((MAX_DENS_VAL - 1.0)>0.0) cntr = 11;
                       /* ring-like + ratio of kinetic and
                          potential energy higher than one
                          half */
    else cntr = 4;
                       /* ratio of kinetic and potential
                          energy higher than one half */
    return;
  }

  else if(modn - MODNUM_MAX > 0) {
    if ((MAX_DENS_VAL - 1.0)>0.0) cntr = 12;
                       /* ring-like + limit of model's number */
    else cntr = 5;
                       /* limit of model's number */
    return;
  }
  else if(modn - MODNUM_MAX <=0) {
    cntr = 6;
                       /* model's number lower than
                          MODNUM_MAX */
    return;
  }
  else if((MAX_DENS_VAL - 1.0)>0.0) {
    cntr = 7;           /* ring_like */
    return;
  }

  return;
}
```

```
/**************************/
/* Routine to print header */
/**************************/

void print_header(int n_a, int n_m){

 printf("\n");
 printf("-------------------------------------------------------------
--------------------------------------\n");
 printf("NTxNR=%dx%d, a=%lf, m=%lf   accuracy in x=%1.0e,
accuracy in f=%1.0e\n"
,NT,NR,aa[n_a],emmeemme[n_m],TOLX,TOLF);

 printf("-------------------------------------------------------------
-------------------------------------\n");
 printf("   NT*r_surf[i]   |     (NT*NR)*rho[i,j]     |    nu_c
|    Phi_g_pol    |    E_kin    |   E_pot   |   U   |   M    |   J
|  eq_effgrav  |  cntr  |  V.T.\n");

 printf("-------------------------------------------------------------
-------------------------------------\n");

 return;
}

/*************/
/* MAIN CYCLE */
/*************/

int main(int argc, char *argv[])
{
  int n_a,
      n_m,
      k,
      i,
      j,
      m,
      l,
      n,
      v,
      c;

  double big,
         temp,
```

```
            MAX_DENS_VAL;

  FILE *TOT;

  theta = dvector(1,NT);
  THETA = dvector(1,NT);
  ERRE = dmatrix(1,NT,1,NR);
  SPH_val = dvector(1,NT*(NR+1)+10);
  init = dvector(1,NT*(NR+1)+10);
  LEG_POL_2c_i = dvector(1,DNMAX*NT);
  fgr = dvector(1,DNMAX*NT*NT*NR*NR);
  OUT_FILE = dmatrix(1,Na*Nm,1,3);

  for(i=1;i<=NT;i++) {
     theta[i] = (PI/2.0)*((i-1.0)/(NT-1.0));
    }          /* Definition of angular values for
                  computational grid */

  /* Definitions of differential rotation parameters
     values */

  aa = dvector(1,Na);
  emmeemme = dvector(1,Nm);

  for (n_a=1;n_a<=Na;n_a++){      /* Equally linearly spaced */
    aa[n_a] = aa_min
              + (((aa_max - aa_min)/(Na - 1.0))*(n_a - 1.0));
  }

  for (n_m=1;n_m<=Nm;n_m++){      /* Equally linearly spaced */
    emmeemme[n_m] = emmeemme_min
                    + (((emmeemme_max - emmeemme_min)
                    /(Nm - 1.0))*(n_m - 1.0));
  }

  if (argv[1][1]=='f') {sscanf(argv[2],"%s",file_name);}
    else {
      printf("ERROR: please give file with spherical model
as input. The computation will be stopped.");
      return 0;
    }

   for(n_a=1;n_a<=Na;n_a++){
    for(n_m=1;n_m<=Nm;n_m++){
```

```
load_sphere(file_name);

for(i=1;i<=NT*(NR+1)+10;i++) init[i] = SPH_val[i];

a = aa[n_a]/SQ(init[NT]);
  /* By Stoeckly (1965) */
emme = emmeemme[n_m]*SPH_val[NT];
  /* By Eriguchi, Mueller (1985) */

cntr = 0;
modnum = 0;
obl=1.0;

print_header(n_a,n_m);

for (i=1;i<=NT*(NR+1)+10;i++) {
    printf("%f ", init[i]);
    if(i==NT*(NR+1)+10) printf("\n");
  }

make_grid(init);

leg_polyn();

obl = step;
modnum=1;

newt(init,NT*(NR+1)+2,&CKC,vecfunc);

MAX_DENS_VAL = 1.0;

for (k=NT+1;k<=NT*(NR+1);k++){
    if ((init[k] - MAX_DENS_VAL - TOLDENS)>=0.0) {
      MAX_DENS_VAL = init[k];
    }
  } /* check if result of N-R is ring-like */

if(CKC==0) {
  PHYS_QUANT(init,modnum);

  for (i=1;i<=NT*(NR+1)+10;i++) {
       printf("%f ", init[i]);
       if(i==NT*(NR+1)+10) printf("\n");
      }
```

```
    OUT_FILE[Nm*(n_a-1)+n_m][1] = emmeemme[n_m];
    OUT_FILE[Nm*(n_a-1)+n_m][2] = aa[n_a];
    if ((MAX_DENS_VAL-1.0)>0.0) {
      if (init[NT*(NR+1)+8] > 0.0) {
        OUT_FILE[Nm*(n_a-1)+n_m][3] = 10;
      }
      else OUT_FILE[Nm*(n_a-1)+n_m][3] = 7;
    }
    else {OUT_FILE[Nm*(n_a-1)+n_m][3] = cntr;}

    }

  else {printf("ERROR IN N-R METHOD\n");
        OUT_FILE[Nm*(n_a-1)+n_m][1] = emmeemme[n_m];
        OUT_FILE[Nm*(n_a-1)+n_m][2] = aa[n_a];
        if ((MAX_DENS_VAL-1.0)>0.0) {
          OUT_FILE[Nm*(n_a-1)+n_m][3] = 7;
        }
        else {OUT_FILE[Nm*(n_a-1)+n_m][3] = 3;}
        break;}


  while(cntr==0 || cntr==6 || cntr==7){

        obl *= step;
        modnum += 1;
        a = aa[n_a]/SQ(init[NT]);

         /* a change from one configuration
            to the subsequent */

        make_grid(init);

        leg_polyn();

        newt(init,NT*(NR+1)+2,&CKC,vecfunc);

        MAX_DENS_VAL = 1.0;

        for (k=NT+1;k<=NT*(NR+1);k++){
            if ((init[k] - MAX_DENS_VAL - TOLDENS)>=0.0)
            {
              MAX_DENS_VAL = init[k];
            }
        }    /* check if result of N-R is ring-like */
```

```
            if(CKC==0) {
               PHYS_QUANT(init,modnum);

               for (i=1;i<=NT*(NR+1)+10;i++) {
               printf("%f ", init[i]);
               if(i==NT*(NR+1)+10) printf("\n");
                    }
               OUT_FILE[Nm*(n_a-1)+n_m][1] = emmeemme[n_m];
               OUT_FILE[Nm*(n_a-1)+n_m][2] = aa[n_a];
               if ((MAX_DENS_VAL-1.0)>0.0) {
                  if (init[NT*(NR+1)+8] > 0.0) {
                    OUT_FILE[Nm*(n_a-1)+n_m][3] = 10;
                  }
                  else OUT_FILE[Nm*(n_a-1)+n_m][3] = 7;
               }
               else {OUT_FILE[Nm*(n_a-1)+n_m][3] = cntr;}
              }

            else {printf("ERROR IN N-R METHOD\n");
                  OUT_FILE[Nm*(n_a-1)+n_m][1] = emmeemme[n_m];
                  OUT_FILE[Nm*(n_a-1)+n_m][2] = aa[n_a];
                  if ((MAX_DENS_VAL-1.0)>0.0) {
                    OUT_FILE[Nm*(n_a-1)+n_m][3] = 7;
                  }
                  else {OUT_FILE[Nm*(n_a-1)+n_m][3] = 3;}
                  break;}
          }

   }

}

TOT=fopen("CONTROL_PARAM.txt","w");

if (TOT == NULL)
  {
  printf("Error opening file!\n");
  exit(1);
  }

else{
  fprintf(TOT,"   m   |   a   |  cntr  \n");

  for(n_a=1;n_a<=Na;n_a++){
```

```
        for(n_m=1;n_m<=Nm;n_m++){
          fprintf(TOT,"%f %f %f\n",OUT_FILE[Nm*(n_a-1)+n_m][1],
                  OUT_FILE[Nm*(n_a-1)+n_m][2],
                  OUT_FILE[Nm*(n_a-1)+n_m][3]);
        }
      }
    }

    fclose(TOT);

    return 0;

}
```

Bibliography

[1] J. Antoniadis, P. C. C. Freire, N. Wex, T. M. Tauris, R. S. Lynch, M. H. van Kerkwijk, M. Kramer, C. Bassa, V. S. Dhillon, T. Driebe, J. W. T. Hessels, V. M. Kaspi, V. I. Kondratiev, N. Langer, T. R. Marsh, M. A. McLaughlin, T. T. Pennucci, S. M. Ransom, I. H. Stairs, J. van Leeuwen, J. P. W. Verbiest, and D. G. Whelan. A Massive Pulsar in a Compact Relativistic Binary. *Science*, 340:448, April 2013.

[2] J. M. Bardeen. Stability of Circular Orbits in Stationary, Axisymmetric Space-Times. ApJ, 161:103, July 1970.

[3] J. M. Bardeen, W. H. Press, and S. A. Teukolsky. Rotating Black Holes: Locally Nonrotating Frames, Energy Extraction, and Scalar Synchrotron Radiation. ApJ, 178:347–370, December 1972.

[4] M. Bauböck, F. Özel, D. Psaltis, and S. M. Morsink. Rotational Corrections to Neutron-star Radius Measurements from Thermal Spectra. ApJ, 799:22, January 2015.

[5] G. Baym, C. Pethick, and P. Sutherland. The Ground State of Matter at High Densities: Equation of State and Stellar Models. ApJ, 170:299, December 1971.

[6] L. Becerra, F. Cipolletta, C. L. Fryer, J. A. Rueda, and R. Ruffini. Angular Momentum Role in the Hypercritical Accretion of Binary-driven Hypernovae. ApJ, 812:100, October 2015.

[7] R. Belvedere, K. Boshkayev, J. A. Rueda, and R. Ruffini. Uniformly rotating neutron stars in the global and local charge neutrality cases. *Nuclear Physics A*, 921:33–59, January 2014.

[8] R. Belvedere, D. Pugliese, J. A. Rueda, R. Ruffini, and S.-S. Xue. Neutron star equilibrium configurations within a fully relativistic theory with strong, weak, electromagnetic, and gravitational interactions. *Nuclear Physics A*, 883:1–24, June 2012.

[9] O. Benhar, V. Ferrari, L. Gualtieri, and S. Marassi. Perturbative approach to the structure of rapidly rotating neutron stars. Phys. Rev. D, 72(4):044028, August 2005.

[10] J. Boguta and A. R. Bodmer. Relativistic calculation of nuclear matter and the nuclear surface. *Nuclear Physics A*, 292:413–428, December 1977.

[11] K. Boshkayev, H. Quevedo, and R. Ruffini. Gravitational field of compact objects in general relativity. Phys. Rev. D, 86(6):064043, September 2012.

[12] E. M. Butterworth and J. R. Ipser. On the structure and stability of rapidly rotating fluid bodies in general relativity. I - The numerical method for computing structure and its application to uniformly rotating homogeneous bodies. ApJ, 204:200–223, February 1976.

[13] P. A. Caraveo. Gamma-Ray Pulsar Revolution. ARA&A, 52:211–250, August 2014.

[14] S. Chakrabarti, T. Delsate, N. Gürlebeck, and J. Steinhoff. I-Q Relation for Rapidly Rotating Neutron Stars. *Physical Review Letters*, 112(20):201102, May 2014.

[15] S. Chandrasekhar. The equilibrium of distorted polytropes. I. The rotational problem. MNRAS, 93:390–406, March 1933.

[16] S. Chandrasekhar. *Hydrodynamic and hydromagnetic stability*. International Series of Monographs on Physics, Oxford: Clarendon. 1961.

[17] C. Cherubini, S. Filippi, R. Ruffini, A. Sepulveda, and J. I. Zuluaga. Non-Homogeneous Axisymmetric Models of Self-Gravitating Systems. In H. Kleinert, R. T. Jantzen, and R. Ruffini, editors, *The Eleventh Marcel Grossmann Meeting On Recent Developments in Theoretical and Experimental General Relativity, Gravitation and Relativistic Field Theories*, pages 2340–2342, September 2008.

[18] F. Cipolletta, C. Cherubini, S. Filippi, J. A. Rueda, and R. Ruffini. Fast rotating neutron stars with realistic nuclear matter equation of state. Phys. Rev. D, 92(2):023007, July 2015.

[19] G. B. Cook, S. L. Shapiro, and S. A. Teukolsky. Rapidly rotating neutron stars in general relativity: Realistic equations of state. ApJ, 424:823–845, April 1994.

[20] G. B. Cook, S. L. Shapiro, and S. A. Teukolsky. Rapidly rotating polytropes in general relativity. ApJ, 422:227–242, February 1994.

[21] S. M. de Carvalho, R. Negreiros, J. A. Rueda, and R. Ruffini. Thermal evolution of neutron stars with global and local neutrality. Phys. Rev. C, 90(5):055804, November 2014.

[22] P. B. Demorest, T. Pennucci, S. M. Ransom, M. S. E. Roberts, and J. W. T. Hessels. A two-solar-mass neutron star measured using Shapiro delay. Nature, 467:1081–1083, October 2010.

[23] M. Dutra, O. Lourenço, S. S. Avancini, B. V. Carlson, A. Delfino, D. P. Menezes, C. Providência, S. Typel, and J. R. Stone. Relativistic mean-field hadronic models under nuclear matter constraints. Phys. Rev. C, 90(5):055203, November 2014.

[24] Y. Eriguchi and E. Mueller. A general computational method for obtaining equilibria of self-gravitating and rotating gases. A&A, 146:260–268, May 1985.

[25] Y. Eriguchi and E. Mueller. Equilibrium models of differentially rotating polytropes and the collapse of rotating stellar cores. A&A, 147:161–168, June 1985.

[26] M. Fortin, J. L. Zdunik, P. Haensel, and M. Bejger. Neutron stars with hyperon cores: stellar radii and equation of state near nuclear density. A&A, 576:A68, April 2015.

[27] J. L. Friedman, J. R. Ipser, and R. D. Sorkin. Turning-point method for axisymmetric stability of rotating relativistic stars. ApJ, 325:722–724, February 1988.

[28] C. L. Fryer, J. A. Rueda, and R. Ruffini. Hypercritical Accretion, Induced Gravitational Collapse, and Binary-Driven Hypernovae. ApJ, 793:L36, October 2014.

[29] R. Geroch. Multipole Moments. I. Flat Space. *Journal of Mathematical Physics*, 11:1955–1961, June 1970.

[30] R. Geroch. Multipole Moments. II. Curved Space. *Journal of Mathematical Physics*, 11:2580–2588, August 1970.

[31] N. K. Glendenning and S. A. Moszkowski. Reconciliation of neutron-star masses and binding of the Lambda in hypernuclei. *Physical Review Letters*, 67:2414–1417, October 1991.

[32] T. Gold. Rotating Neutron Stars as the Origin of the Pulsating Radio Sources. Nature, 218:731–732, May 1968.

[33] R. O. Hansen. Multipole moments of stationary space-times. *Journal of Mathematical Physics*, 15:46–52, January 1974.

[34] A. K. Harding. The neutron star zoo. *Frontiers of Physics*, 8:679–692, December 2013.

[35] J. B. Hartle. Slowly Rotating Relativistic Stars. I. Equations of Structure. ApJ, 150:1005, December 1967.

[36] C. O. Heinke, H. N. Cohn, P. M. Lugger, N. A. Webb, W. C. G. Ho, J. Anderson, S. Campana, S. Bogdanov, D. Haggard, A. M. Cool, and J. E. Grindlay. Improved mass and radius constraints for quiescent neutron stars in ω Cen and NGC 6397. MNRAS, 444:443–456, October 2014.

[37] C. O. Heinke, G. B. Rybicki, R. Narayan, and J. E. Grindlay. A Hydrogen Atmosphere Spectral Model Applied to the Neutron Star X7 in the Globular Cluster 47 Tucanae. ApJ, 644:1090–1103, June 2006.

[38] J. W. T. Hessels, S. M. Ransom, I. H. Stairs, P. C. C. Freire, V. M. Kaspi, and F. Camilo. A Radio Pulsar Spinning at 716 Hz. *Science*, 311:1901–1904, March 2006.

[39] G. P. Horedt, editor. *Polytropes - Applications in Astrophysics and Related Fields*, volume 306 of *Astrophysics and Space Science Library*, July 2004.

[40] R. A. James. The Structure and Stability of Rotating Gas Masses. ApJ, 140:552, August 1964.

[41] J. H. Jeans. *Astronomy and cosmogony.* Cambridge [Eng.] The University press. 1928.

[42] H. Komatsu, Y. Eriguchi, and I. Hachisu. Rapidly rotating general relativistic stars. I - Numerical method and its application to uniformly rotating polytropes. MNRAS, 237:355–379, March 1989.

[43] G. A. Lalazissis, J. König, and P. Ring. New parametrization for the Lagrangian density of relativistic mean field theory. Phys. Rev. C, 55:540–543, January 1997.

[44] J. M. Lattimer and A. W. Steiner. Constraints on the symmetry energy using the mass-radius relation of neutron stars. *European Physical Journal A*, 50:40, February 2014.

[45] A. Maeder. *Physics, Formation and Evolution of Rotating Stars.* Astronomy and Astrophysics Library. ISBN 978-3-540-76948-4. Springer Berlin Heidelberg. 2009.

[46] R. Meinel, M. Ansorg, A. Kleinwächter, G. Neugebauer, and D. Petroff. *Relativistic Figures of Equilibrium.* Cambridge, UK: Cambridge University Press. September 2012.

[47] F. G. Oliveira, J. A. Rueda, and R. Ruffini. Gravitational Waves versus X-Ray and Gamma-Ray Emission in a Short Gamma-Ray Burst. ApJ, 787:150, June 2014.

[48] J. P. Ostriker and P. Bodenheimer. Rapidly Rotating Stars. II. Massive White Dwarfs. ApJ, 151:1089, March 1968.

[49] J. P. Ostriker and J. W.-K. Mark. Rapidly rotating stars. I. The self-consistent-field method. ApJ, 151:1075–1088, March 1968.

[50] L. A. Pachón, J. A. Rueda, and J. D. Sanabria-Gómez. Realistic exact solution for the exterior field of a rotating neutron star. Phys. Rev. D, 73(10):104038, May 2006.

[51] L. A. Pachón, J. A. Rueda, and C. A. Valenzuela-Toledo. On the Relativistic Precession and Oscillation Frequencies of Test Particles around Rapidly Rotating Compact Stars. ApJ, 756:82, September 2012.

[52] F. Pacini. Energy Emission from a Neutron Star. Nature, 216:567–568, November 1967.

[53] S. Pal, D. Bandyopadhyay, and W. Greiner. Antikaon condensation in neutron stars. *Nuclear Physics A*, 674:553–577, July 2000.

[54] G. Pappas and T. A. Apostolatos. Revising the Multipole Moments of Numerical Spacetimes and its Consequences. *Physical Review Letters*, 108(23):231104, June 2012.

[55] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes in C. The art of scientific computing.* Cambridge: University Press. 1992.

[56] J. A. Rueda and R. Ruffini. On the Induced Gravitational Collapse of a Neutron Star to a Black Hole by a Type Ib/c Supernova. ApJ, 758:L7, October 2012.

[57] J. A. Rueda, R. Ruffini, Y.-B. Wu, and S.-S. Xue. Surface tension of the core-crust interface of neutron stars with global charge neutrality. Phys. Rev. C, 89(3):035804, March 2014.

[58] F. D. Ryan. Gravitational waves from the inspiral of a compact object into a massive, axisymmetric body with arbitrary multipole moments. Phys. Rev. D, 52:5707–5718, November 1995.

[59] N. Stergioulas. Rotating Stars in Relativity. *Living Reviews in Relativity*, 6:3, June 2003.

[60] N. Stergioulas and J. L. Friedman. Comparing models of rapidly rotating relativistic stars constructed by two numerical methods. ApJ, 444:306–311, May 1995.

[61] R. Stoeckly. Polytropic Models with Fast, Non-Uniform Rotation. ApJ, 142:208–228, July 1965.

[62] Y. Sugahara and H. Toki. Relativistic mean-field theory for unstable nuclei with non-linear σ and ω terms. *Nuclear Physics A*, 579:557–572, October 1994.

[63] J.-L. Tassoul. *Theory of rotating stars.* Princeton Series in Astrophysics, Princeton: University Press. 1978.

[64] T. M. Tauris, V. M. Kaspi, R. P. Breton, A. T. Deller, E. F. Keane, M. Kramer, D. R. Lorimer, M. A. McLaughlin, A. Possenti, P. S. Ray, B. W. Stappers, and P. Weltevrede. Understanding the Neutron Star Population with the SKA. *Advancing Astrophysics with the Square Kilometre Array (AASKA14)*, page 39, 2015.

[65] M. Urbanec, J. C. Miller, and Z. Stuchlík. Quadrupole moments of rotating neutron stars and strange stars. MNRAS, 433:1903–1909, August 2013.

[66] R. M. Wald. *General relativity.* Chicago, University of Chicago Press. 1984.

[67] K. Yagi, K. Kyutoku, G. Pappas, N. Yunes, and T. A. Apostolatos. Effective no-hair relations for neutron stars and quark stars: Relativistic results. Phys. Rev. D, 89(12):124013, June 2014.

[68] K. Yagi and N. Yunes. I-Love-Q relations in neutron stars and their applications to astrophysics, gravitational waves, and fundamental physics. Phys. Rev. D, 88(2):023009, July 2013.