



Ministério da  
**Ciência, Tecnologia  
e Inovação**



sid.inpe.br/mtc-m19/2012/07.23.14.31-TDI

## ESTRELAS DE NÊUTRONS COMO FONTES DE ONDAS GRAVITACIONAIS E A DETECÇÃO DO MODO FUNDAMENTAL

Patrick da Rosa Silveira

Dissertação de Mestrado do Curso de Pós-Graduação em Astrofísica, orientada pelos Drs. Cecilia B. M. H. Chirenti, e Odylio Denys de Aguiar, aprovada em 24 de maio de 2012.

URL do documento original:

<<http://urlib.net/8JMKD3MGP7W/3CB8D2H>>

INPE  
São José dos Campos  
2012

**PUBLICADO POR:**

Instituto Nacional de Pesquisas Espaciais - INPE

Gabinete do Diretor (GB)

Serviço de Informação e Documentação (SID)

Caixa Postal 515 - CEP 12.245-970

São José dos Campos - SP - Brasil

Tel.:(012) 3208-6923/6921

Fax: (012) 3208-6919

E-mail: pubtc@sid.inpe.br

**CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO DA PRODUÇÃO INTELLECTUAL DO INPE (RE/DIR-204):****Presidente:**

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

**Membros:**

Dr. Antonio Fernando Bertachini de Almeida Prado - Coordenação Engenharia e Tecnologia Espacial (ETE)

Dr<sup>a</sup> Inez Staciarini Batista - Coordenação Ciências Espaciais e Atmosféricas (CEA)

Dr. Gerald Jean Francis Banon - Coordenação Observação da Terra (OBT)

Dr. Germano de Souza Kienbaum - Centro de Tecnologias Especiais (CTE)

Dr. Manoel Alonso Gan - Centro de Previsão de Tempo e Estudos Climáticos (CPT)

Dr<sup>a</sup> Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação

Dr. Plínio Carlos Alvalá - Centro de Ciência do Sistema Terrestre (CST)

**BIBLIOTECA DIGITAL:**

Dr. Gerald Jean Francis Banon - Coordenação de Observação da Terra (OBT)

**REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:**

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)

**EDITORAÇÃO ELETRÔNICA:**

Ivone Martins - Serviço de Informação e Documentação (SID)



Ministério da  
**Ciência, Tecnologia  
e Inovação**



sid.inpe.br/mtc-m19/2012/07.23.14.31-TDI

## ESTRELAS DE NÊUTRONS COMO FONTES DE ONDAS GRAVITACIONAIS E A DETECÇÃO DO MODO FUNDAMENTAL

Patrick da Rosa Silveira

Dissertação de Mestrado do Curso de Pós-Graduação em Astrofísica, orientada pelos Drs. Cecilia B. M. H. Chirenti, e Odylio Denys de Aguiar, aprovada em 24 de maio de 2012.

URL do documento original:

<<http://urlib.net/8JMKD3MGP7W/3CB8D2H>>

INPE  
São José dos Campos  
2012

Dados Internacionais de Catalogação na Publicação (CIP)

---

Silveira, Patrick da Rosa.  
Si39e Estrelas de nêutrons como fontes de ondas gravitacionais e a  
detecção do modo fundamental / Patrick da Rosa Silveira. – São  
José dos Campos : INPE, 2012.  
xviii + 116 p. ; (sid.inpe.br/mtc-m19/2012/07.23.14.31-TDI)

Dissertação (Mestrado em Astrofísica) – Instituto Nacional de  
Pesquisas Espaciais, São José dos Campos, 2012.

Orientadores : Drs. Cecilia B. M. H. Chirenti, e Odylio Denys  
de Aguiar.

1. estrelas de neutrôns. 2. modos quasinormais. 3. modo fun-  
damental. 4. ondas gravitacionais relatividade geral. I.Título.

CDU 524.354.6

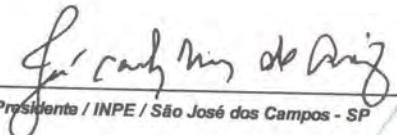
---

Copyright © 2012 do MCT/INPE. Nenhuma parte desta publicação pode ser reproduzida, armazenada em um sistema de recuperação, ou transmitida sob qualquer forma ou por qualquer meio, eletrônico, mecânico, fotográfico, reprográfico, de microfilmagem ou outros, sem a permissão escrita do INPE, com exceção de qualquer material fornecido especificamente com o propósito de ser entrado e executado num sistema computacional, para o uso exclusivo do leitor da obra.

Copyright © 2012 by MCT/INPE. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, microfilming, or otherwise, without written permission from INPE, with the exception of any material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use of the reader of the work.

Aprovado (a) pela Banca Examinadora  
em cumprimento ao requisito exigido para  
obtenção do Título de **Mestre** em  
**Astrofísica**


Dr. José Carlos Neves de Araújo

  
Presidente / INPE / São José dos Campos - SP

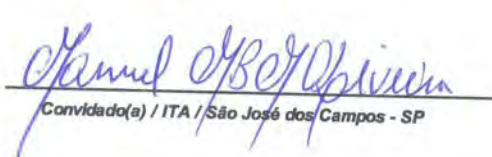
Dr. Odylio Denys de Aguiar

  
Orientador(a) / INPE / SJC Campos - SP

Dra. Cecília Bertoni Martha Hadler Chirenti

  
Orientador(a) / UFABC / Santo André - SP

Dr. Manuel Maximo Bastos Malheiro de  
Oliveira

  
Convidado(a) / ITA / São José dos Campos - SP

*Este trabalho foi aprovado por:*

maioria simples

unanimidade

Aluno (a): *Patrick da Rosa Silveira*

São José dos Campos, 24 de maio de 2012



*“A imaginação é mais importante que a ciência, porque a ciência é limitada, ao passo que a imaginação abrange o mundo inteiro”.*

ALBERT EINSTEIN





## **AGRADECIMENTOS**

A minha família que me apoiou incondicionalmente.

Aos amigos e colegas pela convivência e pela ajuda com detalhes técnicos em Fortran e LaTeX.

Aos professores que dividiram seus conhecimentos durante as aulas.

A minha orientadora Cecília Chirenti pelo aporte científico e ajuda no desenvolvimento de todos os programas em Fortran.

Em especial ao meu orientador Odylio Aguiar que me apoiou/ajudou durante todo o mestrado e nunca desistiu de acreditar em mim.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pelo suporte financeiro.



## RESUMO

O trabalho baseia-se no estudo de estrelas de nêutrons isoladas como fonte de ondas gravitacionais e a análise em particular do modo quasinormal fundamental (modo  $f$ ). Para tanto, obtemos primeiramente um modelo relativístico para uma estrela de nêutrons em equilíbrio, integrando numericamente as equações TOV, as quais descrevem um corpo esfericamente simétrico em equilíbrio gravitacional. Obtemos a solução do modo  $f$  para uma estrela de nêutrons com pulsações não-radiais, através da integração numérica das equações de perturbação. Espera-se que a maior parte da radiação gravitacional emitida pela estrela seja feita através desse modo, o que deverá facilitar a sua detecção. Além disso, a banda de frequências de detecção do detector de ondas gravitacionais Mario Schenberg coincide com uma faixa das frequências previstas para os modos  $f$ , possibilitando a sua observação. A detecção destas frequências fornecerá uma forma de se obter a massa e o raio da estrela de nêutrons emissora da radiação gravitacional observada.



# **NEUTRON STARS AS SOURCES OF GRAVITATIONAL WAVES AND THE DETECTION OF THE FUNDAMENTAL MODE**

## **ABSTRACT**

The work is based on isolated neutron stars as gravitational wave sources and, in particular, the fundamental quasinormal mode (f mode) analysis. Firstly, we obtain a relativistic model for a neutron star in equilibrium by numerically integrating the TOV equations, which describe a spherically symmetric body in gravitational equilibrium. We obtain the f mode solution for a neutron star with non-radial pulsations by the numerical integration of the perturbation equations. We expect that the largest portion of the gravitational radiation emitted by the star be in this mode channel, which must make its detection easier. Furthermore, the Mario Schenberg gravitational wave has its frequency band for detection in the range of some of the frequencies expected for the f modes, making its observation possible for it. The detection of these frequencies will provide a way to obtain the mass and radius of the neutron star that emitted the observed gravitational waves.



## LISTA DE FIGURAS

	<u>Pág.</u>
1.1 Diagrama de fase para matéria a densidade e temperatura extremas . . . .	2
2.1 Joseph Weber trabalhando em seu detector de ondas gravitacionais . . . .	6
2.2 A posição do LISA em relação ás órbitas da Terra e do Sol (esquerda). Órbitas da Terra e do LISA em torno do Sol. . . . .	8
2.3 Pulsar Binário emitindo OGs. . . . .	9
2.4 Órbita de um Pulsar Binário . . . . .	10
2.5 Mudança na hora do Periastro . . . . .	11
2.6 Detector Mario Schenberg. . . . .	15
3.1 Variação da pressão no interior estelar. . . . .	22
3.2 Gráfico $m(r)$ vs $r$ . . . . .	23
3.3 Gráfico mostrando a variação da densidade de energia com o raio. . . . .	23
3.4 Potencial gravitacional no interior estelar. . . . .	24
4.1 Divisão do interior estelar para o cálculo das perturbações. . . . .	28
4.2 $H_{11}(r)$ vs $r$ . . . . .	30
4.3 $K_1(r)$ vs $r$ . . . . .	30
4.4 $W_1(r)$ vs $r$ . . . . .	31
4.5 $X_1(r)$ vs $r$ . . . . .	31
4.6 $H_{12}(r)$ vs $r$ . . . . .	32
4.7 $K_2(r)$ vs $r$ . . . . .	33
4.8 $W_2(r)$ vs $r$ . . . . .	33
4.9 $X_2(r)$ vs $r$ . . . . .	34
4.10 $H_{13}(r)$ vs $r$ . . . . .	34
4.11 $K_3(r)$ vs $r$ . . . . .	35
4.12 $W_3(r)$ vs $r$ . . . . .	35
4.13 $X_3(r)$ vs $r$ . . . . .	36
4.14 $H_{14}(r)$ vs $r$ . . . . .	41
4.15 $K_4(r)$ vs $r$ . . . . .	41
4.16 $W_4(r)$ vs $r$ . . . . .	42
4.17 $X_4(r)$ vs $r$ . . . . .	42
4.18 $H_{15}(r)$ vs $r$ . . . . .	43
4.19 $K_5(r)$ vs $r$ . . . . .	43
4.20 $W_5(r)$ vs $r$ . . . . .	44

4.21	$X_5(r)$ vs $r$ .	44
4.22	$H_1(r)$ vs $r$ .	47
4.23	$K(r)$ vs $r$ .	47
4.24	$W(r)$ vs $r$ .	48
4.25	$X(r)$ vs $r$ .	48
4.26	$Z(r^*)$ vs $r$ e $Z(r^*)$ vs $r^*$ .	51
4.27	$Z'(r^*)$ vs $r$ e $Z'(r^*)$ vs $r^*$ .	52
4.28	$V_z(r^*)$ vs $r$ e $V_z(r^*)$ vs $r^*$ .	53
4.29	$r^*(r)$ vs $r$ .	54
A.1	Gráficos das derivadas das perturbações referentes a solução $Y_1$ .	61
A.2	Gráficos das derivadas das perturbações referentes a solução $Y_2$ .	62
A.3	Gráficos das derivadas das perturbações referentes a solução $Y_3$ .	63
A.4	Gráficos das derivadas das perturbações referentes a solução $Y_4$ .	64
A.5	Gráficos das derivadas das perturbações referentes a solução $Y_5$ .	65



## LISTA DE TABELAS

	<u>Pág.</u>
4.1 Condições Iniciais . . . . .	29



## SUMÁRIO

	<u>Pág.</u>
<b>1 INTRODUÇÃO</b> . . . . .	<b>1</b>
<b>2 ONDAS GRAVITACIONAIS</b> . . . . .	<b>5</b>
2.1 Detectores de Ondas Gravitacionais . . . . .	5
2.2 Prova indireta da observação de OGs (o PSR1913+16) . . . . .	9
2.3 Relatividade Geral e as Ondas Gravitacionais . . . . .	12
2.4 O Detector de Ondas Gravitacionais Mario Schenberg . . . . .	14
<b>3 ESTRELAS DE NÊUTRONS</b> . . . . .	<b>17</b>
3.1 Estrelas de nêutrons e as ondas gravitacionais . . . . .	18
3.2 Equação de Tolman-Oppenheimer-Volkoff . . . . .	18
3.3 Estrutura Interna das Estrelas de Nêutrons . . . . .	20
3.4 Resultados . . . . .	22
<b>4 MODOS QUASINORMAIS</b> . . . . .	<b>25</b>
4.1 Modo Quasinormal Fundamental . . . . .	25
4.2 Cálculo dos Modos Quasinormais . . . . .	26
4.2.1 Perturbações de $\frac{R}{2}$ até a Superfície da estrela . . . . .	27
4.2.2 Perturbações do Centro até $\frac{R}{2}$ . . . . .	32
4.2.3 Perturbações do Centro até a Superfície . . . . .	40
4.2.4 Equação de Zerilli . . . . .	48
4.2.5 Frequência do Modo Quasinormal Fundamental . . . . .	51
<b>5 Conclusões</b> . . . . .	<b>57</b>

REFERÊNCIAS BIBLIOGRÁFICAS . . . . .	59
ANEXO A - DERIVADAS DAS PERTURBAÇÕES . . . . .	61
ANEXO B - ESTRUTURA INTERNA DE UMA ESTRELA DE NÊUTRONS . . . . .	67
ANEXO C - PERTURBAÇÕES DA SUPERFÍCIE DA ESTRELA ATÉ $\frac{R}{2}$ . . . . .	79
ANEXO D- CÁLCULO DAS PERTURBAÇÕES DO CENTRO ATÉ $\frac{R}{2}$ . . . . .	89
ANEXO E - PERTURBAÇÕES EM TODO INTERIOR ESTELAR	99
ANEXO F - CÁLCULO DA FREQUÊNCIA DO MODO F. . . . .	103

# 1 INTRODUÇÃO

Albert Einstein, em 1915, fez uma das mais importantes “descobertas” do século XX, a Relatividade Geral (RG). A Relatividade Geral é uma generalização da Teoria de Gravitação de Isaac Newton, onde a base matemática foi desenvolvida pelo cientista francês Henri Poincaré. Esta generalização mostrou que a gravitação é um efeito da geometria do espaço-tempo, onde a matéria (energia) curva o espaço e o tempo à sua volta.

Uma das previsões da RG é a das ondas gravitacionais (OGs). Na seção 2 veremos como a RG prevê as OGs, assim como uma prova indireta da existência, feita por Hulse e Taylor em 1975.

Os modos quasinormais podem ser calculados quando perturbamos as equações de Einstein que descrevem a estrela e o seu espaço-tempo. No caso de buracos negros, as famílias de modos não dependem da perturbação inicial, sendo assim, existem apenas 2 famílias, uma com paridade axial e outra com paridade polar. As frequências referentes a estes modos, são proporcionais à massa do buraco negro. No entanto, para EN, surgem muitas famílias de modos devido às oscilações do fluido estelar acopladas às oscilações do espaço-tempo. Essas famílias de modos excitadas serão determinadas pela perturbação inicial. Temos também que as frequências de uma dada família muda de acordo com a equação de estado adotada para o interior estelar.

Uma das motivações desse trabalho, assim como do estudo de ondas gravitacionais oriundas de ENs, é a possibilidade de conhecer a sua equação de estado através da observação de OGs. Existem mecanismos matemáticos que possibilitam determinar características da EN, tais como raio, massa e momento de inércia, utilizando as frequências dos modos quasinormais. Além de ser possível determinar a equação de estado, as ENs podem ser um laboratório para o estudo da matéria a altas temperaturas e densidades (Fig. 1.1) (AFORD, 2010).

Na figura 1.1, o eixo horizontal representa o potencial químico de quarks e o eixo vertical é a temperatura. Para baixas temperaturas, temos que à medida que  $\mu$  aumenta, o sistema passa de vácuo para matéria nuclear ( $\mu \approx 310MeV$ ). Para um alto e desconhecido potencial químico existe uma transição para a matéria de quarks. As estrelas de nêutrons encontram-se na parte inferior do diagrama de fase,

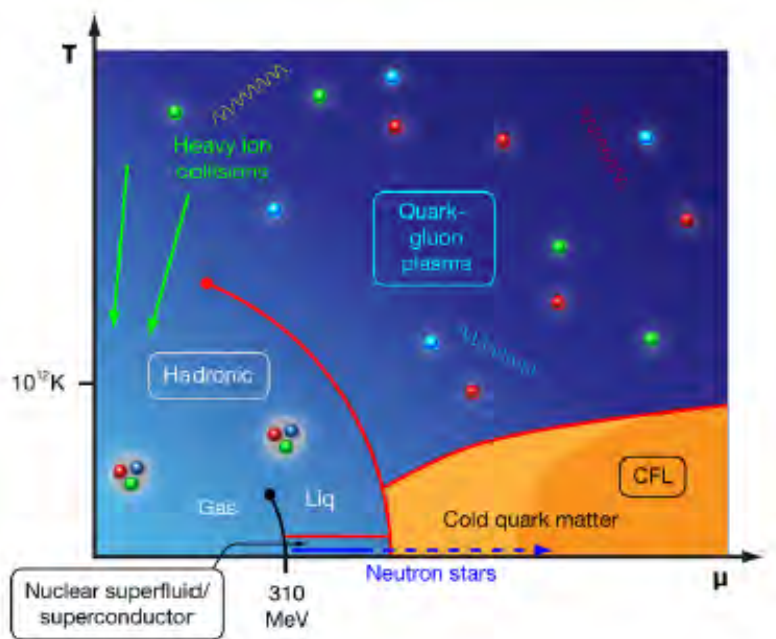


Figura 1.1 - Diagrama de fase para matéria a densidade e temperatura extremas  
 Fonte: Ilustração de Alan Stonebraker (2010)

onde a densidade central dela vai até um máximo desconhecido que depende da matéria nuclear.

Focamos nosso estudo especificamente no modo quasinormal fundamental (modo f). Para estrelas compactas, sem rotação, o modo f é o modo quasinormal (MQN) polar e não-radial com momento angular de índice  $L=2$ .

Um dos motivos para estudar exclusivamente o modo f é devido a sua baixa frequência, sendo assim, o modo f é mais fácil de detectar do que outros modos. Atualmente não existe mecanismo para determinar massa e raio de uma estrela de nêutrons isolada, no entanto, como sabemos, é através desse modo que podemos obter esses parâmetros da estrela compacta.

É através do modo fundamental que a maior parte da radiação gravitacional da estrela será irradiada. Este é o modo não-radial polar e pode ser interpretado como sendo causado pela interface entre a estrela e o seu exterior. Portanto os modos f são úteis para se estimar as massas e raios de estrelas de nêutrons.

Antes de obter os MQNs, realizamos um estudo de uma estrela de nêutrons (ENs) com equação de estado politrópica. Tal estudo foi feito aplicando um método numérico e resolvendo as equações de Tolman-Oppenheimer-Volkoff (TOV) para uma equação de estado politrópica.

Posteriormente desenvolvemos uma série de programas em Fortran 90 para calcular o modo f para este modelo (estrela politrópica). Depois de obtidos os MQNs, vamos estudar a cena inversa, ou seja, como obter parâmetros físicos da estrela através da onda gravitacional observada.

Mas primeiramente vamos falar sobre as OGs e seus potenciais detectores.





## 2 ONDAS GRAVITACIONAIS

Podemos fazer uma simples analogia para entender o que são as OGs. Assim como a luz ou uma onda de rádio são campos eletromagnéticos oscilantes se propagando, emitidos quando partículas carregadas eletricamente são aceleradas, as ondas gravitacionais são oscilações do campo gravitacional se propagando, geradas, neste caso, por massas aceleradas.

Ondas gravitacionais transmitem energia através de deformações do espaço-tempo, isto é, através do campo gravitacional. Da mesma maneira que um objeto ao cair em um lago gera uma série de ondas com origem no ponto onde a superfície da água é inicialmente perturbada, o movimento de corpos massivos num campo gravitacional pode gerar à sua volta pequenas perturbações no espaço-tempo. A estas perturbações que se propagam no "tecido" do espaço-tempo damos o nome de OGs.

Existem diversos experimentos ao redor do mundo que buscam evidências de ondas gravitacionais. Eles se baseiam em tentar detectar alterações da energia interna de corpos massivos a baixíssimas temperaturas, confinados em sistemas amortecidos. Essas alterações da energia interna seriam supostamente causadas por ondas gravitacionais oriundas de megaeventos no espaço, tal como o choque de estrelas (SCHUTZ, 1999).

### 2.1 Detectores de Ondas Gravitacionais

Temos basicamente 4 tipos de técnicas na busca de OGs: detectores de barras ressonantes, detectores interferométricos (espaciais e terrestres), Pulsar Timing Array (PTAs) e Medidas do modo de polarização B da CMB. Veremos na sequência um pouco mais sobre detectores de barras ressonantes e interferométricos.

**Detectores de Barras Ressonantes** - O trabalho pioneiro foi o detector de Weber (Fig. 2.1), construído na década de 60. O detector consistia de um cilindro de uma liga de alumínio com uma massa da ordem de 1 tonelada no interior de uma câmara de vácuo, a fim de isolar a barra de vibrações externas. No entanto, em 1969, Weber após grandes melhorias, possuía sensibilidade de apenas  $10^{-16}$ . O sinal era dominado por ruídos dada a baixa intensidade das ondas gravitacionais a serem observadas.

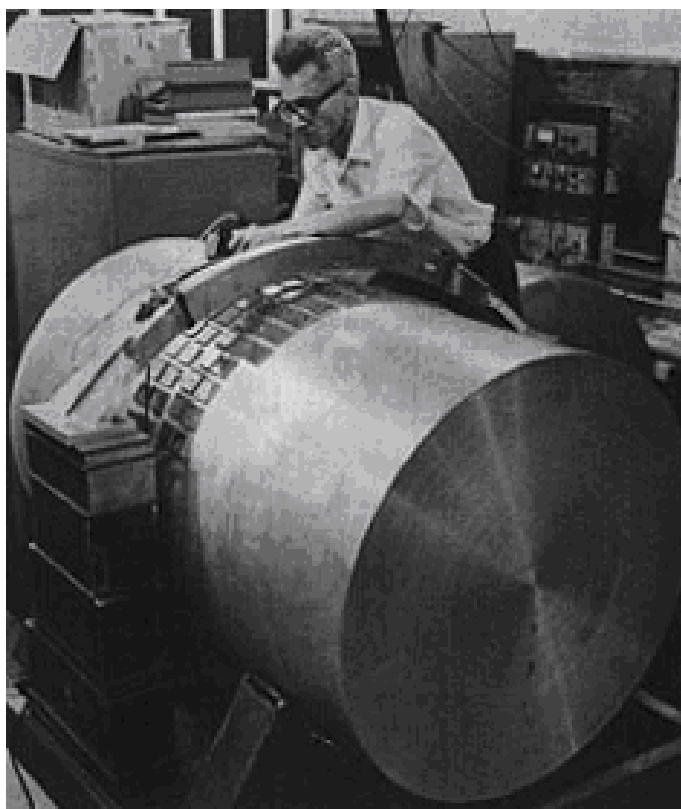


Figura 2.1 - Joseph Weber trabalhando em seu detector de ondas gravitacionais

Atualmente os detectores deste tipo sofreram grandes mudanças, como o desenvolvimento de suspensões melhores que diminuíram a transmissão das vibrações externas, as barras são mantidas a alguns Kelvin de temperatura para diminuir o ruído térmico, e foi introduzido um transdutor ressonante que amplifica as vibrações do modo fundamental da barra. Essas mudanças serviram para diminuir o ruído e amplificar o sinal.

Porém, a sensibilidade dos detectores de barras ressonantes varia também com a direção da fonte. Se a direção da fonte e o eixo do cilindro estiverem separados por um ângulo  $\theta$ , a sensibilidade do detector varia com  $\sin^2\theta$ . Por outro lado se  $\phi$  for o ângulo de polarização no plano da frente de onda medido em relação à projeção do eixo do detector, a sensibilidade à polarização é dada por  $\cos^2\phi$ . Desta forma a orientação do detector em relação à fonte que se pretende observar deve ser cuidadosamente ponderada de forma a maximizar o sinal recebido no detector.

Um dos detectores ressonantes, em funcionamento, mais sensíveis é o AU-

RIGA, um instrumento localizado na Itália, no INFN no Laboratório nacional de Legnaro. A primeira fase de operação que terminou em 1999 conseguiu atingir sensibilidades da ordem de  $10^{-21} Hz^{-\frac{1}{2}}$ .

No entanto a determinação da direção da fonte não é possível quando se utiliza cilindros. Desta forma, existe uma classe de detectores ressonantes, que são os detectores esféricos. Se vários (mais precisamente 6, obedecendo a configuração das faces não opostas de um dodecaedro) transdutores forem colocados na superfície de um detector esférico, poderemos não apenas determinar a direção da fonte, mas também o polarização da mesma (AGUIAR, 2004).

Na Universidade de São Paulo, existe um detector (esférico) de ondas gravitacionais, o Detector Mario Schenberg, que busca tais evidências. Veremos mais detalhes sobre este detector na seção 2.4, o único detector de ondas gravitacionais da América Latina.

**Detectores Interferométricos** - Quando uma onda gravitacional interage com os dois braços perpendiculares de um interferometro, vai induzir pequenas variações no seu comprimento que podem ser medidas através do padrão de interferência resultante da recombinação da luz que se propaga no instrumento. Para que o detector tenha uma sensibilidade da ordem de  $10^{-21} Hz^{-\frac{1}{2}}$ , os braços do interferômetro devem ser da ordem de alguns quilômetros. Ao contrário dos detectores ressonantes, a faixa de frequências em que um detector interferométrico é sensível é bem maior (SCHUTZ, 1999).

No entanto, temos um problema, pois o tempo que a luz demora a percorrer o braço do interferômetro é muito menor que o período das ondas a que este método é sensível. Portanto, para conseguir maximizar a variação do comprimento nos braços do detector, foram desenvolvidas técnicas que permitem que a luz circule algumas dezenas de vezes no instrumento antes que o padrão de interferência seja medido. Essa técnica contribui para o aumento do ruído.

O detector mais interessante é o LIGO nos EUA. Este detector é constituído por três interferômetros: dois em Hanford no estado de Washington com braços de 4 e 2 Km e um terceiro em Livingston na Louisiana com braços de 4 Km. A melhor sensibilidade alcançada pelo LIGO é de  $10^{-23} Hz^{-\frac{1}{2}}$  (em torno de 100-200Hz).

O projeto inicial do LISA (Laser Interferometer Space Antenna) era constituído por três satélites dispostos nos vértices de um triângulo equilátero numa órbita em torno do Sol, com aproximadamente o mesmo raio da órbita da Terra, 20 graus atrás da órbita da Terra e num plano inclinado 60 graus em relação ao plano eclíptico (Fig. 2.2). Para um detector localizado na Terra é quase impensável melhorar a sensibilidade até níveis aceitáveis para frequências abaixo de algumas dezenas de Hz. Sendo assim surgiu a idéia de se colocar um detector no espaço, onde o ruído de baixa frequência é quase inexistente (SCHUTZ, 1999).

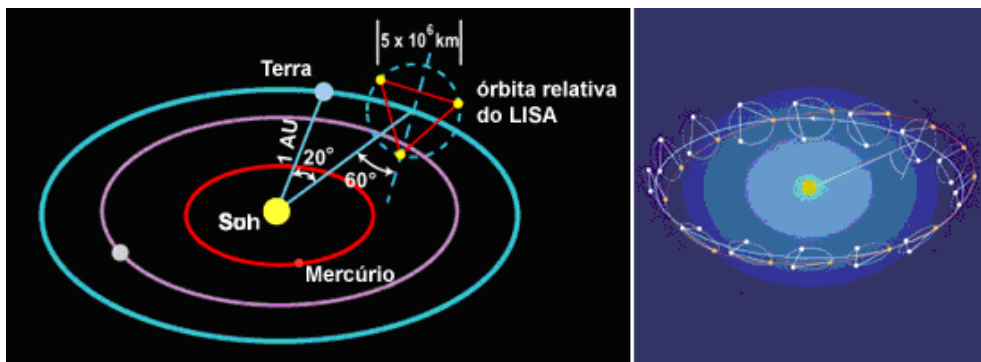


Figura 2.2 - A posição do LISA em relação às órbitas da Terra e do Sol (esquerda). Órbitas da Terra e do LISA em torno do Sol.  
 Fonte: LISA (2010)

O projeto original do LISA teria uma sensibilidade da ordem de  $10^{-23} \text{Hz}^{-\frac{1}{2}}$  para uma banda de frequências de 0.1 mHz a 0.1 Hz. Este poderá ser o primeiro instrumento capaz de fazer medições nesta região do espectro de ondas gravitacionais. Nesta banda de frequências conseguiremos observar a contribuição para o fundo estocástico de ondas gravitacionais de sistemas binários galácticos e a coalescência de buracos negros gigantes que hoje sabemos habitarem os centros de um grande número de galáxias, para além da possibilidade sempre presente de detectar o fundo cosmológico de ondas gravitacionais. Com a sensibilidade prevista, o LISA será capaz de detectar acontecimentos deste tipo com uma razão sinal ruído de 1000 para objectos com desvios para o vermelho até 1.

Como sabemos, um sistema binário é um forte candidato a emissão de ondas gravitacionais (Fig. 2.3), essa emissão causa um decréscimo da órbita do sistema.

Veremos, na seção 2.2, a prova indireta de OGs feita por Hulse e Taylor. Como sabemos, a Relatividade Geral de Einstein prevê a existência de OGs, então na seção 2.3 perturbamos a equação de campo e obtemos a equação de onda de modo a ilustrar essa previsão e introduzir um método de perturbação que será utilizado na obtenção dos MQNs.

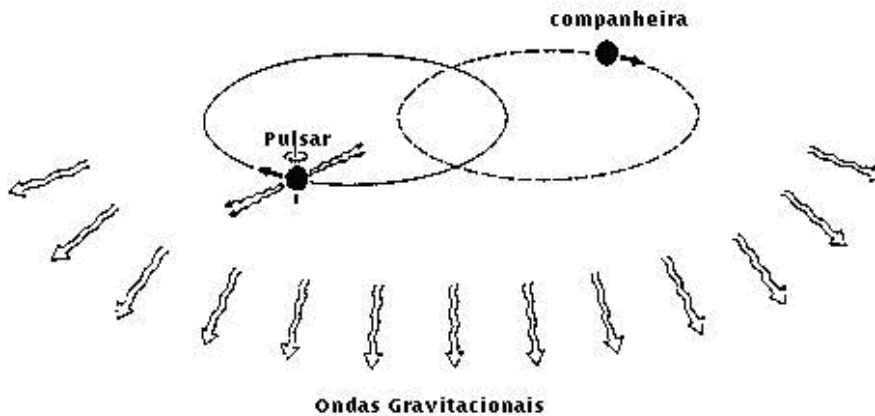


Figura 2.3 - Pulsar Binário emitindo OGs.

## 2.2 Prova indireta da observação de OGs (o PSR1913+16)

O pulsar PSR1913+16 foi descoberto em 1975 por Allan Russell e Joseph Hooton Taylor Jr., ambos da Universidade de Princeton. Tal descoberta abriu novas possibilidades para estudos de ondas gravitacionais, dando a Hulse e Taylor o prêmio Nobel em Física de 1993.

Hulse e Taylor, utilizando a antena de Arecibo de 305m, detectaram emissões pulsantes na banda de rádio em um sistema binário formado por duas ENs, ambas com massas aproximadamente iguais a  $1.4M_{Sol}$  orbitando seu centro de massa comum. Estas emissões eram de uma das estrelas, altamente magnetizada e com rotação de 17 vezes por segundo, possuindo assim um periodo de 59 ms (WEISBERG J. M., 2004).

Cada estrela se move segundo as Leis de Kepler e ambas encontram-se em posições opostas a uma linha que passa pelo centro de massa. Como mostrado na Fig. 2.4, as órbitas das estrelas são excêntricas e o máximo de afastamento entre

as estrelas se dá quando estas estão no apoastro ( $4.8R_{Sol}$ ) e o mínimo no periastro ( $1.1R_{Sol}$ ). O período orbital do sistema determinado em 1974 era de 7.75 horas.

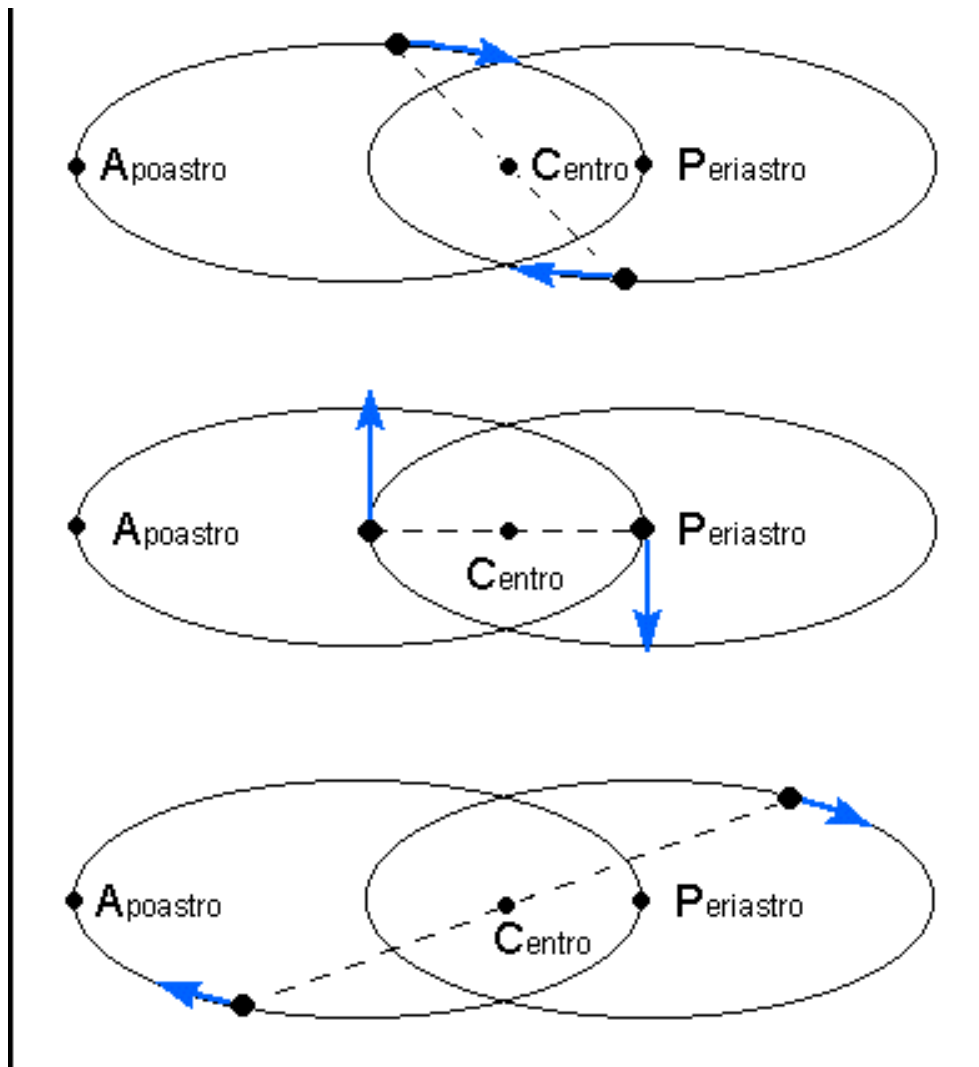


Figura 2.4 - Órbita de um Pulsar Binário  
Fonte: Adaptada de Weisberg et al. (1981)

Como a órbita é elíptica, a velocidade radial varia entre 75km/s (apoastro) e 300km/s (periastro) e a distância entre as estrelas também varia bastante. Desta forma, o campo gravitacional se torna grande no periastro e pequeno no apoastro. Este cenário é muito bom para testar a Teoria da Relatividade de Einstein. A Teoria da RG prediz que o sistema irá perder energia orbital através da emissão de

ondas gravitacionais.

Em 1983, Taylor e seus colaboradores observaram que a hora de periastro havia mudado, conseqüentemente, a órbita não tinha se mantido constante desde a descoberta. A figura 2.5 mostra o resultado de trinta anos de observações, onde vemos a mudança na época do periastro. O decaimento da órbita observado (pontos) é comparado com a previsão feita pela RG (linha) (WEISBERG J. M., 2004).

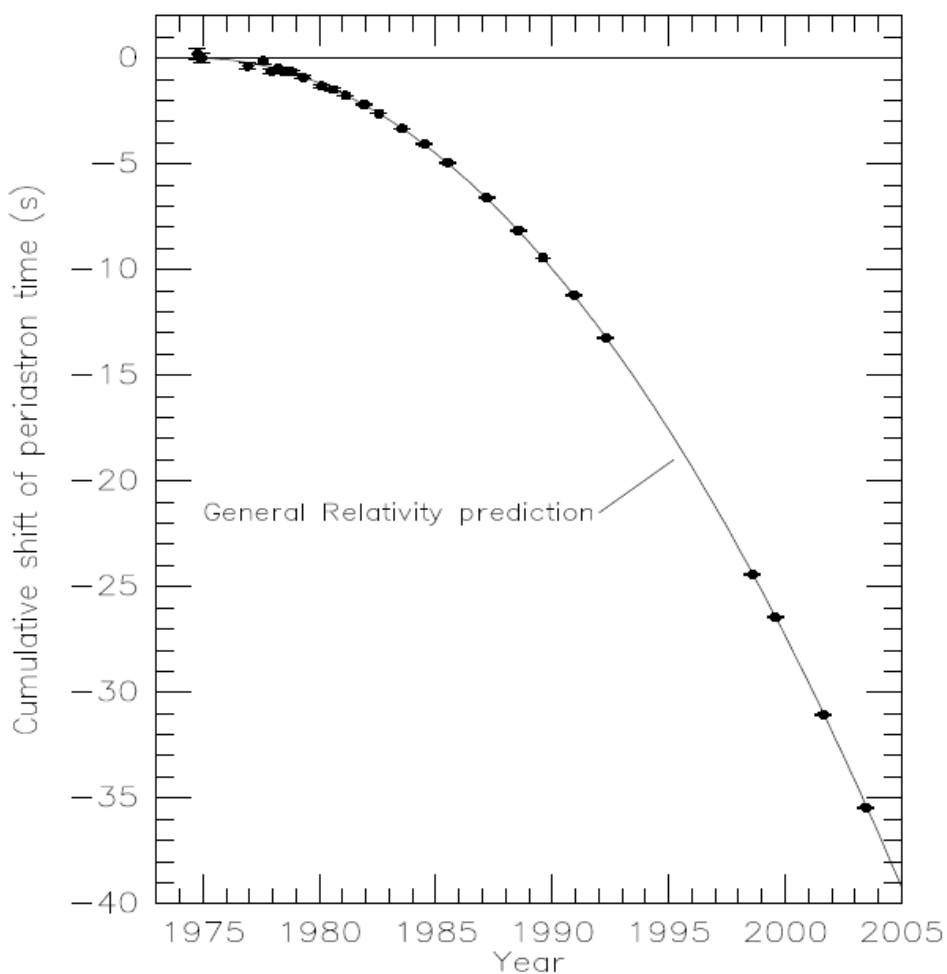


Figura 2.5 - Mudança na hora do Periastro  
Fonte: Figura de Weisberg e Taylor (2004)

O sistema binário vai perdendo energia e sua órbita vai encolhendo até que

as estrelas coalescem. Esse acontecimento pode gerar ondas gravitacionais fortes o suficiente para serem detectadas.

### 2.3 Relatividade Geral e as Ondas Gravitacionais

Podemos ilustrar rapidamente a previsão teórica da RG de ondas gravitacionais. Vamos perturbar a métrica na equação de campo de Einstein, ressaltando que o processo que será realizado na obtenção dos modos quasinormais será muito semelhante a este, no entanto iremos também perturbar o tensor energia-momentum que se encontra no lado direito da equação de campo.

Como sabemos a RG prevê a existência de OGs. Como as OGs são perturbações do espaço-tempo, vamos perturbar a métrica e analisar a equação de campo de Einstein (SCHUTZ, 1985), a qual é dada por

$$R_{\mu\nu} - \frac{1}{2}g_{\mu\nu}R = \frac{8\pi G}{c^4}T_{\mu\nu}. \quad (2.1)$$

Vamos obter a solução para campos fracos, assumindo a métrica perturbada como a soma de uma métrica de fundo não perturbada  $\eta_{\mu\nu}$  e uma perturbação  $h_{\mu\nu}$ , assim

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu}. \quad (2.2)$$

Sendo  $h^{\mu\nu} = \eta^{\mu\alpha}\eta^{\beta\nu}h_{\alpha\beta}$ , temos que  $g^{\mu\nu} = \eta^{\mu\nu} - h^{\mu\nu} + O(h^2)$ . Tomando apenas os termos em primeira ordem em  $h$ ,  $g^{\mu\nu}$  é dado por,

$$g^{\mu\nu} = \eta^{\mu\nu} - h^{\mu\nu}. \quad (2.3)$$

Utilizando a métrica perturbada, o tensor de Ricci pode ser escrito da seguinte forma

$$R_{\mu\nu} = -\frac{1}{2}\eta^{\sigma\lambda}h_{\mu\nu,\lambda\sigma} - \frac{1}{2}(h_{,\mu\nu} - h^\beta{}_{\nu,\mu\beta} - h_\mu{}^\beta{}_{,\beta\nu}). \quad (2.4)$$



Definimos o calibre de Lorentz como

$$\bar{h}_{,\nu}^{\mu\nu} = 0. \quad (2.5)$$

Onde  $\bar{h}^{\mu\nu}$  é chamado de traço-reverso, e é definido como

$$\bar{h}^{\mu\nu} = h^{\mu\nu} - \frac{1}{2}\eta^{\mu\nu}\eta^{\alpha\beta}h_{\alpha\beta} = h^{\mu\nu} - \frac{1}{2}\eta^{\mu\nu}h. \quad (2.6)$$

Utilizando este calibre, podemos reescrever o tensor de Ricci, dado pela equação 2.4, da seguinte forma

$$R_{\mu\nu} = -\frac{1}{2}\eta^{\sigma\lambda}h_{\mu\nu,\sigma\lambda}. \quad (2.7)$$

O escalar de Ricci ou escalar de curvatura, será

$$R = -\frac{1}{2}\eta^{\sigma\lambda}h_{,\sigma\lambda}. \quad (2.8)$$

Substituindo 2.7 e 2.8 em 2.1 teremos

$$\eta^{\sigma\lambda}\bar{h}_{,\sigma\lambda}^{\mu\nu} = -\frac{8\pi G}{c^4}T^{\mu\nu}. \quad (2.9)$$

ou ainda,

$$\square\bar{h}^{\mu\nu} = -\frac{8\pi G}{c^4}T^{\mu\nu}. \quad (2.10)$$

Onde  $\square$  é o d'Alembertiano dado por

$$\square = -\frac{\partial^2}{\partial t^2} - \nabla^2. \quad (2.11)$$

No vácuo, a equação de onda se torna:

$$\square \bar{h}^{\mu\nu} = 0. \quad (2.12)$$

## 2.4 O Detector de Ondas Gravitacionais Mario Schenberg

Localizado no instituto de Física da USP, está o detector de ondas gravitacionais brasileiro, Mario Schenberg. O seu nome dado em homenagem ao físico brasileiro Mario Schenberg, que foi pioneiro em física teórica e astrofísica moderna no País. Ele também foi professor do Instituto de Física da USP.

O detector começou a ser construído em 2000 e é fruto do Projeto Gráviton, projeto que conta com uma parceria entre pesquisadores do Instituto de Física da USP, do Inpe, da Universidade de Leiden, do Instituto Federal de Educação Tecnológica de São Paulo (IFSP), do Instituto Tecnológico de Aeronáutica (ITA), da Unifesp - Campus Diadema, entre outros.

As ondas gravitacionais são captadas/detectadas pela esfera metálica (Fig. 2.6 ) de quase 1 tonelada composta por cobre e 6% de alumínio. O fato do detector ser esférico destaca dos demais detectores, pois pode determinar a forma da onda no espaço (polarização) e a direção de onde ela vem.

A onda gravitacional passa pela esfera, fazendo esta oscilar. Essa oscilação é captada por sensores na superfície. Na fase atual, o detector conta com 6 sensores, que possibilitará a determinação não só da forma da onda, mas a sua direção de origem. Depois de captada, o sinal mecânico é transformado em um sinal elétrico, o qual pode ser enviado para um computador e analisado (AGUIAR, 2004).

Um grande problema é o fato do sinal ser muito fraco, sendo necessário tomar certos cuidados, como por exemplo, suspensão da esfera para que esta não tenha contato com as paredes do aparelho, evitando-se, desta forma, também que esta perceba a vibração sísmica da Terra, isolando-a acusticamente e resfriando-a para minimizar o ruído térmico.

A banda de frequências do detector é de 100Hz em torno de 3200Hz (AGUIAR, 2004). Embora a frequência do modo quasinormal fundamental dependa da equação de estado, acreditamos que essa esteja dentro da banda do Schenberg, sendo assim, será possível observar essa frequência no detector brasileiro.



Figura 2.6 - Detector Mario Schenberg.

No próximo capítulo vamos abordar o objeto astrofísico mais importante no nosso estudo: as estrelas de nêutrons.



### 3 ESTRELAS DE NÊUTRONS

As estrelas de nêutrons são objetos astrofísicos extremamente compactos, resultado da evolução de estrelas com massas entre  $8M_{Sol}$  e  $20M_{Sol}$ , cujo colapso gravitacional, em seus estágios finais de evolução, só para com a pressão de degenerescência exercida pelos nêutrons.

As estrelas de nêutrons foram observadas diretamente apenas em 1967 por Jocelyn Bell, estudante de doutorado da Universidade de Cambridge, que estava testando uma nova técnica em radioastronomia proposta por seu orientador Anthony Hewish. Mas já em 1932, ano em que Sir James Chadwick descobriu a existência do nêutron, acredita-se que Lev Landau e sua equipe especularam a existência de estrelas compostas basicamente de nêutrons. Jocelyn Bell fez a primeira observação de uma estrela de nêutrons analisando os pulsos regulares de um pulsar de rádio.

No entanto, uma Anã Branca tem sua contração gravitacional contrabalançada pela pressão de degenerescência dos elétrons. Em 1934, Baade e Zwicky, dois astrofísicos americanos, sabendo que em um corpo estelar com massa acima do limite de Chandrasekhar essa pressão de degenerescência dos elétrons não é suficiente para parar o colapso gravitacional, sugeriram que nêutrons poderiam ser capazes de exercer uma força capaz de parar o colapso. Em 1939, Robert Oppenheimer e G. Volkoff desenvolveram uma teoria para substanciar a teoria de Baade e Zwicky.

Quando a pressão de degenerescência dos elétrons não é suficiente para se opor à força gravitacional, a estrela começa a se contrair novamente, buscando minimizar sua energia. O processo mais conhecido que para o colapso é chamado neutronização, que acontece quando o núcleo captura um elétron que transforma prótons em nêutrons, este processo emite neutrinos e pode ser mostrado usando o decaimento beta inverso

$$p + e^{-} = n + \nu_e. \quad (3.1)$$

Uma estrela de nêutrons padrão possui um raio em torno de 10Km e uma densidade média da ordem de  $1 \times 10^{14} g/cm^3$  (100 milhões de toneladas por centímetro cúbico), com campo magnético e momento angular gigantescos. Sendo assim, uma

estrela como o Sol, que possui um campo magnético de 1 gauss, ao se contrair até o tamanho de uma estrela de nêutrons, passaria a ter um campo magnético de  $1 \times 10^{12} \text{gauss}$  (1 trilhão de Gauss) ou até mais.

### 3.1 Estrelas de nêutrons e as ondas gravitacionais

De acordo com a teoria da relatividade geral, um objeto assimétrico em rotação produz ondas gravitacionais. Uma pequena assimetria de massa  $m$  na superfície de uma estrela de massa  $M$ , raio  $R$  com uma frequência de rotação  $f_r$  a uma distância  $r$  do observador vai produzir ondas gravitacionais com uma amplitude

$$h \approx 8\pi \frac{R f_r m}{r}. \quad (3.2)$$

e frequência

$$f = 2f_r \quad (3.3)$$

O fator 2 na frequência se deve ao fato de que a assimetria aparece alternadamente nos dois lados do eixo de rotação. Utilizamos unidades naturais nas equações, onde  $c = G = M_{Sol} = 1$ .

Se assumirmos a energia radiada como a conversão da energia rotacional, a observação de ondas gravitacionais fornece uma pista para a compreensão da dinâmica da estrela.

### 3.2 Equação de Tolman-Oppenheimer-Volkoff

A equação que mostraremos aqui foi derivada em 1939 por Richard Chase Tolman, Julius Robert Oppenheimer e George Michael Volkoff, e se reverte à forma usual da Equação do Equilíbrio Hidrostático para um corpo esfericamente simétrico. Para simetria esférica, o elemento de linha é dado por

$$ds^2 = e^\nu dt^2 - e^\lambda dr^2 - r^2(d\theta^2 + \sin^2 \theta d\phi^2). \quad (3.4)$$

Com isso, a métrica é

$$g_{\mu\nu} = \text{diag}(e^\nu, -e^\lambda, -r^2, -r^2 \sin^2 \theta) \quad (3.5)$$

As componentes não nulas do tensor de Ricci são

$$R_{00} = \left[ -\frac{\nu''}{2} + \frac{\lambda'\nu'}{4} - \frac{(\nu')^2}{4} - \frac{\nu'}{r} \right] e^{\nu-\lambda}, \quad (3.6)$$

$$R_{11} = -\frac{\nu''}{2} - \frac{\lambda'\nu'}{4} + \frac{(\nu')^2}{4} - \frac{\lambda'}{r}, \quad (3.7)$$

$$R_{22} = \left( 1 - \frac{r\nu'}{2} - \frac{r\lambda'}{2} \right) e^{-\lambda} - 1, \quad (3.8)$$

$$R_{33} = R_{22} \sin^2 \theta. \quad (3.9)$$

O escalar de Ricci (ou curvatura de Riemann) é

$$R = e^{-\lambda} \left[ -\nu'' + \frac{1}{2}\lambda'\nu' - \frac{1}{2}(\nu')^2 - \frac{2}{r^2} + \frac{2\lambda'}{r} - \frac{2\nu'}{r} \right] + \frac{2}{r}. \quad (3.10)$$

Para um fluido perfeito, o tensor energia-momento é dado por

$$T_{\mu\nu} = (\rho + p)u_\mu u_\nu - pg_{\mu\nu}. \quad (3.11)$$

Onde  $\rho$  é a densidade de energia total,  $p$  a pressão e  $u_\mu$  a quadrivelocidade do fluido.

As componentes não nulas do tensor energia-momento no referencial comóvel com o fluido são dadas por

$$T_{00} = \rho, T_{11} = T_{22} = T_{33} = -p. \quad (3.12)$$

Levando as componentes não nulas do tensor energia-momento (eq. 3.12), o escalar de Ricci (eq. 3.10), as componentes não nulas do tensor de Ricci (eqs. 3.6 a 3.9) e a métrica (eq. 3.4) na equação de Einstein (eq. 2.1) é possível encontrar as seguintes equações diferenciais,

$$4\pi p = e^{-\lambda} \left( \frac{\nu'}{r} + \frac{1}{r^2} \right) - \frac{1}{r^2}, \quad (3.13)$$

$$8\pi p = \frac{1}{r} e^{-\lambda} \left[ \nu'' + \frac{1}{2}(\nu')^2 + \frac{\nu' - \lambda'}{r} - \frac{\nu'\lambda'}{2} \right], \quad (3.14)$$

$$4\pi \rho = e^{-\lambda} \left( \frac{\lambda'}{r} - \frac{1}{r^2} \right) - \frac{1}{r^2}. \quad (3.15)$$

Utilizando a derivada da equação 3.13 em relação a  $r$ , e usando as equações 3.13, 3.14 e 3.15 podemos chegar na equação de TOV para o equilíbrio hidrostático, onde

$$\frac{dp(r)}{dr} = \frac{m(r)}{r^2} \rho \left[ 1 + \frac{p(r)}{\rho} \right] \left[ 1 + \frac{4\pi r^3 p(r)}{m(r)} \right] \left[ 1 - \frac{2m(r)}{r} \right]^{-1}. \quad (3.16)$$

Além da equação de TOV, a equação de massa (eq. 3.17), o potencial gravitacional  $\nu(r)$  (eq. 3.18) e a equação de estado são necessários para descrever uma estrela de nêutrons em equilíbrio hidrostático.

$$\frac{dm(r)}{dr} = 4\pi r^2 \rho. \quad (3.17)$$

$$\frac{d\nu(r)}{dr} = -2(\rho + p) \frac{dp(r)}{dr} \quad (3.18)$$

### 3.3 Estrutura Interna das Estrelas de Nêutrons

Para resolver as equações 3.16, 3.17 e 3.18, desenvolvemos um programa na linguagem Fortran que implementa Runge-Kutta de 4ª ordem. No entanto, é necessário adotar uma equação de estado. Adotamos uma equação de estado politrópica com relação pressão-densidade de energia da seguinte forma (TOOPER, 1965)

$$p = K \rho_b^{1+1/n}, \quad (3.19)$$



onde  $\rho_b$  é a densidade de energia dos bárions,  $K$  é a constante adiabática ( $K$  é uma constante de integração cujo valor depende do objeto em questão e é determinada através da pressão e densidade em um dado ponto) e

$$\rho = \rho_b + np \quad (3.20)$$

Para uma estrela de nêutrons com nêutrons não-relativísticos, temos  $n = 1(\Gamma = 2)$  e  $K = 100(5382m^4/kg^{2/3}s)$  (TOOPER, 1965).

Tendo em mãos as equações de TOV, a equação de estado e um método de integração, a última coisa, a saber, é o valor da densidade no centro da EN (pressão e massa são dados em função desta).

Os programas `tov_rot_z` (Anexo B), `tov_rot_z3` (Anexo B), `tov_rot_z4` (Anexo B) fazem basicamente as mesmas coisas, mas existem distintas rotinas que necessitam de determinados dados. Então, tivemos que fazer algumas alterações nelas e dar-lhes outros nomes. Essas rotinas inicialmente fazem uma varredura para determinar a superfície da estrela e assim obtermos o valor do raio  $R$ . A superfície da estrela é determinada quando a pressão se anula. Desta forma, quando o programa encontra uma pressão nula ou negativa, ele interrompe os cálculos e determina o raio exato da estrela. Com o valor exato de  $R$  é possível redefinir o passo em  $r$  e se obter a estrutura interna da EN. Como já foi comentado, para resolver as equações de TOV é utilizado um método numérico, o qual está contido na rotina `rk4sys` (Anexo B). Na rotina `xpsys` (Anexo B) estão escritas as equações de TOV.

Perto do centro não utilizamos o método Runge-Kutta. Com uma expansão em séries de potência e com os valores iniciais de  $p$ ,  $m$ ,  $\mu$  e  $\rho$  obtemos a estrutura da estrela nessa região. Utilizamos os últimos valores definidos pela expansão como valores iniciais para o método Runge Kutta, obtendo assim a estrutura interna da estrela. Resumindo, temos que, as rotinas `tov_rot` primeiramente possuem um laço que ajuda a determinar o  $R$ , com o  $R$  defini-se o passo em  $r$ , logo após com a série de potencias calcula-se a estrutura interna perto do centro e com o método Runge-Kutta o restante.

### 3.4 Resultados

Para “mostrar” a estrutura interna da estrela e a veracidade do programa geramos arquivos de dados para analisar a variação com o raio de fatores, tais como, pressão, massa, densidade de energia e potencial gravitacional. Na figura 3.1 temos a variação da pressão com o raio. Sabemos que a pressão é máxima no centro da estrela e zero na superfície, isso pode ser facilmente observado no gráfico. O raio que obtemos para a estrela de neutrôns politrópica foi de  $14.1km$ .

Na figura 3.2 temos a massa contida em um raio  $r$ , desta forma, a massa contida em um raio  $r = 0$  é zero e a massa  $m(R)$  (massa contida em um raio  $r = R$ ) é a massa total da estrela (massa  $M$ ), a qual foi de  $1.4M_s$ .

No gráfico da densidade de energia (fig. 3.3) podemos ver que ela tem seu máximo no centro da estrela ( $r = 0$ ) e zero na superfície da estrela ( $r = R$ ). Também temos para “ilustrar” o interior estelar, os gráfico do potencial gravitacional (fig. 3.4).

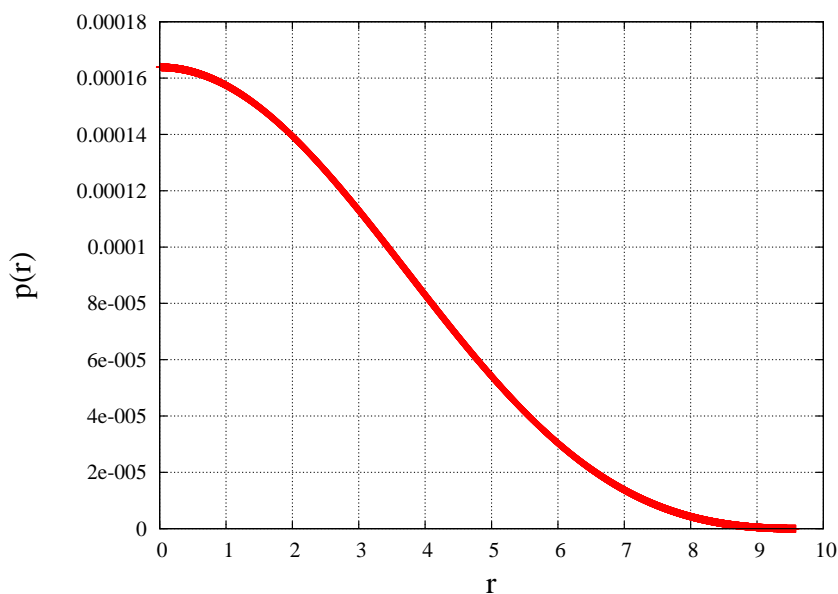


Figura 3.1 - Variação da pressão no interior estelar.

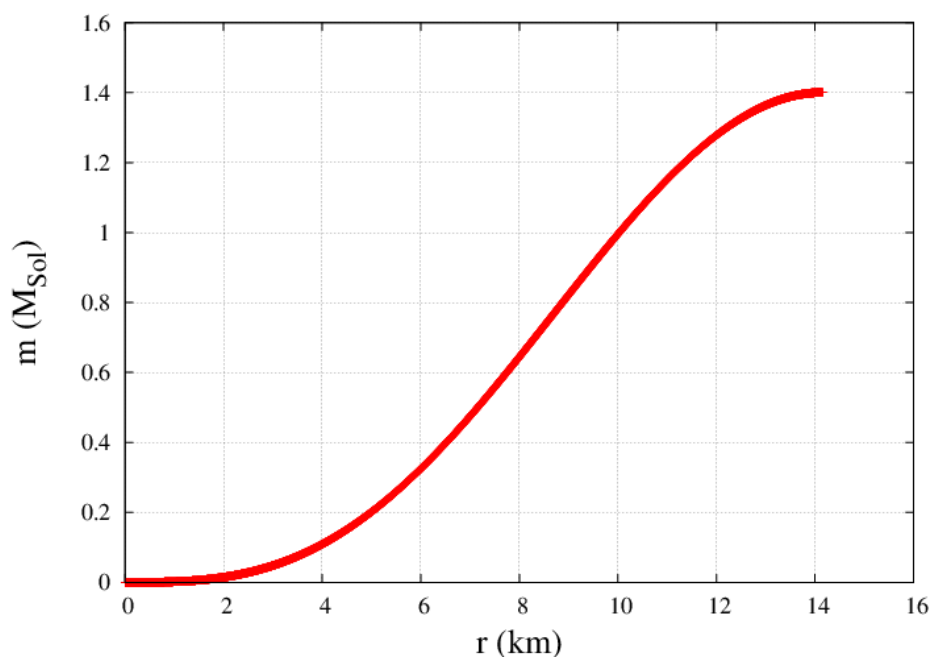


Figura 3.2 - Gráfico  $m(r)$  vs  $r$ .

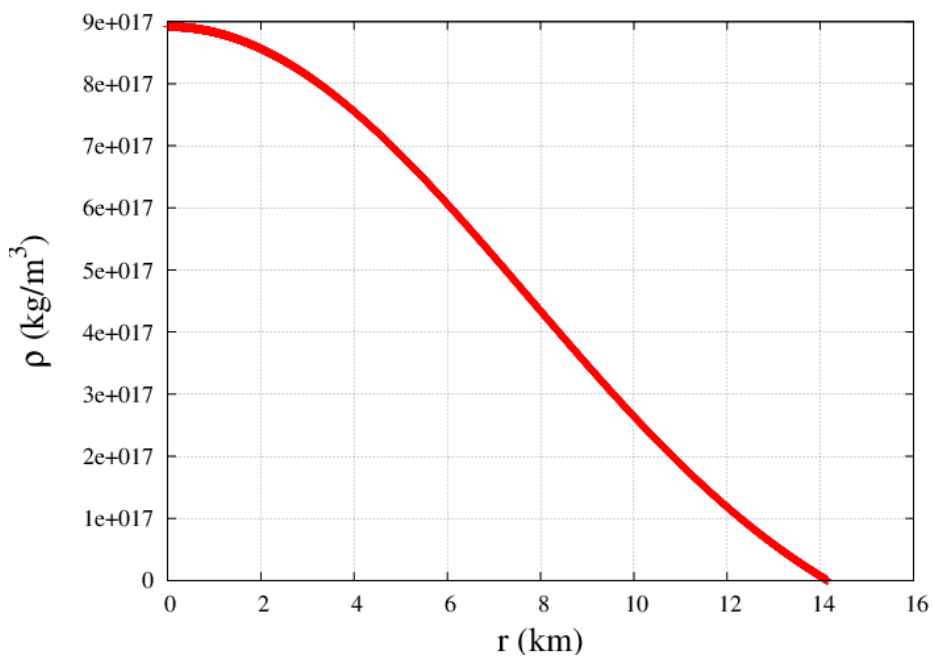


Figura 3.3 - Gráfico mostrando a variação da densidade de energia com o raio.

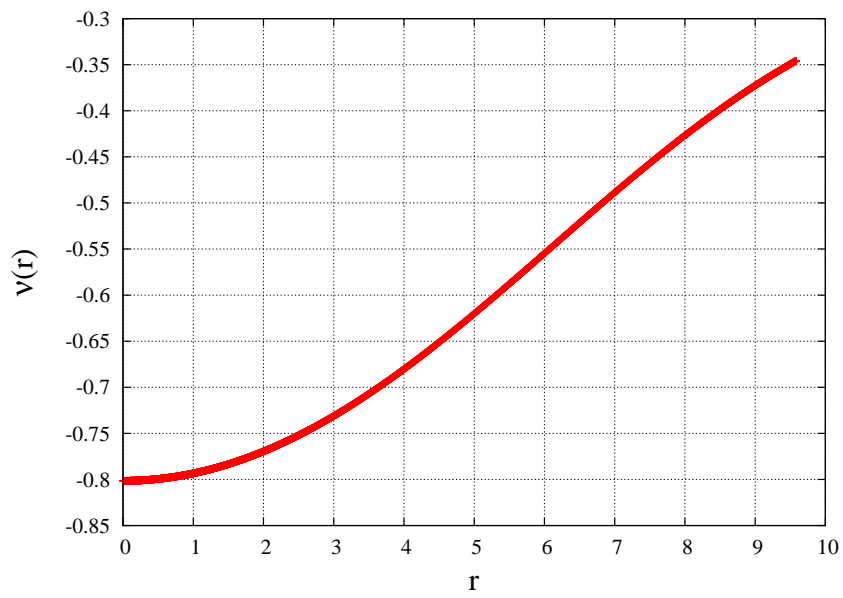


Figura 3.4 - Potencial gravitacional no interior estelar.

## 4 MODOS QUASINORMAIS

Quando perturbamos as equações de Einstein que descrevem uma estrela e o seu espaço-tempo, somamos uma pequena perturbação a todas as quantidades físicas relevantes, tais como densidade, pressão, métrica e quadrivelocidade do fluido. Desta forma:

$$f_0(t, r, \theta, \phi) \longrightarrow f(t, r, \theta, \phi) = f_0(t, r, \theta, \phi) + \delta f(t, r, \theta, \phi), \quad (4.1)$$

Onde,

$$\delta f_{L,m}(t, r, \theta, \phi) = e^{i\omega t} f(r) Y_{Lm}(\theta, \phi). \quad (4.2)$$

As frequências  $\omega$  são números complexos chamados de modos quasinormais, onde:

$$\omega = \omega_r + i\omega_i.$$

Temos 3 modos quasinormais distintos (para perturbações polares e estrelas de nêutrons sem rotação ou campo magnético):

- f-modes (fundamental)  $\Rightarrow$  possuem frequência proporcional à raiz quadrada da densidade média da estrela;

- p-modes (pressão)  $\Rightarrow$  possuem frequências superiores aos modos f e sua força restauradora é a pressão;

- g-modes (gravidade)  $\Rightarrow$  a força restauradora é a flutuação do campo gravitacional (empuxo). Possuem frequências inferiores as do modo fundamental.

### 4.1 Modo Quasinormal Fundamental

Para estrelas compactas, sem rotação, o modo f é o MQN polar e não-radial com momento angular de índice  $l=2$ . Como sabemos, é através desse modo que podemos determinar massa e raio da estrela compacta.

Um dos motivos para estudar especificamente o modo f é devido à sua baixa

frequência ( $\approx 1 - 3kHz$ ), desta forma, é mais fácil detectá-lo do que outros modos.

## 4.2 Cálculo dos Modos Quasinormais

Para o cálculo do modo quasinormal fundamental foi necessário seguir uma série de passos e conseqüentemente criar vários programas antes. Acima de tudo, a primeira coisa foi definir uma equação de estado e obter um modelo relativístico para uma EN, para isso foi necessário utilizar um método numérico para resolver as equações de TOV (Tolman-Oppenheimer-Volkoff) que descrevem uma estrela esfericamente simétrica em equilíbrio gravitacional. Na seção 3.3 obtivemos a estrutura interna da estrela, com isso, o próximo passo é perturbar a equação de Einstein para um fluido perfeito.

No modelo estelar relativístico, o tensor que descreve a métrica perturbada (perturbação com paridade polar) é dado por (LINDBLOM L., 1983)

$$ds^2 = -e^\nu(1 + \mu^l H_0 Y_m^l e^{i\omega t})dt^2 - 2i\omega\mu^l H_1 Y_m^l e^{i\omega t} dt dr + e^\lambda(1 - \mu^l H_0 Y_m^l e^{i\omega t})dr^2 + r^2(1 - \mu^l K Y_m^l e^{i\omega t})(d\theta^2 + \sin\theta d\phi^2). \quad (4.3)$$

As funções  $e^\nu$  e  $e^\lambda$  são componentes da métrica não perturbada,  $Y_m^l$  são harmônicos esféricos usuais e a dependência temporal é  $e^{i\omega t}$ .

A perturbação do fluido é descrita pelo vetor  $\xi^\alpha$  (deslocamento Lagrangeano), cujas componentes são

$$\xi^t = 0, \quad (4.4)$$

$$\xi^r = \mu^l r^{-1} e^{-\lambda/2} W Y_m^l e^{i\omega t}, \quad (4.5)$$

$$\xi^\theta = -\mu^l r^{-2} V \partial_\theta Y_m^l e^{i\omega t}, \quad (4.6)$$

$$\xi^\phi = -\nu^l (r \sin \theta)^{-2} V \partial_\phi Y_m^l e^{i\omega t}. \quad (4.7)$$

As funções de perturbação da métrica  $H_0$ ,  $H_1$  e  $K$ , assim como as funções de perturbação do fluido  $V$  e  $W$  são funções dependentes apenas de  $r$ . As 5 funções de perturbação não são todas independentes e devem satisfazer a relação algébrica abaixo (THORNE K. S. E CAMPOLATTARO, 1967), a qual é uma consequência da Equação de Einstein:

$$\begin{aligned} \left[ 4M + \frac{1}{2}(l+2)(l-1)r + 4\pi r^3 p \right] H_0 &= 8\pi r^3 e^{-\nu/2} X - \left[ \frac{1}{2}(l+1)(M + 4\pi r^3 p) - \omega^2 r^3 e^{-(\lambda+\nu)} \right] H_1, \\ &+ \left[ \frac{1}{2}(l+2)(l-1)r - \omega^2 r^3 e^{-\nu} - r^{-1} e^\lambda (M + 4\pi r^3 p)(3M - r + 4\pi r^3 p) \right] K. \end{aligned} \quad (4.8)$$

onde,

$$X = \omega^2 (\epsilon + p) e^{-\nu/2} V - \frac{p'}{r} e^{(\nu-\lambda)/2} W + \frac{1}{2} (\epsilon + p) e^{\nu/2} H_0, \quad (4.9)$$

a qual é a redução da ordem do sistema.

As perturbações no interior da estrela primeiramente foram obtidas em duas regiões distintas (fig. 4.1). Primeiramente calculamos as perturbações da superfície da estrela até um ponto médio entre o centro e a superfície ( $R/2$ ), depois do centro até esse mesmo ponto. Posteriormente obtemos uma única solução em todo o interior estelar.

#### 4.2.1 Perturbações de $\frac{R}{2}$ até a Superfície da estrela

As perturbações são um conjunto de 4 equações diferenciais acopladas (sistema regula de 4ª ordem) (LINDBLOM L., 1985), dado por

$$H_1' = -r^{-1} [l+1 + 2Me^\lambda r^{-1} + 4\pi r^2 e^\lambda (p-\rho)] H_1 + r^{-1} e^\lambda [H_0 + K - 16\pi (p+\rho) V], \quad (4.10)$$

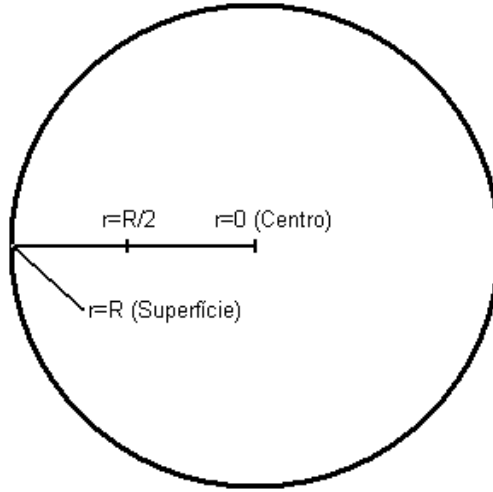


Figura 4.1 - Divisão do interior estelar para o cálculo das perturbações.

$$K' = r^{-1}H_0 + \frac{1}{2}l(l+1)r^{-1}H_1 - \left[ (l+1)r^{-1} - \frac{1}{2}\nu' \right] K - 8\pi(p+\rho)e^{\lambda/2}r^{-1}W, \quad (4.11)$$

$$W' = -(l+1)r^{-1}W + re^{\lambda/2} \left[ \gamma^{-1}p^{-1}e^{-\nu/2}X - l(l+1)r^{-2}V + \frac{1}{2}H_0 + K \right], \quad (4.12)$$

$$\begin{aligned} X' = & -lr^{-1}X + (p+\rho)e^{\lambda/2} \left\{ \frac{1}{2} \left( r^{-1} - \frac{1}{2}\nu' \right) H_0 + \frac{1}{2} \left[ r\omega^2e^{-\nu} + \frac{1}{2}l(l+1)r^{-1} \right] H_1 \right. \\ & + \frac{1}{2} \left( \frac{3}{2}\nu' - r^{-1} \right) K - \frac{1}{2}l(l+1)\nu'r^{-2}V - r^{-1} \left[ 4\pi(p+\rho)e^{\lambda/2} + \omega^2e^{\lambda/2-\nu} \right. \\ & \left. \left. - \frac{1}{2}r^2(r^{-2}e^{-\lambda/2}\nu')' \right] W \right\}. \end{aligned} \quad (4.13)$$

O programa que resolve numericamente essas equações é o `s_m_rot` (Anexo C). O cálculo começa na superfície e termina em  $\frac{R}{2}$ . Para obter as perturbações em um determinado  $r$ , é necessário saber para esse dado  $r$  o valor de pressão e sua



derivada, massa e sua derivada, densidade de energia, potencial e suas derivadas primeira e segunda e também o valor de  $\lambda$  e sua derivada.

Antes de implementar um método numérico e resolver as equações 4.10, 4.11, 4.12 e 4.13 é necessário saber as condições iniciais de  $H_1$ ,  $K$ ,  $W$  e  $X$ , ou seja, seus valores na superfície da estrela. Como temos 3 possíveis conjuntos de valores, teremos 3 soluções para as perturbações.

As condições iniciais podem ser (LINDBLOM L., 1983):

-	$H_1(R)$	$K(R)$	$W(R)$	$X(R)$
$Y_1(R)$	1	0	0	0
$Y_2(R)$	0	1	0	0
$Y_3(R)$	0	0	1	0

Tabela 4.1 - Condições Iniciais

Queremos que a perturbação lagrangeana da pressão se anule na superfície, assim temos que  $X(R) = 0$ . Como uma das quatro funções está fixa (função  $X(R)$ ), temos então 3 soluções linearmente independentes ( $Y_1, Y_2, Y_3$ ).

Com as condições iniciais e com os valores obtidos com `tov_rot_z4` (Anexo B), é possível resolver as equações 4.10, 4.11, 4.12 e 4.13. O método utilizado foi novamente o Runge-Kutta de 4ª ordem, onde a rotina `rk4sys` (Anexo C) foi adaptada e agora é a rotina `rk4sys_p1` e na rotina `xpsys_p1` (Anexo C) estão escritas as equações das derivadas de  $H_1$ ,  $K$ ,  $W$  e  $X$ .

Para cada condição inicial foi criado um laço e realizado o mesmo processo em todos. Vamos chamar as soluções de  $Y_1$ ,  $Y_2$  e  $Y_3$ , isso irá facilitar o entendimento do cálculo das perturbações do centro até  $\frac{R}{2}$  e também da “junção” das perturbações em uma única solução para todo o interior da estrela.

Após resolvermos numericamente as equações 4.10, 4.11, 4.12 e 4.13, geramos um arquivo de dados com o valor das perturbações para 5000 valores de  $r$  (o  $r$  variando de  $R/2$  a  $R$ ), com isso foi possível graficá-las. Nas figuras 4.2, 4.3, 4.4 e 4.5 temos as perturbações referentes à solução  $Y_1$ . Para verificar a consistência dos cálculos numéricos, plotamos a derivada das perturbações, as quais são mostradas

no Anexo A.

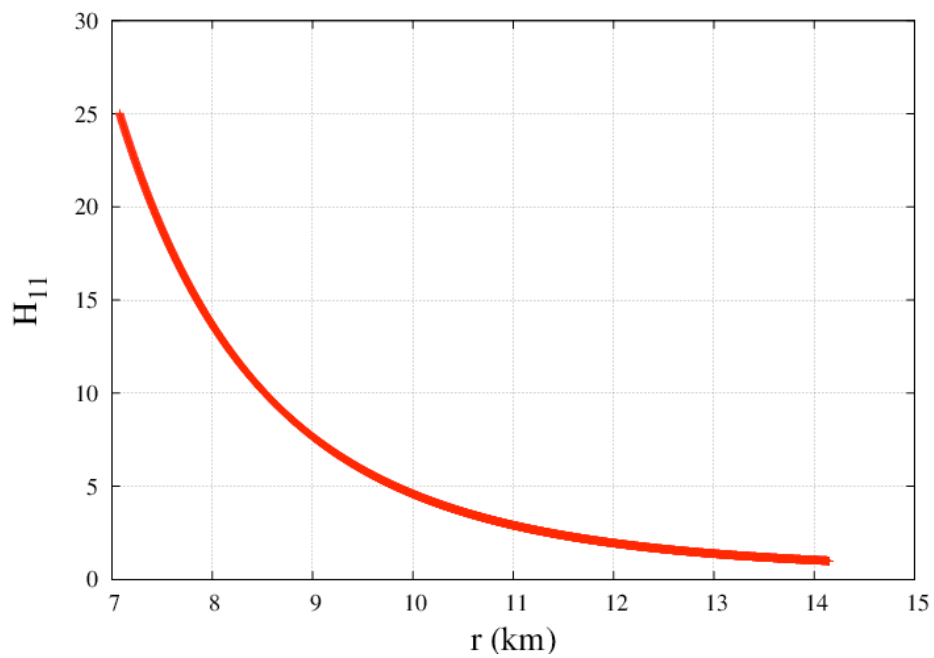


Figura 4.2 -  $H_{11}(r)$  vs  $r$ .

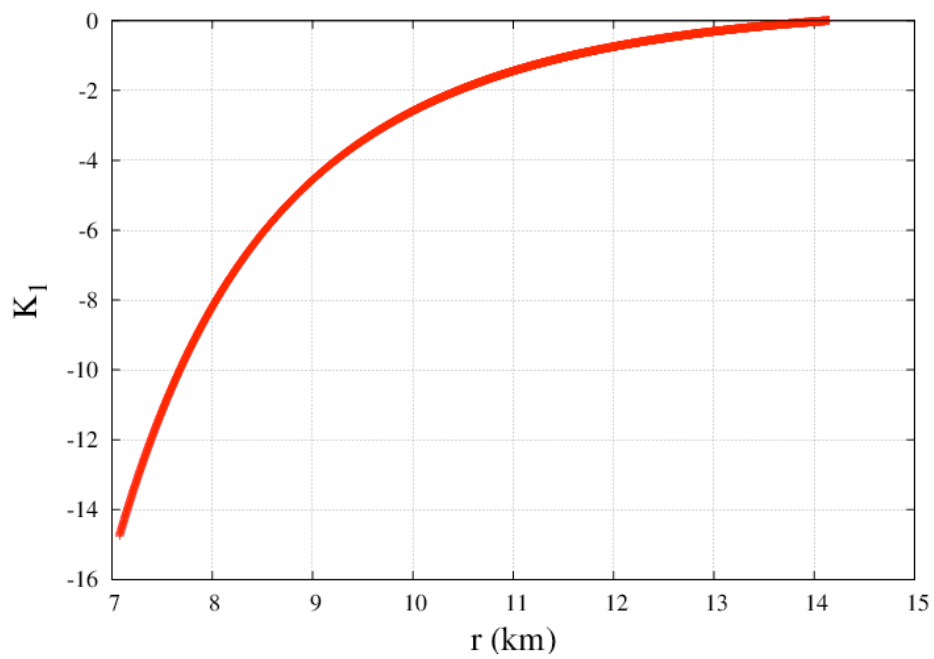


Figura 4.3 -  $K_1(r)$  vs  $r$ .

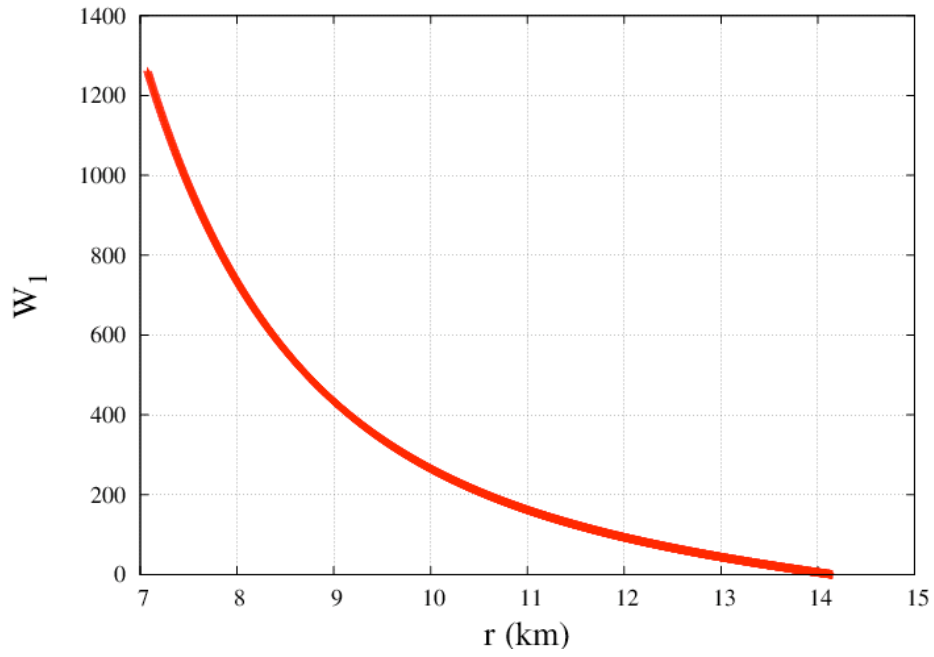


Figura 4.4 -  $W_1(r)$  vs  $r$ .

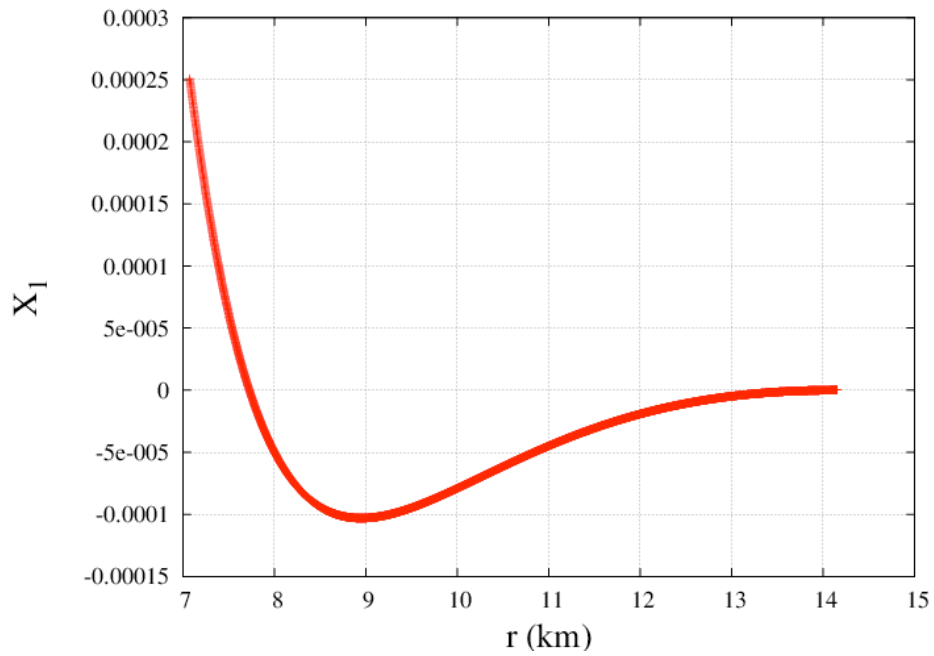


Figura 4.5 -  $X_1(r)$  vs  $r$ .

As figuras 4.6, 4.7, 4.8 e 4.9 são referentes a solução  $Y_2$  e as figuras 4.10, 4.11,

4.12 e 4.13 a  $Y_3$ . As derivadas das perturbações para essas 2 soluções, também são encontradas no ANEXO A.

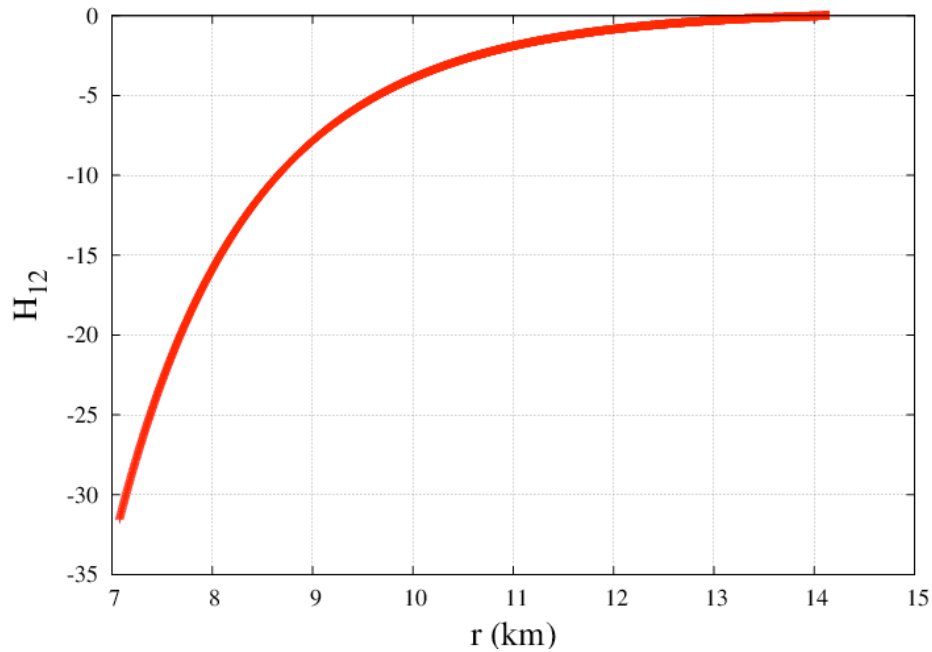


Figura 4.6 -  $H_{12}(r)$  vs  $r$ .

#### 4.2.2 Perturbações do Centro até $\frac{R}{2}$

Perto do centro não é possível apenas resolver as equações 4.10, 4.11, 4.12 e 4.13, pois o sistema é singular para  $r = 0$ . Então as perturbações perto do centro são obtidas através de uma expansão em séries de potência dadas por (LINDBLOM L., 1985),

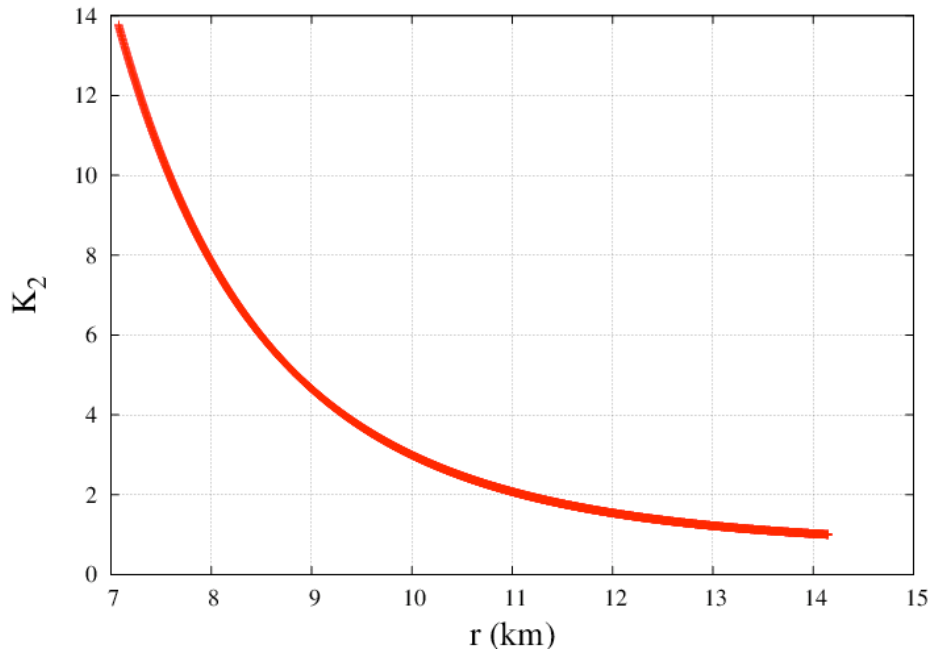


Figura 4.7 -  $K_2(r)$  vs  $r$ .

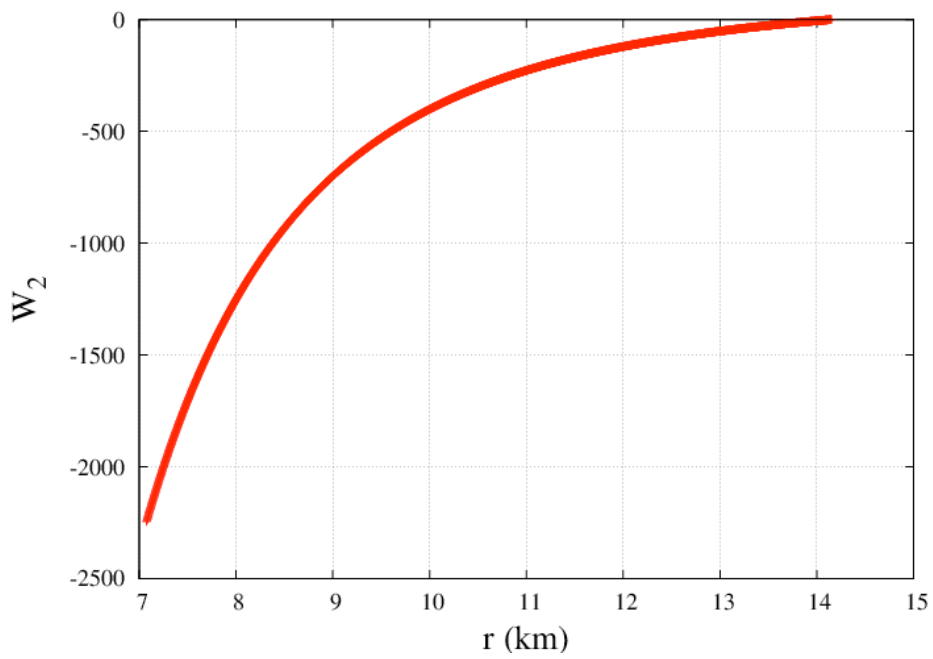


Figura 4.8 -  $W_2(r)$  vs  $r$ .

$$H_1(r) = H_1(0) + \frac{1}{2}r^2 H_1''(0) + O(r^4), \quad (4.14)$$

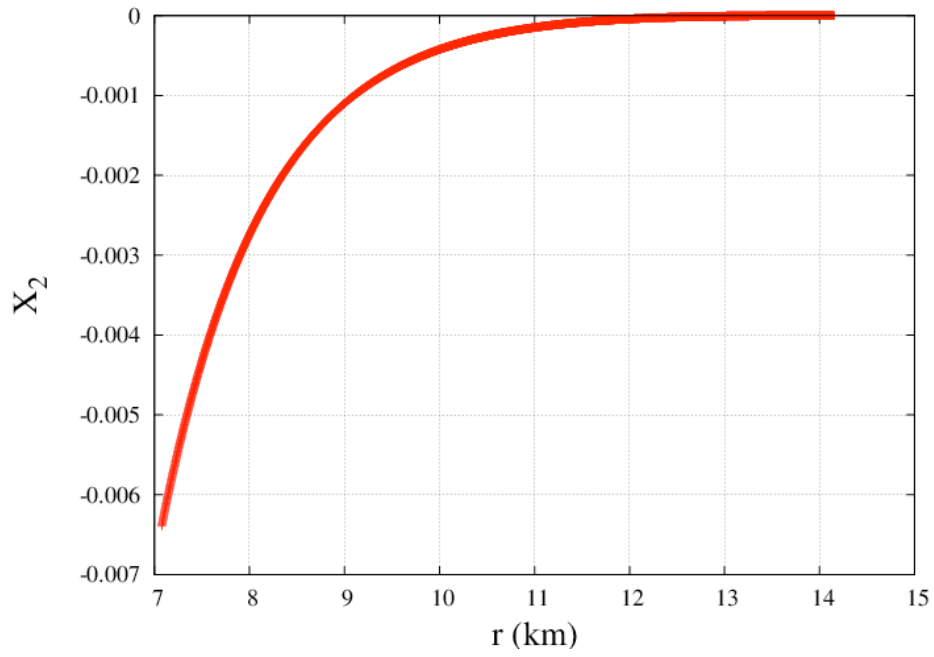


Figura 4.9 -  $X_2(r)$  vs  $r$ .

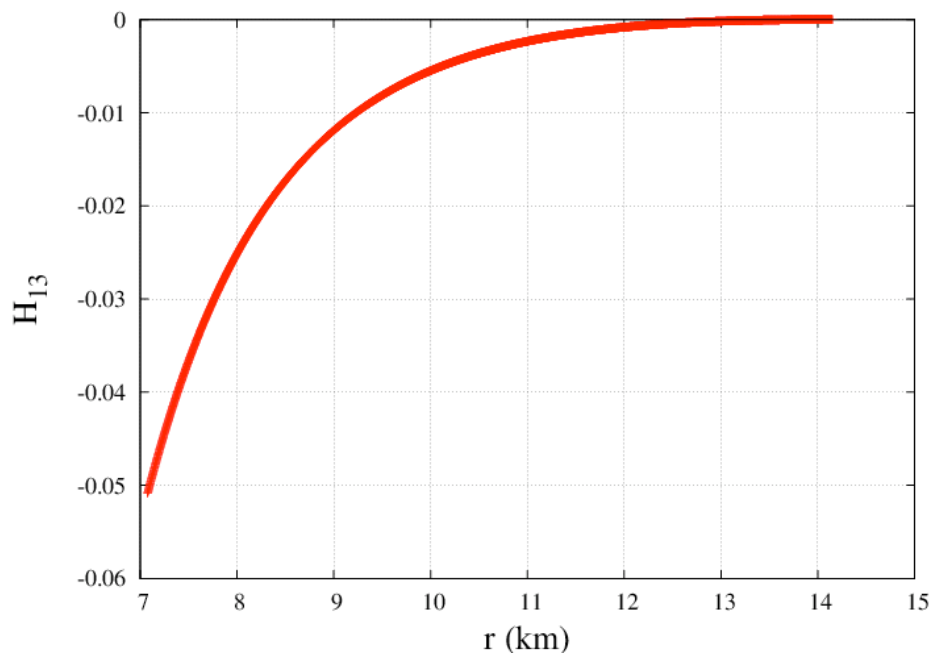


Figura 4.10 -  $H_{13}(r)$  vs  $r$ .

$$K(r) = K(0) + \frac{1}{2}r^2K''(0) + O(r^4), \quad (4.15)$$

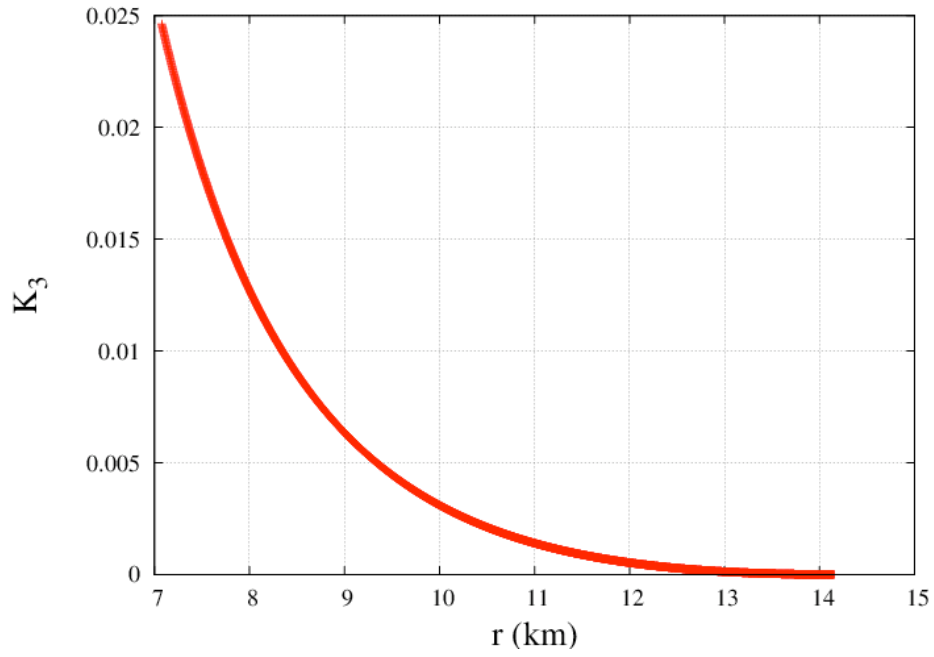


Figura 4.11 -  $K_3(r)$  vs  $r$ .

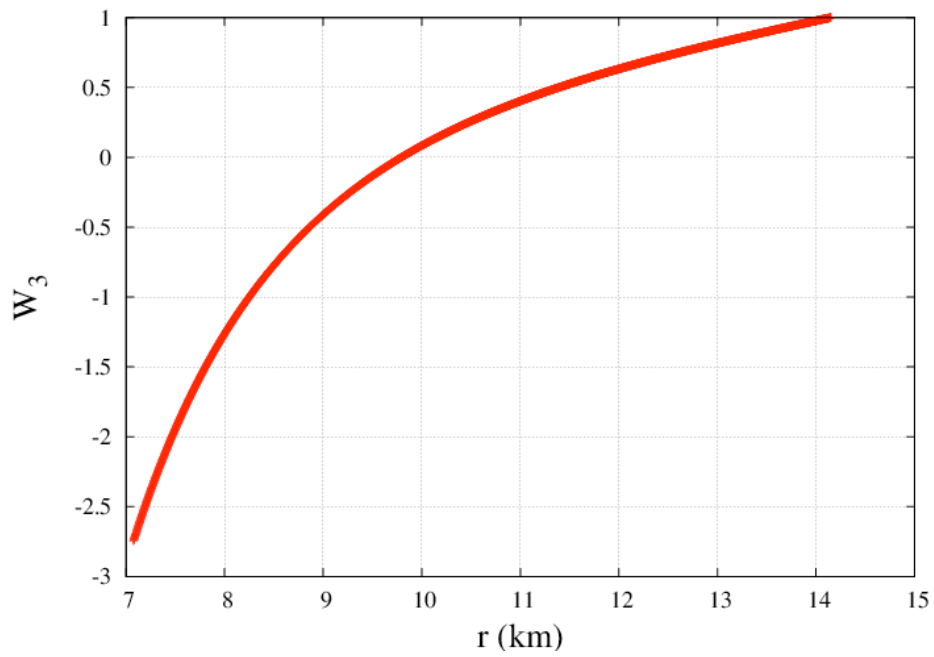


Figura 4.12 -  $W_3(r)$  vs  $r$ .

$$W(r) = W(0) + \frac{1}{2}r^2W''(0) + O(r^4), \quad (4.16)$$

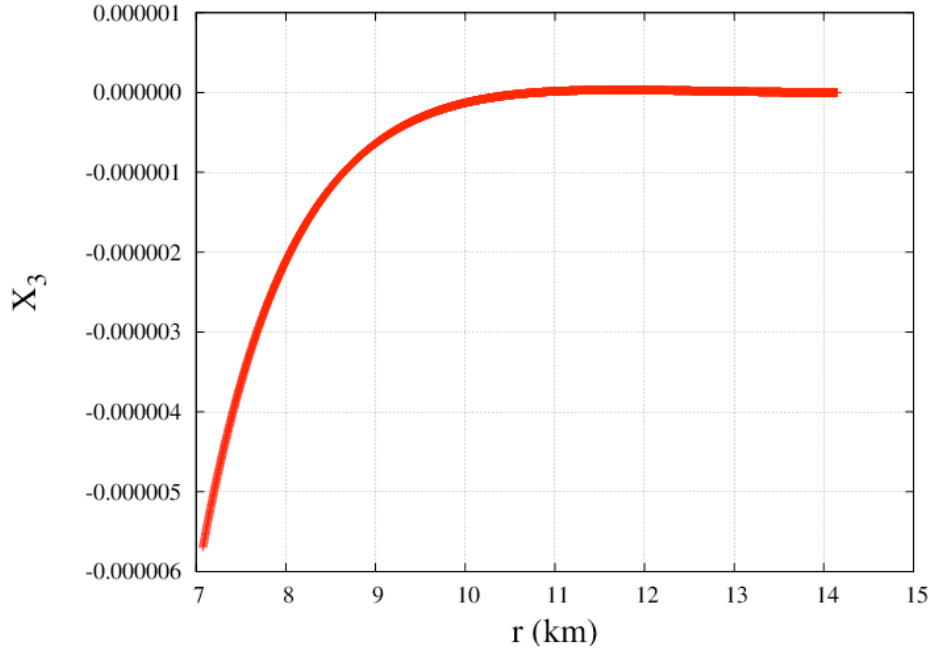


Figura 4.13 -  $X_3(r)$  vs  $r$ .

$$X(r) = X(0) + \frac{1}{2}r^2X''(0) + O(r^4). \quad (4.17)$$

Onde  $H_1(0)$ ,  $K(0)$ ,  $W(0)$  e  $X(0)$  são as condições iniciais e  $H_1''(0)$ ,  $K''(0)$ ,  $W''(0)$  e  $X''(0)$  são determinados resolvendo o seguinte sistema de equações (LINDBLOM L., 1985),

$$\begin{aligned} & -\frac{1}{4}(p_0 + \rho_0)K''(0) + \frac{1}{2} \left[ p_2 + (p_0 + \rho_0) \frac{\omega^2(l+3)}{l(l+1)} e^{-\nu_0} \right] W''(0) + \frac{1}{2} e^{-\nu_0/2} \\ & = \frac{1}{4} \nu_2 e^{-\nu_0/2} X(0) + \frac{1}{4} (p_2 + \rho_2) K(0) + \frac{1}{2} (p_0 + \rho_0) Q_0 + \frac{1}{2} \omega^2 (p_0 + \rho_0) e^{-\nu_0} Q_1 \\ & \quad - \left\{ p_4 - \frac{4\pi}{3} \rho_c p_2 + \frac{\omega^2}{2l} [\rho_2 + p_2 - (\rho_0 + p_0) \nu_2] e^{-\nu_0} \right\} W(0), \quad (4.18) \end{aligned}$$



$$\begin{aligned} & \frac{1}{2}(l+2)K''(0) - \frac{1}{4}l(l+1)H_1''(0) + 4\pi(p_0 + \rho_0)W''(0) \\ &= \frac{4\pi}{3}(3p_0 + \rho_0)K(0) + \frac{1}{2}Q_0 - 4\pi \left[ \nu_2 + p_2 + \frac{8\pi}{3}\rho_0(\rho_0 + p_0) \right] W(0), \end{aligned} \quad (4.19)$$

$$\begin{aligned} & \frac{1}{2}(l+3)H_1''(0) - K''(0) - 8\pi(p_0 + \rho_0)\frac{l+3}{l(l+1)}W''(0) = 4\pi \left[ \frac{1}{3}(2l+3)\rho_0 - p_0 \right] H_1(0) \\ & \quad + \frac{8\pi}{l}(\rho_2 + p_2)W(0) - 8\pi(\rho_0 + p_0)Q_1 + \frac{1}{2}Q_0, \end{aligned} \quad (4.20)$$

$$\begin{aligned} & \frac{1}{2}(l+2)X''(0) - (\rho_0 + p_0)e^{\nu_0/2} \left[ \frac{1}{4}(l+2)\nu_2 - 2\pi(\rho_0 + p_0) - \frac{1}{2}\omega^2 e^{-\nu_0} \right] W''(0) \\ & - \frac{1}{8}l(l+1)(\rho_0 + p_0)e^{\nu_0/2}H_1''(0) = \frac{1}{2} \left[ \rho_2 + p_2 + \frac{1}{2}(\rho_0 + p_0)\nu_2 \right] \frac{lX(0)}{\rho_0 + p_0} \\ & (\rho_0 + p_0)e^{\nu_0/2} \left\{ \frac{1}{2}\nu_2 K(0) + \frac{1}{4}Q_0 + \frac{1}{2}\omega^2 e^{-\nu_0} H_1(0) - \frac{1}{4}l(l+1)\nu_2 Q_1 \right. \\ & \quad \left[ \frac{1}{2}(l+1)\nu_4 - 2\pi(\rho_2 + p_2) - \frac{16\pi^2}{3}\rho_0(\rho_0 + p_0) \right. \\ & \quad \left. \left. + \frac{1}{2} \left( \nu_4 - \frac{4\pi}{3}\rho_0\nu_2 \right) + \frac{1}{2}\omega^2 e^{-\nu_0} \left( \nu_2 - \frac{8\pi}{3}\rho_0 \right) \right] W(0) \right\} \end{aligned} \quad (4.21)$$

Os coeficientes  $Q_0$  e  $Q_1$  são combinações de coeficientes de primeira ordem dados por:

$$\begin{aligned} Q_0 = \frac{4}{(l+2)(l-1)} \left\{ 8\pi e^{-\nu_0/2} X(0) - \left( \frac{8\pi}{3}\rho_0 + \omega^2 e^{-\nu_0} \right) K(0) - \right. \\ \left. \left[ \frac{2\pi}{3}l(l+1)(\rho_0 + 3p_0) - \omega^2 e^{-\nu_0} \right] H_1(0) \right\}, \end{aligned} \quad (4.22)$$

$$Q_1 = \frac{2}{l(l+1)} \left[ \frac{X(0)}{\gamma_0 p_0} e^{-\nu_0/2} + \frac{3}{2} K(0) + \frac{4\pi}{3} (l+1) \rho_0 W(0) \right]. \quad (4.23)$$

Nas equações 4.18, 4.19, 4.20 e 4.21 temos  $p_2$ ,  $p_4$ ,  $\nu_2$ ,  $\nu_4$  e  $\rho_2$ , seus valores são determinados através das equações abaixo (LINDBLOM L., 1983),

$$p_2 = -\frac{4\pi}{3} (\rho_0 + p_0)(\rho_0 + 3p_0), \quad (4.24)$$

$$\rho_2 = p_2 \frac{\rho_0 + p_0}{\gamma_0 p_0}, \quad (4.25)$$

$$\nu_2 = \frac{8\pi}{3} (\rho_0 + 3p_0), \quad (4.26)$$

$$p_4 = \frac{2\pi}{5} (\rho_0 + p_0)(\rho_2 + 5p_2) - \frac{2\pi}{3} (\rho_2 + p_2)(\rho_0 + 3p_0) - \frac{32\pi^2}{9} \rho_0 (\rho_0 + p_0)(\rho_0 + 3p_0), \quad (4.27)$$

$$\nu_4 = \frac{4\pi}{5} (\rho_2 + 5p_2) + \frac{64\pi^2}{9} \rho_0 (\rho_0 + 3p_0). \quad (4.28)$$

As equações 4.18, 4.19, 4.20 e 4.21 podem ser escritas de forma matricial, onde no lado esquerdo da igualdade teremos uma matriz quadrada de ordem 4 ( $A_{4 \times 4}$ ) multiplicando uma matriz coluna ( $Y''(0)$ ) dada pelos termos  $H_1''(0)$ ,  $K''(0)$ ,  $W''(0)$  e  $X''(0)$ . No lado direito temos outra matriz quadrada de ordem 4 ( $B_{4 \times 4}$ ) multiplicando uma matriz coluna ( $Y(0)$ ) com os termos  $H_1(0)$ ,  $K(0)$ ,  $W(0)$  e  $X(0)$ . Teremos então,

$$AY''(0) = BY(0),$$

o que queremos descobrir são os termos da matriz coluna  $Y''(0)$ , então basta

multiplicar ambos os lados da igualdade pela inversa de  $A$ , assim

$$Y''(0) = A^{-1}BY(0).$$

Com isso, o lado esquerdo será apenas a matriz  $Y''(0)$ , ou seja, os termos serão os valores de  $H_1''(0)$ ,  $K''(0)$ ,  $W''(0)$  e  $X''(0)$  que desejamos.

A matriz  $Y(0)$  conhecemos, pois é dada por  $H_1(0)$ ,  $K(0)$ ,  $W(0)$  e  $X(0)$ , que são os valores iniciais das perturbações. As matrizes  $A$  e  $B$  são obtidas das equações 4.18, 4.19, 4.20 e 4.21, elas dependem de  $p_0, p_2, p_4, \nu_0, \nu_2, \nu_4, \rho_0, \rho_2, \omega, l$  e  $n$ .

As perturbações perto do centro são obtidas na rotina `P_C` (Anexo D). Primeiramente foi adaptada uma rotina que resolve as equações de TOV (rotina `tov_rot_z3`, encontrada no Anexo B), e nesta calculamos  $p_2, p_4, \nu_2, \nu_4$  e  $\rho_2$ , sendo assim foi possível obter as matrizes  $A$  e  $B$ . Na rotina `matriz` (Anexo D) estão definidos os termos das matrizes  $A$  e  $B$  logo após serem definidos os termos de  $A$ , essa rotina chama a rotina `inversa` (Anexo D) que inverte a matriz  $A$ . Logo após é realizado o produto da inversa de  $A$  com a matriz  $B$ . Esse produto é passado para a rotina principal e multiplicada pela matriz  $Y(0)$  e dessa forma são obtidos os valores de  $H_1''(0)$ ,  $K''(0)$ ,  $W''(0)$  e  $X''(0)$ .

No entanto, temos 2 conjuntos de condições iniciais, correspondentes às 2 soluções linearmente independentes regulares em  $r=0$ , conseqüentemente vamos ter 2 matrizes  $Y(0)$  distintas e 2 soluções para as perturbações. As condições iniciais são (LINDBLOM L., 1985):

$$H_1(0) = \frac{2lK(0) + 10\pi(\rho_0 + p_0)W(0)}{l(l+1)},$$

$$K(0) = \pm(\rho_0 + p_0),$$

$$W(0) = 1,$$

$$X(0) = (\rho_0 + p_0)e^{\nu_0/2} \left\{ \left[ \frac{4\pi}{3}(\rho_0 + 3p_0) - \frac{\omega^2 e^{-\nu_0}}{l} \right] W(0) + \frac{1}{2}K(0) \right\}.$$

Como vimos, o  $K(0)$  pode assumir 2 valores. Assim como anteriormente, vamos chamar as soluções de  $Y_4$  e  $Y_5$ .

Resumindo, temos que o programa primeiramente chama uma rotina TOV para obter os valores de  $p_0, p_2, p_4, \nu_0, \nu_2, \nu_4, \rho_0, \rho_2, \omega, l$  e  $n$ , com esses valores são montadas as matrizes A e B. Posteriormente a matriz A é invertida e multiplicada pela matriz B. Esse produto é multiplicado pela matriz coluna formada pelas condições iniciais ( $Y(0)$ ) e finalmente são obtidos os valores de  $H_1''(0), K''(0), W''(0)$  e  $X''(0)$ . Usando esses valores nas equações 4.14, 4.15, 4.16 e 4.17 conseguimos determinar as perturbações perto do centro. Determinada as perturbações perto do centro, podemos utilizar o método anterior (solução numérica das equações) e obter as perturbações até  $R/2$ . A rotina `c_m_rot` (Anexo D) resolve as equações 4.10, 4.11, 4.12 e 4.13 do ponto onde a expansão parou até  $R/2$ .

Para mostrar a variação das perturbações nessa região do interior estelar, o programa gera um arquivo de dados com os valores das perturbações para 5000 valores de  $r$ , contidos desde o centro até  $R/2$ . A solução  $Y_4$  é mostrada nas figuras 4.14, 4.15, 4.16 e 4.17. As figuras 4.18, 4.19, 4.20 e 4.21 são referentes a solução  $Y_5$ .

### 4.2.3 Perturbações do Centro até a Superfície

Como vimos, do centro até  $R/2$  temos 2 soluções e da superfície até  $R/2$  temos 3 soluções. No entanto as perturbações devem ser contínuas em todo  $r$ , desta

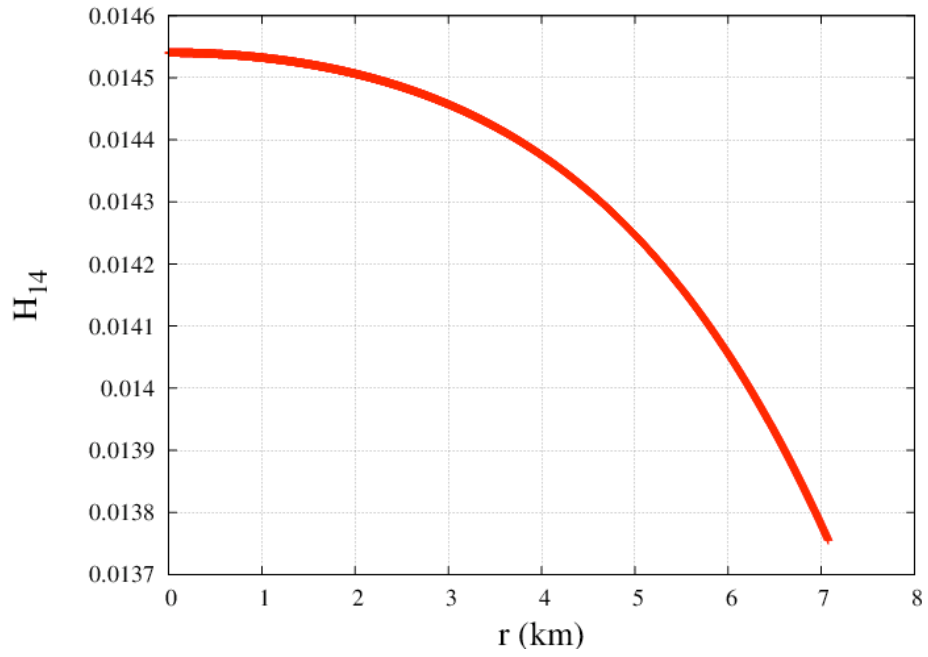


Figura 4.14 -  $H_{14}(r)$  vs  $r$ .

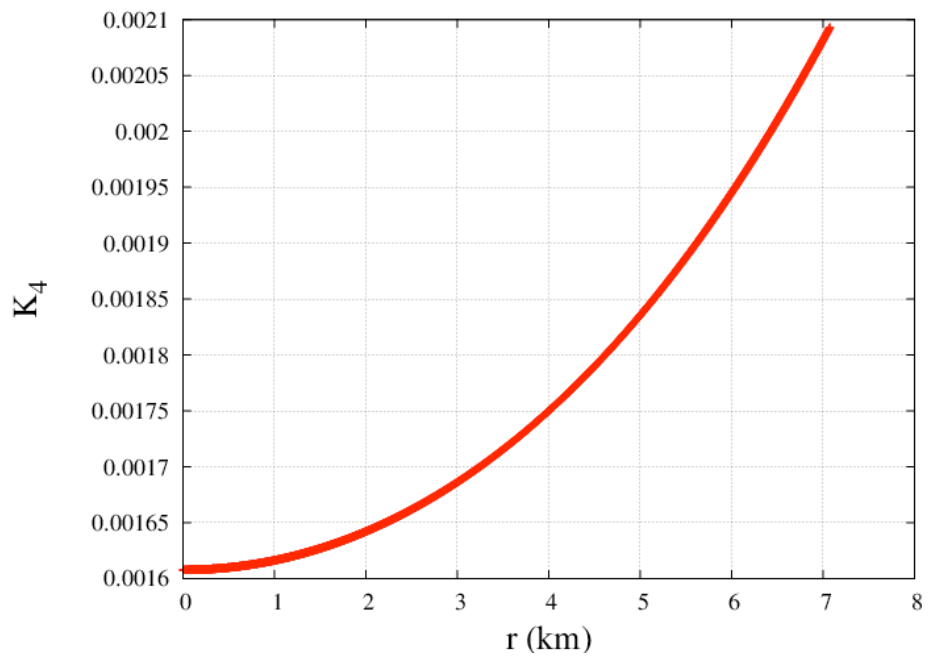


Figura 4.15 -  $K_4(r)$  vs  $r$ .

forma

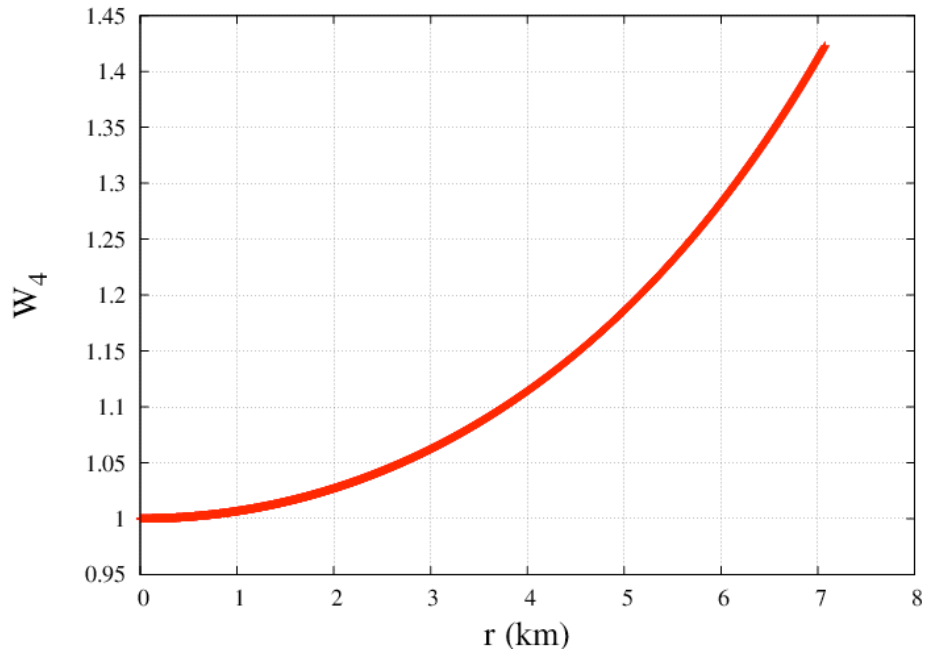


Figura 4.16 -  $W_4(r)$  vs  $r$ .

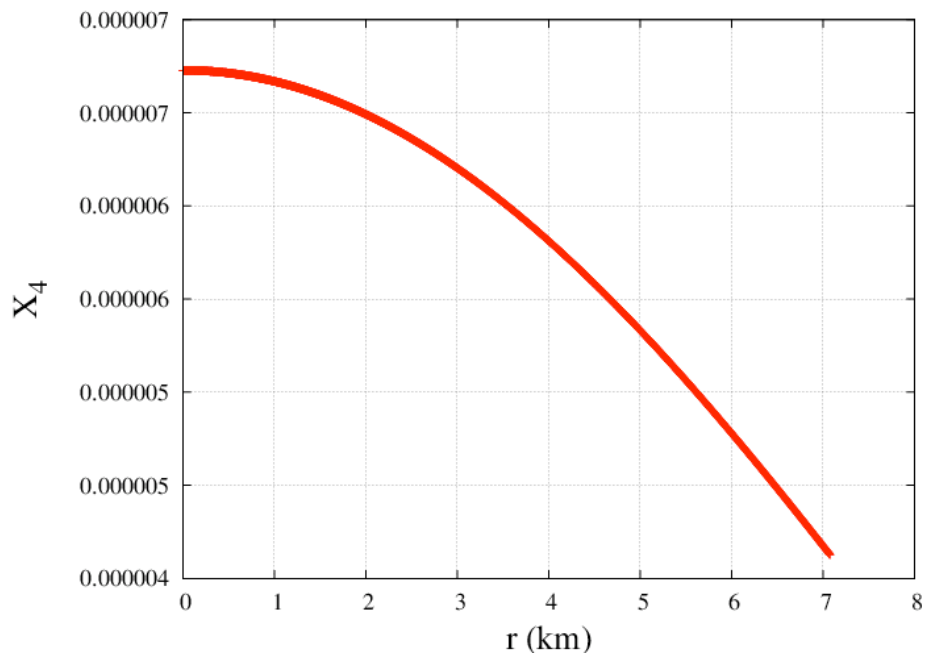


Figura 4.17 -  $X_4(r)$  vs  $r$ .

$$Y(r) = a_1 Y_1(r) + a_2 Y_2(r) + a_3 Y_3(r) \quad \text{para } R \geq r \geq R/2 \quad \text{e}$$

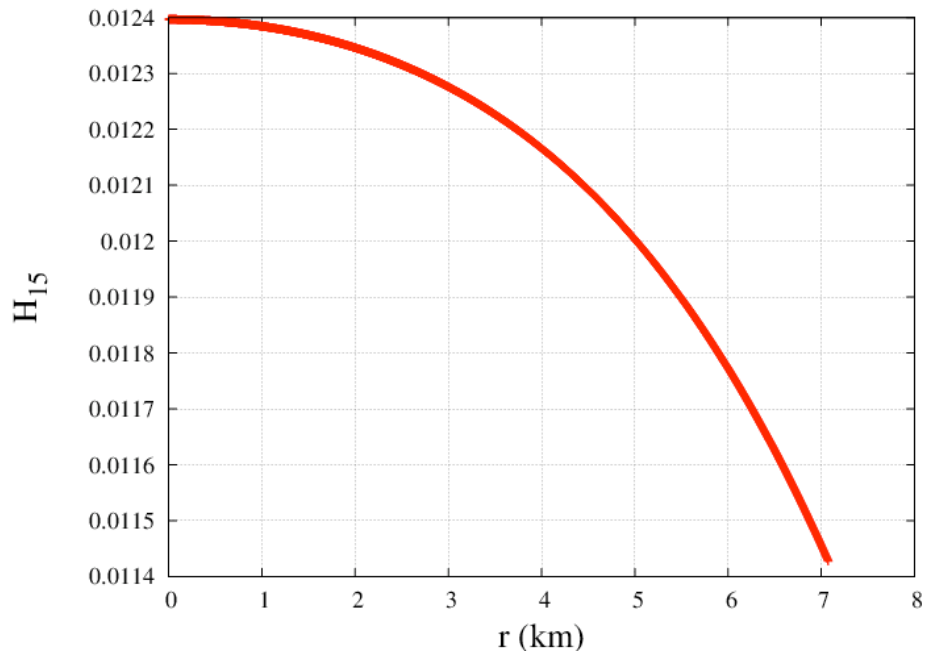


Figura 4.18 -  $H_{15}(r)$  vs  $r$ .

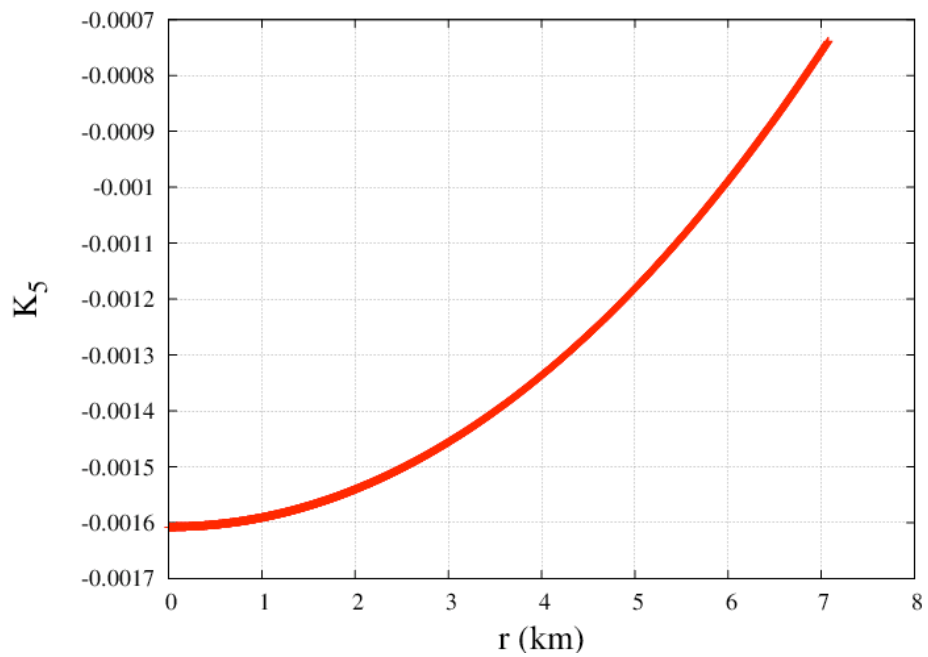


Figura 4.19 -  $K_5(r)$  vs  $r$ .

$$Y(r) = a_4 Y_4(r) + a_5 Y_5(r) \quad \text{para } R/2 \geq r \geq 0$$

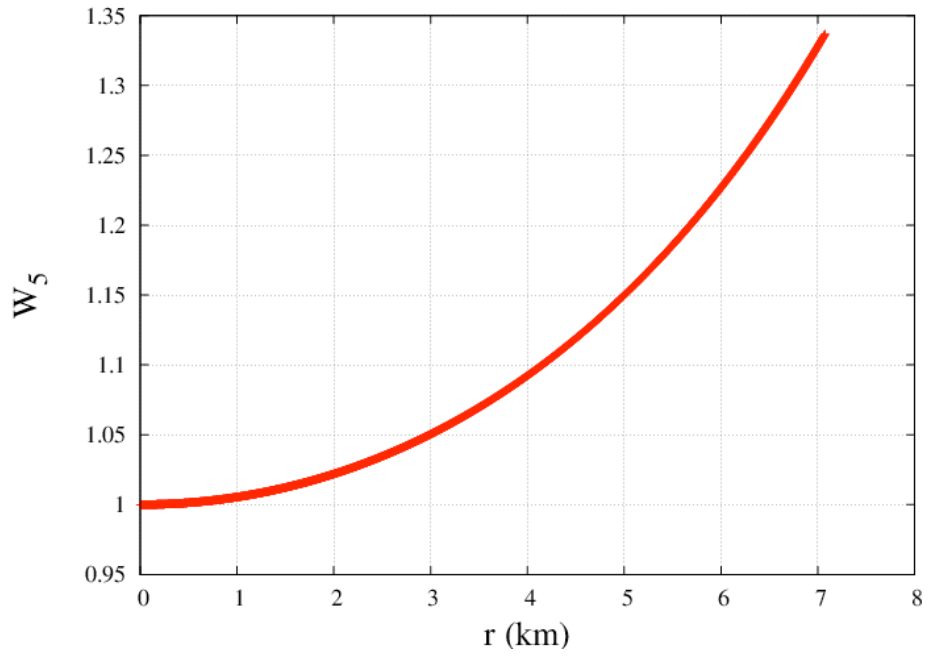


Figura 4.20 -  $W_5(r)$  vs  $r$ .

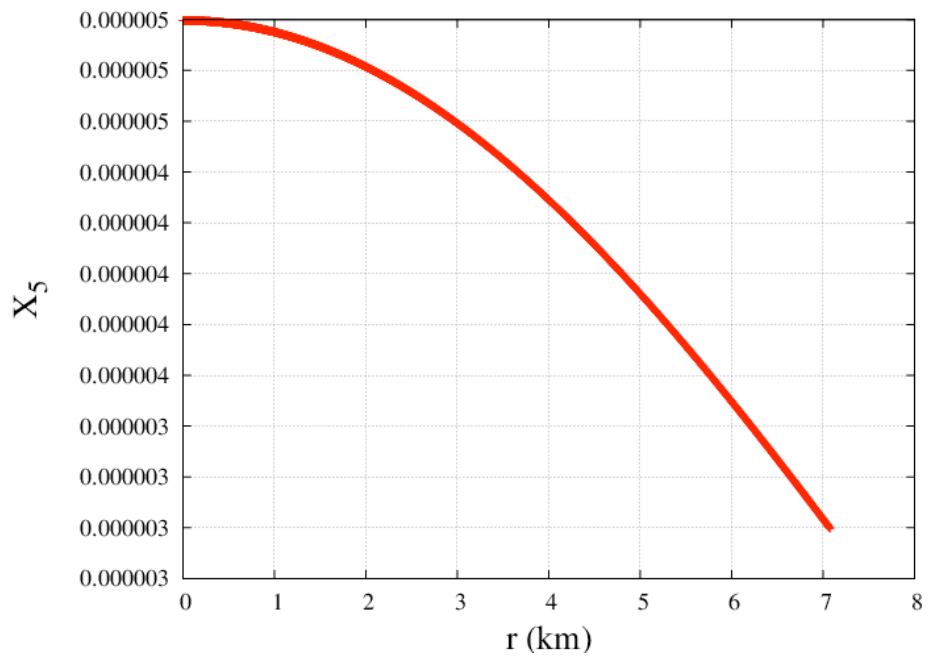


Figura 4.21 -  $X_5(r)$  vs  $r$ .

No entanto em  $R/2$  a solução deve ser a mesma, ou seja



$$a_1 Y_1(R/2) + a_2 Y_2(R/2) + a_3 Y_3(R/2) = a_4 Y_4(R/2) + a_5 Y_5(R/2)$$

Determinando as constantes  $a_1, a_2, a_3, a_4$  e  $a_5$ , foi possível escrever apenas um conjunto de perturbações dentro de toda a estrela. Lembrando que  $Y$  é uma matriz coluna com 4 linhas, dadas por  $H_1, K, W$  e  $X$ , desta forma vamos ter um conjunto de 4 equações, definidas por

$$a_1 H_{11}(R/2) + a_2 H_{12}(R/2) + a_3 H_{13}(R/2) = a_4 H_{14}(R/2) + a_5 H_{15}(R/2)$$

$$a_1 K_1(R/2) + a_2 K_2(R/2) + a_3 K_3(R/2) = a_4 K_4(R/2) + a_5 K_5(R/2)$$

$$a_1 W_1(R/2) + a_2 W_2(R/2) + a_3 W_3(R/2) = a_4 W_4(R/2) + a_5 W_5(R/2)$$

$$a_1 X_1(R/2) + a_2 X_2(R/2) + a_3 X_3(R/2) = a_4 X_4(R/2) + a_5 X_5(R/2)$$

No entanto queremos determinar 5 incógnitas ( $a_1, a_2, a_3, a_4, a_5$ ). Para conseguir resolver o sistema, tomamos  $a_5 = 1$  (podemos fazer isso pois a amplitude total é arbitrária), conseqüentemente

$$a_1 H_{11}(R/2) + a_2 H_{12}(R/2) + a_3 H_{13}(R/2) = a_4 H_{14}(R/2) + H_{15}(R/2)$$

$$a_1 K_1(R/2) + a_2 K_2(R/2) + a_3 K_3(R/2) = a_4 K_4(R/2) + K_5(R/2)$$

$$a_1 W_1(R/2) + a_2 W_2(R/2) + a_3 W_3(R/2) = a_4 W_4(R/2) + W_5(R/2)$$

$$a_1 X_1(R/2) + a_2 X_2(R/2) + a_3 X_3(R/2) = a_4 X_4(R/2) + X_5(R/2)$$

Aqui também podemos escrever estas equações de forma matricial

$$\begin{pmatrix} H_{11}(R/2) & H_{12}(R/2) & H_{13}(R/2) & H_{14}(R/2) \\ K_1(R/2) & K_2(R/2) & K_3(R/2) & K_4(R/2) \\ W_1(R/2) & W_2(R/2) & W_3(R/2) & W_4(R/2) \\ X_1(R/2) & X_2(R/2) & X_3(R/2) & X_4(R/2) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = \begin{pmatrix} H_{15}(R/2) \\ K_5(R/2) \\ W_5(R/2) \\ X_5(R/2) \end{pmatrix}$$

Para determinar as constantes  $a_1$ ,  $a_2$ ,  $a_3$  e  $a_4$  vamos fazer igual a anteriormente: inverter a matriz  $4 \times 4$  da esquerda e multiplicar pela matriz da direita. Calculadas as constantes, as perturbações ficam dadas por,

$$H_1(r) = a_1 H_{11}(r) + a_2 H_{12}(r) + a_3 H_{13}(r) \quad (4.29)$$

$$K(r) = a_1 K_1(r) + a_2 K_2(r) + a_3 K_3(r) \quad (4.30)$$

$$W(r) = a_1 W_1(r) + a_2 W_2(r) + a_3 W_3(r) \quad (4.31)$$

$$X(r) = a_1 X_1(r) + a_2 X_2(r) + a_3 X_3(r) \quad (4.32)$$

para  $R \geq r \geq R/2$  e

$$H_1(r) = a_4 H_{14}(r) + H_{15}(r) \quad (4.33)$$

$$K(r) = a_4 K_4(r) + K_5(r) \quad (4.34)$$

$$W(r) = a_4 W_4(r) + W_5(r)r \quad (4.35)$$

$$X(r) = a_4 X_4(r) + X_5(r) \quad (4.36)$$

para  $R/2 \geq r \geq 0$

O cálculo das constantes, assim como das perturbações são realizadas na rotina `constantes_rot` (Anexo E).

Com os dados obtidos das soluções  $Y_1$  a  $Y_5$  e as equações 4.29 à 4.36 foi possível obter uma única solução em todo o interior da estrela. Essas soluções são mostradas nas figuras 4.22, 4.23, 4.24 e 4.25.

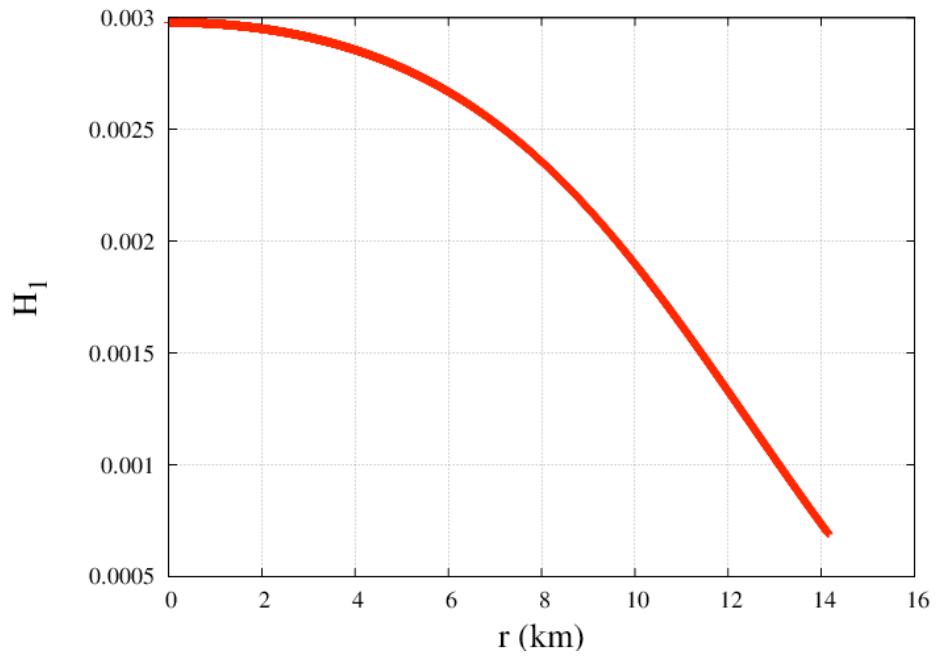


Figura 4.22 -  $H_1(r)$  vs  $r$ .

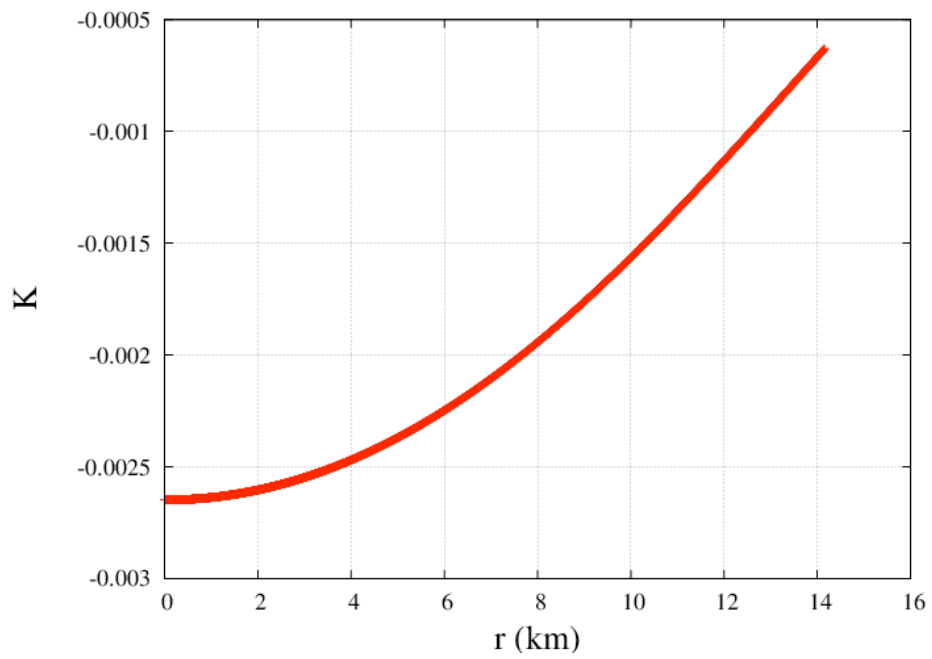


Figura 4.23 -  $K(r)$  vs  $r$ .

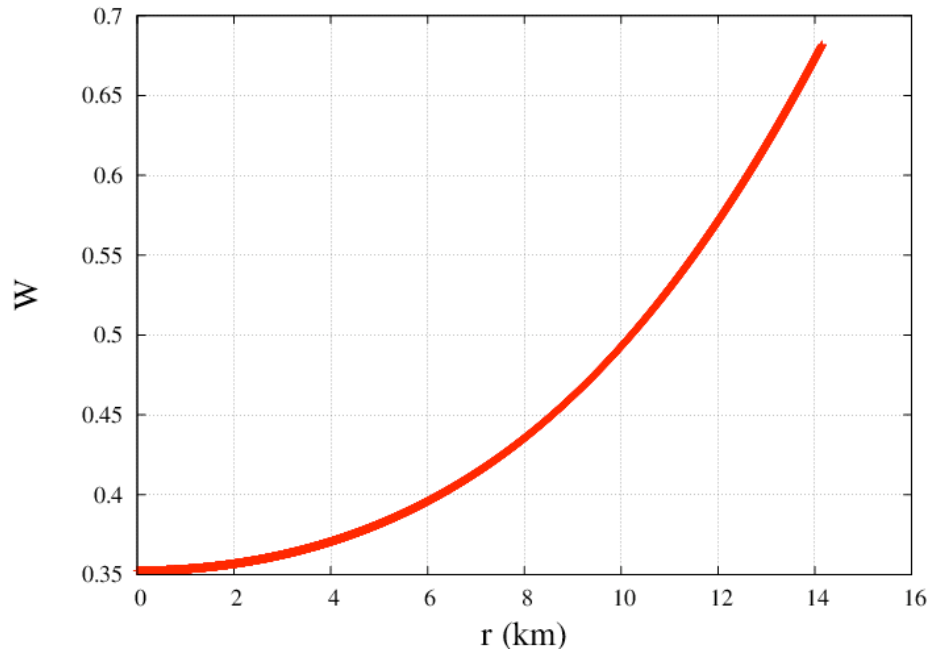


Figura 4.24 -  $W(r)$  vs  $r$ .

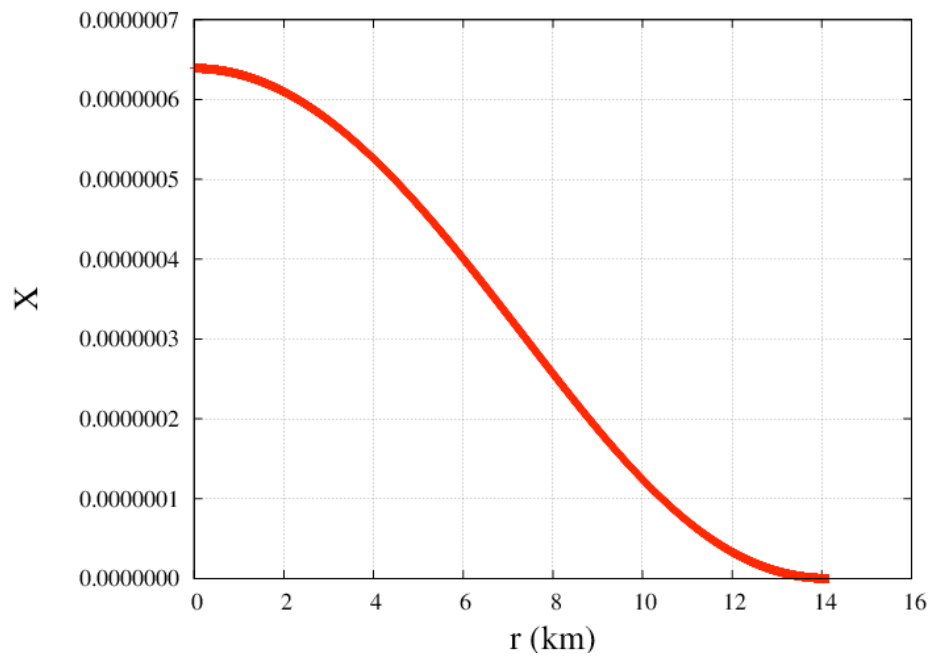


Figura 4.25 -  $X(r)$  vs  $r$ .

#### 4.2.4 Equação de Zerilli

Fora da estrela é mais fácil trabalhar com a equação de Zerilli do que trabalhar com as perturbações. A equação de Zerilli é dada por (LINDBLOM L., 1983),

$$\frac{d^2 Z}{dt^{*2}} = [V_Z(r^*) - \omega^2]Z, \quad (4.37)$$

onde  $r^*$  é a coordenada tartaruga e  $V_Z(r^*)$  é o potencial efetivo definido como

$$V_Z(r^*) = \frac{1 - 2M/r}{r^3(nr + 3M)^2} [2n^2(n + 1)r^3 + 6n^2Mr^2 + 18nM^2r + 18M^3], \quad (4.38)$$

onde,

$$n = \frac{1}{2}(l - 1)(l + 2).$$

No entanto, para resolver a equação de Zerilli (rotina `zerilli_rot` - Anexo F), é necessário o valor de  $Z$  e sua derivada na superfície da estrela. É através da equação 4.39 que a função de zerilli (e sua derivada) se relacionam com  $H_0(R)$  e  $K(R)$  (perturbações da métrica). Tomando  $r = R$  na equação 4.39, é possível determinar as condições iniciais para resolver a equação de Zerilli (LINDBLOM L., 1983)

$$\begin{pmatrix} 0 & 1 \\ a(r) & b(r) \end{pmatrix} \begin{pmatrix} H_0(r) \\ K(r) \end{pmatrix} = \begin{pmatrix} g(r) & 1 \\ h(r) & k(r) \end{pmatrix} \begin{pmatrix} Z(r^*) \\ dZ(r^*)/dr^* \end{pmatrix} \quad (4.39)$$

onde,

$$a(r) = \frac{(nr + 3M)}{\omega^2 r^2 - (n + 1)M/r},$$

$$b(r) = \frac{nr - \omega^2 r^4 + M(r - 3M)}{(r - 2M)[\omega^2 r^2 - (n + 1)M/r]},$$

$$g(r) = \frac{m(n + 1)r^2 + 3nMr + 6M^2}{r^2(nr + 3M)},$$

$$h(r) = \frac{-nr^2 + 3nMr + 3M^2}{(r - 2M)(nr + 3M)},$$

$$k(r) = -\frac{r^2}{r - 2M},$$

Para determinar as condições iniciais para a equação de Zerilli, basta inverter, na equação 4.39, a matriz 2x2 da direita e multiplicar pelo produto das duas matrizes da esquerda.  $H_0(R)$  e  $K(R)$  são obtidos através da rotina `constantes_rot`. Com os valores das condições iniciais, basta aplicar um método numérico e resolver a equação 4.37. A função de Zerilli é obtida desde a superfície ( $r = R$ ) até um  $r = 25\omega^{-1}$ . No entanto, essa equação está variando com  $r^*$  e não com  $r$ . A correspondência entre  $r^*$  e  $r$  é feita através de uma equação transcendental dada por

$$r^* = r + 2M \log(r/2M - 1).$$

Para obter o  $r$  correspondente a um dado  $r^*$ , utilizamos uma rotina do Numerical Recipes que resolve equações transcendentais. A rotina que faz isso é a `rtsafe` (Anexo F) (`rtsafe` calcula a raiz de uma equação dado um chute inicial, usando um método de Zenon-Raplison) e para essa rodar precisa da `funcd` (Anexo F), onde se encontra a equação transcendental.

As figuras 4.26 e 4.27 mostram a função de Zerilli e sua derivada em função de  $r$  e  $r^*$ . Como podemos ver na figura 4.29,  $r^*$  varia linearmente com  $r$  (para  $\mathbf{r} \blacksquare 1$ ), com isso fica explicado o fato de  $Z(r^*)$  em função de  $r^*$  estar apenas deslocada em relação a mesma em função de  $r$ . Na figura 4.28 temos o potencial efetivo, onde podemos observar que o mesmo é máximo na superfície da estrela e vai a zero para  $r \rightarrow \infty$ .

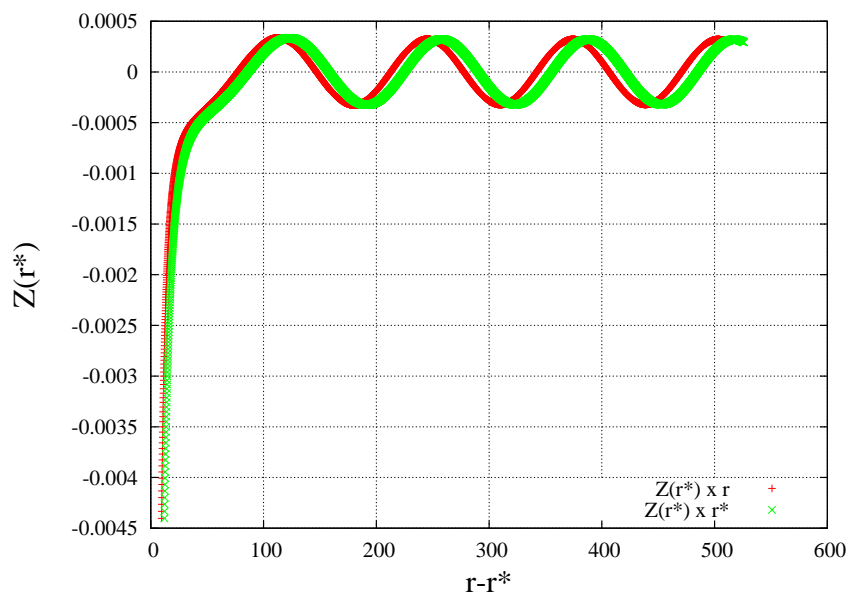


Figura 4.26 -  $Z(r^*)$  vs  $r$  e  $Z(r^*)$  vs  $r^*$ .

#### 4.2.5 Frequência do Modo Quasinormal Fundamental

Existem duas soluções linearmente independentes para a equação de Zerilli (eq. 4.37), as quais podem ser obtidas pelas seguintes séries de potências (para  $r \rightarrow \infty$ , série assintótica) (LINDBLÖM L., 1983):

$$Z_-(r^*) = e^{-i\omega r^*} \sum_{j=0}^{\infty} \alpha_j r^{-j}, \quad (4.40)$$

$$Z_+(r^*) = e^{i\omega r^*} \sum_{j=0}^{\infty} \bar{\alpha}_j r^{-j}. \quad (4.41)$$

Onde  $Z_-$  representa ondas gravitacionais saindo no infinito e  $Z_+$  representa ondas entrando do infinito.

Expandindo as equações 4.40 e 4.41 até  $j = 2$  temos,

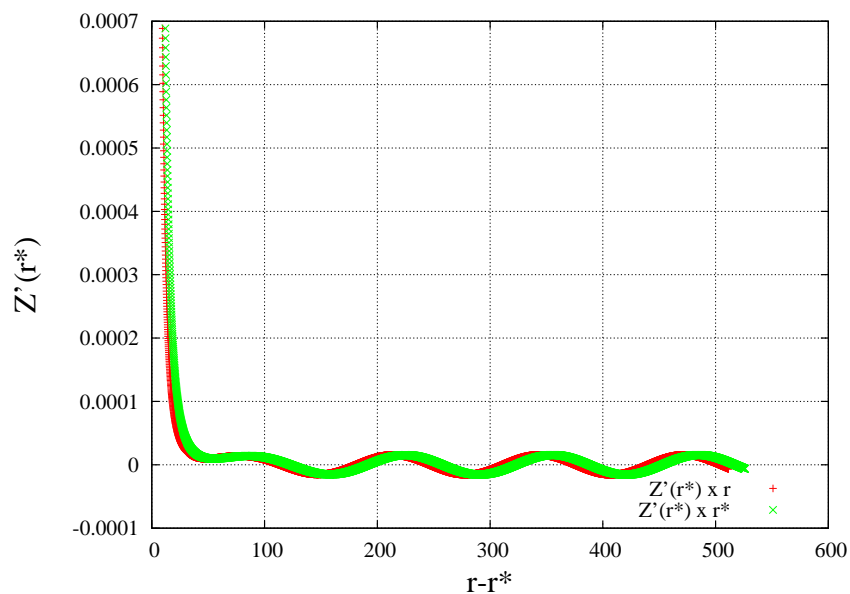


Figura 4.27 -  $Z'(r^*)$  vs  $r$  e  $Z'(r^*)$  vs  $r^*$ .

$$Z_-(r^*) = e^{-i\omega r^*} (\alpha_0 + \alpha_1 r^{-1} + \alpha_2 r^{-2} + \sum_{j=3}^{\infty} \alpha_j r^{-j}),$$

$$Z_+(r^*) = e^{i\omega r^*} (\bar{\alpha}_0 + \bar{\alpha}_1 r^{-1} + \bar{\alpha}_2 r^{-2} + \sum_{j=3}^{\infty} \bar{\alpha}_j r^{-j}).$$

Desconsideramos termos com  $j \geq 3$ , pois teremos terms  $O(1/r^3)$ , assim

$$Z_-(r^*) = e^{-i\omega r^*} (\alpha_0 + \alpha_1 r^{-1} + \alpha_2 r^{-2}), \quad (4.42)$$

$$Z_+(r^*) = e^{i\omega r^*} (\bar{\alpha}_0 + \bar{\alpha}_1 r^{-1} + \bar{\alpha}_2 r^{-2}). \quad (4.43)$$

As constantes  $\alpha_1$  e  $\alpha_2$  são dadas em função de  $\alpha_0$  pelas seguintes relações



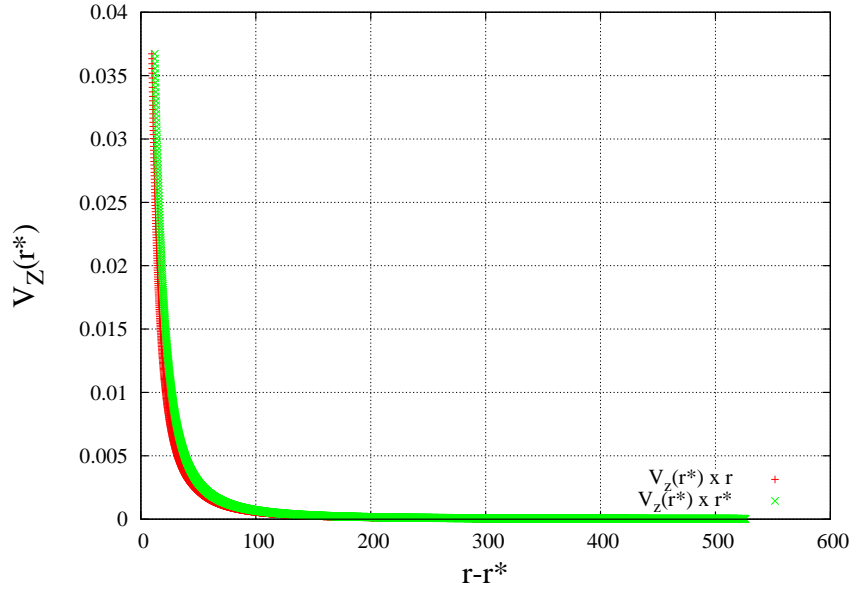


Figura 4.28 -  $V_z(r^*)$  vs  $r$  e  $V_z(r^*)$  vs  $r^*$ .

$$\alpha_1 = -i(n+1)\omega^{-1}\alpha_0, \quad (4.44)$$

$$\alpha_2 = -\frac{1}{2}\omega^{-2}[n(n+1) - \frac{3}{2}iM\omega(1+2/n)]\alpha_0. \quad (4.45)$$

Para obter  $\bar{\alpha}_1$  e  $\bar{\alpha}_2$  basta tomar o complexo conjugado das equações 4.44 e 4.45, desta forma

$$\bar{\alpha}_1 = i(n+1)\omega^{-1}\bar{\alpha}_0, \quad (4.46)$$

$$\bar{\alpha}_2 = -\frac{1}{2}\omega^{-2}[n(n+1) + \frac{3}{2}iM\omega(1+2/n)]\bar{\alpha}_0. \quad (4.47)$$

Usando a fórmula de Euler ( $e^{ix} = \cos(x) + i\sin(x)$ ) e as equações 4.44, 4.45, 4.46 e 4.47,  $Z_-$  e  $Z_+$  serão dados por:

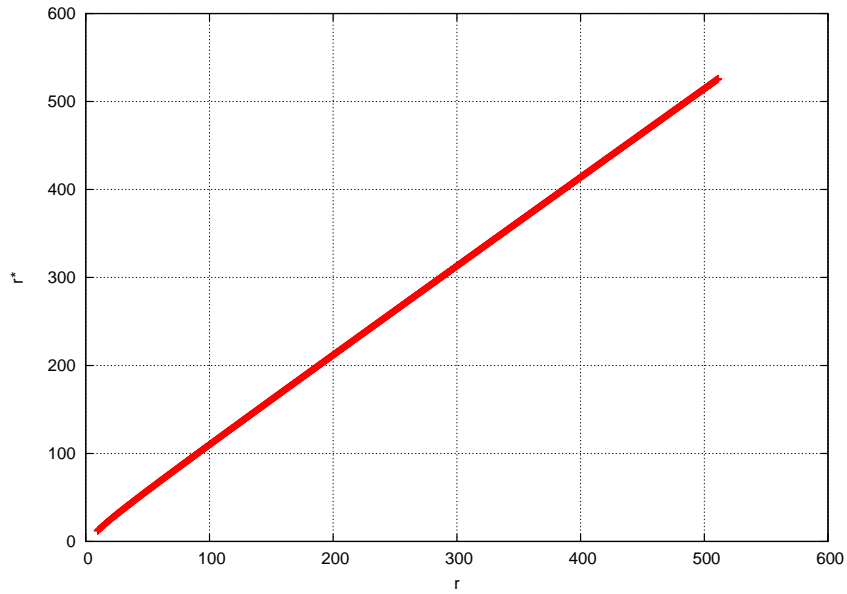


Figura 4.29 -  $r^*(r)$  vs  $r$ .

$$Z_-(r^*) = \alpha_0 \cos(\omega r^*)(Z_1 + iZ_2) + \alpha_0 \sin(\omega r^*)(Z_2 - iZ_1), \quad (4.48)$$

$$Z_+(r^*) = \bar{\alpha}_0 \cos(\omega r^*)(Z_1 - iZ_2) + \bar{\alpha}_0 \sin(\omega r^*)(Z_2 + iZ_1). \quad (4.49)$$

Se  $\omega$  é real, temos  $Z_- = Z_+$ . Os coeficientes  $Z_1$  e  $Z_2$  são definidos por,

$$Z_1 = 1 - \frac{n(n+1)}{2\omega^2 r^2} \quad e$$

$$Z_2 = \frac{3M}{4\omega r^2} (1 + 2/n) - \frac{n+1}{r\omega}.$$

Uma solução arbitrária da equação de Zerilli (eq. 4.50) é dada pela combinação linear de  $Z_-$  e  $Z_+$ .

$$Z(r^*) = \beta(\omega)Z_-(r^*) + \gamma(\omega)Z_+(r^*). \quad (4.50)$$

Porém, para determinar  $\beta$  e  $\gamma$  necessitamos de 2 equações, com isso derivamos  $Z_-$  e  $Z_+$  para obter a segunda equação, assim

$$\frac{dZ(r^*)}{dr^*} = \beta(\omega)\frac{dZ_-(r^*)}{dr^*} + \gamma(\omega)\frac{dZ_+(r^*)}{dr^*}. \quad (4.51)$$

As derivadas  $\frac{dZ_-}{dr^*}$  e  $\frac{dZ_+}{dr^*}$  são:

$$\frac{dZ_-(r^*)}{dr^*} = \alpha_0 \cos(\omega r^*)(Z_4 + iZ_3) + \alpha_0 \sin(\omega r^*)(Z_3 - iZ_4) \quad e \quad (4.52)$$

$$\frac{dZ_+(r^*)}{dr^*} = \bar{\alpha}_0 \cos(\omega r^*)(Z_4 - iZ_3) + \bar{\alpha}_0 \sin(\omega r^*)(Z_3 + iZ_4). \quad (4.53)$$

Onde,

$$Z_3 = \frac{n(n+1)}{2\omega r^2} - \left[ \frac{3M}{2\omega r^3}(1+2/n) - \frac{n+1}{\omega r^2} \right] \left( 1 + \frac{2M}{r-2M} \right)^{-1} - \omega \quad e$$

$$Z_4 = \frac{n(n+1)}{\omega^2 r^3} \left( 1 + \frac{2M}{r-2M} \right)^{-1} + \frac{3M}{4r^2}(1+2/n) - \frac{n+1}{r}.$$

Escrevendo na forma matricial o sistema formado pelas equações 4.50 e 4.51, temos

$$\begin{pmatrix} Z(r^*) \\ \frac{dZ(r^*)}{dr^*} \end{pmatrix} = \begin{pmatrix} Z_-(r^*) & Z_+(r^*) \\ \frac{dZ_-(r^*)}{dr^*} & \frac{dZ_+(r^*)}{dr^*} \end{pmatrix} \begin{pmatrix} \beta(\omega) \\ \gamma(\omega) \end{pmatrix} \quad (4.54)$$

Para determinar  $\beta$  e  $\gamma$  (matriz coluna da direita da igualdade), temos que inverter a matriz 2x2 da direita e multiplicar pela matriz coluna da esquerda. Mas

primeiro é necessário determinar os termos dessas duas matrizes. Utilizamos os valores de  $Z$  e  $\frac{dZ}{dr^*}$  em  $r^*(r) = r^*(r = 25\omega^{-1})$  que foram obtidos no programa `zerili_rot` (Anexo F) e os valores de  $Z_-$ ,  $Z_+$  e suas derivadas foram calculadas utilizando o mesmo  $r^*$ . Os números complexos  $\beta$  e  $\gamma$  são calculados na rotina `zerili2_rot` (Anexo F), a qual posteriormente é utilizada para determinar a frequência e o tempo de decaimento do modo quasinormal fundamental.

Utilizamos inicialmente a frequência Newtoniana para um modo fundamental, onde

$$\omega_i = \sqrt{\frac{2\ell(\ell-1)}{2\ell+1} \left(\frac{M}{R^3}\right)} \quad (4.55)$$

Logo após definimos 3 valores para a frequência  $\omega$ :

$$\omega_A = 0.9\omega_N$$

$$\omega_B = \omega_N$$

$$\omega_C = 1.1\omega_N$$

Logo após, calculamos os coeficientes  $\gamma$  para cada um dos 3 omegas. Os modos quasinormais correspondem a OGs que somente “saem” no infinito, ou seja,  $\gamma = 0$ . Com os 3 gamas, ajustamos um polinômio de 2º grau em  $\gamma(\omega)$  e calculamos as raízes  $\omega_+$  e  $\omega_-$  (complexas), posteriormente escolhemos a raiz adequada para ser a próxima estimativa. Utilizamos a parte real da raiz selecionada e substituímos em um dos omegas (A, B ou C) e repetimos o procedimento até a frequência ser determinada com a precisão desejada.

Através do programa `Zerilli4` (Anexo F), conseguimos calcular a frequência complexa  $\omega = 2\pi f + i/\tau$  para o modo quasinormal fundamental. Onde o  $f$  obtido foi de  $1.5795kHz$  e o tempo de decaimento  $\tau$  de  $0.2987$  s.

## 5 Conclusões

Infelizmente, no decorrer do trabalho tivemos que mudar de estratégia várias vezes, assim como limitar/diminuir os conteúdos a serem abordados. Isso aconteceu devido a complexidade do assunto. Para obter (calcular) os modos é necessário muito conhecimento científico sobre o assunto, assim como ter grande facilidade em programação, para ser possível resolver as equações diferenciais acopladas através de algum método numérico.

Antes de mais nada, para calcular os modos quasinormais foi necessário obter a estrutura interna de uma ENs esfericamente simétrica em equilíbrio hidrostático. Resolvemos as equações de TOV para uma equação de estado politrópica (seção 3.3). Os resultados obtidos (seção 3.4) foram satisfatórios e aceitáveis de acordo com ENs padrões. Para a equação de estado politrópica obtivemos uma estrela de neutrões estável com raio de  $14.1km$  e massa de  $1.4M_s$ .

Obtivemos um código em fortran funcionando perfeitamente, desta forma, e calculamos o modo quasinormal fundamental para estrela de neutrões politrópica. Utilizando este código é possível calcular os modos quasinormais para outros modelos, basta alterar a equação de estado e resolver as equações de TOV, obtendo assim uma nova estrutura estelar, da qual serão extraídos os modos quasinormais.

A frequência complexa  $\omega = 2\pi f + i/\tau$  para o modo quasinormal fundamental de uma estrela de neutrões, possui tipicamente  $f = (1.5 - 3)kHz$  e  $\tau = (0.1 - 0.5)s$ . Os valores que obtivemos são  $f = 1.5795kHz$  e  $\tau = 0.2987s$ . Dessa forma, podemos constatar que os resultados obtidos para uma equação de estado simplificada (equação politrópica) são fisicamente corretos, mostrando o quão robusto é o método que utilizamos. Além da frequência do modo  $f$  ser um valor satisfatório, também obtivemos a forma da onda (função radial), que mostramos no gráfico da figura 4.26.

Utilizando o código, é possível fazer uma série de investigações, como variar a densidade central (mantendo  $K$  e  $n$  fixos) ou, então, variar  $K$  e/ou  $n$  e obter os modos para outras equações de estado politrópicas. Também é possível fazer uma adaptação para equações de estado realistas, e com problema inverso comparar  $M$  e  $R$  para diferentes fits.

Para uma sequência de estrelas com nossa equação de estado politrópica e

aumentando a densidade central, foi possível fazer um fit (eq 5.1) para a frequência  $f$  e o tempo de decaimento  $\tau$  em função da massa e raio da estrela. Dessa forma, sabendo (observando/determinando)  $f$  e  $\tau$ , podemos inferir a massa e raio da ENs.

$$f = 7.36 \times 10^{-2} + 55.80 \sqrt{\frac{M}{R^3}} \quad \text{e} \quad \frac{1}{\tau} = \frac{M^3}{R^4} \left[ 9.91 \times 10^{-2} - 0.33 \left( \frac{M}{R} \right) \right], \quad (5.1)$$

## REFERÊNCIAS BIBLIOGRÁFICAS

- AFORD, M. Cool quarks. **Physics**, v. 3, n. 44, 2010. 1
- AGUIAR O. D., e. a. The brazilian spherical detector: progress and plans. **Classical and Quantum Gravity**, v. 21, p. 457–463, 2004.
- LINDBLOM L., D. S. L. The quadrupole oscillations of neutron stars. **ApJS:1983ApJS...53...73L**, 1983. 26, 29, 38, 48, 49, 51
- \_\_\_\_\_. On the nonradial pulsations of general relativistic stellar models. **ApJ:1967ApJ...149..591T**, 1985. 27, 32, 36, 39
- SCHUTZ, B. F. **A First Course in General Relativity**. [S.l.]: Cambridge, 1985. 12
- \_\_\_\_\_. Gravitational wave astronomy. **Classical and Quantum Gravity**, v. 16, p. 131–156, 1999. 5, 7, 8
- THORNE K. S. E CAMPOLATTARO, A. Non-radial pulsation of general-relativistic stellar models. i. analytic analysis for  $l \geq 2$ . **The Astrophysical Journal**, v. 149, p. 591–611, 1967. 27
- TOOPER, R. F. Adiabatic fluid spheres in general relativity. **The Astrophysical Journal**, v. 142, p. 1541–1560, 1965. 20, 21
- WEISBERG J. M., T. J. H. Relativistic binary pulsar b1913 + 16: Thirty years of observations and analysis. **arXiv:astro-ph/0407149v1**, 2004. 9, 11





## ANEXO A - DERIVADAS DAS PERTURBAÇÕES

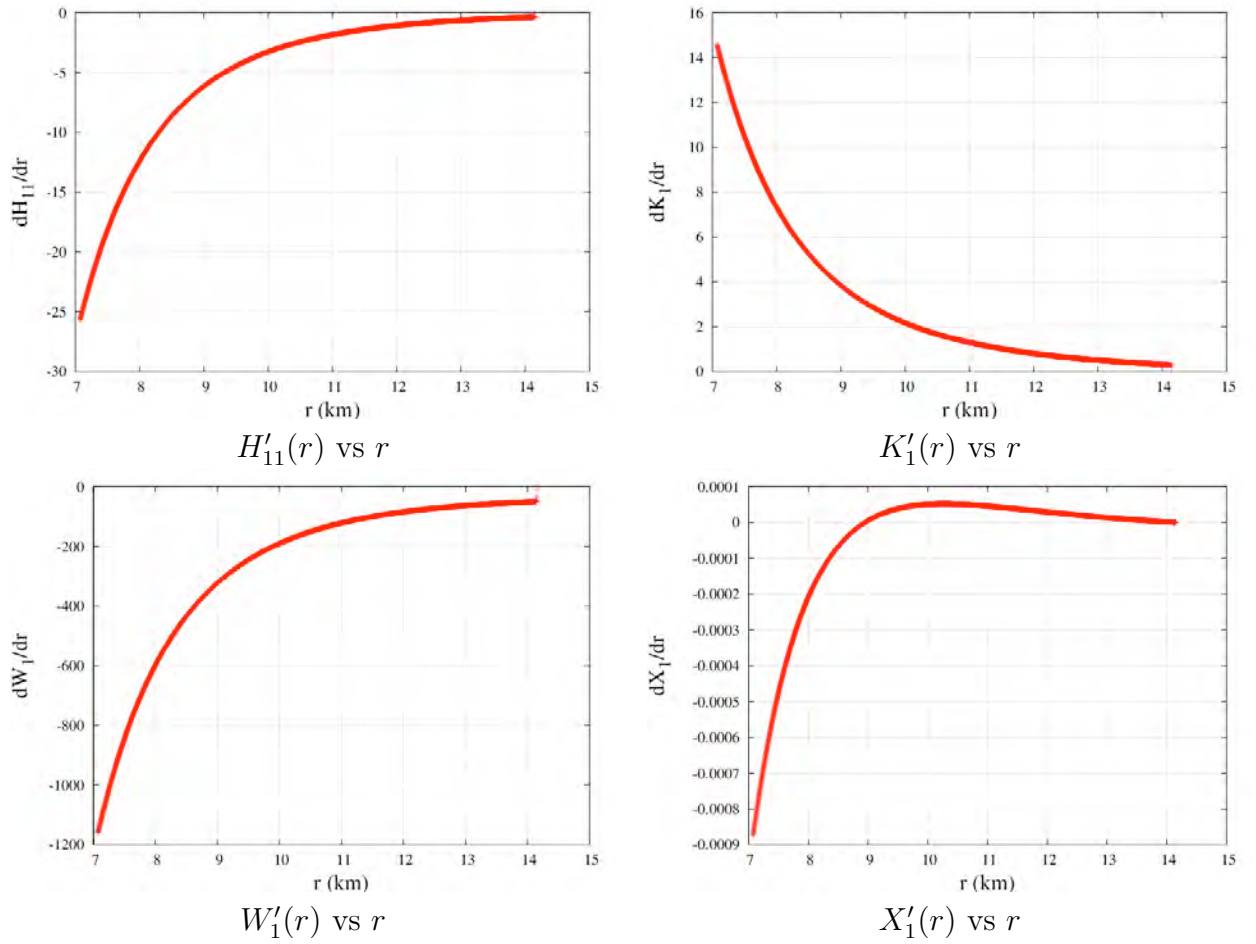
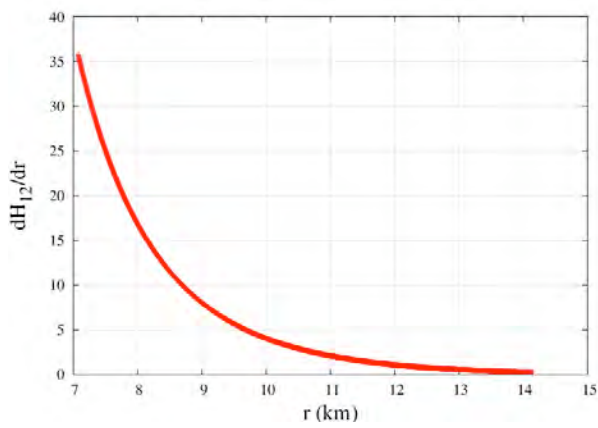
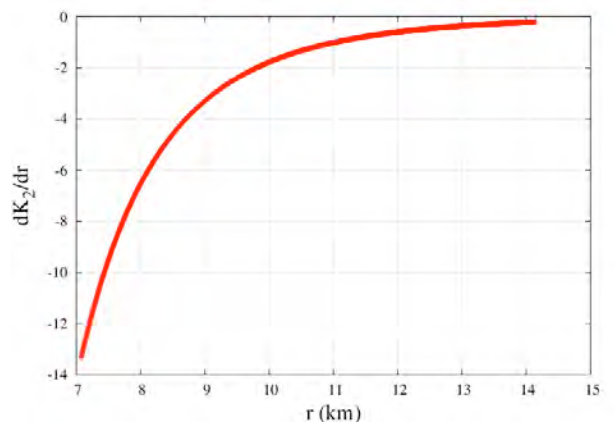


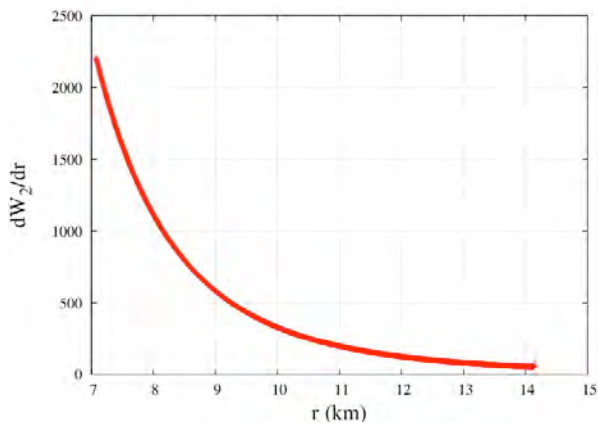
Figura A.1 - Gráficos das derivadas das perturbações referentes a solução  $Y_1$ .



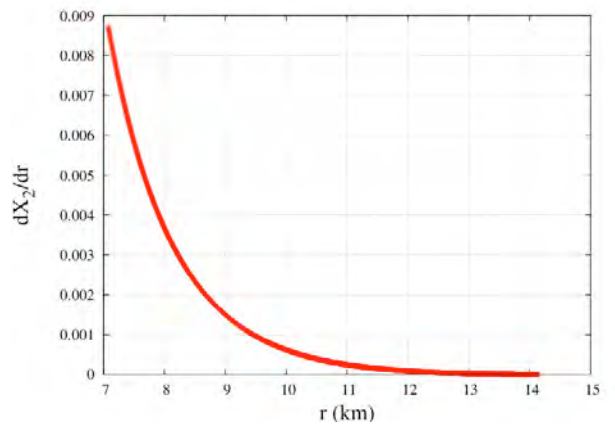
$H'_{12}(r)$  vs  $r$



$K'_2(r)$  vs  $r$

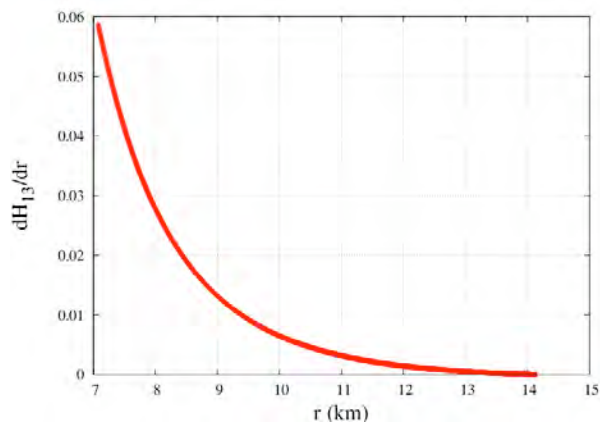


$W'_2(r)$  vs  $r$

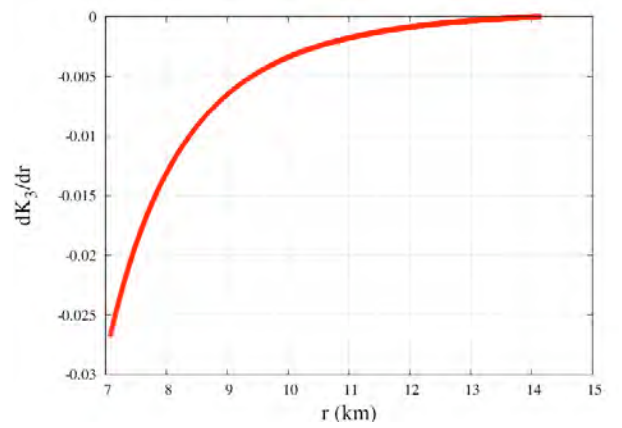


$X'_2(r)$  vs  $r$

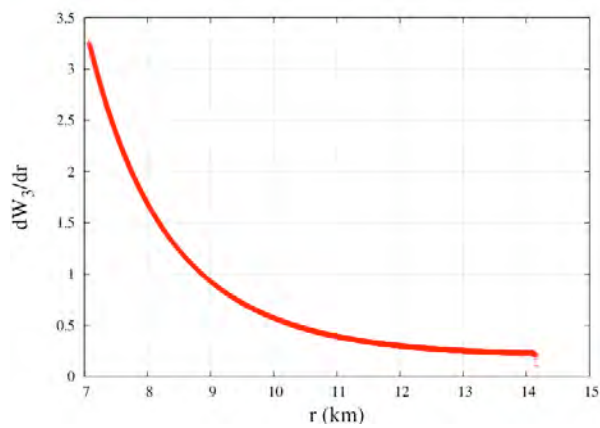
Figura A.2 - Gráficos das derivadas das perturbações referentes a solução  $Y_2$ .



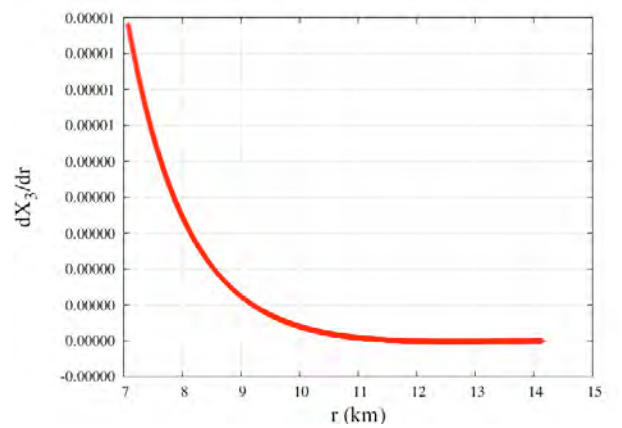
$H'_{13}(r)$  vs  $r$



$K'_3(r)$  vs  $r$

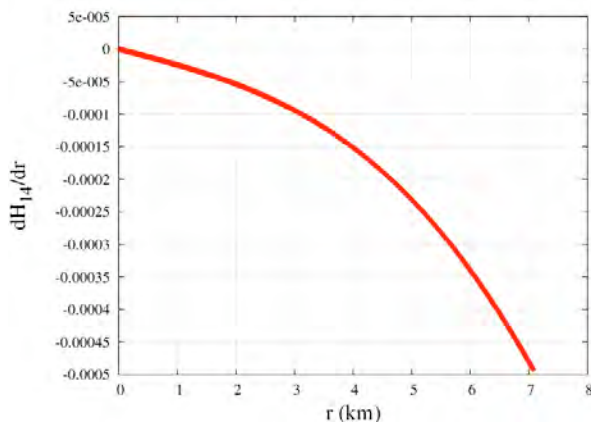


$W'_3(r)$  vs  $r$

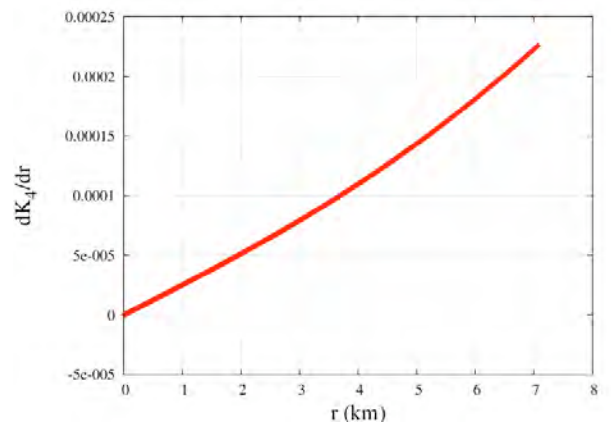


$X'_3(r)$  vs  $r$

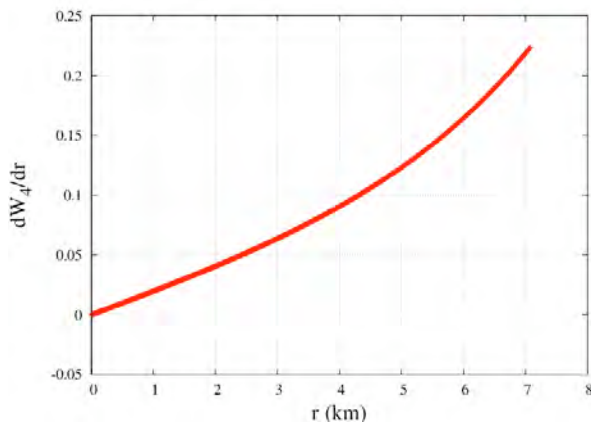
Figura A.3 - Gráficos das derivadas das perturbações referentes a solução  $Y_3$ .



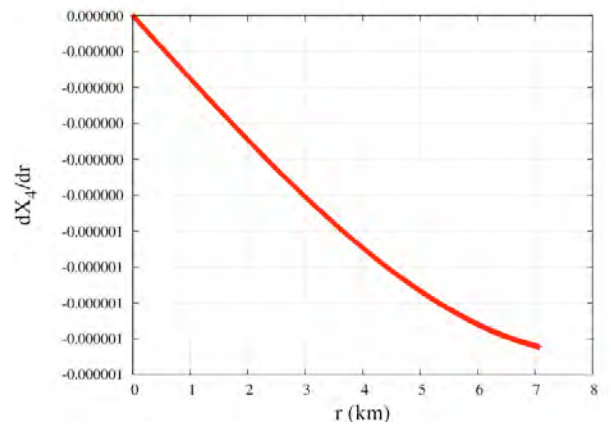
$H'_{14}(r)$  vs  $r$



$K'_4(r)$  vs  $r$

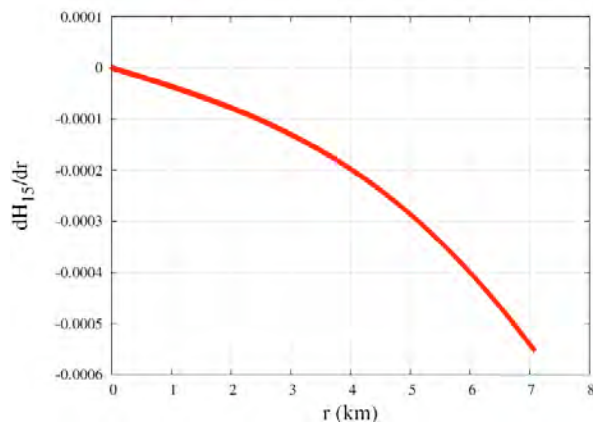


$W'_4(r)$  vs  $r$

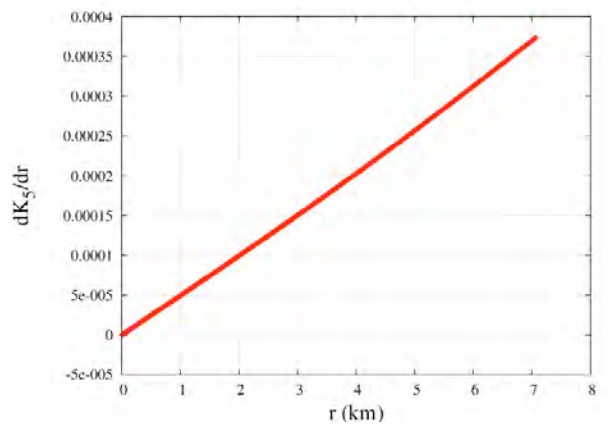


$X'_4(r)$  vs  $r$

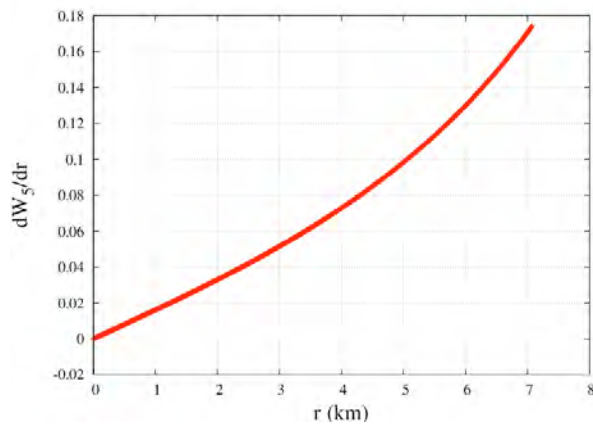
Figura A.4 - Gráficos das derivadas das perturbações referentes a solução  $Y_4$ .



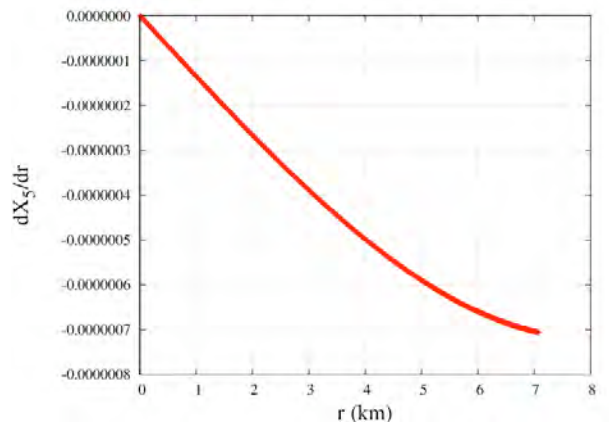
$H'_{15}(r)$  vs  $r$



$K'_5(r)$  vs  $r$



$W'_5(r)$  vs  $r$



$X'_5(r)$  vs  $r$

Figura A.5 - Gráficos das derivadas das perturbações referentes a solução  $Y_5$ .



## ANEXO B - ESTRUTURA INTERNA DE UMA ESTRELA DE NÊUTRONS

### Rotina tov\_rot\_z

```

subroutine tov_rot_z(nn,K,rhob0,l,omega,nsteps)
implicit none
integer, parameter :: dp= SELECTED_REAL_KIND(15,300)
integer, parameter :: n = 2, ns=2
integer :: i, j, nsteps
real(kind= dp) :: rv(0:nsteps), pv(0:nsteps), mv(0:nsteps), rhov(0:nsteps), pvl(0:nsteps), lamb(0:nsteps)
real(kind= dp) :: ni(0:nsteps), nil(0:nsteps), nill(0:nsteps), lambl(0:nsteps)
real(kind= dp), parameter :: a = 0.0, b =20.0
real(kind= dp) :: d, e, f, rhob0, rhob, r, rho, h, RR1, hh, m_linha, MM, p1
real(kind= dp) :: l, pi, K, nn, Rg, prg, RR, prg_linha, delta_ni, ni0, p0
real(kind= dp) :: ni1, ni2, ni3, ni4, ni5, p2, ni22, rho2, p4, ni44, omega
real(kind= dp) :: rmin
real(kind= dp) :: x(0:n)

pi=4.0d0*atan(1.0d0)
rhob=rhob0
d=0.0d0
e=K*(rhob)**(1.0d0+1.0d0/nn)
f=0.0d0
h = (b - a)/nsteps
r=0.0d0
j=nsteps-1
rho=rhob+nn*e

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!Expansao em serie para TOV

p2=-(4.0d0*pi/3.0d0)*(rho+e)*(rho+3.0d0*e)
rho2=p2*(rho+e)/((1.0d0+1.0d0/nn)*e)
ni22=(8.0d0*pi/3.0d0)*(rho+3.0d0*e)
p4=-(2.0d0*pi/5.0d0)*(rho+e)*(rho2+5.0d0*p2)-(2.0d0*pi/3.0d0)*(rho2+p2)*(rho+3.0d0*e)-
((32.0d0*pi**2)/9.0d0)*rho*(rho+e)*(rho+3.0d0*e)
ni44=(4.0d0*pi/5.0d0)*(rho2+5.0d0*p2)+((64.0d0*pi**2)/9.0d0)*rho*(rho+3.0d0*e)

do i=0, ns-1
r=(i)*h
rv(i)=r

pv(i) = e + p2*(r**2)/2.0d0 + p4*(r**4)/4.0d0
rhov(i) = rho + rho2*(r**2)/2.0d0
mv(i) = 4.0d0*pi*rho*(r**3)/3.0d0 + 2.0d0*pi*rho2*(r**5)/5.0d0
ni(i) = f + ni22*r**2/2.0d0 + ni44*r**4/4.0d0
nil(i) = ni22*r + ni44*r**3

```

```

nill(i) = ni22 + 3.0d0*ni44*r**2
pvl(i) = p2*r + p4*r**3
lamb(i)=-log(1.0d0 - (8.0d0/3.0d0)*pi*rho*(r**2) - (4.0d0/5.0d0)*pi*rho2*(r**4))
lamb1(i)=(16.0d0*pi*rho*r/3.0d0+16.0d0*pi*rho2*r**3/5.0d0)/(1.0d0-(8.0d0/3.0d0)*pi*rho*(r**2)-
(4.0d0/5.0d0)*pi*rho2*(r**4))

! print *, i, rv(i), nill(i)

end do

print *, "loop 1: end of series expansion for TOV at r = ",r

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!1a Integracao numerica para TOV

x = (/mv(ns-1), pv(ns-1), ni(ns-1)/)

do i=ns, j
  r = r + h
  call rk4sys(n,h,x,r,K,nn)
  if(x(1)<=0.0d0) then
    print *, "found surface"
    go to 50
  end if
  rhob = (x(1)/K)**(nn/(nn+1.0d0))
  rho=rhob+nn*x(1)
  p1=x(1)*5.54d5
  prg=x(1)
  prg_linha=-((1.0d0/r**2)*(x(0) + 4.0d0*pi*(r**3)*x(1))*(rho + x(1)))/(1.0d0-2.0d0*x(0)/r)
end do

50 continue

Rg=r-h
RR=Rg-(nn+1.0d0)*(prg/prg_linha)

print *, "R = ", RR

delta_ni=log(1.0d0-2.0d0*x(0)/RR)-x(2)
rhob=rhob0
d=0.0d0
e=K*(rhob)**(1.0+1.0/nn)
f=delta_ni
rho=rhob+nn*e
rmin=1.0d-6*RR
hh=(RR-rmin)/(nsteps-1)

j=nsteps-1

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! 2o. loop para TOV
! Recalculando a expansao em série com novo passo
! Começando em rmin = 1e-6*RR

```



```

do i=0, ns-1
  r=rmin+i*hh
  rv(i)=r

  pv(i) = e + p2*(r**2)/2.0d0 + p4*(r**4)/4.0d0
  rhov(i) = rho + rho2*(r**2)/2.0d0
  mv(i) = 4.0d0*pi*rho*(r**3)/3.0d0 + 2.0d0*pi*rho2*(r**5)/5.0d0
  ni(i) = f + ni22*r**2/2.0 + ni44*r**4/4.0d0
  nil(i) = ni22*r + ni44*r**3
  nill(i) = ni22 + 3.0d0*ni44*r**2
  pvl(i) = p2*r + p4*r**3
  lamb(i)=-log(1.0d0 - (8.0d0/3.0d0)*pi*rho*(r**2) - (4.0d0/5.0d0)*pi*rho2*(r**4))
  lambl(i)=(16.0d0*pi*rho*r/3.0d0+16.0d0*pi*rho2*r**3/5.0d0)/(1.0d0-(8.0d0/3.0d0)*pi*rho*(r**2)-
(4.0d0/5.0d0)*pi*rho2*(r**4))

  print *, "loop2: end of series expansion for TOV at r = ",r

  x = (/mv(ns-1), pv(ns-1), ni(ns-1)/)

do i=ns-1, j
  r =rmin+i*hh
  rhob = (x(1)/K)**(nn/(nn+1.0))
  rho=rhob+nn*x(1)
  rv(i)=r
  mv(i)=x(0)
  pv(i)=x(1)
  rhov(i)=rho
  m_linha=4.0d0*pi*(r**2)*rho
  ni(i)=x(2)
  pvl(i)=-((1.0d0/r**2)*(x(0) + 4.0d0*pi*(r**3)*x(1))*(rho + x(1)))/(1.0d0-2.0d0*x(0)/r)
  nil(i)=2.0d0*(1.0/r**2)*(x(0) + 4.0d0*pi*(r**3)*x(1))/(1.0d0-2.0d0*x(0)/r)
  ni1=2.0d0/r**2
  ni2=1.0d0-2.0d0*x(0)/r
  ni3=x(0)+4.0d0*pi*(r**3)*x(1)
  ni4=(m_linha+4.0*pi*r**2*(r*pvl(i)+3.0d0*x(1)))/ni2
  ni5=-2.0d0*m_linha/r+ni1*x(0)
  nill(i)=ni1*ni4-(2.0d0*ni1/r)*ni3/ni2-ni1*ni3*ni5/ni2**2
  lamb(i)=-log(ni2)
  lambl(i)=-ni5/ni2

  call rk4sys(n, hh, x, r, K, nn)

end do

print *, "final grid point at ",rv(j)

MM=mv(nsteps-1)
omega=sqrt(2.0d0*1*(1-1.0d0)/(2.0d0*1+1.0d0))*sqrt(MM/RR**3)

end subroutine tov_rot_z

```

## Rotina tov\_rot\_z3

```

subroutine tov_rot_z3(nn,K,rhob0,hh,p0,p2,p4,ni0,ni22,ni44,rho2,mv,pv,rhov,pv1,ni,nil,nill,lamb,lambl,nsteps,ns,rmin)
implicit none
integer, parameter :: dp= SELECTED_REAL_KIND(15,300)
integer, parameter :: n = 2
integer :: i, j, nsteps, ns
real(kind= dp) :: rv(0:nsteps), pv(0:nsteps), mv(0:nsteps), rhov(0:nsteps), pv1(0:nsteps), lamb(0:nsteps)
real(kind= dp) :: ni(0:nsteps), nil(0:nsteps), nill(0:nsteps), lambl(0:nsteps)
real(kind= dp), parameter :: a = 0.0d0, b =20.0d0
real(kind= dp) :: d, e, f, rhob0, rhob, r, rho, h, RR1, hh, m_linha, MM, p1
real(kind= dp) :: pi, K, nn, Rg, prg, RR, prg_linha, delta_ni, ni0, p0
real(kind= dp) :: ni1, ni2, ni3, ni4, ni5, p2, ni22, rho2, p4, ni44
real(kind= dp) :: rmin
real(kind= dp) :: x(0:n)

open(10,file="Test.dat")
open(14,file="RaioMassa.dat")
open(16,file="RaioPressao.dat")
open(18,file="RaioDensidade.dat")
open(11,file="RaioNi.dat")
open(13,file="RaioNiLinha.dat")
open(15,file="RaioNi2Linhas.dat")
open(17,file="RaioLambda.dat")
open(19,file="RaioLambdaLinha.dat")
open(12,file="RaioPressaoLinha.dat")

pi=4.0d0*atan(1.0d0)
rhob=rhob0
d=0.0d0
e=K*(rhob)**(1.0+1.0/nn)
f=0.0
h = (b - a)/nsteps
r=0.0d0
j=nsteps-1
rho=rhob+nn*e

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!Expansao em serie para TOV

p2=-(4.0d0*pi/3.0d0)*(rho+e)*(rho+3.0d0*e)
rho2=p2*(rho+e)/((1.0d0+1.0d0/nn)*e)
ni22=(8.0d0*pi/3.0d0)*(rho+3.0d0*e)
p4=-(2.0d0*pi/5.0d0)*(rho+e)*(rho2+5.0d0*p2)-(2.0d0*pi/3.0d0)*(rho2+p2)*(rho+3.0d0*e)-
((32.0d0*pi**2)/9.0d0)*rho*(rho+e)*(rho+3.0d0*e)
ni44=(4.0d0*pi/5.0d0)*(rho2+5.0d0*p2)+((64.0d0*pi**2)/9.0d0)*rho*(rho+3.0d0*e)

do i=0, ns-1
  r=(i)*h
  rv(i)=r

  pv(i) = e + p2*(r**2)/2.0d0 + p4*(r**4)/4.0d0
  rhov(i) = rho + rho2*(r**2)/2.0d0
  mv(i) = 4.0d0*pi*rho*(r**3)/3.0d0 + 2.0d0*pi*rho2*(r**5)/5.0d0

```

```

ni(i) = f + ni22*r**2/2.0d0 + ni44*r**4/4.0d0
nil(i) = ni22*r + ni44*r**3
nill(i) = ni22 + 3.0d0*ni44*r**2
pvl(i) = p2*r + p4*r**3
lamb(i)=-log(1.0d0 - (8.0d0/3.0d0)*pi*rho*(r**2) - (4.0d0/5.0d0)*pi*rho2*(r**4))
lamb1(i)=(16.0d0*pi*rho*r/3.0d0+16.0d0*pi*rho2*r**3/5.0d0)/(1.0d0-(8.0d0/3.0d0)*pi*rho*(r**2)-
(4.0d0/5.0d0)*pi*rho2*(r**4))

end do

print *, "loop 1: end of series expansion for TOV at r = ",r

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!1a Integracao numerica para TOV

x = (/mv(ns-1), pv(ns-1), ni(ns-1)/)

do i=ns, j
  r = r + h
  call rk4sys(n,h,x,r,K,nn)
  if(x(1)<=0.0d0) then
    print *, "found surface"
    go to 50
  end if
  rhob = (x(1)/K)**(nn/(nn+1.0d0))
  rho=rhob+nn*x(1)
  p1=x(1)*5.54d5
  prg=x(1)
  prg_linha=-((1.0d0/r**2)*(x(0) + 4.0d0*pi*(r**3)*x(1))*(rho + x(1)))/(1.0d0-2.0d0*x(0)/r)
  write(10,fmt='(4(e12.5,1x))') r, x(0), x(1), rho
end do

50 continue

Rg=r-h
RR=Rg-(nn+1.0d0)*(prg/prg_linha)

print *, "R = ", RR

RR1=RR*1.4767d0

delta_ni=log(1.0d0-2.0d0*x(0)/RR)-x(2)
rhob=rhob0
d=0.0d0
e=K*(rhob)**(1.0d0+1.0d0/nn)
f=delta_ni
rho=rhob+nn*e
rmin=1.0d-6*RR
hh=(RR-rmin)/(nsteps-1)

print *, "hh = ", hh
print *, "(nsteps-1)*hh = ", (nsteps-1)*hh

j=nsteps-1

```

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! 2o. loop para TOV
! Recalculando a expansao em série com novo passo
! Começando em rmin = 1e-6*RR

do i=0, ns-1
  r=rmin+i*hh
  rv(i)=r

  pv(i) = e + p2*(r**2)/2.0d0 + p4*(r**4)/4.0d0
  rhov(i) = rho + rho2*(r**2)/2.0d0
  mv(i) = 4.0d0*pi*rho*(r**3)/3.0d0 + 2.0d0*pi*rho2*(r**5)/5.0d0
  ni(i) = f + ni22*r**2/2.0d0 + ni44*r**4/4.0d0
  nil(i) = ni22*r + ni44*r**3
  nill(i) = ni22 + 3.0d0*ni44*r**2
  pvl(i) = p2*r + p4*r**3
  lamb(i)=-log(1.0d0 - (8.0d0/3.0d0)*pi*rho*(r**2) - (4.0d0/5.0d0)*pi*rho2*(r**4))
  lambl(i)=(16.0d0*pi*rho*r/3.0d0+16.0d0*pi*rho2*r**3/5.0d0)/(1.0d0-(8.0d0/3.0d0)*pi*rho*(r**2)-
(4.0d0/5.0d0)*pi*rho2*(r**4))

end do

print *, "loop2: end of series expansion for TOV at r = ",r

x = (/mv(ns-1), pv(ns-1), ni(ns-1)/)

do i=ns-1, j
  r =rmin+i*hh
  rhob = (x(1)/K)**(nn/(nn+1.0d0))
  rho=rhob+nn*x(1)
  rv(i)=r
  mv(i)=x(0)
  pv(i)=x(1)
  rhov(i)=rho
  m_linha=4.0d0*pi*(r**2)*rho
  ni(i)=x(2)
  pvl(i)=-((1.0d0/r**2)*(x(0) + 4.0d0*pi*(r**3)*x(1))*(rho + x(1)))/(1.0d0-2.0d0*x(0)/r)
  nil(i)=2.0d0*(1.0d0/r**2)*(x(0) + 4.0d0*pi*(r**3)*x(1))/(1.0d0-2.0d0*x(0)/r)
  ni1=2.0d0/r**2
  ni2=1.0d0-2.0d0*x(0)/r
  ni3=x(0)+4.0d0*pi*(r**3)*x(1)
  ni4=(m_linha+4.0d0*pi*r**2*(r*pvl(i)+3.0d0*x(1)))/ni2
  ni5=-2.0d0*m_linha/r+ni1*x(0)
  nill(i)=ni1*ni4-(2.0*ni1/r)*ni3/ni2-ni1*ni3*ni5/ni2**2
  lamb(i)=-log(ni2)
  lambl(i)=-ni5/ni2

  call rk4sys(n, hh, x, r, K, nn)

end do

MM=mv(nsteps-1)
p0=pv(0)

```

```

ni0=ni(0)

print *, "final grid point at ",rv(j)

do i=0, j
  write(14,fmt='(2(e12.5,1x))') rv(i), mv(i)
  write(16,fmt='(2(e12.5,1x))') rv(i), pv(i)
  write(18,fmt='(2(e12.5,1x))') rv(i), rhov(i)
  write(11,fmt='(2(e12.5,1x))') rv(i), ni(i)
  write(13,fmt='(2(e12.5,1x))') rv(i), nil(i)
  write(15,fmt='(2(e12.5,1x))') rv(i), nill(i)
  write(17,fmt='(2(e12.5,1x))') rv(i), lamb(i)
  write(19,fmt='(2(e12.5,1x))') rv(i), lambl(i)
  write(12,fmt='(2(e12.5,1x))') rv(i), pvl(i)
end do

close(10)
close(14)
close(16)
close(18)
close(11)
close(13)
close(15)
close(17)
close(19)

end subroutine tov_rot_z3

```

## Rotina tov\_rot\_z4

```

subroutine tov_rot_z4(nn,K,rhob0,hh,mv,pv,rhov,pvl,ni,nil,nill,lamb,lambl,nsteps,ns,RR)
implicit none
integer, parameter :: dp= SELECTED_REAL_KIND(15,300)
integer, parameter :: n = 2
integer :: i, j, nsteps, ns
real(kind= dp) :: rv(0:nsteps), pv(0:nsteps), mv(0:nsteps), rhov(0:nsteps), pvl(0:nsteps), lamb(0:nsteps)
real(kind= dp) :: ni(0:nsteps), nil(0:nsteps), nill(0:nsteps), lambl(0:nsteps)
real(kind= dp), parameter :: a = 0.0d0, b =20.0d0
real(kind= dp) :: d, e, f, rhob0, rhob, r, rho, h, RR1, hh, m_linha, MM, p1
real(kind= dp) :: pi, K, nn, Rg, prg, RR, prg_linha, delta_ni, ni0, p0
real(kind= dp) :: ni1, ni2, ni3, ni4, ni5, p2, ni22, rho2, p4, ni44
real(kind= dp) :: rmin
real(kind= dp) :: x(0:n)

open(10,file="Test.dat")
open(14,file="RaioMassa.dat")
open(16,file="RaioPressao.dat")
open(18,file="RaioDensidade.dat")
open(11,file="RaioNi.dat")

```

```

open(13,file="RaioNiLinha.dat")
open(15,file="RaioNi2Linhas.dat")
open(17,file="RaioLambda.dat")
open(19,file="RaioLambdaLinha.dat")
open(12,file="RaioPressaoLinha.dat")

pi=4.0d0*atan(1.0d0)
rhob=rhob0
d=0.0d0
e=K*(rhob)**(1.0d0+1.0d0/nn)
f=0.0d0
h = (b - a)/nsteps
r=0.0
j=nsteps-1
rho=rhob+nn*e

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!Expansao em serie para TOV

p2=- (4.0d0*pi/3.0d0)*(rho+e)*(rho+3.0d0*e)
rho2=p2*(rho+e)/((1.0d0+1.0d0/nn)*e)
ni22=(8.0d0*pi/3.0d0)*(rho+3.0d0*e)
p4=- (2.0d0*pi/5.0d0)*(rho+e)*(rho2+5.0d0*p2)- (2.0d0*pi/3.0d0)*(rho2+p2)*(rho+3.0d0*e)-
(32.0d0*pi**2)/9.0d0)*rho*(rho+e)*(rho+3.0d0*e)
ni44=(4.0d0*pi/5.0d0)*(rho2+5.0d0*p2)+((64.0d0*pi**2)/9.0d0)*rho*(rho+3.0d0*e)

do i=0, ns-1
  r=(i)*h
  rv(i)=r

  pv(i) = e + p2*(r**2)/2.0d0 + p4*(r**4)/4.0d0
  rhov(i) = rho + rho2*(r**2)/2.0d0
  mv(i) = 4.0d0*pi*rho*(r**3)/3.0d0 + 2.0d0*pi*rho2*(r**5)/5.0d0
  ni(i) = f + ni22*r**2/2.0d0 + ni44*r**4/4.0d0
  nil(i) = ni22*r + ni44*r**3
  nill(i) = ni22 + 3.0d0*ni44*r**2
  pvl(i) = p2*r + p4*r**3
  lamb(i)=-log(1.0d0 - (8.0d0/3.0d0)*pi*rho*(r**2) - (4.0d0/5.0d0)*pi*rho2*(r**4))
  lamb1(i)=(16.0d0*pi*rho*r/3.0d0+16.0d0*pi*rho2*r**3/5.0d0)/(1.0d0-(8.0d0/3.0d0)*pi*rho*(r**2)-
(4.0d0/5.0d0)*pi*rho2*(r**4))

end do

print *, "loop 1: end of series expansion for TOV at r = ",r

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!1a Integracao numerica para TOV

x = (/mv(ns-1), pv(ns-1), ni(ns-1)/)

do i=ns, j
  r = r + h
  call rk4sys(n,h,x,r,K,nn)
  if(x(1)<=0.0d0) then

```

```

        print *, "found surface"
        go to 50
    end if
    rhob = (x(1)/K)**(nn/(nn+1.0d0))
    rho=rhob+nn*x(1)
    p1=x(1)*5.54d5
    prg=x(1)
    prg_linha=-((1.0d0/r**2)*(x(0) + 4.0d0*pi*(r**3)*x(1))*(rho + x(1)))/(1.0d0-2.0d0*x(0)/r)
    write(10,fmt='(4(e12.5,1x))') r, x(0), x(1), rho
end do

50 continue

Rg=r-h
RR=Rg-(nn+1.0d0)*(prg/prg_linha)

print *, "R = ", RR

RR1=RR*1.4767d0

delta_ni=log(1.0d0-2.0d0*x(0)/RR)-x(2)
rhob=rhob0
d=0.0d0
e=K*(rhob)**(1.0d0+1.0d0/nn)
f=delta_ni
rho=rhob+nn*e
rmin=1.0d-6*RR
hh=(RR-rmin)/(nsteps-1)

print *, "hh = ", hh
print *, "(nsteps-1)*hh = ", (nsteps-1)*hh

j=nsteps-1

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! 2o. loop para TOV
! Recalculando a expansao em série com novo passo
! Começando em rmin = 1e-6*RR

do i=0, ns-1
    r=rmin+i*hh
    rv(i)=r

    pv(i) = e + p2*(r**2)/2.0d0 + p4*(r**4)/4.0d0
    rhov(i) = rho + rho2*(r**2)/2.0d0
    mv(i) = 4.0d0*pi*rho*(r**3)/3.0d0 + 2.0*pi*rho2*(r**5)/5.0d0
    ni(i) = f + ni22*r**2/2.0d0 + ni44*r**4/4.0d0
    nil(i) = ni22*r + ni44*r**3
    nill(i) = ni22 + 3.0d0*ni44*r**2
    pvl(i) = p2*r + p4*r**3
    lamb(i)=-log(1.0d0 - (8.0d0/3.0d0)*pi*rho*(r**2) - (4.0d0/5.0d0)*pi*rho2*(r**4))
    lambl(i)=(16.0d0*pi*rho*r/3.0d0+16.0d0*pi*rho2*r**3/5.0d0)/(1.0d0-(8.0d0/3.0d0)*pi*rho*(r**2)-
(4.0d0/5.0d0)*pi*rho2*(r**4))

```

```

end do

print *, "loop2: end of series expansion for TOV at r = ",r

x = (/mv(ns-1), pv(ns-1), ni(ns-1)/)

do i=ns-1, j
  r =rmin+i*hh
  rhob = (x(1)/K)**(nn/(nn+1.0d0))
  rho=rhob+nn*x(1)
  rv(i)=r
  mv(i)=x(0)
  pv(i)=x(1)
  rhov(i)=rho
  m_linha=4.0d0*pi*(r**2)*rho
  ni(i)=x(2)
  pv1(i)=-((1.0d0/r**2)*(x(0) + 4.0d0*pi*(r**3)*x(1))*(rho + x(1)))/(1.0d0-2.0d0*x(0)/r)
  nil(i)=2.0d0*(1.0d0/r**2)*(x(0) + 4.0d0*pi*(r**3)*x(1))/(1.0d0-2.0d0*x(0)/r)
  ni1=2.0d0/r**2
  ni2=1.0d0-2.0d0*x(0)/r
  ni3=x(0)+4.0d0*pi*(r**3)*x(1)
  ni4=(m_linha+4.0d0*pi*r**2*(r*pv1(i)+3.0d0*x(1)))/ni2
  ni5=-2.0d0*m_linha/r+ni1*x(0)
  nill(i)=ni1*ni4-(2.0d0*ni1/r)*ni3/ni2-ni1*ni3*ni5/ni2**2
  lamb(i)=-log(ni2)
  lambl(i)=-ni5/ni2

  call rk4sys(n, hh, x, r, K, nn)

end do

MM=mv(nsteps-1)
p0=pv(0)
ni0=ni(0)

print *, "final grid point at ",rv(j)

do i=0, j
  write(14,fmt='(2(e12.5,1x))') rv(i), mv(i)
  write(16,fmt='(2(e12.5,1x))') rv(i), pv(i)
  write(18,fmt='(2(e12.5,1x))') rv(i), rhov(i)
  write(11,fmt='(2(e12.5,1x))') rv(i), ni(i)
  write(13,fmt='(2(e12.5,1x))') rv(i), nil(i)
  write(15,fmt='(2(e12.5,1x))') rv(i), nill(i)
  write(17,fmt='(2(e12.5,1x))') rv(i), lamb(i)
  write(19,fmt='(2(e12.5,1x))') rv(i), lambl(i)
  write(12,fmt='(2(e12.5,1x))') rv(i), pv1(i)
end do

close(10)
close(14)
close(16)
close(18)
close(11)

```



```

close(13)
close(15)
close(17)
close(19)

end subroutine tov_rot_z4

```

## Rotina rk4sys

```

subroutine rk4sys(n,h,x,r,K,nn)
  implicit none
  integer, parameter :: dp= SELECTED_REAL_KIND(15,300)
  real(kind= dp) :: x(0:n)
  real(kind= dp), allocatable :: y(:), f(:, :)
  integer :: i, n
  real(kind= dp) :: h,r,K,nn
  allocate (y(0:n), f(0:n,4))

  call xpsys(n,x,f(0,1),r,K,nn)

  do i = 0,n
    y(i) = x(i) + 0.5d0*h*f(i,1)
    enddo
    call xpsys(n,y,f(0,2),r+0.5d0*h,K,nn)

  do i = 0,n
    y(i) = x(i) + 0.5d0*h*f(i,2)
    enddo
    call xpsys(n,y,f(0,3),r+0.5d0*h,K,nn)

  do i = 0,n
    y(i) = x(i) + h*f(i,3)
    enddo
    call xpsys(n,y,f(0,4),r+h,K,nn)

  do i = 0,n
    x(i) = x(i) + (h/6.0d0)* (f(i,1) + 2.0d0*(f(i,2) + f(i,3)) + f(i,4))
  enddo
end subroutine rk4sys

```

## Rotina xpsys

```
subroutine xpsys(n,x,f,r,K,nn)
implicit none
integer, parameter :: dp= SELECTED_REAL_KIND(15,300)
real(kind= dp), dimension (0:n) :: x, f
real(kind= dp) :: fp,fp1,fp2,fp3,fp4,pi,rho,r,rhob,K,nn
integer n

pi=4.0d0*atan(1.0d0)

if(x(1)<=0.0d0) then
    x(1) = 0.0d0
end if

rhob = (x(1)/K)**(nn/(nn+1.0))
rho=rhob+nn*x(1)

fp1 = 1.0d0/r**2
fp2 = rho + x(1)
fp3 = x(0) + 4.0d0*pi*(r**3)*x(1)
fp4 = 1.0d0-2.0d0*x(0)/r
fp = fp1*fp3/fp4

f(0) = 4.0d0*pi*(r**2)*rho
f(2) = 2.0d0*fp
f(1) = -0.5d0*f(2)*fp2
end subroutine xpsys
```

## ANEXO C - PERTURBAÇÕES DA SUPERFÍCIE DA ESTRELA ATÉ

$\frac{R}{2}$

### Rotina s\_m\_rot

```
subroutine s_m_rot(nn,K,rhob0,l,h1v1,kv1,wv1,xv1,h1v2,kv2,wv2,xv2,h1v3,kv3,wv3,xv3,mF1,mF2,mF3,cons,omega,ns,ns2)
implicit none
integer, parameter :: dp= SELECTED_REAL_KIND(15,300)
integer, parameter :: n1 = 3, ns3=2
integer :: i, j, q, ns, ns2
real(kind= dp) :: l, d, e, f, g, rhob0, r, hh, pi, K, nn, RR, omega, h1
real(kind= dp) :: ni2, nil2, nill2, lb1, lbl1, M2, p2, rho2, pl
real(kind= dp) :: ni2n, nil2n, nill2n, lb1n, lbl1n, M2n, p2n, rho2n, pln
real(kind= dp) :: h1v1(0:ns), kv1(0:ns), wv1(0:ns), xv1(0:ns), rv1(0:ns)
real(kind= dp) :: h1v2(0:ns), kv2(0:ns), wv2(0:ns), xv2(0:ns), rv2(0:ns)
real(kind= dp) :: h1v3(0:ns), kv3(0:ns), wv3(0:ns), xv3(0:ns), rv3(0:ns)
real(kind= dp) :: h1v1l(0:ns), kv1l(0:ns), wv1l(0:ns), xv1l(0:ns)
real(kind= dp) :: h1v2l(0:ns), kv2l(0:ns), wv2l(0:ns), xv2l(0:ns)
real(kind= dp) :: h1v3l(0:ns), kv3l(0:ns), wv3l(0:ns), xv3l(0:ns)
real(kind= dp) :: pv(0:ns2), mv(0:ns2), rhov(0:ns2), lb(0:ns2), lbl(0:ns2)
real(kind= dp) :: ni(0:ns2), nil(0:ns2), nill(0:ns2), pvl(0:ns2), rv(0:ns2)
real(kind= dp) :: x(0:n1), Yl(0:n1), mF1(0:n1), mF2(0:n1), mF3(0:n1)
real(kind= dp) :: cons(0:4)

open(20,file="TestY1t.dat")
open(21,file="TestY2t.dat")
open(22,file="TestY3t.dat")
open(23,file="TestY1lt.dat")
open(24,file="TestY2lt.dat")
open(25,file="TestY3lt.dat")

pi= 4.0d0*atan(1.0d0)

call tov_rot_z4(nn,K,rhob0,hh,mv,pv,rhov,pvl,ni,nil,nill,lb,lbl,ns2,ns3,RR)

! d=H10, e=K0, f=W0, g=X0
! x(0)=H1, x(1)=K, x(2)=W, x(3)=X
! Laço 1 (obter Y1)

d=1.0d0
e=0.0d0
f=0.0d0
g=0.0d0
x = (/d, e, f, g/)
h1=-hh
r=RR
q=ns2-1
```

```

h1v1(0)=d
kv1(0)=e
wv1(0)=f
xv1(0)=g
rv1(0)=RR

h1v1l(0)=0.0d0
kv1l(0)=0.0d0
wv1l(0)=0.0d0
xv1l(0)=0.0d0

M2=mv(ns2-1)
p2=pv(ns2-1)
rho2=rhov(ns2-1)
pl=pvl(ns2-1)
ni2=ni(ns2-1)
nil2=nil(ns2-1)
nill2=nill(ns2-1)
lb1=lb(ns2-1)
lb1l=lb1(ns2-1)

cons(0)=mv(ns2-1) !MR
cons(1)=pv(ns2-1) !pR
cons(2)=ni(ns2-1) !niR
cons(3)=lb(ns2-1) !lbR
cons(4)=RR !RR

M2n=mv(ns2-2)
p2n=pv(ns2-2)
rho2n=rhov(ns2-2)
pln=pvl(ns2-2)
ni2n=ni(ns2-2)
nil2n=nil(ns2-2)
nill2n=nill(ns2-2)
lb1n=lb(ns2-2)
lb1ln=lb1(ns2-2)

j=ns

do i=1, j
  q=q-1
  call rk4sys_p1(n1,h1,x,r,nn,ni2,nil2,nill2,lb1,lb1l,M2,p2,rho2,pl,ni2n,nil2n,nill2n,lb1n,lb1ln,M2n,p2n,rho2n,pln,ome

  M2=mv(q)
  p2=pv(q)
  rho2=rhov(q)
  pl=pvl(q)
  ni2=ni(q)
  nil2=nil(q)
  nill2=nill(q)
  lb1=lb(q)
  lb1l=lb1(q)

```

```

M2n=mv(q-1)
p2n=pv(q-1)
rho2n=rhov(q-1)
p1n=pv1(q-1)
ni2n=ni(q-1)
nil2n=nil(q-1)
nill2n=nill(q-1)
lb1n=lb(q-1)
lbl1n=lbl(q-1)

r=RR-i*hh

h1v1(i)=x(0)
kv1(i)=x(1)
wv1(i)=x(2)
xv1(i)=x(3)
rv1(i)=r

h1v1l(i)=Y1(0)
kv1l(i)=Y1(1)
wv1l(i)=Y1(2)
xv1l(i)=Y1(3)

mF1(0)=h1v1(i)
mF1(1)=kv1(i)
mF1(2)=wv1(i)
mF1(3)=xv1(i)
end do

print*, ""
print*, ""
print*, ""
print*, "Ultimo r (Superficie Metade):" , r
print*, ""
print*, ""
print*, ""

do i=0, j
  write(20,fmt='(5(f20.14,1x))') h1v1(i), kv1(i), wv1(i), xv1(i), rv1(i)
  write(23,fmt='(5(f20.14,1x))') h1v1l(i), kv1l(i), wv1l(i), xv1l(i), rv1(i)
end do

! Laço 2 (obter Y2)

d=0.0d0
e=1.0d0
f=0.0d0
g=0.0d0
x = (/d, e, f, g/)
h1=-hh
r=RR

```

```

q=ns2-1

h1v2(0)=d
kv2(0)=e
wv2(0)=f
xv2(0)=g
rv2(0)=RR

h1v2l(0)=0.0d0
kv2l(0)=0.0d0
wv2l(0)=0.0d0
xv2l(0)=0.0d0

M2=mv(ns2-1)
p2=pv(ns2-1)
rho2=rhov(ns2-1)
pl=pvl(ns2-1)
ni2=ni(ns2-1)
nil2=nil(ns2-1)
nill2=nill(ns2-1)
lb1=lb(ns2-1)
lb1l=lb1(ns2-1)

M2n=mv(ns2-2)
p2n=pv(ns2-2)
rho2n=rhov(ns2-2)
pln=pvl(ns2-2)
ni2n=ni(ns2-2)
nil2n=nil(ns2-2)
nill2n=nill(ns2-2)
lb1n=lb(ns2-2)
lb1ln=lb1(ns2-2)

j=ns

do i=1, j
  q=q-1
  call rk4sys_p1(n1,h1,x,r,nn,ni2,nil2,nill2,lb1,lb1l,M2,p2,rho2,pl,ni2n,nil2n,nill2n,lb1n,lb1ln,M2n,p2n,rho2n,pln,ome

  M2=mv(q)
  p2=pv(q)
  rho2=rhov(q)
  pl=pvl(q)
  ni2=ni(q)
  nil2=nil(q)
  nill2=nill(q)
  lb1=lb(q)
  lb1l=lb1(q)

  M2n=mv(q-1)
  p2n=pv(q-1)
  rho2n=rhov(q-1)
  pln=pvl(q-1)
  ni2n=ni(q-1)

```

```

nil2n=nil(q-1)
nil12n=nil1(q-1)
lb1n=lb(q-1)
lb11n=lb1(q-1)

r=RR-i*hh

h1v2(i)=x(0)
kv2(i)=x(1)
wv2(i)=x(2)
xv2(i)=x(3)
rv2(i)=r

h1v21(i)=Y1(0)
kv21(i)=Y1(1)
wv21(i)=Y1(2)
xv21(i)=Y1(3)

mF2(0)=h1v2(i)
mF2(1)=kv2(i)
mF2(2)=wv2(i)
mF2(3)=xv2(i)
end do

do i=0, j
  write(21,fmt='(5(f20.14,1x))') h1v2(i), kv2(i), wv2(i), xv2(i), rv2(i)
  write(24,fmt='(5(f20.14,1x))') h1v21(i), kv21(i), wv21(i), xv21(i), rv2(i)
end do

! Laço 3 (obter Y3)

d=0.0d0
e=0.0d0
f=1.0d0
g=0.0d0
x = (/d, e, f, g/)
h1=-hh
r=RR
q=ns2-1

h1v3(0)=d
kv3(0)=e
wv3(0)=f
xv3(0)=g
rv3(0)=RR

h1v31(0)=0.0d0
kv31(0)=0.0d0
wv31(0)=0.0d0
xv31(0)=0.0d0

M2=mv(ns2-1)
p2=pv(ns2-1)
rho2=rhov(ns2-1)

```

```

pl=pvl(ns2-1)
ni2=ni(ns2-1)
nil2=nil(ns2-1)
nill2=nill(ns2-1)
lb1=lb(ns2-1)
lb11=lbl(ns2-1)

M2n=mv(ns2-2)
p2n=pv(ns2-2)
rho2n=rhov(ns2-2)
pln=pvl(ns2-2)
ni2n=ni(ns2-2)
nil2n=nil(ns2-2)
nill2n=nill(ns2-2)
lb1n=lb(ns2-2)
lb11n=lbl(ns2-2)

j=ns

do i=1, j
  q=q-1
  call rk4sys_p1(n1,h1,x,r,nn,ni2,nil2,nill2,lb1,lb11,M2,p2,rho2,pl,ni2n,nil2n,nill2n,lb1n,lb11n,M2n,p2n,rho2n,pln,ome

  M2=mv(q)
  p2=pv(q)
  rho2=rhov(q)
  pl=pvl(q)
  ni2=ni(q)
  nil2=nil(q)
  nill2=nill(q)
  lb1=lb(q)
  lb11=lbl(q)

  M2n=mv(q-1)
  p2n=pv(q-1)
  rho2n=rhov(q-1)
  pln=pvl(q-1)
  ni2n=ni(q-1)
  nil2n=nil(q-1)
  nill2n=nill(q-1)
  lb1n=lb(q-1)
  lb11n=lbl(q-1)

  r=RR-i*hh

  h1v3(i)=x(0)
  kv3(i)=x(1)
  ww3(i)=x(2)
  xv3(i)=x(3)
  rv3(i)=r

  h1v3l(i)=Yl(0)
  kv3l(i)=Yl(1)
  ww3l(i)=Yl(2)

```



```

xv3l(i)=Yl(3)

mF3(0)=h1v3(i)
mF3(1)=kv3(i)
mF3(2)=wv3(i)
mF3(3)=xv3(i)
end do

do i=0, j
  write(22,fmt='(5(f20.14,1x))') h1v3(i), kv3(i), wv3(i), xv3(i), rv3(i)
  write(25,fmt='(5(f20.14,1x))') h1v3l(i), kv3l(i), wv3l(i), xv3l(i), rv3l(i)
end do

close(20)
close(21)
close(22)
close(23)
close(24)
close(25)

end subroutine s_m_rot

```

## Rotina rk4sys\_p1

```

subroutine rk4sys_p1(n,h,x,r,nn,ni2,nil,nill,lb1,lb1l,M2,p2,rho2,pl,ni2n,nil2n,nill2n,lb1n,lb1ln,M2n,p2n,rho2n,pln,omega)
implicit none
integer, parameter :: dp= SELECTED_REAL_KIND(15,300)
real(kind= dp) :: x(0:n), Yl(0:n)
real(kind= dp), allocatable :: y(:), f(:, :)
integer :: i, n
real(kind= dp) :: l, h, r, nn, ni2, nil, nill, lb1, lb1l, M2, p2, rho2, pl, omega
real(kind=dp) :: ni2n, nil2n, nill2n, lb1n, lb1ln, M2n, p2n, rho2n, pln
real(kind=dp) :: ni2x, nilx, nillx, lb1x, lb1lx, M2x, p2x, rho2x, plx

allocate (y(0:n), f(0:n,4))
call xpsys_p1(n,x,f(0,1),r,nn,ni2,nil,nill,lb1,lb1l,M2,p2,rho2,pl,omega,l)

do i = 0,n
  y(i) = x(i) + 0.5*h*f(i,1)
  enddo
  ni2x = 0.5*(ni2+ni2n)
  nilx = 0.5*(nil+nil2n)
  nillx = 0.5*(nill+nill2n)
  lb1x = 0.5*(lb1+lb1n)
  lb1lx = 0.5*(lb1l + lb1ln)
  M2x = 0.5*(M2 + M2n)
  p2x = 0.5*(p2 + p2n)
  rho2x = 0.5*(rho2 + rho2n)
  plx = 0.5*(pl + pln)

```

```

        call xpsys_p1(n,y,f(0,2),r+0.5*h,nn,ni2x,nilx,nillx,lb1x,lb11x,M2x,p2x,rho2x,plx,omega,l)

do i = 0,n
  y(i) = x(i) + 0.5*h*f(i,2)
enddo
  call xpsys_p1(n,y,f(0,3),r+0.5*h,nn,ni2x,nilx,nillx,lb1x,lb11x,M2x,p2x,rho2x,plx,omega,l)

do i = 0,n
  y(i) = x(i) + h*f(i,3)
enddo
  call xpsys_p1(n,y,f(0,4),r+h,nn,ni2n,nil2n,nill2n,lb1n,lb11n,M2n,p2n,rho2n,pln,omega,l)

do i = 0,n
x(i) = x(i) + (h/6.0)* (f(i,1) + 2.0*(f(i,2) + f(i,3)) + f(i,4))
enddo

Y1(0)= f(0,1)
Y1(1)= f(1,1)
Y1(2)= f(2,1)
Y1(3)= f(3,1)

end subroutine rk4sys_p1

```

## Rotina xpsys\_p1

```

subroutine xpsys_p1(n,x,f,r,nn,ni,nil,nill,lamb,lamb1,M,p,rho,pl,omega,l)
implicit none
integer, parameter :: dp= SELECTED_REAL_KIND(15,300)
integer :: n
real(kind= dp), dimension (0:n) :: x, f
real(kind= dp) :: f0, f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11, f12, f13, f14, r
real(kind= dp) :: f15, f16, f17, f18, f19, f20, f21, f22, f23, f24, f25, f26
real(kind= dp) :: l, p, M, rho, ni, nil, nill, lamb, lamb1, pl, omega, nn, V, H0, pi

pi= 4.0*atan(1.0)

! x(0)=H1, x(1)=K, x(2)=W, x(3)=X,
! nn= indice adiabático
! p_linha= dp/dr=pl
! ni_linha= d(ni)/dr=nil
! lambda_linha= d(lambda)/dr=lambda_linha1
! ni_2linhas= d(ni_linha)/dr=nill

f0= 0.5*l*(1+1.0)
f1= 1.0/r
f2= exp(lamb)
f3= -f1*(1+1.0+2.0*M*f2*f1+4.0*pi*(r**2)*f2*(p-rho))
f4= 3.0*M+0.5*(1+2.0)*(1-1.0)*r+4.0*pi*(r**3)*p

```

```

f5= 8.0*pi*(r**3)*(exp(-0.5*ni))
f6= f0*(M+4.0*pi*(r**3)*p)-(omega**2)*(r**3)*(exp(-lamb-ni))
f7= (M+4.0*pi*(r**3)*p)*(3.0*M-r+4.0*pi*(r**3)*p)
f8= 0.5*(1+2.0)*(1-1.0)*r-(omega**2)*(r**3)*(exp(-ni))-f1*f2*f7
H0= (f5*x(3)-f6*x(0)+f8*x(1))/f4
f9= (omega**2)*(rho+p)*(exp(-0.5*ni))
f10= f1*p1*(exp(0.5*(ni-lamb)))
f11= 0.5*(rho+p)*(exp(0.5*ni))
V= (x(3)+f10*x(2)-f11*H0)/f9
f12= f0*f1
f13= (1+1.0)*f1-0.5*nil
f14= 8.0*pi*(rho+p)*exp(0.5*lamb)*f1
f14= 8.0*pi*(rho+p)*exp(0.5*lamb)*f1
f15= (1+1.0)*f1
f16= r*(exp(0.5*lamb))

f17= ((nn/(1.0+nn))*(exp(-0.5*ni)))/p

f18= 2.0*f0/(r**2)
f19= (rho+p)*(exp(0.5*ni))
f20= 0.5*(f1-0.5*nil)
f21= 0.5*(r*(omega**2)*(exp(-ni))+f0*f1)
f22= 0.5*(1.5*nil-f1)
f23= f0*nil/(r**2)
f24= 4.0*pi*(rho+p)*(exp(0.5*lamb))+omega**2*exp(0.5*lamb-ni)
f25= 0.5*exp(-0.5*lamb)*(-2.0*f1*nil-0.5*lamb*nil+nil1)
f26= f24-f25

! f(0)=>H1, f(1)=>K, f(2)=>W, f(3)=>X

f(0) = f3*x(0)+f1*f2*(H0+x(1))-16.0*pi*(rho+p)*V)
f(1) = f1*H0+f12*x(0)-f13*x(1)-f14*x(2)
f(2) = -f15*x(2)+f16*(f17*x(3)-f18*V+0.5*H0+x(1))
f(3) = -1*f1*x(3)+f19*(f20*H0+f21*x(0)+f22*x(1)-f23*V-f1*f26*x(2))

end subroutine xpsys_p1

```



## ANEXO D- CÁLCULO DAS PERTURBAÇÕES DO CENTRO ATÉ $\frac{R}{2}$

### Rotina c\_m\_rot

```
subroutine c_m_rot(nn,K,rhob,l,h1v4,kv4,wv4,xv4,h1v5,kv5,wv5,xv5,hh,rmin,mF4,mF5,omega,ns,ns2,mv,pv,ni,lb)
implicit none
integer, parameter :: n1 = 3, n3=2
integer, parameter :: dp= SELECTED_REAL_KIND(15,300)
integer :: i, j, ns, ns2
real(kind= dp) :: l, d, e, f, g, rhob, r, hh, pi, K, nn, omega, H6, K6, W6, X6, H7, K7, W7, X7
real(kind= dp) :: ni2, nil2, nill2, lb1, lbl1, M2, p2, rho2, pl
real(kind= dp) :: ni2n, nil2n, nill2n, lb1n, lbl1n, M2n, p2n, rho2n, pln
real(kind= dp) :: H1l, K1, W1, X1
real(kind= dp) :: h1v4(0:ns), kv4(0:ns), wv4(0:ns), xv4(0:ns), rv4(0:ns)
real(kind= dp) :: h1v5(0:ns), kv5(0:ns), wv5(0:ns), xv5(0:ns), rv5(0:ns),X5(0:n3)
real(kind= dp) :: pv(0:ns2), mv(0:ns2), rhov(0:ns2), lb(0:ns2), lbl(0:ns2)
real(kind= dp) :: ni(0:ns2), nil(0:ns2), nill(0:ns2), pvl(0:ns2)
real(kind= dp) :: H14(0:n3),K4(0:n3),W4(0:n3),X4(0:n3),H15(0:n3),K5(0:n3),W5(0:n3)
real(kind= dp) :: x(0:n1), Y1(0:n1), mF4(0:n1), mF5(0:n1)
real(kind= dp) :: h1v4l(0:ns), kv4l(0:ns), wv4l(0:ns), xv4l(0:ns), rv4l(0:ns)
real(kind= dp) :: h1v5l(0:ns), kv5l(0:ns), wv5l(0:ns), xv5l(0:ns), rv5l(0:ns)
real(kind= dp) :: rmin

pi=4.0*atan(1.0)

call P_C(nn,K,rhob,l,hh,omega,mv,pv,rhov,pvl,ni,nil,nill,lb,lbl,H14,K4,W4,X4,H15,K5,W5,X5,n3,ns2,rmin)
open(10,file="TestY4t9.dat")
open(11,file="TestY5t9.dat")
open(12,file="TestY4lt9.dat")
open(13,file="TestY5lt9.dat")

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Laço 1 (obter Y4)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

do i=0, n3-1
r=rmin+i*hh
h1v4(i)=H14(i)
kv4(i)=K4(i)
wv4(i)=W4(i)
xv4(i)=X4(i)
rv4(i)=r
h1v4l(i)=0.0
kv4l(i)=0.0
wv4l(i)=0.0
xv4l(i)=0.0
end do
```

```

print *, "r = ", r, i

d=H14(n3-1)
e=K4(n3-1)
f=W4(n3-1)
g=X4(n3-1)
x = (/d, e, f, g/)
M2=mv(n3-1)
p2=pv(n3-1)
rho2=rhov(n3-1)
pl=pvl(n3-1)
ni2=ni(n3-1)
nil2=nil(n3-1)
nill2=nill(n3-1)
lb1=lb(n3-1)
lb11=lbl(n3-1)

M2n=mv(n3)
p2n=pv(n3)
rho2n=rhov(n3)
pln=pvl(n3)
ni2n=ni(n3)
nil2n=nil(n3)
nill2n=nill(n3)
lb1n=lb(n3)
lb11n=lbl(n3)

j=ns-1

do i=n3, j
call rk4sys_p1(n1, hh, x, r, nn, ni2, nil2, nill2, lb1, lb11, M2, p2, rho2, pl, ni2n, nil2n, nill2n, lb1n, lb11n, M2n, p2n,
rho2n, pln, omega, l, Y1)
    M2=mv(i)
    p2=pv(i)
    rho2=rhov(i)
    pl=pvl(i)
    ni2=ni(i)
    nil2=nil(i)
    nill2=nill(i)
    lb1=lb(i)
    lb11=lbl(i)

    M2n=mv(i+1)
    p2n=pv(i+1)
    rho2n=rhov(i+1)
    pln=pvl(i+1)
    ni2n=ni(i+1)
    nil2n=nil(i+1)
    nill2n=nill(i+1)
    lb1n=lb(i+1)
    lb11n=lbl(i+1)

```

```

        r=rmin+i*hh

        h1v4(i)=x(0)
        kv4(i)=x(1)
        wv4(i)=x(2)
        xv4(i)=x(3)
        rv4(i)=r
        h1v4l(i)=Yl(0)
        kv4l(i)=Yl(1)
        wv4l(i)=Yl(2)
        xv4l(i)=Yl(3)

        mF4(0)=h1v4(i)
        mF4(1)=kv4(i)
        mF4(2)=wv4(i)
        mF4(3)=xv4(i)

    end do

    print*, ""
    print*, ""
    print*, ""
    print*, "Ultimo r (Centro Metade):" , r
    print*, ""
    print*, ""
    print*, ""

    do i=0, j
        write(10,fmt='(5(f15.10,1x))') h1v4(i), kv4(i), wv4(i), xv4(i), rv4(i)
        write(12,fmt='(5(f15.10,1x))') h1v4l(i), kv4l(i), wv4l(i), xv4l(i), rv4(i)
    end do

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Laço 2 (obter Y5)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

    do i=0, n3-1
        r=rmin+i*hh
        h1v5(i)=H15(i)
        kv5(i)=K5(i)
        wv5(i)=W5(i)
        xv5(i)=X5(i)
        rv5(i)=r
        h1v5l(i)=0.0
        kv5l(i)=0.0
        wv5l(i)=0.0
        xv5l(i)=0.0
    end do

    d=H15(n3-1)
    e=K5(n3-1)
    f=W5(n3-1)

```

```

g=X5(n3-1)
x = (/d, e, f, g/)
M2=mv(n3-1)
p2=pv(n3-1)
rho2=rhov(n3-1)
pl=pvl(n3-1)
ni2=ni(n3-1)
nil2=nil(n3-1)
nill2=nill(n3-1)
lb1=lb(n3-1)
lb11=lbl(n3-1)

M2n=mv(n3)
p2n=pv(n3)
rho2n=rhov(n3)
pln=pvl(n3)
ni2n=ni(n3)
nil2n=nil(n3)
nill2n=nill(n3)
lb1n=lb(n3)
lb11n=lbl(n3)

j=ns-1

do i=n3, j
call rk4sys_p1(n1,hh,x,r,nm,ni2,nil2,nill2,lb1,lb11,M2,p2,rho2,pl,ni2n,nil2n,nill2n,lb1n,lb11n,M2n,p2n,
rho2n,pln,omega,l,Y1)
    M2=mv(i)
    p2=pv(i)
    rho2=rhov(i)
    pl=pvl(i)
    ni2=ni(i)
    nil2=nil(i)
    nill2=nill(i)
    lb1=lb(i)
    lb11=lbl(i)

    M2n=mv(i+1)
    p2n=pv(i+1)
    rho2n=rhov(i+1)
    pln=pvl(i+1)
    ni2n=ni(i+1)
    nil2n=nil(i+1)
    nill2n=nill(i+1)
    lb1n=lb(i+1)
    lb11n=lbl(i+1)

    r=rmin+i*hh

    h1v5(i)=x(0)
    kv5(i)=x(1)
    wv5(i)=x(2)
    xv5(i)=x(3)
    rv5(i)=r

```



```

        h1v5l(i)=Y1(0)
        kv5l(i)=Y1(1)
        wv5l(i)=Y1(2)
        xv5l(i)=Y1(3)

        mF5(0)=h1v5(i)
        mF5(1)=kv5(i)
        mF5(2)=wv5(i)
        mF5(3)=xv5(i)

    end do

    do i=0, j
        write(11,fmt='(5(f15.10,1x))') h1v5(i), kv5(i), wv5(i), xv5(i), rv5(i)
        write(13,fmt='(5(f15.10,1x))') h1v5l(i), kv5l(i), wv5l(i), xv5l(i), rv5(i)
    end do

close(10)
close(11)
close(12)
close(13)

end subroutine c_m_rot

```

## Rotina P\_C

```

subroutine P_C(nn,K,rhob0,l,hh,omega,mv,pv,rhov,pvl,ni,nil,nill,lb,lbl,H11,K1,W1,X1,H12,K2,W2,X2,ns,ns2,rmin)
implicit none
integer, parameter :: dp= SELECTED_REAL_KIND(15,300)
integer :: i, j, ns, ns2
real(kind= dp) :: l, rhob0, r, hh, pi, K, nn, omega, p0, p2, p4, ni0, ni2, ni4, rho2, rho0
real(kind= dp) :: H11(0:ns), K1(0:ns), W1(0:ns), X1(0:ns)
real(kind= dp) :: H12(0:ns), K2(0:ns), W2(0:ns), X2(0:ns)
real(kind= dp), dimension (4,4) :: mH
real(kind= dp) :: pv(0:ns2), mv(0:ns2), rhov(0:ns2), lb(0:ns2), lbl(0:ns2)
real(kind= dp) :: ni(0:ns2), nil(0:ns2), nill(0:ns2), pvl(0:ns2)
real(kind= dp) :: H10111, K0111, W0111, X0111, H101, K01, W01, X01
real(kind= dp) :: H10112, K0112, W0112, X0112, H102, K02, W02, X02
real(kind= dp) :: rmin

pi=4.0d0*atan(1.0d0)

call tov_rot_z3(nn,K,rhob0,hh,p0,p2,p4,ni0,ni2,ni4,rho2,mv,pv,rhov,pvl,ni,nil,nill,lb,lbl,ns2,ns,rmin)

rho0=rhob0+nn*p0

call matriz(mH,nn,p0,p2,p4,ni0,ni2,ni4,rho0,rho2,omega,l)

```

```

!      Laço 1

W01=1.0d0
K01=rho0+p0
X01=(rho0+p0)*exp(0.5d0*ni0)*(((4.0d0*pi/3.0d0)*(rho0+3.0d0*p0)-omega**2*exp(-ni0)/1)*W01+0.5d0*K01)
H101=(2.0d0*1*K01+16.0d0*pi*(rho0+p0)*W01)/(1*(1+1.0d0))

H10111=mH(1,1)*H101+mH(1,2)*K01+mH(1,3)*W01+mH(1,4)*X01
K0111=mH(2,1)*H101+mH(2,2)*K01+mH(2,3)*W01+mH(2,4)*X01
W0111=mH(3,1)*H101+mH(3,2)*K01+mH(3,3)*W01+mH(3,4)*X01
X0111=mH(4,1)*H101+mH(4,2)*K01+mH(4,3)*W01+mH(4,4)*X01

!      print*, "H=", H10111,"K=", K0111,"W=", W0111,"X=", X0111

!      H4=H10111
!      K4=K0111
!      W4=W0111
!      X4=X0111

j=ns-1
r=0.0d0
do i=0, j
    H11(i)=H101
    K1(i)=K01
    W1(i)=W01
    X1(i)=X01
end do

!      Laço 2

W02=1.0d0
K02=-rho0-p0
X02=(rho0+p0)*exp(0.5d0*ni0)*(((4.0d0*pi/3.0d0)*(rho0+3.0d0*p0)-omega**2*exp(-ni0)/1)*W02+0.5d0*K02)
H102=(2.0d0*1*K02+16.0d0*pi*(rho0+p0)*W02)/(1*(1+1.0d0))

H10112=mH(1,1)*H102+mH(1,2)*K02+mH(1,3)*W02+mH(1,4)*X02
K0112=mH(2,1)*H102+mH(2,2)*K02+mH(2,3)*W02+mH(2,4)*X02
W0112=mH(3,1)*H102+mH(3,2)*K02+mH(3,3)*W02+mH(3,4)*X02
X0112=mH(4,1)*H102+mH(4,2)*K02+mH(4,3)*W02+mH(4,4)*X02

!      H5=H10112
!      K5=K0112
!      W5=W0112
!      X5=X0112

j=ns-1
r=0.0d0
do i=0, j
    H12(i)=H102
    K2(i)=K02
    W2(i)=W02
    X2(i)=X02
end do

```

```
end do
```

```
end subroutine P_C
```

## Rotina matriz

```
subroutine matriz(mH,nn,p0,p2,p4,ni0,ni2,ni4,rho0,rho2,omega,l)
implicit none
integer, parameter :: dp= SELECTED_REAL_KIND(15,300)
real(kind= dp), dimension (4,4) :: mG, mF, mH, mJ
real(kind= dp) :: l, p0, p2, p4, ni0, ni2, ni4, rho0, rho2, omega, pi
real(kind= dp) :: qh, qk, qk1, qw1, qx, qx1, q0, q1, q2, q3, q4, nn
integer :: i, j

open(24,file="Inversa.dat")
open(25,file="Produto.dat")
open(26,file="mF.dat")
open(27,file="mJ.dat")

pi=4.0*atan(1.0)
print *, omega

mF(1,1)=0.0
mF(2,1)=-0.25*1*(1+1.0)
mF(3,1)=0.5*(1+3.0)
mF(4,1)=-0.125*1*(1+1.0)*(p0+rho0)*exp(0.5*ni0)
mF(1,2)=-0.25*(p0+rho0)
mF(2,2)=0.5*(1+2.0)
mF(3,2)=-1.0
mF(4,2)=0.0
mF(1,3)=0.5*(p2+(p0+rho0)*omega**2*(1+3.0)*exp(-ni0)/(1*(1+1.0)))
mF(2,3)=4.0*pi*(p0+rho0)
mF(3,3)=-8.0*pi*(p0+rho0)*(1+3.0)/(1*(1+1.0))
mF(4,3)=-(p0+rho0)*exp(0.5*ni0)*(0.25*(1+2.0)*ni2-2.0*pi*(rho0+p0)
-0.5*omega**2*exp(-ni0))
mF(1,4)=0.5*exp(-0.5*ni0)
mF(2,4)=0.0
mF(3,4)=0.0
mF(4,4)=0.5*(1+2.0)

qh=-4.0*((2.0/3.0)*pi*1*(1+1.0)*(rho0+3.0*p0)-omega**2*exp(-ni0))/((1+2.0)*(1-1.0))
qk=-4.0*((8.0*pi/3.0)*rho0+omega**2*exp(-ni0))/((1+2.0)*(1-1.0))
qk1=3.0/(1*(1+1.0))
qw1=8.0*pi*rho0/(3.0*1)
qx=(32.0*pi*exp(-0.5*ni0))/((1+2.0)*(1-1.0))
qx1=(2.0*exp(-0.5*ni0))/(1*(1+1.0)*p0*(1.0+1.0/nn))
q0=0.25*(rho0+p0)
q1=0.5*omega**2*(p0+rho0)*exp(-ni0)
q2=-8.0*pi*(rho0+p0)
```

```

q3=0.25*(rho0+p0)*exp(0.5*ni0)
q4=-0.25*(rho0+p0)*exp(0.5*ni0)*1*(1+1.0)*ni2

mJ(1,1)=q0*qh
mJ(2,1)=0.5*qh
mJ(3,1)=4.0*pi*((1/3)*(2*1+3.0)*rho0-p0)+0.5*qh
mJ(4,1)=q3*qh+0.5*(rho0+p0)*exp(-0.5*ni0)*omega**2
mJ(1,2)=0.25*(p2+rho2)+q0*qk+q1*qk1
mJ(2,2)=(4.0*pi/3.0)*(rho0+3.0*p0)+0.5*qk
mJ(3,2)=q2*qk1+0.5*qk
mJ(4,2)=q3*qk+q4*qk1+0.5*(rho0+p0)*exp(-0.5*ni0)*omega**2
mJ(1,3)=q1*qw1-(p4-(4.0*pi/3.0)*rho0*p2+(0.5/1)*(rho2+p2-(rho0+p0)*ni2))*exp(-ni0)*omega**2
mJ(2,3)=-4.0*pi*(rho2+p2+(8.0*pi/3.0)*rho0*(rho0+p0))
mJ(3,3)=(8.0*pi/1)*(rho2+p2)+q2*qw1
mJ(4,3)=q4*qw1+(rho0+p0)*exp(0.5*ni0)*(0.5*(1+1.0)*ni4-2.0*pi*(rho2+p2)-(16.0*pi**2/3.0)*rho0*(rho0+p0)+ &
& 0.5*(ni4-(4.0*pi/3.0)*rho0*ni2)+0.5*omega**2*exp(-ni0)*(ni2-(8.0*pi/3.0)*rho0))
mJ(1,4)=0.25*ni2*exp(-0.5*ni0)+q0*qx+q1*qx1
mJ(2,4)=0.5*qx
mJ(3,4)=q2*qx1+0.5*qx
mJ(4,4)=0.5*(rho2+p2+0.5*(rho0+p0)*ni2)*1/(rho0+p0)

call inversa(mJ,mG)

do i=1, 4
    write(24,*) mG(i,1),mG(i,2),mG(i,3),mG(i,4)
    write(26,*) mF(i,1),mF(i,2),mF(i,3),mF(i,4)
    write(27,*) mJ(i,1),mJ(i,2),mJ(i,3),mJ(i,4)
end do

do i=1, 4
    do j=1, 4
        mH(i,j)=mG(i,1)*mJ(1,j)+mG(i,2)*mJ(2,j)+mG(i,3)*mJ(3,j)+mG(i,4)*mJ(4,j)
    end do
end do

do i=1, 4
    write(25,*) mH(i,1),mH(i,2),mH(i,3),mH(i,4)
end do

end subroutine matriz

```

## Rotina inversa

```

subroutine inversa(mF,mG)
implicit none
integer, parameter :: dp= SELECTED_REAL_KIND(15,300)
real(kind= dp), dimension (4,4) :: mG, mF
real(kind= dp) :: detF, detF1, detF2, detF3, detF4

```

```

real(kind= dp) :: c11, c12, c13, c14, c21, c22, c23, c24, c31, c32, c33, c34, c41, c42, c43, c44

detF1=mF(2,2)*(mF(3,3)*mF(4,4)-mF(3,4)*mF(4,3))-mF(2,3)*(mF(3,2)*mF(4,4)-mF(3,4)*mF(4,2))+ &
& mF(2,4)*(mF(3,2)*mF(4,3)-mF(3,3)*mF(4,2))

detF2=mF(2,1)*(mF(3,3)*mF(4,4)-mF(3,4)*mF(4,3))-mF(2,3)*(mF(3,1)*mF(4,4)-mF(3,4)*mF(4,1))+ &
& mF(2,4)*(mF(3,1)*mF(4,3)-mF(3,3)*mF(4,1))

detF3=mF(2,1)*(mF(3,2)*mF(4,4)-mF(3,4)*mF(4,2))-mF(2,2)*(mF(3,1)*mF(4,4)-mF(3,4)*mF(4,1))+ &
& mF(2,4)*(mF(3,1)*mF(4,2)-mF(3,2)*mF(4,1))

detF4=mF(2,1)*(mF(3,2)*mF(4,3)-mF(3,3)*mF(4,2))-mF(2,2)*(mF(3,1)*mF(4,3)-mF(3,3)*mF(4,1))+ &
& mF(2,3)*(mF(3,1)*mF(4,2)-mF(3,2)*mF(4,1))

detF=mF(1,1)*detF1-mF(1,2)*detF2+mF(1,3)*detF3-mF(1,4)*detF4

print *, "det F = ", detF

c11=mF(2,2)*(mF(3,3)*mF(4,4)-mF(3,4)*mF(4,3))-mF(2,3)*(mF(3,2)*mF(4,4)-mF(3,4)*mF(4,2))+ &
& mF(2,4)*(mF(3,2)*mF(4,3)-mF(3,3)*mF(4,2))

c12=-mF(2,1)*(mF(3,3)*mF(4,4)-mF(3,4)*mF(4,3))+mF(2,3)*(mF(3,1)*mF(4,4)-mF(3,4)*mF(4,1))- &
& mF(2,4)*(mF(3,1)*mF(4,3)-mF(3,3)*mF(4,1))

c13=mF(2,1)*(mF(3,2)*mF(4,4)-mF(3,4)*mF(4,2))-mF(2,2)*(mF(3,1)*mF(4,4)-mF(3,4)*mF(4,1))+ &
& mF(2,4)*(mF(3,1)*mF(4,2)-mF(3,2)*mF(4,1))

c14=-mF(2,1)*(mF(3,2)*mF(4,3)-mF(3,3)*mF(4,2))+mF(2,2)*(mF(3,1)*mF(4,3)-mF(3,3)*mF(4,1))- &
& mF(2,3)*(mF(3,1)*mF(4,2)-mF(3,2)*mF(4,1))

c21=-mF(1,2)*(mF(3,3)*mF(4,4)-mF(3,4)*mF(4,3))+mF(1,3)*(mF(3,2)*mF(4,4)-mF(3,4)*mF(4,2))- &
& mF(1,4)*(mF(3,2)*mF(4,3)-mF(3,3)*mF(4,2))

c22=mF(1,1)*(mF(3,3)*mF(4,4)-mF(3,4)*mF(4,3))-mF(1,3)*(mF(3,1)*mF(4,4)-mF(3,4)*mF(4,1))+ &
& mF(1,4)*(mF(3,1)*mF(4,3)-mF(3,3)*mF(4,1))

c23=-mF(1,1)*(mF(3,2)*mF(4,4)-mF(3,4)*mF(4,2))+mF(1,2)*(mF(3,1)*mF(4,4)-mF(3,4)*mF(4,1))- &
& mF(1,4)*(mF(3,1)*mF(4,2)-mF(3,2)*mF(4,1))

c24=mF(1,1)*(mF(3,2)*mF(4,3)-mF(3,3)*mF(4,2))-mF(1,2)*(mF(3,1)*mF(4,3)-mF(3,3)*mF(4,1))+ &
& mF(1,3)*(mF(3,1)*mF(4,2)-mF(3,2)*mF(4,1))

c31=mF(1,2)*(mF(2,3)*mF(4,4)-mF(2,4)*mF(4,3))-mF(1,3)*(mF(2,2)*mF(4,4)-mF(2,4)*mF(4,2))+ &
& mF(1,4)*(mF(2,2)*mF(4,3)-mF(2,3)*mF(4,2))

c32=-mF(1,1)*(mF(2,3)*mF(4,4)-mF(2,4)*mF(4,3))+mF(1,3)*(mF(2,1)*mF(4,4)-mF(2,4)*mF(4,1))- &
& mF(1,4)*(mF(2,1)*mF(4,3)-mF(2,3)*mF(4,1))

c33=mF(1,1)*(mF(2,2)*mF(4,4)-mF(2,4)*mF(4,2))-mF(1,2)*(mF(2,1)*mF(4,4)-mF(2,4)*mF(4,1))+ &
& mF(1,4)*(mF(2,1)*mF(4,2)-mF(2,2)*mF(4,1))

c34=-mF(1,1)*(mF(2,2)*mF(4,3)-mF(2,3)*mF(4,2))+mF(1,2)*(mF(2,1)*mF(4,3)-mF(2,3)*mF(4,1))- &
& mF(1,3)*(mF(2,1)*mF(4,2)-mF(2,2)*mF(4,1))

```

```
c41=-mF(1,2)*(mF(2,3)*mF(3,4)-mF(2,4)*mF(3,3))+mF(1,3)*(mF(2,2)*mF(3,4)-mF(2,4)*mF(3,2))- &  
& mF(1,4)*(mF(2,2)*mF(3,3)-mF(2,3)*mF(3,2))
```

```
c42=mF(1,1)*(mF(2,3)*mF(3,4)-mF(2,4)*mF(3,3))-mF(1,3)*(mF(2,1)*mF(3,4)-mF(2,4)*mF(3,1))+ &  
& mF(1,4)*(mF(2,1)*mF(3,3)-mF(2,3)*mF(3,1))
```

```
c43=-mF(1,1)*(mF(2,2)*mF(3,4)-mF(2,4)*mF(3,2))+mF(1,2)*(mF(2,1)*mF(3,4)-mF(2,4)*mF(3,1))- &  
& mF(1,4)*(mF(2,1)*mF(3,2)-mF(2,2)*mF(3,1))
```

```
c44=mF(1,1)*(mF(2,2)*mF(3,3)-mF(2,3)*mF(3,2))-mF(1,2)*(mF(2,1)*mF(3,3)-mF(2,3)*mF(3,1))+ &  
& mF(1,3)*(mF(2,1)*mF(3,2)-mF(2,2)*mF(3,1))
```

```
mG(1,1)=c11/detF
```

```
mG(1,2)=c21/detF
```

```
mG(1,3)=c31/detF
```

```
mG(1,4)=c41/detF
```

```
mG(2,1)=c12/detF
```

```
mG(2,2)=c22/detF
```

```
mG(2,3)=c32/detF
```

```
mG(2,4)=c42/detF
```

```
mG(3,1)=c13/detF
```

```
mG(3,2)=c23/detF
```

```
mG(3,3)=c33/detF
```

```
mG(3,4)=c43/detF
```

```
mG(4,1)=c14/detF
```

```
mG(4,2)=c24/detF
```

```
mG(4,3)=c34/detF
```

```
mG(4,4)=c44/detF
```

```
end subroutine inversa
```

## ANEXO E - PERTURBAÇÕES EM TODO INTERIOR ESTELAR

### Rotina constantes\_rot

```
subroutine constantes_rot(nn,K,l,rhob,RR,MR,omega,HOR,KR,H1R,niR,K1R,H11R,nilR,nsteps,nsteps2)
implicit none
integer, parameter :: n1=3
integer, parameter :: dp= SELECTED_REAL_KIND(15,300)
integer :: i, j, q, nsteps, nsteps2
real(kind= dp), dimension (4,4) :: mG, mF
real(kind= dp) :: l, rhob, r, hh, pi, K, nn, a1, a2, a3, a4, rmin
real(kind= dp) :: h1v1(0:nsteps),kv1(0:nsteps),wv1(0:nsteps),xv1(0:nsteps)
real(kind= dp) :: h1v2(0:nsteps),kv2(0:nsteps),wv2(0:nsteps),xv2(0:nsteps)
real(kind= dp) :: h1v3(0:nsteps),kv3(0:nsteps),wv3(0:nsteps),xv3(0:nsteps)
real(kind= dp) :: h1v4(0:nsteps),kv4(0:nsteps),wv4(0:nsteps),xv4(0:nsteps)
real(kind= dp) :: h1v5(0:nsteps),kv5(0:nsteps),wv5(0:nsteps),xv5(0:nsteps)
real(kind= dp) :: h1v(0:nsteps2),kv(0:nsteps2),wv(0:nsteps2),xv(0:nsteps2), rv(0:nsteps2),h0v(0:nsteps2)
real(kind= dp) :: mv(0:nsteps2),pv(0:nsteps2),niv(0:nsteps2),lbv(0:nsteps2)
real(kind= dp) :: m, p, ni, lb
real(kind= dp) :: KR,H1R,XR,omega,HOR,f4,f5,f6,f7,f8
real(kind= dp) :: MR,pR,niR,lbR,RR,K1R,H11R,nilR,cons(0:4)
real(kind= dp) :: mF1(0:n1),mF2(0:n1),mF3(0:n1),mF4(0:n1),mF5(0:n1)
open(28,file="TestY_t1.dat")

pi=4.0d0*atan(1.0d0)

call s_m_rot(nn,K,rhob,l,h1v1,kv1,wv1,xv1,h1v2,kv2,wv2,xv2,h1v3,kv3,wv3,xv3,mF1,mF2,mF3,cons,omega,nsteps,nsteps2)
call c_m_rot(nn,K,rhob,l,h1v4,kv4,wv4,xv4,h1v5,kv5,wv5,xv5,hh,rmin,mF4,mF5,omega,nsteps,nsteps2,mv,pv,niv,lbv)

mF(1,1)=mF1(0)
mF(2,1)=mF1(1)
mF(3,1)=mF1(2)
mF(4,1)=mF1(3)
mF(1,2)=mF2(0)
mF(2,2)=mF2(1)
mF(3,2)=mF2(2)
mF(4,2)=mF2(3)
mF(1,3)=mF3(0)
mF(2,3)=mF3(1)
mF(3,3)=mF3(2)
mF(4,3)=mF3(3)
mF(1,4)=-mF4(0)
mF(2,4)=-mF4(1)
mF(3,4)=-mF4(2)
mF(4,4)=-mF4(3)

call inversa(mF,mG)
```

```

a1=mG(1,1)*mF5(0)+mG(1,2)*mF5(1)+mG(1,3)*mF5(2)+mG(1,4)*mF5(3)
a2=mG(2,1)*mF5(0)+mG(2,2)*mF5(1)+mG(2,3)*mF5(2)+mG(2,4)*mF5(3)
a3=mG(3,1)*mF5(0)+mG(3,2)*mF5(1)+mG(3,3)*mF5(2)+mG(3,4)*mF5(3)
a4=mG(4,1)*mF5(0)+mG(4,2)*mF5(1)+mG(4,3)*mF5(2)+mG(4,4)*mF5(3)

print*, "a1=", a1, "a2=", a2, "a3=", a3, "a4=", a4

j=nsteps-1

do i=0, j
  h1v(i)=a4*h1v4(i)+h1v5(i)
  kv(i)=a4*kv4(i)+kv5(i)
  wv(i)=a4*wv4(i)+wv5(i)
  xv(i)=a4*xv4(i)+xv5(i)

  m = mv(i)
  p = pv(i)
  ni = niv(i)
  lb = lbv(i)
  r = rv(i)

  f4= 3.0d0*m+0.5d0*(1+2.0d0)*(1-1.0d0)*r+4.0d0*pi*(r**3)*p
  f5= 8.0d0*pi*(RR**3)*(exp(-0.5d0*ni))
  f6= 0.5d0*1*(1+1.0d0)*(m+4.0d0*pi*(r**3)*p)-(omega**2)*(r**3)*(exp(-lb-ni))
  f7= (m+4.0d0*pi*(r**3)*p)*(3.0d0*m-r+4.0d0*pi*(r**3)*p)
  f8= 0.5d0*(1+2.0d0)*(1-1.0d0)*r-(omega**2)*(r**3)*(exp(-ni))-exp(lb)*f7/r
  h0v(i)=(f5*xv(i)-f6*h1v(i)+f8*kv(i))/f4
end do

q=nsteps+1

do i=nsteps, nsteps2-1
  q=q-1
  h1v(i)=a1*h1v1(q)+a2*h1v2(q)+a3*h1v3(q)
  kv(i)=a1*kv1(q)+a2*kv2(q)+a3*kv3(q)
  wv(i)=a1*wv1(q)+a2*wv2(q)+a3*wv3(q)
  xv(i)=a1*xv1(q)+a2*xv2(q)+a3*xv3(q)

  H1R=h1v(i)
  KR=kv(i)
  XR=xv(i)

  m = mv(i)
  p = pv(i)
  ni = niv(i)
  lb = lbv(i)
  r = rv(i)

  f4= 3.0d0*m+0.5d0*(1+2.0d0)*(1-1.0d0)*r+4.0d0*pi*(r**3)*p
  f5= 8.0d0*pi*(RR**3)*(exp(-0.5d0*ni))
  f6= 0.5d0*1*(1+1.0d0)*(m+4.0d0*pi*(r**3)*p)-(omega**2)*(r**3)*(exp(-lb-ni))
  f7= (m+4.0d0*pi*(r**3)*p)*(3.0d0*m-r+4.0d0*pi*(r**3)*p)
  f8= 0.5d0*(1+2.0d0)*(1-1.0d0)*r-(omega**2)*(r**3)*(exp(-ni))-exp(lb)*f7/r
  h0v(i)=(f5*XR-f6*H1R+f8*KR)/f4

```



```

end do

MR=cons(0)
pR=cons(1)
niR=cons(2)
lbR=cons(3)
RR=cons(4)

f4= 3.0d0*MR+0.5d0*(1+2.0d0)*(1-1.0d0)*RR+4.0d0*pi*(RR**3)*pR
f5= 8.0d0*pi*(RR**3)*(exp(-0.5d0*niR))
f6= 0.5d0*1*(1+1.0d0)*(MR+4.0d0*pi*(RR**3)*pR)-(omega**2)*(RR**3)*(exp(-lbR-niR))
f7= (MR+4.0d0*pi*(RR**3)*pR)*(3.0d0*MR-RR+4.0d0*pi*(RR**3)*pR)
f8= 0.5d0*(1+2.0d0)*(1-1.0d0)*RR-(omega**2)*(RR**3)*(exp(-niR))-exp(lbR)*f7/RR
HOR=(f5*XR-f6*H1R+f8*KR)/f4

do i=0, nsteps2-1
  rv(i)=rmin+i*hh
  write(28,fmt='(6(f20.14,1x))') h1v(i), kv(i), wv(i), xv(i), rv(i), h0v(i)
end do

close(28)

end subroutine constantes_rot

```



## ANEXO F - CÁLCULO DA FREQUÊNCIA DO MODO F

### Rotina Zerilli4

```
program Zerilli4
implicit none
integer, parameter :: dp= SELECTED_REAL_KIND(15,300), max_it = 20
real(kind= dp) :: l, nn, K, rhob, pi, omega, omegaA, omegaB, omegaC
real(kind= dp) :: fHz, Ts, omegaA1, omegaB1, omegaC1, test_omega
complex(kind= dp) :: gamaA, betaA, gamaB, betaB, gamaC, betaC, delta
complex(kind= dp) :: gama1, gama2, gama3, omega_up, omega_dw, omega_raiz
complex(kind= dp) :: omega_old
integer :: i
real(kind= dp), dimension (3,3) :: mG, mH

open(51,file="Teste_omega_raiz.dat")
open(52,file="Teste_gama.dat")
open(53,file="Teste_gama123.dat")
open(54,file="Teste_omegaABC.dat")

nn= 1.0
l=2.0
K=100
rhob=1.28e-3
pi= 4.0*atan(1.0)

call tov_rot_z(nn,K,rhob,l,omega)

! Definição dos Omegas
omegaA=1.0*omega
omegaB=0.9*omegaA
omegaC=1.1*omegaA

! Calculo de gamaA, gamaB, gamaC

call zerilli2_rot(nn,K,rhob,l,omegaA,gamaA,betaA)
call zerilli2_rot(nn,K,rhob,l,omegaB,gamaB,betaB)
call zerilli2_rot(nn,K,rhob,l,omegaC,gamaC,betaC)

! Inverte a matriz

mG(1,1)= 1.0
mG(1,2)= omegaA
mG(1,3)= omegaA**2

mG(2,1)= 1.0
```

```

mG(2,2)= omegaB
mG(2,3)= omegaB**2

mG(3,1)= 1.0
mG(3,2)= omegaC
mG(3,3)= omegaC**2

call inversa3x3(mG,mH)

! Calculo de gama1, gama2 e gama3

gama1 = mH(1,1)*gamaA + mH(1,2)*gamaB + mH(1,3)*gamaC
gama2 = mH(2,1)*gamaA + mH(2,2)*gamaB + mH(2,3)*gamaC
gama3 = mH(3,1)*gamaA + mH(3,2)*gamaB + mH(3,3)*gamaC

! Primeiro calculo de omega_raiz

delta=gama2**2-4.0*gama3*gama1

omega_up = (-gama2+sqrt(delta))/(2.0*gama3)
omega_dw = (-gama2-sqrt(delta))/(2.0*gama3)

! Escolhe a raiz

if(aimag(omega_up)*aimag(omega_dw).gt.0.0)then
  if(abs(aimag(omega_up)).gt.abs(aimag(omega_dw)))then
    omega_raiz = omega_dw
  elseif(abs(aimag(omega_up)).lt.abs(aimag(omega_dw)))then
    omega_raiz = omega_up
  endif
elseif(aimag(omega_up)*aimag(omega_dw).lt.0.0)then
  if(aimag(omega_up).gt.0.0)then
    omega_raiz = omega_up
  elseif(aimag(omega_up).lt.0.0)then
    omega_raiz = omega_dw
  endif
endif

write(51,'(6(f15.10,1x))')real(omega_raiz), aimag(omega_raiz), real(omega_up), aimag(omega_up),
real(omega_dw), aimag(omega_dw)

do i=1,max_it

  ! Seleção dos omegas

  omegaA1=abs(real(omega_raiz)-omegaA)
  omegaB1=abs(real(omega_raiz)-omegaB)
  omegaC1=abs(real(omega_raiz)-omegaC)

  if (omegaA1<omegaB1.and.omegaA1<omegaC1) then
    if (omegaB1<omegaC1) then
      omegaC = real(omega_raiz)
    else

```

```

        omegaB = omegaC
        gamaB = gamaC
        omegaC = real(omega_raiz)
    endif
else if (omegaB1<omegaA1.and.omegaB1<omegaC1) then
    if (omegaA1<omegaC1) then
        omegaC = real(omega_raiz)
    else
        omegaA = omegaC
        gamaA = gamaC
        omegaC = real(omega_raiz)
    end if
else if (omegaC1<omegaA1.and.omegaC1<omegaB1) then
    if (omegaA1<omegaB1) then
        omegaB = omegaC
        gamaB = gamaC
        omegaC = real(omega_raiz)
    else
        omegaA = omegaB
        gamaA = gamaB
        omegaB = omegaC
        gamaB = gamaC
        omegaC = real(omega_raiz)
    end if
end if

write(54,'(3(f15.10,1x))')omegaA, omegaB, omegaC

! Recalcula gamaC

call zerilli2_rot(nn,K,rhob,l,omegaC,gamaC,betaC)

write(52,'(5(f15.10,1x))')real(gamaC),aimag(gamaC),real(betaC),aimag(betaC),omegaC

! Inverte a Matriz

mG(1,1)= 1.0
mG(1,2)= omegaA
mG(1,3)= omegaA**2

mG(2,1)= 1.0
mG(2,2)= omegaB
mG(2,3)= omegaB**2

mG(3,1)= 1.0
mG(3,2)= omegaC
mG(3,3)= omegaC**2

call inversa3x3(mG,mH)

! Calculo de gama1, gama2 e gama3

gama1 = mH(1,1)*gamaA + mH(1,2)*gamaB + mH(1,3)*gamaC
gama2 = mH(2,1)*gamaA + mH(2,2)*gamaB + mH(2,3)*gamaC

```

```

gama3 = mH(3,1)*gamaA + mH(3,2)*gamaB + mH(3,3)*gamaC

write(53,'(6(f15.6,1x))')real(gama1),aimag(gama1),real(gama2),aimag(gama2),real(gama3),aimag(gama3)

! Calculo de omega_raiz

delta=gama2**2-4.0*gama3*gama1

omega_up = (-gama2+sqrt(delta))/(2.0*gama3)
omega_dw = (-gama2-sqrt(delta))/(2.0*gama3)

omega_old = omega_raiz

! Escolhe a nova raiz

if(aimag(omega_up)*aimag(omega_dw).gt.0.0)then
  if(abs(aimag(omega_up)).gt.abs(aimag(omega_dw)))then
    omega_raiz = omega_dw
  elseif(abs(aimag(omega_up)).lt.abs(aimag(omega_dw)))then
    omega_raiz = omega_up
  endif
elseif(aimag(omega_up)*aimag(omega_dw).lt.0.0)then
  if(aimag(omega_up).gt.0.0)then
    omega_raiz = omega_up
  elseif(aimag(omega_up).lt.0.0)then
    omega_raiz = omega_dw
  endif
endif

! Teste de omega

test_omega = abs(omega_old-omega_raiz)/abs(omega_old+omega_raiz)

write(51,'(7(f15.10,1x))')real(omega_raiz),aimag(omega_raiz),real(omega_up),aimag(omega_up),real(omega_dw),
aimag(omega_dw),test_omega

if (test_omega.lt.1.0e-8) then
  goto 50
end if

enddo

50 continue

call zerilli2_rot(nn,K,rhob,l,real(omega_raiz),gamaC,betaC)

fHz = 32.3105*real(omega_raiz)
Ts = 1/(32310.5*aimag(omega_raiz))

print*, ""
print*, ""
print*, ""
print*, "f(kHz)=" , fHz
print*, ""

```

```

        print*, "T(s)=" , Ts
        print*, ""

close(51)
close(52)
close(53)
close(54)

end program Zerilli4

```

## Rotina zerilli2\_rot

```

subroutine zerilli2_rot(nn,K,rhob,l,omega,gama,beta,ns,ns2)
implicit none
integer, parameter :: dp= SELECTED_REAL_KIND(15,300)
real(kind= dp) :: l, nR, nn, K, rhob, MR, omega, rmax, rtmax, Zmax, Zlmax
real(kind= dp) :: Z_real_out, Zl_real_out, Z_comp_out, Zl_comp_out
real(kind= dp) :: Z_real_in, Zl_real_in, Z_comp_in, Zl_comp_in
real(kind= dp) :: z1, z2, z3, z4, pi, drdrt, r, rt
integer :: i, j, ns, ns2
complex(kind= dp) :: Z_out, Zl_out, Z_in, Zl_in, gama, beta, detE, ii, alfa1, alfa2,Z_out2, Zl_out2, Z_in2, Zl_in2
complex(kind= dp), dimension (2,2) :: mE, mF

open(32,file="Zerilli2.dat")

pi= 4.0d0*atan(1.0d0)

call zerilli_rot(nn,K,rhob,l,MR,omega,rmax,rtmax,Zmax,Zlmax,ns,ns2)

r=rmax
rt=rtmax
nR=0.5d0*(1-1.0d0)*(1+2.0d0)
drdrt=1.0d0 - 2.0d0*MR/r

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!11
ii = cmplx(0.0d0,1.0d0)

alfa1 = -ii*(nR + 1.0d0)/omega
alfa2 = -(nR*(nR+1.0d0)-1.5d0*ii*MR*omega*(1.0d0+2.0d0/nR))/(2.0d0*omega**2)

Z_out2 = exp(-ii*omega*rt)*(1.0d0 + alfa1/r + alfa2/r**2)
Zl_out2 = -ii*omega*Z_out2 &
& - exp(-ii*omega*rt)*(alfa1/r**2 + 2.0*alfa2/r**3)*drdrt
Z_in2 = conjg(Z_out2)
Zl_in2 = conjg(Zl_out2)

! Inicio do calculo de beta e gama

mE(1,1)=Z_out2

```

```

mE(2,1)=Zl_out2
mE(1,2)=Z_in2
mE(2,2)=Zl_in2

write(32,'(8(f12.9,1x))')real(mE(1,1)),aimag(mE(1,1)),real(mE(2,1)),aimag(mE(2,1)),real(mE(1,2)),
aimag(mE(1,2)),real(mE(2,2)),aimag(mE(2,2))

detE= mE(1,1)*mE(2,2)-mE(1,2)*mE(2,1)

write(32,'(2(f12.9,1x))')real(detE), aimag(detE)

mF(1,1)=mE(2,2)/detE
mF(1,2)=-mE(1,2)/detE
mF(2,1)=-mE(2,1)/detE
mF(2,2)=mE(1,1)/detE

write(32,'(8(f12.8,1x))')real(mF(1,1)),aimag(mF(1,1)),real(mF(2,1)),aimag(mF(2,1)),real(mF(1,2)),
aimag(mF(1,2)),real(mF(2,2)),aimag(mF(2,2))

! Calculo de beta e gama

write(32,'(2(f12.9,1x))')Zmax, Zlmax

beta = mF(1,1)*Zmax + mF(1,2)*Zlmax
gama = mF(2,1)*Zmax + mF(2,2)*Zlmax

print*, ""
print*, ""
print*, ""
print*, "beta=" , beta
print*, ""
print*, "gama=" , gama
print*, ""

write(32,'(4(f12.9,1x))')real(gama),aimag(gama),real(beta),aimag(beta)

close(32)

end subroutine zerilli2_rot

```

## Rotina zerilli\_rot

```

subroutine zerilli_rot(nn,K,rhob,l,MR,omega,rmax1,rtmax1,Zmax,Zlmax,ns,ns2)
implicit none
integer, parameter :: dp= SELECTED_REAL_KIND(15,300)
real(kind= dp) :: aR, bR, HOR, KR, gR, hR, kRp, ZRt, ZRt1, r, rt, nR, pi, v2m, v3m, rm, rtm, v5, v6,
Zrtt, Zrtt1, H1R, niR, K1R, H11R, nilR
real(kind= dp) :: MR, RR, omega, detA, detC, rmin, rmax, rtmin, rtmax, v1, v2, v3, v4
real(kind= dp) :: l, nn, K, rhob, x1, x2, xacc, h, Zmax, Zlmax, rmax1, rtmax1

```



```

integer :: i, ns, ns2
integer, parameter :: n1=1
real(kind= dp), dimension (2,2) :: mA, mB, mC, mD, mC_inv, mE
real(kind= dp) :: x(0:n1)
real(kind= dp) :: Ztv(0:ns), Ztv1(0:ns), Vtv(0:ns), rtv(0:ns), rv(0:ns), Z11(0:ns), H0v(0:ns), Kv(0:ns)
real(kind= dp), external :: rtsafe

open(31,file="Zerilli_v1.dat")
open(32,file="H0_K.dat")

pi= 4.0d0*atan(1.0d0)

call constantes_rot(nn,K,l,rhob,RR,MR,omega,HOR,KR,H1R,niR,K1R,H11R,nilR,ns,ns2)

nR=0.5d0*(1-1.0d0)*(1+2.0d0)
aR=- (nR*RR+3.0d0*MR)/(omega**2*RR**2-(nR+1.0d0)*MR/RR)
bR=(nR*RR*(RR-2.0d0*MR)-omega**2*RR**4+MR*(RR-3.0d0*MR))/((RR-2.0d0*MR)*
(omega**2*RR**2-(nR+1.0d0)*MR/RR))
gR=(nR*(nR+1.0d0)*RR**2+3.0d0*nR*MR*RR+6.0d0*MR**2)/((nR*RR+3.0d0*MR)*RR**2)
hR=(-nR*RR**2+3.0d0*nR*MR*RR+3.0d0*MR**2)/((RR-2.0d0*MR)*(nR*RR+3.0d0*MR))
kRp=-RR**2/(RR-2.0d0*MR)
rt=RR+2.0d0*MR*log(RR/(2.0d0*MR)-1.0d0)

mA(1,1)=gR
mA(2,1)=hR
mA(1,2)=1.0d0
mA(2,2)=kRp

!      Inicio do calculo da inversa

detA= mA(1,1)*mA(2,2)-mA(1,2)*mA(2,1)

mB(1,1)=mA(2,2)/detA
mB(1,2)=-mA(1,2)/detA
mB(2,1)=-mA(2,1)/detA
mB(2,2)=mA(1,1)/detA

!      Fim do calculo da inversa

mC(1,1)=0.0d0
mC(2,1)=aR
mC(1,2)=1.0d0
mC(2,2)=bR

!      Produto das Matrizes

mD(1,1)=mB(1,1)*mC(1,1)+mB(1,2)*mC(2,1)
mD(2,1)=mB(2,1)*mC(1,1)+mB(2,2)*mC(2,1)
mD(1,2)=mB(1,1)*mC(1,2)+mB(1,2)*mC(2,2)
mD(2,2)=mB(2,1)*mC(1,2)+mB(2,2)*mC(2,2)

!      Calculo das condições iniciais para resolver a equação de Zerilli

ZRt=mD(1,1)*HOR+mD(1,2)*KR

```

```

ZRtl=mD(2,1)*HOR+mD(2,2)*KR

! Intervalo de integracao para a eq. de Zerilli

rmin=RR
rmax=25.0d0/omega

rtmin=rmin+2.0d0*MR*log(rmin/(2.0d0*MR)-1.0d0)
rtmax=rmax+2.0d0*MR*log(rmax/(2.0d0*MR)-1.0d0)

h=(rtmax-rtmin)/ns

print*, ""
print*, ""
print*, ""
print*, "h=" , h
print*, ""
print*, "ZRt=" , ZRt
print*, ""
print*, "ZRtl=" , ZRtl
print*, ""

Zrtt = RR**(1+2.0d0)*(KR - exp(niR)*H1R)/(nR*RR+3.0d0*MR)

Zrttl = (1+2.0d0)*RR**(1+1.0d0)*(KR- exp(niR)*H1R)/(nR*RR+3.0d0*MR) &
& - RR**(1+2.0d0)*nR*(KR- exp(niR)*H1R)/(nR*RR+3.0d0*MR)**2 &
& + RR**(1+2.0d0)*(K1R - exp(niR)*H11R - ni1R*exp(niR)*H1R)/(nR*RR+3.0d0*MR)
print*, "ZRtt=" , ZRtt
print*, ""
print*, "Zrttl=" , Zrttl
print*, ""

! Condições Iniciais do RK

x(0)=ZRt
x(1)=ZRtl

! Componentes zero dos vetores

Ztv(0)=ZRt
Ztv1(0)=ZRtl
rv(0)=rmin
rtv(0)=rtmin

v1=2.0d0*nR**2*(nR+1.0d0)*RR**3+6.0d0*nR**2*MR*RR**2+18.0d0*nR*MR**2*RR+18.0d0*MR**3
Vtv(0)=v1*(1.0d0-2.0d0*MR/RR)/(RR**3*(nR*RR+3.0d0*MR)**2)
v4=Vtv(0)-omega**2

Z11(0) = (v4-omega**2)*ZRt

! Valores necessários para o calculo de r, através de r*, utilizados em rtsafe

xacc=1.0d-9
x1=rmin-xacc

```

```

x2=rmax+xacc

do i=1, ns

  ! Calculo de V em rt+h

  rt=rtmin+i*h
  r=rtsafe(x1,x2,xacc,rt,MR)

  v2=2.0d0*nR**2*(nR+1.0d0)*r**3+6.0d0*nR**2*MR*r**2+18.0d0*nR*MR**2*r+18.0d0*MR**3
  v3=v2*(1.0d0-2.0d0*MR/r)/(r**3*(nR*r+3.0d0*MR)**2)
  Vtv(i)=v3
  v6=v3-omega**2

  ! Calculo de V em rt+0.5*h

  rtm=rtmin+i*h-0.5d0*h
  rm=rtsafe(x1,x2,xacc,rtm,MR)

  v2m=2.0d0*nR**2*(nR+1.0d0)*rm**3+6.0d0*nR**2*MR*rm**2+18.0d0*nR*MR**2*rm+18.0d0*MR**3
  v3m=v2*(1.0d0-2.0d0*MR/rm)/(rm**3*(nR*rm+3.0d0*MR)**2)
  v5=v3m-omega**2

  call rk4sys_fora(n1,h,x,rt,v4,v5,v6)

  Ztv(i)=x(0)
  Ztv1(i)=x(1)
  rv(i)=r
  rtv(i)=rt
  Z11(i) = v6*x(1)

  v4=v6

  Zmax=x(0)
  Z1max=x(1)
  rmax1=r
  rtmax1=rt

end do

do i=0, ns
  write(31,fmt='(6(f15.10,1x))') Ztv(i), Ztv1(i), Vtv(i), rv(i), rtv(i), Z11(i)
end do

! Check H0 and K!

do i=0, ns
  r = rv(i)
  nR=0.5d0*(1-1.0d0)*(1+2.0d0)
  aR=- (nR*r+3.0d0*MR)/(omega**2*r**2-(nR+1.0d0)*MR/r)
  bR=(nR*r*(r-2.0d0*MR)-omega**2*r**4+MR*(r-3.0d0*MR))/((r-2.0d0*MR)*
(omega**2*r**2-(nR+1.0d0)*MR/r))
  gR=(nR*(nR+1.0d0)*r**2+3.0d0*nR*MR*r+6.0d0*MR**2)/((nR*r+3.0d0*MR)*r**2)
  hR=(-nR*r**2+3.0d0*nR*MR*r+3.0d0*MR**2)/((r-2.0d0*MR)*(nR*r+3.0d0*MR))

```

```

kRp=-r**2/(r-2.0d0*MR)

mC(1,1)=0.0d0
mC(2,1)=aR
mC(1,2)=1.0d0
mC(2,2)=bR

detC = mC(1,1)*mC(2,2)-mC(1,2)*mC(2,1)

mC_inv(1,1)=mC(2,2)/detC
mC_inv(1,2)=-mC(1,2)/detC
mC_inv(2,1)=-mC(2,1)/detC
mC_inv(2,2)=mC(1,1)/detC

mA(1,1)=gR
mA(2,1)=hR
mA(1,2)=1.0d0
mA(2,2)=kRp

mE(1,1)=mC_inv(1,1)*mA(1,1)+mC_inv(1,2)*mA(2,1)
mE(2,1)=mC_inv(2,1)*mA(1,1)+mC_inv(2,2)*mA(2,1)
mE(1,2)=mC_inv(1,1)*mA(1,2)+mC_inv(1,2)*mA(2,2)
mE(2,2)=mC_inv(2,1)*mA(1,2)+mC_inv(2,2)*mA(2,2)

HOv(i)=mE(1,1)*Ztv(i)+mE(1,2)*Ztv1(i)
Kv(i) =mE(2,1)*Ztv(i)+mE(2,2)*Ztv1(i)

write(32,fmt='(3(f15.10,1x))') rv(i), HOv(i), Kv(i)
end do

close(31)
close(32)

end subroutine zerilli_rot

```

## Rotina rk4sys\_fora

```

subroutine rk4sys_fora(n,h,x,rt,v4,v5,v6)
implicit none
integer, parameter :: dp= SELECTED_REAL_KIND(15,300)
real(kind= dp) :: x(0:n)
real(kind= dp), allocatable :: y(:), f(:, :)
integer :: i, n
real(kind= dp) :: h,rt,v4,v5,v6
allocate (y(0:n), f(0:n,4))

call xpsys_fora(n,x,f(0,1),rt,v4)

```

```

do i = 0,n
  y(i) = x(i) + 0.5d0*h*f(i,1)
  enddo
  call xpsys_fora(n,y,f(0,2),rt+0.5d0*h,v5)

do i = 0,n
  y(i) = x(i) + 0.5d0*h*f(i,2)
  enddo
  call xpsys_fora(n,y,f(0,3),rt+0.5d0*h,v5)

do i = 0,n
  y(i) = x(i) + h*f(i,3)
  enddo
  call xpsys_fora(n,y,f(0,4),rt+h,v6)

do i = 0,n
x(i) = x(i) + (h/6.0d0)* (f(i,1) + 2.0d0*(f(i,2) + f(i,3)) + f(i,4))
enddo
end subroutine rk4sys_fora

```

## Rotina xpsys\_fora

```

subroutine xpsys_fora(n,x,f,rt,v4)
implicit none
integer, parameter :: dp= SELECTED_REAL_KIND(15,300)
real(kind= dp), dimension (0:n) :: x, f
real(kind= dp) :: rt, v4
integer :: n
! x(0)=Z1, x(1)=Z

f(0) = x(1)
f(1) = v4*x(0)

end subroutine xpsys_fora

```

## Rotina rtsafe

```

FUNCTION rtsafe(x1,x2,xacc,rt,MR)
IMPLICIT NONE
  integer, parameter :: dp= SELECTED_REAL_KIND(15,300)
REAL(dp), INTENT(IN) :: x1,x2,xacc,rt,MR
REAL(dp) :: rtsafe

```

```

INTEGER, PARAMETER :: MAXIT=1000
INTEGER :: j
REAL(dp) :: df,dx,dxold,f,fh,f1,temp,xh,xl
call funcd(x1,rt,MR,f1,df)
call funcd(x2,rt,MR,fh,df)
if ((f1 > 0.0 .and. fh > 0.0) .or. &
    (f1 < 0.0 .and. fh < 0.0)) &
print*, 'root must be bracketed in rtsafe'
if (f1 == 0.0) then
rtsafe=x1
RETURN
else if (fh == 0.0) then
rtsafe=x2
RETURN
else if (f1 < 0.0) then
xl=x1
xh=x2
else
xh=x1
xl=x2
end if
rtsafe=0.5_dp*(x1+x2)
dxold=abs(x2-x1)
dx=dxold
call funcd(rtsafe,rt,MR,f,df)
do j=1,MAXIT
if (((rtsafe-xh)*df-f)*((rtsafe-xl)*df-f) > 0.0 .or. &
    abs(2.0_dp*f) > abs(dxold*df) ) then
dxold=dx
dx=0.5_dp*(xh-xl)
rtsafe=xl+dx
if (xl == rtsafe) RETURN
else
dxold=dx
dx=f/df
temp=rtsafe
rtsafe=rtsafe-dx
if (temp == rtsafe) RETURN
end if
if (abs(dx) < xacc) RETURN
call funcd(rtsafe,rt,MR,f,df)
if (f < 0.0) then
xl=rtsafe
else
xh=rtsafe
end if
end do
print*, 'rtsafe: exceeded maximum iterations'
END FUNCTION rtsafe

```

## Rotina funcd

```
SUBROUTINE funcd(x,rt,MR,fval,fderiv)
IMPLICIT NONE
    integer, parameter :: dp= SELECTED_REAL_KIND(15,300)
REAL(dp), INTENT(IN) :: x, rt, MR
REAL(dp), INTENT(OUT) :: fval,fderiv

    fval=x+2.0*MR*log(x/(2.0*MR)-1.0)-rt
    fderiv=1.0+2.0*MR/(x-2.0*MR)

END SUBROUTINE funcd
```

## Rotina inversa3x3

```
subroutine inversa3x3(mG,mH)
implicit none
integer, parameter :: dp= SELECTED_REAL_KIND(15,300)
real(kind= dp), dimension (3,3) :: mG, mH, mI
real(kind= dp) :: detG, omegaA, omegaB, omegaC
integer i,j

open(70,file="Teste_matriz_inversa.dat")

omegaA = mG(1,2)
omegaB = mG(2,2)
omegaC = mG(3,2)

detG = omegaA*omegaB*(omegaB-omegaA) + omegaB*omegaC*(omegaC-omegaB) + omegaC*omegaA*(omegaA - omegaC)

mH(1,1) = (mG(2,2)*mG(3,3)-mG(2,3)*mG(3,2))/detG
mH(1,2) = -(mG(1,2)*mG(3,3)-mG(1,3)*mG(3,2))/detG
mH(1,3) = (mG(1,2)*mG(2,3)-mG(2,2)*mG(1,3))/detG

mH(2,1) = -(mG(2,1)*mG(3,3)-mG(2,3)*mG(3,1))/detG
mH(2,2) = (mG(1,1)*mG(3,3)-mG(1,3)*mG(3,1))/detG
mH(2,3) = -(mG(1,1)*mG(2,3)-mG(1,3)*mG(2,1))/detG

mH(3,1) = (mG(2,1)*mG(3,2)-mG(2,2)*mG(3,1))/detG
mH(3,2) = -(mG(1,1)*mG(3,2)-mG(1,2)*mG(3,1))/detG
mH(3,3) = (mG(1,1)*mG(2,2)-mG(1,2)*mG(2,1))/detG

! Teste da inversa

do i=1,3
    do j=1,3
        mI(i,j) = mG(i,1)*mH(1,j) + mG(i,2)*mH(2,j) + mG(i,3)*mH(3,j)
        write(70,*)"mI(",i,",",j,") = ",mI(i,j)
    enddo
enddo
```

```
enddo  
  
close(70)  
  
end subroutine inversa3x3
```



## **PUBLICAÇÕES TÉCNICO-CIENTÍFICAS EDITADAS PELO INPE**

### **Teses e Dissertações (TDI)**

Teses e Dissertações apresentadas nos Cursos de Pós-Graduação do INPE.

### **Manuais Técnicos (MAN)**

São publicações de caráter técnico que incluem normas, procedimentos, instruções e orientações.

### **Notas Técnico-Científicas (NTC)**

Incluem resultados preliminares de pesquisa, descrição de equipamentos, descrição e ou documentação de programas de computador, descrição de sistemas e experimentos, apresentação de testes, dados, atlas, e documentação de projetos de engenharia.

### **Relatórios de Pesquisa (RPQ)**

Reportam resultados ou progressos de pesquisas tanto de natureza técnica quanto científica, cujo nível seja compatível com o de uma publicação em periódico nacional ou internacional.

### **Propostas e Relatórios de Projetos (PRP)**

São propostas de projetos técnico-científicos e relatórios de acompanhamento de projetos, atividades e convênios.

### **Publicações Didáticas (PUD)**

Incluem apostilas, notas de aula e manuais didáticos.

### **Publicações Seriadas**

São os seriados técnico-científicos: boletins, periódicos, anuários e anais de eventos (simpósios e congressos). Constam destas publicações o Internacional Standard Serial Number (ISSN), que é um código único e definitivo para identificação de títulos de seriados.

### **Programas de Computador (PDC)**

São a seqüência de instruções ou códigos, expressos em uma linguagem de programação compilada ou interpretada, a ser executada por um computador para alcançar um determinado objetivo. Aceitam-se tanto programas fonte quanto os executáveis.

### **Pré-publicações (PRE)**

Todos os artigos publicados em periódicos, anais e como capítulos de livros.