



Gibran Henrique de Souza

O modo fundamental de emissão de ondas gravitacionais

Campinas-SP
2013



UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE FÍSICA *Gleb Wataghin*

Gibran Henrique de Souza

O modo fundamental de emissão de ondas
gravitacionais

Orientador: Prof. Dr. Anderson Campos Fauth (IFGW-UNICAMP)

Co-Orientador: Profa. Dra. Cecilia Bertoni Martha Hadler Chirenti (CMCC-UFABC)

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação
em Física do Instituto de Física *Gleb Wataghin* da Universidade Estadual
de Campinas para obtenção do título de Mestre em Física.

Este exemplar corresponde à versão final da dissertação
defendido pelo aluno Gibran Henrique de Souza e orientado
pelo Prof. Dr. Anderson Campos Fauth.

Campinas-SP

2013

**FICHA CATALOGRÁFICA ELABORADA POR
VALKÍRIA SUCCI VICENTE – CRB8/5398 - BIBLIOTECA DO IFGW
UNICAMP**

| | |
|-------|--|
| So89m | <p>Souza, Gibran Henrique de, 1989- O modo fundamental de emissão de ondas gravitacionais / Gibran Henrique de Souza. -- Campinas, SP : [s.n.], 2013.</p> <p>Orientadores: Anderson Campos Fauth, Cecilia Bertoni Martha Hadler Chirenti. Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Física “Gleb Wataghin”.</p> <p>1. Estrelas de nêutrons. 2. Ondas gravitacionais. 3. Astrofísica. I. Fauth, Anderson Campos, 1957- II. Chirenti, Cecilia Bertoni Martha Hadler. III. Universidade Estadual de Campinas. Instituto de Física “Gleb Wataghin”. IV. Título.</p> |
|-------|--|

Informações para Biblioteca Digital

Título em inglês: The fundamental mode emission of gravitational waves

Palavras-chave em inglês:

Neutron stars

Gravitational waves

Astrophysics

Área de Concentração: Física

Titulação: Mestre em Física

Banca Examinadora:

Anderson Campos Fauth [Orientador]

Jorge Ernesto Horvath

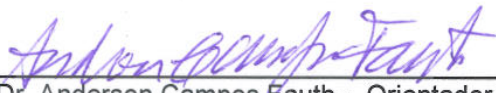
Pedro Cunha de Holanda

Data da Defesa: 30-04-2013

Programa de Pós-Graduação em: Física

MEMBROS DA COMISSÃO JULGADORA DA DISSERTAÇÃO DE MESTRADO DE **GIBRAN HENRIQUE DE SOUZA – RA 070995** APRESENTADA E APROVADA AO INSTITUTO DE FÍSICA “GLEB WATAGHIN”, DA UNIVERSIDADE ESTADUAL DE CAMPINAS, EM 30/04/2013.

COMISSÃO JULGADORA:



Prof. Dr. Anderson Campos Fauth - Orientador do Candidato
DRCC/IFGW/UNICAMP



Prof. Dr. Jorge Ernesto Horvath – IAGCA/USP



Prof. Dr. Pedro Cunha de Holanda – DRCC/IFGW/UNICAMP

Dedico este trabalho
ao meu pai que embora tenha nos deixado nunca será esquecido.

Uma equação para tudo unificar, uma equação para tudo narrar;
uma equação para todos os conceitos conter, e o universo descrever.

Agradecimentos

Primeiramente gostaria de agradecer à Deus pelo dom da vida e inteligência, à minha mãe que nunca desistiu de mim e sempre soube me apoiar e incentivar em meu caminho na ciência, ao meu orientador por ter me aceito como orientando e por me instruir durante a realização deste projeto, à minha co-orientadora que gentilmente me guiou e me mostrou os caminhos corretos, agradeço também ao Instituto de Física Gleb Wataghin, IFGW, que através da CAPES financiou este projeto e possibilitou a sua realização, por fim agradeço a todos que me apoiaram e que me incentivaram.

Resumo

Usando como base um código computacional que integra numericamente as equações TOV, que descrevem o interior de corpos relativísticos de simetria esférica, com a equação de estado SLy, que fornece a pressão em função da densidade para a matéria nuclear em condições extremas se comparadas à matéria nuclear convencional, conseguimos descrever uma estrela de nêutrons realista e com esta simular a emissão de ondas gravitacionais, com a previsão de como seria seu tempo de decaimento e frequência.

Abstract

Using as base a computer code that integrates numerically the TOV equations, which describe the interior of relativistic bodies of spherical symmetry, with the SLy equation of state, which provides the pressure in function of density for nuclear matter under extreme conditions when compared with conventional nuclear matter, we describe a realistic neutron star and simulate the emission of gravitational waves, with the predictions of how its decay rate and frequency will be.

Sumário

| | |
|---------------------------------------|----------|
| Agradecimentos | ix |
| Resumo | xi |
| Abstract | xiii |
| 1 Introdução | 1 |
| 2 Estrelas de Nêutrons | 3 |
| 2.1 Origem | 3 |
| 2.2 Características Físicas | 5 |
| 2.2.1 Pulsares | 6 |
| 2.2.2 Radio-quietos | 8 |
| 2.3 Estrutura | 8 |
| 2.3.1 Crosta Externa | 10 |
| 2.3.2 Crosta Interna | 10 |
| 2.3.3 Núcleo Externo | 11 |
| 2.3.4 Núcleo Interno | 11 |
| 2.4 Equação de Estado | 12 |

| | | |
|----------|---|-----------|
| 2.4.1 | Politropos | 12 |
| 2.4.2 | Equações de estado realistas | 13 |
| 2.4.3 | Equações TOV | 15 |
| 2.5 | Resultados e comparação | 19 |
| 2.5.1 | Massa | 19 |
| 2.5.2 | Pressão | 21 |
| 2.5.3 | Densidade | 23 |
| 2.5.4 | Índice Adiabático | 25 |
| 3 | Ondas Gravitacionais | 27 |
| 3.1 | Perspectivas | 27 |
| 3.2 | Dedução Matemática | 27 |
| 3.2.1 | Ondas no vácuo | 29 |
| 3.2.2 | Ondas nas Estrelas | 32 |
| 3.2.2.1 | Região $R/2 \leq r \leq R$ | 35 |
| 3.2.2.2 | Região $0 \leq r \leq R/2$ | 36 |
| 3.2.2.3 | Perturbação Total | 39 |
| 3.2.3 | As perturbações fora da estrela e as ondas emitidas | 40 |
| 3.2.3.1 | A equação de Zerilli | 40 |
| 3.2.3.2 | A Frequência do Modo Quasinormal Fundamental | 42 |
| 3.3 | Detecção | 44 |
| 3.3.1 | Observação Indireta | 44 |
| 3.3.1.1 | O sistema PSR1913+16 | 45 |
| 3.3.2 | Observação Direta | 46 |
| 3.3.2.1 | Interferometria à laser | 46 |
| 3.3.2.2 | Detectores de Massas Ressonantes | 49 |

| | | |
|----------|---|-----------|
| 4 | Metodologia | 54 |
| 4.1 | Dentro da Estrela | 54 |
| 4.2 | Fora da Estrela | 56 |
| 5 | Resultados das simulações | 59 |
| 5.1 | Diferenças entre as estrelas simuladas | 59 |
| 5.2 | As ondas gravitacionais simuladas | 62 |
| 5.2.1 | As ondas em função da densidade central | 62 |
| 5.2.2 | As ondas em função da massa da estrela | 64 |
| 5.3 | As Linearizações | 67 |
| 5.3.1 | Frequência | 67 |
| 5.3.2 | Tempo de decaimento | 71 |
| 5.3.3 | Comparação em relação à outros autores. | 74 |
| 5.4 | O Problema Inverso | 81 |
| 6 | Conclusões | 83 |
| 7 | Apoio | 86 |
| | Bibliografia | 86 |
| | Apêndice | 91 |

Lista de Figuras

| | | |
|------|--|----|
| 2.1 | <i>A Estrutura de cebola evidenciando as camadas de fusões sucessivas, cortesia de R.J.Hall.</i> | 4 |
| 2.2 | <i>A Supernova SN 1994D (ponto brilhante na esquerda inferior), à 108 milhões de anos-luz. Cortesia da NASA.</i> | 5 |
| 2.3 | <i>A Nebulosa do Caranguejo, imagens em cores falsas. Cortesia da NASA.</i> | 7 |
| 2.4 | <i>O pulsar da Nebulosa do Caranguejo e a nuvem que gira junto com ele. Foto obtido pelo observatório Chandra, cortesia da NASA.</i> | 9 |
| 2.5 | <i>O esquema de um pulsar, mostrando as linhas de campo e o feixe de radiação. Cortesia de www.clubedecienciasastronomia.blogspot.com.br.</i> | 9 |
| 2.6 | <i>Esquematização da estrutura de um pulsar. P.Haensel.</i> | 12 |
| 2.7 | <i>A pressão em função da densidade total de energia.</i> | 15 |
| 2.8 | <i>O índice adiabático em função da densidade total de energia.</i> | 16 |
| 2.9 | <i>A distribuição de massa para as estrelas.</i> | 20 |
| 2.10 | <i>A distribuição de pressão para as estrelas.</i> | 22 |
| 2.11 | <i>A distribuição de densidade para as estrelas.</i> | 24 |
| 2.12 | <i>A distribuição do índice adiabático para a estrela SLy.</i> | 25 |
| 3.1 | <i>A distorção causada por uma onda gravitacional ao passar por um objeto.</i> | 32 |

| | | |
|------|--|----|
| 3.2 | Representação do sistema PSR1913+16. | 46 |
| 3.3 | Mudança de fase no periastro. | 47 |
| 3.4 | Um detector pelo princípio do interferômetro de Michelson. | 48 |
| 3.5 | Esquema do projeto LISA. | 49 |
| 3.6 | Joseph Weber e seu detector. | 50 |
| 3.7 | O detector Mário Schenberg. | 52 |
| 3.8 | A esfera detectora. | 53 |
| 4.1 | A sequência para as rotinas. | 57 |
| 4.2 | A sequência para as rotinas (continuação). | 58 |
| 5.1 | A massa da estrela em função da densidade central. | 60 |
| 5.2 | A massa da estrela em função do seu raio total. | 61 |
| 5.3 | A frequência da onda emitida em função da densidade central. | 63 |
| 5.4 | O tempo de decaimento da onda emitida em função da densidade central. | 64 |
| 5.5 | A frequência da onda emitida em função da massa da estrela. | 65 |
| 5.6 | O tempo de decaimento da onda emitida em função da massa da estrela. | 66 |
| 5.7 | O comportamento de nossas duas equações de estado quando submetidas a relação 5.1. | 68 |
| 5.8 | O ajuste linear para a curva SLy. | 69 |
| 5.9 | O ajuste linear para a curva politrópica. | 71 |
| 5.10 | O comportamento de nossas duas equações de estado quando submetidas a relação 5.4. | 72 |
| 5.11 | O ajuste linear para a curva SLy. | 74 |
| 5.12 | O ajuste linear para a curva politrópica. | 75 |
| 5.13 | Comparando as linearizações obtidas. | 77 |
| 5.14 | Comparando as linearizações obtidas. | 79 |

Lista de Tabelas

| | | |
|-----|---|----|
| 3.1 | Condições iniciais | 35 |
| 5.1 | Parâmetros para a figura 5.13 | 80 |
| 5.2 | Parâmetros para a figura 5.14 | 80 |
| 5.3 | Nossos resultados para a equação SLy comparados com Bernuzzi. | 80 |

Introdução

Embora propostas teoricamente desde o começo da década de 1930, a existência das estrelas de nêutrons foi comprovada observacionalmente na década de 1960 com o desenvolvimento da tecnologia dos rádio-telescópios. Inicialmente os primeiros trabalhos teóricos eram apenas uma forma de aplicação da Teoria da Relatividade Geral a sistemas nucleares em que a densidade média de energia permitiria fenômenos relativísticos, mas essas propostas não foram levadas a sério. Conforme o conhecimento avançou nas áreas de física nuclear, de partículas e estatística, novos fenômenos foram descobertos e outros propostos sobre o que ocorreria com a matéria ordinária quando esta passa aos limites relativísticos de densidade e energia, como o de-confinamento de quarks e a proposta de um plasma de quarks-glúons. Mas as comprovações observacionais surgiram na década de 1960 quando os rádio-telescópios permitiram a varredura do espaço próximo numa faixa muito mais ampla do espectro eletromagnético do que a faixa do visível à qual os telescópios comuns estão limitados. Teoricamente já havia a predição de que estes corpos podiam resultar de núcleos estelares que sobreviveram à morte da estrela e que, se existissem, exibiriam um campo magnético intenso que geraria radiação periódica, e a detecção desta radiação seria a forma pela qual tais corpos seriam localizados. E foi assim que ocorreu, em 1965 a primeira fonte de rádio extra-solar foi detectada na Nebulosa do Caranguejo e assim

com as primeiras evidências observacionais esta área de estudo ganhou destaque crescente e gerou grandes contribuições ao nosso conhecimento astrofísico. Atualmente o estudo destes corpos revelou novos estados de condensação da matéria, revelando como ela se comporta no limite em que é comprimida ao ponto em que os núcleos atômicos ficam separados por uns poucos raios nucleares ; além de evidências experimentais para a Teoria da Relatividade Geral, como o decaimento da órbita de pulsares binários, e a busca por ondas gravitacionais, pois embora haja observações indiretas ainda não há uma observação direta que as confirme de uma vez por todas; além disso observacionalmente usam-se os pulsares como marcadores astronômicos de distância entre as galáxias.

Estrelas de Nêutrons

2.1 Origem

Durante o processo de formação estelar algumas estrelas chegam a adquirir massas que variam desde poucas vezes a massa solar até dezenas de vezes, essas estrelas são chamadas de estrelas gigantes. Justamente devido à sua grande massa essas estrelas apresentam uma vida na sequência principal de poucas centenas de milhões de anos, o que é uma vida curta se comparada à vida de bilhões de anos que as estrelas pouco massivas, como o Sol, apresentam. Essa curta vida acontece pois a estrela apresenta uma alta taxa de produção de energia por meio da fusão nuclear para balancear a pressão exercida pela força da gravidade, sem o equilíbrio gerado pela fusão a estrela acabaria por desabar sobre si mesma. Primeiramente a cadeia de fusão começa com a conversão de hidrogênio em hélio no núcleo da estrela. Quando o hidrogênio acaba o núcleo sofre uma contração em que sua temperatura e densidade aumentam, possibilitando a conversão de hélio em carbono, a camada logo acima do núcleo aquece-se e passa a converter hidrogênio em hélio, com isso ela se expande e a estrela aumenta seu volume, passando à fase de gigante. Este processo prossegue em uma sequência em que cada camada converte um elemento em outro, avançando para os elementos mais pesados, até que finalmente o núcleo

estelar passa a converter silício em ferro. Nessa fase a conversão cessa, pois sendo o ferro o elemento mais estável, a sua fusão em outros elementos consome energia em vez de produzi-la, e as camadas acima do núcleo assumem o que os astrônomos chamam de “estrutura de cebola”, em que a camada de cima fornece o combustível para a camada de baixo. Esta pode ser vista esquematicamente na figura 2.1.

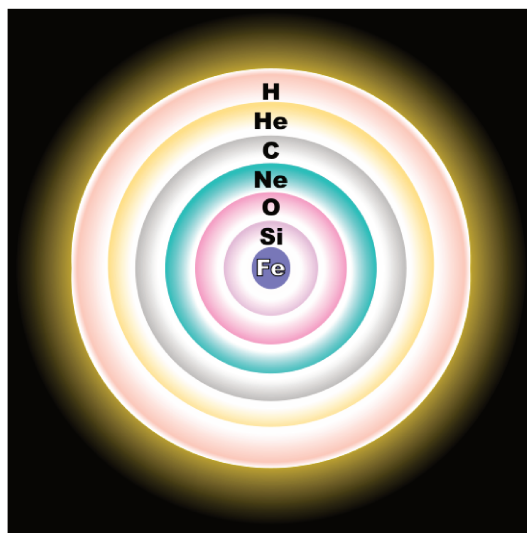


Figura 2.1: *A Estrutura de cebola evidenciando as camadas de fusões sucessivas, cortesia de R.J.Hall.*

Sem a energia vinda da fusão o núcleo estelar não é mais capaz de sustentar a atração da gravidade, e assim as camadas superiores acabam por desabar sobre o núcleo, essa compressão repentina leva a um super-aquecimento do núcleo, o que reativa a fusão por um curto período de tempo, isso leva a uma expansão violenta das camadas superiores que são ejetadas para o espaço aberto num fenômeno chamado de supernova. Seguindo a classificação astronômica atual esta é uma supernova do tipo II[1], resultante do colapso de uma estrela massiva. A supernova costuma durar alguns dias até que se apague totalmente, mas durante esse período a estrela emite mais energia do que nos milhões de anos em que brilhou normalmente, mas estima-se

que 99% dessa energia seja emitida na forma de neutrinos, mesmo assim a fração emitida na forma de radiação eletromagnética chega a superar o brilho da galáxia na qual a supernova se localiza. Um exemplo é a da supernova SN 1994D, que esta na figura 2.2. Quanto ao núcleo estelar sobrevivente, se ele apresentar uma massa (observada) entre 1,4 e 3,2 massas solares, ele poderá se tornar uma estrela de nêutrons sobrevivendo graças à pressão de degeneração dos nêutrons, como estes são férmions há um limite do número de estados que podem ser ocupados, assim surge uma força de repulsão que balança a gravidade; caso apresente uma massa maior ou não consiga deter o colapso, o núcleo consequentemente irá se tornar um buraco negro estelar.



Figura 2.2: A Supernova SN 1994D (ponto brilhante na esquerda inferior), à 108 milhões de anos-luz. Costesia da NASA.

2.2 Características Físicas

No momento de seu nascimento a estrela de nêutrons possui uma temperatura da ordem de $10^{14}K$, mas devido à intensa emissão de neutrinos a temperatura cai para 10^6K [1]. Como dito anteriormente, este corpo é um núcleo estelar sobrevivente, uma massa pouco maior que a

massa solar num raio médio de 12km, assim sua densidade fica da ordem de $5 \times 10^{15} g/cm^3$, o que já é acima da densidade nuclear. Logo após seu nascimento a imensa atração da gravidade faz com que os átomos constituintes sejam literalmente comprimidos ao ponto em que suas eletrosferas se sobrepõem e é superada a repulsão de Fermi entre os elétrons. Nesse ponto os elétrons são forçados a órbitas cada vez mais próximas ao núcleo atômico, assim são capturados pelos prótons e são convertidos em nêutrons pelo processo beta-inverso:

$$p^+ + e^- \rightarrow n + \nu_e. \quad (2.1)$$

Este processo gera um excesso de nêutrons, que são absorvidos pelos núcleos de ferro próximos até que estes fiquem saturados, e assim os nêutrons residuais são deixados livres pelo meio circundante, é por causa desse excesso de nêutrons que ela recebe a denominação de estrela de nêutrons. Este processo acentua-se conforme avança-se para o interior da estrela, levando a uma divisão em camadas que determinam as características observacionais das estrelas de nêutrons observadas. Observacionalmente dividem-se as estrelas de nêutrons em duas categorias, descritas à seguir.

2.2.1 Pulsares

Estrelas jovens que apresentam um gigantesco campo magnético ($\sim 10^{40} Gauss$) aliado a um curto período de rotação ($\sim 1ms$), este gigantesco campo magnético captura as partículas carregadas próximas à estrela e as acelera até os polos magnéticos, onde colidem brutalmente com a superfície emitindo toda a energia cinética na forma de radiação polarizada, podendo emitir raios-x e gama. Devido à alta rotação da estrela este feixe de radiação projeta um cone no espaço que quando detectado é visto como um pulso periódico de radiação (ver figura 2.5), por isso leva o nome de pulsar. Convém dizer que em inglês pulsar é um neologismo resultante da contração de *pulsating star*, que foi a denominação original dada a esses corpos recém descobertos, mas em português há o verbo *pulsar* que coincidiu ter um significado apropriado, por isso não houve a tradução literal de *pulsating star*. Aqui gostaria de falar um pouco sobre

o pulsar mais bem estudado, o pulsar dentro da Nebulosa do Caranguejo. Esta nebulosa é

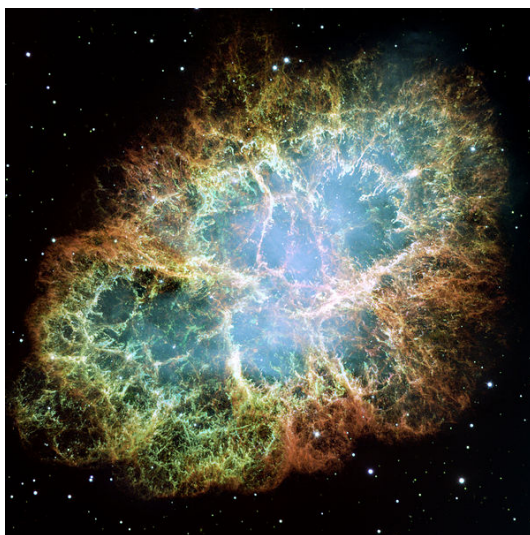


Figura 2.3: A Nebulosa do Caranguejo, imagens em cores falsas. Cortesia da NASA.

formada pelos restos de uma supernova que foi observada no ano de 1054, e foi catalogada por astrônomos chineses e árabes. O interessante é que como na Europa da época havia a crença de que o céu astronômico era imutável, esta supernova que perdeu seu brilho em poucos dias foi simplesmente ignorada, e interpretada pelos astrônomos como uma “pequena brincadeira por parte dos céus”. Mas enfim com o desenvolvimento dos rádio-telescópios no começo da década de 1950, voltou-se à observar essa nebulosa devido aos sinais de rádio extremamente colimado e periódico que vinham de seu centro. Uma imagem moderna é mostrada na figura 2.3. Como nessa época começaram os estudos sérios a respeito de objetos astronômicos superdensos (anãs brancas, estrelas de nêutrons e buracos negros), voltaram-se para essa nebulosa, pois como foi proposto na época que esses objetos nasceriam de supernovas, nada mais óbvio do que procurar por um deles nos restos de uma. Seguindo as estimativas da massa da nebulosa pode-se estimar a massa da estrela progenitora, que atualmente é estimada entre 9 e 11 massas solares, e uma estrela dessa massa apresenta grandes chances de deixar uma estrela de nêutrons

como remanescente. Os rádio-telescópios da época estimaram o diâmetro da fonte emissora de rádio como sendo aproximadamente 1 ano-luz, o que é extremamente grande para uma estrela de nêutrons, assim a hipótese de que fosse uma foi descartada e o mistério sobre o que havia dentro desta nebulosa perdurou por mais algumas décadas. Atualmente graças ao observatório de Raios-X Chandra sabemos dos detalhes sobre esta nebulosa e seu pulsar. Em seu interior há um pulsar com raio ótico estimado em 25km, com dois poderosos jatos de radiação que saem dos polos magnéticos do pulsar. Como este pulsar é muito jovem do ponto de vista astronômico, possui menos de mil anos de idade, ele ainda conserva grande parte de seu momento angular e de seu poderosíssimo campo magnético, este campo se estendo por todo o espaço e conforme segue a rotação do pulsar ele arrasta consigo todo o gás ionizado que esta em volta do pulsar. Como os elétrons são as partículas carregadas mais leves eles são acelerados até velocidades próximas a da luz, e é isto que gera a radiação de rádio que é aqui detectada. Como os rádio-telescópios da década de 1950 não eram tão sensíveis se comparados aos atuais, eles não foram capazes de resolver toda a nuvem, assim não viram que era o pulsar que estava dentro da nuvem que gerava toda essa emissão de rádio. A figura 2.4 mostra uma imagem obtida pelo Chandra em que pode-se distinguir facilmente o pulsar central, os feixes de radiação e a nuvem circundante.

2.2.2 Radio-quietos

São estrelas mais antigas que perderam seu campo magnético e tiveram seu período de rotação reduzido, não podendo produzir mais pulsos de radiação. Como apresentam temperatura média de $10^6 K$, elas emitem radiação de corpo negro no pico do raio-x e assim podem ser detectadas.

2.3 Estrutura

Embora nunca se tenha estudado diretamente a estrutura interna de uma estrela de nêutrons, ela é convencionalmente dividida em quatro camadas, que são propostas segundo nossos conhecimentos atuais sobre como a matéria se comporta nesses estados de densidade, além das



Figura 2.4: O pulsar da Nebulosa do Caranguejo e a nuvem que gira junto com ele. Foto obtido pelo observatório Chandra, cortesia da NASA.

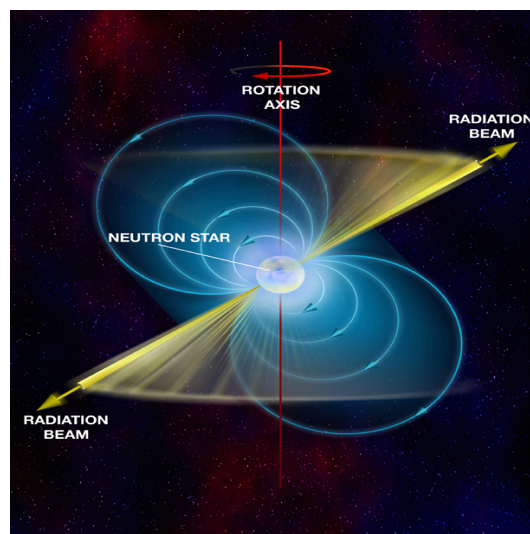


Figura 2.5: O esquema de um pulsar, mostrando as linhas de campo e o feixe de radiação. Cortesia de www.clubedecienciasastronomia.blogspot.com.br.

leis básicas da termodinâmica. Essas camadas são definidas segundo sua faixa de densidade e variam segundo a definição de cada autor, mas em geral concordam segundo as características gerais, um esquema geral pode ser visto na figura 2.6([2],[3]).

2.3.1 Crosta Externa

A primeira das camadas é que esta em contato direto com o espaço aberto, numa estrela de nêutrons típica de 12km de raio ela costuma ter apenas 0,5km e retém somente $10^{-10}\%$ da massa da estrela. Sua faixa de densidade é de $3,3 \times 10^7$ até $9,6 \times 10^{10}$ (em g/cm^3)([3]). É constituída principalmente de íons de ferro e níquel que sobreviveram ao colapso do antigo núcleo estelar imersos numa rede cristalina de elétrons degenerados devido à alta pressão à qual as eletrosferas foram submetidas. Esta camada está sujeita a súbitas variações em sua estrutura resultantes da ação do campo magnético externo e até de impactos de corpos externos à estrela, essas variações recebem o nome de *starquakes*. Qualquer matéria que caia na estrela será rapidamente comprimida e esmagada até que se converta na mesma matéria constituinte da crosta, é importante lembrar que isso só é possível devido à gigantesca atração gravitacional das camadas inferiores, se fosse possível retirar uma amostra da estrela, essa amostra iria se desfazer devido à repulsão fermiônica, por isso a denominação de *estrela*, pois ela se mantém graças à atração gravitacional, e não devido à ação da força forte como pensava-se no início dos estudos destes corpos.

2.3.2 Crosta Interna

Logo após vem a crosta interna, caracterizada pelo *Neutron drip* (gotejamento de nêutrons), nessa região a densidade e a concentração de nêutrons são tais que pequenos agrupamentos de nêutrons ligados através da atração forte sobrevivem fora dos núcleos atômicos e criam uma rede cristalina semelhante à dos elétrons. Os elétrons ganham energia necessária para ficarem livres e se aproximam do limite relativístico. Possui uma profundidade média de 1km e densidade de $3,5 \times 10^{11} \text{g/cm}^3$ até $1,3 \times 10^{14} \text{g/cm}^3$. Nessa região os núcleos de ferro e níquel absorvem o excesso de nêutrons até atingirem o máximo possível e ficam supermassivos em relação aos

seus isótopos terrestres([3]).

2.3.3 Núcleo Externo

Tendo profundidade de 2,5km e densidade de $1,3 \times 10^{14} g/cm^3$ até $4,0 \times 10^{15} g/cm^3$ é a região onde o gigantesco campo magnético tem origem. Nessa faixa de densidade os núcleos atômicos foram literalmente esmagados uns contra os outros, fazendo os prótons e nêutrons se dissociarem num superfluido nuclear. Os prótons degenerados passam a ser um fluido supercondutor, pois os prótons possuem carga elétrica, ao girarem na alta rotação da estrela eles geram um campo magnético, que é imediatamente expulso para as camadas externas devido a propriedade que os supercondutores apresentam de expulsar linhas de campo magnético. Os nêutrons passam a ser um superfluido ligado pela força forte. Nesta alta densidade os elétrons livres se tornam ultra-relativísticos e atingem a energia mínima para se converterem em múons, passando a viver em equilíbrio com estes([3]).

2.3.4 Núcleo Interno

Constitui a maior parte da estrela, onde se concentra a maior parte de sua massa e energia, com densidade maior que $4,0 \times 10^{14} g/cm^3$, não há consenso sobre como é sua composição ou qual seria o estado da matéria em seu interior. Cada autor se vê livre para deduzir sua composição, alguns argumentam que os prótons e nêutrons se dissociam em quarks e passam a conviver num plasma de quarks-glúons; ainda argumentam que os quarks *up* e *down* constituintes dos prótons e nêutrons transformam-se através da interação fraca em quarks *strange*. Como esses são mais massivos, a conversão de energia cinética em energia de repouso iria estabilizar a matéria, que assim existiria nesse estado “estranho”; há ainda a hipótese que prótons e nêutrons se rearranjariam como mésons Káons e Píons, pois como mésons todos poderiam ocupar o mesmo espaço de fase e assim estabilizar a estrela([3],[4]).

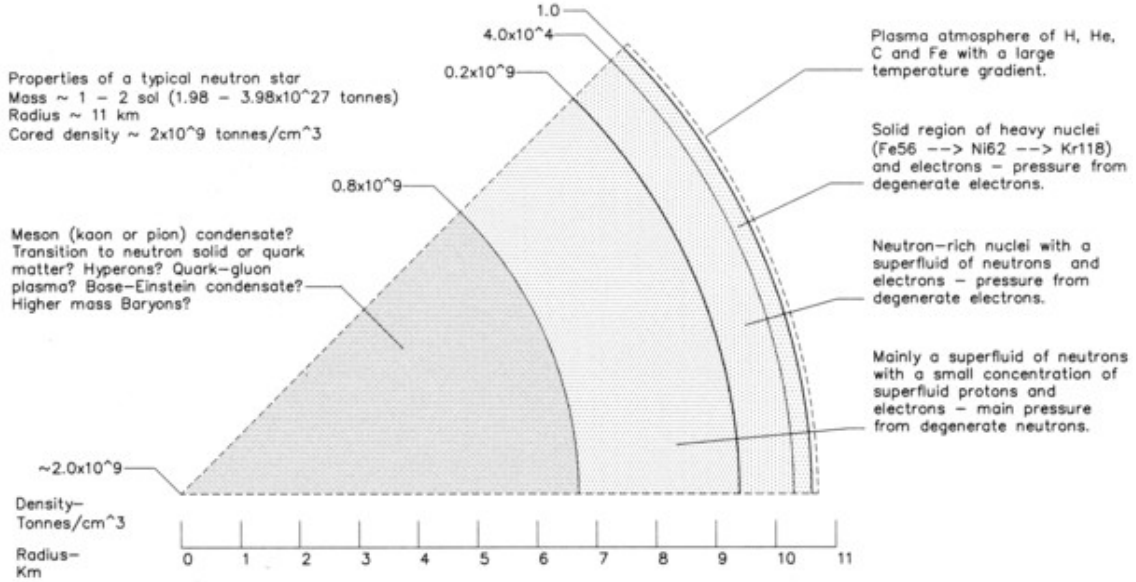


Figura 2.6: Esquematização da estrutura de um pulsar. P.Haensel.

2.4 Equação de Estado

Para a descrição completa é necessário ligar a densidade num dado ponto à sua respectiva pressão, há várias maneiras de se fazer isso, como o uso de equações bariônicas, equações politrópicas e ainda através de tabelas com resultados experimentais de física nuclear aliada à física de altas energia e partículas (obtidos de modelos teóricos com os resultados dos experimentos com aceleradores de partículas). Um dos modelos mais simples é o do politropo([5]).

2.4.1 Politropos

Politropos são simples nuvens de gás que se mantêm coesas pela atração gravitacional([6]). Foram um dos primeiros modelos utilizados em astronomia devido à sua simplicidade, por utilizarem a teoria da gravitação Newtoniana são aplicados principalmente em corpos cujos os efeitos da relatividade geral são desprezados, e assim são descritos pela Equação de Lane-Emden:

$$\frac{1}{\xi^2} \frac{d}{d\xi} \left(\xi^2 \frac{d\theta}{d\xi} \right) + \theta^n = 0, \quad (2.2)$$

sendo ξ um raio adimensional e θ esta ligado à densidade por $\rho = \rho_c \theta^n$, sendo ρ_c a densidade bariônica central e com as condições de contorno $\theta(0) = 1$ e $\theta'(0) = 0$ ([6],[7]). A densidade bariônica fica conectada à pressão pela equação politrópica

$$P = K \rho^{1+\frac{1}{n}}, \quad (2.3)$$

onde n é o índice politrópico e K é a constante que define a pressão central. Para estrelas de nêutrons politrópicas, n varia de 0,5 à 1,0 ([8]) (essa descrição falha em não dividir a estrelas em camadas e ignora sua composição química). Aqui usaremos este modelo simples como controle para a nossa equação de estado realista.

2.4.2 Equações de estado realistas

Equações de estado realistas são obtidas da análise dos dados experimentais dos colisores de partículas. No nosso caso, como estamos tratando de estrelas de nêutrons precisamos de uma matéria rica em nêutrons, mas a matéria comum a que temos acesso não cumpre esse requisito, por isso precisamos de dados vindos de experiências com núcleos atômicos. Nos colisores de partículas núcleos de átomos pesados são acelerados e colididos entre si, durante o curto período em que ocorre a colisão obtêm-se um regime de densidade e pressão esperado para uma estrela de nêutrons, depois que o sistema se dispersa os resultados do detectores são comparados e analisado segundo nosso conhecimento sobre física nuclear no regime das forças forte e fraca. São graças a esses experimentos que podemos prever a estrutura de uma estrela de nêutrons mesmo nunca tendo acesso direto a uma. Para nosso trabalho usamos as equações de estado de P. Haensel e B. Pichon ([9]), para a descrição da crosta externa e F. Douchin e P. Hanesel ([10]) para a descrição da crosta interna e núcleo, e assim como neste trabalho citado denominaremos esta equação de estado com SLy. Com estes dados tabelados encontramos todos os resultados intermediários entre dois pontos com a interpolação logarítmica descrita em W. David Arnett

e Richard L. Bowers ([11]),

$$\frac{\log P - \log P_j}{\log \rho - \log \rho_j} = \frac{\log P_{j+1} - \log P_j}{\log \rho_{j+1} - \log \rho_j}, \quad (2.4)$$

sendo P e ρ a pressão e a densidade que desejamos calcular, aqui pode-se calcular um usando o outro parâmetro como entrada, sendo ρ_j e ρ_{j+1} os pontos tabelados onde ρ esta inserido e P_j e P_{j+1} os pontos onde P esta inserido. Nós usaremos esta equação de estado pois ela é bem conhecida e utilizada na área, sendo que várias de nossas bibliografias a usam como equação de estado. Nas figuras 2.7 e 2.8 vemos uma comparação de nossas duas equações de estado utilizadas, apresentando a pressão e o índice adiabático em função da densidade total de energia. Vejam que como comentado a equação politrópica é a mais simples e isso fica claro nas comparações.

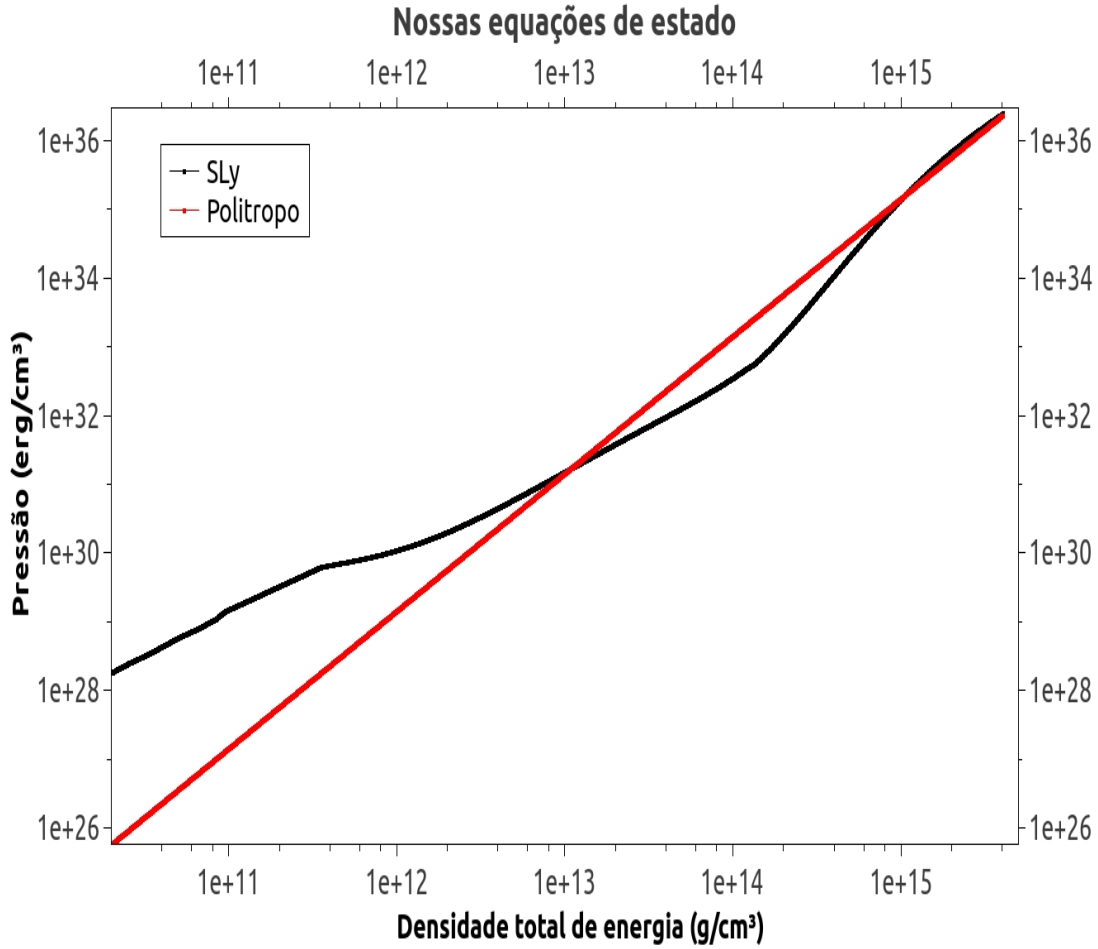


Figura 2.7: A pressão em função da densidade total de energia.

2.4.3 Equações TOV

Como estrelas de nêutrons são corpos em que os efeitos da Relatividade Geral não podem ser ignorados, a Equação de Lane-Emden não é mais apropriada para a sua descrição, por isso usam-se as equações TOV (Tolman–Oppenheimer–Volkoff) ([12],[13]), pois descrevem corretamente corpos de simetria esférica nos quais os efeitos da Relatividade Geral não podem ser desprezados. Iniciemos com a métrica de Schwarzschild, que descreve a métrica exterior a um corpo esférico

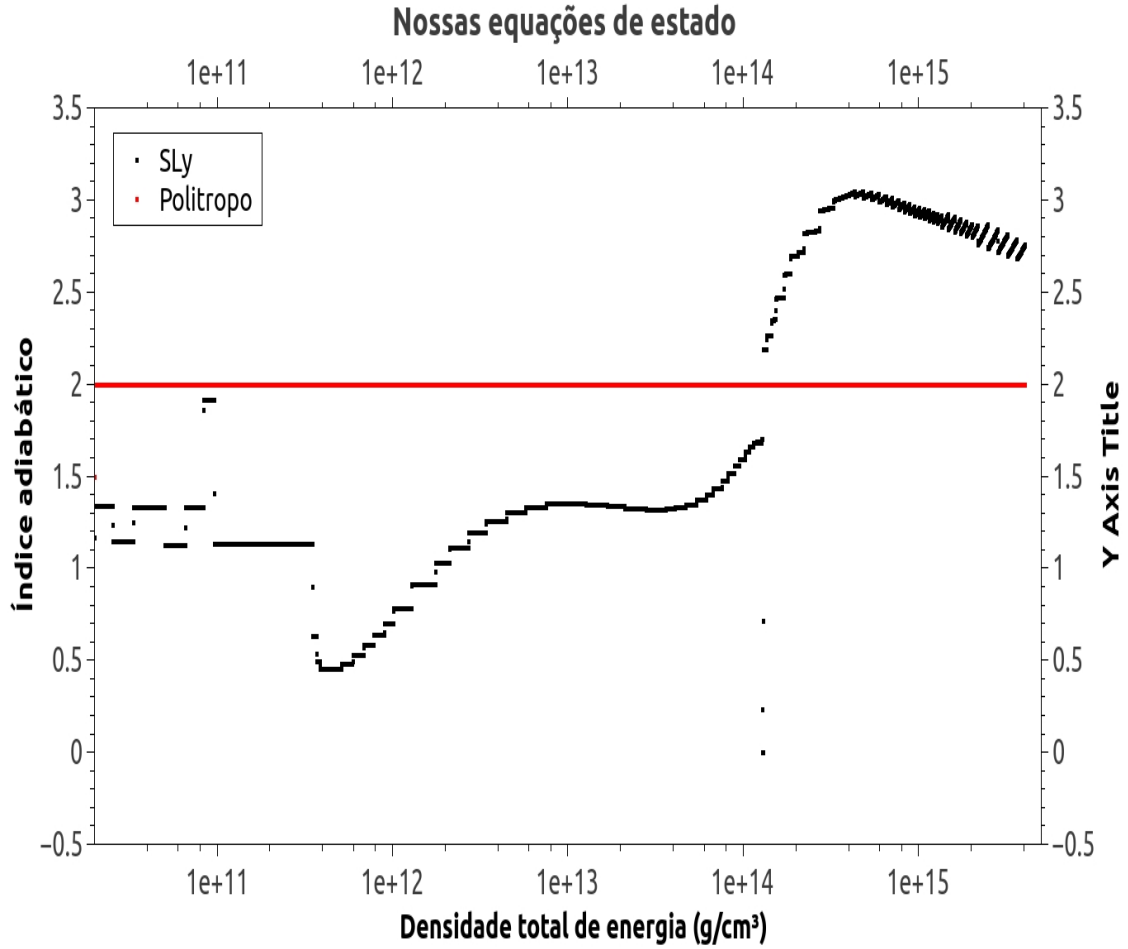


Figura 2.8: O índice adiabático em função da densidade total de energia.

que não rotaciona:

$$ds^2 = -\left(1 - \frac{2GM}{c^2 r}\right)(cdt)^2 + \left(1 - \frac{2GM}{c^2 r}\right)^{-1} dr^2 + r^2(d\theta^2 + \sin^2\theta d\phi^2), \quad (2.5)$$

seguindo a notação matemática convencional da Teoria da Relatividade Geral, essa métrica pode ser escrita na seguinte forma (esta notação tem por objetivo facilitar a manipulação matemática das equações):

$$ds^2 = -e^{2\Phi} dt^2 + e^{2\Lambda} dr^2 + r^2 d\Omega^2. \quad (2.6)$$

Convém dizer que por enquanto estamos usando o sistema de unidades naturais em que $c=G=1$, após o final da dedução as equações serão apresentadas nas unidades convencionais. Usando esta métrica o Tensor de Einstein toma a forma de uma matriz diagonal com os seguintes elementos:

$$G_{00} = \frac{1}{r^2} e^{2\Phi} \frac{d}{dr} [r(1 - e^{-2\Lambda})], \quad (2.7)$$

$$G_{rr} = -\frac{1}{r^2} e^{2\Lambda} (1 - e^{-2\Lambda}) + \frac{2}{r} \Phi', \quad (2.8)$$

$$G_{\theta\theta} = r^2 e^{-2\Lambda} \left[\Phi'' + (\Phi')^2 + \frac{\Phi'}{r} - \Phi' \Lambda' - \frac{\Lambda'}{r} \right], \quad (2.9)$$

$$G_{\phi\phi} = \sin^2 \theta G_{\theta\theta}, \quad (2.10)$$

Onde o índice ' indica derivação em relação à coordenada radial r .

O Tensor de Einstein é um dos componentes da Equação de Einstein, que descreve como a estrutura do espaço-tempo irá se comportar numa região próxima a uma dada distribuição de massa-energia, convém dizer que na Teoria da Relatividade Geral o que causa a distorção local na métrica é a concentração de energia, e a isto está ligada a famosa relação $E = mc^2$ que liga a massa de repouso de um objeto à sua energia de repouso, na maior parte dos casos a energia resultante da massa supera em muitas ordens os outros tipos de energia (cinética, térmica, elétrica...) mas no nosso regime estas não podem ser desprezadas mas já se encontram implícitas na equação de estado. Na teoria Newtoniana somente a massa de repouso é a fonte da gravidade. Além do Tensor de Einstein precisamos do Tensor de Energia-Momento que descreve a distribuição e o fluxo de energia-momento pela estrutura 4-dimensional do espaço-tempo, uma vez que na relatividade geral os fluxos de energia também geram contribuições a distorção do espaço-tempo. Para um fluido perfeito de 4-velocidade \vec{U} numa métrica $g^{\alpha\beta}$, o Tensor Energia-Momento assume a forma matemática genérica:

$$T^{\alpha\beta} = (\rho + p) \vec{U} \otimes \vec{U} + p(g^{\alpha\beta})^{-1}, \quad (2.11)$$

onde ρ é sua densidade de matéria e p é sua pressão, os índices α e β correm de 0 a 3.

Aqui convém dizer que a densidade de matéria (bariônica) é a densidade de massa de repouso medida por um observador distante, enquanto que a densidade total de energia é a distribuição de energia por todo o corpo.

Com esta definição o tensor assume a forma de uma matriz diagonal com os seguintes elementos:

$$T_{00} = \rho e^{2\Phi}, \quad (2.12)$$

$$T_{rr} = p e^{2\Lambda}, \quad (2.13)$$

$$T_{\theta\theta} = r^2 p, \quad (2.14)$$

$$T_{\phi\phi} = \sin^2 \theta T_{\theta\theta}, \quad (2.15)$$

aplicando as estas equações as regras de conservação e substituindo-as ao Tensor de Einstein encontraremos as equações TOV.

Iniciemos com a conservação do Tensor energia-momento num referencial local:

$$T^{\alpha\beta}_{;\beta} = 0, \quad (2.16)$$

devido a simetria da métrica somente o termo para $\alpha = r$ não é nulo, e isto gera:

$$(\rho + p) \frac{d\Phi}{dr} = -\frac{dp}{dr}, \quad (2.17)$$

esta equação nos diz o gradiente de pressão necessário para equilibrar o campo gravitacional e impedir que o corpo colapse.

Agora conectemos o Tensor de Einstein ao Tensor energia-momento através da Equação de Einstein:

$$G^{\alpha\beta} = 8\pi T^{\alpha\beta}, \quad (2.18)$$

além disso faz-se esta conveniente definição:

$$e^{2\Lambda} = \left(1 - \frac{2m(r)}{r}\right)^{-1}. \quad (2.19)$$

Agora resolvendo as equações 2.7 e 2.12 através da equação 2.18 encontramos:

$$\frac{dM(r)}{dr} = 4\pi r^2 \rho, \quad (2.20)$$

que descreve a massa em função do raio.

Por fim substituindo as equações 2.8 e 2.13 encontramos:

$$\frac{d\Phi}{dr} = \frac{m(r) + 4\pi r^3 p}{r[r - 2m(r)]}. \quad (2.21)$$

Agora recolocando as constantes finalmente encontraremos as Equações TOV que descrevem nossa estrela:

$$\frac{dM(r)}{dr} = 4\pi r^2 \rho(r), \quad (2.22)$$

$$\frac{dP(r)}{dr} = -\frac{G}{r^2} \left(\rho(r) + \frac{P(r)}{c^2} \right) \left(M(r) + 4\pi r^3 \frac{P(r)}{c^2} \right) \left(1 - \frac{2GM(r)}{c^2 r} \right)^{-1}, \quad (2.23)$$

$$\frac{d\Phi(r)}{dr} = -\left(\frac{2}{P(r) + \rho c^2} \right) \frac{dP(r)}{dr}. \quad (2.24)$$

Uma das condições de estabilidade vinda das equações precedentes é que uma estrela é estável se a sua massa aumentar com a densidade central, ou seja $\frac{dM}{d\rho_c} > 0$. Esse é um conjunto de equações acopladas e não apresentam solução analítica, assim usam-se métodos numéricos para integrá-la e descrever o interior da estrela com as seguintes condições de contorno: $\rho(r=0) = \rho_c$ e $P(r=R_{max}) = 0$.

2.5 Resultados e comparação

A seguir mostraremos os resultados da integração das equações TOV usando as equações de estado politrópica e a equação de estado SLy tabelada ([9],[10]). A densidade de energia total central usada para ambos os casos é de $7,32 \times 10^{14} g/cm^3$, esta densidade foi escolhida pois com ela obtemos para a equação de estado SLy uma estrela de uma massa solar. Para a equação politrópica usou-se $K = 145358 dyne.cm^4 g^{-2}$, e $n = 1$ que é o índice para uma estrela de nêutrons composta somente de nêutrons não relativísticos e não interagentes.

2.5.1 Massa

Para a equação de estado politrópica a massa e o raio estelar calculados para esta estrela de nêutrons foram de $1,307M_\odot$ e $14,65km$, enquanto que para a SLy a massa e o raio estelar

calculados foram de $1,00M_{\odot}$ e $11,88km$. Na figura 2.9 apresentamos os dois gráfico juntos para comparação, com a massa em massas solares e o raio em quilômetros.

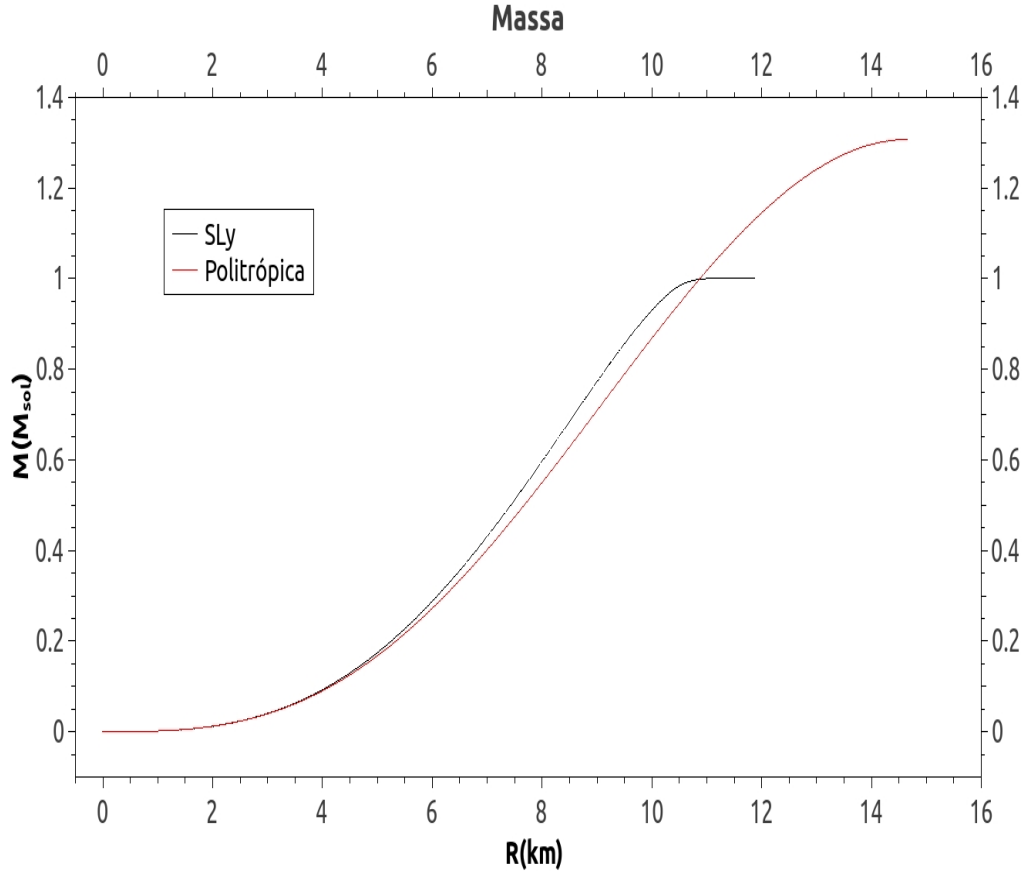


Figura 2.9: A distribuição de massa para as estrelas.

Vemos que embora as massas sejam diferentes, o comportamento da distribuição de massa é semelhante para os dois casos pois a equação de distribuição de massa é a mesma para ambos os casos (eq. 2.22). A suavidade da distribuição politrópica em relação a SLy deve-se ao fato da estrela politrópica ter um raio maior, assim há um volume maior por onde a massa se distribui quase linearmente, como mostra o comportamento do gráfico. Essa distribuição quase linear de massa também aparece na estrela SLy, sendo um pouco mais acentuada.

A região próxima da origem onde a massa cresce rapidamente com o raio para ambas as curvas, a concavidade positiva representa o núcleo das estrelas, onde as densidades extremas, várias ordens de grandeza acima das densidades nucleares convencionais, causam uma grande variação da massa em função do raio, mas como o núcleo apresenta um pequeno volume se comparado com o resto da estrela, ele não contribui grandemente com a massa total.

A região logo após, onde a massa cresce quase que linearmente é o manto da estrela, onde se concentra a maior parte da massa, nesta região a densidade apresenta um comportamento bem comportado, por isso a distribuição de massa também o é.

A última região, próxima à superfície é a crosta da estrela, a concavidade negativa indica que a massa esta se estabilizando, a densidade não mais se altera e para os padrões estelares ela fica praticamente constante. Veja que o fim é mais acentuado na estrela SLy pois na sua equação de estado já estava implícito a fina crosta, enquanto que a estrela politrópica é mais suave.

A diferença de massa é devido ao comportamento da matéria que está implícito em cada equação, no caso da equação politrópica o índice escolhido, $n = 1$, é usado para o tratamento de nêutrons não relativísticos e não interagentes, neste regime não há grande confinamento de matéria e a densidade se torna menor. Já a equação SLy, por ser mais realista, contém estes comportamentos e naturalmente prediz uma matéria mais densa que a da equação politrópica.

2.5.2 Pressão

Na figura 2.10 podemos ver a distribuição de pressão pelo interior da estrela para os dois casos.

Estas distribuições são semelhantes entre si mas a pressão para a equação politrópica é maior para um mesmo raio se comparada a SLy, veja que como o raio da estrela politrópica é maior ela apresenta uma região de baixa pressão maior do que a estrela SLy. Veja também que elas apresentam a pressão central semelhantes quanto à ordem de grandeza, isso significa que a estrela SLy é mais compacta do que a estrela politrópica. Como num caso semelhante à da massa, vemos que na primeira região do gráfico, o núcleo estelar, a pressão varia fortemente com

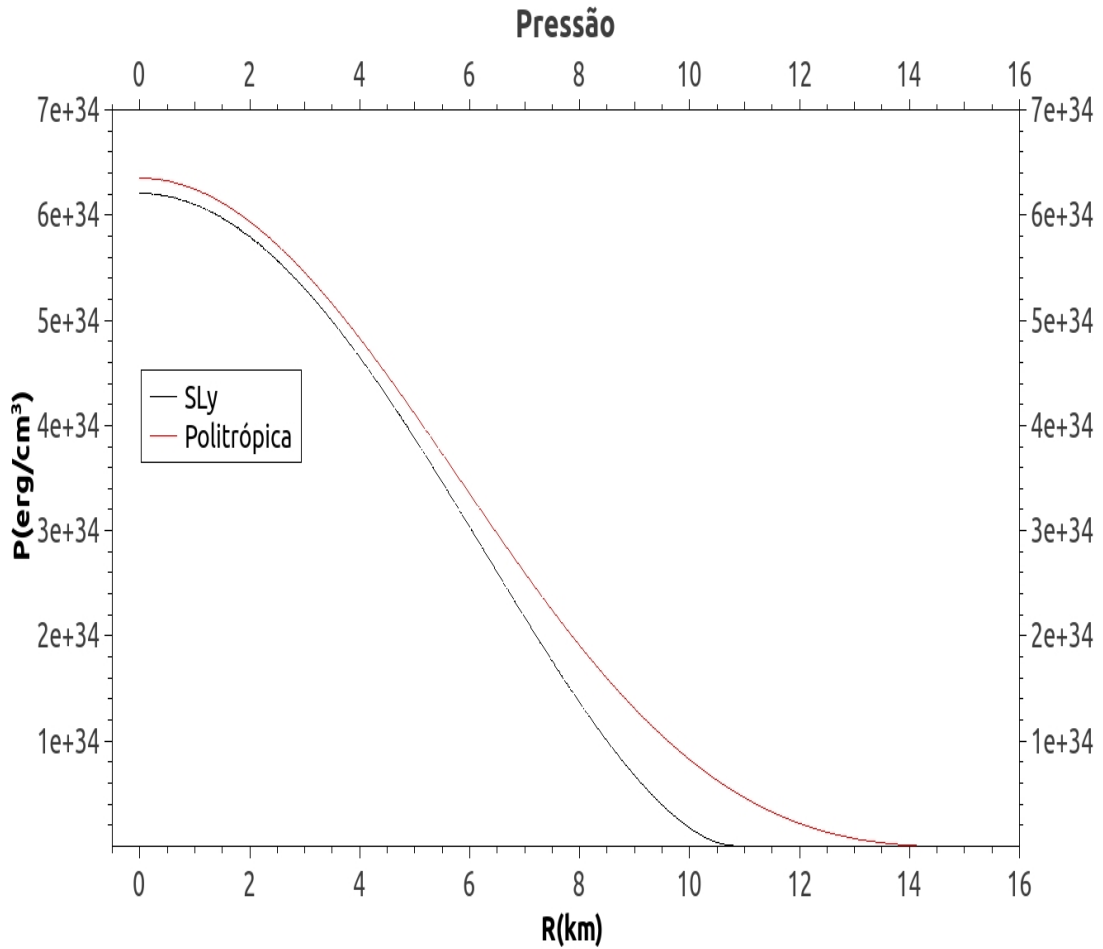


Figura 2.10: A distribuição de pressão para as estrelas.

o raio, pois aí a densidade também varia fortemente. Na próxima região, o manto, a pressão cai de forma contínua e constante, sendo bem próxima a um decaimento linear. Por fim na última região, a crosta, a pressão decai rapidamente até atingir a superfície, sendo que este decaimento é mais suave na estrela politrópica do que na estrela SLy devido ao comportamento das diferentes equações de estado.

Convém esclarecer algo, pela definição de estrela politrópica esta é uma massa contínua de

matéria sem interfaces, logo uma simples estrela politrópica não apresenta atmosfera, portanto durante a integração usamos como condição para determinar o raio da estrela o ponto onde a pressão se anula, pois daí em diante seria o espaço vazio. Uma estrela real possui sim uma região que poderíamos definir como atmosfera, mas sua composição e tamanho diferem muito da nossa noção corrente de atmosfera usada para o nosso planeta. Observações precisas do espectro da luz emitida por estrelas de nêutrons sugerem que exista uma fina camada atmosférica composta principalmente de vapor de ferro ionizado, embora não saibamos quais são os efeitos dessa atmosfera nas nossas simulações, suspeitamos que ela influencia na medição do raio nas estrelas de pouca massa, pois aí ela teria uma espessura maior, enquanto que nas estrelas massivas, ela passaria por um efeito de compactação graças à alta gravidade e se reduziria à uma pequena camada de alguns metros. Para a estrela SLy a condição para determinar o raio foi o ponto onde a pressão encontraria essa atmosfera de ferro, sendo aproximadamente $3,8330 \times 10^{25} \text{ erg/cm}^3$ ([3], [9]).

Por este motivo, o gráfico da pressão SLy não chega a tocar o eixo zero.

A unidade de pressão utilizada, erg/cm^3 , é a unidade padrão para pressões muito acima dos limites nucleares, pois nestes regimes a energia média por partícula é da ordem da energia de repouso, assim pode-se dizer que é a própria energia de interação das partículas que gera a pressão a que estão submetidas, por isso o uso de erg , que é unidade de energia, e cm^3 , que é unidade de volume.

A unidade erg/cm^3 é equivalente à dyne/cm^2 .

2.5.3 Densidade

Na figura 2.11 vemos a distribuição de densidade de energia pelo interior da estrela para os dois casos.

Desta vez as distribuições não compartilham das mesmas semelhanças se comparadas às distribuições anteriores.

Vemos que para a distribuição da estrela politrópica é mais semelhante as distribuições anteriores, com a parte inicial do gráfico caracterizada por uma forte dependência em relação

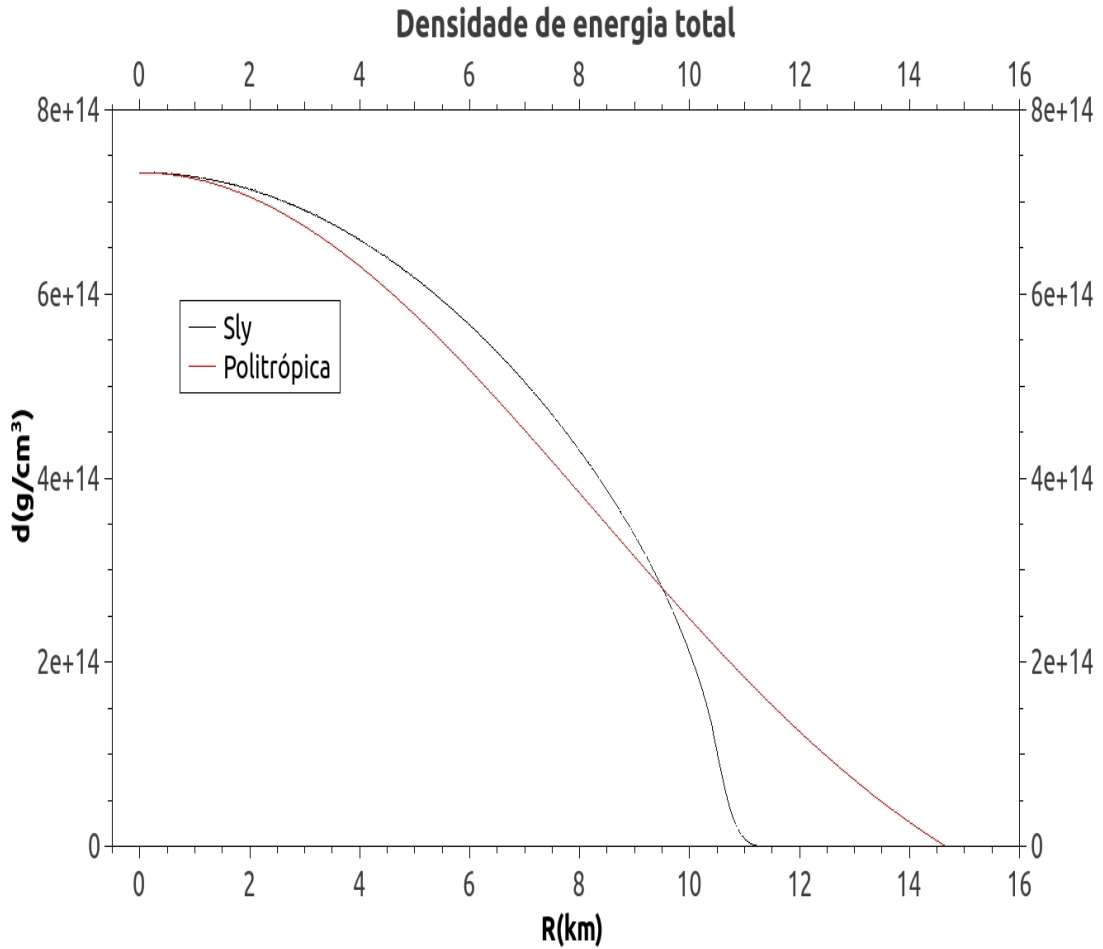


Figura 2.11: A distribuição de densidade para as estrelas.

ao raio, seguido de uma queda suave e linear e depois uma queda súbita característica da crosta.

A distribuição SLy também apresenta uma forte dependência no início, seguida de uma queda acentuada na região do manto, perto de $11 km$ vemos um pequeno desvio onde há outra queda brusca da densidade que rapidamente chega à zero. Isto decorre da integração ter atingido a atmosfera de ferro, e a curta região antes dela é a fina crosta.

Aqui as unidades são as usuais, g/cm^3 .

2.5.4 Índice Adiabático

Na figura 2.12 vemos a distribuição do índice adiabático pelo interior da estrela SLy.

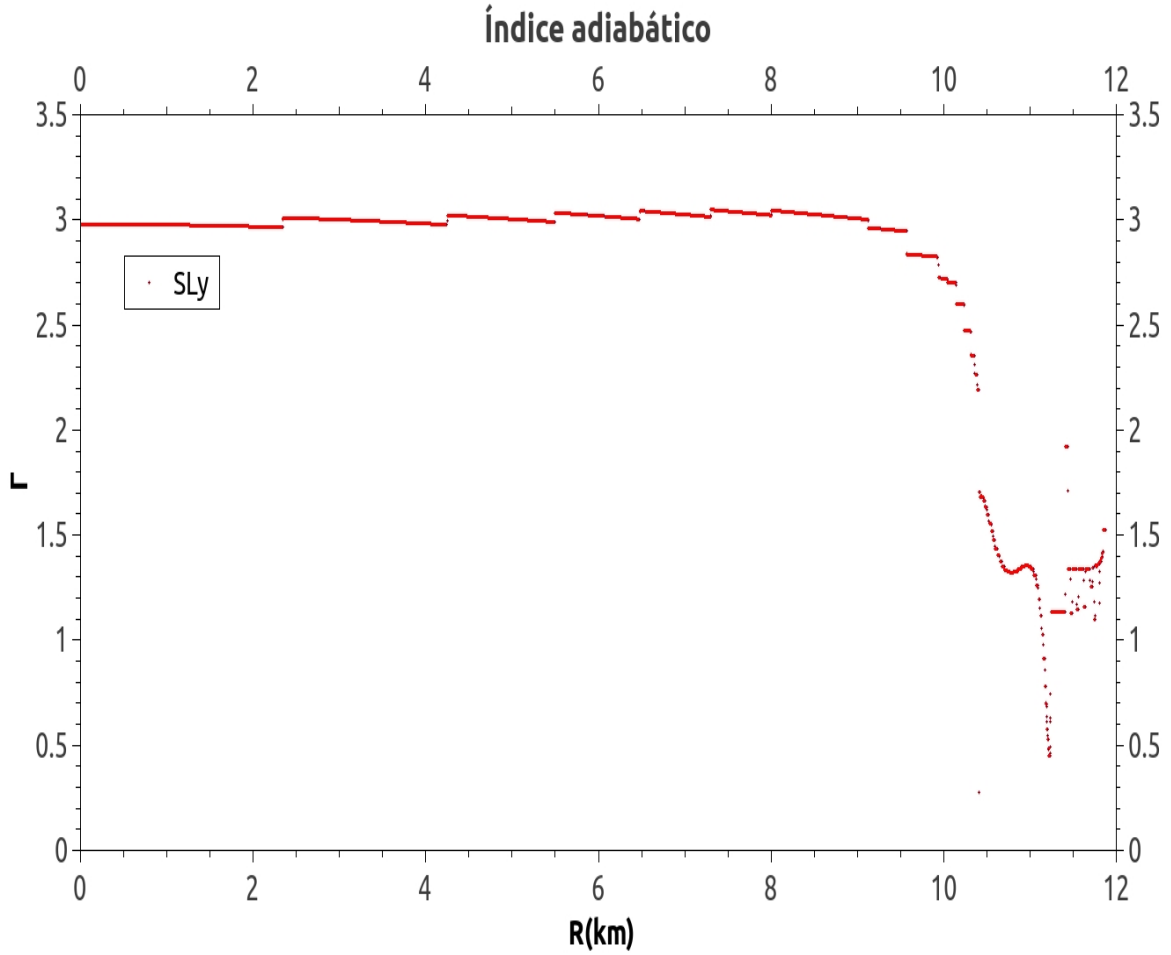


Figura 2.12: A distribuição do índice adiabático para a estrela SLy.

O índice adiabático é o fator γ nas transformações adiabáticas da forma $PV^\gamma = Cte$, este será utilizado para o cálculo das ondas gravitacionais emitidas pela duas estrelas. Veja que pela definição de transformação adiabática e de processo politrópico temos que $n = \frac{1}{\gamma-1}$, assim nossa estrela politrópica possui seu índice adiabático constante e igual a 2.

Para a estrela SLy usamos a seguinte definição[10]:

$$\gamma = \left(\frac{\rho + Pc^{-2}}{P} \right) \frac{\Delta P}{\Delta \rho};$$

donde vemos que o índice é variável conforme avança-se pelo interior da estrela.

Na distribuição da estrela SLy vemos que na maior parte da estrela o índice mantém-se na faixa de 3, sendo que logo após $10km$ há uma súbita queda, esta é a região onde o manto termina e onde a crosta começa, ai os núcleos atômicos voltam a existir. Vejam que já próximo à superfície o índice varia radicalmente e não há uma curva analítica que possa ajustá-lo. O índice médio, 3, é bem diferente do índice para a estrela politrópica, 2. Segundo nossa bibliografia ([8]), este índice corresponde a $n = 0,5$, o que corresponde a nêutrons ultra-relativísticos, assim vemos que até aproximadamente o raio de $9,0km$ nossa estrela é composta principalmente de nêutrons ultra-relativísticos, sendo que depois há grandes mudanças em sua composição.

Ondas Gravitacionais

3.1 Perspectivas

Uma das consequências matemáticas da Relatividade Geral é a previsão das Ondas Gravitacionais, distorções na métrica que se propagariam pela estrutura do espaço-tempo à velocidade da luz e que localmente afetariam a métrica plana, alterando as dimensões dos corpos. Matematicamente são um paralelo às ondas eletromagnéticas que surgem quando um corpo dotado de carga sofre uma aceleração, só que no caso seriam corpos dotados de massa que ao serem acelerados emitiriam ondas gravitacionais mas com o diferencial de que somente momentos de quadrupolo emitiriam ondas.

3.2 Dedução Matemática

Iniciemos com a Equação de Campo de Einstein([12],[13]),

$$R_{\mu\nu} - \frac{1}{2}g_{\mu\nu}R = \frac{8\pi G}{c^4}T_{\mu\nu}, \quad (3.1)$$

que descreve a distorção na métrica gerada por uma dada concentração de energia (a massa de um corpo se relaciona com sua energia de repouso através de $E = mc^2$, assim as várias formas de energia causam alterações na métrica, mas a massa é a maior responsável, tanto que no limite

da gravitação Newtoniana a massa é a única responsável pelo campo gravitacional). Agora tomemos o limite do campo fraco, onde o potencial gravitacional deve retornar ao clássico newtoniano, em primeira ordem, e a métrica local é a métrica plana $\eta_{\mu\nu}$ acrescida de uma pequena perturbação, assim:

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu}, \quad (3.2)$$

com

$$|h_{\mu\nu}| \ll 1, \quad (3.3)$$

esta é a chamada gravidade linearizada, pois leva em conta somente termos lineares na expansão, assim para o cálculo dos termos do tensor de Ricci temos:

$$g^{\mu\nu} = \eta^{\mu\nu} - h^{\mu\nu}, \quad (3.4)$$

que quando aplicado ao tensor gera:

$$R_{\mu\nu} = -\frac{1}{2}\eta^{\sigma\lambda}h_{\mu\nu,\lambda\sigma} - \frac{1}{2}(h_{,\mu\nu} - h^\beta{}_{\nu,\mu\beta} - h^\beta{}_{\mu,\nu\beta}). \quad (3.5)$$

Aqui seguimos a convenção matemática da Relatividade Geral onde os índices gregos correm de zero a três, enquanto que os latinos correm de um a três, índices repetidos numa equação indicam soma. Há também a definição das derivadas: Dada uma função qualquer ϕ , a sua derivada em relação à uma coordenada de índice α é denotada por:

$$\frac{\partial\phi}{\partial x^\alpha} \equiv \phi_{,\alpha}; \quad (3.6)$$

por extensão, cada índice a mais é uma nova diferenciação em relação àquela coordenada. As matrizes que apresentam índices mistos, inferiores e superiores, possuem uma multiplicação implícita pela matriz métrica $\eta_{\mu\nu}$ para alterar um índice, assim, por exemplo:

$$T^\alpha{}_\beta = \eta_{\beta\mu}T^{\alpha\mu}. \quad (3.7)$$

Seguindo com as definições, definimos o Calibre de Lorentz como sendo:

$$\bar{h}^{\mu\nu}_{,\nu} = 0, \quad (3.8)$$

ele recebe este nome por ser análogo ao calibre de Lorentz do eletro-magnetismo, e $\bar{h}^{\mu\nu}$ é definido como:

$$\bar{h}^{\mu\nu} = h^{\mu\nu} - \frac{1}{2}\eta^{\mu\nu}h^\alpha_\alpha. \quad (3.9)$$

Com isto o tensor e o escalar de Ricci ficam:

$$R_{\mu\nu} = -\frac{1}{2}\eta^{\sigma\lambda}h_{\mu\nu,\sigma\lambda}, \quad (3.10)$$

$$R = -\frac{1}{2}\eta^{\sigma\lambda}h_{,\sigma\lambda}, \quad (3.11)$$

que ao serem substituídos na equação de Einstein (3.1) geram:

$$\square \bar{h}^{\mu\nu} = -\frac{8\pi G}{c^4}T^{\mu\nu}, \quad (3.12)$$

Onde \square é o operador D’Lambertiano. Esta é a chamada teoria linearizada pois leva em conta somente termos lineares na métrica, no limite Newtoniano para campos fracos, $|\phi| \ll 1$, e velocidades muito abaixo da velocidade da luz, ela retorna a métrica para o campo fraco.

3.2.1 Ondas no vácuo

Continuaremos nosso estudo com as ondas no vácuo pois são mais simples, e depois passaremos para as ondas dentro da estrela. No vácuo a equação 3.12 torna-se:

$$\square \bar{h}^{\mu\nu} = 0, \quad (3.13)$$

pois na ausência de matéria o Tensor energia-momento anula-se, abrindo o operador D’Lambertiano a equação assume a forma:

$$\left(-\frac{\partial^2}{\partial t^2} + \nabla^2\right)\bar{h}^{\mu\nu} = 0, \quad (3.14)$$

vemos que esta é a equação da onda no vácuo, e pode ser escrita na forma complexa como:

$$\bar{h}^{\mu\nu} = A^{\mu\nu} \exp(ik_\alpha x^\alpha), \quad (3.15)$$

onde k_α é o vetor de onda, este deve ser constante e ao ser decomposto em sua parte temporal e espacial apresenta a forma análoga ao vetor de onda eletromagnética:

$$\vec{k} \rightarrow (\omega, \mathbf{k}), \quad (3.16)$$

e é assim que vemos que a onda gravitacional no vácuo propaga-se na velocidade da luz.

Derivando a equação 3.15 obtemos:

$$\bar{h}_{,\alpha}^{\mu\nu} = ik_\alpha A^{\mu\nu}, \quad (3.17)$$

mas lembrando que impomos o Calibre de Lorentz (3.8), temos que:

$$k_\nu A^{\mu\nu} = 0, \quad (3.18)$$

que restringe os valores possíveis à matriz $A^{\mu\nu}$ a somente aqueles que são ortogonais à \vec{k} . Com estas imposições vemos que a onda obtida é uma onda plana, assim podemos decompô-la em uma análise de Fourier como uma superposição de ondas planas.

Como esta onda foi deduzida no Calibre de Lorentz precisamos recuperar nossa liberdade de calibre, e para isso utilizamos um novo vetor genérico ξ_α que satisfaça a equação da onda 3.14, com a escolha conveniente de ξ_α como sendo:

$$\xi_\alpha = B_\alpha \exp(ik_\mu x^\mu), \quad (3.19)$$

encontramos a seguinte relação que liga as novas coordenadas livres com as antigas definidas no Calibre de Lorentz:

$$\bar{h}_{\mu\nu}^{LIV} = \bar{h}_{\mu\nu}^{LOR} - \xi_{\mu,\nu} - \xi_{\nu,\mu} + \eta_{\mu\nu} \xi^\alpha_{,\alpha}, \quad (3.20)$$

substituindo o vetor ξ_α , e dividindo os termos exponenciais obtemos:

$$A_{\mu\nu}^{LIV} = A_{\mu\nu}^{LOR} - iB_\mu k_\nu - iB_\nu k_\mu + i\eta_{\mu\nu} B^\alpha k_\alpha. \quad (3.21)$$

Lembrando que B_α pode ser arbitrário, impomos duas condições que geram o Calibre transversal sem traço, que é muito útil em descrever ondas polarizadas que se propagam por um eixo \vec{z} definido. Estas condições são:

$$A^\alpha_\alpha = 0, \quad (3.22)$$

e

$$A_{\alpha\beta}U^\beta = 0, \quad (3.23)$$

onde U^β é uma 4-velocidade fixa. Aplicando uma transformação de Lorentz à base de coordenadas, e impondo que o vetor U^β seja somente a componente temporal, $U^\beta = \delta^\beta_0$, e orientando a onda para que viaje pelo eixo z , $k^\alpha = (\omega, 0, 0, \omega)$, encontramos que somente as componentes com $\alpha, \beta = 1, 2$ são não nulas, sendo os termos diagonais opostos em sinal e os anti-diagonais idênticos. A seguir a matriz resultante da perturbação:

$$(A_{\mu\nu}{}^{TT}) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & A_{xx} & A_{xy} & 0 \\ 0 & A_{xy} & -A_{xx} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Quando uma onda destas passar por uma partícula livre, ela alterará a métrica local e distorcerá a distância própria entre ela e uma outra partícula vizinha. Embora a localização destas partículas não seja alterada no sistema de coordenadas local, a distância entre elas é dependente da métrica e isto pode ser detectado. Por exemplo, uma partícula no eixo x numa métrica plana sofrerá uma pequena variação ε com a passagem da onda, colocando esta partícula na origem, a variação Δl que ela sofrerá em seu comprimento próprio será de, em primeira ordem:

$$\Delta l \equiv \int |ds^2|^{1/2} = \int_0^\varepsilon |g_{xx}|^{1/2} dx, \quad (3.24)$$

aproximando obtemos:

$$|g_{xx}(x=0)|^{1/2}\varepsilon \approx [1 + \frac{1}{2}h_{xx}^{TT}(x=0)]\varepsilon. \quad (3.25)$$

Embora o conceito seja simples, as estimativas para a amplitude desta variação para quando uma onda energética, do ponto de vista gravitacional, passar é da ordem de $10^{-21}m$, o que no momento esta próximo da nossa sensibilidade experimental. Devido a sua forma matricial a onda é dividida em duas polarizações:

Plus:

$$A_{xx}^{TT} \neq 0 \quad e \quad A_{xy}^{TT} = 0; \quad (3.26)$$

Cros:

$$A_{xx}^{TT} = 0 \quad e \quad A_{xy}^{TT} \neq 0. \quad (3.27)$$

Os efeitos da distorção seriam semelhantes ao da imagem 3.1, retirada de Ruthen(1992, p.74):

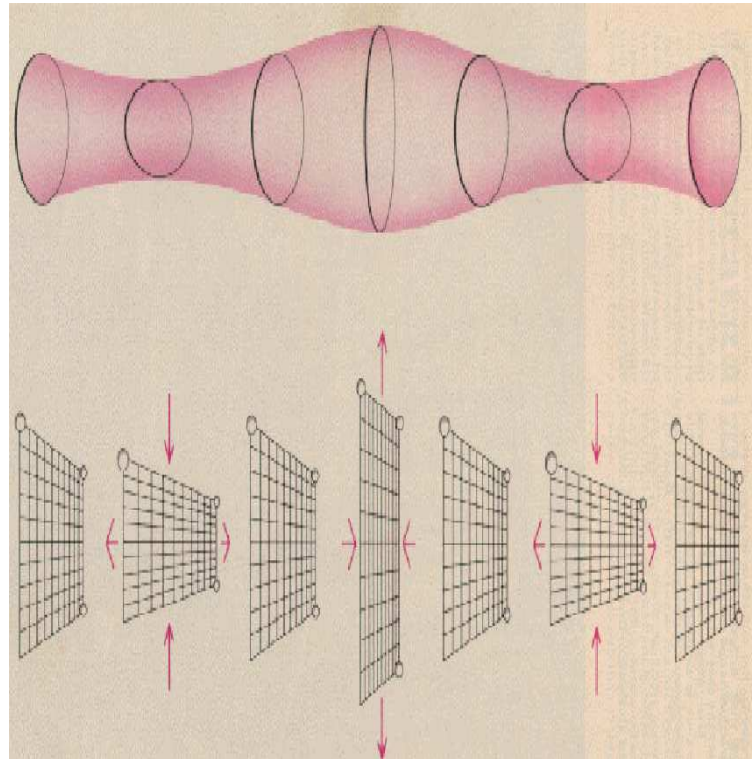


Figura 3.1: A distorção causada por uma onda gravitacional ao passar por um objeto.

3.2.2 Ondas nas Estrelas

As ondas dentro das estrelas são mais complicadas do que no vácuo pois elas interagem com a distribuição de matéria dentro da estrela, além de interagirem com o próprio fluxo de matéria, o que leva a correções de ordens superiores que complicam muito e impedem as

simples assunções levadas em conta para a onda no vácuo. Para o nosso estudo nos limitaremos a uma perturbação radial se propagando pela estrela até sua superfície, também consideraremos a estrela como ausente de campo magnético e sem influências externas. Seguiremos segundo o esquema de ([14]), ([15]) e ([16]), onde a métrica perturbada dentro da estrela é dada por:

$$ds^2 = -e^\nu (1 + \mu^l H_0 Y_m^l e^{i\omega t}) dt^2 - 2i\omega \mu^l H_1 Y_m^l e^{i\omega t} dt dr + e^\lambda (1 - \mu^l H_0 Y_m^l e^{i\omega t}) dr^2 + r^2 (1 - \mu^l K Y_m^l e^{i\omega t}) (d\theta^2 + \sin^2 \theta d\phi^2), \quad (3.28)$$

onde e^ν e e^λ são os coeficientes da métrica não perturbada, Y_m^l são os harmônicos esféricos usuais, $e^{i\omega t}$ é a dependência temporal, com ω a frequência do modo quasinormal de oscilação, l é o momento angular considerado e μ é definido como

$$\mu = \begin{cases} r/R, & \text{para } 0 \leq r < R \\ 1, & \text{para } r \geq R \end{cases}.$$

H_0, H_1 e K são funções somente de r ([15]).

Antes de continuarmos, gostaria de chamar a atenção para esta métrica, veja que ela se assemelha em forma à métrica do campo fraco, os coeficientes nos termos dt^2 e dr^2 diferem por apenas um sinal, e a simetria esférica fica evidenciada pelo mesmo coeficiente que multiplica os termos $d\theta^2$ e $d\phi^2$, mas há um termo misto $dt dr$ que evidencia a dependência desta onda em relação ao fluxo de energia que existe dentro da estrela. A onda apresenta três modos de oscilação que se acoplam à estrela:

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

p-modes(pressão): possuem frequência superior ao modo f e sua força restauradora é a pressão;

g-modes(gravidade): a força restauradora é a flutuação do campo gravitacional (empuxo), e possuem frequência inferior à do modo fundamental.

Nossa análise se restringe ao chamado *Modo Quasinormal Fundamental*, onde computamos pequenas perturbações para o modo f, com as perturbações no fluido que excitam o momento

de quadrupolo, de índice $l = 2$.

Como esta onda é dependente do fluxo de energia, é necessário definir um 4-vetor para a perturbação do fluido no interior da estrela, para isto usa-se deslocamento Lagrangiano ξ^α , dado por ([17]) e ([18]):

$$\xi^t = 0, \quad (3.29)$$

$$\xi^r = \mu^l r^{-1} e^{-\lambda/2} W Y_m^l e^{i\omega t}, \quad (3.30)$$

$$\xi^\theta = -\mu^l r^{-2} V \partial_\theta Y_m^l e^{i\omega t}, \quad (3.31)$$

$$\xi^\phi = -\nu^l (r \sin \theta)^{-2} V \partial_\phi Y_m^l e^{i\omega t}. \quad (3.32)$$

Voltando à nossa métrica (3.28), todos os seus coeficientes são dados por [19] e serão aqui reproduzidos:

e^ν e e^λ são os coeficientes da métrica não perturbada, Y_m^l são os harmônicos esféricos e $e^{i\omega t}$ é a dependência temporal. As funções H_0 , H_1 e K , que perturbam a métrica, e V e W , que perturbam o fluido, são dependentes somente da coordenada r e devem satisfazer a seguinte relação que é consequência da equação de Einstein:

$$\begin{aligned} \left[4M + \frac{1}{2}(l+2)(l-1)r + 4\pi r^3 p \right] H_0 &= 8\pi r^3 e^{-\nu/2} X - \left[\frac{1}{2}l(l+1)(M + 4\pi r^3 p) - \omega^2 r^3 e^{-(\lambda+\nu)} \right] H_1 \\ &+ \left[\frac{1}{2}(l+2)(l-1)r - \omega^2 r^3 e^{-\nu} - r^{-1} e^\lambda (M + 4\pi r^3 p)(3M - r + 4\pi r^3 p) \right] K, \end{aligned} \quad (3.33)$$

sendo

$$X = \omega^2(\rho + p)e^{-\nu/2}V - \frac{p'}{r}e^{(\nu-\lambda)/2}W + \frac{1}{2}(\rho + p)e^{\nu/2}H_0, \quad (3.34)$$

a redução de ordem do sistema. Para nosso estudo tratamos a perturbação em duas regiões, $0 \leq r \leq R/2$ e $R/2 \leq r \leq R$, sendo R o raio total da estrela.

3.2.2.1 Região $R/2 \leq r \leq R$

Aqui as perturbações formam um sistema de 4 equações diferenciais acopladas, a descrição segue como em [14]:

$$H_1' = -r^{-1}[l+1+2Me^\lambda r^{-1}+4\pi r^2 e^\lambda(p-\rho)]H_1 + r^{-1}e^\lambda[H_0+K-16\pi(p+\rho)V], \quad (3.35)$$

$$K' = r^{-1}H_0 + \frac{1}{2}(l+1)r^{-1}H_1 - [(l+1)r^{-1} - \frac{\nu'}{2}]K - 8\pi(p+\rho)e^{\lambda/2}r^{-1}W, \quad (3.36)$$

$$W' = -(l+1)r^{-1}W + re^{\lambda/2}[\gamma^{-1}p^{-1}e^{\nu/2}X - l(l+1)r^{-2}V + H_0/2 + K], \quad (3.37)$$

$$X' = -lr^{-1}X + (p+\rho)e^{\nu/2}\left\{\frac{1}{2}(r^{-1} - \nu'/2)H_0 + \frac{1}{2}[r\omega^2 e^{-\nu} + \frac{l(l+1)}{2}r^{-1}]H_1 + (\frac{3\nu'}{2} - r^{-1})\frac{K}{2} - \frac{l}{2}(l+1)\nu'r^{-2}V - r^{-1}[4\pi(p+\rho)e^{\lambda/2} + \omega^2 e^{\lambda/2-\nu} - \frac{1}{2}r^2(r^{-2}e^{-\lambda/2}\nu')']W\right\}. \quad (3.38)$$

Este sistema necessita de condições iniciais para que a integração seja possível, estas condições são os valores das funções na superfície da estrela, $r = R$, e aqui usaremos como está descrito em [15]:

Tabela 3.1: Condições iniciais

| | $H_1(R)$ | 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 |

Uma outra condição é que perturbação lagrangiana da pressão se anule na superfície da estrela, pois como acabou a estrela não há mais pressão, assim $X(R) = 0$, isto gera um conjunto de três soluções independentes que serão nomeadas como Y_1 , Y_2 e Y_3 , elas serão utilizadas na junção das duas regiões de perturbação em seu ponto comum $R/2$.

3.2.2.2 Região $0 \leq r \leq R/2$

Aqui surge um grave problema nas equações de 3.35 à 3.38, elas são singulares em $r = 0$, por isso são expandidas em séries de potência em torno deste ponto:

$$H_1(r) = H_1(0) + \frac{1}{2}r^2 H_1''(0) + \mathcal{O}(r^4), \quad (3.39)$$

$$K(r) = K_1(0) + \frac{1}{2}r^2 K_1''(0) + \mathcal{O}(r^4), \quad (3.40)$$

$$W(r) = W_1(0) + \frac{1}{2}r^2 W_1''(0) + \mathcal{O}(r^4), \quad (3.41)$$

$$X(r) = X_1(0) + \frac{1}{2}r^2 X_1''(0) + \mathcal{O}(r^4), \quad (3.42)$$

onde $H_1(0)$, $K(0)$, $W(0)$ e $X(0)$ são as condições iniciais. Os coeficientes $H_1''(0)$, $K''(0)$, $W''(0)$ e $X''(0)$ são determinados à partir de um sistema de equações dado por [14]:

$$\begin{aligned}
& -\frac{1}{4}(p_0 + \rho_0)K''(0) + \frac{1}{2}\left[p_2 + (p_0 + \rho_0)\frac{\omega^2(l+3)}{l(l+1)}e^{-\nu_0}\right]W''(0) + \frac{1}{2}e^{-\nu_0/2} = \frac{1}{4}\nu_2e^{-\nu_0/2}X(0) \\
& + \frac{1}{4}(p_2 + \rho_2)K(0) + \frac{1}{2}(p_0 + \rho_0)Q_0 + \frac{1}{2}\omega^2(p_0 + \rho_0)e^{-\nu_0}Q_1 \\
& - \left\{p_4 - \frac{4\pi}{3}\rho_0p_2 + \frac{\omega^2}{2l}[\rho_2 + p_2 - (p_0 + \rho_0)\nu_2]e^{-\nu_0}\right\}W(0), \quad (3.43)
\end{aligned}$$

$$\begin{aligned}
& \frac{1}{2}(l+2)K''(0) - \frac{1}{4}l(l+1)H_1''(0) + 4\pi(p_0 + \rho_0)W''(0) = \\
& \frac{4\pi}{3}(3p_0 + \rho_0)K(0) + \frac{1}{2}Q_0 - 4\pi\left[\rho_2 + p_2 + \frac{8\pi}{3}\rho_0(p_0 + \rho_0)\right]W(0), \quad (3.44)
\end{aligned}$$

$$\begin{aligned}
& \frac{1}{2}(l+3)H_1''(0) - K''(0) - 8\pi(p_0 + \rho_0)\frac{l+3}{l(l+1)}W''(0) = 4\pi\left[\frac{1}{3}(2l+3)\rho_0 - p_0\right]H_1(0) + \\
& + \frac{8\pi}{l}(p_2 + \rho_2)W(0) - 8\pi(p_0 + \rho_0)Q_1 + \frac{1}{2}Q_0, \quad (3.45)
\end{aligned}$$

$$\begin{aligned}
& \frac{1}{2}(l+2)X''(0) - \frac{1}{8}l(l+1)(p_0 + \rho_0)e^{\nu_0/2}H_1''(0) - \\
& (p_0 + \rho_0)e^{\nu_0/2}\left[\frac{1}{4}(l+2)\nu_2 - 2\pi(p_0 + \rho_0) - \frac{1}{2}\omega^2e^{\nu_0}\right] = \frac{1}{2}\left[\rho_2 + p_2 + \frac{1}{2}(\rho_0 + p_0)\nu_2\right]\frac{lX(0)}{\rho_0 + p_0} + \\
& + (p_0 + \rho_0)e^{\nu_0/2}\left\{\frac{1}{2}\nu_2K(0) + \frac{1}{4}Q_0 + \frac{1}{2}\omega^2e^{-\nu_0}H_1(0) - \frac{1}{4}l(l+1)\nu_2Q_1 + \left[\frac{1}{2}(l+1)\nu_4 - 2\pi(p_2 + \rho_2)\right.\right. \\
& \left.\left. - \frac{16\pi^2}{3}\rho_0(p_0 + \rho_0) + \frac{1}{2}\left(\nu_4 - \frac{4\pi}{3}\rho_0\nu_2\right) + \frac{1}{2}\omega^2e^{-\nu_0}\left(\nu_2 - \frac{8\pi}{3}\rho_0\right)\right]W(0)\right\} \quad (3.46)
\end{aligned}$$

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

$$\begin{aligned}
Q_0 = \frac{4}{(l+2)(l-1)} & \left\{ 8\pi e^{-\nu_0/2}X(0) - \left(\frac{8\pi}{3}\rho_0 + \omega^2e^{-\nu_0}\right)K(0) - \right. \\
& \left. \left[\frac{2\pi}{3}l(l+1)(\rho_0 + 3p_0) - \omega^2e^{-\nu_0}\right]H_1(0) \right\}, \quad (3.47)
\end{aligned}$$

$$Q_1 = \frac{2}{l(l+1)} \left[\frac{X(0)}{\gamma_0 p_0} e^{-\nu_0/2} + \frac{3}{2}K(0) + \frac{4\pi}{3}(l+1)\rho_0W(0) \right]. \quad (3.48)$$

Os termos p_2 , p_4 , ν_2 , ν_4 e ρ_2 são dados por [15], e estão listados abaixo:

$$p_2 = -\frac{4\pi}{3}(\rho_0 + p_0)(\rho_0 + 3p_0), \quad (3.49)$$

$$\rho_2 = p_2 \frac{\rho_0 + p_0}{\gamma_0 p_0}, \quad (3.50)$$

$$\nu_2 = \frac{8\pi}{3}(\rho_0 + 3p_0), \quad (3.51)$$

$$p_4 = \frac{2\pi}{5}(\rho_0 + p_0)(\rho_2 + 5p_2) - \frac{2\pi}{3}(\rho_2 + p_2)(\rho_0 + 3p_0) - \frac{32\pi^2}{9}\rho_0(\rho_0 + p_0)(\rho_0 + 3p_0), \quad (3.52)$$

$$\nu_4 = \frac{4\pi}{5}(\rho_2 + 5p_2) + \frac{64\pi^2}{9}\rho_0(\rho_0 + 3p_0). \quad (3.53)$$

Todo o conjunto de equações (3.43) até (3.46) pode ser escrito na forma matricial, sendo que no lado esquerdo da igualdade teremos uma matriz quadrada de ordem 4 multiplicando um vetor coluna ($Y''(0)$) formado por $H_1''(0)$, $K''(0)$, $W''(0)$ e $X''(0)$. No lado direito teremos uma outra matriz quadrada de ordem 4 multiplicando um vetor coluna ($Y(0)$) formado por $H_1(0)$, $K(0)$, $W(0)$ e $X(0)$. Assim:

$$AY''(0) = BY(0), \quad (3.54)$$

isolando o vetor $Y''(0)$ obtemos a solução:

$$Y''(0) = A^{-1}BY(0), \quad (3.55)$$

temos o vetor $Y(0)$ que é dado 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 do conjunto de equações (3.43) até (3.46). Mas surge uma pequena ambivalência nas soluções, segundo nossa referência [14] os valores das condições iniciais são:

$$H_1(0) = \frac{2lK(0) + 10\pi(\rho_0 + p_0)W(0)}{l(l+1)}, \quad (3.56)$$

$$K(0) = \pm(\rho_0 + p_0), \quad (3.57)$$

$$W(0) = 1, \quad (3.58)$$

$$X(0) = (\rho_0 + p_0)e^{\nu_0/2} \left(\left[\frac{4\pi}{3}(\rho_0 + 3p_0) - \frac{\omega^2 e^{\nu_0}}{l} \right] W(0) + \frac{1}{2}K(0) \right), \quad (3.59)$$

assim vemos que $K(0)$ pode assumir dois valores, o que gera duas soluções distintas, e estas serão nomeadas como Y_4 e Y_5 .

3.2.2.3 Perturbação Total

Agora que temos as duas soluções que vão do centro até $R/2$, e as três que vão de $R/2$ até a superfície, precisamos juntá-las para obter uma única descrição das perturbações na estrela, para isso impomos que a solução deve ser contínua em $r = R/2$, assim temos o conjunto de equações:

$$Y(r) = a_1 Y_1(r) + a_2 Y_2(r) + a_3 Y_3(r) \quad R/2 \leq r \leq R, \quad (3.60)$$

$$Y(r) = a_4 Y_4(r) + a_5 Y_5(r) \quad 0 \leq r \leq R/2, \quad (3.61)$$

$$a_1 Y_1(R/2) + a_2 Y_2(R/2) + a_3 Y_3(R/2) = a_4 Y_4(R/2) + a_5 Y_5(R/2), \quad (3.62)$$

para resolver este sistema precisamos determinar todas as constantes a 's, para isso voltamos à escrever Y como sendo uma matriz coluna de quatro linhas, com as funções dadas H_1, K, W, X , assim teremos quatro equações e cinco incógnitas:

$$a_1 H_1(R/2) + a_2 H_{12}(R/2) + a_3 H_{13}(R/2) = a_4 H_{14}(R/2) + a_5 H_{15}(R/2), \quad (3.63)$$

$$a_1 K_1(R/2) + a_2 K_2(R/2) + a_3 K_3(R/2) = a_4 K_4(R/2) + a_5 K_5(R/2), \quad (3.64)$$

$$a_1 W_1(R/2) + a_2 W_2(R/2) + a_3 W_3(R/2) = a_4 W_4(R/2) + a_5 W_5(R/2), \quad (3.65)$$

$$a_1 X_1(R/2) + a_2 X_2(R/2) + a_3 X_3(R/2) = a_4 X_4(R/2) + a_5 X_5(R/2). \quad (3.66)$$

Para que este sistema tenha solução exata tomamos $a_5 = 1$, isso pode ser feito pois a amplitude total é arbitrária. Escrevendo apropriadamente este sistema de equações na forma matricial obtemos:

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

As constantes são calculadas invertendo a matriz quadrada e multiplicando-a pela coluna da direita, assim com as constantes calculadas as perturbações ficam dadas por:

Para a região $R/2 \leq r \leq R$:

$$H_1(r) = a_1 H_{11}(r) + a_2 H_{12}(r) + a_3 H_{13}(r), \quad (3.67)$$

$$K(r) = a_1 K_1(r) + a_2 K_2(r) + a_3 K_3(r), \quad (3.68)$$

$$W(r) = a_1 W_1(r) + a_2 W_2(r) + a_3 W_3(r), \quad (3.69)$$

$$X(r) = a_1 X_1(r) + a_2 X_2(r) + a_3 X_3(r). \quad (3.70)$$

E para a região $0 \leq r \leq R/2$:

$$H_1(r) = a_4 H_{14}(r) + H_{15}(r), \quad (3.71)$$

$$K(r) = a_4 K_4(r) + K_5(r), \quad (3.72)$$

$$W(r) = a_4 W_4(r) + W_5(r), \quad (3.73)$$

$$X(r) = a_4 X_4(r) + X_5(r). \quad (3.74)$$

Assim com todas as equações e constantes em mãos as perturbações podem ser computadas em seus respectivos regimes.

3.2.3 As perturbações fora da estrela e as ondas emitidas

3.2.3.1 A equação de Zerilli

Com a descrição das perturbações na estrela e a descrição das ondas no vácuo agora é necessário fazer a descrição das ondas perto da fonte emissora, a região intermediária entre o observador e a fonte, e depois unir esta região à fonte e tomar o limite para infinito (propagação no vácuo). Nessa região trabalharemos com a equação de Zerilli, dada por [15]:

$$\frac{d^2 Z}{dr^{*2}} = [V_Z(r^*) - \omega^2]Z, \quad (3.75)$$

onde r^* é a coordenada tartaruga:

$$r^* = r + 2GM \ln \left| \frac{r}{2GM} - 1 \right|; \quad (3.76)$$

que é usada para corrigir a singularidade geométrica que ocorre em $r = 2MG$ na métrica de Schwarzschild.

$V_Z(r^*)$ é o potencial efetivo e é definido como:

$$V_Z(r^*) = \frac{1 - 2M/r}{r^3(nr + 3M)^2} [2n^2(n + 1)r^3 + 6n^2Mr^2 + 18nM^2r + 18M^3], \quad (3.77)$$

onde:

$$n = \frac{1}{2}(l - 1)(l + 2). \quad (3.78)$$

Mas precisamos da equação de Zerilli e sua derivada na superfície da estrela, e para isso usaremos as perturbações $H_0(R)$ e $K(R)$ que foram descritas anteriormente, assim ligamos a região perto da fonte ao interior da estrela. Seguindo com o esquema de [15] definimos o sistema matricial abaixo:

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

Onde os termos são:

$$a(r) = \frac{nr + 3M}{\omega^2 r^2 - (n + 1)M/r}, \quad (3.79)$$

$$b(r) = \frac{nr - \omega^2 r^4 + M(r - 3M)}{(r - 2M)[\omega^2 r^2 - (n + 1)M/r]}, \quad (3.80)$$

$$g(r) = \frac{n(n + 1)r^2 + 3nMr + 6M^2}{r^2(nr + 3M)}, \quad (3.81)$$

$$h(r) = \frac{-nr^2 + 3nMr + 3M^2}{(r - 2M)(nr + 3M)}, \quad (3.82)$$

$$k(r) = \frac{-r^2}{r - 2M}. \quad (3.83)$$

A partir deste sistema de equações consegue-se a equação de Zerilli isolando-se o vetor do lado esquerdo da igualdade, como a equação de Zerilli é transcendental usa-se métodos numéricos para isso, no nosso estudo ela foi calculada desde $r = R$ até $r = 25\omega^{-1}$ ([16]).

3.2.3.2 A Frequência do Modo Quasinormal Fundamental

Agora calcularemos a frequência do modo quasinormal fundamental, que foi comentado no começo de toda esta discussão sobre as ondas na estrela, usaremos uma expansão em séries de Fourier para a equação de Zerilli, representando uma onda vindo do infinito e outra indo para o infinito:

$$Z_-(r*) = e^{-i\omega r*} \sum_{j=0}^{\infty} \alpha_j r^{-j}, \quad (3.84)$$

$$Z_+(r*) = e^{i\omega r*} \sum_{j=0}^{\infty} \bar{\alpha}_j r^{-j}. \quad (3.85)$$

Onde Z_- são as ondas vindo do infinito e Z_+ são as ondas indo para o infinito. Como dito no começo de tudo, estamos interessado somente no momento de quadrupolo, por isso a expansão irá até o segundo termo, com os coeficientes:

$$\alpha_1 = -i(n+1)\omega^{-1}\alpha_0, \quad (3.86)$$

$$\alpha_2 = -\frac{1}{2}\omega^{-2} \left[n(n+1) - \frac{3}{2}iM\omega(1+2/n) \right] \alpha_0. \quad (3.87)$$

Usando a fórmula de Euler as duas expansões podem ser escritas como:

$$Z_-(r*) = \alpha_0 \cos(\omega r*)(Z_1 + iZ_2) + \alpha_0 \sin(\omega r*)(Z_2 - iZ_1), \quad (3.88)$$

$$Z_+(r*) = \bar{\alpha}_0 \cos(\omega r*)(Z_1 - iZ_2) + \bar{\alpha}_0 \sin(\omega r*)(Z_2 + iZ_1). \quad (3.89)$$

Onde os coeficientes são:

$$Z_1 = 1 - \frac{n(n+1)}{2\omega^2 r^2}, \quad (3.90)$$

$$Z_2 = \frac{3M}{4\omega r^2} (1 + 2/n) - \frac{n+1}{\omega r}, \quad (3.91)$$

assim a equação de Zerilli completa pode ser escrita como uma combinação linear de suas duas expansões:

$$Z(r*) = \beta(\omega)Z_-(r*) + \gamma(\omega)Z_+(r*). \quad (3.92)$$

Precisamos descobrir as constantes β e γ , para isso usamos a derivada da equação de Zerilli e as derivadas de suas expansões:

$$\frac{dZ(r*)}{dr*} = \beta(\omega) \frac{dZ_-}{dr*}(r*) + \gamma(\omega) \frac{dZ_+}{dr*}(r*), \quad (3.93)$$

$$\frac{dZ_-}{dr*}(r*) = \alpha_0 \cos(\omega r*)(Z_4 + iZ_3) + \alpha_0 \sin(\omega r*)(Z_3 - iZ_4), \quad (3.94)$$

$$\frac{dZ_+}{dr*}(r*) = \bar{\alpha}_0 \cos(\omega r*)(Z_4 - iZ_3) + \bar{\alpha}_0 \sin(\omega r*)(Z_3 + iZ_4). \quad (3.95)$$

sendo:

$$Z_3 = \frac{n(n+1)}{2\omega r^2} - \left[\frac{3M(1+2/n) - 2r(n+1)}{2\omega r^3} \right] \left(1 + \frac{2M}{r-2M} \right)^{-1} - \omega, \quad (3.96)$$

$$Z_4 = \frac{n(n+1)}{\omega^2 r^3} \left(1 + \frac{2M}{r-2M} \right)^{-1} + \frac{3M(1+2/n)}{4r^2} - \frac{n+1}{r}. \quad (3.97)$$

Mais uma vez todo esse sistema pode ser escrito na forma matricial:

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

Isolando e resolvendo o vetor do lado direito da igualdade, no intervalo em que Z e sua derivada estão sujeitos, que foi dito anteriormente, encontramos as constantes procuradas. Como precisamos de um modo fundamental de oscilação como parâmetro de entrada, escolhemos a frequência Newtoniana:

$$\omega_N = \sqrt{\frac{2l(l-1)}{2l+1} \left(\frac{M}{R^3} \right)}, \quad (3.98)$$

e com este definimos três valores para a frequência ω :

$$\omega_A = 0, 9\omega_N, \quad (3.99)$$

$$\omega_B = \omega_N, \quad (3.100)$$

$$\omega_C = 1, 1\omega_N, \quad (3.101)$$

e com estes calculamos os coeficientes γ para cada ω . Os modos quasinormais correspondem as ondas que somente vem do infinito, ou seja, $\gamma = 0$, assim com estes três valores de γ

ajustamos um polinômio do segundo grau e resolvendo-o encontramos as raízes complexas ω_+ e ω_- . Escolhemos uma destas raízes para ser a próxima estimativa, pegamos sua parte real e substituímos em um dos três ômega ditos anteriormente e repetimos o procedimento até que a frequência seja determinada com a precisão necessária. A frequência complexa ω é escrita como:

$$\omega = 2\pi f + i/\tau, \quad (3.102)$$

onde f é a frequência de oscilação da onda gravitacional emitida e τ é o seu tempo de decaimento([16]).

3.3 Detecção

Depois de toda a exaustiva descrição matemática das ondas gravitacionais, e da descrição dos métodos computacionais por nós usados, nos voltaremos às técnicas de detecção destas ondas, pois tão importante quanto descrevê-las é detectá-las, pois somente a observação experimental é que pode validar uma teoria ou refutá-la. Embora ainda não tenhamos evidências diretas da detecção das ondas gravitacionais, possuímos evidências indiretas que concordam grandemente com a teoria e que motivam a criação de detectores ainda mais precisos.

3.3.1 Observação Indireta

A observação indireta se dá através de previsões que a relatividade geral tem sobre corpos que estariam supostamente emitindo ondas gravitacionais, como estas ondas carregam parte da energia do corpo para o espaço, este corpo teria seus parâmetros variando com o tempo, por exemplo, uma estrela à alta rotação que estaria emitindo ondas gravitacionais também estaria diminuindo sua frequência de rotação, assim um observador que medisse a taxa de desaceleração poderia medir a quantidade de energia emitida por ondas gravitacionais. Um outro exemplo são os pulsares binários, como o sistema PSR1913+16 que será descrito à seguir.

3.3.1.1 O sistema PSR1913+16

Este sistema foi descoberto em 1975 por Allan Russell e Joseph Hooton Taylor Jr., ambos da Universidade de Princeton, usando o rádio-telescópio de Arecibo de 305m medindo as pulsações de rádio vindo deste sistema. Ele consiste de duas estrelas de nêutrons de massa aproximadamente $1,4M_{\odot}$ orbitando o centro de massa do sistema numa elipse acentuada, seguindo as Leis de Kepler. A distância entre os dois varia entre $1,1R_{\odot}$ no periastro e $4,8R_{\odot}$ no apoastro, resultando num período orbital de 7,75 horas. Uma das estrelas é altamente magnetizada e emite pulsos de rádio com período de 59ms. Uma representação esquemática é dada pela figura 3.2 abaixo.

Segundo as predições da relatividade geral este sistema emitirá boa parte de sua energia potencial gravitacional na forma de ondas gravitacionais conforme as estrelas orbitam uma à outra, e assim suas distâncias relativas diminuirão. Como consequência o período orbital também diminuirá. Isso foi observado em 1983 quando Taylor e seus colaboradores mediram que a hora do periastro havia mudado, confirmando que a órbita não se manteve estável. As observações continuaram por mais trinta anos e os resultados de medida de decaimento da órbita são resumidos na figura 3.3 retirada de [20] .

Os pontos correspondem às observações e a curva é a predição da relatividade geral:

$$\dot{P}_{o,GR} = -\frac{192\pi G^{5/3}}{5c^5} \left(\frac{P_o}{2\pi}\right)^{-5/3} (1-e^2)^{-7/2} \left(1 + \frac{73}{24}e^2 + \frac{37}{96}e^4\right) m_p m_c (m_p + m_c)^{-1/3}. \quad (3.103)$$

Sendo e a excentricidade da órbita, P_o o período orbital, m_p e m_c a massa da estrela pulsar e sua companheira. Vejam que as medidas coincidem perfeitamente com a teoria, sendo um forte indicio para as ondas gravitacionais. Se o sistema continuar a perder energia a essa taxa prevê-se que daqui 300 milhões de anos as duas estrelas se choquem num evento extremamente energético, tanto em ondas eletromagnéticas, gravitacionais e emissão de neutrinos. Se nesse futuro distante ainda houver humanidade, ciência e tecnologia; isto será um evento importante para a ciência da época.

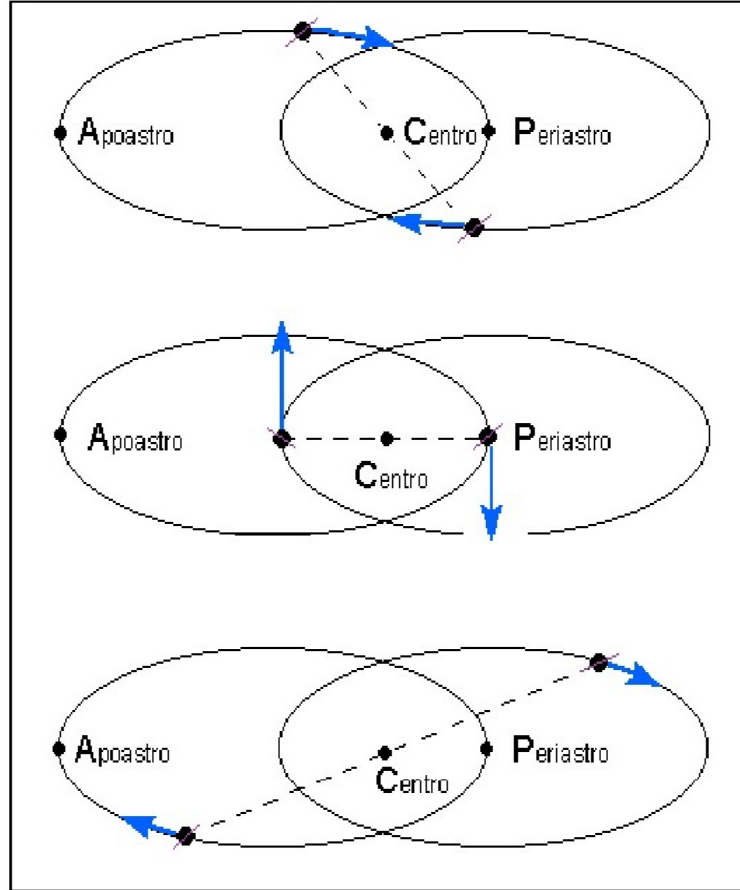


Figura 3.2: Representação do sistema PSR1913+16.

3.3.2 Observação Direta

Os métodos de observação direta correspondem a interferômetros que detectam a variação da distância entre dois pontos, como descrito pela equação 3.25.

3.3.2.1 Interferometria à laser

Funcionam no princípio do Interferômetro de Michelson, dois lasers perpendiculares cruzam seus feixes e qualquer variação na distância percorrida pela luz é vista como um padrão de interferência. Embora o conceito seja simples a alta sensibilidade requerida ($10^{-21} Hz^{-1/2}$) faz

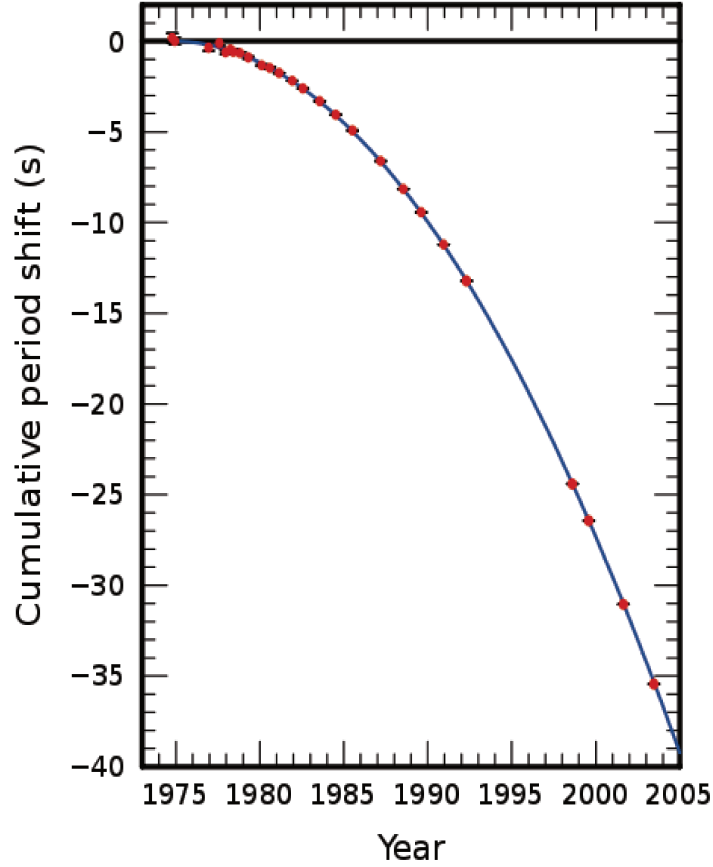


Figura 3.3: Mudança de fase no periastro.

com que os braços do laser sejam da ordem de quilômetros, o que necessita de um alinhamento ultra-preciso, além do isolamento de qualquer forma de perturbação mecânica externa. O melhor detector deste tipo é o LIGO (Laser Interferometer Gravitational Wave Observatory), localizado nos EUA com três interferômetros dois em Hanford no estado de Washington com braços de 4 e 2Km e um terceiro em Livingston na Louisiana com braços de 4Km, sua sensibilidade é de $10^{-21} Hz^{-1/2}$.

Uma nova abordagem que visa resolver os problemas da limitação do comprimento do feixe do laser e as perturbações mecânicas é a construção dos detectores no espaço sideral, onde não

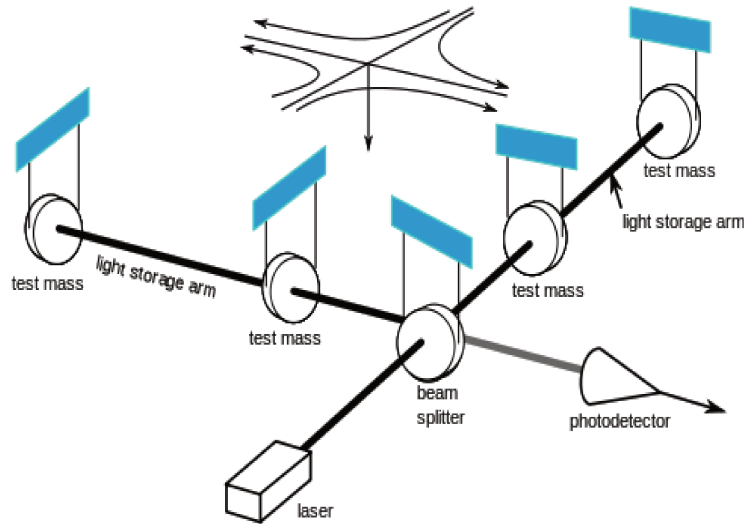


Figura 3.4: Um detector pelo princípio do interferômetro de Michelson.

existe interferência de corpos externos, pode-se criar os mais diversos arranjos experimentais por centenas de milhares de quilômetros sem limitação de espaço físico. Um projeto recente que vem mobilizando a comunidade internacional é o LISA (Laser Interferometer Space Antenna), três satélites dispostos num triângulo equilátero com lado de comprimento $5 \times 10^5 km$ numa órbita próxima à da Terra. Este audacioso projeto prevê a resolução necessária para medir ondas vindas de buracos negros super massivos, pulsares binários e até mesmo o proposto fundo em ondas gravitacionais vindo do Big Bang.

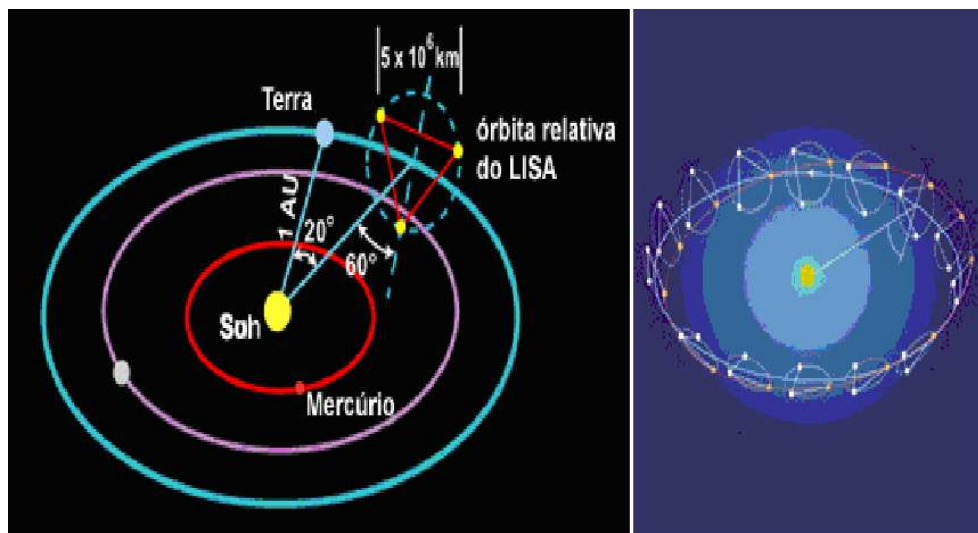


Figura 3.5: Esquema do projeto LISA.

3.3.2.2 Detectores de Massas Ressonantes

Consistem em massas simétricas de metal, cilíndricas ou esféricas, onde o efeito acumulativo da onda seria medido como uma distorção em toda a estrutura cristalina do metal, o trabalho pioneiro foi de Joseph Weber que construiu na década de 60 um cilindro de liga de alumínio com massa da ordem de uma tonelada. Embora o cilindro tenha sido colocado em uma câmara de vácuo e isolado de todos os ruídos externos sua sensibilidade não ultrapassou 10^{-16} pois mesmo resfriado a temperaturas criogênicas o movimento térmico caótico dos átomos de que era constituído ultrapassava a sensibilidade do aparelho([21]).



Figura 3.6: Joseph Weber e seu detector.

Aqui gostaria de focar no detector Mario Schenberg, que faz parte do projeto Gráviton, um projeto brasileiro feito em parceria com a Universidade Estadual de São Paulo (USP) e o Instituto Nacional de Pesquisas Espaciais (INPE), junto com o apoio de ITA, UNICAMP e CEFET-SP([22]). Este detector constitui de uma esfera de aproximadamente uma tonelada feita de uma liga de 94% de cobre e 6% de alumínio, resfriada até aproximadamente a temperatura do hélio líquido e suspensa para impedir a interferência de vibrações externas. O projeto começou no ano de 2000 e possibilitou o intercâmbio científico entre as instituições que colaboraram com suas áreas de atuação tecnológica. Além do isolamento mecânico três detectores de raios cósmicos filtram perturbações vindas de chuviscos atmosféricos de partículas carregadas, mesmo

que a energia carregada por estas seja mínima, ela é suficiente para perturbar o detector quando ele estiver em pleno funcionamento. Devido ao seu formato o detector é capaz de detectar ondas vindas de qualquer direção sem perda de eficiência e ainda com a capacidade de indicar de onde vem o sinal. Este sinal é captado pelos seis detectores que sentem as pequenas variações no diâmetro da esfera e transformam em impulsos elétricos que são analisados pelo computador. Um esquema é mostrado na figura 3.7 à seguir. A figura 3.8 mostra a esfera detectora.

A banda de frequência do detector é de $(3200 \pm 100)\text{Hz}$ ([23]).

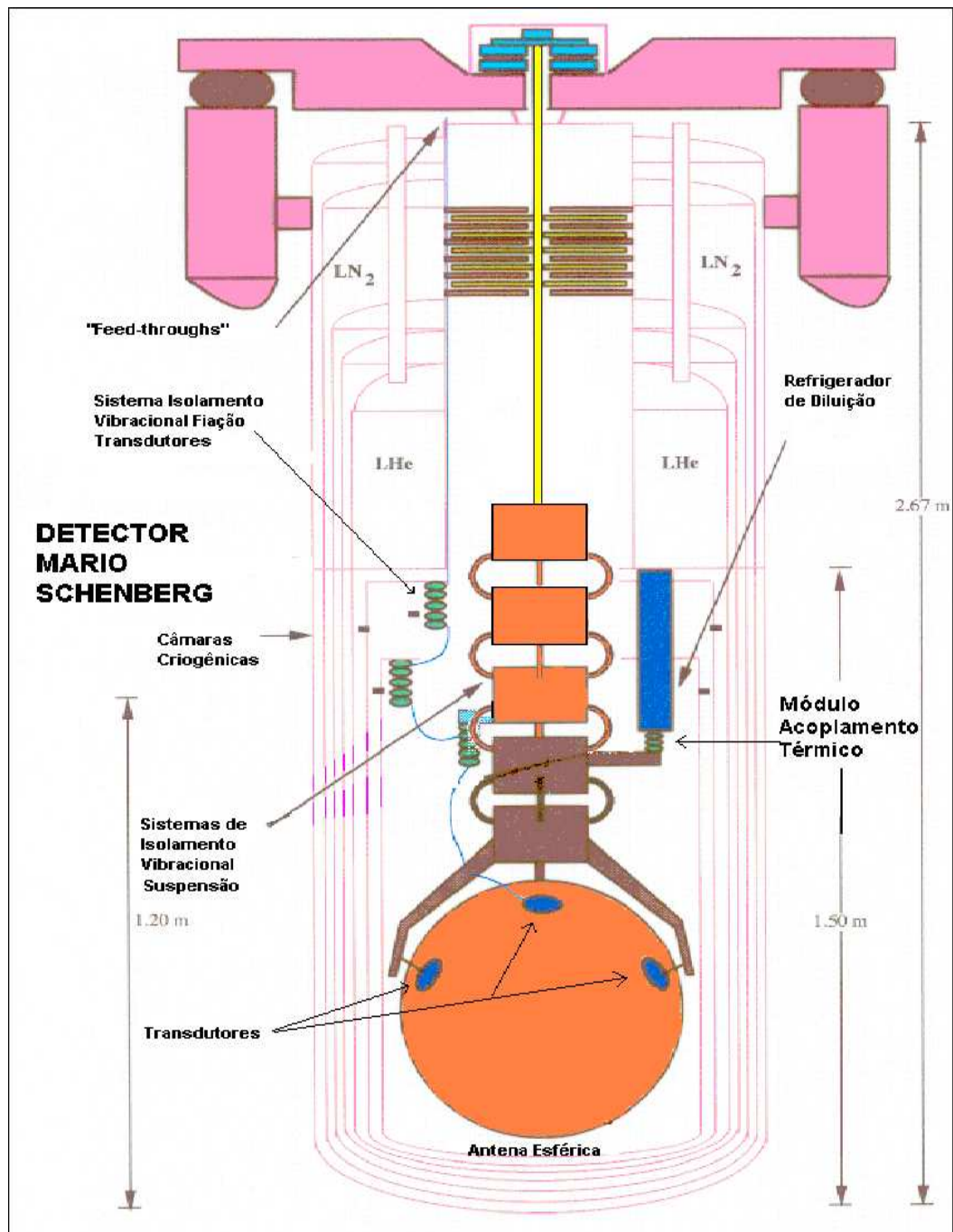


Figura 3.7: O detector Mário Schenberg.

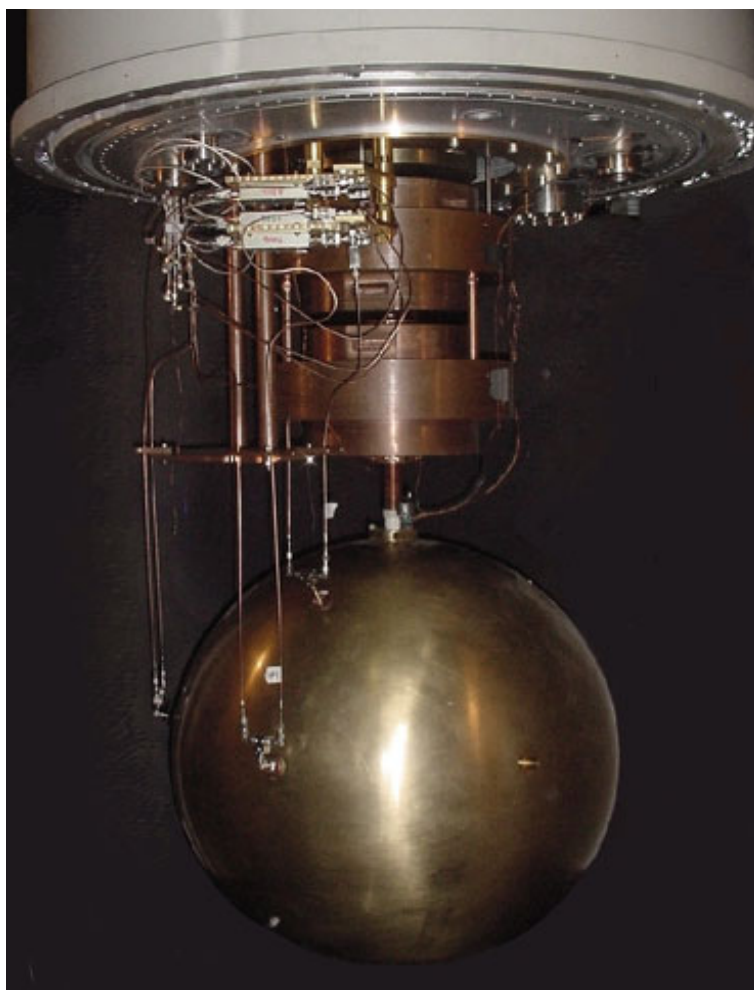


Figura 3.8: A esfera detectora.

Metodologia

Aqui discutiremos o método por nós utilizado para o cálculo numérico de todas as equações do capítulo anterior, seguiremos como está descrito em [16].

4.1 Dentro da Estrela

Para calcular as ondas nosso programa divide a estrela em dez mil pontos pelo seu interior, sendo cinco mil pontos na região $0 \leq r \leq R/2$, e cinco mil na região $R/2 \leq r \leq R$. Para isso nosso programa inicia-se com a rotina mestre `zerilli4`, onde são definidas todas as variáveis utilizadas e onde todos os cálculos são arranjados para fornecer o resultado final. Este programa depende de todas as variáveis internas da estrela para um dado r , por isso ele chama a rotina `tov_rot_z` para que esta simule a estrela. Nesta rotina estão as expansões numéricas para a resolução das equações TOV dadas no capítulo 2 (2.22, 2.23 e 2.24). Esta rotina chama as rotinas `gamaSLy1` e `gamaSLy2` que são nossa equação de estado SLy, sendo que na primeira ela fornece a pressão em função da densidade e na outra o contrário, densidade em função da pressão. Para o cálculo dos pontos intermediários aos tabelados seguimos a interpolação descrita na seção 2.4.2.

A rotina `tov_rot_z` chama a rotina `rk4sys`, que possui nosso método de Runge-Kutta de

quarta ordem, para resolver numericamente as equações TOV; esta por sua vez chama a rotina **xpsys** que contém as equações TOV, aqui está também nossa condição de que as equações devem ser calculadas até que tenham encontrado a superfície. Por fim esta chama a rotina **gamaSLy2** que possui nossa equação de estado. Com a estrela em mãos o programa está habilitado à calcular as perturbações e a equação de Zerilli. A equação de Zerilli (3.75) e sua derivada são calculadas na rotina **zerilli_rot**, as funções $\beta(\omega)$ e $\gamma(\omega)$ (3.92 e 3.93), e a condição de que $\gamma(\omega)$ deve se anular para ω são calculadas na rotina **zerilli2_rot**. A rotina **zerilli_rot** possui como auxiliares as rotinas **rk4sys_fora** e **xpsys_fora**, que também são os métodos de integração de quarta ordem.

Para que as rotinas calculem a equação de Zerilli é necessário que tenham todas as constantes de integração dadas na dedução da parte teórica, todo o conjunto de equações 3.63, 3.64, 3.65 e 3.66, para isso usa-se a rotina **constantes_rot**, onde todas as perturbações da estrela são unidas para o ponto $r = R/2$ e a perturbação total é calculada, através das equações 3.67, 3.68, 3.69, 3.70, 3.71, 3.72, 3.73 e 3.74. Esta rotina é dependente das rotinas **s_m_rot**, que calcula as cinco mil perturbações entre $R/2 \leq r \leq R$, e **c_m_rot** que calcula as cinco mil perturbações entre $0 \leq r \leq R/2$.

A rotina **s_m_rot** calcula as equações 3.35, 3.36, 3.37 e 3.38, e é dependente das rotinas **tov_rot_z4**, que simula a estrela para a região consideradas, e das rotinas **rk4sys_p1** e **xpsys_p1** que são o método numérico utilizado. As condições iniciais para a resolução do sistema numérico são aquelas dadas na tabela 3.1, lembrando que esta rotina faz o cálculo para Y_1 , Y_2 e Y_3 em separado.

A rotina **c_m_rot** apresenta a singularidade na origem e para corrigir este problema ela chama a rotina **P_C**, que resolve as equações 3.39, 3.40, 3.41 e 3.42 na origem, sendo que para isso utiliza-se das rotinas **matriz**, que resolve a equação 3.55, e **inversa**, que inverte a matriz A . A estrela para esta região é calculada pela rotina **tov_rot_z4**. O método numérico é novamente aquele utilizado por **rk4sys_p1** e **xpsys_p1**.

4.2 Fora da Estrela

Após todas as perturbações interiores à estrela terem sido calculadas, retornamos à rotina `zerilli_rot`, que agora calcula a equação de Zerilli (3.75) e sua derivada, sendo que agora o termo potencial $V_Z(r^*)$ é a estrela perturbada, que dará origem as ondas. A rotina resolve a equação de $r = R$ até $r = 25\omega^{-1}$, sendo que após este ponto a frequência mantém-se constante. Veja que agora a equação tem a forma de uma equação de onda com a fonte emissora. A rotina `zerilli2_rot` calcula os valores de $\beta(\omega)$ e $\gamma(\omega)$ (3.92 e 3.93) antes de devolvê-los à rotina anterior. Estes resultados são devolvidos à rotina `zerilli4` que então calcula a equação do segundo grau que dará nosso valor para a frequência complexa ω presente na equação de Zerilli, sendo que a parte real corresponde à frequência da onda emitida e a parte imaginária ao tempo de decaimento. Para isso usam-se os três valores iniciais 3.99, 3.100 e 3.101. Usando estes valores finalmente encontra-se o ω dado na equação 3.102 para assim determinar f e τ .

Abaixo temos dois fluxogramas que indicam a interligação das rotinas utilizadas nos cálculos.

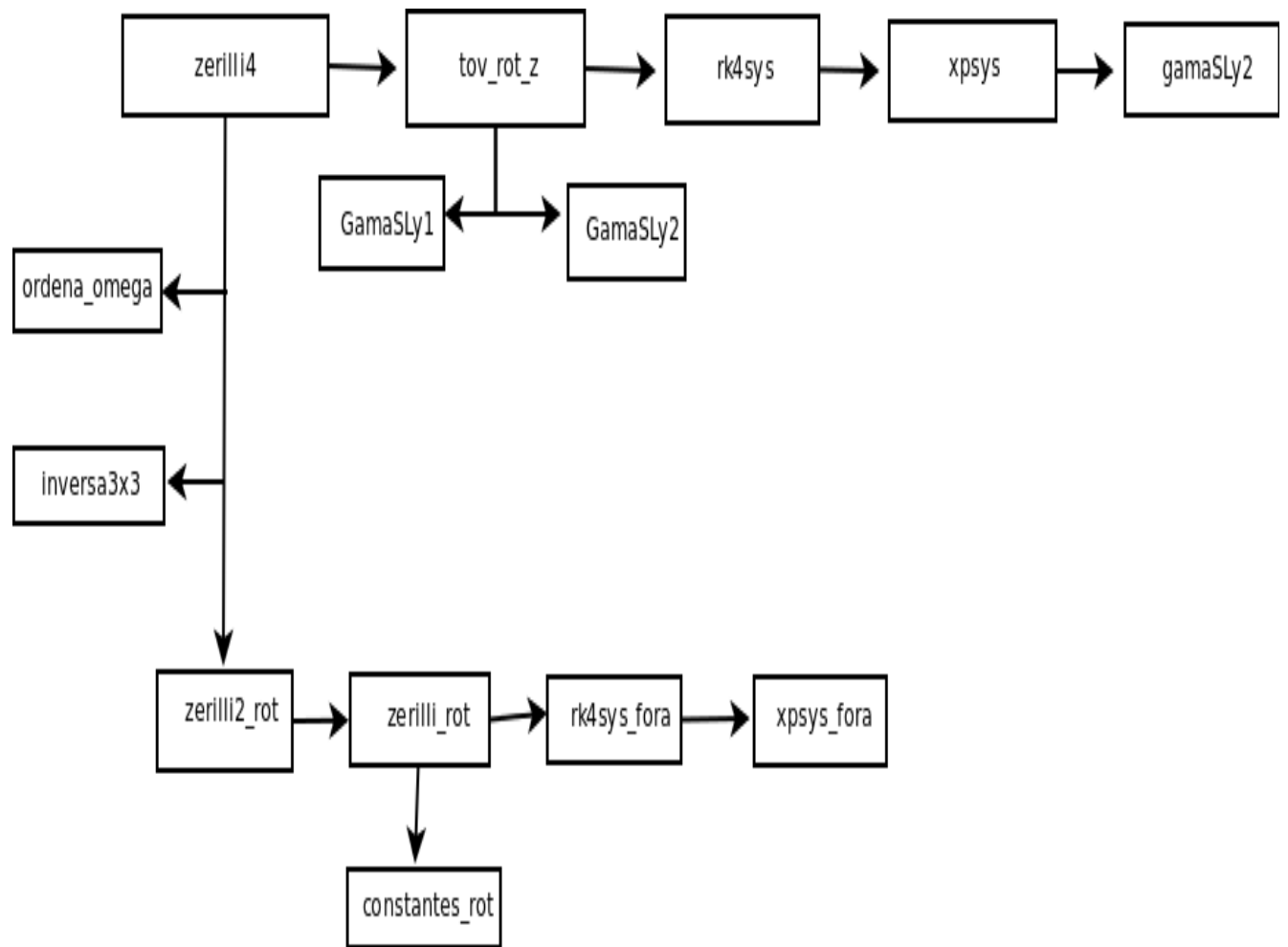


Figura 4.1: A sequência para as rotinas.

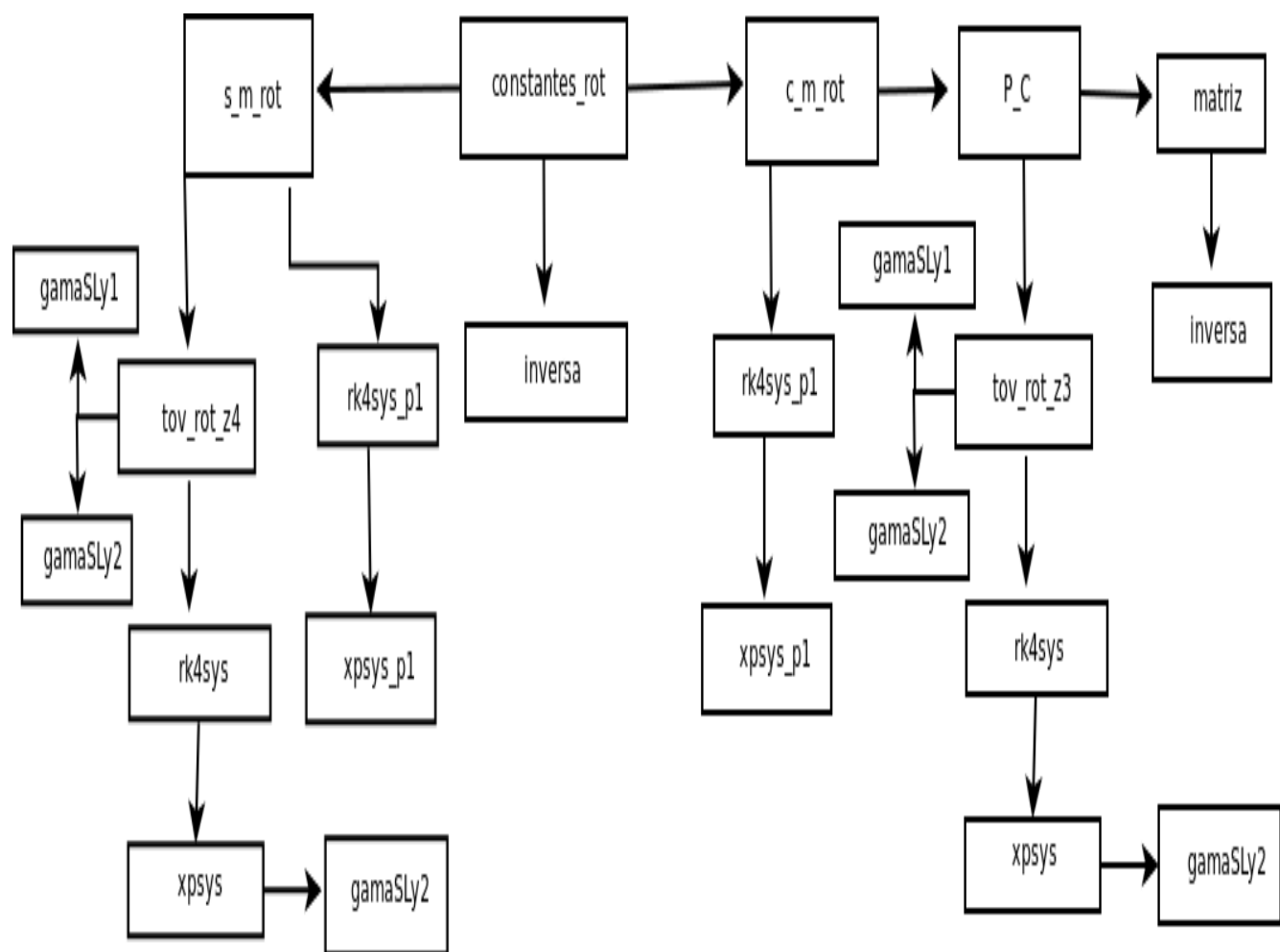


Figura 4.2: A sequência para as rotinas (continuação).

Resultados das simulações

Agora compararemos as simulações para as equações politrópica e SLy e discutiremos nossos resultados.

5.1 Diferenças entre as estrelas simuladas

Na seção 2.5 discutimos um caso específico para mostrar as diferenças na estrutura interna causada pela escolha de uma ou outra equação de estado, mas na prática não podemos medir a estrutura de uma estrela devido às nossas limitações astronômicas e tecnológicas (não há como lançar uma sonda para explorar uma estrela de nêutrons cujo tempo de espera não seja de bilhões de anos e mesmo assim não dispomos de um material capaz de suportar uma temperatura de milhões de Kelvins). Por isso nos interessamos mais pela massa total e pelo raio da estrela que são grandezas que podem ser facilmente medidas pelos nossos telescópios. A massa pode ser deduzida a partir da órbita de estrelas binárias e o raio pode ser medido durante uma ocultação. Nossos resultados predizem que a massa, o raio e a densidade central possuem uma relação unívoca, sendo que com a medição de um pode-se deduzir o outro. Isso fica claro pelas figuras 5.1 e 5.2 logo a seguir.

Para a geração destes dados pegamos 200 pontos entre as densidades de $1,6 \times 10^{14} g/cm^3$

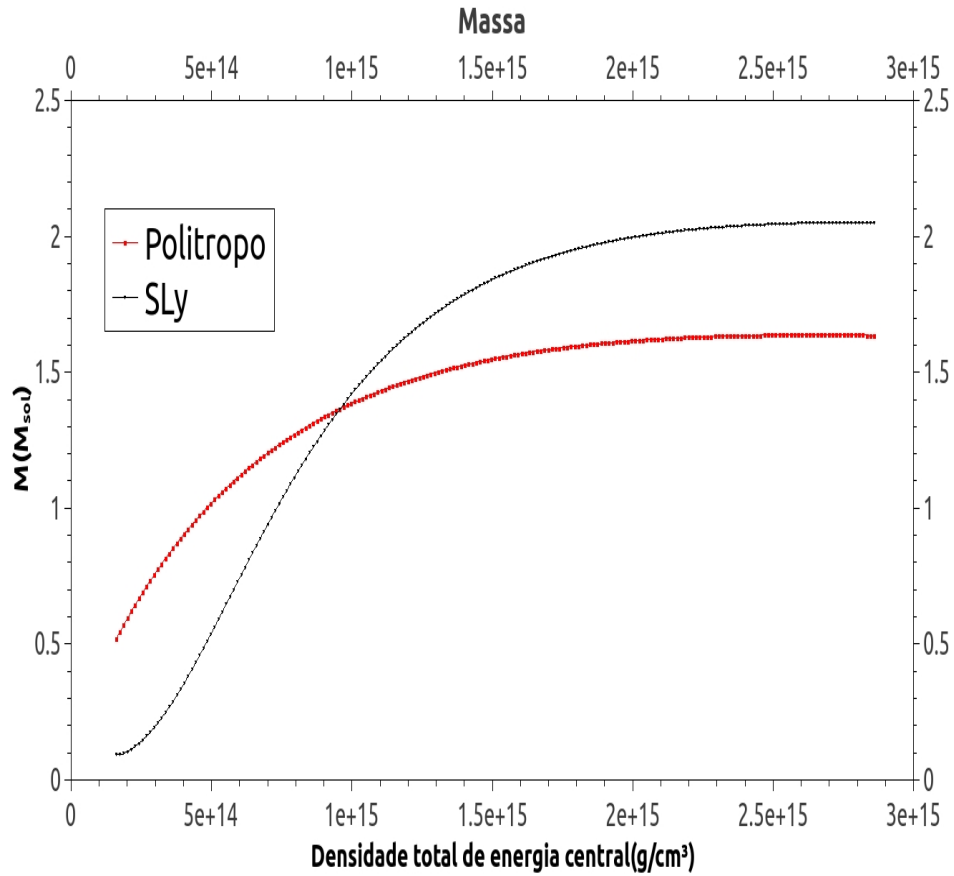


Figura 5.1: A massa da estrela em função da densidade central.

e $2,86 \times 10^{15} g/cm^3$, que é a faixa de densidade operacional da equação SLy. Para a equação politrópica temos $n = 1$ e $K = 145358 dyne.cm^4 g^{-2}$. Vemos que é bem determinada a massa em função da densidade, e que no começo a equação politrópica fornece estrelas mais massivas do que a equação SLy, mas esse quadro logo se reverte e a partir de aproximadamente da densidade de $1,1 \times 10^{15} g/cm^3$ a equação SLy passa a fornecer estrelas mais massivas. É interessante ver que a curva é crescente e quando atinge o valor da massa máxima ela se estabiliza momentaneamente para depois levemente começar a decair. Vale lembrar que as configurações após a massa máxima são de estrelas instáveis, que logo após seu nascimento colapsam em buracos negros.

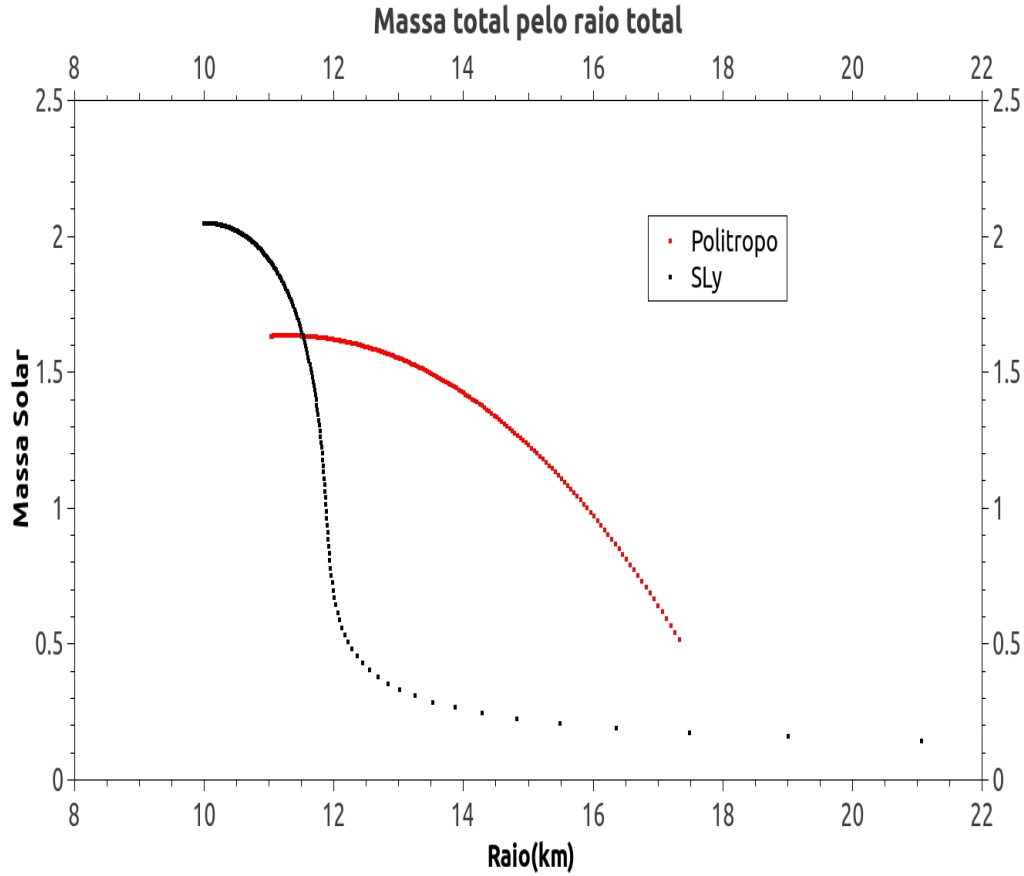


Figura 5.2: A massa da estrela em função do seu raio total.

Vemos que existe uma relação unívoca entre a massa da estrela e seu raio, assim com apenas um dado observacional podemos inferir o outro. Veja que numa grande faixa do gráfico, para uma mesma massa, a estrela politrópica apresenta um raio maior que a estrela SLy, isso ocorre pelo fato da equação politrópica ignorar algumas interações entre as partículas constituintes, que quando levadas em conta levam a uma compactação maior da estrela pois são forças atrativas. Vemos também que a equação SLy apresenta duas regiões onde não há massa equivalente para a equação politrópica, a primeira região nas altas densidades é bem comportada e fornece grandes massas, acima da massa máxima da equação politrópica; a segunda região é nas baixas

densidades e é mal comportada, a massa varia lentamente com o raio e chega a casa dos cinquenta quilômetros, é provável que essa banda necessite de um outro tipo de equação de estado mas devido à sua baixa massa e alto raio alguns autores, [3], discutem se na verdade seriam corpos transitórios entre as anãs brancas e as estrelas de nêutrons. Lembrando que a região à esquerda do ponto de massa máxima é a região de instabilidade.

5.2 As ondas gravitacionais simuladas

5.2.1 As ondas em função da densidade central

Pois bem finalmente chegamos onde gostaríamos, a comparação entre as ondas simuladas para nossas equações de estado. Iniciemos com os resultados simples da frequência da onda emitida em função da densidade central, embora na prática não há como determinar com exatidão a densidade central da estrela, já temos uma boa indicação de como a onda deve se comportar. Vejamos a figura 5.3. Ambas as curvas são bem comportadas. Vemos que neste gráfico a frequência emitida pela estrela SLy é maior do que a da estrela politrópica, sendo importante que ambas as curvas são bem distintas entre si, no futuro, quando os detectores terão a resolução necessária, será possível estimar a densidade central, e a equação de estado, a partir da onda medida. Vejam que o formato das duas curvas é semelhante, sendo que no início a curva SLy apresenta um aumento mais sensível do que a curva politrópica, sendo que logo após ambas aumentam suavemente, mas visualmente há constatação de que nesta região a curva politrópica cresce mais rapidamente do que a curva SLy.

Na figura 5.4 vemos o tempo de decaimento, que mede a taxa de atenuação da onda, em função da densidade central.

Vemos que as curvas são semelhantes entre si, mas a curva relativa à equação de estado SLy está um pouco mais deslocada para a direita em relação à outra. Vemos também que na região de baixas densidades elas apresentam valores distintos que variam fortemente com a densidade central, sendo que logo há uma estabilização onde ambas as curvas convergem para o valor de 0,14s. A resolução dos detectores atuais ainda não é suficiente para uma boa resolução do

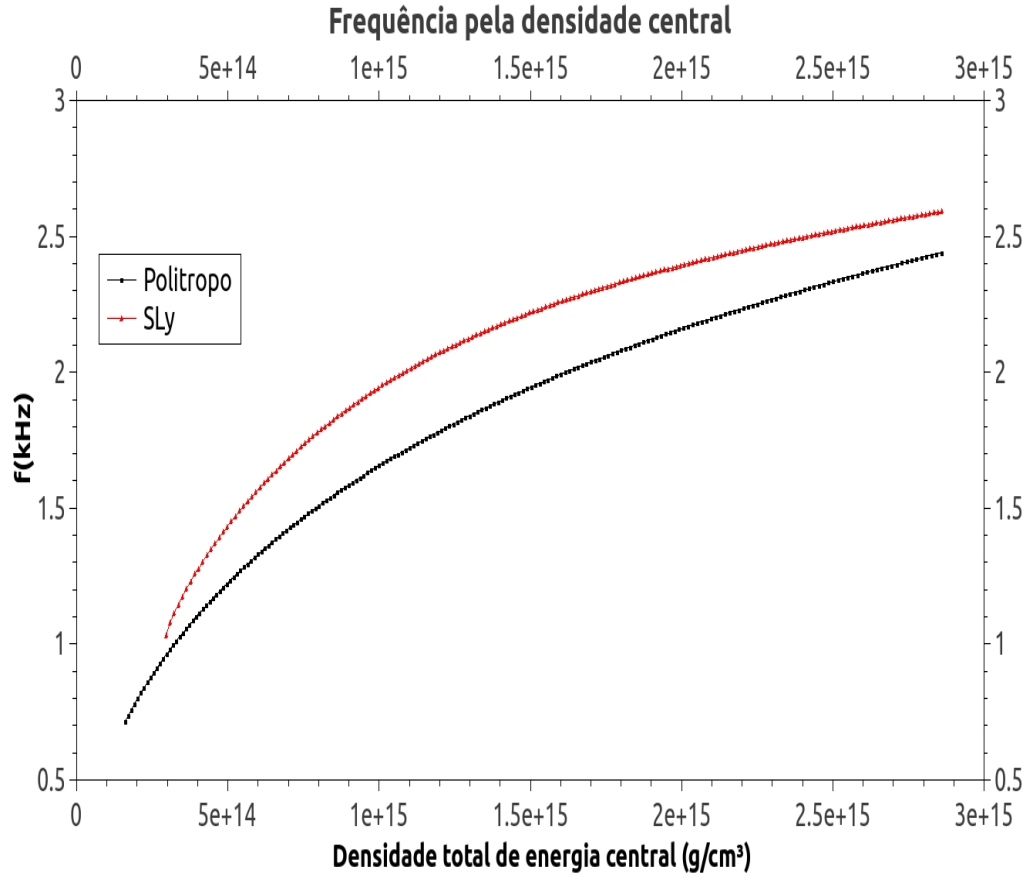


Figura 5.3: A frequência da onda emitida em função da densidade central.

tempo de decaimento, portanto aqui o tempo de decaimento não é um bom parâmetro para a definição da equação de estado de uma estrela.

No momento não dispomos de grandes detectores capazes de medir em detalhes as ondas gravitacionais, mas aqui abre-se uma janela para que no futuro, a partir de uma onda medida possamos calcular as propriedades da estrela emissora (massa, raio e densidade) e com isso estimar sua equação de estado. Note que isso seria o reverso do que fizemos neste trabalho, assim é uma via de mão dupla para a determinação do comportamento da matéria.

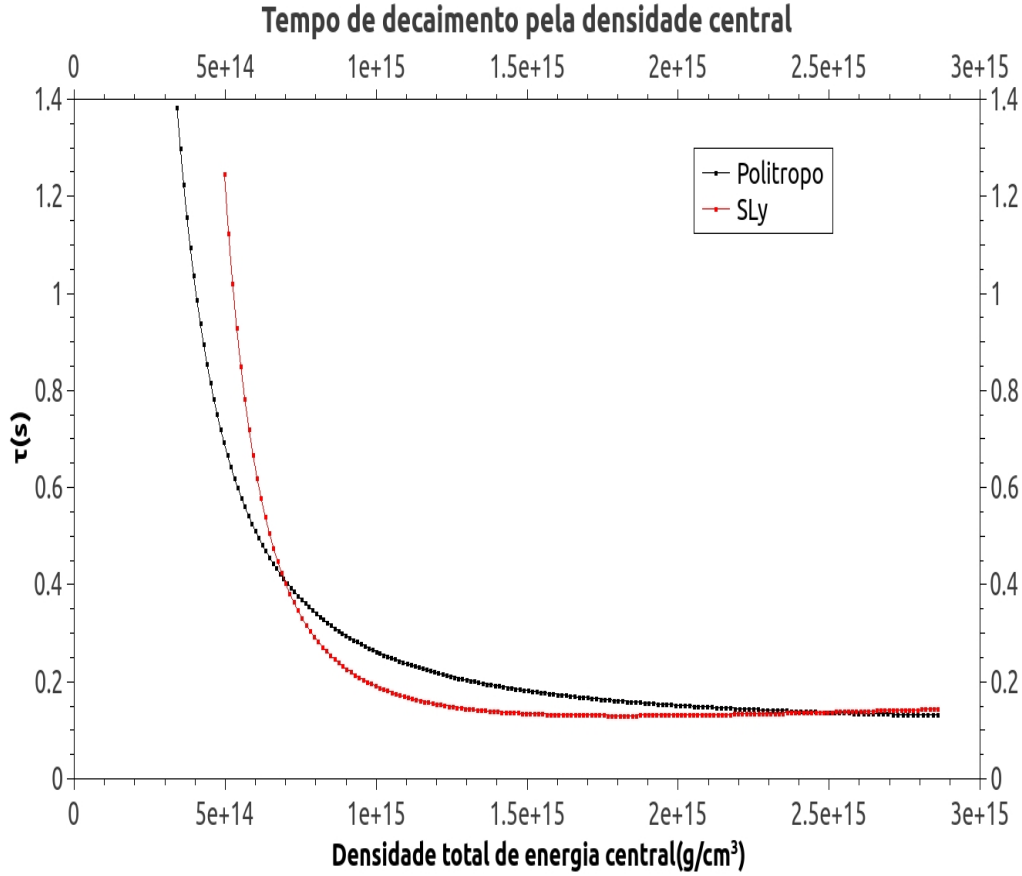


Figura 5.4: O tempo de decaimento da onda emitida em função da densidade central.

5.2.2 As ondas em função da massa da estrela

Agora apresentaremos os resultados em função da massa da estrela emissora, isso faz mais sentido do ponto de vista observacional pois a massa é mais fácil e mais precisa de ser determinada do que a densidade central. Vejamos a figura 5.5.

Vemos que é bem determinada a frequência da onda em função da massa da estrela, sendo que a curva SLy no começo apresenta um grande aumento com a massa, seguido duma região de aumento linear por fim chegando numa região de aumento rápido e acumulativo que culmina

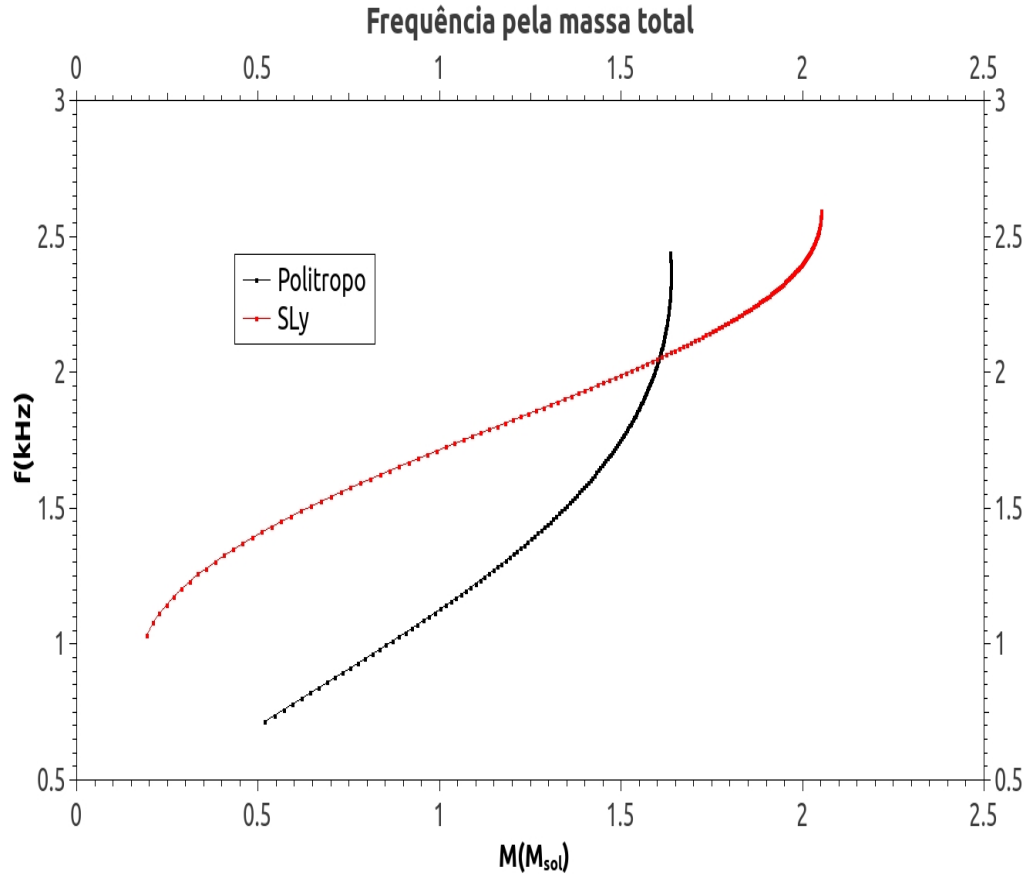


Figura 5.5: A frequência da onda emitida em função da massa da estrela.

na estrela de massa máxima. A curva politrópica já começa com uma região de aumento linear e depois um aumento rápido, vejam que interessante o que acontece após a massa máxima, há um retorno, uma diminuição na massa mas um aumento na frequência emitida. Como já discutido antes, após a massa máxima a estrela torna-se instável e colapsa num buraco negro, mas aqui vemos que as frequências produzidas são diferentes daquelas produzidas por uma estrela de nêutrons estável. Isso é interessante pois além do método óptico para se distinguir uma estrela de nêutrons de um buraco negro, agora pode ser proposto distingui-los pelas suas ondas gravitacionais emitidas.

Agora vejamos a figura 5.6 que apresenta o tempo de decaimento em função da massa da estrela.

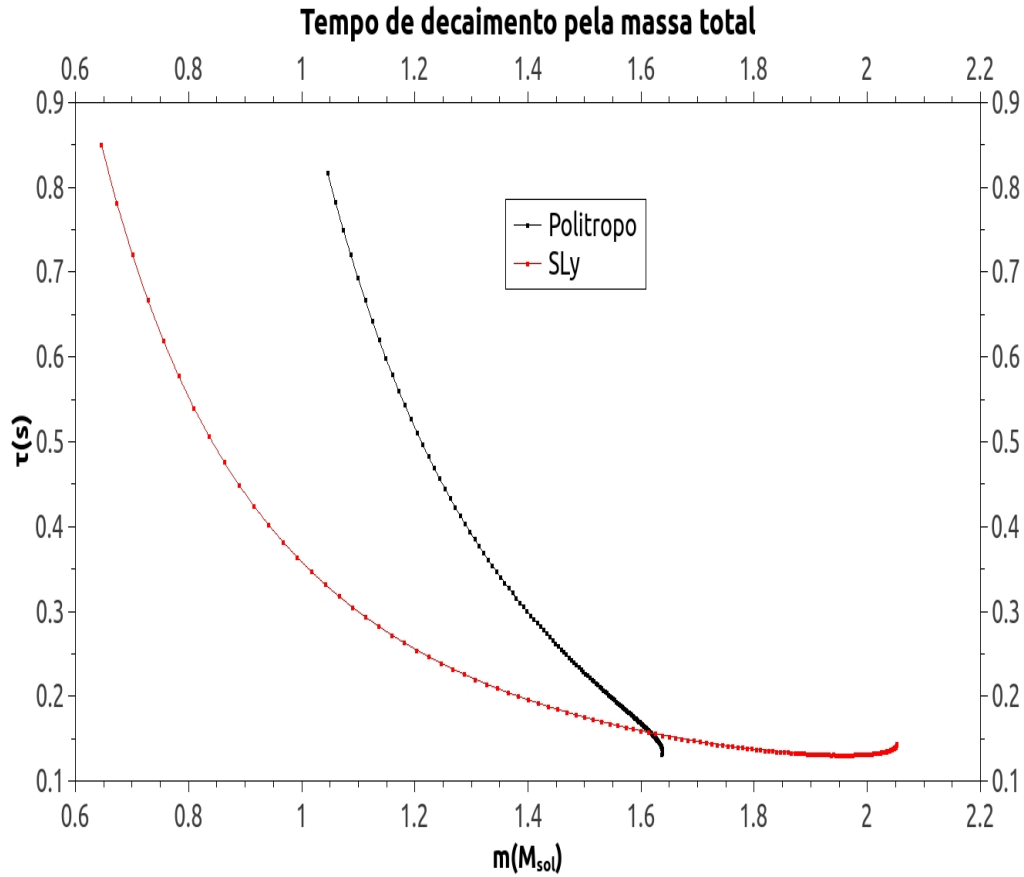


Figura 5.6: O tempo de decaimento da onda emitida em função da massa da estrela.

Vemos que ambas as curvas decrescem rapidamente, sendo que no início há uma forte dependência, uma pequena variação na massa causa uma grande variação no tempo de decaimento, mas conforme chegamos próximos à massa máxima, a distância entre os pontos diminui. A curva politrópica apresenta um decrescimento mais acentuado, sendo próximo a um decaimento linear nas regiões finais, sua distribuição é menor do que a da curva SLy pois a equação politrópica não fornece estrelas de baixa massa. A curva SLy possui um decrescimento acentuado logo no

início, se assemelhando à um decaimento exponencial, após isso a curva decresce lentamente de forma quase linear, até variar lentamente no final da curva.

5.3 As Linearizações

5.3.1 Frequência

Seguindo com nossas referências ([24], [25]) na teoria Newtoniana de perturbações estelares encontramos uma relação linear entre a frequência de oscilação do modo fundamental-f e a raiz quadrada da densidade média da estrela, tendo como inspiração a equação 3.98 ao se fazer a análise fenomenológica dos dados obtidos de simulações anteriores, ou seja:

$$f_f = b + a\sqrt{\frac{M}{R^3}}; \quad (5.1)$$

esta simples relação é uma poderosa fonte de informações pois diferentes equações de estado fornecem diferentes linearizações, assim com o espectro de radiação gravitacional vindo de um corpo podemos inferir sua equação de estado. Vejamos como nossas ondas se comportam perante esta relação segundo a figura 5.7. Importante, aqui a massa está em quilômetros, para obter este resultado basta multiplicar a massa, em unidades de massa solares, pelo fator de conversão 1,475km.

Veja que o início da curva SLy ainda não segue uma linearização perfeita, isso deve-se ao fato de que nossa equação de estado não é totalmente confiável no regime de baixas densidades, mas esta curva claramente segue um comportamento linear conforme a densidade aumenta. Note ainda que os pontos ficam menos afastados na região de altas densidades, o que mostra a maior precisão da nossa equação de estado nessa região. Por outro lado a equação politrópica apresenta um comportamento melhor no geral do que a equação SLy, especialmente na região de baixas densidades. Isso ocorre devido ao fato dela ser analítica em todos os pontos. Veja que há o mesmo comportamento dos pontos se ajuntarem mais na região de altas densidades. O interessante é que a equação de estado SLy, que é mais realista do que a politrópica, apresenta uma inclinação menor do que a politrópica. Isto pode ser um indicativo de que equações mais

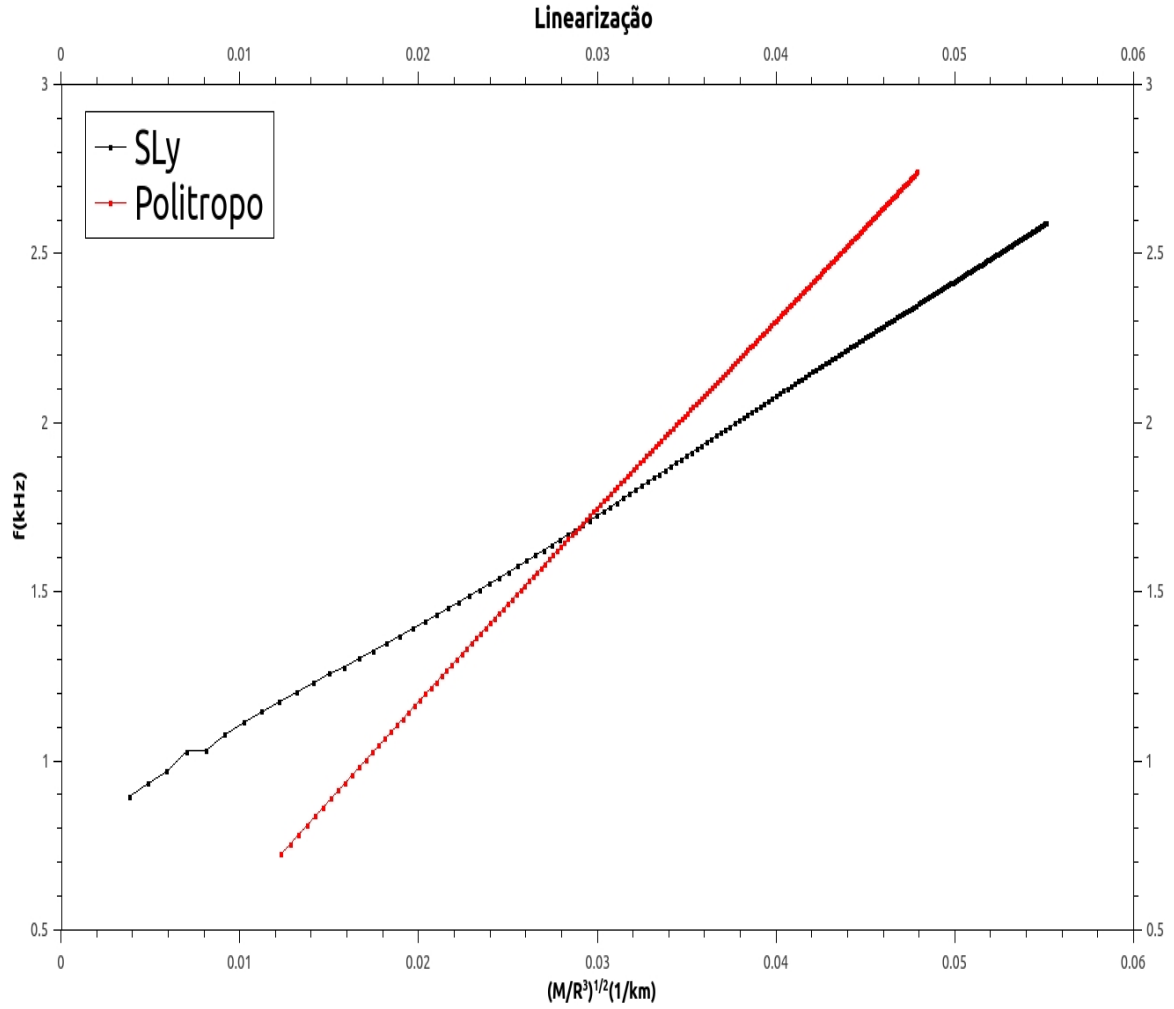


Figura 5.7: O comportamento de nossas duas equações de estado quando submetidas a relação 5.1.

realistas produzam uma curva mais suave do que equações menos realistas, mas isso deve ser comprovado via observação futura pelos detectores de ondas gravitacionais. Vejam que as retas cruzam-se apenas uma vez e que os pontos são bem diferenciados entre si, assim não há como confundir as duas curvas caso haja uma medida experimental, e como comentado anteriormente,

no futuro poderemos deduzir a equação de estado da fonte à partir da comparação entre as diferentes curvas.

Agora colocaremos o resultado de um ajuste linear aplicado às curvas, elas foram postas em gráficos separados para melhor visualização.

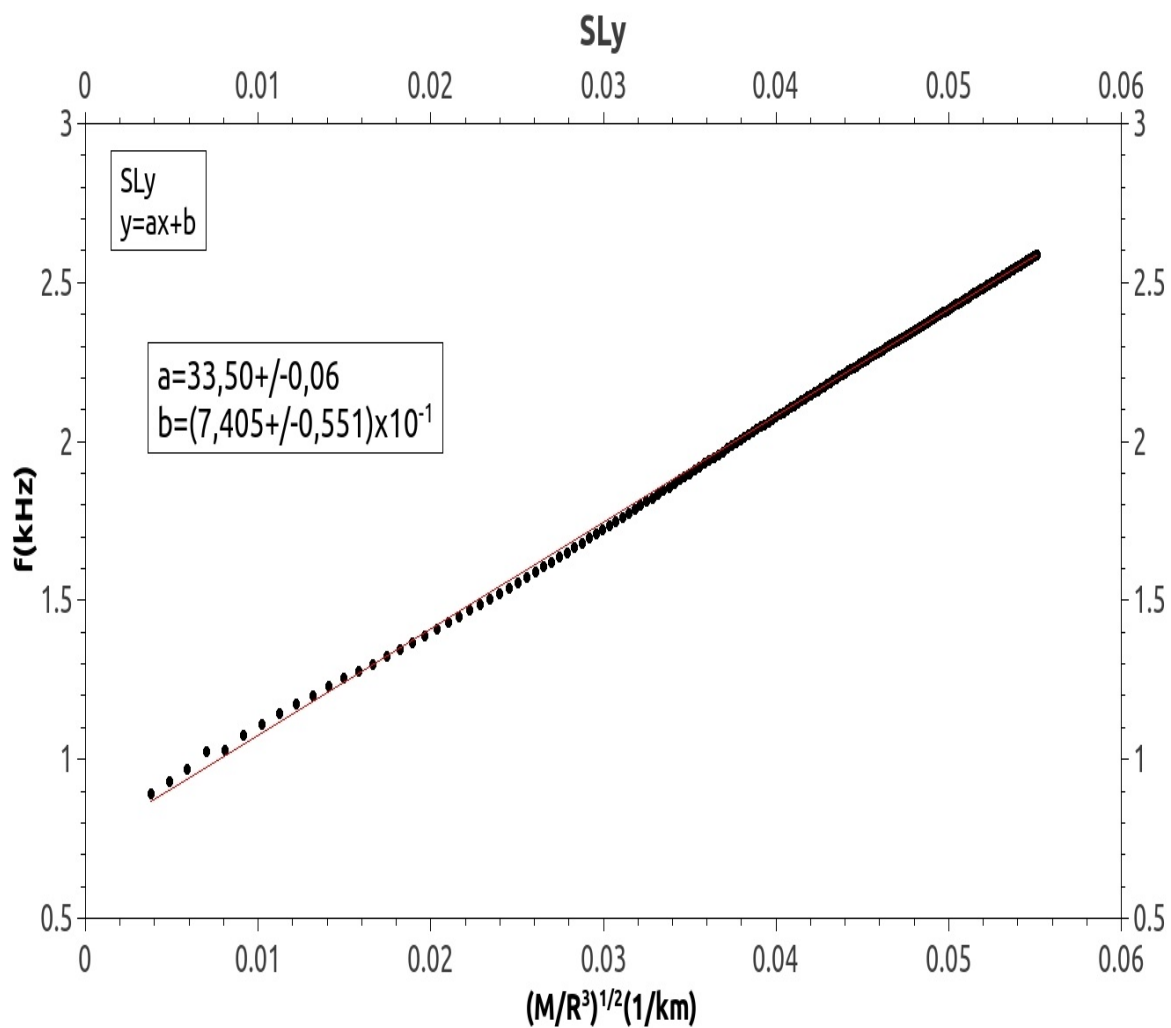


Figura 5.8: O ajuste linear para a curva SLy.

A curva 5.8 apresenta o ajuste feito para a curva referente à equação SLy, veja que a reta

não ajusta perfeitamente a região de baixas densidades mas ela cabe perfeitamente na região de altas densidades. Estatisticamente os pontos na região de baixa densidade se aproximam cada vez mais da reta, ora por cima, ora por baixo, assim a reta fornece resultados satisfatórios na região de baixas densidades. Os coeficientes angulares e lineares são, respectivamente:

$$a = (33,500 \pm 0,061)kHz \quad b = (7,405 \pm 0,551) \times 10^{-1}km \times kHz. \quad (5.2)$$

A curva 5.9 apresenta o ajuste feito para a curva referente à equação politrópica. Vejam que este ajuste é melhor na região de baixas densidades do que a equação SLy, mas mesmo assim o ajuste não é perfeito. Comparando com a equação SLy a reta está melhor ajustada pois ela tangencia os pontos na região de baixa densidade, enquanto que a reta SLy não chega a isto. No final da reta vemos que ela começa a se distanciar do centro dos pontos, começando ai a escapar o ajuste, isto é natural pois nessa região estamos nos aproximando do limite do buraco negro, e é natural que precisemos de uma outra análise. Mas no geral o ajuste gera resultados satisfatórios e é excelente na região de grandes e médias densidades. Os coeficientes angulares e lineares são, respectivamente:

$$a = (56,160 \pm 0,044)kHz \quad b = (5,658 \pm 0,160) \times 10^{-2}km \times kHz; \quad (5.3)$$

como dito anteriormente, agora nossos resultados confirmam que a curva politrópica apresenta um coeficiente angular maior do que a curva SLy.

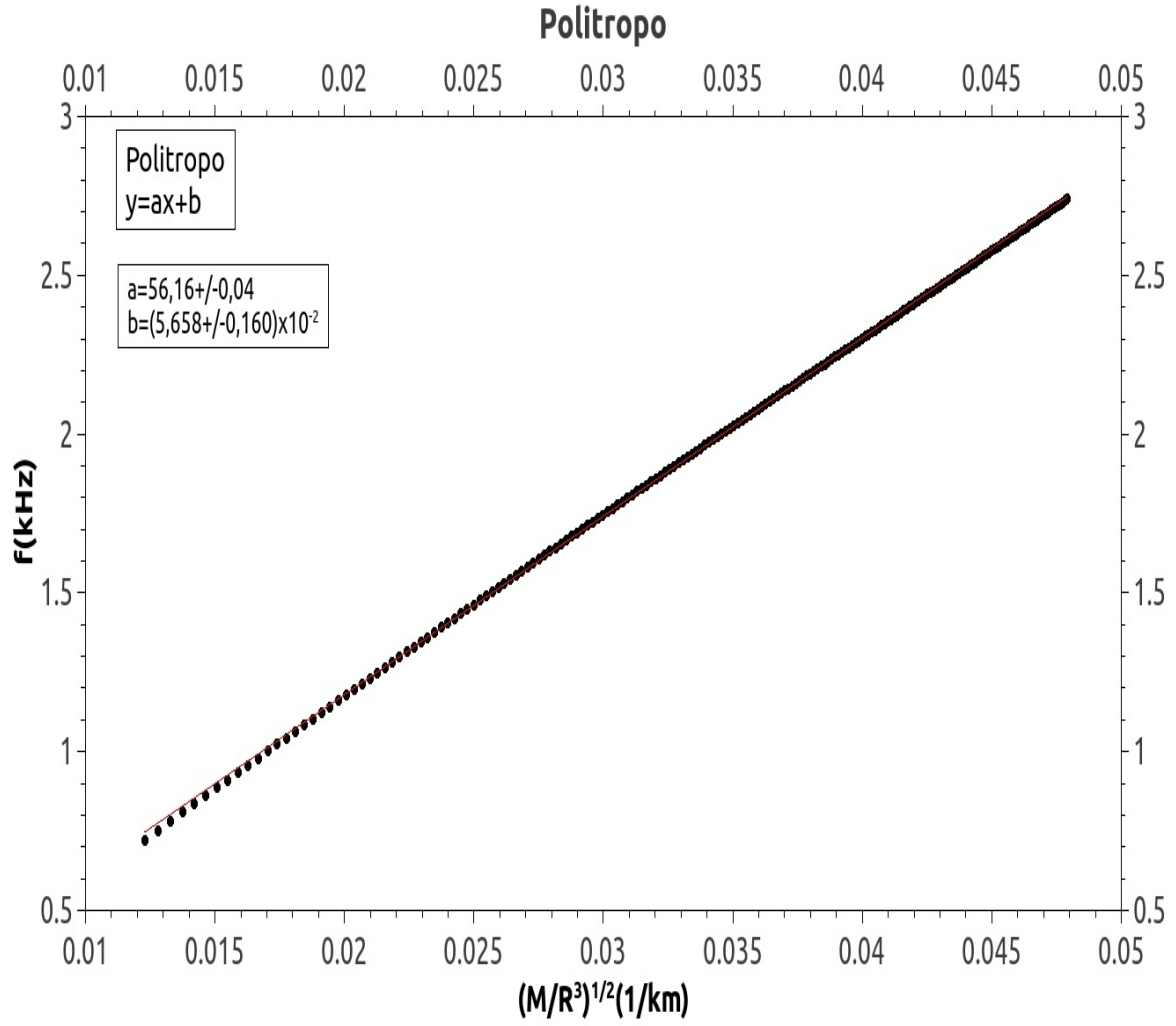


Figura 5.9: O ajuste linear para a curva politrópica.

5.3.2 Tempo de decaimento

Seguindo com a teoria Newtoniana temos que o tempo de decaimento do modo fundamental-f pode ser ajustado em função da massa e do raio do objeto como([24],[25]):

$$\tau_f = \frac{R^4}{cM^3} \left[b + a \frac{M}{R} \right]^{-1}; \quad (5.4)$$

mas esta relação pode ser linearizada na forma:

$$\frac{R^4}{\tau_f c M^3} = b + a \frac{M}{R}; \quad (5.5)$$

e é com esta linearização que apresentamos os resultados na curva 5.10.

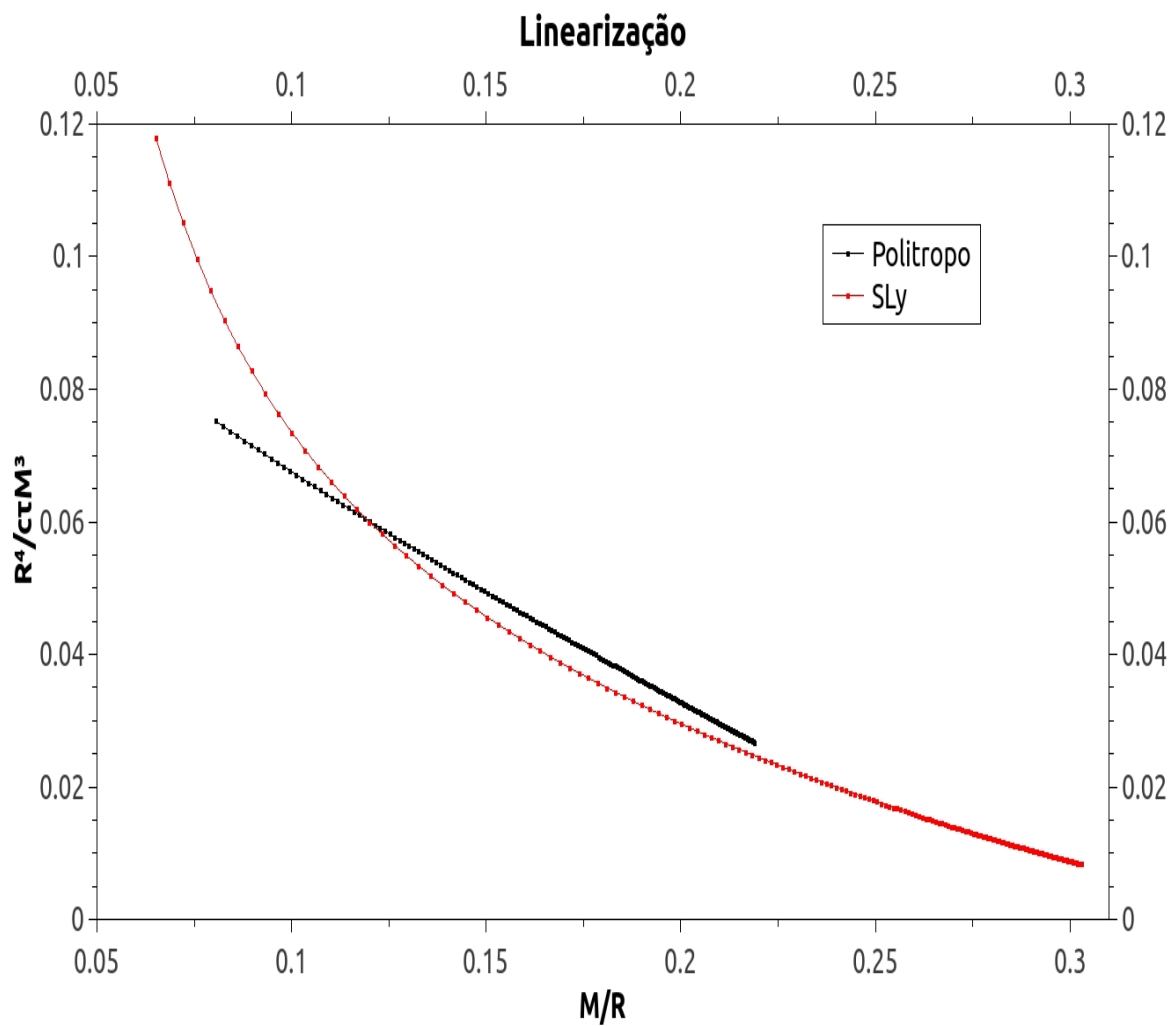


Figura 5.10: O comportamento de nossas duas equações de estado quando submetidas a relação 5.4.

Vemos que a curva politrópica apresenta uma perfeita linearização, assim ela gera excelentes

resultados que não se confundem com os resultados da curva SLy, mas como o modelo politrópico é um modelo simples não é de se esperar tamanha facilidade em dados observacionais. Por outro lado a curva SLy não apresenta um comportamento linear, ela se assemelha mais a um decaimento exponencial, somente na região de altas densidades é possível um ajuste linear. Como o modelo SLy é mais realista, são esses os resultados que esperamos obter numa futura medição, mas é necessário cuidado na região de baixas densidades. Vemos também que as curvas são bem distintas entre si, assim não há como confundi-las.

Na figura 5.11 vemos o ajuste para a curva SLy. É muito importante salientar que para este ajuste removemos o início da região de baixa densidade pois esta não corresponde de forma alguma a uma reta. Vemos que embora a curva não se ajusta a uma reta perfeita, sendo que nas regiões de baixas e médias densidades a reta escapa aos pontos para baixo e depois para cima, ainda há um bom ajuste que se torna excelente na região de altas compacidades. Os coeficientes angulares e lineares são, respectivamente:

$$a = (-1,915 \pm 0,010) \times 10^{-1} \quad b = (6,596 \pm 0,027) \times 10^{-2}; \quad (5.6)$$

tanto a quanto b são adimensionais.

Na figura 5.12 vemos o ajuste para a curva politrópica, aqui há um comportamento semelhante à curva SLy, onde nas regiões de baixas e médias densidades a reta escapa aos pontos para baixo e depois para cima, porém a diferença é menos acentuada do que na equação SLy, logo depois há um bom ajuste que se torna excelente na região de altas densidades. Esta persistência dos ajustes relacionados à equação SLy serem menos precisos do que aqueles relacionados à equação politrópica deve-se ao fato da equação politrópica ser analítica em todos os pontos, enquanto que a equação SLy é originária de dados experimentais, o que a torna intrinsecamente disposta a erros. Os coeficientes angulares e lineares são, respectivamente:

$$a = (-3,476 \pm 0,010) \times 10^{-1} \quad b = (1,026 \pm 0,002) \times 10^{-1}. \quad (5.7)$$

Vemos que novamente o coeficiente angular da curva politrópica é maior do que a da curva SLy, assim podemos novamente inferir que curvas realistas são mais suaves do que as outras.

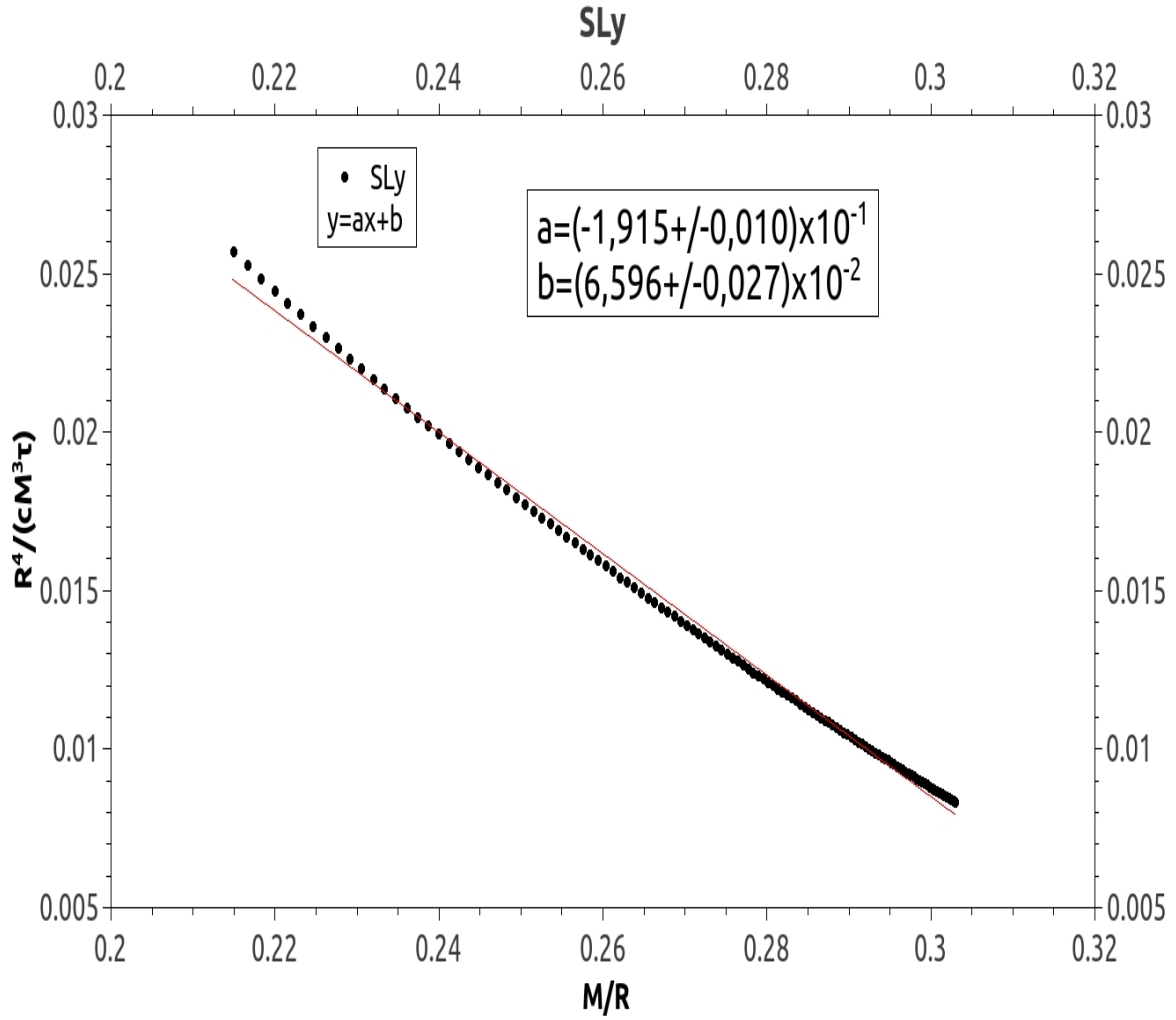


Figura 5.11: O ajuste linear para a curva SLy.

5.3.3 Comparação em relação à outros autores.

Aqui compararemos nossos resultados com os resultados dos trabalhos de Nils Andersson e Kostas D. Kokkotas ([24]), que fizeram uns dos primeiros estudos sobre a distribuição do espectro das ondas gravitacionais em função de diferentes equações de estado; e de Valeria Ferrari ([26]), que apresenta um pequeno resumo dos resultados obtidos pelo seu grupo.

Iniciemos com a frequência, na figura 5.13 vemos nossas retas politrópica e SLy junto com as retas obtidas por Andersson & Kokkotas, AK, e por Ferrari e seu grupo, OBG. Aqui colocamos somente as retas obtidas pela linearização dos resultados e omitimos os pontos para não poluir visualmente o gráfico devido ao excesso de pontos. Inicialmente vemos que as retas são bem distintas entre si, não há retas sobrepostas e isso torna fácil sua distinção quando se for analisar

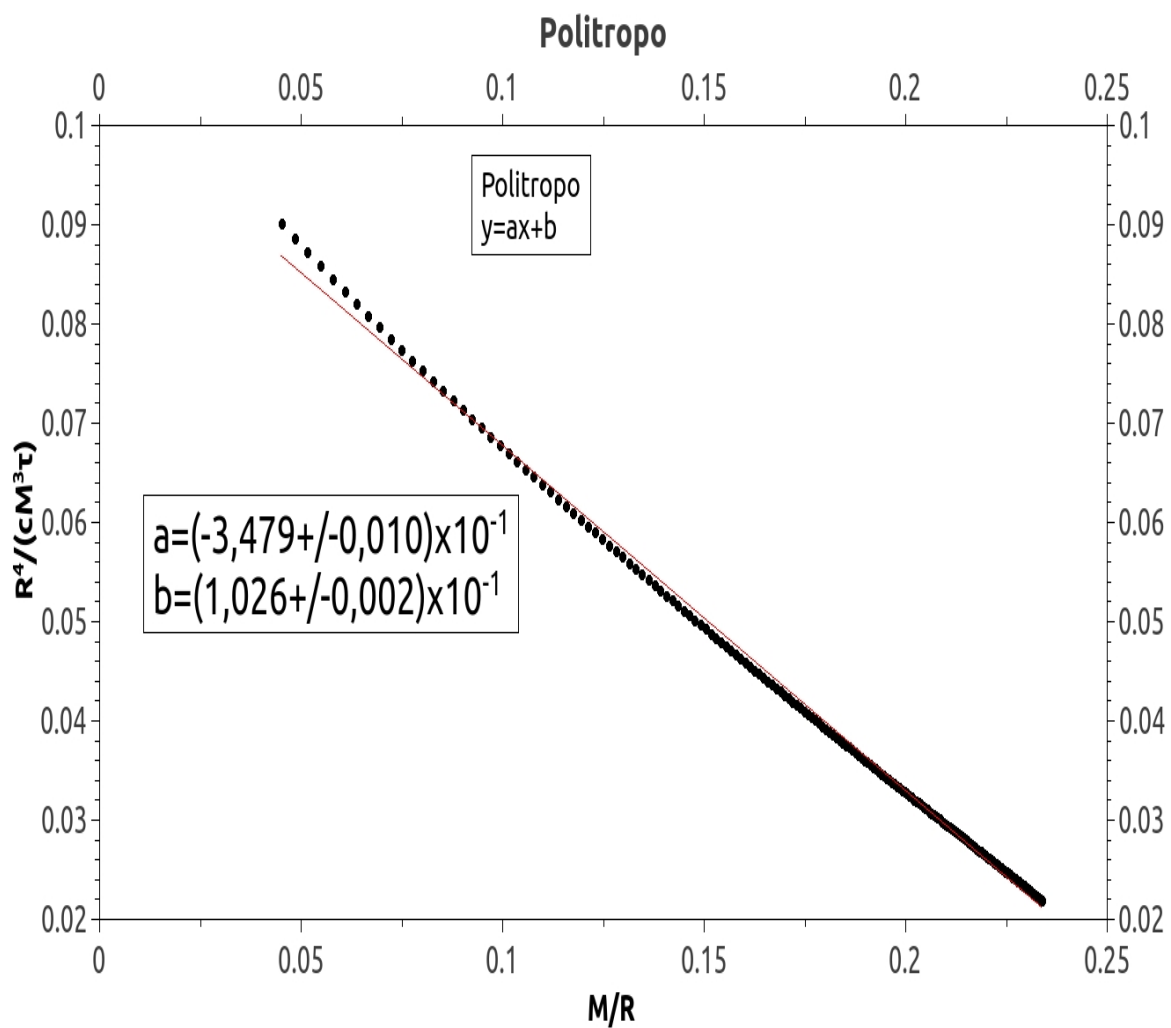


Figura 5.12: O ajuste linear para a curva politrópica.

os resultados experimentais. Nossa reta politrópica apresenta o maior coeficiente angular, como dito anteriormente ela é a menos realista das equações de estado, assim mais uma vez temos o indício de que equações menos realistas geram retas com maior coeficiente angular. Em seguida as retas OBG e AK são semelhantes entre si, entrando na região de altas frequências onde nossas estrela produzidas são instáveis. Como comentado no trabalho de Ferrari, era seu objetivo refinar um pouco mais os resultados anteriores de Andersson & Kokkotas, por isso a semelhança entre as duas retas. Por fim nossa reta SLy, ela é a que apresenta o menor coeficiente angular mas se aproxima da reta OBG para a região de altas frequências, sendo um indicativo de que nesta região nossas equações de estado tenham um mesmo comportamento. Nesta comparação vimos que a faixa geral de frequência vai do 0,6kHz até o 3,0kHz, o que está muito acima da faixa de detecção estimada para o LISA, assim a futura geração de detectores espaciais por interferometria não é capaz de medir as ondas vindas de estrelas de nêutrons perturbadas. Felizmente os detectores de massa ressonante operam numa faixa na ordem de unidades de quilo Hertz, especialmente neste caso o detector Mario Schenberg que operará num pico de 3,0kHz que será capaz de medir estas ondas com um erro razoável para separar o sinal do ruído de fundo([27]). Logo caberá aos detectores terrestres futuros detectar as ondas gravitacionais vindas de estrelas.

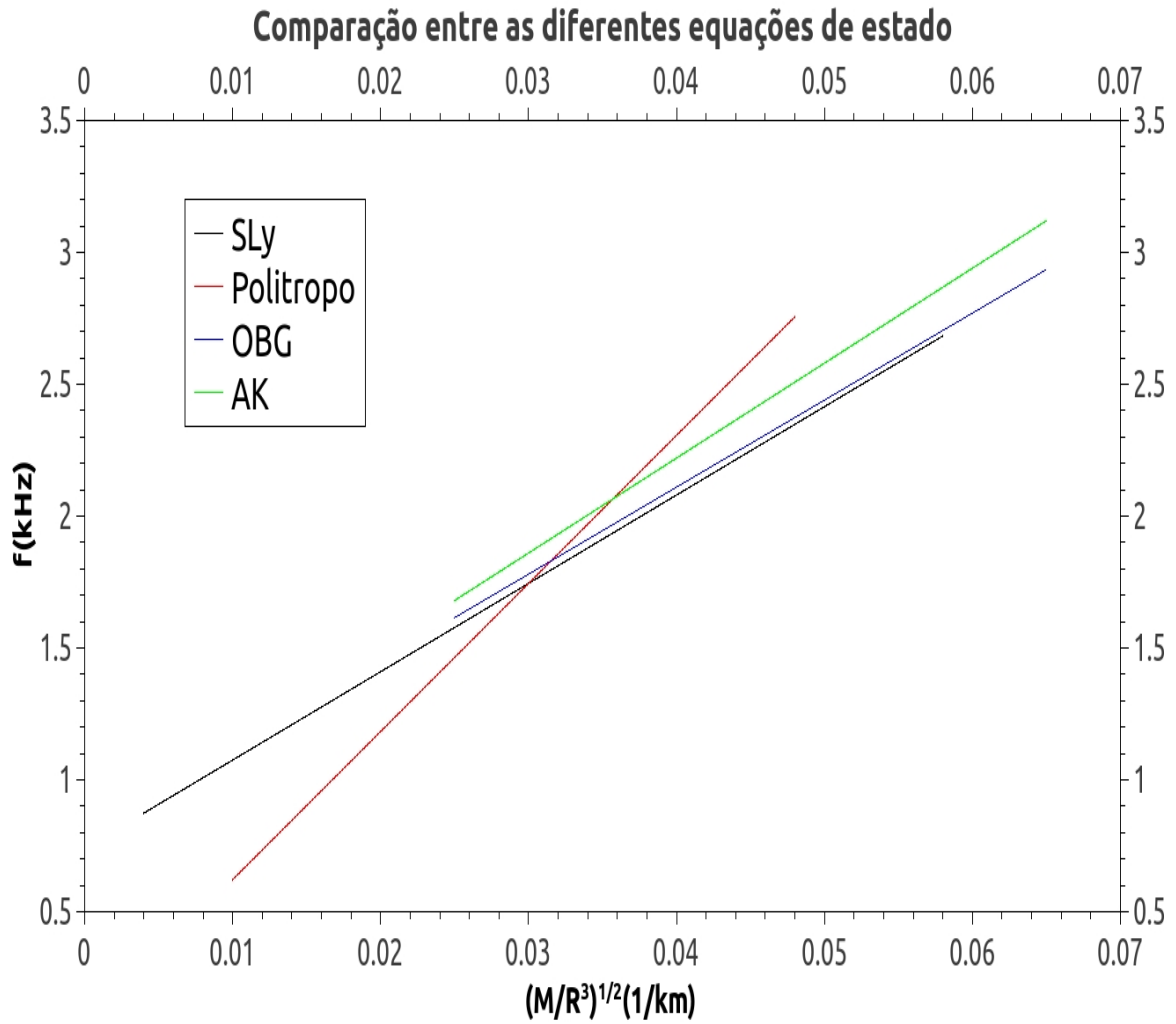


Figura 5.13: Comparando as linearizações obtidas.

Agora vejamos o tempo de decaimento que está exposto na figura 5.14. Vemos que nossa reta politrópica novamente apresenta o maior coeficiente angular enquanto que a reta SLy apresenta o menor coeficiente angular. O destaque aqui é para as retas OBG e AK que são idênticas, concordando com o trabalho de Ferrari. Nossa reta SLy apresenta um comprimento muito curto quando comparado com as outras, isso deve-se à retirada dos pontos iniciais onde

os resultados não concordam com uma linearização, como não havia sentido em calculá-los a sua faixa foi retirada do gráfico. Nossa reta politrópica segue o começo das retas OBG e AK mas logo afasta-se destas e torna-se bem distinta. As retas OBG e AK continuam para a região de alta compactação onde nossas estrelas entram no ramo instável. Pelo gráfico nossas retas e de nossas referências são bem distintas, assim numa análise experimental será fácil distingui-las e determinar a equação de estado do corpo irradiante.

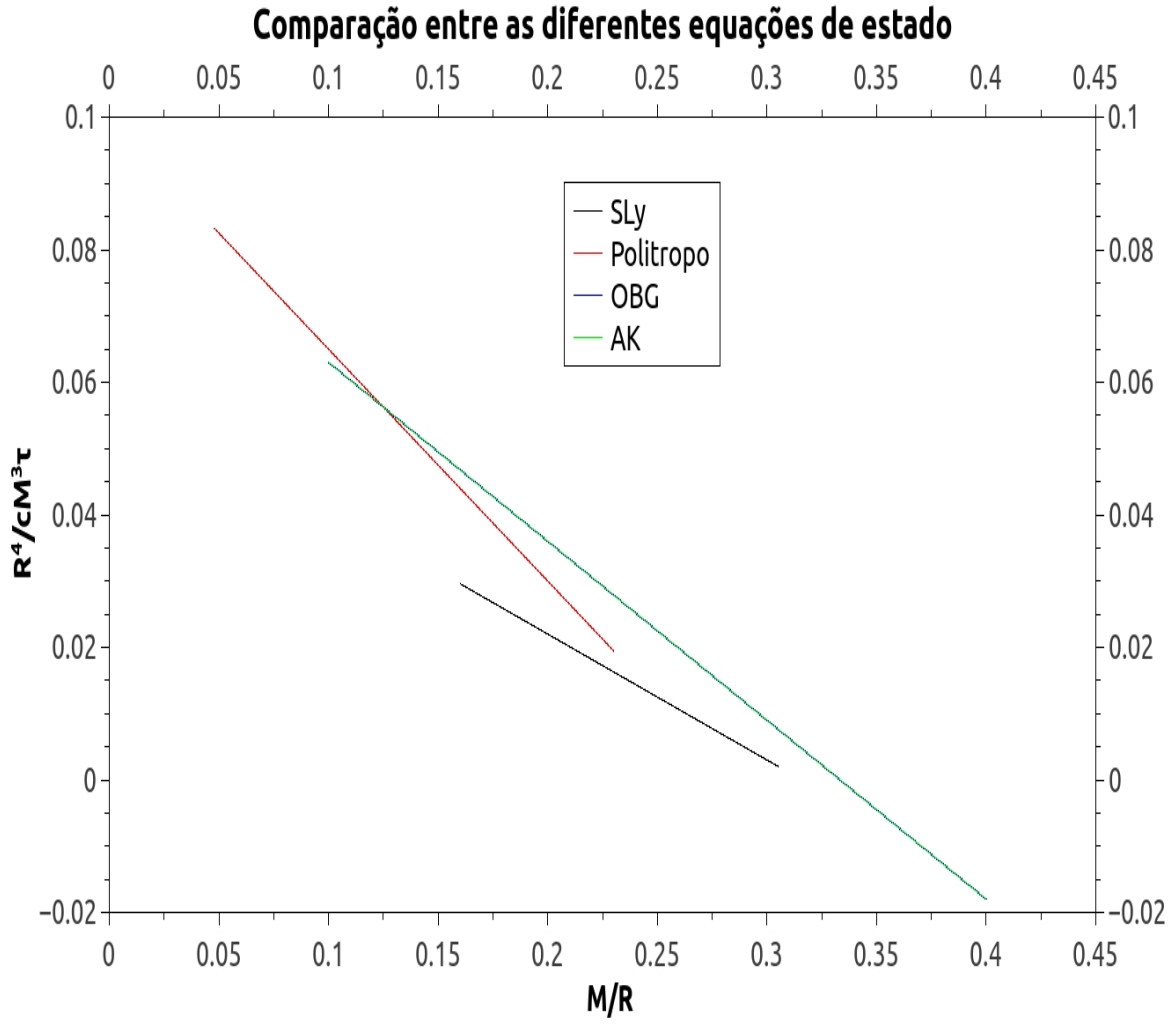


Figura 5.14: Comparando as linearizações obtidas.

Na tabela 5.1 apresentamos nossos parâmetros e dos dos autores utilizados para a figura 5.13, seguindo a nomenclatura da equação 5.1 e dentro da precisão do gráfico.

Agora na tabela 5.2 apresentamos nossos parâmetros e dos dos autores utilizados para a figura 5.14, seguindo a nomenclatura da equação 5.5 e dentro da precisão do gráfico.

Tabela 5.1: Parâmetros para a figura 5.13

| | a(kHz) | b(km.kHz) |
|-----------|------------------|-----------------|
| SLy | $33,50 \pm 0,06$ | $0,74 \pm 0,05$ |
| Politropo | $56,16 \pm 0,04$ | $0,06 \pm 0,01$ |
| OBG | $33,00 \pm 2,00$ | $0,79 \pm 0,09$ |
| AK | $36,00 \pm 0,10$ | $0,78 \pm 0,01$ |

Tabela 5.2: Parâmetros para a figura 5.14

| | a | b |
|-----------|--------------------|-------------------|
| SLy | $-0,191 \pm 0,001$ | $0,066 \pm 0,002$ |
| Politropo | $-0,350 \pm 0,001$ | $0,10 \pm 0,01$ |
| OBG | $-0,270 \pm 0,001$ | $0,09 \pm 0,01$ |
| AK | $-0,270 \pm 0,001$ | $0,09 \pm 0,01$ |

Ainda comparando nossos resultados, SFC, agora apresentaremos nossas frequências e tempos de decaimento para cinco estrelas SLy, comparando com aqueles obtidos por Bernuzzi, ([28], [29]), junto com a diferença relativa destes com os nossos resultados. Tudo é apresentado na tabela 5.3.

Tabela 5.3: Nossos resultados para a equação SLy comparados com Bernuzzi.

| M_{\odot} | SFC f(Hz) | Bernuzzi f(Hz) | $\Delta f(\%)$ | SFC $\tau(s)$ | Bernuzzi $\tau(s)$ | $\Delta \tau(\%)$ |
|-------------|-----------|----------------|----------------|---------------|--------------------|-------------------|
| 1,00 | 1718 | 1691 | 1,6 | 0,3540 | 0,3300 | 7,3 |
| 1,20 | 1824 | 1804 | 1,1 | 0,2549 | 0,2250 | 13,3 |
| 1,40 | 1932 | 1932 | 0 | 0,1957 | 0,1890 | 3,5 |
| 1,60 | 2048 | 2029 | 0,8 | 0,1592 | 0,1600 | 0,5 |
| 2,05 | 2591 | 2607 | 0,6 | 0,1440 | 0,1452 | 0,8 |

Vemos que nossos resultados concordam grandemente com os obtidos por Bernuzzi, o que

ajuda a validar nosso estudo, as diferenças de poucas unidades nas frequências obtidas devem-se ao outro método computacional por ele utilizado, mas como ainda não dispomos de dados observacionais e com as estimativas de sensibilidade dos futuros detectores, estes resultados estão na mesma faixa de erro.

5.4 O Problema Inverso

Embora nosso estudo tenha se limitado a duas equações de estado, há outros onde várias outras equações de estado são utilizadas e onde os resultados são apresentados nas forma das linearizações aqui utilizadas. Com o avanço da área logo teremos muitos modelos e seus resultados esperando por verificação, e com os detectores cada vez mais avançados breve será o dia em que finalmente detectaremos as ondas gravitacionais e colocaremos à prova todas nossas teorias. Atualmente há outras equações de estado que também derivam de dados experimentais vindo de colisores, nos quais usaram-se regimes de energia diferentes uns dos outros, e onde também aplicou-se diferentes modelos de física de partículas para ajustar os dados intermediários entre dois pontos experimentais. Por isso esperamos que com as medições de ondas gravitacionais possamos filtrar esses modelos e obter aqueles que correspondem à estrelas reais. Quando detectarmos uma onda com sua frequência e tempo de decaimento seremos capazes de determinar o raio e a massa da estrela emissora e com isso sua equação de estado. Iniciemos isolando a massa M na equação 5.1:

$$M = R^3 \left(\frac{f_f - b}{a} \right)^2; \quad (5.8)$$

substituindo na equação 5.5, tomando o cuidado de substituir os coeficientes a e b desta equação por e e d para não confundir com os anteriores, obtemos:

$$\frac{1}{cR^5 \left(\frac{f_f - b}{a} \right)^6 \tau} = eR^2 \left(\frac{f_f - b}{a} \right)^2 + d; \quad (5.9)$$

reescrevendo obtemos:

$$ce\tau \left(\frac{f_f - b}{a} \right)^8 R^7 + cd\tau \left(\frac{f_f - b}{a} \right)^6 R^5 - 1 = 0. \quad (5.10)$$

Agora lembramos que f_f e τ foram medidos pelo nosso detector, são dados experimentais, e que c é a velocidade da luz, e temos que a , b , d e e são os parâmetros obtidos pelas nossas linearizações. Substituindo convenientemente os coeficientes para melhor visualização obtemos:

$$\alpha R^7 + \beta R^5 - 1 = 0. \quad (5.11)$$

Resolvendo esta equação obtemos o valor do raio da estrela que emitiu a onda detectada, mas como esta é uma equação sétima temos que tomar cuidado para filtrar os valores obtidos para aqueles aceitos para uma estrela de nêutrons, lembrando que esta equação depende de qual equação de estado utilizamos para obter os coeficientes α e β , assim como esperado equações de estado diferentes darão raios diferentes. Uma vez obtido o raio correto basta substituí-lo na equação 5.8 para encontrar o valor da massa da estrela que produziu a onda. As ondas gravitacionais ainda facilitarão a detecção da massa e do raio de estrela de nêutrons solitárias, pois como vimos cada estrela possui um raio, uma massa e uma frequência bem definidos, e é muito difícil medir o raio e a massa de estrelas que não tenham uma companheira estelar para comparação.

Isto é interessante para a previsão de *Quarkstars*, estrelas teoricamente feitas de quarks desconfiados e que estariam entre as estrelas de nêutrons e os buracos negros, embora ainda exista muita controvérsia se são apenas especulação teórica ou passíveis de existência, da mesma forma que aconteceu com as estrelas de nêutrons no começo do século passado, previu-se que somente elas seriam capazes de emitir ondas gravitacionais perfeitamente esféricas, assim caberia a detecção destas confirmar ou não tais estrelas novas. Ainda neste ramo, nos modelos que descrevem a interação entre os quarks constituintes aparece o termo “constante de bolsa”, que define o raio médio de interação entre estes quarks e foi visto que variações neste termo geram estrelas de massas e raios diferentes, assim medindo as ondas por estas emitidas teríamos mais dados experimentais sobre o valor deste termo que governa sobre uma das partes do mundo quântico([30]).

Conclusões

Graças aos nossos estudos conseguimos especificar as ondas gravitacionais vindas de estrelas de nêutrons descritas a partir de duas equações de estado bem distintas. Comparando primeiramente a estrutura de duas estrelas simuladas tomamos noção de como estas duas equações de estado produzem corpos diferentes, sendo que esta descrição interna serve para prever como a matéria nuclear se comportaria sobre tais condições e qual seria o corpo resultante de uma situação de equilíbrio. Como dito no corpo da tese as equações TOV asseguram que um corpo relativístico mantenha-se em equilíbrio hidrostático, cabendo à equação de estado garantir o equilíbrio microscópico entre suas partículas interagentes. É claro que dada a infinidade de estrelas de densidades diferentes que podemos simular nós não poderíamos descrever todas neste trabalho. Por isso no capítulo dos resultados das simulações optamos por apresentar os resultados externos para 200 estrelas simuladas, pois assim é mais fácil distinguir as estrelas observadas em relação a sua equação de estado.

Toda a matemática envolvendo ondas gravitacionais já está desenvolvida desde o meio do século *XX*, o que facilitou enormemente o desenvolvimento de nossos códigos computacionais para calcular as ondas, uma vez que já dispúnhamos de uma extensa bibliografia sobre como resolver as equações por quais métodos e até onde prosseguir com o cálculo pra obter precisão

sem perda de tempo. Por se tratar de um complexo sistema de equações atreladas e integrações não analíticas não haveria como desenvolver tudo manualmente, e por isso mais uma vez devemos nosso sucesso à computação.

Como a maioria das previsões da relatividade geral, desde a dilatação do tempo até os buracos negros, precisou-se de anos até que tais propostas, vistas na época como loucuras teóricas, fossem gradualmente aceitas com o número crescente de experimentos e observações que concordavam com tais predições. E assim está sendo com as ondas gravitacionais, toda sua elegância matemática e analogias às ondas eletromagnéticas tornaram-nas mais fáceis de aceitação, ao mesmo tempo que as estimativas de detecção diziam que a tecnologia da época era incapaz de medi-las, não houve outra escolha senão esperar que com o tempo a tecnologia avançasse e permitisse isso. E agora parece que chegamos na época em que a tecnologia está apta a detectar as ondas gravitacionais.

Nossas simulações já oferecem uma janela de observação para os detectores e uma estimativa de que tipo de onda esperar vindo de objetos astronômicos, como vimos nossa faixa de frequência de 0,7kHz até 3,0kHz é um forte indicativo de que os detectores de interferometria são incapazes de detectar as ondas vindas destes corpos. Assim com a faixa de operação estimada para estes detectores é mais prudente pensar que eles são capazes de detectar as ondas gravitacionais vindas de buracos negros super massivos. Felizmente os detectores de barra ressonante são capazes de operar nesta faixa mas a grande acurácia tecnológica necessária para isso ainda torna inviável seu pleno funcionamento. Uma das grandes expectativas atuais gira em torno dos detectores esféricos, como o miniGRAIL e o Mario Schenberg, que operariam nesta faixa de frequência e apresentariam a grande vantagem de indicar a direção de origem da onda.

Embora nossas linearizações relativas às estrelas que seguem a equação de estado SLy não tenham atingindo a perfeição, devemos ter em mente que nossos modelos não conhecem em detalhes estes objetos astronômicos e que ainda faltam vários aspectos a serem averiguados em relação ao comportamento da matéria nestas condições. Ao comparar nossos resultados com os de outros autores percebemos a tendência de que equações de estado realistas tendam

à gerar ondas de frequência menor do que aquelas geradas por equações menos realistas. No futuro esperamos que à partir dos dados obtidos pelas ondas gravitacionais sejamos capazes de estimar a equação de estado e a estrutura do corpo que a produziu.

Apoio

Este projeto foi financiado pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, CAPES.

Referências Bibliográficas

- [1] Kepler S. O. Filho, Maria F. O, Saraiva, *Astronomia e Astrofísica*, Editora Livraria da Física, São Paulo, 2004, 2ªed.;
- [2] Rüdiger Pakmor, Neutron Star Radii and Equation of State, 21/01/2005;
- [3] P. Haensel, A.Y. Potekhin, and D.G. Yakovlev, *Neutron Stars 1: Equation of State and Structure*, Astrophysics and Space Science Library, Springer;
- [4] F. Douchin and P. Haensel; A unified equation of state of dense matter and neutron star structure; *A&A* 380, 151–167 (2001);
- [5] P. Haensel and A. Y. Potekhin; Analytical representations of unified equations of state of neutron-star matter; *AA* 428, 191–197 (2004);
- [6] Maciel, Walter J.; *Introdução à estrutura e evolução estelar*. Editora da Universidade de São Paulo, 1999;
- [7] Tooper, Robert F.; *Adiabatic fluid spheres in general relativity*; IIT Research Institute, Chicago, 1965;
- [8] Shapiro, L. S.; Teukolsky, S. A.; *Black holes, white dwarfs, and neutron stars, the physics of compact objects*. New York, NY: John Wiley, c1983;

- [9] P. Haensel and B. Pichon; Experimental nuclear masses and the ground state of cold dense matter; *A&A* 283, 313-318(1994);
- [10] F. Douchin and P. Haensel; A unified equation of state of dense matter and neutron star structure; *A&A* 380, 151–167 (2001);
- [11] W. David Arnett and Richard L. Bowers; A microscopic interpretation of neutron star structure; *The astrophysical Journal Supplement Series*, 33: 415-436, 1977;
- [12] Schutz, B.F.; A first course in general relativity. Cambridge: Cambridge University Press, c2009;
- [13] Hartle, J. B.; Gravity: An introduction to Einstein’s General Relativity. San Francisco: Addison-Wesley, c2003;
- [14] Detweiler, S. Lidblom, L; On the nonradial pulsations of general relativistic stellar models; *The Astrophysical journal*, 292: 12-15, 1985;
- [15] Detweiler, S. Lidblom, L; The quadrupole oscillations of neutrons stars; *The Astrophysical journal supplement Series*, 53: 73-92, 1983;
- [16] Patrick da Rosa Silveira; Estrelas de nêutrons como fontes de ondas gravitacionais e a detecção do modo fundamental; Instituto Nacional de pesquisas espaciais-INPE, Março-2012;
- [17] Richard Price, Kip S. Thorne, Non-radial pulsation of general relativistic stellar models. II. Properties of the gravitational waves, *The astrophysical journal*, Vol.155, January 1969;
- [18] Kip S. Thorne, Non-radial pulsation of general relativistic stellar models.III Analytic and numerical results for neutron stars, *The astrophysical journal*, Vol.158, October 1969;
- [19] Thorne, Kip S.; Campolattaro, Alfonso; Non-radial pulsation of general relativistic stellar models. I.Analytic analysis for $l \geq 2$, *The astrophysical journal*, Vol.149, September 1967;

- [20] Joel M. Weisberg, Joseph H. Taylor, Relativistic Binary Pulsar B1913+16: Thirty Years of Observations and Analysis, ASP Conference Series, Vol. TBD, 2004;
- [21] J. Weber, Anisotropy and Polarization in the Gravitational-Radiation Experiments. *Phys. Rev. Lett.* 25, 180–184 (1970);
- [22] O. D. Aguiar et al, The Brazilian spherical detector: progress and plans, O D Aguiar et al 2004 *Class. Quantum Grav.* 21 S457;
- [23] José Luiz Melo, Sistemas de isolamento vibracional e de Acoplamento antena-transdutores para o Protótipo de um detector de ondas gravitacionais, São José dos Campos, INPE, 2002. 199p. – (INPE-9693-TDI/854);
- [24] Nils Andersson, Kostas D. Kokkotas, Towards gravitational wave asteroseismology, *Mon. Not. R. Astron. Soc.* 299, 1059–1068 (1998);
- [25] Omar Benhar, Valeria Ferrari, Leonardo Gualtieri, Gravitational wave asteroseismology reexamined, *Physical review d* 70, 124015 (2004);
- [26] Valeria Ferrari, Gravitational waves from perturbed stars, *Bull. Astr. Soc. India* (2011) 39, 203–224;
- [27] Cecilia Chirenti, Patrick r. Silveira, Odylio d. Aguiar, Non-radial oscillations of neutron stars and the detection of gravitational waves, 5th International Workshop on Astronomy and Relativistic Astrophysics (IWARA2011), *International Journal of Modern Physics: Conference Series* Vol. 18 (2012) 48–52;
- [28] Sebastiano Bernuzzi, Alessandro Nagar, Gravitational waves from pulsations of neutron stars described by realistic Equations of State, *General Relativity and Quantum Cosmology (gr-qc); Astrophysics (astro-ph)*, *Phys.Rev.D*78:024024, 2008;

- [29] Sebastiano Bernuzzi, Numerical simulations of relativistic star oscillations, Gravitational waveforms from perturbative and 3-dimensional codes, Thesis submitted for the award of the degree of PhD in Physics, XXI ciclo - January 2009, Università degli Studi di Parma;
- [30] GWIC, The gravitational wave international committee roadmap, The future of gravitational wave astronomy, June 2010.

Apêndice

Aqui colocaremos todos os programas utilizados com uma breve explicação sobre eles.

Zerilli4

Este é o programa mestre que concatena todos os outros programas.

```
program Zerilli4
  implicit none
  integer, parameter :: dp= SELECTED_REAL_KIND(15,300), max_it = 20
  integer, parameter :: ns = 5000, ns2 = 10000
  real(kind = dp) :: l, nn, K, rhob, pi, omega, omegaA, omegaB, omegaC
  real(kind = dp) :: fHz, Ts, MM, RR1, rho
  complex(kind = dp) :: gamaA, betaA, gamaB, betaB, gamaC, betaC
  complex(kind = dp) :: omega_raiz, omega_old
  complex(kind = dp) :: f0,f1,f2,p,a,b,c
  real(kind = dp) :: x0,x1,x2, teste_omega
  real(kind = dp),parameter :: TOL = 1.0d-8, CactusDens = 6.176d+17
  integer :: i, j
  real(kind= dp), dimension (3,3) :: mG, mH
  open(51,file="Teste_omega_raiz_new.dat")
  open(52,file="Teste_omega_raiz_new2.dat")
  open(53,file="Teste_gama_raiz_new.dat")
  !open(54,file="rhofreqt.dat")
  nn= 1.00d0
  l=2.0d0
  K=100.0d0
  rhob=(7.32d+14/CactusDens)
  pi= 4.0d0*atan(1.0d0)
  rho = rhob
  call tov_rot.z(nn,K,rho,l,omega,ns2,MM,RR1)
  ! Definição dos Omegas
  omegaA=0.99d0*omega
  omegaB=1.0d0*omega
  omegaC=1.01d0*omega
  ! Calcula os gamas
```



```

call zerilli2_rot(nn,K,rho,l,omegaA,gamaA,betaA,ns,ns2)
call zerilli2_rot(nn,K,rho,l,omegaB,gamaB,betaB,ns,ns2)
call zerilli2_rot(nn,K,rho,l,omegaC,gamaC,betaC,ns,ns2)
write(51,'(9(f15.6,1x))')omegaA, omegaB, omegaC, abs(gamaA), abs(betaA), abs(gamaB), abs(betaB), abs(gamaC), abs(betaC)
! Ordena os gamas
! gamaA = gamaA/betaA
! gamaB = gamaB/betaB
! gamaC = gamaC/betaC
call ordena_omega(omegaA,omegaB,omegaC,gamaA,gamaB,gamaC,x0,x1,x2,f0,f1,f2)
omega_old = x2
do i=1, max_it
write(51,'(6(f12.9,1x))')x0,x1,x2,abs(f0),abs(f1),abs(f2)
! Inverte a Matriz
mG(1,1)= (x0-x2)**2
mG(1,2)= x0-x2
mG(1,3)= 1.0d0
mG(2,1)= (x1-x2)**2
mG(2,2)= x1-x2
mG(2,3)= 1.0d0
mG(3,1)= 0.0d0
mG(3,2)= 0.0d0
mG(3,3)= 1.0d0
call inversa3x3(mG,mH)
! Calculo de a, b e c
a = mH(1,1)*f0 + mH(1,2)*f1 + mH(1,3)*f2
b = mH(2,1)*f0 + mH(2,2)*f1 + mH(2,3)*f2
c = mH(3,1)*f0 + mH(3,2)*f1 + mH(3,3)*f2
! Calculo da raiz
if(abs(b+sqrt(b**2 - 4.0d0*a*c)).gt.abs(b-sqrt(b**2 - 4.0d0*a*c)))then
p = x2 - 2.0d0*c/(b+sqrt(b**2 - 4.0d0*a*c))
else
p = x2 - 2.0d0*c/(b-sqrt(b**2 - 4.0d0*a*c))
endif
omega_raiz = p
teste_omega = abs(omega_raiz - omega_old)/abs(omega_old)
write(52,'(5(f12.9,1x))')dble(omega_raiz),dimag(omega_raiz),dble(omega_old),dimag(omega_old),
teste_omega
write(53,'(9(f12.4,1x))')a,b,c,abs(f0), abs(f1), abs(f2)
if(teste_omega.lt.TOL)then
print*, "reached the desired accuracy!"Table5
exit
endif
x0 = x1
x1 = x2
x2 = dble(p)
omegaA = x0
omegaB = x1
omegaC = x2
omega_old = omega_raiz
call zerilli2_rot(nn,K,rho,l,omegaA,gamaA,betaA,ns,ns2)
call zerilli2_rot(nn,K,rho,l,omegaB,gamaB,betaB,ns,ns2)
call zerilli2_rot(nn,K,rho,l,omegaC,gamaC,betaC,ns,ns2)
f0 = gamaA
f1 = gamaB
f2 = gamaC
enddo

```

```

! call zerilli2_rot(nn,K,rhob,l,dble(omega_raiz),gamaC,betaC)
fHz = 32.3105d0*dble(omega_raiz)
! Ts = 1/(32310.5d0*dimag(omega_raiz))
Ts = 1/(203012.9d0*dimag(omega_raiz))
print*, ""
print*, ""
print*, ""
print*, "f(kHz)=" , fHz
print*, ""
print*, "T(s)=" , Ts
print*, ""
print*, "Msolar=" , MM
print*, "R(km)=" , RR1
print*, ""
close(51)
close(52)
close(53)
end program Zerilli4

```

ordena_omega

Ordena os ômega calculados para a equação de Zerilli.

```

subroutine ordena_omega(omegaA,omegaB,omegaC,gamaA,gamaB,gamaC,x0,x1,x2,f0,f1,f2)
implicit none
integer, parameter :: dp= SELECTED_REAL_KIND(15,300)
real(kind= dp) :: omegaA, omegaB, omegaC, x0, x1, x2
complex(kind= dp) :: gamaA, gamaB, gamaC, f0, f1, f2
! Ordena gamaA, gamaB e gamaC
if(abs(gamaA).gt.abs(gamaB).and.abs(gamaA).gt.abs(gamaC))then
if(abs(gamaB).gt.abs(gamaC))then
x0 = omegaA
x1 = omegaB
x2 = omegaC
f0 = gamaA
f1 = gamaB
f2 = gamaC
else
x0 = omegaA
x1 = omegaC
x2 = omegaB
f0 = gamaA
f1 = gamaC
f2 = gamaB
endif
elseif(abs(gamaB).gt.abs(gamaA).and.abs(gamaB).gt.abs(gamaC))then
if(abs(gamaA).gt.abs(gamaC))then
x0 = omegaB
x1 = omegaA
x2 = omegaC
f0 = gamaB
f1 = gamaA
f2 = gamaC
else

```

```

x0 = omegaB
x1 = omegaC
x2 = omegaA
f0 = gamaB
f1 = gamaC
f2 = gamaA
endif
elseif(abs(gamaC).gt.abs(gamaA).and.abs(gamaC).gt.abs(gamaB))then
if(abs(gamaA).gt.abs(gamaB))then
x0 = omegaC
x1 = omegaA
x2 = omegaB
f0 = gamaC
f1 = gamaA
f2 = gamaB
else
x0 = omegaC
x1 = omegaB
x2 = omegaA
f0 = gamaC
f1 = gamaB
f2 = gamaA
endif
endif
end subroutine ordena_omega

```

inversa3x3

Inverte a matriz utilizada para o cálculo de ômega.

```

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
close(70)
end subroutine inversa3x3

```

tov_rot_z

Este programa expande as equações TOV e resolve o interior estelar.

```

subroutine tov_rot_z(nn,K,rhob0,l,omega,nsteps,MM,RR1)
implicit none
integer, parameter :: dp= SELECTED_REAL_KIND(15,300)
integer, parameter :: n = 2, ns=2
integer :: i, j, nsteps
real(kind= dp), parameter :: CactusPressao = 5.542683d+38, CactusDens = 6.176d+17,
cluz=299792458d+2
real(kind= dp) :: rv(0:nsteps), pv(0:nsteps), mv(0:nsteps), rhov(0:nsteps), pvl(0:nsteps), lamb(0:nsteps), gama(0:nsteps)
real(kind= dp) :: ni(0:nsteps), nil(0:nsteps), nill(0:nsteps), lambl(0:nsteps), vetord(0:nsteps), vetorp(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, denss, press, gam, rhox
real(kind= dp) :: x(0:n)
pi=4.0d0*atan(1.0d0)
rhob=rhob0
d=0.0d0
rhox=CactusDens*rhob
call gamaSLy1(rhox,press)
e=press/CactusPressao
f=0.0d0
h = (b - a)/nsteps
r=0.0d0
j=nsteps-1
rho=rhob
!!!!!!!!!!!!!!!!!!!!!!
!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)<=((0.38330E+25)/CactusPressao)) then
!print *, "found surface"
go to 50
end if
vetorp(i)=CactusPressao*x(1)
press=CactusPressao*x(1)
call gamaSLy2(denss,press)
rho= denss/CactusDens
vetord(i)=denss
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
gam=((vetord(i-1)+vetorp(i-1)/(cluz*cluz))/vetorp(i-1))*(vetorp(i-1)-vetorp(i-2))/(vetord(i-1)-vetord(i-2))
Rg=r-h
RR=Rg-(gam/(gam-1.0))*(prg/prg_linha)
RR1=RR*1.4767
!print *, "R = ", RR
delta_ni=log(1.0d0-2.0d0*x(0)/RR)-x(2)
rhob=rhob0
d=0.0d0
rhop=CactusDens*rhob
call gamaSLy1(rhop,press)
e=press/CactusPressao
f=delta_ni
rho=rhob
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))
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 *, "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
  press=CactusPressao*x(1)
  call gamaSLy2(denss,press)
  rho= denss/CactusDens
  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)
  lamb1(i)=-ni5/ni2
!print *, i, rv(i), nill(i)
  call rk4sys(n,hh,x,r,K,nn)
end do
print *, "final grid point at ",rv(j)
MM=mv(nsteps-1)
! omega=0.894*((MM/RR**3)**(0.5))
omega=sqrt(2.0d0*1*(1-1.0d0)/(2.0d0*1+1.0d0))*sqrt(MM/RR**3)
end subroutine tov_rot_z

```

rk4sys

Cria e guarda os vetores de integração.

```

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

```

xpsys

Resolve as equações TOV.

```

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), parameter :: CactusPressao = 5.542683d+38, CactusDens = 6.176d+17
real(kind= dp) :: fp,fp1,fp2,fp3,fp4,pi,rho,r,rhob,K,nn, press, denss
integer n
! x(0)=M, x(1)=p, x(2)=ni
pi=4.0d0*atan(1.0d0)
if(x(1)<=((0.38330E+25)/CactusPressao)) then
x(1) = (0.38330E+25)/CactusPressao !a estrela termina aqui
rho = 0.33030E+08/CactusDens!!VER AQUI!
go to 100
end if
press=CactusPressao*x(1)
call gamaSLy2(denss,press)
rho= denss/CactusDens
100 continue
!!print *, "rhob = ", rhob, " rho = ", rho
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

```

gamaSLy1

É a nossa equação de estado, aqui ela calcula a pressão em função da densidade.

```
subroutine gamaSLy1(rho,press)
```

```

! rho = valor da densidade a ser comparada
! gam = valor de gama obtido da tabela SLy
implicit none
integer, parameter :: dp= SELECTED_REAL_KIND(15,300)
integer i
real(kind= dp) :: rho, gam, lpress, press, l1, l2, l3, pres1, rho1
real(kind= dp) :: dens(103), gama(78), pres(103)
gam = 1.177
press=0.62150E+30
lpress=29.79344
!dens g/cm3 pres erg/cm3 converter depois para as unidades Cactus
! tabela SLy
data dens(1:103)&
/0.33030E+08,0.65920E+08,0.13150E+09,0.26250E+09,0.33050E+09,0.52390E+09,0.83030E+09,
0.10460E+10,0.12120E+10,0.16060E+10,&
0.25450E+10,0.41660E+10,0.66060E+10,0.80310E+10,0.10110E+11,0.13190E+11,0.16610E+11,
0.20910E+11,0.25330E+11,0.33150E+11,&
0.41740E+11,0.50390E+11,0.66190E+11,0.83370E+11,0.96310E+11,&
0.34951E+12,0.36883E+12,0.38650E+12,0.44199E+12,0.51080E+12,0.59119E+12,0.68224E+12,
0.78339E+12,0.89426E+12,0.10146E+13,&
0.12831E+13,0.17543E+13,0.21141E+13,0.27232E+13,0.34178E+13,0.44827E+13,0.57153E+13,
0.75106E+13,0.96148E+13,0.12593E+14,&
0.16774E+14,0.21042E+14,0.27844E+14,0.36043E+14,0.40688E+14,0.47001E+14,0.53843E+14,
0.61153E+14,0.67284E+14,0.75224E+14,&
0.81738E+14,0.88350E+14,0.95022E+14,0.10173E+15,0.10845E+15,0.11351E+15,0.11859E+15,
0.12372E+15,0.12720E+15,0.12845E+15,&
0.13038E+15,0.13531E+15,0.14381E+15,0.15232E+15,0.16935E+15,0.18641E+15,0.20350E+15,
0.22063E+15,0.27223E+15,0.32424E+15,&
0.37675E+15,0.42983E+15,0.48358E+15,0.53808E+15,0.59340E+15,0.64963E+15,0.70684E+15,
0.76510E+15,0.82450E+15,0.88509E+15,&
0.94695E+15,0.10102E+16,0.10748E+16,0.11408E+16,0.12085E+16,0.12777E+16,0.13486E+16,
0.14706E+16,0.15977E+16,0.17302E+16,&
0.18683E+16,0.20123E+16,0.21624E+16,0.24820E+16,0.28289E+16,0.32048E+16,0.36113E+16,
0.40498E+16/
data pres(1:103)&
/0.38330E+25,0.10060E+26,0.26040E+26,0.66760E+26,0.87380E+26,0.16290E+27,0.30290E+27,
0.41290E+27,0.50360E+27,0.68600E+27,&
0.12720E+28,0.23560E+28,0.43620E+28,0.56620E+28,0.77020E+28,0.10480E+29,0.14250E+29,
0.19380E+29,0.25030E+29,0.34040E+29,&
0.46280E+29,0.59490E+29,0.80890E+29,0.11000E+30,0.14500E+30,&
0.62150E+30,0.64304E+30,0.65813E+30,0.69945E+30,0.74685E+30,0.80149E+30,0.86443E+30,
0.93667E+30,0.10191E+31,0.11128E+31,&
0.13370E+31,0.17792E+31,0.21547E+31,0.28565E+31,0.37461E+31,0.52679E+31,0.72304E+31,
0.10405E+32,0.14513E+32,0.20894E+32,&
0.30720E+32,0.41574E+32,0.60234E+32,0.84613E+32,0.99286E+32,0.12023E+33,0.14430E+33,
0.17175E+33,0.19626E+33,0.23024E+33,&
0.26018E+33,0.29261E+33,0.32756E+33,0.36505E+33,0.40509E+33,0.43681E+33,0.46998E+33,
0.50462E+33,0.52856E+33,0.53739E+33,&
0.53739E+33,0.58260E+33,0.66828E+33,0.76443E+33,0.99146E+33,0.12701E+34,0.16063E+34,
0.19971E+34,0.35927E+34,0.59667E+34,&
0.92766E+34,0.13668E+35,0.19277E+35,0.26235E+35,0.34670E+35,0.44702E+35,0.56451E+35,
0.70033E+35,0.85561E+35,0.10315E+36,&
0.12289E+36,0.14491E+36,0.16930E+36,0.19616E+36,0.22559E+36,0.25769E+36,0.29255E+36,
0.35702E+36,0.42981E+36,0.51129E+36,&
0.60183E+36,0.70176E+36,0.81139E+36,0.10609E+37,0.13524E+37,0.16876E+37,0.20679E+37,
0.24947E+37/

```



```

if(rho < dens(1)) then
press=0.38330E+25
write(*,fmt='(a,e11.5,a,f5.3)') "ATENCAO! ERRO?! rho menor do que PRIMEIRO valor da tabela! rho = ",rho, " press = ",press
go to 100
end if
do i=1,103
if(rho <= dens(i)) then
rho1 = log10(rho)
!!press=logP
l1= log10(rho)-log10(dens(i-1))
l2= log10(pres(i))-log10(pres(i-1))
l3= log10(dens(i))-log10(dens(i-1))
lpress= (l1*l2)/l3 + log10(pres(i-1))
press = 10**(lpress)
pres1=pres(i)
go to 100
end if
end do
100 continue
if(i==103) then
l1= log10(rho)-log10(dens(102))
l2= log10(pres(103))-log10(pres(102))
l3= log10(dens(103))-log10(dens(102))
lpress= (l1*l2)/l3 + log10(pres(102))
press = 10**(lpress)
end if
if(i == 104) then
press=pres(103)
write(*,fmt='(a,e11.5,a,f5.3)') "ATENCAO! ERRO?! rho alcançou o FINAL da tabela! rho = ",rho, " press = ",press
end if
end subroutine gamaSLy1

```

gamaSLy2

É nossa equação de estado, aqui ela calcula a densidade em função da pressão.

```

subroutine gamaSLy2(denss,press)!esta subrotina calcula densidade em função da pressão
! rho = valor da densidade a ser comparada
! gam = valor de gama obtido da tabela SLy
implicit none
integer, parameter :: dp= SELECTED_REAL_KIND(15,300)
integer i
real(kind= dp) :: rho, gam, ldens, press, l1, l2, l3, lpres, denss, rho1
real(kind= dp) :: dens(103), pres(103)
gam = 1.177
!dens g/cm3 pres erg/cm3 converter depois para as unidades Cactus
! tabela SLy
data dens(1:103)&
/0.33030E+08,0.65920E+08,0.13150E+09,0.26250E+09,0.33050E+09,0.52390E+09,0.83030E+09,
0.10460E+10,0.12120E+10,0.16060E+10,&
0.25450E+10,0.41660E+10,0.66060E+10,0.80310E+10,0.10110E+11,0.13190E+11,0.16610E+11,
0.20910E+11,0.25330E+11,0.33150E+11,&
0.41740E+11,0.50390E+11,0.66190E+11,0.83370E+11,0.96310E+11,&

```

```

0.34951E+12,0.36883E+12,0.38650E+12,0.44199E+12,0.51080E+12,0.59119E+12,0.68224E+12,
0.78339E+12,0.89426E+12,0.10146E+13,&
0.12831E+13,0.17543E+13,0.21141E+13,0.27232E+13,0.34178E+13,0.44827E+13,0.57153E+13,
0.75106E+13,0.96148E+13,0.12593E+14,&
0.16774E+14,0.21042E+14,0.27844E+14,0.36043E+14,0.40688E+14,0.47001E+14,0.53843E+14,
0.61153E+14,0.67284E+14,0.75224E+14,&
0.81738E+14,0.88350E+14,0.95022E+14,0.10173E+15,0.10845E+15,0.11351E+15,0.11859E+15,
0.12372E+15,0.12720E+15,0.12845E+15,&
0.13038E+15,0.13531E+15,0.14381E+15,0.15232E+15,0.16935E+15,0.18641E+15,0.20350E+15,
0.22063E+15,0.27223E+15,0.32424E+15,&
0.37675E+15,0.42983E+15,0.48358E+15,0.53808E+15,0.59340E+15,0.64963E+15,0.70684E+15,
0.76510E+15,0.82450E+15,0.88509E+15,&
0.94695E+15,0.10102E+16,0.10748E+16,0.11408E+16,0.12085E+16,0.12777E+16,0.13486E+16,
0.14706E+16,0.15977E+16,0.17302E+16,&
0.18683E+16,0.20123E+16,0.21624E+16,0.24820E+16,0.28289E+16,0.32048E+16,0.36113E+16,
0.40498E+16/
data pres(1:103)&
/0.38330E+25,0.10060E+26,0.26040E+26,0.66760E+26,0.87380E+26,0.16290E+27,0.30290E+27,
0.41290E+27,0.50360E+27,0.68600E+27,&
0.12720E+28,0.23560E+28,0.43620E+28,0.56620E+28,0.77020E+28,0.10480E+29,0.14250E+29,
0.19380E+29,0.25030E+29,0.34040E+29,&
0.46280E+29,0.59490E+29,0.80890E+29,0.11000E+30,0.14500E+30,&
0.62150E+30,0.64304E+30,0.65813E+30,0.69945E+30,0.74685E+30,0.80149E+30,0.86443E+30,
0.93667E+30,0.10191E+31,0.11128E+31,&
0.13370E+31,0.17792E+31,0.21547E+31,0.28565E+31,0.37461E+31,0.52679E+31,0.72304E+31,
0.10405E+32,0.14513E+32,0.20894E+32,&
0.30720E+32,0.41574E+32,0.60234E+32,0.84613E+32,0.99286E+32,0.12023E+33,0.14430E+33,
0.17175E+33,0.19626E+33,0.23024E+33,&
0.26018E+33,0.29261E+33,0.32756E+33,0.36505E+33,0.40509E+33,0.43681E+33,0.46998E+33,
0.50462E+33,0.52856E+33,0.53739E+33,&
0.53739E+33,0.58260E+33,0.66828E+33,0.76443E+33,0.99146E+33,0.12701E+34,0.16063E+34,
0.19971E+34,0.35927E+34,0.59667E+34,&
0.92766E+34,0.13668E+35,0.19277E+35,0.26235E+35,0.34670E+35,0.44702E+35,0.56451E+35,
0.70033E+35,0.85561E+35,0.10315E+36,&
0.12289E+36,0.14491E+36,0.16930E+36,0.19616E+36,0.22559E+36,0.25769E+36,0.29255E+36,
0.35702E+36,0.42981E+36,0.51129E+36,&
0.60183E+36,0.70176E+36,0.81139E+36,0.10609E+37,0.13524E+37,0.16876E+37,0.20679E+37,
0.24947E+37/
if(press<=0) then
denss=0
go to 100
end if
if(press < pres(1)) then
denss=dens(1)
go to 100
write(*,fmt='(a,e11.5,a,f5.3)') "ATENCAO! ERRO?! press menor do que PRIMEIRO valor da tabela! pres = ",press, " dens = ",denss
end if
do i=1,103
if(press <= pres(i)) then
rho1 = dens(i)
!ldens=logrho
lpres=log10(press)
l1= log10(press)-log10(pres(i-1))
l2= log10(dens(i))-log10(dens(i-1))
l3= log10(pres(i))-log10(pres(i-1))
ldens= (l1*l2)/l3 + log10(dens(i-1))

```

```

denss = 10**(ldens)
!pres1=pres(i)
go to 100
end if
end do
100 continue
!if(i==103) then
! l1= log10(press)-log10(pres(102))
! l2= log10(dens(103))-log10(dens(102))
! l3= log10(pres(103))-log10(pres(102))
! ldens= (l1*l2)/l3 + log10(dens(102))
! denss = 10**(ldens)
!end if
if(i == 104) then
denss = dens(103)
print*, "ATENCAO! ERRO?! pres alcançou o FINAL da tabela! pres = ",press, " dens = ",denss
end if
end subroutine gamaSLy2

```

zerilli2_rot

```

Calcula gama e ômega.
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,'(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

```

zerilli_rot

Continua com o calculo de gama e ômega.

```

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, H0R, KR, gR, hR, kRp, ZRt, ZRtl, r, rt, nR, pi, v2m, v3m, rm, rtm, v5, v6
real(kind= dp) :: Zrtt, Zrttl, H1R, niR, K1R, H1lR, 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
!, ns=5000
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), Zll(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,H0R,KR,H1R,niR,K1R,H1lR,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 Matrices
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)*H0R+mD(1,2)*KR
ZRtl=mD(2,1)*H0R+mD(2,2)*KR
! Intervalo de integracao para a eq. de Zerilli
rmin=RR
rmax=25.0d0/omega
! rmax=50.0d0/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)
v4=Vtv(0)-omega**2
Zll(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(1)
! Ztv1(i)=x(0)
Ztv(i)=x(0)
Ztv1(i)=x(1)
rv(i)=r
rtv(i)=rt
Zll(i) = v6*x(1)
v4=v6
! Zmax=x(1)
! Zlmax=x(0)
Zmax=x(0)
Zlmax=x(1)
rmax1=r
rtmax1=rt
end do
do i=0, ns
write(31,fmt='(6(f15.10,lx))') Ztv(i), Ztv1(i), Vtv(i), rv(i), rtv(i), Zll(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)
H0v(i)=mE(1,1)*Ztv(i)+mE(1,2)*Ztv(i)
Kv(i) =mE(2,1)*Ztv(i)+mE(2,2)*Ztv(i)
write(32,fmt='(3(f15.10,1x))') rv(i), H0v(i), Kv(i)
end do
close(31)
close(32)
end subroutine zerilli_rot

```

constantes_rot

Guarda todos os valores da matriz para o cálculo de gama e ômega.

```

subroutine constantes_rot(nn,K,l,rhob,RR,MR,omega,H0R,KR,H1R,niR,KlR,H1lR,nilR,nsteps,nsteps2)
implicit none
integer, parameter :: n1=3
!nsteps = 5000, nsteps2=10000,
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,H0R,f4,f5,f6,f7,f8
real(kind= dp) :: MR,pR,niR,lbR,RR,KlR,H1lR,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+1.0d0)*(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
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
H0R=(f5*XR-f6*H1R+f8*KR)/f4
! do i=0, nsteps2
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

```

s_m_rot

Aqui são calculadas as perturbações de $R/2$ até R .

```

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
!, ns = 5000, ns2 = 10000
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, gam
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), gama(0:ns2)
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,lb1,ns2,ns3,RR,gama)
! d=H10, e=K0, f=W0, g=X0 são as condições iniciais
! x(0)=H1, x(1)=K, x(2)=W, x(3)=X
! Laço 1 (obter Y1)
d=1.0d0! aqui igual a 1 é uma das condições na tese do patrick
e=0.0d0
f=0.0d0
g=0.0d0
x = (/d, e, f, g/)
h1=-hh
r=RR
q=ns2-1
h1v1(0)=d !condição inicial das equações
kv1(0)=e
wv1(0)=f
xv1(0)=g
rv1(0)=RR
h1v1l(0)=0.0d0 !condição inicial das derivadas das equações
kv1l(0)=0.0d0
wv1l(0)=0.0d0
xv1l(0)=0.0d0
M2=mv(ns2-1) !massa !dando valor aos escalares
p2=pv(ns2-1) !pressão
rho2=rhov(ns2-1) !densidade
pl=pvl(ns2-1) !derivada da pressão
ni2=ni(ns2-1) !potencial
nil2=nil(ns2-1) !1 derivada potencial
nill2=nill(ns2-1)!2 derivada potencial
lb1=lb(ns2-1) !lambda
lb1l=lb1(ns2-1) !derivada lambda
gam=gama(ns2-1) !gama
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 !5000
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,omega,l,Y1,gam)
!q=ns2-1=9999 aqui a iteração decresce, ela vai da superfície até o meio
M2=mv(q)
p2=pv(q)
rho2=rhov(q)
pl=pvl(q)

```

```

ni2=ni(q)
nil2=nil(q)
nill2=nill(q)
lb1=lb(q)
lbl1=lbl(q)
gam=gama(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)
lbl1n=lbl(q-1)
r=RR-i*hh
h1v1(i)=x(0)!aqui x devolve os valores das 4 equações bizarras originais
kv1(i)=x(1)
wv1(i)=x(2)
xv1(i)=x(3)
rv1(i)=r
h1v1l(i)=Yl(0)!aqui são suas derivadas
kv1l(i)=Yl(1)
wv1l(i)=Yl(2)
xv1l(i)=Yl(3)
mF1(0)=h1v1(i)!o que faz este vetor?
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!aqui igual a 1 é uma das condições na tese do patrick
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)
lbl1=lbl(ns2-1)
gam=gama(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)
lbl1n=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,omega,l,Y1,gam)
!do meio até a superficie
M2=mv(q)!é tudo igual ao laço 1
p2=pv(q)
rho2=rhov(q)
pl=pvl(q)
ni2=ni(q)
nil2=nil(q)
nill2=nill(q)
lb1=lb(q)
lbl1=lbl(q)
gam=gama(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)
lbl1n=lbl(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
h1v2l(i)=Y1(0)
kv2l(i)=Y1(1)
wv2l(i)=Y1(2)

```

```

xv2l(i)=Yl(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))') h1v2l(i), kv2l(i), wv2l(i), xv2l(i), rv2l(i)
end do
! Lago 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
h1v3l(0)=0.0d0
kv3l(0)=0.0d0
wv3l(0)=0.0d0
xv3l(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)
lbl1=lbl(ns2-1)
gam=gama(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)
lbl1n=lbl(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,omega,l,Y1,gam)
M2=mv(q)
p2=pv(q)
rho2=rhov(q)
pl=pvl(q)

```

```

ni2=ni(q)
nil2=nil(q)
nill2=nill(q)
lb1=lb(q)
lbl1=lbl(q)
gam=gama(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)
lbl1n=lbl(q-1)
r=RR-i*hh
h1v3(i)=x(0)
kv3(i)=x(1)
wv3(i)=x(2)
xv3(i)=x(3)
rv3(i)=r
h1v3l(i)=Yl(0)
kv3l(i)=Yl(1)
wv3l(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), rv3(i)
end do
close(20)
close(21)
close(22)
close(23)
close(24)
close(25)
end subroutine s_m_rot

```

tov_rot_z4

Aqui calculamos a estrela para o programa s_m_rot.

```

subroutine tov_rot_z4(nn,K,rhob0,hh,mv,pv,rhov,pvl,ni,nil,nill,lamb,lambl,nsteps,ns,RR,gama)
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), gama(0:nsteps)
real(kind= dp) :: ni(0:nsteps), nil(0:nsteps), nill(0:nsteps), lambl(0:nsteps),
vetord(0:nsteps), vetorp(0:nsteps)

```

```

real(kind= dp), parameter :: a = 0.0d0, b =20.0d0
real(kind= dp), parameter :: CactusPressao = 5.542683d+38, CactusDens = 6.176d+17,
cluz=299792458d+2
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, denss, press, gam, rhox
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")
open(31,file="gamaz4.dat")
pi=4.0d0*atan(1.0d0)
rhob=rhob0
d=0.0d0
rhox=CactusDens*rhob
call gamaSLy1(rhox,press)
e=press/CactusPressao
f=0.0
h = (b - a)/nsteps
r=0.0d0
j=nsteps-1
rho=rhob
!!!!!!!!!!!!!!!!!!!!!!
!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)<=((0.38330E+25)/CactusPressao)) then
    !print *, "found surface"
  go to 50
  end if
  vetorp(i)=CactusPressao*x(1)
  press=CactusPressao*x(1)
  call gamaSLy2(denss,press)
  rho= denss/CactusDens
  vetord(i)=denss
  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
  gam=((vetord(i-1)+vetorp(i-1)/(cluz*cluz))/vetorp(i-1))
*(vetorp(i-1)-vetorp(i-2))/(vetord(i-1)-vetord(i-2))
  Rg=r-h
  RR=Rg-(gam/(gam-1.0))*(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
  rhox=CactusDens*rhob
  call gamaSLy1(rhox,press)
  e=press/CactusPressao
  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))
    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 *, "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
  press=CactusPressao*x(1)
  call gamaSLy2(denss,press)
  rho= denss/CactusDens
  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-(2.0d0*ni1/r)*ni3/ni2-ni1*ni3*ni5/ni2**2
  lamb(i)=-log(ni2)
  lambl(i)=-ni5/ni2
  ! !print *, i, rv(i), nill(i)
  call rk4sys(n,hh,x,r,K,nn)
end do
do i=ns-1, j
  gama(i) = ((CactusDens*rhov(i)+ (CactusPressao/(cluz*cluz))*pv(i))/pv(i))*((pv(i)-pv(i-1))/
((CactusDens)*(rhov(i)-rhov(i-1))))
  gama(0)=gama(2)
  gama(1)=gama(2)
  gama(9999)=gama(9998)
  if (rhov(i) == rhov(i-1)) then
    gama(i) = gama(i-1)
  end if
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)
  write(31,fmt='(3(e12.5,1x))') rv(i), gama(i), rhov(i)
end do
close(10)
close(14)
close(16)

```

```

close(18)
close(11)
close(13)
close(15)
close(17)
close(19)
close(31)
end subroutine tov_rot_z4

```

rk4sys_p1

Calcula e guarda todos os vetores e matrizes para a rotina s_m_rot.

```

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,l,Yl,gam)
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, gam
! real(kind= dp) :: H1l, Kl, Wl, Xl
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,gam)
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,lb1lx,M2x,p2x,rho2x,plx,omega,l,gam)
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,lb1lx,M2x,p2x,rho2x,plx,omega,l,gam)
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,lb1ln,M2n,p2n,rho2n,pln,omega,l,gam)
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
Yl(0)= f(0,1)
Yl(1)= f(1,1)
Yl(2)= f(2,1)
Yl(3)= f(3,1)

```

```

! Y1(0)=H1l, Y1(1)=Kl, Y1(2)=Wl,Y1(3)=Xl
! H1l=(1.0/6.0)* (f(1,1) + 2.0*(f(1,2) + f(1,3)) + f(1,4))
! Kl=(1.0/6.0)* (f(2,1) + 2.0*(f(2,2) + f(2,3)) + f(2,4))
! Wl=(1.0/6.0)* (f(3,1) + 2.0*(f(3,2) + f(3,3)) + f(3,4))
! Xl=(1.0/6.0)* (f(4,1) + 2.0*(f(4,2) + f(4,3)) + f(4,4))
end subroutine rk4sys_p1

```

xpsys_p1

Apresenta os cálculos complementares para a rotina rk4sys_p1.

```

subroutine xpsys_p1(n,x,f,r,nn,ni,nil,nill,lamb,lambl,M,p,rho,pl,omega,l,gam)
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, gam
real(kind= dp) :: l, p, M, rho, ni, nil, nill, lamb, lambl, 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*1*(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*pl*(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
f17= ((1/gam)*(exp(-0.5*ni)))/p
f18= 2.0*f0/(r**2)
! f19= (rho+p)*(exp(0.5*lamb))
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*lamb1*nil+nill)
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.pl

```

c_m_rot

Aqui são calculadas as perturbações de 0 até $R/2$.

```

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, lb1l, M2, p2, rho2, pl
real(kind= dp) :: ni2n, nil2n, nill2n, lb1n, lb1ln, M2n, p2n, rho2n, pln
real(kind= dp) :: H1l, Kl, Wl, Xl
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), lb1(0:ns2)
real(kind= dp) :: ni(0:ns2), nil(0:ns2), nill(0:ns2), pvl(0:ns2), gama(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), Yl(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, gam
!omega vem desde o início do programa em tov_rot.z
pi=4.0*atan(1.0)
!calcula do meio até o núcleo
call P_C(nn,K,rhob,l,hh,omega,mv,pv,rhov,pvl,ni,nil,nill,lb,lb1,H14,K4,W4,
X4,H15,K5,W5,X5,n3,ns2,rmin,gama)
open(10,file="TestY4t9.dat")
open(11,file="TestY5t9.dat")
open(12,file="TestY4lt9.dat")
open(13,file="TestY5lt9.dat")
! open(24,file="TestY1lr.dat")
! d=H10, e=K0, f=W0, g=X0
! x(0)=H1, x(1)=K, x(2)=W, x(3)=X
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Laço 1 (obter Y4)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
do i=0, n3-1 !definindo condições iniciais
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
! h1v4l(i)=r*H6
! kv4l(i)=r*K6
! wv4l(i)=r*W6
! xv4l(i)=r*X6
end do
!print *, "r = ", r, i
d=H14(n3-1)!condições iniciais
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)
lbl1=lbl(n3-1)
gam=gama(n3-1)!por aqui o gama!
M2n=mv(n3)
p2n=pv(n3)
rho2n=rhov(n3)
pln=pvl(n3)
ni2n=ni(n3)
nil2n=nil(n3)
nill2n=nill(n3)
lb1n=lb(n3)
lbl1n=lbl(n3)
j=ns-1
do i=n3, j
call rk4sys_p1(n1,hh,x,r,nn,ni2,nil2,nill2,lb1,lbl1,M2,p2,rho2,pl,ni2n,nil2n,nill2n,lb1n,lbl1n,
M2n,p2n,rho2n,pln,omega,l,Y1,gam)
M2=mv(i)
p2=pv(i)
rho2=rhov(i)
pl=pvl(i)
ni2=ni(i)
nil2=nil(i)
nill2=nill(i)
lb1=lb(i)
lbl1=lbl(i)
gam=gama(i)!colocar gama aqui
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)
lbl1n=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
! h1v4l(i)=r*H6
! kv4l(i)=r*K6
! wv4l(i)=r*W6
! xv4l(i)=r*X6
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)
lbl1=lbl(n3-1)
gam=gama(n3-1)!colocar gama aqui
M2n=mv(n3)
p2n=pv(n3)
rho2n=rhov(n3)
pln=pvl(n3)
ni2n=ni(n3)
nil2n=nil(n3)
nill2n=nill(n3)
lb1n=lb(n3)
lbl1n=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,gam)
  M2=mv(i)
  p2=pv(i)
  rho2=rhov(i)
  pl=pvl(i)
  ni2=ni(i)
  nil2=nil(i)
  nill2=nill(i)
  lb1=lb(i)
  lbl1=lbl(i)
  gam=gama(i)!colocar gama aqui
  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)
  lbl1n=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

```

P_C

Aqui calcula-se e guarda-se as matrizes necessárias para a rotina c_m_rot.

```

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,gama)
implicit none
integer, parameter :: dp= SELECTED_REAL_KIND(15,300)
! integer, parameter :: nsteps2=100000
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 !é uma matriz
real(kind= dp) :: pv(0:ns2), mv(0:ns2), rhov(0:ns2), lb(0:ns2), lbl(0:ns2), gama(0:ns2)
real(kind= dp) :: ni(0:ns2), nil(0:ns2), nill(0:ns2), pvl(0:ns2)
! real(kind= dp) :: H4, K4, W4, X4, H5, K5, W5, X5
real(kind= dp) :: H10l1, K0l1, W0l1, X0l1, H101, K01, W01, X01
real(kind= dp) :: H10l2, K0l2, W0l2, X0l2, H102, K02, W02, X02
real(kind= dp) :: rmin, gam
pi=4.0d0*atan(1.0d0)
!se eu entendi corretamente, aqui eu usarei o gama(0)
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,gama)
!(nn,K,rhob0,hh,p0,p2,p4,ni0,ni2,ni44,rho2,mv,pv,rhov,pvl,ni,nil,nill,lamb,lambl,nsteps,ns,rmin,gam)
rho0=rhob0
gam=gama(0)
! !print *, p0,p2,p4,ni0,ni2,ni4,rho0,rho2,l,omega
call matriz(mH,nn,p0,p2,p4,ni0,ni2,ni4,rho0,rho2,omega,l,gam)
! 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)/l)*W01+0.5d0*K01)
H101=(2.0d0*1*K01+16.0d0*pi*(rho0+p0)*W01)/(l*(1+1.0d0))
H10l1=mH(1,1)*H101+mH(1,2)*K01+mH(1,3)*W01+mH(1,4)*X01
K0l1=mH(2,1)*H101+mH(2,2)*K01+mH(2,3)*W01+mH(2,4)*X01
W0l1=mH(3,1)*H101+mH(3,2)*K01+mH(3,3)*W01+mH(3,4)*X01
X0l1=mH(4,1)*H101+mH(4,2)*K01+mH(4,3)*W01+mH(4,4)*X01
! !print*, "H=", H10l1,"K=", K0l1,"W=", W0l1,"X=", X0l1
! H4=H10l1
! K4=K0l1
! W4=W0l1
! X4=X0l1
j=ns-1
r=0.0d0

```



```

do i=0, j
  H11(i)=H101
  K1(i)=K01
  W1(i)=W01
  X1(i)=X01
  ! r=i*hh
  ! H11(i)=H101+0.5*H1011*r**2
  ! K1(i)=K01+0.5*K011*r**2
  ! W1(i)=W01+0.5*W011*r**2
  ! X1(i)=X01+0.5*X011*r**2
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)/l)*W02+0.5d0*K02)
H102=(2.0d0*1*K02+16.0d0*pi*(rho0+p0)*W02)/(l*(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
  ! r=i*hh
  ! H12(i)=H102+0.5*H10112*r**2
  ! K2(i)=K02+0.5*K0112*r**2
  ! W2(i)=W02+0.5*W0112*r**2
  ! X2(i)=X02+0.5*X0112*r**2
end do
end subroutine P_C

```

tov_rot_z3

Aqui calcula-se a estrela necessária para a rotina c_m_rot e suas auxiliares.

```

subroutine tov_rot_z3(nn,K,rhob0,hh,p0,p2,p4,ni0,ni22,ni44,rho2,mv,pv,rhov,pvl,
ni,nil,nill,lamb,lambl,nsteps,ns,rmin,gama)
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), gama(0:nsteps)
real(kind= dp) :: ni(0:nsteps), nil(0:nsteps), nill(0:nsteps), lambl(0:nsteps),
vetord(0:nsteps), vetorp(0:nsteps)
real(kind= dp), parameter :: a = 0.0d0, b =20.0d0

```

```

real(kind= dp), parameter :: CactusPressao = 5.542683d+38, CactusDens = 6.176d+17,
cluz=299792458d+2
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, denss, press, gam, rhox
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")
open(31,file="gamaz3.dat")
pi=4.0d0*atan(1.0d0)
rhob=rhob0
d=0.0d0
rhox=CactusDens*rhob
call gamaSLy1(rhox,press)
e=press/CactusPressao
f=0.0
h = (b - a)/nsteps
r=0.0d0
j=nsteps-1
rho=rhob
!!!!!!!!!!!!!!!!!!!!!!
!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)<=((0.38330E+25)/CactusPressao)) then
    !print *, "found surface"
  go to 50
  end if
  vetorp(i)=CactusPressao*x(1)
  press=CactusPressao*x(1)
  call gamaSLy2(denss,press)
  rho= denss/CactusDens
  vetord(i)=denss
  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
  gam=((vetord(i-1)+vetorp(i-1)/(cluz*cluz))/vetorp(i-1))*(vetorp(i-1)-vetorp(i-2))
  /(vetord(i-1)-vetord(i-2))
  Rg=r-h
  RR=Rg-(gam/(gam-1.0))*(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
  rhox=CactusDens*rhob
  call gamaSLy1(rhox,press)
  e=press/CactusPressao
  f=delta_ni
  rho=rhob
  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))
    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 *, "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
  press=CactusPressao*x(1)
  call gamaSLy2(denss,press)
  rho= denss/CactusDens
  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
  ! !print *, i, rv(i), nill(i)
  call rk4sys(n,hh,x,r,K,nn)
end do
do i=ns-1, j
  gama(i) = ((CactusDens*rhov(i)+ (CactusPressao/(cluz*cluz))*pv(i))/pv(i))*((pv(i)-pv(i-1))/
((CactusDens)*(rhov(i)-rhov(i-1))))
  gama(0)=gama(1)
  gama(9999)=gama(9998)
  if (rhov(i) == rhov(i-1)) then
    gama(i) = gama(i-1)
  end if
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)
  write(31,fmt='(3(e12.5,1x))') rv(i), gama(i), rhov(i)
end do
close(10)
close(14)
close(16)
close(18)

```

```

close(11)
close(13)
close(15)
close(17)
close(19)
close(31)
end subroutine tov_rot_z3

```

matriz

Aqui são calculados e guardados os vetores e matrizes para a rotina P_C.

```

subroutine matriz(mH,nn,p0,p2,p4,ni0,ni2,ni4,rho0,rho2,omega,l,gam)
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, gam
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)
! l=2
! !print *, p0,p2,p4,ni0,ni2,ni4,rho0,rho2,l,omega
!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*(gam))
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

```

inversa

Inverte a matriz utilizada na rotina anterior.

```

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))
! c32=-mF(1,1)*(mF(2,3)*mF(4,4)-mF(2,4)*mF(4,1))+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))
! c43=-mF(1,1)*(mF(2,2)*mF(3,4)-mF(2,4)*mF(3,3))+mF(1,2)*(mF(2,1)*mF(3,4)-mF(2,4)*mF(3,1))-
! mF(1,4)*(mF(2,1)*mF(3,3)-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

```