

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E
COMPUTAÇÃO CIENTÍFICA
DEPARTAMENTO DE MATEMÁTICA APLICADA

DISSERTAÇÃO DE MESTRADO

Soluções de Weyl:
Aspectos Globais, Singularidades
e Geodésicas

9913791

Autor: LUIS ALBERTO D'AFONSECA

Orientador: PROF. DR. PATRÍCIO A. LETELIER SOTOMAYOR

UNICAMP
BIBLIOTECA CENTRAL

UNIDADE	BC
N.º CHAMADA:	
V.	Ex.
TOMBO BC/	38079
PROC.	229/99
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREÇO	11,00
DATA	17-07-99
N.º CPD	

CM-00125859-1

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**

D'Afonseca, Luis Alberto

D131s Soluções de Weyl: propriedades globais, singularidades e geodésicas / Luis Alberto D'Afonseca – Campinas, [S.P. :s.n.], 1999.

Orientador : Patricio A. Letelier Sotomayor

Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1. Relatividade geral (Física). 2. Singularidades (Matemática). 3. Física matemática. I. Sotomayor, Patricio A. Letelier. II. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. III. Título.

Soluções de Weyl: Propriedades Globais, Singularidades e Geodésicas

Este exemplar corresponde à redação final da dissertação devidamente corrigida e defendida por **Luis Alberto D'Afonseca** e aprovada pela comissão julgadora.

Campinas, 20 de Abril de 1999



Prof. Dr. Patrício A. Letelier Sotomayor
Orientador

Banca Examinadora:

1. Prof. Dr. Patricio Aníbal Letelier Sotomayor
2. Prof. Dr. Samuel Rocha de Oliveira
3. Prof. Dr. George Emanuel Avraam Matsas

Dissertação apresentada ao Instituto de Matemática, Estatística e Computação Científica, UNICAMP, Como requisito parcial para a obtenção do Título de MESTRE em Matemática Aplicada.

Tese de Mestrado defendida em 20 de abril de 1999 e aprovada

Pela Banca Examinadora composta pelos Profs. Drs.



Prof (a). Dr (a). PATRÍCIO ANÍBAL LETELIER SOTOMAYOR



Prof (a). Dr (a). SAMUEL ROCHA DE OLIVEIRA



Prof (a). Dr (a). GEORGE EMANUEL AVRAAM MATSAS

Aos Meus Pais

Por serem um exemplo
a ser seguido.

Agradecimentos

Seria impossível listar todos que merecem ser citados, portanto começo com um agradecimento geral, a todos os que direta ou indiretamente contribuíram para a conclusão dessa dissertação.

Devo agradecer ao meu orientador, professor Letelier, idealizador do projeto.

Aos colegas pela parceria, por todo esse tempo. Entre os colegas aqueles que posso chamar de amigos: Cantão, Portugal, Biloti, Ernesto e Luiz.

Cantão e Portugal também devem ser agradecidos pela leitura dos manuscritos.

Ernesto por ceder suas bibliotecas para diferenciação automática.

Agradeço a FAPESP (Fundação de Amparo a Pesquisa do Estado de São Paulo) pelo suporte financeiro, que viabilizou este projeto, e pelas críticas aos relatórios parciais, que indicaram erros a serem corrigidos e pontos a serem melhorados.

Fora do ambiente acadêmico, a todos aqueles cujo incentivo foi indispensável. Em especial aos amigos: José, Ana Clara, Lucia, Renata e Ildelfonso e ao professor Pedro Russo, embora nunca tenha sido seu aluno.

Sumário

Notações	5
Resumo	7
Abstract	9
1 Introdução	11
2 Métricas de Weyl	13
2.1 Equações de Weyl	13
2.2 Geodésicas nas Coordenadas de Weyl	17
2.3 Coordenadas Especiais	19
3 Singularidades	23
3.1 Introdução	23
3.2 Invariantes Escalares	23
3.3 Singularidades Intermediárias	24
3.4 Horizontes nas Métricas de Weyl	25
3.5 Singularidades Estruturais	25
3.6 Extensões Maximais	26
4 Soluções de Weyl	35
4.1 Introdução	35
4.2 Campo Homogêneo	36
4.3 Solução de Chazy-Curzon	36
4.4 Barra de Comprimento Finito	38
4.5 Solução de Schwarzschild	38
4.6 Linha Infinita	40
4.7 Linha Semi-Infinita	40
4.8 Discos	43
4.9 Solução de Appell	51

4.10	Anéis	54
4.11	Soluções Multipolares	57
4.12	Soluções em Sistemas R-Separáveis	60
4.13	Soluções de Godfrey	63
4.14	Métricas de Kinnersley	64
5	Soluções Sobrepostas	65
5.1	Sobreposição de Partículas de Curzon	65
5.2	Barras Sobrepostas	66
5.3	Solução Geral para Dois Corpos	67
5.4	Discos Circundando Buracos Negros	68
5.5	Partícula de Curzon e Disco de Appell	70
5.6	Sobreposições com o Anel de Letelier-Oliveira	71
5.7	Sobreposições com o Anel de Weyl-Bach	73
5.8	Corpos Acelerados	76
5.9	A Métrica C	78
6	Conclusões	87
	Bibliografia	89
A	Teoremas Sobre Singularidades	93
A.1	Estrutura Causal	93
A.2	Orientação Temporal	93
A.3	Pontos Extremos e Curvas Inextensíveis	95
A.4	Conceitos de Estabilidade	96
A.5	Congruências Geodésicas	98
A.6	Pontos Conjugados	100
A.7	Curvas de Comprimento Máximo	103
A.8	Teoremas Sobre Singularidades	104
B	Sistemas de Coordenadas	107
B.1	Cilíndricas	107
B.2	Esféricas	108
B.3	Esferoidais Prolatas	108
B.4	Esferoidais Oblatas	109
B.5	Toroidais	110
B.6	Esferas Tangentes	111
B.7	Bi-Esféricas	111

C Soluções Multipolares	113
C.1 Multipólos em Coordenadas Cilíndricas	113
C.2 Multipólos em Coordenadas Prolatas	114
D Métodos Numéricos	119
D.1 Integração Numérica	119
D.2 Problemas de Valor Inicial	123
D.3 Listagens dos Programas	125

Notações

Ao longo deste trabalho utilizaremos as seguintes notações:

\mathcal{J}_a Função de Bessel do primeiro tipo e de ordem a ,

\mathcal{Y}_a Função de Bessel do segundo tipo e de ordem a ,

\mathcal{P}_a Polinômio de Legendre de grau a ,

\mathcal{Q}_a Função de Legendre de segundo tipo e de ordem a ,

$\text{sn}_\kappa x$ Função elíptica de Jacobi,

$\text{cn}_\kappa x$ Função elíptica de Jacobi,

$K(\kappa)$ Integral elíptica do primeiro tipo e módulo κ ,

Ei Integral exponencial,

$\delta(x)$ Distribuição delta de Dirac,

f_x Derivada parcial de f por x , $f_x = \frac{\partial f}{\partial x}$,

\bar{z} Complexo conjugado de z ,

$a!!$ $a!! = a(a-2)(a-4) \cdots k$ onde k é $0!!$ ou $(-1)!!$, tendo definido $0!! = (-1)!! = 1$,

$\mathcal{O}(\cdot)$ Da ordem de, ou seja, se $f(t) = \mathcal{O}(g(t))$ quando $t \rightarrow t_0$ então $\lim_{t \rightarrow t_0} \frac{f(t)}{g(t)} = 0$,

x^a Coordenadas de um evento, onde $(x^0, x^1, x^2, x^3) = (t, r, z, \varphi)$.

Resumo

Esta dissertação é uma revisão das soluções das equações de Einstein para sistemas estáticos com simetria axial, que são conhecidas como soluções de Weyl.

Os primeiros capítulos tratam dos aspectos gerais destas soluções. Inicialmente apresentando a forma das equações de Einstein para este caso e as equações para suas geodésicas. Em seguida tratamos das singularidades que podem ser encontradas.

Nos capítulos seguintes são apresentadas várias soluções de Weyl, com algumas informações sobre suas interpretações físicas e geométricas. Começando com soluções para distribuições de matéria em equilíbrio estático, isto é, que não contradizem a estaticidade imposta a estas métricas. Depois apresentamos uma série de soluções onde a hipótese de estaticidade não é compatível com a distribuição de matéria, essa discrepância é resolvida pelo aparecimento de singularidades estruturais. Algumas destas soluções são originais sendo obtidas numericamente.

Por fim segue um conjunto de apêndices. Entre os quais, um com a descrição dos métodos numéricos empregados e as listagens dos programas.

Abstract

This dissertation is a review of static axial-symmetric solutions of Einstein equations also known as Weyl solutions.

The first chapters deal with general aspects of these solutions, the form of Einstein equation and its geodesic equations. Next we study the singularities at these metrics.

In the following chapters several Weyl solutions, also some information about its physic and geometric interpretations are presented. Beginning with solutions for matter distributions with static equilibrium, that is, which agree with the imposed staticity of the metric. We show a list of solutions which contradict this hypothesis, this discrepancy is solved by the existence of strut singularities. Some of this solutions are originals and are obtained using numerical methods.

At the end follows a set of appendices. Among them, there are a description of numerical methods employed and the programs listings.

Introdução

Este trabalho foi motivado pela falta, na literatura científica, de uma revisão das soluções de Weyl, que são soluções, para a equação de Einstein, estáticas e com simetria axial. Mesmo o livro de Kramer et. al. "*Exact Solutions of Einstein's Field Equations*" [41] contém apenas uma curta seção sobre estas soluções. Outras revisões, também parciais, podem ser encontradas em alguns artigos, tais como Bonnor [12], Bonnor e Martins [10] ou Letelier e Oliveira [51]. Tentamos então produzir uma revisão o mais completa possível, englobando as soluções com suas propriedades físicas e geométricas.

Uma descrição geral das métricas de Weyl é feita no primeiro capítulo. Começando pela descrição matemática de suas simetrias e apresentando a forma das equações de Einstein quando estas simetrias são impostas, conhecidas como equações de Weyl. Incluímos, em seguida as equações para as geodésicas destas métricas, estas equações serão depois resolvidas numericamente para algumas métricas. Por fim, apresentamos alguns sistemas de coordenadas especiais onde também podemos escrever as soluções de Weyl.

O capítulo seguinte trata do estudo das singularidades. Nele apresentamos as fórmulas para alguns invariantes escalares nas coordenadas de Weyl, tratamos resumidamente das singularidades intermediárias e da caracterização de horizontes nos espaços de Weyl. Em seguida, descrevemos as singularidades estruturais¹, que são uma característica de espaço-tempos estáticos. E dedicamos a última seção ao estudo das extensões maximais e dos diagramas de Carter-Penrose.

Os terceiro e quarto capítulos são listas de soluções. O primeiro deles trata de soluções em equilíbrio, ou seja, que não contêm singularidades estruturais. Enquanto o segundo apresenta soluções que são sobreposições das anteriores e em geral são sustentadas por estruturas. Para muitas soluções foram incluídos os gráficos das funções ϕ e ν , que definem um espaço de Weyl. Estes gráficos além de nos dar uma noção qualitativa das soluções, permitem-nos comparar os resultados conhecidos analiticamente com os obtidos numericamente.

O primeiro apêndice contém um resumo dos teoremas sobre singularidades de Hawking e Ellis. As definições e teoremas são bastante úteis para a compreensão do conceito de singulari-

¹Em inglês o termo empregado é *strut* que significa escorar. Porém como a expressão *singularidade de escora* nos parece cacofônica preferimos uma tradução incorreta. Porém devemos esclarecer que não há nenhuma relação destas singularidades com a estabilidade estrutural da teoria de equações diferenciais.

dade, mas os principais resultados tratam de universos dinâmicos, não se aplicando a soluções estáticas, como as de Weyl.

Em seguida incluímos uma lista com diversos sistemas de coordenadas, que foram empregados ao longo do texto. Preferimos apresentá-los reunidos, pois desta forma servem como um pequeno inventário, com os sistemas de coordenadas empregados para o estudo das soluções de Weyl.

No Apêndice C incluímos alguns resultados sobre as soluções multipolares, em coordenadas cilíndricas e prolatas. Por fim, no Apêndice D, apresentamos os métodos numéricos e as listagens dos programas implementados, bem como um pequeno manual sobre seu uso. A linguagem empregada para a implementação destes programas foi C++, enquanto os gráficos e outros resultados foram gerados empregando os pacotes matemáticos Matlab e Mathematica.

Métricas de Weyl

2.1 Equações de Weyl

A relatividade geral postula que a presença de matéria, ou energia, distorce o espaço e o tempo a sua volta. Essa distorção pode ser descrita através da definição de uma métrica, que ao ser expressa como um elemento de linha assume a forma

$$ds^2 = g_{ab} dx^a dx^b.$$

Aqui foi empregada a notação tensorial de soma dos índices mudos, dx^a representa um deslocamento infinitesimal com relação à coordenada x^a , onde $a = 0, 1, 2, 3$ e g_{ab} é o tensor métrico, com assinatura de Lorentz, $(+ - - -)$ ou $(- + + +)$. Como ambas são equivalentes podendo ser escolhidas arbitrariamente, trabalharemos com o primeiro caso.

Dada uma distribuição de matéria e energia a métrica fica determinada pelas equações de Einstein, que fundamentam toda a teoria da relatividade geral. Estas equações são

$$G_{ab} = \frac{8\pi G}{c^2} T_{ab}, \quad (2.1)$$

onde G_{ab} é o tensor de Einstein, definido como $R_{ab} - \frac{1}{2}g_{ab}R$, onde R_{ab} é o tensor de Ricci e $R = R^a_a$ seu traço. Deste modo, o tensor de Einstein está diretamente relacionado com a curvatura do espaço-tempo. Enquanto T_{ab} é o tensor de energia momento, ou seja, é o elemento que contém as informações sobre a distribuição de matéria e energia no espaço. Para que não seja necessário manipular constantes ao longo dos cálculos é comum em relatividade trabalharmos com sistemas de unidades nos quais a constante gravitacional, G , e a velocidade da luz, c , sejam iguais a 1. Também seguiremos essa convenção.

Como (2.1) é uma equação tensorial ela é composta por 16 equações, mas como estamos tratando de espaços sem torsão os tensores são simétricos, reduzindo este número a apenas 10 equações independentes. Por outro lado, o tensor de Ricci, R_{ab} , é escrito em função das derivadas segundas de g_{ab} , que são as incógnitas do problema e assim chegamos a dez equações diferenciais parciais não lineares de segunda ordem para determinar as dez componentes independentes de g_{ab} . Enquanto o tensor T_{ab} em muitos casos também depende de g_{ab} o que torna o problema ainda mais complexo.

Para resolver estas equações, foram impostas condições de simetria aos espaço-tempos, o que é justificado, pois os objetos astronômicos possuem simetrias. Outra simplificação é considerar a métrica estática, isto é, supor que as fontes do campo gravitacional são imóveis. Porém, esta última simplificação nem sempre é compatível com a realidade observada; por exemplo, se dois corpos estiverem sob gravitação mútua, eles tenderão ao movimento, a menos que alguma força não gravitacional os sustente imóveis. Essa incompatibilidade possui implicações que serão tratadas na Seção 3.5.

Outro fator que simplifica consideravelmente as equações é buscar soluções apenas para o vácuo, ou seja, considerar $T_{ab} = 0$. Isso equivale, no caso Newtoniano, a tomar a equação de Laplace $\nabla^2 V = 0$, ao invés da equação de Poisson $\nabla^2 V = \rho$. Com isso não só tornamos as equações homogêneas, como também podemos simplificar as próprias equações que passam a ser escritas como

$$R_{ab} = 0. \quad (2.2)$$

Em 1916 surgiu a primeira solução exata das equações de Einstein, a solução de Schwarzschild. Esta métrica, que depois veio a definir o que hoje conhecemos como buracos negros, é a solução das equações de Einstein para o vácuo (2.2) estática e com simetria esférica, dependendo apenas de coordenada radial, o que transforma as equações diferenciais parciais em ordinárias. Retornaremos a esta solução como um caso particular das soluções de Weyl (veja a Seção 4.5).

A simetria esférica é muito restritiva pois a métrica só pode depender de uma das quatro coordenadas do espaço-tempo. Esta restrição é tão forte que existem teoremas de unicidade, impondo que a solução de Schwarzschild seja a única solução estática com simetria esférica para o vácuo. Um caso menos restritivo, e conseqüentemente mais rico em soluções, é o que considera espaços estáticos com simetria axial. Com esta hipótese a métrica pode depender de duas coordenadas do espaço-tempo, sendo independente apenas do tempo e do ângulo de rotação em relação ao eixo de simetria. Este foi o caso tratado por Weyl, e que recebeu seu nome.

Vamos agora mostrar os passos principais para a obtenção das equações de Weyl. Inicialmente serão impostas as simetrias. A forma matematicamente correta de impor estas condições é determinar os campos vetoriais que correspondem a estas simetrias, ou seja, os campos de Killing. No nosso caso teremos um campo vetorial tipo-tempo, ζ , e um tipo-espaço, ξ , hiper-superfície ortogonais. Então impomos que a métrica seja invariante sob translações ao longo das linhas de fluxo destes campos. Empregando a derivação de Lie podemos escrever estas restrições como

$$\mathcal{L}_\zeta g_{ab} = \mathcal{L}_\xi g_{ab} = 0, \quad (2.3)$$

onde $\mathcal{L}_{\zeta, \xi}$ denota a derivada de Lie com relação ao campo ζ ou ξ .

Como estamos buscando um sistema de coordenadas específico, não precisamos trabalhar com caracterizações invariantes; podemos desde já determinar que as coordenadas serão adaptadas a esse campos de Killing. Escolhendo x^0 como deslocamento ao longo de ζ e x^3 ao longo

de ξ e então substituindo x^0 por t e x^3 por φ , que definimos como ângulo de rotação em relação ao eixo de simetria. Conseqüentemente a métrica não terá dependências com relação a estas coordenadas. Além disso, os termos cruzados que envolvem dt ou $d\varphi$ devem ser nulos, pois caso contrário ao invertermos o sinal de uma destas coordenadas, $t \rightarrow -t$ ou $\varphi \rightarrow -\varphi$, alteraríamos a métrica. Impostas estas restrições podemos escrever o elemento de linha correspondente a um espaço de Weyl como

$$ds^2 = A dt^2 - \gamma_{ij} dx^i dx^j - B d\varphi^2, \quad (2.4)$$

onde A , B e γ_{ij} são funções apenas de x^1 e x^2 e os índices i, j assumem os valores 1 e 2.

Como o elemento de linha deixa de ser diagonal apenas com relação às coordenadas x^1 e x^2 , e qualquer métrica bidimensional é conformalmente plana podemos aplicar uma transformação de $(x^1, x^2) \rightarrow (r, z)$ que leve (2.4) em

$$ds^2 = A dt^2 - D(dr^2 + dz^2) - B d\varphi^2, \quad (2.5)$$

onde A , B e D são funções de r e z . A partir deste ponto empregaremos a seguinte numeração das coordenadas $(x^0, x^1, x^2, x^3) = (t, r, z, \varphi)$.

Até aqui, apenas pela aplicação das simetrias impostas, reduzimos o problema de determinar as 10 funções de g_{ab} para o problema de determinar apenas as funções A , B e D . Podemos agora, através da imposição das equações de Einstein para o vácuo (2.2), reduzir para apenas duas incógnitas, ϕ e ν , ficando com

$$ds^2 = e^{2\phi} dt^2 - e^{2(\nu-\phi)}(dr^2 + dz^2) - r^2 e^{-2\phi} d\varphi^2, \quad (2.6)$$

onde as exponenciais foram incluídas por conveniência e as funções ϕ e ν dependem apenas das coordenadas r e z . Para completar a definição das soluções de Weyl devemos fixar os intervalos de definição das coordenadas. Tomaremos então os intervalos

$$-\infty < t < \infty, \quad -\infty < z < \infty, \quad 0 < r < \infty, \quad 0 \leq \varphi \leq 2\pi,$$

onde identificamos $\varphi = 0$ com $\varphi = 2\pi$. Com esta identificação impomos a topologia cilíndrica, $\mathcal{R} \times (\mathcal{R}^2 \times S)$, que caracteriza as soluções de Weyl.

Sob as condições descritas acima as equações de Einstein para o vácuo (2.2) são reduzidas ao sistema de equações

$$\nabla^2 \phi = \phi_{rr} + \frac{\phi_r}{r} + \phi_{zz} = 0, \quad (2.7)$$

$$\nu[\phi] = \int r[(\phi_r^2 - \phi_z^2)dr + 2\phi_r \phi_z dz], \quad (2.8)$$

que são conhecidas como equações de Weyl.

Note que a integral que define a função ν será sempre bem definida pois (2.7) é exatamente a condição de integrabilidade de (2.8). Uma dedução mais detalhada destes resultados pode ser encontrada no livro de Synge [66]. As funções ϕ e ν algumas vezes são chamadas de potenciais de Weyl. Para campos fracos $\frac{1}{2} \ln g_{tt} = \phi$ aproxima o potencial Newtoniano.

Na verdade, para que possamos escrever a métrica na forma (2.6), não é necessário que as equações de Einstein para o vácuo (2.2) sejam satisfeitas integralmente. Basta que a relação

$$R_1^1 + R_2^2 = 0 \quad (2.9)$$

seja válida. Quando esta relação e as condições de simetria descritas acima são satisfeitas, podemos escrever as equações de Einstein (2.1) como segue [58]

$$\nabla^2 \phi = -4\pi e^{2(\nu-\phi)} (T_0^0 - T_3^3), \quad (2.10)$$

$$\nu_r = r(\phi_r^2 - \phi_z^2), \quad (2.11)$$

$$\nu_z = 2r\phi_r \phi_z, \quad (2.12)$$

$$\nu_{rr} + \nu_{zz} - \nabla^2 \phi + \phi_r^2 + \phi_z^2 = 4\pi e^{2(\nu-\phi)} (T_0^0 + T_3^3). \quad (2.13)$$

Um dos aspectos que tornaram as métricas de Weyl importantes é que a equação (2.7) é exatamente o Laplaciano em coordenadas cilíndricas com simetria axial, ou seja, é linear e suas soluções já foram estudadas, no contexto da gravitação Newtoniana, por isso a função ϕ é muitas vezes chamada de Imagem Newtoniana da solução, como foi sugerido por Letelier e Oliveira [48]. Mas devemos tomar cuidado com esta nomenclatura, pois, embora aos pontos singulares de ϕ correspondam as fontes da solução, o significado real da métrica pode diferir consideravelmente de sua Imagem Newtoniana, como veremos nos casos da métrica de Schwarzschild, cuja Imagem Newtoniana é uma barra, e do espaço de Minkowski acelerado, cuja função ϕ é singular sobre uma linha semi-infinita.

O processo de obter uma solução de Weyl partindo de uma solução conhecida da equação de Laplace é matematicamente simples. Porém torna impossível uma interpretação a priori da solução obtida. Por outro lado, a escolha do sistema de coordenadas foi determinada por critérios puramente matemáticos, o que obscurece as propriedades físicas da solução. Assim surge o problema de interpretar corretamente uma solução de Weyl. Além disso como estamos fixos a um sistema de coordenadas particular estamos presos à carta deste sistema, conseqüentemente toda a informação disponível refere-se a apenas um aberto contido na variedade do espaço-tempo. Este aspecto será tratado com mais detalhes na Seção 3.6, sobre extensões maximais.

A integral (2.8), que chamaremos de integral de Weyl, engloba os aspectos não lineares das equações de Einstein. Deste modo, ao contrário da equação de Laplace, a soma de soluções não é uma solução. O que dificulta a sobreposição de métricas já conhecidas, entretanto existem algumas identidades [48] que podem ser úteis

$$\nu[\phi, \psi] = \int r [(\phi_r \psi_r - \phi_z \psi_z) dr + (\phi_r \psi_z + \phi_z \psi_r) dz], \quad (2.14)$$

$$\nu[\phi + \psi] = \nu[\phi] + \nu[\psi] + 2\nu[\phi, \psi], \quad (2.15)$$

$$\nu[\alpha\phi] = \alpha^2 \nu[\phi], \quad (2.16)$$

$$\nu[\phi(a)] = \lim_{b \rightarrow a} \nu[\phi(b), \phi(a)], \quad (2.17)$$

$$\nu[\phi, \psi] = \nu[\psi, \phi], \quad (2.18)$$

$$\overline{\nu[\phi, \psi]} = \nu[\overline{\phi}, \overline{\psi}], \quad (2.19)$$

$$\frac{\partial}{\partial a} \nu[\phi(a), \psi] = \nu \left[\frac{\partial \phi(a)}{\partial a}, \psi \right]. \quad (2.20)$$

α é uma constante, a e b são parâmetros.

2.2 Geodésicas nas Coordenadas de Weyl

Uma característica que diferencia a relatividade geral da gravitação Newtoniana é que a primeira já traz, incluída em suas equações de campo, as equações de movimento. Embora ambos os casos tenham uma certa equivalência matemática, a trajetória de uma partícula de teste, ou seja, uma partícula sujeita ao campo gravitacional mas cuja massa é desprezível, será dada por uma curva estacionária de um funcional, conhecido como integral de ação. Porém na gravitação Newtoniana este funcional consiste de um adendo à teoria, baseado na função Lagrangeana, enquanto na relatividade geral ele é o próprio comprimento de arco da trajetória, dado por

$$S = \int \sqrt{g_{ab} \dot{x}^a \dot{x}^b} du, \quad (2.21)$$

Nesta expressão o ponto denota derivação com relação ao parâmetro u .

Como na relatividade geral a métrica não é definida positiva, podemos ter três situações: $g_{ab} \dot{x}^a \dot{x}^b$ maior, menor ou igual a zero. Quando $g_{ab} \dot{x}^a \dot{x}^b > 0$ as curvas estacionárias de S são trajetórias de partículas de teste. Se $g_{ab} \dot{x}^a \dot{x}^b = 0$ são raios de luz. Enquanto o caso $g_{ab} \dot{x}^a \dot{x}^b < 0$ não tem significado físico.

O conhecimento das trajetórias de partículas teste é de grande utilidade para a compreensão das forças gravitacionais inerentes a uma determinada solução. Isso nos permite inferir algumas propriedades relativas à fonte deste campo gravitacional. Simultaneamente, em termos da geometria, o conhecimento das geodésicas nos permite uma melhor compreensão do espaço-tempo.

Apresentamos agora as equações para o cálculo das geodésicas com base em princípios variacionais. Poderíamos também calcular as geodésicas com base em suas propriedades geométricas, como ser a curva cujo vetor tangente é transportado paralelamente.

Primeiramente, chamamos a atenção para o fato de que uma curva será estacionária com relação a (2.21) se e, somente se, também o for com relação a

$$S' = \int g_{ab} \dot{x}^a \dot{x}^b du, \quad (2.22)$$

Conseqüentemente, como este funcional é mais simples, é nele que os cálculos baseiam-se.

Agora, se definirmos $K = g_{ab} \dot{x}^a \dot{x}^b$, podemos dizer que, com uma escolha adequada do parâmetro u , que chamaremos parâmetro afim, o valor de K não se altera ao longo de uma

geodésica, pois ele é o quadrado da norma do vetor tangente a curva. E como sabemos, este vetor é propagado paralelamente. Como estamos em um espaço pseudo-Riemanniano, com assinatura de Lorentz, K poderá assumir qualquer valor real. Quando $K = 0$ a geodésica associada é uma geodésica nula, e representa a trajetória de um raio de luz. Caso contrário, devido á escolha do parâmetro u , K pode assumir os valores 1 ou -1. De acordo com a assinatura que estamos adotando quando $K = 1$, esta geodésica será tipo-tempo e representará a trajetória de uma partícula teste, que como no caso Newtoniano não será influenciada pela massa da partícula. O caso $K = -1$ que representa uma geodésica tipo-espaço. Sintetizando temos

$$K = g_{ab}\dot{x}^a\dot{x}^b = \alpha, \quad (2.23)$$

onde α é a constante que determina o tipo da geodésica

$$\alpha = \begin{cases} -1 & \text{geodésica tipo-espaço} \\ 0 & \text{geodésica nula} \\ 1 & \text{geodésica tipo-tempo.} \end{cases}$$

Aplicando os princípios variacionais a (2.22) chegamos às equações de Euler-Lagrange

$$\frac{\partial K}{\partial x^a} - \frac{d}{du} \frac{\partial K}{\partial \dot{x}^a} = 0. \quad (2.24)$$

Temos então cinco equações (2.23)–(2.24) para determinar as quatro funções que parametrizam a geodésica. Podemos, assim, ignorar uma delas.

De posse das equações genéricas vamos aplicá-las ao elemento de linha (2.6) para obtermos as equações que determinam as geodésicas em um espaço de Weyl. Expandindo (2.23) obtemos a seguinte expressão

$$e^{2\phi}\dot{t}^2 - e^{2(\nu-\phi)}(\dot{r}^2 + \dot{z}^2) - r^2 e^{-2\phi}\dot{\phi}^2 = \alpha.$$

Do mesmo modo substituindo $x^a = t, r, z, \phi$ em (2.24) teremos

$$\ddot{t} = -2\dot{t}(\phi_r\dot{r} + \phi_z\dot{z}),$$

$$\ddot{\phi} = 2\dot{\phi}(\phi_r\dot{r} + \phi_z\dot{z} - r^{-1}),$$

$$\ddot{r} = \dot{\phi}^2 e^{-2\phi}(2r^2\phi_r + r) - 2\dot{r}\dot{z}(\nu_z - \phi_z) - \dot{r}^2\nu_r + \dot{z}^2(\nu_r - 2\phi_r) - \alpha\phi_r e^{-2(\nu-\phi)},$$

$$\ddot{z} = \dot{\phi}^2 e^{-2\phi}(2r^2\phi_z + r) - 2\dot{r}\dot{z}(\nu_r - \phi_r) - \dot{r}^2\nu_z + \dot{z}^2(\nu_z - 2\phi_z) - \alpha\phi_z e^{-2(\nu-\phi)}.$$

Os subscritos indicam derivação parcial com relação à coordenada indicada e os pontos indicam a derivada total com relação ao parâmetro afim u .

Embora o cálculo das geodésicas seja um problema de valor inicial para equações diferenciais ordinárias é difícil encontrar suas soluções analíticas, pois as funções ϕ e ν podem ser muito complexas. Deste modo desenvolvemos um programa que resolve numericamente estas equações. Foram empregados métodos convencionais para equações diferenciais ordinárias, como os métodos de Runge-Kutta. No Apêndice D apresentamos tanto os algoritmos quanto as listagens dos programas, em C++ empregados. Para algumas soluções dos Capítulos 4 e 5 foram incluídas as trajetórias espaciais de suas geodésicas tipo-tempo.

2.3 Coordenadas Especiais

Coordenadas de Lewis

O elemento de linha (2.6) é o mais usado, as vezes com pequenas alterações, para descrever os espaços de Weyl, mas não é o único. Outro elemento, utilizado por alguns autores, é obtido empregando as coordenadas de Lewis, x e y

$$ds^2 = e^{2\chi} dt^2 - e^{2(\gamma-\chi)}(dx^2 + dy^2) - \rho^2 e^{-2\chi} d\phi^2. \quad (2.25)$$

As funções ρ , χ e γ , dependem apenas das coordenadas x e y , e satisfazem as equações de campo

$$\rho_{xx} + \rho_{yy} = 0, \quad (2.26)$$

$$\rho_x \chi_x + \rho_y \chi_y + \rho(\chi_{xx} + \chi_{yy}) = 0, \quad (2.27)$$

$$\gamma_{xx} + \gamma_{yy} = \chi_y^2 - 2i\chi_y \chi_x - \chi_x^2, \quad (2.28)$$

$$\partial\gamma\partial\rho = \frac{1}{2}\partial^2\rho + \rho(\partial\chi)^2, \quad (2.29)$$

onde $\partial = \partial_x + i\partial_y$.

As métricas (2.6) e (2.25) são totalmente compatíveis, pois como a função ρ é harmônica sempre existe f tal que

$$2\rho(x, y) = f(x + iy) + f(x - iy).$$

Então, definindo a transformação de coordenadas

$$2r = f(x + iy) + f(x - iy),$$

$$2iz = f(x + iy) - f(x - iy),$$

podemos mostrar que as soluções nas coordenadas de Weyl e de Lewis são relacionadas por

$$e^{2\gamma} = \left| \frac{\partial f}{\partial x} + i \frac{\partial f}{\partial y} \right|^2 e^{2\nu}, \quad \chi = \phi.$$

Coordenadas de Schwarzschild

Como a simetria esférica é um caso especial da simetria axial a métrica de Schwarzschild será um caso particular das métricas de Weyl. Na Seção 4.5 veremos que a solução de Schwarzschild, em coordenadas de Weyl, é dada por

$$\phi = \frac{1}{2} \log \frac{R_- + R_+ - 2m}{R_- + R_+ + 2m}, \quad \nu = \frac{1}{2} \log \frac{(R_- + R_+)^2 - 4m^2}{4R_- R_+}, \quad (2.30)$$

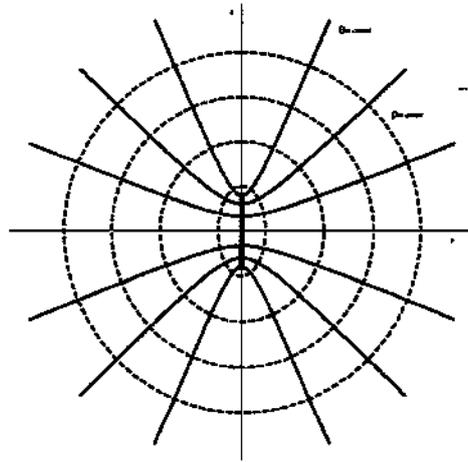


Figura 2.1: Transformação para coordenadas de Schwarzschild.

onde definimos $R_{\pm}^2 = r^2 + (z \pm m)^2$. O elemento de linha (2.6) para esta métrica assume a forma

$$ds^2 = \Phi dt^2 - \frac{\Upsilon}{\Phi} (dr^2 + dz^2) - \frac{r^2 d\varphi^2}{\Phi},$$

onde

$$\Phi = \frac{R_- + R_+ - 2m}{R_- + R_+ + 2m}, \quad \Upsilon = \frac{(R_- + R_+)^2 - 4m^2}{4R_- R_+}.$$

Para colocarmos esta solução na sua forma convencional

$$ds^2 = \left(1 - \frac{2m}{\rho}\right) dt^2 - \left(1 - \frac{2m}{\rho}\right)^{-1} d\rho^2 - \rho^2 (d\theta^2 + \sin^2 \theta d\varphi^2) \quad (2.31)$$

basta aplicarmos a transformação

$$r = \sqrt{\rho^2 - 2m\rho} \sin \theta, \quad z = (\rho - m) \cos \theta. \quad (2.32)$$

É importante notar que esta transformação só é válida para $\rho > 2m$, isto é, a solução (2.30) está relacionada apenas com o exterior do horizonte de eventos da métrica de Schwarzschild.

A relação entre as coordenadas de Weyl e de Schwarzschild pode ser vista na Figura 2.1. Note que o raio de Schwarzschild torna-se a barra $r = 0$ e $-m \leq z \leq m$, enquanto a órbita de Schwarzschild, no plano equatorial, dada por $\rho = 3m$, torna-se $r = \sqrt{3}m$.

Podemos agora aplicar a transformação (2.32) a uma métrica de Weyl arbitrária. Este processo pode ser útil para compreendermos algumas métricas, principalmente as multipolares, descritas na Seção 4.11, pois são, em certo sentido, próximas da métrica de Schwarzschild. Deste modo, obtemos as solução de Weyl em coordenadas de Schwarzschild

$$ds^2 = e^{2\phi} dt^2 - e^{2(\nu-\phi)} \left(S + \frac{m^2}{\rho^2} \sin^2 \theta\right) \left(\frac{d\rho^2}{S} + \rho^2 d\theta^2\right) - \rho^2 S \sin^2 \theta e^{-2\phi} d\varphi^2, \quad (2.33)$$

onde $S = 1 - 2m\rho^{-1}$ e as funções ϕ e ν estão escritas em termos das novas coordenadas ρ e θ . Notando que $R_{\pm} = |\rho + m(\pm \cos \theta - 1)|$ e que $\rho > 2m$ podemos reescrever as funções ϕ e ν relativas a solução de Schwarzschild como

$$\phi = \frac{1}{2} \ln S, \quad \nu = \frac{1}{2} \ln S - \frac{1}{2} \ln \left(S + \frac{m^2}{\rho^2} \sin^2 \theta \right).$$

Singularidades

3.1 Introdução

Singularidades são elementos comuns nas soluções das equações de Einstein. Porém, as soluções com que estamos acostumados a lidar são altamente simétricas. Observando-se isso levantou-se a questão de que estas singularidades seriam frutos puramente das simetrias impostas e não resultados que representem fenômenos naturais. Esta questão foi respondida por Hawking e Ellis [31] que, sob algumas condições, demonstraram que toda solução das equações de Einstein possui singularidades. Porém como estes teoremas se aplicam a espaço-tempos evolutivos as soluções de Weyl não satisfazem suas condições. Mesmo assim, apresentamos um resumo dos resultados relacionados com estes teoremas no Apêndice A, como a definição de singularidades baseada na incompletude geodésica, que se aplica a qualquer espaço-tempo.

Como as soluções são sempre dadas em um sistema de coordenadas específico, enquanto as propriedades físicas de uma métrica devem ser invariantes, temos o problema de determinar se as singularidades que encontramos são de natureza física ou provocadas pelo sistema de coordenadas. Para conseguirmos fazer essa distinção, podemos empregar os invariantes escalares, definidos na próxima seção. Como eles não dependem do sistema de coordenadas, suas singularidades são consideradas reais.

Além das singularidades invariantes, isto é, que não dependem do observador, podemos pensar em singularidades que sejam sentidas por observadores especiais. Por exemplo, aqueles em queda livre. Estas singularidades não são invariantes, pois dependem do sistema de coordenadas, porém são físicas, pois poderiam, ao menos em princípio, serem testadas experimentalmente. Tratamos sucintamente deste caso na seção sobre singularidades intermediárias.

As duas últimas seções deste capítulo tratam de características específicas dos espaços de Weyl, como a determinação de seus horizontes e a definição das singularidades estruturais, que são oriundas da estaticidade.

3.2 Invariantes Escalares

Qualquer função escalar é invariante sob transformações de coordenadas. Assim se pudermos contruir escalares com significado geométrico teremos alguma informação invariante sobre a geometria. Os invariantes que apresentamos aqui são construídos por contrações do tensor de

Riemann, que é definido pela expressão

$$R_{jkl}^i = \partial_k \Gamma_{jl}^i - \partial_l \Gamma_{jk}^i + \Gamma_{pk}^i \Gamma_{jl}^p - \Gamma_{pl}^i \Gamma_{jk}^p.$$

Como estamos tratando de soluções sem torsão, Γ é o símbolo de Christoffel

$$\Gamma_{ij}^k = -\frac{1}{2} g^{kl} (\partial_i g_{lj} + \partial_j g_{il} - \partial_l g_{ij}).$$

Vale lembrar que as métricas de Weyl são soluções para o vácuo e portanto o traço do tensor de Riemann é nulo. O tensor de Weyl C_{ijkl} é o tensor de Riemann subtraído de seu traço, então este dois tensores são idênticos nestes espaços.

Nas coordenadas de Weyl (2.6) podemos escrever as componentes não nulas do tensor de Riemann como [10]

$$\begin{aligned} e^{-2(\nu-\phi)} R_{2121} &= -r^{-2} e^{2(\nu-\phi)} R_{3030} = Q, \\ -r^{-2} e^{2\phi} R_{2323} &= e^{-2\phi} R_{1010} = P, \\ e^{2\phi} R_{2020} &= -r^{-2} e^{2\phi} R_{1313} = S, \\ e^{-2\phi} R_{2010} &= -r^{-2} e^{2\phi} R_{2313} = T, \end{aligned} \quad (3.1)$$

$$\text{onde } P = \phi_{rr} + 2\phi_r^2 - \nu_r \phi_r + \nu_z \phi_z - \phi_z^2, \quad Q = \phi_z^2 + \phi_r^2 - \frac{\phi_r}{r},$$

$$S = \phi_{zz} + 2\phi_z^2 - \nu_z \phi_z + \nu_r \phi_r - \phi_r^2, \quad T = \phi_{zr} + 3\phi_z \phi_r - \nu_z \phi_r - \nu_r \phi_z.$$

Com estes resultados podemos definir os invariantes desejados. Apresentamos aqui as fórmulas para dois deles: o escalar de Kretschmann

$$K = R^{ijkl} R_{ijkl} = 8e^{-4(\nu-\phi)} (P^2 + Q^2 + S^2 + 2T^2) \quad (3.2)$$

e o invariante cúbico

$$N = R^{ijkl} R_{ijmn} R^{mn}_{ij} = 48e^{-6(\nu-\phi)} Q(PS - T^2). \quad (3.3)$$

3.3 Singularidades Intermediárias

Como dissemos, uma singularidade sentida por um observador em queda livre pode ser considerada física. Para estudarmos o que seria visto por este observador utilizamos as projeções do tensor de Riemann sobre as tetradas ortogonais naturais, que são definidas a partir de uma geodésica tipo-tempo, tomando o vetor unitário tangente a curva e três vetores espaciais unitários, todos ortogonais entre si e propagados paralelamente ao longo da curva. Nestas condições as componentes não nulas e independentes do tensor de Riemann são [51]

$$R_{\hat{t}\hat{r}\hat{t}\hat{r}} = \kappa \left[\phi_{rr} + \phi_r^2 (2 - r\phi_r) - \phi_r^2 (1 - 3r\phi_r) \right],$$

$$R_{\hat{t}\hat{z}\hat{t}\hat{z}} = \kappa \left[\phi_{zz} + \phi_r^2 (1 - r\phi_r) + \phi_r^2 (2 - 3r\phi_r) \right],$$

$$R_{\hat{t}\hat{r}\hat{z}\hat{z}} = \kappa \left[\phi_{rz} + 3\phi_r \phi_z (1 - 3r\phi_r) + r\phi_z^3 \right],$$

onde definimos $\kappa = \exp 2(\phi - \nu)$.

3.4 Horizontes nas Métricas de Weyl

Na relatividade geral o conceito de horizonte é modelado pelas propriedades do horizonte de eventos da métrica de Schwarzschild. Tentando generalizar esse conceito para aplicá-lo a outras métricas, Carter [14] definiu os horizontes de Killing. Estes horizontes são definidos como superfícies nulas, tais que seus geradores sejam proporcionais aos vetores de Killing de uma isometria tipo-tempo. No caso das soluções de Weyl a norma de vetor de Killing tipo-tempo é dada por e^ϕ . De modo que, nas regiões onde $\phi \rightarrow -\infty$ temos horizontes de Killing.

3.5 Singularidades Estruturais

As singularidades estruturais surgem da incompatibilidade de algumas distribuições de matéria com a restrição de estaticidade. Esta incompatibilidade também existe na física Newtoniana, porém ela não é detectada pelas equações de campo, enquanto que na relatividade, como as equações de campo incorporam as equações de movimento, esta incompatibilidade não só é detectada como também resolvida pelo surgimento de singularidades que sustentem a estaticidade. Obviamente estas singularidades não representam nenhum fenômeno físico, elas são apenas o resultado de uma restrição matemática inadequada ao modelo.

Nas métricas de Weyl estas singularidades podem aparecer sob duas formas. No eixo z ligando dois corpos, ou um corpo e o infinito, como violação da condição de suavidade elementar¹. Ou, quando um dos corpos que tende ao movimento for um anel, esta singularidade aparece como uma descontinuidade da função ν , no interior deste anel. Chamaremos esta descontinuidade de membrana [51]. O Capítulo 5 contém várias soluções que servem como exemplo para ambos os casos.

A condição de suavidade elementar é a exigência de que em cada ponto do espaço-tempo exista uma vizinhança difeomórfica ao espaço de Minkowski. Olhando para uma superfície no \mathbf{R}^3 esta condição é equivalente à existência de um plano tangente em cada ponto da superfície. Deste modo, o equivalente a uma singularidade estrutural ocorre no vértice de um cone usual em \mathbf{R}^3 . Por este motivo esta singularidade também é chamada de singularidade cônica.

Como os espaços com que estamos lidando são estáticos e com simetria axial, a condição de suavidade elementar equivale a exigir que a coordenada angular, φ , tenha a periodicidade convencional, 2π , sobre o eixo de simetria. Para isso é necessário que [41]

$$\lim_{\text{eixo}} \frac{X_{,a} X^{,a}}{4X} = 1,$$

onde $X = \xi_a \xi^a$ é norma do vetor de Killing espacial. Nas coordenadas de Weyl esta condição simplifica-se em [51]

$$\lim_{r \rightarrow 0} \nu = 0. \tag{3.4}$$

Como estas singularidades sustentam os corpos em oposição à força gravitacional, podemos calcular as forças de compressão que elas exercem. Para as estruturas estas forças F_z , são

¹Em inglês *Elementary Flatness Condition*.

calculadas integrando T_z^z em um disco centrado em $r = 0$ e com as coordenadas t e z fixas

$$F_z = \int_{\text{disco}} T_z^z d\sigma,$$

onde $d\sigma$ é um elemento de superfície. Aplicando diferentes definições para este elemento teremos diferentes expressões para a força, mas que ao menos em primeira ordem são equivalentes.

Tomando o elemento de superfície da sub-variedade bidimensional, definido quando fixamos t e z em (2.6), que é dado por $d\sigma = e^{\nu-2\phi} r dr d\varphi$ obtemos então a expressão [51]

$$F_z = \frac{1}{4} (e^{-\nu(0,z)} - 1). \quad (3.5)$$

Se utilizássemos o elemento de superfície $d\sigma = \sqrt{|g|} dr d\varphi = e^{2(\nu-\phi)} dr d\varphi$ a força de compressão seria dada por

$$F_z = \frac{1}{4} (1 - e^{\nu(0,z)}).$$

Note que para valores de ν pequenos as duas definições coincidem.

Estas estruturas são gravitacionalmente inertes, pois a densidade Newtoniana associada a um corpo é

$$\rho_N = \rho + p_1 + p_2 + p_3,$$

onde ρ e p_i são os autovalores do seu tensor de energia momento. Então como no caso das estruturas temos que $-\rho = p_z > 0$, conseqüentemente $\rho_N = 0$. Por outro lado $-\rho = p_z > 0$ é a equação de estado para as cordas cósmicas [49], [51], [35], [45] e [55].

3.6 Extensões Maximais

Nas seções anteriores falamos que as singularidades poderiam ser físicas ou geradas pelo sistema de coordenadas. Com os invariantes escalares podemos distinguir entre estes dois tipos de singularidades, mas além de saber que uma singularidade não é física desejamos também eliminá-la, de modo que possamos estudar as características do espaço diretamente.

Em termos topológicos, as singularidades falsas estão ligadas ao fato de que os sistemas de coordenadas são definidos como um difeomorfismo entre um aberto, contido na variedade que desejamos estudar, e um aberto contido em \mathbf{R}^n , no caso da relatividade em \mathbf{R}^4 . Elas simplesmente marcam o fim do domínio de definição do sistema de coordenadas, ou seja, são pontos em que o mapeamento deixa de ser difeomórfico. Nestas regiões somos obrigados a considerar outra carta, ou seja, outro difeomorfismo definido em um aberto que contenha esta região.

Na teoria da relatividade surge o problema de não sabermos, a priori, qual a topologia da variedade onde estamos resolvendo as equações. De modo que não podemos definir um conjunto de cartas que cubra toda a variedade e então escrever a solução em cada uma destas cartas. O que normalmente ocorre é que uma solução é obtida em um sistema de coordenadas escolhido

arbitrariamente, como no caso das soluções de Weyl, e então tentamos encontrar uma variedade que a contenha.

Um exemplo deste processo é a obtenção da métrica de Kruskal [42]. Schwarzschild resolveu as equações de Einstein impondo que ela possuisse simetria esférica obtendo o elemento de linha (2.31). Porém esta métrica possui uma singularidade de coordenadas sobre a esfera $\rho = 2m$, como podemos ver no primeiro gráfico da Figura 3.1, onde os cones de luz não interceptam a hipersuperfície $\rho = 2m$. Porém se calcularmos o tempo próprio ao longo da trajetória de uma partícula em queda livre radial vemos que ela atinge a singularidade real, em $\rho = 0$, com tempo finito.

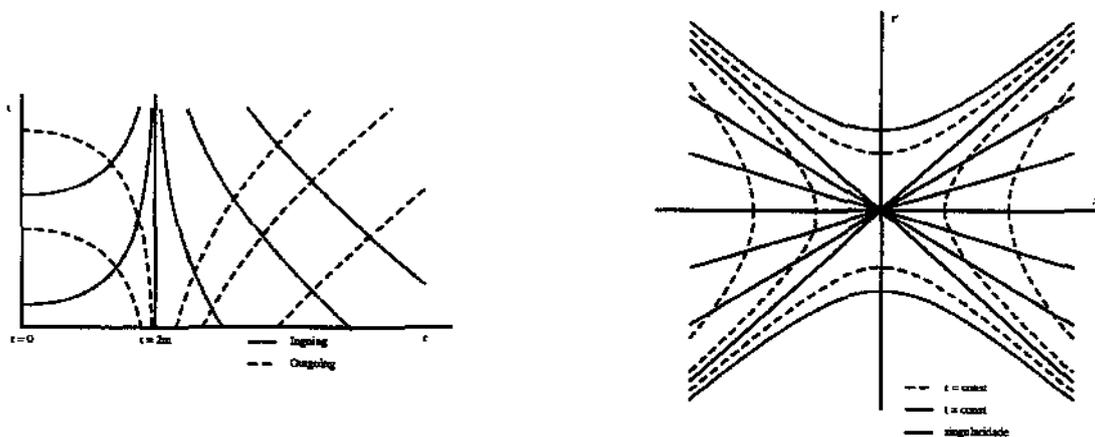


Figura 3.1: Geodésicas nulas da métrica de Schwarzschild e métrica de Kruskal.

A transformação que Kruskal [42] encontrou leva a região exterior desta métrica, $\rho > 2m$, para a região $|t'| < x'$ do plano $x' - t'$, exposta no segundo gráfico da Figura 3.1, onde as linhas tracejadas indicam as linhas $\rho = const$ e as pontilhadas $t = const$. Porém a nova métrica não é singular quando $t' = \pm x'$, o que permite a expansão de seu domínio de definição para toda a região entre as linhas que correspondem à $\rho = 0$ e que são singularidades físicas. Temos assim um espaço que contém a métrica de Schwarzschild e que não possui singularidades de coordenadas. Dizemos então que a métrica de Kruskal é a extensão maximal da métrica de Schwarzschild.

Na verdade a métrica de Kruskal é mais do que apenas a extensão maximal da métrica de Schwarzschild. Ela é também conformalmente plana, ou seja, a sub-métrica da superfície $t' - x'$ pode ser escrita como

$$ds^2 = \Omega^2 (dt^2 - dx^2),$$

onde Ω^2 é uma função não nula e positiva. Isso significa que os ângulos e proporções entre vetores na métrica de Kruskal são os mesmos da métrica de Minkowski. Conseqüentemente, as geodésicas nulas são linhas retas ortogonais entre si.

De posse da extensão maximal de uma métrica, escrita em coordenadas conformalmente planas, podemos, aplicando transformações conformais, reduzir os intervalos de definição das

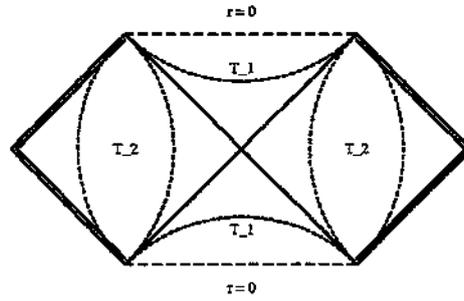


Figura 3.2: Diagrama de Kruskal.

coordenadas a intervalos finitos, obtendo os diagramas de Carter-Penrose. Para a métrica de Kruskal temos o diagrama da Figura 3.2.

Em geral encontrar o diagrama de uma métrica é uma tarefa complexa. Porém para métricas que satisfaçam algumas condições, podemos aplicar o método descrito por Walker [69].

Este método sistematiza os passos necessários para obter o diagrama de Carter-Penrose de superfícies tipo-tempo, bidimensionais, estáticas e totalmente geodésicas. Vale lembrar que uma superfície é considerada totalmente geodésica quando todas as geodésicas que a tangenciam em algum ponto estão totalmente contidas em seu interior. Daqui para frente trataremos as superfícies que satisfaçam estas restrições por T .

Como esta é uma superfície bidimensional sua métrica induzida pode ser escrita em um sistema de coordenadas tal que

$$ds^2 = F dt^2 - \frac{dr^2}{F}, \quad (3.6)$$

onde F é a norma do vetor de Killing tipo-tempo em T . Esta função depende somente da coordenada r , pois como a métrica é estática, F não pode depender de t , e como a superfície é totalmente geodésica podemos escolher um sistema de coordenadas tal que as demais coordenadas sejam constantes em T .

Para podermos determinar quais singularidades são físicas, precisamos definir um invariante escalar em T ; uma escolha adequada é a curvatura intrínseca de Gauss, que neste caso será escrita como

$$K = \frac{1}{2} \frac{d^2 F}{dr^2}. \quad (3.7)$$

A suavidade de K será exigida em todas as extensões realizadas.

Como veremos, se F e K forem suaves, para todos os valores de r todas as geodésicas em T poderão ser estendidas até a completude. Porém se uma destas funções for singular para algum valor $r = r_0$ então as geodésicas que alcançarem r_0 , com comprimento afim finito, serão incompletas e inextensíveis, enquanto os pontos onde $F = 0$ definem horizontes de Killing, implicando que as geodésicas que chegarem a estes pontos serão incompletas mas extensíveis. É por estas regiões que poderemos estender o domínio de definição da métrica.

Quando observamos o infinito $r \rightarrow \pm\infty$ devemos analisar se a métrica é assintoticamente plana ou não. No primeiro caso ela é bem comportada nesta região, enquanto o contrário indica

a existência de singularidades no infinito. Avaliando o elemento de linha (3.6) notamos que a métrica será assintoticamente plana se $F \rightarrow 1$ quando $r \rightarrow \infty$, valendo o mesmo para o infinito negativo. Porém, na verdade, basta que F aproxime-se de um valor finito e constante, pois neste caso é possível aplicar uma transformação que reescale as coordenadas, fazendo com que nas novas o espaço seja assintoticamente plano.

Para a aplicação do método o primeiro passo é identificar as raízes da função F , que trataremos por a_i , onde $i = 1, 2, 3, \dots, n$, escolhidas de modo que

$$-\infty < a_1 < a_2 < a_3 < \dots < a_n < \infty.$$

Então, definindo $a_0 = -\infty$ e $a_{n+1} = \infty$ dividimos a superfície T em $n + 1$ regiões

$$T_i = \{(t, r) \mid -\infty < t < \infty, a_i \leq r \leq a_{i+1}\}.$$

Introduziremos agora coordenadas nulas em cada uma destas regiões. Para isso fixamos r_i entre a_i e a_{i+1} . No caso em que T_i for singular escolhemos $r_i = r_0$, caso contrário r_i é arbitrário. E então definimos as transformações

$$u_i = t + \int_{r_i}^r \frac{dw}{F(w)}, \quad v_i = t - \int_{r_i}^r \frac{dw}{F(w)}, \quad (3.8)$$

que levam a métrica de T em $ds^2 = F du_i dv_i$. Este novo elemento de linha será suave em $-\infty < u_i < \infty$ e $-\infty < v_i < \infty$ se F for suave em T_i .

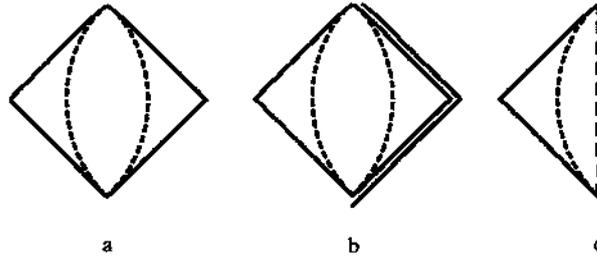


Figura 3.3: Blocos básicos.

Para obtermos o diagrama correspondente à superfície compactificamos independentemente cada região de T . Quando $F > 0$ definimos

$$\varphi_i = \tan^{-1} u_i, \quad \psi_i = \tan^{-1} v_i.$$

Esta transformação leva cada região de T em um dos blocos básicos, que são vistos na Figura 3.3.

- Se na região T_i a função F for regular mas não assintoticamente plana T_i será transformada em um bloco do tipo 'a';
- Se T_i for tal que $a_i < r < \infty$ e F for assintoticamente plana no infinito então o bloco correspondente será o 'b', onde o infinito é representado pelas linhas duplas. Se o intervalo da coordenada r for $-\infty < r < a_{i+1}$ então o bloco resultante será também o 'b' porém agora ele estará refletido horizontalmente;

- Por último, quando T_i for singular, isto é, quando ela estiver restrita ao intervalo $r_0 < r < a_{i+1}$, o bloco resultante será o 'c', onde a linha tracejada representa a singularidade $r = r_0$. Se o intervalo correspondente for $a_i < r < r_0$ então, como no caso anterior o bloco 'c' será refletido horizontalmente.

As linhas pontilhadas em cada bloco representam regiões onde $r = const$, sendo portanto órbitas do vetor de Killing.

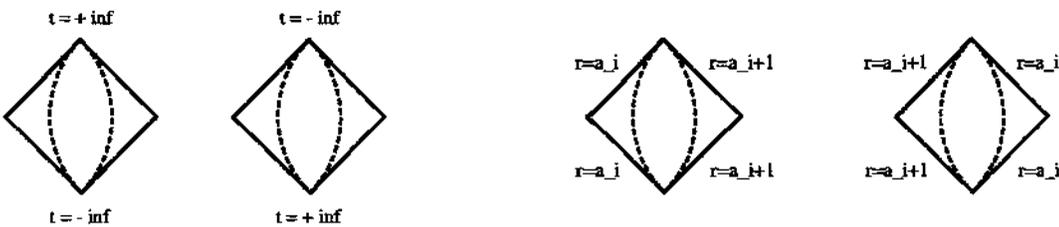


Figura 3.4: Reflexões em t e em r .

Nas regiões onde $F < 0$ as coordenadas r e t têm suas características de tempo ou espaço invertidas. Os blocos correspondentes a estas regiões são obtidos da mesma maneira, exceto pelo fato das coordenadas u_i e v_i serem substituídas por seus negativos. Portanto estes blocos serão rotacionados com relação aos blocos onde $F > 0$, isso é com a coordenada r variando verticalmente e t horizontalmente. As órbitas do vetor de Killing, que agora são tipo-espaço, servem para explicitar em quais blocos F é maior ou menor que zero.

Como conseqüência da simetria discreta $t \rightarrow -t$ podemos refletir os blocos no sentido de variação da coordenada t , veja o primeiro gráfico da Figura 3.4. Ao passo que invertendo os papéis das coordenadas u_i e v_i , na transformação (3.8), refletimos o bloco com relação a coordenada r , como no segundo gráfico da Figura 3.4. Estas reflexões serão úteis para permitir um número maior de ligações entre os blocos.

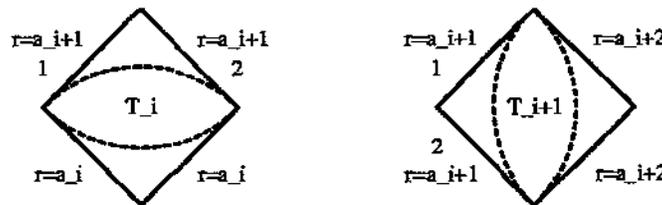


Figura 3.5: Ligações possíveis.

Tendo listado os blocos relacionados com a superfície T o próximo passo, para obter o seu diagrama de Carter-Penrose, é unir estes blocos de todas as maneiras possíveis. Porém estas ligações devem ser feitas de modo que a coordenada tipo-tempo, r ou t dependendo do bloco, sempre varie verticalmente e a curvatura intrínseca de Gauss seja suave através de todas as uniões.

Para esclarecer como estas ligações podem ser feitas apresentamos a Figura 3.5 como exemplo. Estes dois blocos podem ser ligados de duas formas distintas. A primeira unindo as arestas etiquetadas com o número 1. Note que para realizar esta ligação primeiro devemos refletir o

bloco T_i com relação à coordenada r . A outra possibilidade é unir as arestas marcadas com 2. Os dois resultados possíveis são apresentados na Figura 3.6. O diagrama 'a' é o resultado da união pelas arestas 1 e 'b' pelas arestas 2.



Figura 3.6: Unindo os blocos.

Como foi assumido que K é suave através de $r = a_{i+1}$, para mostrarmos que estes blocos podem ser unidos adequadamente basta mostrarmos que existe um sistema de coordenadas em uma vizinhança da aresta $r = a_{i+1}$.

Considerando primeiramente a configuração 1, definimos

$$du = dt + \frac{dr}{F}.$$

Então, após manipulações simples temos o elemento de linha

$$ds^2 = F du^2 - 2 du dr, \quad (3.9)$$

que é suave em $r = a_{i+1}$, permitindo assim que a coordenada r agora assumia valores em todo o intervalo (a_i, a_{i+2}) . Deste modo, o sistema de coordenadas cobre suavemente ambos os blocos. No caso da segunda configuração, basta definirmos a nova coordenada como

$$dv = dt - \frac{dr}{F},$$

obtendo o elemento de linha

$$ds^2 = F dv^2 - 2 dv dr, \quad (3.10)$$

que cobre agora os dois blocos unidos pela segunda configuração.

Percebendo que ambas as ligações descritas podem ser realizadas simultaneamente podemos montar o diagrama apresentado na Figura 3.7. Além disso, podemos refletir o bloco T_{i+1} em relação à coordenada r e acrescentá-lo ao lado esquerdo do diagrama, que passa assim a conter quatro blocos básicos. Construindo o diagrama desta forma obtemos a maior extensão possível, com estes dois blocos. Porém, este arranjo introduz um problema: o ponto p , no centro do diagrama, não é coberto por nenhum dos sistemas de coordenadas definidos até agora. Precisamos então de um novo sistema de coordenadas capaz de mapear uma vizinhança deste ponto.

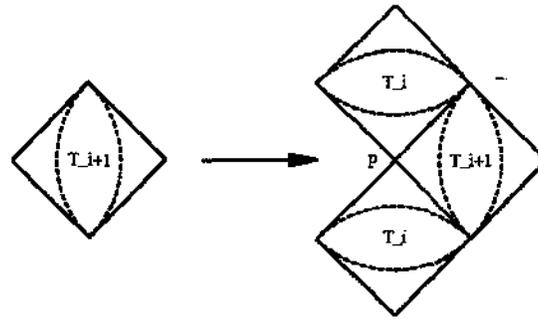


Figura 3.7: Ligações simultâneas.

Walker demonstra que este sistema de coordenadas existe, desde que a função F possa ser escrita como $H^{-1}(r - a_1)(r - a_2) \dots (r - a_n)$, onde H é um polinômio em r de grau n cujas raízes não coincidam com as de F . Para funções mais gerais a existência deste sistema fica em aberto. Estas novas coordenadas são definidas como

$$U = \alpha e^{\beta t} \exp\left(\beta \int \frac{dr}{F}\right), \quad V = \alpha^{-1} e^{-\beta t} \exp\left(\beta \int \frac{dr}{F}\right),$$

onde α é uma constante de integração e β é determinada de forma que o elemento de linha seja regular em todo o interior do diagrama. A restrição em F foi imposta justamente para garantir a existência desta constante. Deste modo o elemento de linha no interior do diagrama passa a ser escrito como

$$ds^2 = \frac{F}{\beta^2} \exp\left(-2\beta \int \frac{dr}{F}\right) dU dV.$$

Apresentar um sistema de coordenadas equivalente para o caso de diagramas com mais blocos torna-se difícil, talvez impossível, pois temos que impor mais condições de regularidade para cada ponto de interseção contido no interior do diagrama. Entretanto este sistema não é necessário, pois podemos cobrir todo o diagrama, empregando uma carta para cada conjunto de quatro blocos, montando assim um mapa para toda a variedade.

Reverendo os passos descritos temos:

- Escrever a métrica da superfície T em um sistema de coordenadas em que seu elemento de linha assuma a forma da expressão (3.6);
- Então de posse da fórmula da função F podemos avaliar suas raízes e seus pontos singulares. O que nos dá a divisão da superfície T nas regiões T_i ;
- Basta agora determinar o bloco correspondente a cada região e uni-los convenientemente.

Tendo apresentado o método apresentamos agora um exemplo, construindo o diagrama de Carter-Penrose para a métrica de Schwarzschild.

Tomando esta métrica em sua forma canônica (2.31) e nos restringindo à superfície $t = r$, vemos que seu elemento de linha já se encontra na forma desejada (3.6)

$$ds^2 = \left(1 - \frac{2m}{r}\right) dt^2 - \left(1 - \frac{2m}{r}\right)^{-1} dr^2.$$

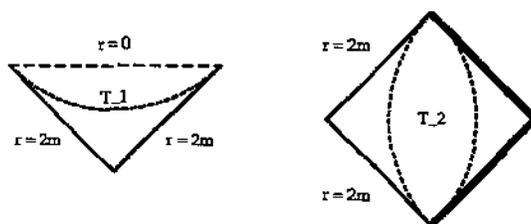


Figura 3.8: Blocos básicos da métrica de Schwarzschild.

De onde extraímos imediatamente as expressões

$$F = 1 - \frac{2m}{r}, \quad K = \frac{2m}{r^3}.$$

É fácil ver que a raiz de F é dada por $a_1 = 2m$ e que $r = 0$ é uma singularidade real. Desta forma, a superfície T é dividida em duas regiões, cujos blocos correspondentes estão na Figura 3.8. Reproduzindo os passos descritos para a união dos blocos obtemos diretamente o diagrama da Figura 3.2.

Soluções de Weyl

4.1 Introdução

Neste capítulo e no próximo apresentamos várias soluções de Weyl e algumas de suas características. Neste incluímos aquelas que estão associadas a distribuições de massa isoladas e no próximo sobreposições destas.

Outras revisões das soluções de Weyl podem ser encontradas na literatura científica, podemos citar como exemplos: Bonnor [12], Bonnor e Martins [10], Letelier e Oliveira [51] e Kramer et. al. [41]. Porém estas revisões são muito restritas.

Na próxima seção comentamos sobre os métodos numéricos, que foram empregados para a obtenção de algumas novas soluções. Os cálculos necessários para exibir as demais soluções não foram incluídos. Porém estão indicadas as referências onde eles podem ser encontrados.

Soluções Numéricas

Para obtermos uma solução de Weyl precisamos resolver a equação de Laplace (2.7) e integrar (2.8). Como a equação de Laplace é uma das equações clássicas da física-matemática suas soluções são bem conhecidas e pela sua linearidade podemos obter a sobreposição de soluções de modo trivial. Porém a integral (2.8) não é linear podendo representar um obstáculo para obter novas soluções analíticas.

Porém conhecendo ϕ analiticamente é fácil integrar (2.8) numericamente. Temos então uma ferramenta útil e simples para gerar novas soluções, permitindo o estudo de características das métricas de Weyl, tais como estruturas e membranas.

Entretanto, ao computar ν numericamente o que temos são seus valores para determinados pontos de uma malha, o que é suficiente para que possamos descrever seu comportamento em, praticamente, qualquer região desejada. Porém ficamos presos as coordenadas de Weyl, o que torna inviável um estudo de suas características globais, que em geral dependem de transformações de coordenadas singulares.

Como os resultados numéricos são apresentados graficamente incluímos também os gráficos das soluções analíticas para que possamos compará-los. Como os métodos numéricos que foram empregados são muito simples e de conhecimento geral, incluímos sua descrição apenas no Apêndice D, bem com as listagens dos programas implementados.

4.2 Campo Homogêneo

Esta é a mais simples solução de Weyl

$$\phi = \lambda z, \quad \nu = -\frac{\lambda^2 r^2}{2}. \quad (4.1)$$

Na gravitação Newtoniana ϕ é o potencial para um plano no infinito. Porém ele pode ser transformado em uma métrica que representa uma onda gravitacional, como veremos na Seção 5.8.

4.3 Solução de Chazy-Curzon

Resolvendo o Laplaciano para uma fonte pontual e calculando ν pela integral (2.8), obtemos a solução de Weyl conhecida como solução de Chazy-Curzon [17], [22]

$$\phi = -\frac{m}{R}, \quad \nu = -\frac{m^2 r^2}{2R^4}, \quad (4.2)$$

aqui $R^2 = r^2 + (z - z_0)^2$ e z_0 é a posição, no eixo z , onde ϕ é singular.

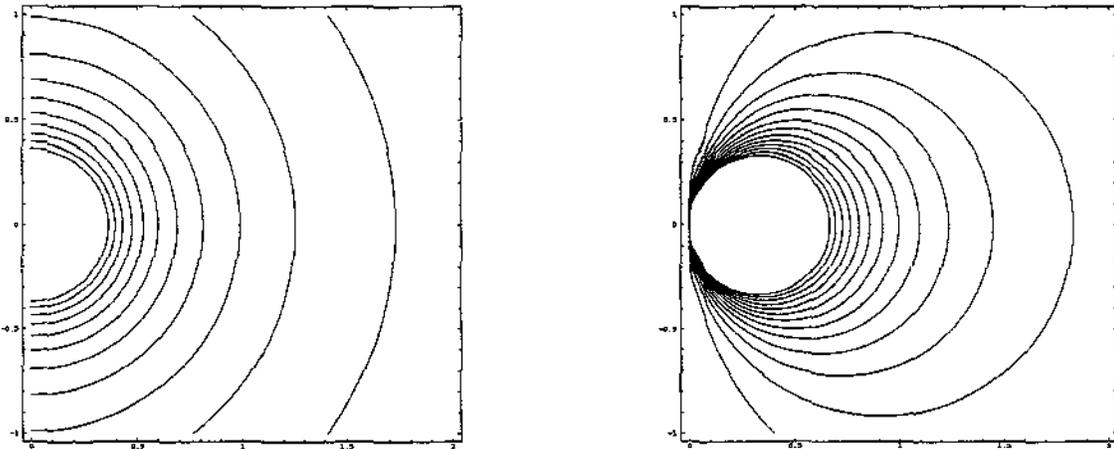


Figura 4.1: Funções $\phi(r, z)$ e $\nu(r, z)$ da métrica de Curzon.

Na Figura 4.1 estão os gráficos das funções (4.2). Mesmo com a função ϕ sendo o potencial Newtoniano para uma massa pontual, quando observamos o gráfico de ν , fica evidente que esta solução não contém a uma distribuição esférica de matéria. O que será confirmado mais adiante ao analisarmos seus invariantes e suas geodésicas. Esta solução é tratada por partícula de Curzon.

Para obtemos alguma informação sobre as propriedades físicas deste espaço precisamos calcular seus invariantes escalares. Aplicando as fórmulas apresentadas no Capítulo 3 obtemos o escalar de Kretschmann (3.2), onde fixamos $z_0 = 0$,

$$K = \frac{16m^2 e^{2\psi}}{R^{12}} (Ar^2 + 9Br^2 z^2 + 3Dz^4 - 3mER),$$

$$\psi = \frac{m}{R^4} (mr^2 - 2R^3),$$

$$A = m^4 + 6m^2r^2 + 3r^4, \quad B = m^2 + r^2 + z^2,$$

$$D = m^2 + z^2, \quad E = m^2r^2 + 2r^4 + 4r^2z^2 + 2z^4$$

e o invariante cúbico (3.3), também com $z_0 = 0$,

$$N = \frac{48m^3e^{3\psi}}{R^{27}} (r^4 + 2r^2z^2 + z^4 - mR^3) \Phi,$$

$$\Phi = 2r^{14} + Ar^{12} + Br^{10} + 2z^2Cr^8 + 2z^4Dr^6 + z^6Er^4 + Fr^2 + G,$$

$$B = m^4 + 27m^2z^2 + 42z^4, \quad C = 2m^4 + 30m^2z^2 + 35z^4,$$

$$D = 3m^4 + 35m^2z^2 + 35z^4, \quad E = 4m^4 + 45m^2z^2 + 42z^4,$$

$$A = 5m^2 + 14z^2, \quad G = 2z^{12} (m^2 + z^2) - 4mR^{13},$$

$$F = z^8 (m^4 + 15m^2z^2 + 14z^4) - 3m^3R^9$$

e a função ϕ é a mesma que foi definida para o invariante de Kretschmann.

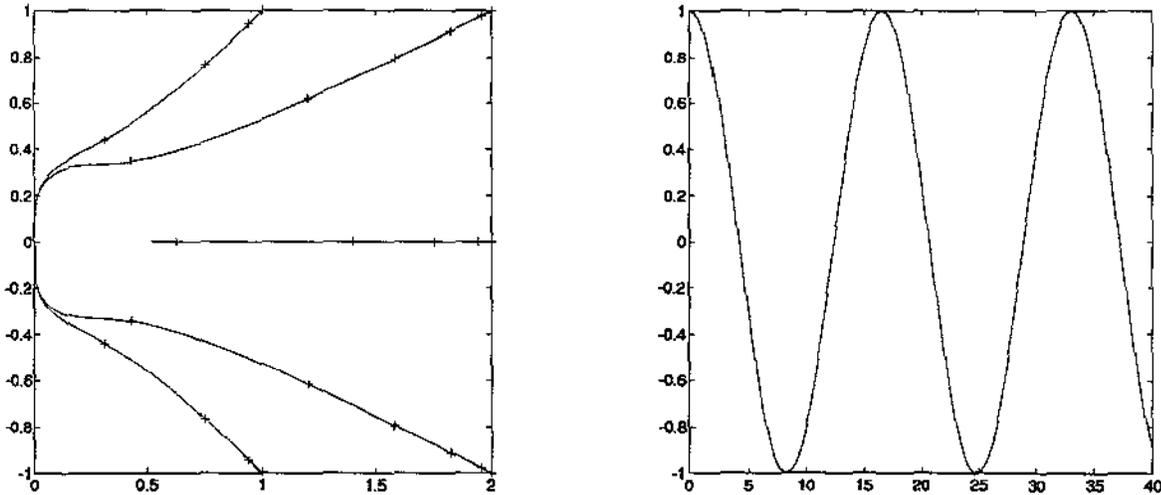


Figura 4.2: Geodésicas da métrica de Curzon.

Se adotarmos coordenadas esféricas (B.8) estes escalares são reescritos como

$$K = \frac{8m^2e^{2\psi}}{R^{10}} (6R^4 - 12mR^3 + 3(A+2)m^2R^2 - 3Am^3R + Am^4),$$

$$N = \frac{24m^3e^{3\psi}}{R^{14}} (m-R) (-4R^4 + 8mR^3 - (3A+4)m^2R^2 + 3Am^3R - Am^4)$$

sendo que agora ψ e A são definidas como

$$\psi = \frac{m}{R^2} (\pi \sin^2 \theta - 2R), \quad A = 1 - \cos 2\theta.$$

Uma característica destes escalares é que a singularidade na origem é direcional, ou seja, ela é dependente da trajetória de aproximação. Por exemplo se nos aproximarmos por uma linha reta, diferente do eixo z , os escalares tornam-se infinitos. Porém se nos aproximarmos pelo eixo z eles se anulam. Uma possível interpretação para estes resultados é que a solução de Curzon esteja associada a métrica de um anel colapsado. Quando traçarmos as geodésicas desta métrica, essa característica se tornará mais clara.

No primeiro gráfico da Figura 4.2 apresentamos as trajetórias espaciais das geodésicas tipo-tempo que partiram do repouso. As marcas indicam a passagem de uma unidade do tempo próprio. Todas estas curvas terminam abruptamente, isto é, os métodos numéricos são interrompidos ou geram resultados absurdos, indicando que eles sentiram a presença da singularidade.

Quando soltamos uma partícula exatamente sobre o eixo z ela cairá viajando sobre este eixo. Como os invariantes de curvatura se anulam quando nos aproximamos da singularidade pelo o eixo z , a partícula ignora a singularidade. No segundo gráfico confrontamos a coordenada z desta trajetória com o seu tempo próprio, evidenciando seu comportamento oscilatório. O comportamento desta geodésica consiste em mais um argumento para a interpretação desta métrica como um anel cujo raio foi colapsado. Scott e Szekeres [64], [65] estudaram o comportamento das geodésicas espaciais e globais desta métrica.

4.4 Barra de Comprimento Finito

A solução cuja imagem Newtoniana é uma barra de comprimento $2l$, sobre o eixo z com extremidades em $z = \pm l$, é definida pelas funções [62]

$$\phi = \frac{m}{2l} \ln \frac{R_1 + R_2 + 2l}{R_1 + R_2 - 2l}, \quad \nu = \frac{m^2}{2l^2} \ln \frac{(R_1 + R_2)^2 - 4l^2}{4R_1 R_2}, \quad (4.3)$$

onde R_a são as distâncias Euclidianas dos extremos da barra ao ponto em que se está calculando o potencial e a constante m está associada a massa do barra. Algumas vezes esta solução também é chamada de barra de Weyl, solução γ de Weyl ou solução de Zipoy-Voorhees.

4.5 Solução de Schwarzschild

Um caso particular, de grande interesse, das soluções (4.3) ocorre quando tomamos $l = m$, ou seja, o comprimento é igual a duas vezes a massa do corpo. Esta nova solução é escrita como

$$\phi = \frac{1}{2} \ln \frac{R_1 + R_2 - 2m}{R_1 + R_2 + 2m}, \quad \nu = \frac{1}{2} \ln \frac{(R_1 + R_2)^2 - 4m^2}{4R_1 R_2}. \quad (4.4)$$

Esta métrica é importante, pois representa a solução de Schwarzschild nas coordenadas de Weyl [9] [44] [46], como foi dito na Seção 2.3, a transformação (2.32) leva essa métrica no

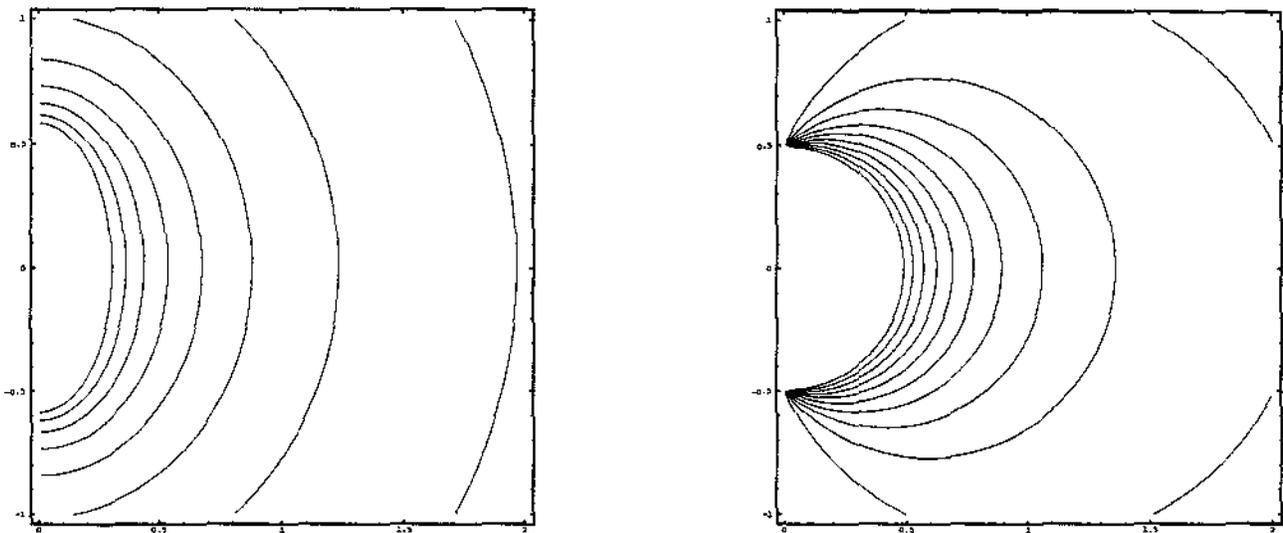


Figura 4.3: Solução de Schwarzschild em coordenadas de Weyl.

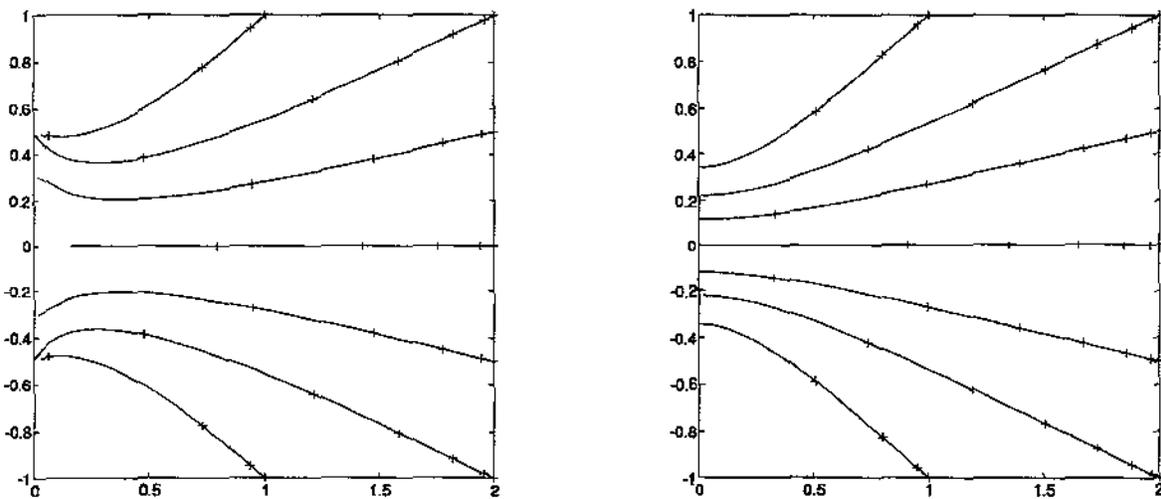


Figura 4.4: Geodésicas da barra de Weyl e da solução de Schwarzschild.

exterior da métrica de Schwarzschild. Este fato serve como alerta quanto a interpretação de uma métrica de Weyl com base em sua Imagem Newtoniana.

Podemos também escrever os potenciais dessa métrica como

$$\phi = \frac{1}{2} \ln \left(\frac{\mu_1}{\mu_2} \right), \quad \nu = \frac{1}{2} \ln \left(\frac{r^2 + \mu_1 \mu_2}{(\mu_1^2 + r^2)(\mu_2^2 + r^2)} \right)$$

através das definições

$$\mu_1 = m - z + \sqrt{(m - z)^2 + r^2}, \quad \mu_2 = -m - z + \sqrt{(m + z)^2 + r^2}.$$

Embora a solução para uma barra de densidade arbitrária possa parecer muito parecida com a solução de Schwarzschild, que correspondente a uma barra com densidade igual a 1/2, esta semelhança é um mero artifício do sistema de coordenadas. A diferença fica evidente quando

traçamos as trajetórias espaciais das geodésicas tipo-tempo, vistas na Figura 4.4. Note que no primeiro gráfico, onde foi considerada uma barra de densidade igual a $1/4$, as trajetórias sempre se dirigem aos extremos da barra, situados nos pontos $\pm 1/2$, indicando a presença de matéria repulsiva no interior da barra. No segundo gráfico, relativo a uma solução de Schwarzschild, também entre os pontos $\pm 1/2$, as geodésicas incidem diretamente sobre a barra, indicando que ela é composta de matéria normal.

4.6 Linha Infinita

Tomando o potencial de uma linha infinita com densidade constante σ

$$\phi = 2\sigma \ln r + \ln C, \quad (4.5)$$

onde C é uma constante arbitrária, e integrando (2.8) temos

$$\nu = 4\sigma^2 \ln r. \quad (4.6)$$

Esta é a solução de Levi-Civita, e seu elemento de linha [10] [11] é

$$ds^2 = C^2 r^{4\sigma} dt^2 - r^{(8\sigma^2 - 4\sigma)} (dr^2 + dz^2) - C^{-2} r^{2-4\sigma} d\varphi^2. \quad (4.7)$$

Seus invariantes de Kretschmann e Cúbico são

$$K = 64P\sigma^2(2\sigma - 1)^2 r^{-4P}, \quad N = -768\sigma^4(2\sigma - 1)^4 r^{-6P},$$

onde definimos $P = 4\sigma^2 - 2\sigma + 1$.

Note que para $\sigma = 0$ ou $\sigma = 1/2$ estes invariantes anulam-se, ou seja, para estes valores estamos lidando com um espaço-tempo plano. Para outros valores de σ eles tendem ao infinito quando $r \rightarrow 0$ e tendem a zero quando $r \rightarrow \infty$.

Na Seção 4.14 listaremos algumas métricas obtidas por Kinnersley, que também são soluções de Weyl, embora em outros sistemas de coordenadas. Entre elas existe uma que é exatamente a solução para uma linha infinita com densidade $\sigma = 1/4$, e $C = 1$.

4.7 Linha Semi-Infinita

A solução para uma linha semi-infinita de densidade constante, σ , que parte do ponto z_0 e estende-se ao infinito é dada por

$$\phi = \sigma \ln X + \ln C, \quad \nu = 2\sigma^2 \ln \frac{X}{2R} \quad (4.8)$$

a função X é definida como $X = R + \epsilon(z - z_1)$, onde a constante ϵ assume o valor 1 quando a linha encontra-se nos valores de z maiores que z_0 e -1 caso contrário, enquanto C é arbitrária.

Substituindo estas funções no elemento de linha (2.6) temos

$$ds^2 = C^2 X^{2\sigma} dt^2 - \frac{X^{(4\sigma^2 - 2\sigma)}}{(2R)^{4\sigma^2}} (dr^2 + dz^2) - C^{-2} X^{-2\sigma} r d\varphi^2. \quad (4.9)$$

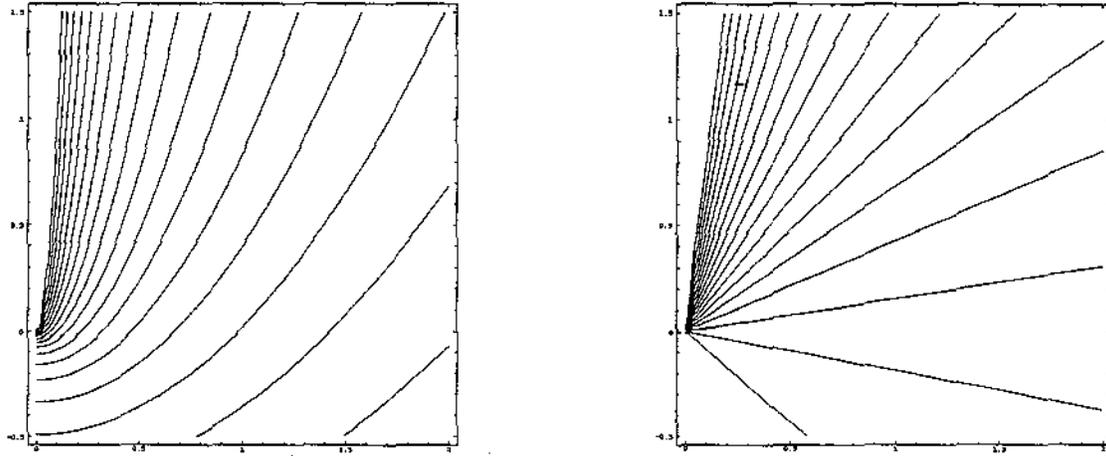


Figura 4.5: Potenciais ϕ e ν de uma linha semi-infinita.

Quando $\sigma \neq 1$, podemos definir um sistema de coordenadas equivalente onde a constante C seja igual a unidade. Chegamos a esse novo sistema pela transformação

$$z = C^{(1-\sigma)^{-1}} z^*, \quad r = C^{(1-\sigma)^{-1}} r^*, \quad t = C^{-(1-\sigma)^{-1}} t^*,$$

que coloca os potenciais métricos em sua forma mais conhecida

$$\phi = \sigma \ln X^*, \quad \nu = 2\sigma^2 \ln \frac{X^*}{2R^*}. \quad (4.10)$$

Para esta métrica os invariantes escalares definidos na Seção 3.2 resultam em

$$K = 12G^2 \left(1 - \frac{2}{3}Y\right), \quad N = -12G^3(1 - Y),$$

onde as funções G e Y são definidas como

$$G = \frac{2^{2\sigma} C^2 \sigma (2\sigma - 1)}{R^{2(1-\sigma)}} \left(\operatorname{cosec} \frac{\theta}{2} \right)^{2(4\sigma^2 - 2\sigma + 1)}, \quad Y = (1 - \sigma)(1 + 2\sigma) \cos^2 \frac{\theta}{2}.$$

O ângulo θ é medido entre a linha e o segmento de reta que une o extremo da linha com o ponto onde estamos calculando os invariantes.

Bonnor e Martins [10] fazem um estudo das características desta métrica em função de σ , obtendo os resultados:

$\sigma = 0$ Os invariantes escalares se anulam e estamos lidando com um espaço-tempo plano, na verdade a métrica se reduz ao espaço de Minkowski;

$\sigma = 1/2$ Os invariantes também se anulam, mas neste caso estamos lidando com um espaço-tempo uniformemente acelerado, este aspecto será tratado com mais detalhes na próxima sub-seção;

$\sigma = -1/2$ Neste caso podemos identificar, esta métrica, com a solução de Taub, pela transformação

$$r = Z\rho \quad 2\epsilon(z - z_0) = Z^2 - \rho^2;$$

$\sigma > 1$ Quando $R \rightarrow \infty$ os invariante divergem indicando a presença de fontes adicionais no infinito.

Caso contrário os invariantes divergem apenas sobre a linha. Então se quisermos uma solução assintoticamente plana devemos então impor que $\sigma \leq 1$, ou no sistema de unidades internacional $\sigma \leq 10^{28} \text{ g/cm}$.

Espaço-Tempo Acelerado

Como foi dito a métrica de um linha semi-infinita com densidade $\sigma = 1/2$ caracteriza um espaço-tempo uniformemente acelerado. Podemos formalisar esta afirmação aplicando uma transformação de coordenadas que leve (4.9) em um espaço de Minkowski.

Reescrevendo as expressões para as funções ϕ e ν com $\sigma = 1/2$ temos

$$\phi = \frac{1}{2} \ln AX, \quad \nu = \frac{1}{2} \ln \frac{X}{2R} \quad (4.11)$$

a constante A é definida como $A = C^2$ e sua unidade é $(\text{comprimento})^{-1}$ assim podemos associá-la com uma aceleração. A constante aditiva na função ν foi escolhida de modo que o eixo z seja regular fora da linha. Se fixarmos $z_0 = (2\epsilon A)^{-1}$ então a aceleração na origem terá intensidade igual a A .

Em coordenadas cilíndricas a métrica de Minkowski tem a forma

$$ds^2 = d\tau^2 - d\zeta^2 - d\eta^2 - \eta^2 d\varphi^2$$

e a transformação que leva (4.11) nesta métrica é [7] [20]

$$z - a = \frac{B}{2} (\eta^2 + \tau^2 - \zeta^2), \quad \tau = B\eta\sqrt{\zeta^2 - \tau^2}, \quad \tanh Bqt = \frac{\tau}{\zeta} \quad (4.12)$$

a coordenada φ é mantida inalterada, a e B são constantes e q é um fator de escala para a coordenada t que será útil para acomodar condições de regularidade, como veremos na Seção 5.8.

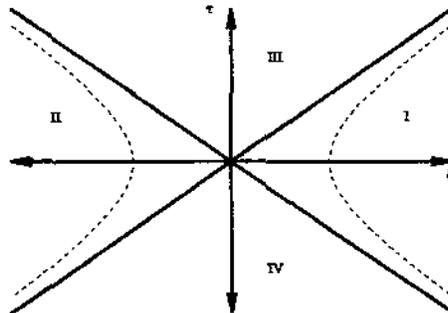


Figura 4.6: História de uma partícula originalmente no eixo z fora da linha semi-infinita.

Pela expressão (4.12) vemos que a história de um ponto fixo no eixo- z , fora da linha, aparece nas novas coordenadas como duas linhas dadas por

$$\tau = \pm \sqrt{\frac{2}{B}(z - a) + \zeta^2}, \quad \eta = 0, \quad (4.13)$$

como mostra a Figura 4.6. Por outro lado os pontos sobre a linha com $|t| < \infty$ são mapeados em $\tau = \zeta = 0$ embora quando $t = \pm\infty$ serão mapeados em $\zeta = \pm\tau$. Em todos os casos $\eta = 0$, pois para qualquer valor de t a restrição $\tau = 0$ implica em $\eta = 0$.

A transformação (4.12), além de evidenciar a ausência de curvatura desta métrica, estende seu domínio. Observe que o plano $\tau \in (0, \infty)$ e $t \in (-\infty, \infty)$ é mapeado, devido a multivaloração em (4.12), simultaneamente nas regiões I e II, da Figura 4.6. Além disso, nenhum ponto do plano original é levado as regiões III e IV, caracterizadas por $|\tau| > |\zeta|$. Como a métrica resultante é regular nas retas $\tau^2 = \zeta^2$, para qualquer valor não nulo de q , podemos tomar todo o plano $\tau - \zeta$ como domínio desta métrica, que não é mais estática, mas é invariante sob as transformações de Lorentz no plano $\zeta - \tau$. Além de ser simétrica com relação as reflexões das coordenadas $\zeta \rightarrow -\zeta$ e $\tau \rightarrow -\tau$ e as rotações em torno do eixo $\eta = 0$.

Como veremos na Seção 5.8 esta métrica pode ser usada para gerarmos soluções exatas das equações de Einstein para corpos acelerados.

4.8 Discos

Solução Geral Newtoniana

Na teoria Newtoniana da gravitação, qualquer disco, cuja matéria é distribuída de modo axialmente simétrico, possui um campo gravitacional dado por [4]

$$\phi = - \sum_{n=0}^{\infty} C_{2n} P_{2n}(\eta) q_{2n}(\xi), \quad (4.14)$$

onde empregamos as coordenadas esferoidais oblatas (B.30), $q_{2n}(\xi) = i^{2n+1} Q_{2n}(i\xi)$ e C_{2n} são constantes arbitrárias.

A densidade Newtoniana associada com cada um destes discos é $\mu = S(r)\delta(z)$, sendo que

$$S(r) = 2\pi a \left[1 - \left(\frac{r}{a} \right)^2 \right]^{-\frac{1}{2}} \sum_{n=0}^{\infty} \alpha_n P_{2n} \left(\sqrt{1 - \frac{r^2}{a^2}} \right), \quad (4.15)$$

onde a constante α é definida como $\alpha_n = (2n + 1)C_{2n}q_{2n}(0)$.

Tensor de Energia Momento de um Disco Fino

Considerando um disco infinitesimalmente fino sobre o plano $z = 0$ descrito pelas funções métricas ϕ e ν , as componentes não nulas de seu tensor de energia momento serão [58] [46]

$$\epsilon = -T^t_t = 4e^{2(\phi-\nu)}(1 - r\phi_r)\phi_z\delta(z),$$

$$p_{\varphi\varphi} = T^{\varphi}_{\varphi} = 4e^{2(\phi-\nu)}r\phi_r\phi_z\delta(z).$$

Uma maneira de lidarmos com a restrição de estaticidade, imposta pelo formalismo das soluções de Weyl, é pensarmos no disco como sendo composto por inúmeras partículas que giram em diferentes sentidos, de modo que o momento angular do disco, como um todo, seja

nulo. Estes discos serão chamados de discos em contrarotação¹. Com base neste modelo podemos calcular a velocidade de rotação de cada partícula, que compõe o disco

$$v^2 = \frac{p_{\varphi\varphi}}{\epsilon} = \frac{r\phi_r}{1 - r\phi_r}. \quad (4.16)$$

Uma solução fisicamente viável deve satisfazer a condição fraca de energia $\epsilon > 0$ como também a condição dominante de energia $v^2 \leq 1$.

Discos de Morgan-Morgan

Thomas Morgan e Lesley Morgan [58] definiram uma família de discos em 1969 com o intuito de estudar o desvio para o vermelho em suas proximidades. Estes discos foram definidos tomando-se a fórmula geral (4.14) e escolhendo constantes C_{2n} tais que

$$C_{2n} = \frac{(-1)^{n+1}(4n+1)(2m+1)!}{2^{2n+1}(2n+1)(2m-2n)!} \frac{\Gamma(m-n+\frac{1}{2})}{\mathcal{Q}_{2n+1}(0)\Gamma(m+n+\frac{3}{2})} \frac{M}{a} \quad (4.17)$$

para todo $n \leq m$ e $C_{2n} = 0$ quando $n > m$, onde m é um número inteiro positivo e fixo. Desta maneira os potenciais e densidades Newtonianas são dadas por

$$\phi = - \sum_{n=0}^m i^{2n+1} C_{2n} \mathcal{P}_{2n}(\eta) \mathcal{Q}_{2n}(i\xi), \quad (4.18)$$

$$S^{D_m}(r) = \frac{(2m+1)M}{2\pi a^2} \left[1 - \left(\frac{r}{a}\right)^2 \right]^{m-1/2} \quad (r \leq a), \quad (4.19)$$

onde a é o raio do disco, os gráficos das densidades dos primeiros 4 discos estão na Figura 4.7. Note que a densidade associada ao disco D_0 diverge quando $r \rightarrow 1$, implicando que este disco não é fisicamente aceitável. Embora suas geodésicas sejam bem comportadas, como a Figura 4.10 mostrará.

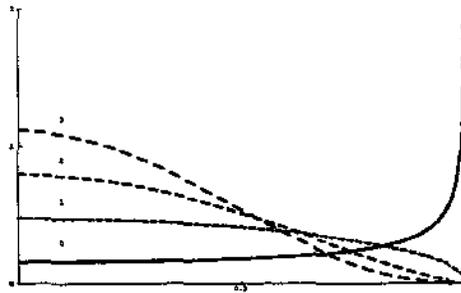


Figura 4.7: Densidades dos primeiros 4 discos de Morgan-Morgan.

Os potenciais e as densidades Newtonianas, destes discos, estão relacionados pelas relações de recorrência [48]

$$\phi^{D_{n+1}} = \frac{2n+3}{a^{2n+3}} \int a^{2n+2} \phi^{D_n} da, \quad (4.20)$$

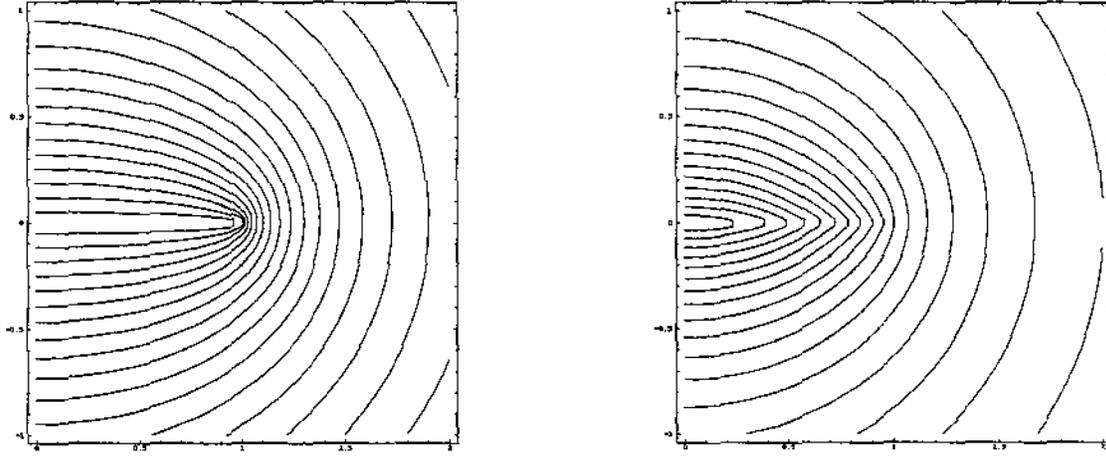


Figura 4.8: A função ϕ dos 2 primeiros discos de Morgan-Morgan.

$$S^{D_{n+1}} = \frac{2n+3}{a^{2n+3}} \int a^{2n+2} S^{D_n} da. \quad (4.21)$$

Explicitamente o potencial para os primeiros três discos é dado por

$$\phi^{D_0} = -\frac{M}{a} \operatorname{arccot} \xi, \quad (4.22)$$

$$\phi^{D_1} = -\frac{M}{a} \left\{ W + \frac{1}{4} [(3\xi^2 + 1)W - 3\xi] (3\eta^2 - 1) \right\}, \quad (4.23)$$

$$\phi^{D_2} = -\frac{M}{a} \left\{ W + (2\eta^2 - 1)K + (35\eta^4 - 30\eta^2 + 3)L \right\}, \quad (4.24)$$

sendo que $W = \operatorname{arccot} \xi$ e

$$K = \frac{5}{14} [(3\xi^2 + 1) \operatorname{arccot} \xi - 3\xi],$$

$$L = \frac{3}{448} \left[(35\xi^4 + 30\xi^2 + 3) \operatorname{arccot} \xi - 35\xi^2 + \frac{55}{3}\xi \right].$$

Em coordenadas cilíndricas estes potenciais são reescritos como [48]

$$\phi^{D_0} = -\frac{M}{a} \operatorname{Im} \ln \mu, \quad (4.25)$$

$$\phi^{D_1} = -\frac{3M}{2a^3} \operatorname{Im} \left[\left(a^2 + z^2 - \frac{r^2}{2} \right) \ln \mu + \frac{3z + ia}{2} R \right], \quad (4.26)$$

$$\phi^{D_2} = -\frac{15M}{4a^5} \operatorname{Im} \left[A \ln \mu - BR - \frac{1}{4} CR^3 \right], \quad (4.27)$$

onde $R^2 = (ia - z)^2 + r^2$, $\mu = ia - z + R$ e

$$A = \frac{1}{2} \left(a^4 - \frac{r^4}{8} - z^4 + z^2 r^2 \right) + \left(z^2 - \frac{r^2}{2} \right) \left(a^2 + z^2 - \frac{r^2}{2} \right),$$

¹Em inglês dizemos que o disco é composto por partículas em *Counterrotation*

$$B = \frac{1}{8}(ia - z)^3 + (ia - z) \left(3z^2 - \frac{r^2}{16} \right) - zr^2,$$

$$C = \frac{25}{3}z + ia.$$

Os gráficos de ϕ_0 e ϕ_1 podem ser vistos na Figura 4.8.

Discos de Morgan-Morgan Relativísticos

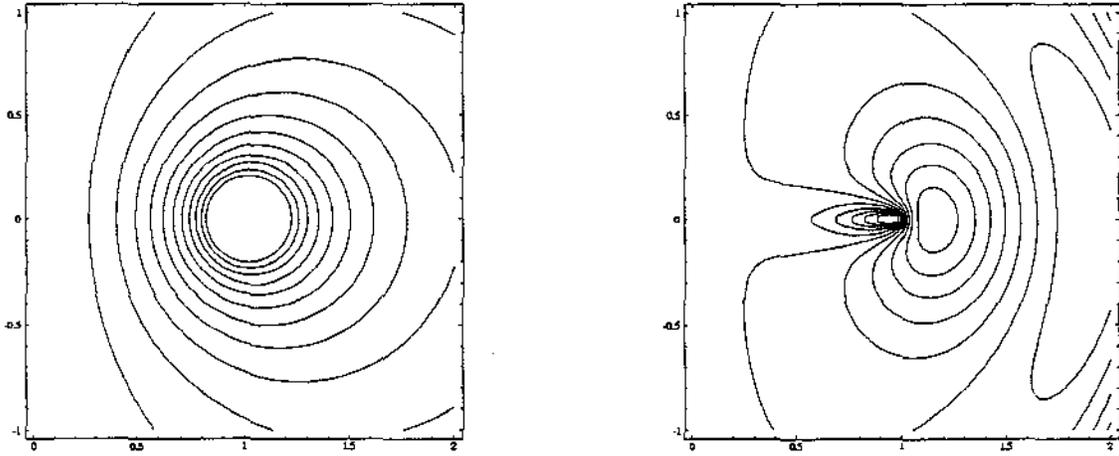


Figura 4.9: A função ν dos 2 primeiros discos de Morgan-Morgan.

Partindo da solução definida em (4.18) podemos calcular a solução de Weyl correspondente. Para as funções ϕ (4.22) e (4.23) as funções ν associadas são [48]

$$\nu[\phi^{D_0}] = -\frac{M^2}{a^2} \ln \frac{r^2 + |\mu|^2}{|r^2 + \mu^2|}, \quad (4.28)$$

$$\nu[\phi^{D_1}] = -\frac{9M^2r^2}{4a^4} \left[\frac{r^2B^2}{a^2} - (1 - \eta^2) (A^2 - 2\xi AB) \right], \quad (4.29)$$

onde definimos $A = \xi \operatorname{arctot} \xi - 1$ e $B = \frac{1}{2} \left[\frac{\xi}{1 + \xi} - \operatorname{arctot} \xi \right]$.

Os gráficos destas funções podem ser vistos na Figura 4.9. É interessante notar que o comportamento das funções é radicalmente diferente. Indicando que o segundo disco não é perturbação do primeiro.

Em coordenadas cilíndricas ϕ^{D_1} é dado por (4.26), e a expressão para $\nu[\phi^{D_1}]$ no plano $z = 0$ torna-se

$$\phi = \phi^{D_1} = -\frac{3\pi}{4} \left(1 - \frac{r^2}{2a^2} \right),$$

$$\nu = \nu[\phi^{D_1}] = \frac{9M^2r^2}{4a^4} \left[\left(1 + \frac{\pi^2}{16} \right) \frac{r^2}{a^2} - 2 \right].$$

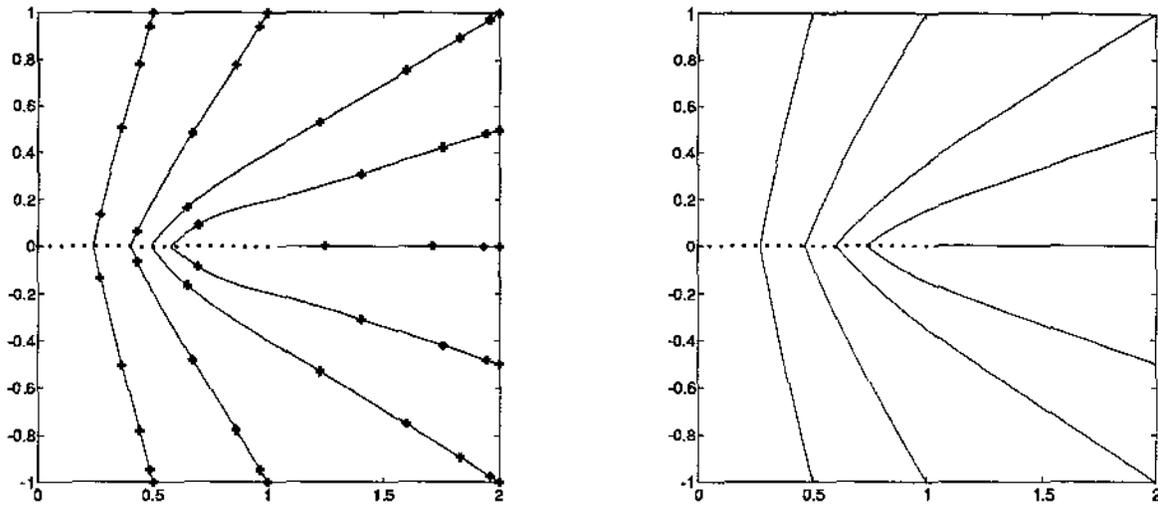


Figura 4.10: Geodésicas e queda Newtoniana no primeiro disco de Morgan-Morgan ϕ^{D_0} .

Podemos então calcular as componentes não nulas do tensor de energia momento deste disco

$$\epsilon = \sqrt{1 - \frac{r^2}{a^2}} \left[1 - \frac{3\pi}{4} \frac{Mr^2}{a^3} \right] F\delta(z), \quad p_{\varphi\varphi} = \sqrt{1 - \frac{r^2}{a^2}} \frac{3\pi}{4} \frac{Mr^2}{a^3} F\delta(z),$$

onde $F = 3M(2\pi a)^{-1}e^{2(\phi-\nu)}$. Calculando a velocidade de rotação (4.16) obtemos

$$v^2 = \frac{3\pi}{4} \frac{Mr^2}{a^3} \left(1 - \frac{3\pi}{4} \frac{Mr^2}{a^3} \right)^{-1}.$$

A condição fraca de energia, $\epsilon > 0$, exige que

$$\frac{M}{a} < \frac{4}{3\pi}. \quad (4.30)$$

Além disso notando que a maior velocidade ocorrerá na borda $r = a$ e impondo que $v^2 \leq 1$ então teremos a restrição

$$\frac{M}{a} < \frac{2}{3\pi}, \quad (4.31)$$

que é mais forte que (4.30) e é chamada de condição de Morgan-Morgan [58].

No primeiro gráfico da Figura 4.10 apresentamos as projeções espaciais de algumas geodésicas tipo-tempo, correspondentes a partículas de teste inicialmente em repouso, sobre do disco D_0 . A linha pontilhada indica a posição do disco. No segundo gráfico estão as equivalentes Newtonianas, que são muito similares.

Discos de Poeira

Estes discos [46] são gerados pela aglutinação de dois espaço-tempos, associados com dipolos opostos. Estas métricas não são fisicamente aceitáveis, pois contém massas negativas. Podemos entretanto definir toda uma família de discos, com base no mesmo processo, apenas empregando outros harmônicos com simetria axial.

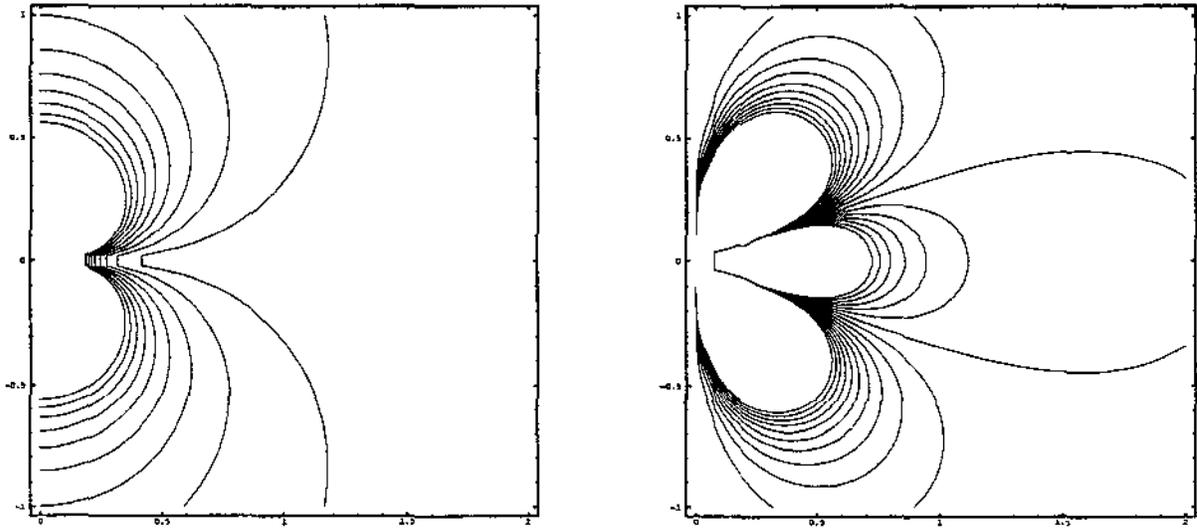


Figura 4.11: Funções ϕ e ν para o discos de noeira.

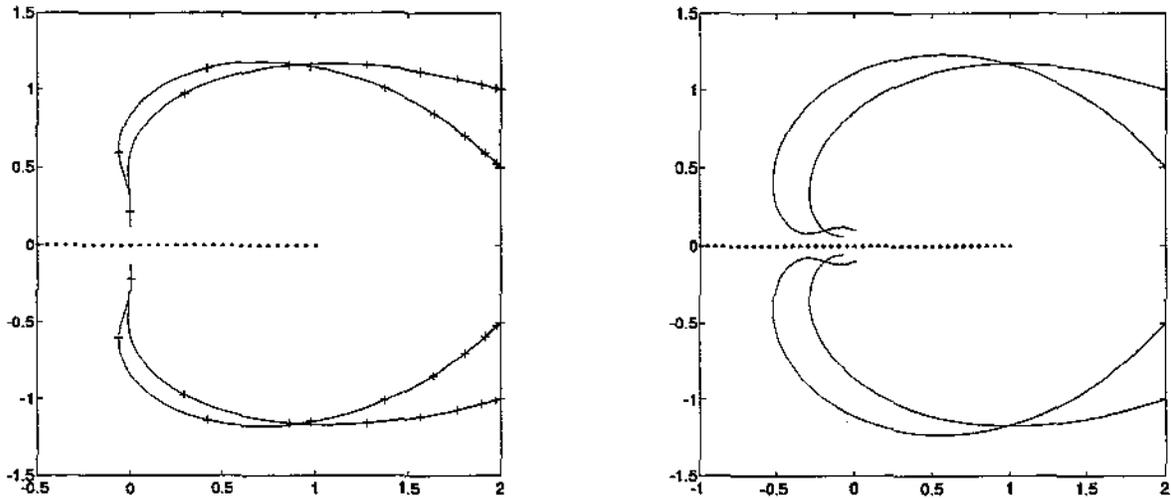


Figura 4.12: Geodésicas e queda Newtoniana no Disco de Poeira.

Considerando o primeiro harmônico esférico, com simetria axial $\mathcal{P}_1(\cos\theta) = \cos\theta$ teremos para $z > 0$

$$\phi^+ = -\frac{\alpha^2 z}{2(z^2 + r^2)^{3/2}}, \quad (4.32)$$

onde α é uma constante. Enquanto que para $z < 0$ definimos ϕ^- substituindo z por $-z$, assim calculando ν temos

$$\nu = \frac{\alpha^2 r^4}{16(z^2 + r^2)^4} \left(1 - 8\frac{z^2}{r^2}\right). \quad (4.33)$$

Os gráficos desta funções estão na Figura 4.11. Os componentes do tensor de energia momento são

$$\epsilon = 2\frac{\alpha^2}{r^3} e^{-\alpha^4/8r^4} \delta(z), \quad p_{\varphi\varphi} = 0.$$

No primeiro gráfico da Figura 4.12 estão as projeções espaciais das geodésicas tipo-tempo para partículas, inicialmente em repouso, sob o efeito do campo gravitacional desse disco. A linha pontilhada indica a posição do anel. No segundo gráfico apresenta as trajetórias Newtonianas.

Discos Anulares

Esta família [46] foi criada através de uma inversão nos discos de Morgan-Morgan. Considere $\Lambda(r, z)$ definida como a função ϕ dos discos de Morgan-Morgan (4.18), cuja densidade Newtoniana associada é \bar{S} , como em (4.19). Definindo agora a inversão

$$r^2 + z^2 \rightarrow \frac{a^4}{r^2 + z^2}, \quad (4.34)$$

sendo que a é o raio da borda interior do disco anular, temos, deste modo, a função ϕ dos novos discos

$$\phi = \frac{a}{R} \Lambda \left(\frac{a^2 r}{r^2 + z^2}, \frac{a^2 z}{r^2 + z^2} \right).$$

R é a coordenada radial, $R^2 = r^2 + z^2$. E a nova densidade é $S = \frac{a^3}{r^3} \bar{S} \left(\frac{a^2}{r} \right)$.

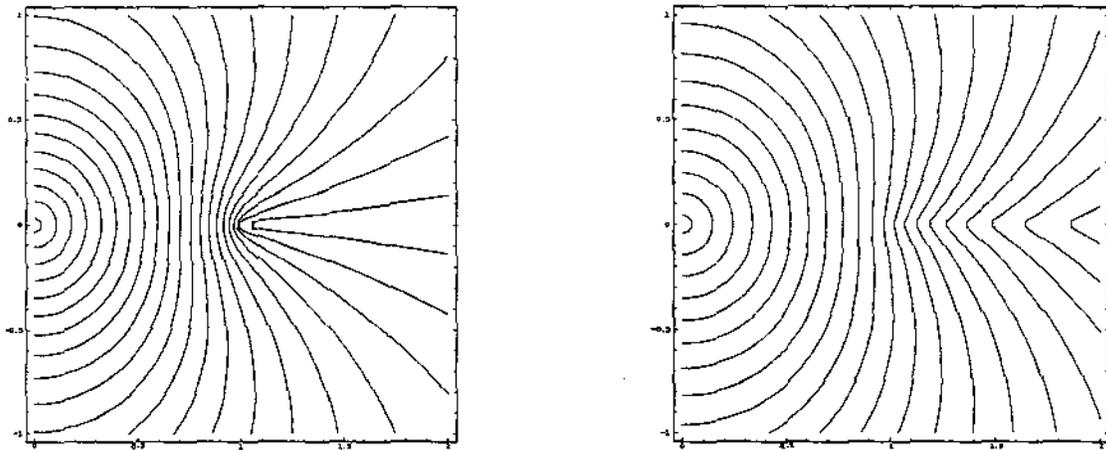


Figura 4.13: A função ϕ dos 2 primeiros discos anulares.

Tomando Λ como o primeiro membro da família dos discos de Morgan-Morgan (4.26) teremos o primeiro membro da nova família

$$\phi = \frac{2Ma^2}{\pi R^5} \operatorname{Im} \left[\left(z^2 - \frac{r^2}{2} + \frac{R^4}{a^2} \right) \ln \mu + \frac{s}{2} \left(3z + i \frac{R^2}{a} \right) \right]. \quad (4.35)$$

M é a massa do disco e $s = \sqrt{\left(-z + i \frac{R^2}{a} \right)^2 + r^2}$, $\mu = -\frac{ziR^2}{a} + s$.

Quando, $r \rightarrow \infty$, $\phi \rightarrow -2M/R$, comportamento esperado para um potencial Newtoniano. A função ν satisfará a condição de regularidade, $\nu \rightarrow 0$ quando $r \rightarrow 0$, bastando para isso uma escolha adequada das constantes de integração.

Discos de Letelier-Oliveira

A mais simples solução da equação de Laplace, em coordenadas toroidais (B.32), é

$$\phi = -\frac{M}{\sqrt{2}} \sqrt{\cos \xi - \cos \eta} \cos \frac{\eta}{2}. \quad (4.36)$$

Em coordenadas cilíndricas ele torna-se

$$\phi^{R_0} = -M \operatorname{Re} \frac{1}{R} \quad (4.37)$$

com $R^2 = (ia - z)^2 + r^2$. Comparando este resultado com com o primeiro disco de Morgan-Morgan (4.25) podemos observar as relações [46]

$$\phi^{R_0} = \frac{\partial}{\partial a} (a\phi^{D_0}), \quad \phi^{D_0} = \frac{1}{a} \int \phi^{R_0} da, \quad (4.38)$$

repetindo o processo para os demais discos de Morgan-Morgan podemos gerar os potenciais para essa nova família

$$\phi^{R_{n+1}} = \frac{\partial}{\partial a} (a\phi^{R_n}). \quad (4.39)$$

Os próximos membros, ϕ^{R_1} e ϕ^{R_2} , são

$$\phi^{R_1} = -M \operatorname{Re} \left(\frac{1}{R} + \frac{a(a + iz)}{R^3} \right), \quad (4.40)$$

$$\phi^{R_2} = -M \operatorname{Re} \left(\frac{1}{R} + \frac{a(4a + 3iz)}{R^3} + \frac{3a^2(a + iz)^2}{R^5} \right). \quad (4.41)$$

Enquanto as funções ν associadas são

$$\nu[\phi^{R_0}] = -M^2 r^2 \left[\frac{|\tau|^2}{(r^2 + |\tau|^2)^2 R^2} + \operatorname{Re} \left(\frac{1}{4R^4} \right) \right], \quad (4.42)$$

$$\nu[\phi^{R_1}] = \nu[\phi^{R_0}] - M^2 a [a(A + \operatorname{Re} B) - (\operatorname{Im} D + \operatorname{Im} E)], \quad (4.43)$$

onde $\mu = ia - z + R$ e

$$\begin{aligned} A &= 2\nu \left[\frac{ia - z}{R^3}, \frac{ia + z}{R^3} \right] = \frac{r^2 |\tau|^2 |R - ia + z|^2}{|R^6| (r^2 + |\tau|^2)^2} \\ &\quad - \frac{6r^4 |\tau|^4}{|R|^4 (r^2 + |\tau|^2)^4} + \frac{2r^2 |\tau|^4}{|R|^4 (r^2 + |\tau|^2)^3} \operatorname{Re} \left(\frac{ia - z}{R} \right), \\ B &= 2\nu \left[\frac{ia - z}{R^3} \right] = -\frac{r^2 (R - ia + z)^2}{4R^8} + \frac{3r^4}{8R^8} - \frac{r^2 \tau (ia - z)}{4R^8}, \\ D &= 2\nu \left[\frac{1}{R}, \frac{ia - z}{R^3} \right] = -\frac{r^2 (R - ia + z)}{4R^6} + \frac{r^2 \tau}{4R^6}, \\ E &= 2\nu \left[\frac{1}{R}, \frac{ia + z}{R^3} \right] = -\frac{r^2 |\tau|^2 (R - ia + z)}{R^2 |R|^2 (r^2 + |\tau|^2)^2} + \frac{2r^2 |\tau|^4}{R |R|^2 (r^2 + |\tau|^2)^3}. \end{aligned}$$

4.9 Solução de Appell

Em 1887 Appell [2] propôs uma solução para o Laplaciano, que consistia em tomar o potencial para uma massa pontual com massa e posição complexas, obtendo

$$\phi = -\frac{\mathcal{M}}{\mathcal{R}}, \quad (4.44)$$

onde $\mathcal{R}^2 = r^2 + (z - \mathcal{Z})^2$ enquanto \mathcal{M} e \mathcal{Z} são números complexos. Agora se tomarmos a parte real desta expressão ela também será solução do Laplaciano e além disso decairá com $(r^2 + z^2)^{-1/2}$ quando $r^2 + z^2 \rightarrow \infty$, indicando que esta solução está associada a uma distribuição de matéria isolada. Desta forma tomaremos como solução de Appell a parte real de (4.44).

Como este potencial é singular em uma região em forma de anel ele foi chamado, por muitos autores, de anel de Appell. Mas ele não é exatamente o potencial gerado por anel, como veremos adiante. Se impusermos que \mathcal{Z} seja imaginário puro, isso é $\mathcal{Z} = ia$, o potencial será singular em um anel de raio a sobre o plano $z = 0$. Então escrevendo $\mathcal{M} = me^{i\alpha}$ teremos

$$\phi = -m \operatorname{Re} \left(\frac{e^{i\alpha}}{\mathcal{R}} \right), \quad (4.45)$$

onde agora $\mathcal{R}^2 = r^2 + (z - ia)^2$. Sem perda de generalidade podemos impor que $|\alpha| \leq \frac{\pi}{2}$.

Escrevendo os primeiros termos da expansão em série desta solução temos que quando $r^2 + z^2 \gg a$,

$$\phi \sim -\frac{m \cos \alpha}{r^2 + z^2} + \frac{m \operatorname{sen} \alpha}{(r^2 + z^2)^3}. \quad (4.46)$$

Portanto a massa total da distribuição de massa é $m \cos \alpha$ e o termo dipolar é proporcional a $m \operatorname{sen} \alpha$.

Podemos expressar o potencial ϕ em coordenadas toroidais como

$$\phi = -\frac{m}{\sqrt{2a}} \sqrt{\cosh \eta - \cos \xi} \cos \left(\frac{\xi}{2} + \alpha \right)$$

enquanto a função ν correspondente [48] é

$$\nu = \frac{m^2}{a^2} \left[2(1 - \cos \xi) - \operatorname{senh}^2 \eta \cos 2(\xi + \alpha) \right]. \quad (4.47)$$

Porém, embora a função ϕ e suas derivadas sejam localmente bem definidas, em termos das coordenadas cilíndricas r e z , ϕ e ϕ_ξ são multivaloradas quando consideradas globalmente, isso é, se partirmos de um ponto e efetuarmos uma curva fechada em volta da singularidade do anel não recuperaremos os valores de ϕ ou de seu gradiente [28].

Para eliminar este inconveniente devemos impor restrições sobre ξ . Mas se tomarmos $0 \leq \xi \leq 2\pi$ a função ϕ terá uma descontinuidade tipo salto em $\xi = 0$ o que implica que ϕ_ξ terá uma descontinuidade tipo delta. Não podemos aceitar esse tipo de descontinuidade pois o tensor de curvatura é definido com as derivadas segundas da métrica.

Por outro lado notando que $\phi = 0$ para $\xi = -2\alpha \pm \pi$ e restringindo ξ ao intervalo $-2\alpha - \pi \leq \xi \leq -2\alpha + \pi$ a função ϕ será univalorada e contínua. Enquanto ϕ_ξ terá uma descontinuidade tipo salto finita, e a distribuição tipo delta aparece apenas nas derivadas segundas de ϕ , o que nos permite definir o tensor de curvatura em um sentido distribucional.

Esta função ϕ é, na verdade, solução da equação de Poisson e não do Laplaciano, pois a descontinuidade na derivada ϕ_ξ corresponde a uma membrana de matéria na superfície $\xi = -2\alpha \pm \pi$ [28]. Estas superfícies são capas esféricas com bordas no anel $z = 0$ e $r = a$. Quando $\alpha = 0$ estas capas colapsam na superfície plana no interior do anel, este caso particular é tratado por disco de Appell e será melhor estudado na próxima sub-seção.

Se tomarmos $\alpha = \pm\pi/2$ a membrana ficará sobre a superfície $\xi = 0$, ou seja, a região do plano $z = 0$ exterior ao anel. Mas neste caso a massa total do anel será nula restando apenas o termo dipolar, veja a expressão (4.46).

Disco de Appell

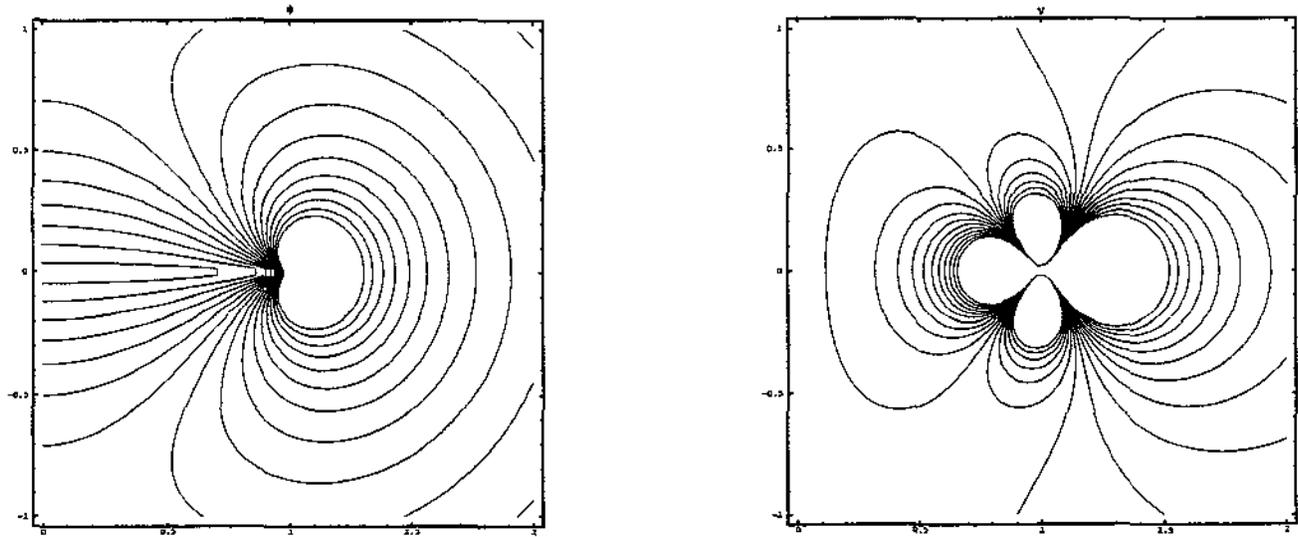


Figura 4.14: Potenciais do disco de Appell.

O disco de Appell é o caso particular da solução de Appell quando tomamos a constante $\alpha = 0$ e ele corresponde exatamente ao primeiro disco de Letelier e Oliveira (4.39). Os gráficos das funções ϕ e ν para esta solução estão na Figura 4.14.

Como a função ν é suave todas as contribuições ao tensor de energia-momento são oriundas da função ϕ , portanto se calcularmos a única componente não nula deste tensor teremos

$$T_t^t = -e^{\phi-\nu} \Big|_{z=0} [\phi_z] \delta(z),$$

onde $[\phi_z]$ é a magnitude do salto da função ϕ_z em $z = 0$. Podemos agora calcular a densidade superficial no interior do anel

$$\sigma = e^{\frac{\phi-\nu}{2}} \Big|_{z=0} [\phi_z]$$

No limite Newtoniano esse resultado coincide com o obtido quando consideramos ϕ como solução da equação de Poisson

$$\sigma = -\frac{16\pi m a}{\sqrt{a^2 + r^2}}$$

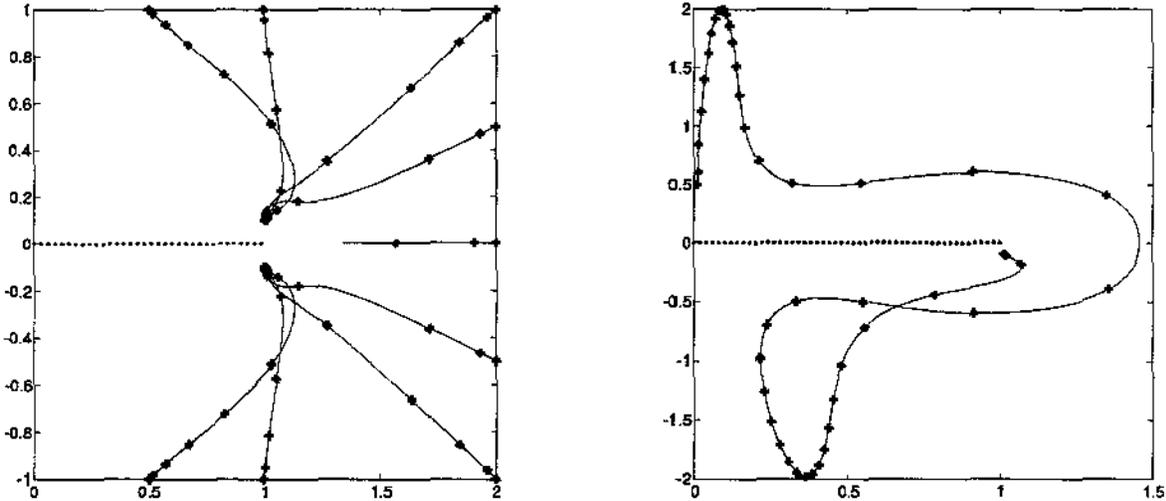


Figura 4.15: Geodésicas do disco de Appell.

Note que se desejarmos que a massa total do disco seja positiva devemos tomar m maior que zero o que implica que σ será negativa. Além disso $\sigma \rightarrow -\infty$ quando $r \rightarrow a$ pelo interior do disco o que torna a massa da superfície no interior do anel infinitamente negativa, como a massa total do disco é positiva a massa sobre o anel deve ser infinitamente positiva.

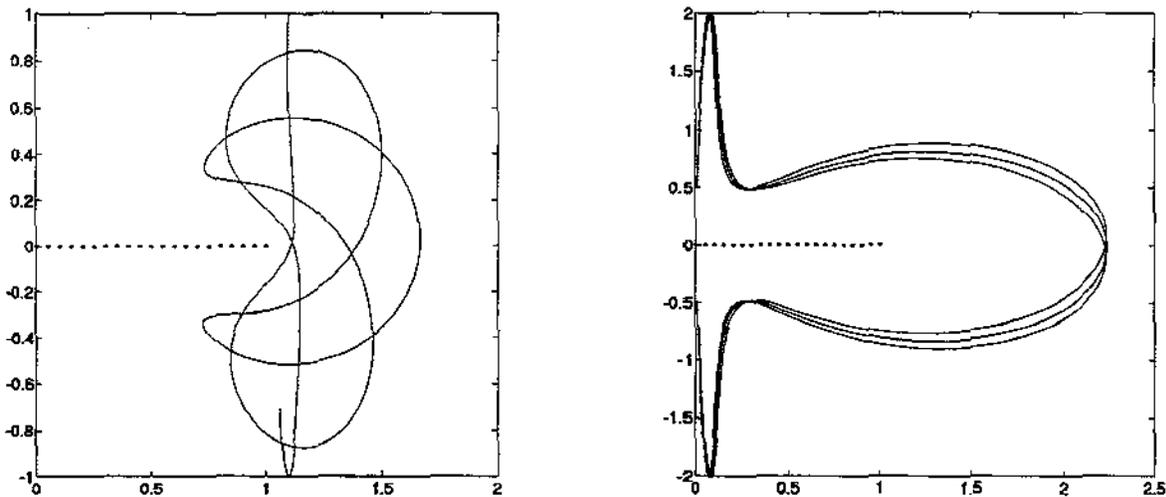


Figura 4.16: Queda Newtoniana no disco de Appell.

As propriedades desta métrica podem ser melhor compreendidas quando observamos o comportamento de partículas em queda livre. Na Figura 4.15 colocamos as projeções espaciais das trajetórias destas partículas, os pontos marcados ao longo de cada linha indicam a passagem

de uma unidade do tempo próprio. O segundo gráfico apresenta a trajetória de uma partícula solta no ponto (0.01, 0.5). Devido a presença de matéria negativa no interior do disco seu comportamento é exótico. A linha pontilhada marca a posição do disco.

Na Figura 4.16 estão duas trajetórias para partículas em queda livre Newtoniana. No primeiro gráfico a partícula foi solta do ponto (1, 1) e no segundo do ponto (0.01, 0.5). Ao contrário dos discos estudados anteriormente, estas trajetórias são totalmente diferentes dos seus equivalentes relativísticos.

Como o disco é composto tanto de matéria positiva como negativa o ponto $r = 0$ e $z = a$ encontra-se em equilíbrio entre as forças atrativas e repulsivas. Retomaremos esse tópico na Seção 5.5, quando sobreponemos esse disco com a solução de Curzon.

4.10 Anéis

Anel de Letelier-Oliveira

Este anel é o resultado da sobreposição de um disco de Morgan-Morgan com um disco de Appell [51]. Pois o primeiro anula a massa negativa contida no interior do último. Como os discos são colocados sobre o mesmo plano a solução resultante estará em equilíbrio, não apresentando, portanto, singularidades estruturais ou membranas.

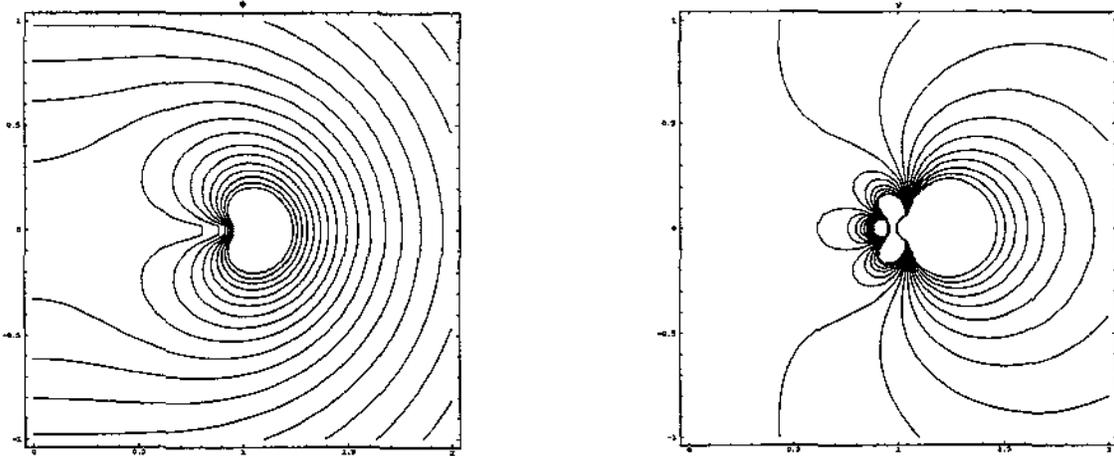


Figura 4.17: Potenciais do anel de Letelier-Oliveira.

Tomando os potenciais de Appell e Morgan-Morgan definidos como

$$\phi_{Appell} = -\operatorname{Re} \left(\frac{M}{2R_a} \right), \quad \phi_{Morgan} = -\operatorname{Im} \left(\frac{M}{2a} \ln \mu_a \right),$$

onde $R_a^2 = r^2 + (z - ia)^2$ e $\mu_a = ia - z + R_a$. E fixando uma escolha para as funções multivaloradas envolvidas, temos

$$\phi_{Appell} = -\frac{m}{2\sqrt{2}} \sqrt{\frac{A}{B}}, \quad \phi_{Morgan} = -\frac{m}{2a} \arctan \sqrt{\frac{2a^2}{A}},$$

onde definimos

$$A = \sqrt{B} + K, \quad B = K^2 + 4z^2a^2, \quad K = r^2 + z^2 - a^2.$$

Portanto, a solução para o anel é

$$\phi = \phi_{Appell} + \phi_{Morgan}, \quad (4.48)$$

$$\nu[\phi] = -\frac{M^2}{4a^2} \left[1 + \ln \left(\frac{r^2 + |\mu_a|^2}{|r^2 + \mu_a^2|} \right) \right] - \frac{M^2}{32a^2} \left(\frac{r^2 + |z - ia|^2}{|R_a|^2} - 1 \right) \quad (4.49)$$

$$-\frac{M^2}{16} \operatorname{Re} \left(\frac{r^2}{R_a^4} \right) + \frac{M^2}{2a} \operatorname{Im} \left(\frac{r^2}{R_a(r^2 + \mu_a^2)} - \frac{\mu_a}{R_a(\mu_a - \mu_a^*)} \right).$$

Os gráficos destas funções estão na Figura 4.17.

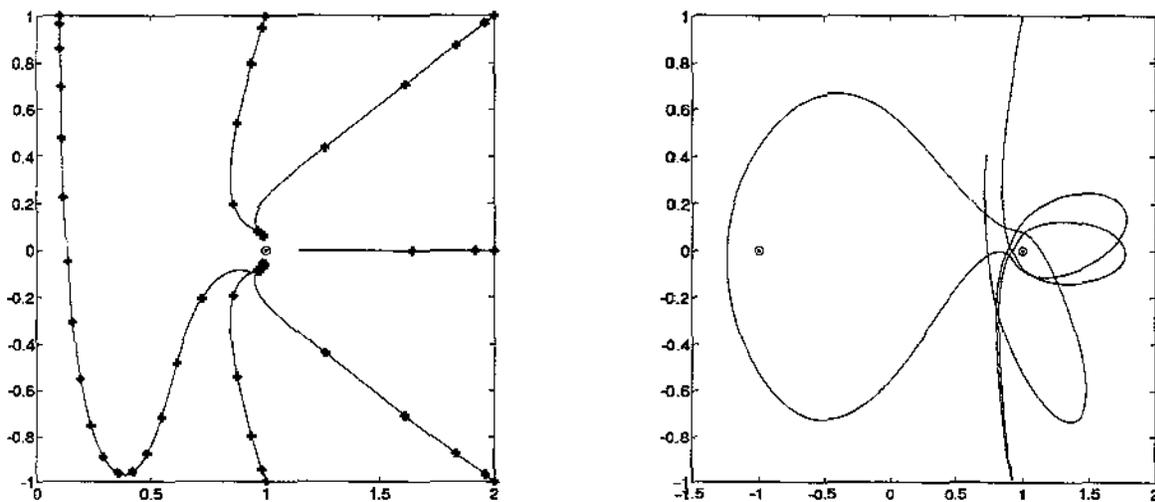


Figura 4.18: Geodésicas e queda Newtoniana no anel de Letelier-Oliveira.

No primeiro gráfico da Figura 4.18 apresentamos as projeções espaciais das geodésicas tipo-tempo desta métrica. Neste gráfico estão as trajetórias das partículas que partiram do repouso, as marcas indicam a passagem de uma unidade do tempo próprio e o símbolo \odot indica a posição do anel.

A trajetória que parte do ponto $(2, 0)$ termina bruscamente após a passagem de 2,7 unidades de tempo, indicando que o método numérico encontrou algum tipo de singularidade. Porém, as outras geodésicas podem ser prolongadas indefinidamente, indicando que estas partículas só atingirão a singularidade após a passagem de um intervalo infinito do seu tempo próprio.

No sistema de coordenadas cilíndrico, esta solução apresenta uma singularidade de coordenadas no plano interno ao anel, que foi sentida pelos métodos numéricos. Assim, para podermos acompanhar a geodésica que parte do ponto $(0.01, 1)$ foi necessário calcular manualmente a iteração que atravessa este plano.

No segundo gráfico da Figura 4.18 é apresentada a trajetória de uma partícula em queda livre a partir do repouso no ponto $(1, 1)$, segundo as equações de movimento Newtonianas.

Anel de Weyl-Bach

O potencial Newtoniano para um anel deve ser calculado pela integração das contribuições de cada partícula que o compõe, como vemos na expressão

$$\phi = \int_{\text{anel}} \frac{\sigma}{R_a} d\xi.$$

ξ é a variável de integração, σ a densidade linear do anel e R_a é a distância entre o ponto de integração e o ponto onde desejamos calcular o potencial.

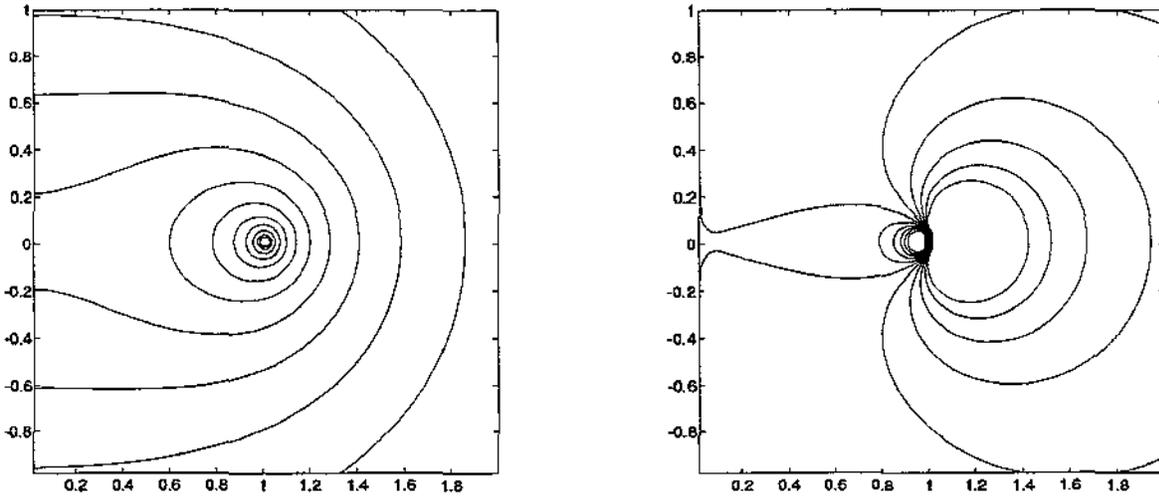


Figura 4.19: Potenciais do anel de Weyl-Bach.

Em coordenadas cilíndricas esta integral torna-se

$$\phi = -\frac{4a\sigma}{R_a} K\left(\frac{2\sqrt{ar}}{R_a}\right), \quad (4.50)$$

onde $K(\kappa)$ é a integral elíptica do primeiro tipo e fixando o anel no plano $z = z_0$ com raio a a distância R_a pode ser expressa tomando a raiz positiva de $R_a^2 = (a+r)^2 + (z-z_0)^2$.

Em coordenadas toroidais (B.32) este potencial escreve-se como

$$\phi = \sigma \sqrt{\cosh \eta - \cos \xi} H(\eta), \quad (4.51)$$

$$\text{onde } H(\eta) = e^{-\eta/2} K(\kappa), \quad \kappa^2 = 1 - e^{2\eta}. \quad (4.52)$$

A função ν associada é dada pela expressão [70] [36]

$$\nu = -\ln(\cosh \eta - \cos \xi) \quad (4.53)$$

$$-\sigma^2 \sinh \eta \left[\sinh \eta \cos \xi \left(\frac{H^2}{4} + H'^2 \right) - (1 - \cosh \eta \cos \xi) H H'^2 \right].$$

Os gráficos destas funções estão na Figura 4.19.

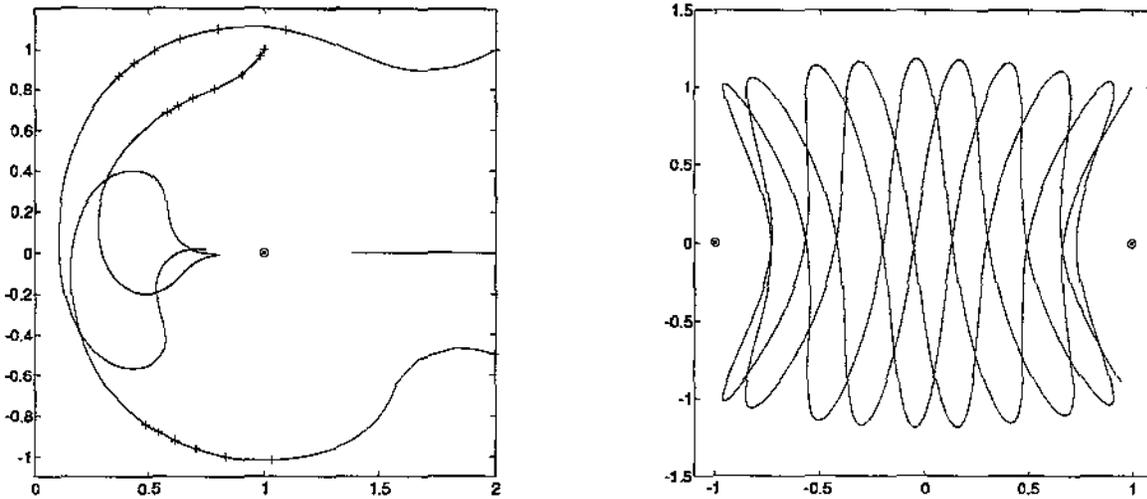


Figura 4.20: Geodésicas e queda Newtoniana no anel de Weyl-Bach

Embora tenha sido gerado a partir do potencial Newtoniano de um anel, esta métrica não corresponde a uma distribuição de massa realista, pois apresenta uma singularidade direcional sobre o anel, como mostra o primeiro gráfico da Figura 4.20. Neste gráfico incluímos as projeções espaciais das trajetórias de partículas em queda livre, o símbolo \odot indica a posição do anel. Note que a trajetória que inicia no ponto $(2, 0)$ cai retilineamente em direção a singularidade pelo seu lado externo, porém após um intervalo de 0,2 unidades de seu tempo próprio o método numérico é interrompido bruscamente, possivelmente indicando a presença de uma singularidade de curvatura.

Todas as outras trajetórias, apresentam intervalos onde foram repelidas pelo anel, o que indica a presença de matéria negativa, além disso elas só se aproximam do anel pelo seu lado interno. As marcas sobre as trajetórias indicam a passagem de uma unidade do tempo próprio. Foram indicadas apenas as primeiras unidades, pois a partir destes pontos elas se aglutinam rapidamente, por exemplo as trajetórias iniciadas nos pontos $(1, 1)$, $(2, 1)$ e $(2, -0.5)$ foram simuladas até atingirem, respectivamente, 200, 50 e 100 unidades do tempo próprio.

4.11 Soluções Multipolares

O Caso Newtoniano

Na teoria gravitacional Newtoniana quaisquer corpos com simetria axial têm seu potencial externo descrito por

$$\phi = \sum_{n=0}^{\infty} \frac{D_n}{R^{n+1}} \mathcal{P}_n(\cos \theta), \quad (4.54)$$

onde (ρ, θ, φ) são coordenadas esféricas (B.8) e D_n são os momentos multipolares da fonte. Estes momentos são definidos como

$$D_n = \int_V \mu(\xi, \beta) \xi^n \mathcal{P}_n(\cos \beta) dv. \quad (4.55)$$

μ é a densidade de massa, V é o volume do corpo e (ξ, β, φ) são as coordenadas do ponto de integração. $D_0 = M$ é a massa total do corpo e os demais momentos determinam como esta massa está distribuída.

No caso relativístico a definição de momentos é muito mais complicada, veja por exemplo [27]. O que apresentamos a seguir são expansões em série das soluções de Weyl, embora seja comum chamar-las de multipolares elas não correspondem aos multipolos relativísticos nem aos Newtonianos, como veremos.

Momentos de Weyl

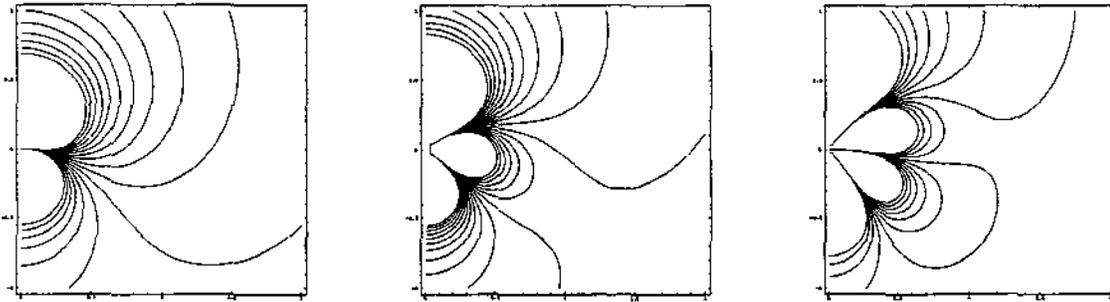


Figura 4.21: Funções ϕ_1 , ϕ_2 e ϕ_3 .

Podemos escrever todas as soluções assintoticamente planas das equações de Weyl empregando as séries [61]

$$\phi = - \sum_{n=0}^{\infty} \frac{a_n}{R^{n+1}} \mathcal{P}_n \left(\frac{z}{R} \right), \quad (4.56)$$

$$\nu = \sum_{n,k=0}^{\infty} \frac{(n+1)(k+1)}{n+k+2} \frac{a_n a_k}{R^{n+k+2}} (\mathcal{P}_{n+1} \mathcal{P}_{k+1} - \mathcal{P}_n \mathcal{P}_k), \quad (4.57)$$

onde $R^2 = r^2 + z^2$. As constantes arbitrárias a_n fazem o mesmo papel dos momentos gravitacionais Newtonianos D_n , sendo portanto chamadas de momentos de Weyl.

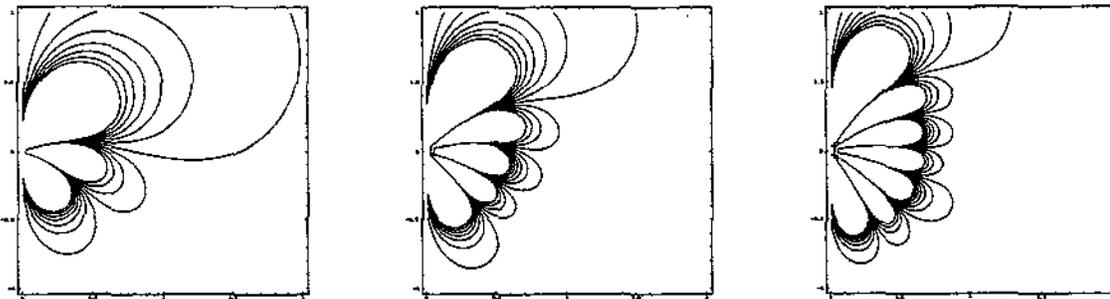


Figura 4.22: Funções ν_1 , ν_2 e ν_3 .

Porém não podemos identificar estes momentos aos verdadeiros. Para esclarecer este ponto apresentamos a expansão, em momentos de Weyl, da métrica de Schwarzschild

$$a_{2n} = \frac{M^{2n+1}}{2n+1}, \quad a_{2n+1} = 0. \quad (4.58)$$

Observe que são necessários infinitos termos para obter esta solução, porém ela é, por definição, a solução monopolar da relatividade geral. Por outro lado, se tomarmos apenas o primeiro termo da série (4.56) teremos a solução de Curzon (4.2), que nitidamente possui uma composição mais complexa que a monopolar.

Nas Figuras 4.21 e 4.22 apresentamos os gráficos das funções ϕ e ν para primeiras somas parciais de (4.56)–(4.57). Excluímos a primeira ϕ_0 e ν_0 pois ela é exatamente a solução de Curzon e seu gráfico está na Figura 4.1.

Momentos de Erez-Rosen

Como a solução de Schwarzschild é a solução monopolar, na gravitação relativística, Erez e Rosen [25], propuseram uma outra expansão em série para as soluções de Weyl assintoticamente planas, de modo que o primeiro termo desta série fosse o monopolo relativístico. Para isso utilizaram o sistema de coordenadas esferoidais prolatas (B.21), fixando o parâmetro a como a massa do corpo M . Neste sistema de coordenadas, a solução de Schwarzschild (4.4) é escrita como

$$\phi = \frac{1}{2} \ln \frac{x-1}{x+1}, \quad \nu = \frac{1}{2} \ln \frac{x^2-1}{x^2+y^2}.$$

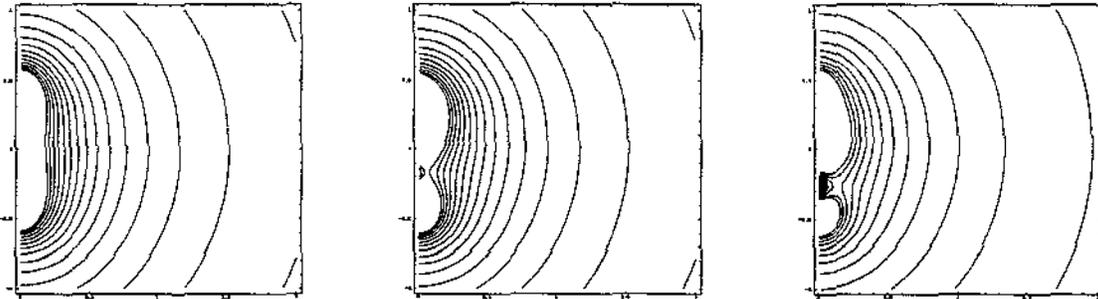


Figura 4.23: Função ϕ para o quadrupolo, octopolo e 16-polo.

Esta expansão é descrita pelas fórmulas [61]

$$\phi = \sum_{n=0}^{\infty} (-1)^{n+1} q_n Q_n(x) P_n(y), \quad (4.59)$$

$$\nu = \sum_{m,n=0}^{\infty} (-1)^{m+n} q_m q_n \Gamma^{(mn)}. \quad (4.60)$$

$\Gamma^{(mn)}$ é apresentada no Apêndice C, bem como as primeiras soluções e algumas propriedades das soluções de Weyl quando escritas no sistema de coordenadas esferoidais prolatas. Enquanto q_n são constantes arbitrárias, que chamaremos de momentos de Erez-Rosen. Este momentos estão relacionados com os momentos de Weyl pela fórmula [60]

$$a_n = \sum_{j=0}^T \frac{(-M)^{n+1} n! q_k}{(n+k+1)!! (n-k)!!}, \quad (4.61)$$

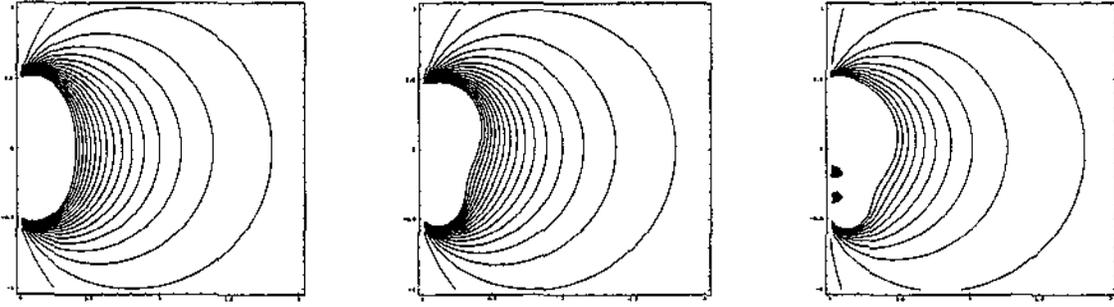


Figura 4.24: Função ν para o quadrupolo, octopolo e 16-polo

onde $k = 2j$ e $T = \frac{n}{2}$ quando n for par e $k = 2j + 1$ e $T = \frac{n-1}{2}$ quando for ímpar.

Além do primeiro termo da série ser a solução de Schwarzschild a solução composta dos dois primeiros termos é a solução de Erez-Rosen, que é definida escolhendo $q_0 = 1$, $q_2 \neq 0$ e $q_j = 0$ para todo $j \neq 0, 2$. O termo dipolar q_1 é descartado, pois sempre é possível realizar uma transformação de coordenadas que o anule. As Figuras 4.23 e 4.24 contêm os gráficos das funções ϕ e ν das primeiras soluções multipolares, exceto a primeira que como é a solução de Schwarzschild pode ser vista na Figura 4.3.

Momentos de Gutsunaev-Manko

Outra representação possível, também em coordenadas esferoidais prolatas [61] é

$$\phi = \frac{1}{2} \ln \frac{x-1}{x+1} + \sum_{n=0}^{\infty} b_{n+1} \left(\frac{P_n^+}{(x+y)^{n+1}} - \frac{P_n^-}{(x-y)^{n+1}} \right), \quad (4.62)$$

onde definimos

$$P_n^{\pm} = \mathcal{P}_n \left(\frac{xy \pm 1}{x \pm y} \right)$$

e as constantes b_n são os momentos de Gutsunaev-Manko, que se relacionam com os momentos de Weyl pelas fórmulas

$$a_{2n} = -\frac{M^{2n+1}}{2n+1} - 2M^{2n+1} \sum_{k=1}^n \binom{2n}{2k-1} b_{2k},$$

$$a_{2n+1} = -2M^{2n+2} \sum_{k=1}^n \binom{2n+1}{2k} b_{2k+1}.$$

4.12 Soluções em Sistemas R-Separáveis

Como nos casos apresentados na seção anterior, estas soluções baseiam-se nas autofunções do Laplaciano em sistemas de coordenadas adequados. O termo R-separável é empregado, pois nas coordenadas que serão tratadas aqui a separação de variáveis só é possível quando incluímos uma função R na expressão para ϕ , ou seja, $\phi(u, v) = R(u, v)U(u)V(v)$. Esta função é dependente do sistema de coordenadas. Incluímos abaixo duas famílias de soluções que foram calculadas por Carminati e Sarracino [13] e que empregam esse tipo de soluções do Laplaciano.

Não incluímos estas soluções como multipolos, pois nas coordenadas de esferas tangentes, possuímos apenas as soluções correspondentes a cada autofunção isolada e não para suas combinações. E nas coordenadas bi-esféricas a solução associada a sobreposição das autofunções corresponde a solução geral para dois corpos, sendo portanto apresentada no próximo capítulo.

Soluções em Coordenadas de Esferas Tangentes

Neste sistema de coordenadas (B.40), a solução geral para o Laplaciano é dada por

$$\phi = \sqrt{\chi^2 + \psi^2} X(\chi) P(\psi)$$

e as funções X e P são (B.44)–(B.45)

$$\begin{aligned} q = 0 \quad X &= A + B \log \chi, \\ P &= C + D\psi, \\ q \neq 0 \quad X &= A \mathcal{J}_0(q\chi) + B \mathcal{Y}_0(q\chi), \\ P &= C e^{q\psi} + D e^{-q\psi}, \end{aligned}$$

A, B, C e D são constantes arbitrárias e \mathcal{J}_0 e \mathcal{Y}_0 são as, primeira e segunda, funções de Bessel de ordem zero.

O potencial métrico ν para o caso $q = 0$ é dado por

$$\begin{aligned} \nu &= 2B\psi(B + X) \left[\frac{D^2\psi^3}{4} + \frac{2CD\psi^2}{3} + \frac{\psi}{2} \left(C^2 - \frac{D^2\chi^2 X}{B} \right) - \frac{CD\chi^2 X}{B} \right] \\ &\quad + 2D\psi^2 X \left(\frac{BD\psi^2}{4} + \frac{BC\psi}{3} - \frac{D\chi^2 X}{2} \right) + F(\chi) \end{aligned}$$

nesta expressão $F(\chi)$ é uma função de integração, que pode ser tomada como

$$\begin{aligned} F(\chi) &= \frac{\chi^2}{2} \left[\frac{D^2\chi^2}{2} \left(A^2 + \frac{B^2}{8} - \frac{AB}{2} \right) - C^2 \left(A^2 + \frac{B^2}{2} + AB \right) \right] \\ &\quad - \frac{BC^2\chi^2}{2} \ln \chi (2A + B + B \ln \chi) + \frac{BD^2\chi^4}{2} \ln \chi \left(A - \frac{B}{4} + \frac{B}{2} \ln \chi \right) + \mathcal{E}. \end{aligned}$$

Enquanto no caso $q \neq 0$ a expressão para ν é

$$\begin{aligned} \nu &= -\alpha_+ \chi \left[q\psi^2 X X_1 + \frac{\chi}{2} (X^2 + X_1^2 - 2q\chi X X_1) \right] - \frac{CD\chi^2}{3} (3X^2 - X_1^2) \\ &\quad + q\alpha_- \chi^2 \psi (X_1^2 - X^2) + 2q^2 CD\chi^2 \left(\psi^2 - \frac{\chi^2}{3} \right) \left(X^2 + X_1^2 - \frac{X X_1}{q\chi} \right) + \mathcal{E}, \end{aligned}$$

\mathcal{E} é uma constante arbitrária e definimos as funções

$$X_1 = A \mathcal{J}_1(q\chi) + B \mathcal{Y}_1(q\chi), \quad \alpha_{\pm} = C^2 e^{2q\psi} \pm D^2 e^{-2q\psi}.$$

Soluções em Coordenadas Bi-Esféricas

O elemento de linha para as soluções Weyl neste sistema de coordenada assume a forma

$$ds^2 = e^{2\phi} dt^2 - \frac{\alpha^2 e^{2(\nu-\phi)}}{(\cosh \eta - \cos \theta)^2} (d\eta^2 + d\theta^2) - \frac{\alpha^2 \sin^2 \theta e^{-2\phi}}{(\cosh \eta - \cos \theta)^2} d\varphi^2.$$

A solução geral para o Laplaciano é (B.47)

$$\phi(\eta, \theta) = \sqrt{\cosh \eta - \cos \theta} H(\eta) \Theta(\theta),$$

onde $H = Ae^{\eta(s+\frac{1}{2})} Be^{-\eta(s+\frac{1}{2})}$,

$$\Theta = C \mathcal{P}_s(\cos \theta) + D \mathcal{Q}_s(\cos \theta).$$

Se $s \neq 1/2, 0$ a função ν é dada por

$$\begin{aligned} \nu = \sin \theta \left[\Lambda \sin \theta \left(\Theta'^2 + s(s+1)\Theta^2 \right) - HH'\Theta^2 \sinh \eta \sin \theta \right. \\ \left. + H^2 \Theta \Theta' (\cos \theta \cosh \eta - 1) \right] + F, \end{aligned}$$

sendo que Λ é definida pela integral

$$\Lambda = \int H^2 \sinh \eta d\eta$$

e a função de integração F pode ser escrita como

$$F(\theta) = -4AB \left(s + \frac{1}{2} \right)^2 \int_{\xi}^1 \Theta^2(\xi) d\xi + \mathcal{E},$$

onde $\xi = \cos \theta$. Quando $s = 0$, ν pode ser simplificado em

$$\nu = \Lambda D^2 + \Theta^2 \sin^2 \theta \cosh \eta \left(AB - \frac{H^2}{2} \right) - DH^2 \Theta (\cos \theta \cosh \eta - 1) + F(\theta)$$

agora F é escrita como

$$\begin{aligned} F(\theta) = -2AB \int_{\xi}^1 \Theta^2(\xi) d\xi + \frac{A^2 + B^2}{2} [2D \cos \theta (C + D \mathcal{Q}_0(\cos \theta)) \\ - C^2 \cos^2 \theta + 2D^2 \ln |\sin \theta| + D \sin^2 \theta \mathcal{Q}_0(\cos \theta)] + \mathcal{E}. \end{aligned}$$

Chegamos agora ao último caso $s = 1/2$, onde H torna-se uma constante, que fixaremos como $H = 1$ obtendo então

$$\nu = \sin \theta \cosh \eta \left(\Theta'^2 \sin \theta + \Theta \Theta' \cos \theta - \frac{1}{4} \Theta^2 \sin \theta \right) - \Theta \Theta' \sin \theta + \mathcal{E}.$$

A função de Legendre do segundo tipo $\mathcal{Q}_s(x)$ é singular quando $x = 1$, deste modo a autofunção $\Theta(\theta)$ será sempre singular quando $\theta = 0$, ou seja, ao longo do eixo z . Conseqüentemente as soluções em que $D \neq 0$ terão uma linha infinita como imagem Newtoniana, por outro lado se $D = 0$ a solução será assintoticamente plana com massa total $m = -a(A+B)/\sqrt{2}$ e representando a solução para duas partículas, situadas em $z = \pm a$, com massas $m_1 = -aA/\sqrt{2}$ e $m_2 = -aB/\sqrt{2}$. A sobreposição de autofunções com $D = 0$ representa a solução geral para duas partículas, este caso será tratado na Seção 5.3, do próximo capítulo.

4.13 Soluções de Godfrey

Estas são soluções de Weyl que contêm outras simetrias além das exigidas pela definição [29]. A primeira classe de soluções, que é caracterizada por dois parâmetros α e C , é definida como

$$\phi = \alpha \ln r + z, \quad \nu = \alpha^2 \ln r + 2\alpha z - \frac{r^2}{2} + C. \tag{4.63}$$

E a segunda, que depende de três parâmetros α , β e C é descrita por

$$\phi = \beta \ln r + \frac{\alpha - \beta}{2 \ln(\sqrt{r^2 + z^2} + z)},$$

$$\nu = \beta^2 \ln r + \frac{\alpha^2 - \beta^2}{2 \ln(\sqrt{r^2 + z^2} + z)} - \frac{(\alpha - \beta)^2}{2 \ln(r^2 + z^2) + C}.$$

Note que escolhendo $\alpha = \beta \neq -1, 0, 1/2, 1$ ou 2 então essa métrica torna-se a solução de Levi-Civita (4.5)–(4.6).

O elemento de linha para para a métrica (4.63) com $\alpha = 1$ é escrito como

$$ds^2 = r^2 e^{2z} dt^2 - e^{2z-r^2+2C} (dr^2 + dz^2) - e^{-2z} d\phi^2,$$

que não é assintoticamente plano. Por outor lado a norma de vetor de Killing tipo-tempo, $|\zeta| = r^2 e^{2z}$ se anula para todo o eixo z , enquanto a norma do vetor de Killing tipo-espaco, $|\xi| = -e^{-2z}$, é finita para qualquer valor de z portanto todo o eixo z é um horizonte extensível.

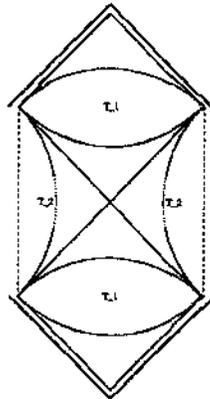


Figura 4.25: Diagrama de Carter-Penrose da métrica de Godfrey.

Para obtermos a extensão maximal desta métrica aplicamos a transformação

$$u = \exp \left[\text{Ei} \left(\frac{r^2}{2} \right) \right] \cosh \left(\frac{t}{2e^C} \right), \quad v = \exp \left[\text{Ei} \left(\frac{r^2}{2} \right) \right] \sinh \left(\frac{t}{2e^C} \right).$$

O elemento de linha assim obtido é

$$ds^2 = 4r^2 \exp \left(2z + 2C - \text{Ei} \left(\frac{r^2}{2} \right) \right) (dv^2 - du^2) - e^{2z-r^2+2C} dz^2 - e^{-2z} d\phi^2$$

e podemos ver que ele é regular para $r = 0$. O que nos permite estender seu domínio de definição para todo o plano $u - v$. Na Figura 4.25 exibimos o diagrama de Carter-Penrose desta métrica, onde o domínio original é representado pela região I enquanto a região II é uma extensão. Note que o infinito espacial é constituído por duas linhas tipo-tempo isso é uma consequência da não assintoticidade da métrica e indica a presença de singularidades no infinito.

4.14 Métricas de Kinnersley

Kinnersley [38] apresentou diversas métricas, sendo que as contidas no caso IV-B são soluções para o vácuo e entre elas K1 e K2 são axialmente simétricas e estáticas, desde que $C = 0$ ou $C = -1/2$, estas são, portanto, métricas de Weyl.

A solução K1 é dada por

$$ds^2 = \frac{k^2 p^2}{x^2} dw^2 + 2dw dp - \frac{4p}{x} dw dx - \frac{\Delta}{2} dx^2 - 2\Delta d\varphi^2,$$

onde $\Delta = mx^{-1} - k^2/2$, ou depois de transformada

$$ds^2 = k^2(R+z)^2 dt^2 - \frac{k^2(R+z)^2}{(2R)^4} (dr^2 + dz^2) - \frac{k^2 r^2}{(R+z)^2} d\varphi^2.$$

Assim comparando com (2.6) podemos extrair os potenciais de Weyl

$$\phi = \ln(R+z) + \ln k, \quad \nu = 2 \ln \frac{R+z}{2R} + 2 \ln k.$$

A outra solução, K2, é definida como

$$ds^2 = 2dw dp - \frac{4p}{x} dw dx - \frac{x}{2m} dx^2 - \frac{2m}{x} d\xi$$

podendo ser transformada em

$$ds^2 = r dt^2 - r^{-1/2} (dr^2 + dz^2) - r d\varphi^2,$$

onde os potenciais de Weyl tornaram-se

$$\phi = \frac{1}{2} \ln r, \quad \nu = \frac{1}{4} \ln r,$$

que são exatamente as funções que definem a métrica para uma linha infinita (4.7) com $C = 1$ e $\sigma = 1/4$.

Soluções Sobrepostas

Neste capítulo listamos várias soluções de Weyl cujos potenciais ϕ são somas dos apresentados no capítulo anterior. Como a integral (2.8) não é linear, a função ν para uma sobreposição conterá um termo de interação, a principal característica este termo é a introdução de singularidades estruturais ou membranas, como descrito na Seção 3.5.

5.1 Sobreposição de Partículas de Curzon

Partindo da métrica (4.2) podemos calcular a solução para n partículas no eixo z [22]

$$\phi = - \sum_{p=1}^n \frac{m_p}{R_p}, \quad (5.1)$$

$$\nu = -\frac{1}{2} \sum_{p=1}^n \left(\frac{m_p \sin \theta_p}{R_p} \right)^2 - \sum_{\substack{p,q=1 \\ p \neq q}}^n \frac{m_p m_q}{(z_p - z_q)^2} \sin \left(\frac{\theta_p - \theta_q}{2} \right), \quad (5.2)$$

onde R_i é a distância Euclidiana da i -ésima singularidade até o ponto P e θ_p é o ângulo formado entre o segmento que une estes pontos e o eixo z .

Se tomarmos apenas duas partículas fixas em $z_{1,2}$ com massas $m_{1,2}$ as expressões acima simplificam-se e a função ϕ pode ser escrita como

$$\phi = -\frac{m_1}{R_1} - \frac{m_2}{R_2}, \quad (5.3)$$

enquanto ν torna-se $\nu = \nu_1 + \nu_2 + \nu_{12}$ onde $\nu_{1,2}$ são os potenciais métricos associados a cada uma das partículas isoladas e ν_{12} fica sendo

$$\nu_{12} = -\frac{m_1 m_2}{(z_2 - z_1)^2} \sin^2 \left(\frac{\theta_1 - \theta_2}{2} \right) \quad (5.4)$$

ou substituindo $\sin \theta_i$ por r/R_i

$$\nu = \frac{2m_1 m_2}{(z_2 - z_1)^2} \left(\frac{r^2 + (z - z_1)(z - z_2)}{R_1 R_2} - 1 \right). \quad (5.5)$$

Nos dois primeiros gráficos da Figura 5.1 podemos ver, respectivamente, as curvas de nível das funções ϕ e ν , relativas a sobreposição de duas partículas de Curzon. Esta solução apresenta

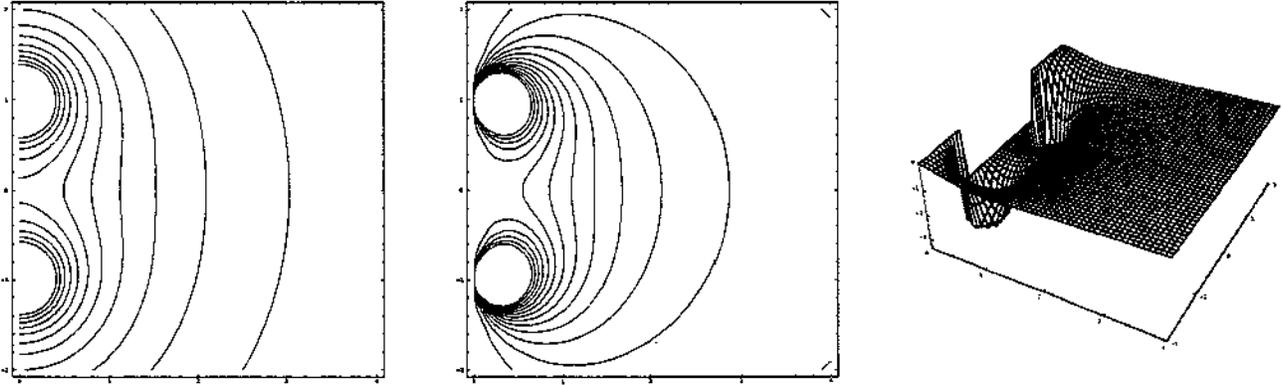


Figura 5.1: Sobreposição de duas partículas de Curzon.

uma singularidade estrutural, pois ν não se anula no eixo z entre as partículas. Esta estrutura pode ser melhor observada no terceiro gráfico da Figura 5.1, onde apresentamos o gráfico de superfície de ν .

Na região no eixo z entre as partículas a esta função assume o valor

$$\nu(0, z) = -4 \frac{m_1 m_2}{(z_1 - z_2)^2} \quad \text{para } z_1 < z < z_2. \quad (5.6)$$

Definindo $D = z_1 - z_2$ como a distância entre as partículas a força de compressão na estrutura será [51]

$$F_z = \frac{1}{4} \left[\exp \left(4 \frac{m_1 m_2}{D^2} \right) - 1 \right] \approx \frac{m_1 m_2}{D^2} + 2 \frac{(m_1 m_2)^2}{D^4} \quad (5.7)$$

Note que o primeiro termo desta série é exatamente a força Newtoniana que seria necessária para manter as partículas separadas.

Cooperstock [19] aplicando esta solução obteve um resultado aproximado para a radiação emitida por dois corpos em queda retilínea. Tema que foi retomado por Araújo et. al. [3].

5.2 Barras Sobrepostas

Do mesmo modo que obtivemos a solução para a sobreposição de partículas de Curzon podemos calcular a sobreposição de soluções de Schwarzschild em coordenadas de Weyl [62], [51].

A solução para duas barras, tais que $l_i = m_i$ para $i = 1, 2$, separadas por $2D$, é dada por

$$\phi = \frac{1}{2} \ln \left(\frac{R_1 + R_2 - 2\mu_1}{R_1 + R_2 + 2\mu_1} \frac{R_3 + R_4 - 2\mu_2}{R_3 + R_4 + 2\mu_2} \right), \quad (5.8)$$

$$\nu = \frac{1}{2} \left[\ln \frac{(R_1 + R_2)^2 - 4\mu_1^2}{4R_1 R_2} + \ln \frac{(R_3 + R_4)^2 - 4\mu_2^2}{4R_3 R_4} + \ln \left(\frac{D}{D + \mu_2} \right)^2 + \ln \left(\frac{(\mu_2 + D)R_1 + (\mu_1 + \mu_2 + D)R_2 - \mu_1 R_4}{DR_1 + (\mu_1 + D)R_2 - \mu_1 R_3} \right)^2 \right], \quad (5.9)$$

onde R_1 e R_2 , são as distâncias no espaço de base até pontos extremos da primeira barra, enquanto R_3 e R_4 são as distâncias relativas a segunda barra.

5.3 Solução Geral para Dois Corpos

Como citado na Seção 4.12, a solução obtida pela sobreposição das autofunções do Laplaciano em coordenadas bi-esféricas é a solução geral para dois corpos [13].

O potencial ϕ é dado simplesmente pela soma das autofunções

$$\phi = \sqrt{\cosh \eta - \cos \theta} \sum_{s=0}^{\infty} H_s(\eta) \Theta_s(\theta),$$

onde definimos as funções

$$H_s(\eta) = A_s e^{\eta(s+\frac{1}{2})} + B_s e^{-\eta(s+\frac{1}{2})},$$

$$\Theta_s(\theta) = \mathcal{P}_s(\cos \theta).$$

Pela integração de (2.8) obtemos a função ν

$$\begin{aligned} \nu = \operatorname{sen} \theta \sum_{p,s=0}^{\infty} & \left\{ \operatorname{sen} \theta \Lambda_{ps} \left[\Theta'_p \Theta'_s + \Theta_p \Theta_s \left(s(s+1) + \frac{1}{2} \right) \right] \right. \\ & - \operatorname{sen} \theta \Theta_p \Theta_s \left(H_p H'_s \operatorname{senh} \eta + \frac{1}{2} H_p H_s \cosh \eta - \Omega_{ps} \right) \\ & \left. + \Theta'_p \Theta_s \left[\cos \theta (2\Omega_{ps} + \Lambda_{ps}) - 2\Pi_{ps} \right] \right\} + F(\theta), \end{aligned}$$

onde foram definidas

$$\Lambda_{ps} = \int H_p H_s \cosh \eta \, d\eta, \quad \Pi_{ps} = \int H_p H'_s \, d\eta, \quad \Omega_{ps} = \int H_p H'_s \cosh \eta \, d\eta$$

e a função de integração pode ser

$$\begin{aligned} F(\theta) = -2 \sum_{s=0}^{\infty} A_s B_s & \left[2 \left(s + \frac{1}{2} \right)^2 I_s^s - (1 - \xi^2) \Theta_s \Theta'_s \right] \\ & + \frac{1}{2} \sum_{s=0}^{\infty} K_s \left[\frac{4(s+1)}{2s+1} \left((s+1) I_{s+1}^{s+1} + s I_{s+1}^{s-1} \right) + \xi \left(\Theta_{s+1}^2 \Theta_s^2 \right) + I_{s+1}^{s+1} - I_s^s \right], \end{aligned}$$

onde $\xi = \cos \theta$, $K_s = A_{s+1} B_s + A_s B_{s+1}$ enquanto que $I_p^s = \int_{\xi}^1 \Theta_p \Theta_s \, d\xi$.

Como esta solução é assintoticamente plana sua massa total é dada pela soma das massas de cada termo da série

$$m = -\frac{a}{\sqrt{2}} \sum_{s=0}^{\infty} (A_s + B_s)$$

e a massa de cada partícula é

$$m_1 = -\frac{a}{\sqrt{2}} \sum_{s=0}^{\infty} A_s, \quad m_2 = -\frac{a}{\sqrt{2}} \sum_{s=0}^{\infty} B_s.$$

5.4 Discos Circundando Buracos Negros

Estas métricas podem ser consideradas como modelos de galáxias, pois de acordo com as observações astronômicas a maioria das galáxias é composta por um núcleo denso, um disco e um halo. O núcleo será representado pelo buraco negro, o disco de estrelas por um disco infinitesimalmente fino porém o halo é desconsiderado.

O tensor de energia momento de um disco infinitesimalmente fino, sobre o eixo z , com um buraco negro de Schwarzschild no centro. Seus componentes não nulos são [46]

$$\epsilon = -T_{\dot{t}\dot{t}}^t = 4e^{2(\Psi+\Lambda-\nu)}[1 - r(\Psi + \Lambda)_r] \Lambda_z \delta(z), \quad (5.10)$$

$$p_{\varphi\varphi} = T_{\varphi\varphi}^\varphi = 4re^{2(\Psi+\Lambda-\nu)}(\Psi + \Lambda)_r \Lambda_z \delta(z), \quad (5.11)$$

onde Λ e $\nu[\Lambda]$ são os potenciais métricos do disco, Ψ e $\nu[\Psi]$ são os potenciais métricos do buraco negro, e $\nu = \nu[\Psi + \Lambda]$.

Uma característica destas soluções é que a transformação (2.32), que leva a solução de Schwarzschild das coordenadas de Weyl para suas coordenadas canônicas, leva o disco infinitesimal em outro disco infinitesimal.

Apresentamos agora uma lista de soluções onde diferentes discos foram sobrepostos ao buraco negro.

Discos de Morgan-Morgan

Para o caso dos discos de Morgan-Morgan (4.18) sobrepostos com um buraco negro o tensor de energia momento (5.10)–(5.11) assume os valores

$$\epsilon = \sqrt{1 - \frac{r^2}{a^2}} \left[1 - \left(\frac{3\pi r^2}{4a^3} + \frac{m}{\sqrt{m^2 + r^2}} \right) \right] G \delta(z),$$

$$p_{\varphi\varphi} = \sqrt{1 - \frac{r^2}{a^2}} \left(\frac{3\pi r^2}{4a^3} + \frac{m}{\sqrt{m^2 + r^2}} \right) G \delta(z),$$

onde M é a massa do disco e m é a massa do buraco negro. G é uma função dos potenciais métricos [46]. A condição fraca de energia, $\epsilon > 0$, aplicada a este caso torna-se

$$0 \leq \frac{m^2}{a^2} \leq \left(1 - \frac{3\pi M}{4a} \right)^2 \left[\frac{3\pi M}{2a} \left(1 - \frac{3\pi M}{8a} \right)^2 \right]^{-1}$$

e a velocidade de rotação das partículas (4.16) é

$$v^2 = \left(\frac{3\pi M r^2}{4a^3} + \frac{m}{\sqrt{m^2 + r^2}} \right) \left(1 - \frac{3\pi M r^2}{4a^3} - \frac{m}{\sqrt{m^2 + r^2}} \right)^{-1}.$$

Disco de Poeira

Para obter a sobreposição da solução de Schwarzschild (4.4) com a solução de um disco de poeira (4.32)–(4.33) nós primeiro reescrevemos (4.32) como $\phi_D^+ = \alpha \ln \mu^+$, então a função ϕ do conjunto é escrita como

$$\phi^+ = \phi_{BH} + \phi_D^+ = \frac{1}{2} \ln \left(\frac{\mu^+ \mu_2}{\mu_1} \right).$$

No plano $z = 0$ e a função ν é

$$\nu|_{z=0} = \frac{\alpha^4}{16r^4} + \frac{1}{2} \ln \left(\frac{r^2}{r^2 + m^2} \right) + \frac{\alpha^2}{2} \left(\frac{\mu_1}{r(r - \mu_1)^2} - \frac{\mu_2}{r(r - \mu_2)^2} \right).$$

Notando que neste plano $\mu_1 = m + \sqrt{r^2 + m^2}$ e $\mu_2 = -m + \sqrt{r^2 + m^2}$ e aplicando estes potenciais para calcular as componentes do tensor de energia momento obtemos

$$\epsilon = \frac{2\alpha^2}{r^3} e^{2(\phi-\nu)} \left(1 - \frac{m}{\sqrt{r^2 + m^2}} \right), \quad p_{\varphi\varphi} = \frac{2\alpha^2}{r^3} \frac{m}{\sqrt{r^2 + m^2}} e^{2(\phi-\nu)},$$

como também a velocidade de rotação

$$v^2 = \frac{m}{\sqrt{r^2 + m^2} - m}.$$

Com esta expressão vemos que quando $r < \sqrt{3}m$ a velocidade se torna superluminal $v^2 > 1$, note que $\sqrt{3}m$ é exatamente o raio de Schwarzschild, no plano equatorial, em coordenadas de Weyl. Esta região nos obriga a abandonar esta solução como um modelo razoável para uma galáxia real.

Discos Anulares

Usando as expressões (5.10)–(5.11) e (4.35) e nos restringindo a $r \geq a$, pode-se mostrar que

$$\epsilon = G \sqrt{1 - \frac{a^2}{r^2}} \left(1 - \left[\frac{M}{r} \left(1 - \frac{3a^2}{2r^2} \right) + \frac{m}{\sqrt{r^2 + m^2}} \right] \right),$$

$$p_{\varphi\varphi} = G \sqrt{1 - \frac{a^2}{r^2}} \left[\frac{M}{r} \left(1 - \frac{3a^2}{2r^2} \right) + \frac{m}{\sqrt{r^2 + m^2}} \right],$$

onde G é uma expressão envolvendo os potenciais de Weyl, veja [46].

Quando $r = a$ temos que $\epsilon = p_{\varphi\varphi} = 0$, enquanto que para $r \rightarrow \infty$ estas funções assumem os comportamentos assintóticos definidos por $\epsilon = O(r^{-2})$ e $p_{\varphi\varphi} = O(r^{-4})$. Desta forma a condição $\epsilon \geq 0$ é sempre satisfeita para todos os valores de M/a e m/a .

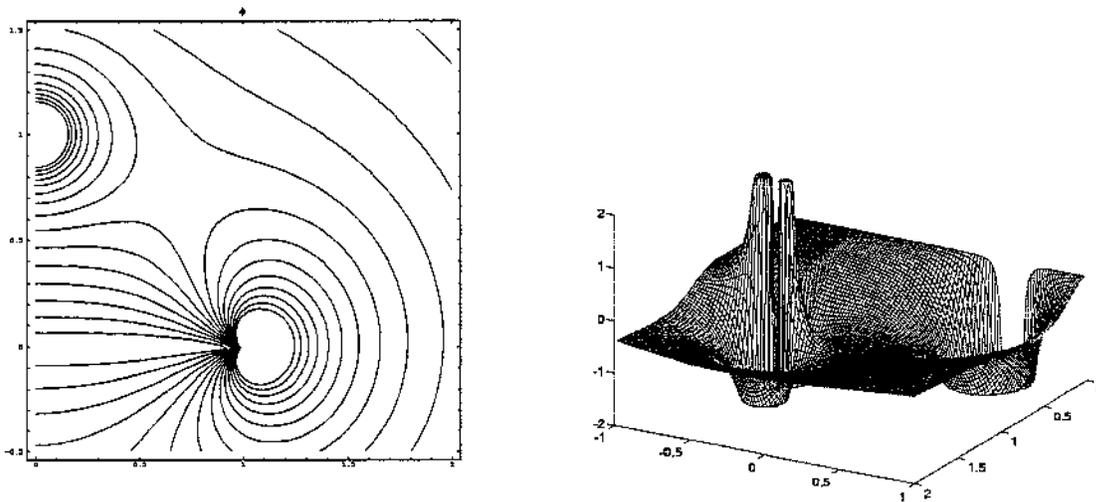


Figura 5.2: Partícula de Curzon e Disco de Appell em equilíbrio. Funções ϕ e ν .

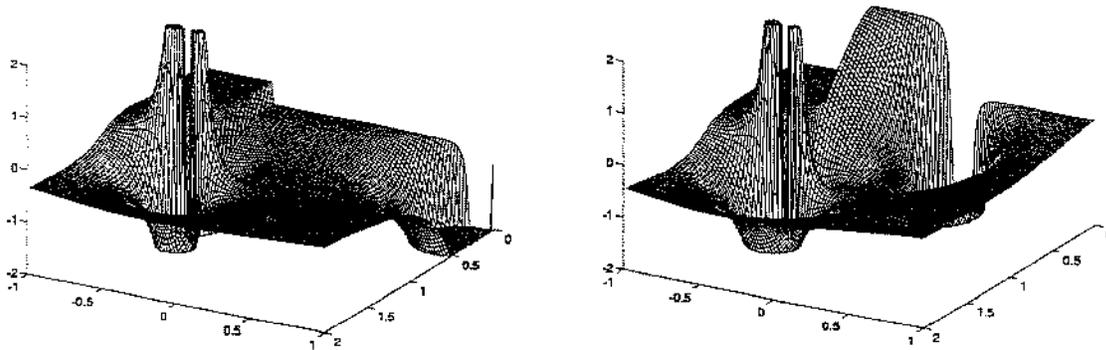


Figura 5.3: Função ν para uma partícula de Curzon e um disco de Appell, atração e repulsão.

5.5 Partícula de Curzon e Disco de Appell

A sobreposição de um disco de Appell, no plano $z = 0$ com raio a ϕ^a e ν^a , (4.37) e (4.42), e uma partícula de Curzon em z_0 , ϕ^c e ν^c , (4.2), é dada pelas funções [49]

$$\phi = \phi^a + \phi^c, \quad \nu = \nu^a + \nu^c + \nu^{ac},$$

onde

$$\nu^{ac} = \frac{2mM}{(z_0^2 + a^2)^2} \left(-B + [AB + 2z_0a^2D] P + [BD - 2z_0A] Q \right),$$

$$P = \frac{\sqrt{\cosh \eta - \cos \xi}}{\sqrt{2aR}} \cos \frac{\xi}{2}, \quad Q = \frac{\sqrt{\cosh \eta - \cos \xi}}{\sqrt{2R}} \sin \frac{\xi}{2},$$

$A = r^2 + (z - z_0)z$, $B = z_0^2 - a^2$ e $D = z - z_0$ enquanto que η e ξ são as coordenadas toroidais definidas na Seção B.5.

O primeiro gráfico da Figura 5.2 exibe as curvas de nível da função ϕ . Porém a função ν terá diferenças significantes dependendo da posição da partícula. Como o interior do disco é composto de matéria negativa e sua borda por matéria positiva [28], o ponto $r = 0$ e $z = a$ será uma posição de equilíbrio. Então posicionando a partícula de Curzon neste ponto temos uma solução em equilíbrio. A função ν correspondente não possui a singularidade estrutural, como mostra o segundo gráfico de Figura 5.2. Porém, existe uma membrana no interior do disco, que se anula nas proximidades do eixo z .

Quando posicionamos a partícula fora deste ponto, a solução final será instável e possuirá tanto a membrana quanto a estrutura, como podemos ver nos gráficos da Figura 5.3. No primeiro a partícula está mais afastada do disco que o ponto de equilíbrio, portanto, a atração é mais forte e a estrutura, como nas outras soluções, é negativa. Enquanto no segundo arranjo, com a partícula mais próxima que o ponto de equilíbrio, a estrutura é positiva.

5.6 Sobreposições com o Anel de Letelier-Oliveira

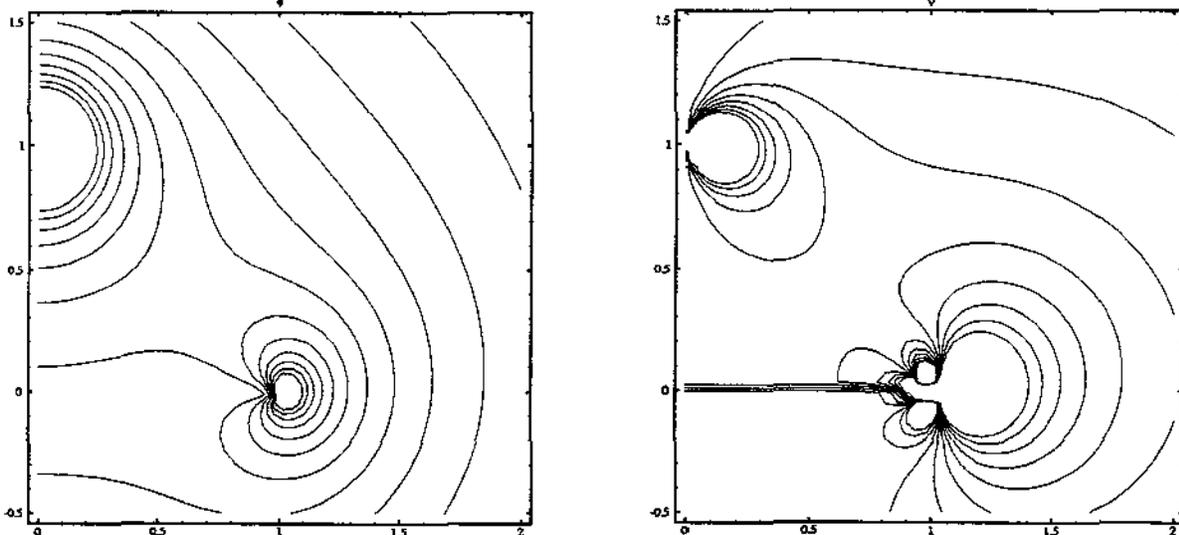


Figura 5.4: Anel de Letelier-Oliveira e uma partícula de Curzon.

Sobrepondo uma partícula de Curzon com um anel de Letelier e Oliveira [51] obtemos uma métrica que contém uma estrutura e uma membrana. Sejam ϕ^c e ν^c os potenciais métricos da partícula de Curzon (4.2) no ponto z_0 , com massa m , enquanto ϕ^a e ν^a são a solução para o anel (4.48)–(4.49) no plano $z = 0$ com raio a e massa M . Então a sobreposição é dada por

$$\phi = \phi^c + \phi^a, \quad \nu = \nu^c + \nu^a + \nu^{ca}, \quad (5.12)$$

onde a função ν^{ca} é escrita como

$$\nu^{ca} = mM \left\{ \text{Re} \left[\frac{1}{(z - z_0 - ia)} \left(\frac{r^2 + (z - ia)(z - z_0)}{R_a R_0} - 1 \right) \right] + \text{Im} \left[\frac{2\mu_0}{aR_0(\mu_0 - \mu_a)} \right] - \frac{2}{z_0^2 + a^2} \right\}.$$

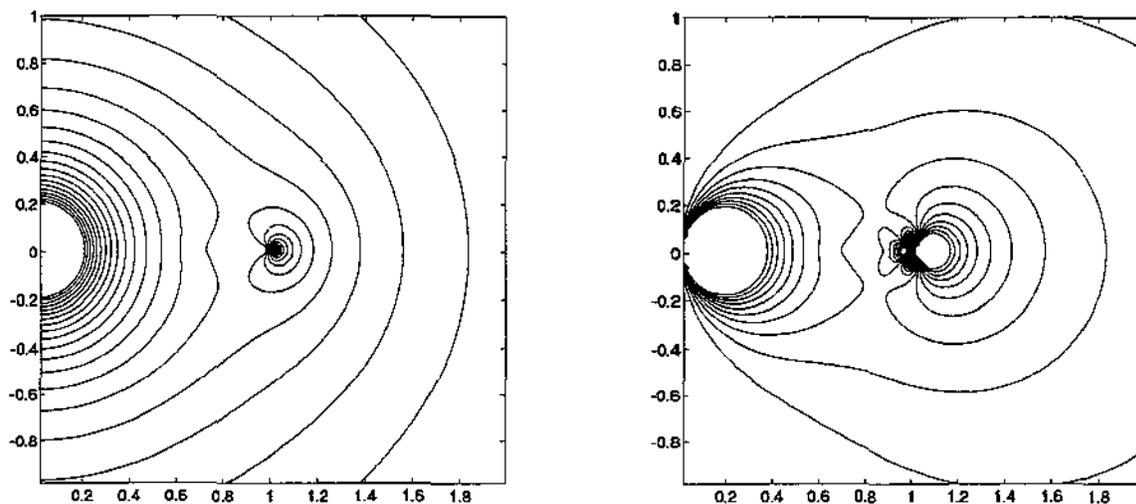


Figura 5.5: Anel de Letelier-Oliveira e uma partícula de Curzon em equilíbrio.

Os gráficos da Figura 5.4 exibem esta solução para o caso $z_0 = 1$ e $a = 1$, neste arranjo temos o aparecimento tanto da membrana quanto da estrutura, porém na Figura 5.5, onde $z_0 = 0$, os corpos estão em equilíbrio e as singularidades não aparecem.

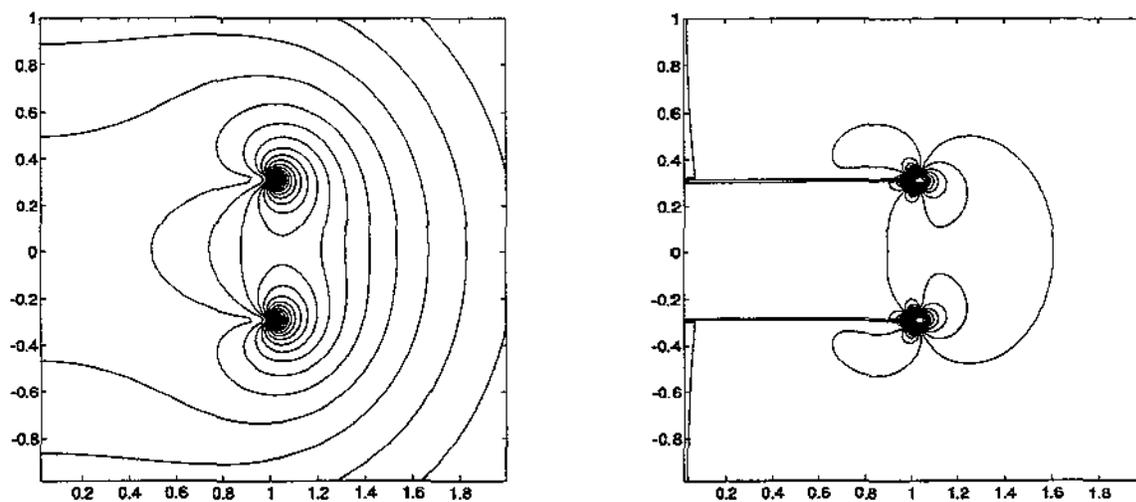


Figura 5.6: Dois anéis de Letelier-Oliveira.

Por métodos numéricos calculamos também a sobreposição de dois anéis paralelos, como mostra a Figura 5.6. Esta métrica contém duas membranas, uma no interior de cada anel e uma estrutura ligando-as.

5.7 Sobreposições com o Anel de Weyl-Bach

A sobreposição de um anel de Weyl (4.51) e (4.53), ϕ^w e ν^w , com um campo homogêneo (4.1) ϕ^h e ν^h é dada pelas funções [36]

$$\phi = \phi^w + \phi^h, \quad \nu = \nu^w + \nu^h + \nu^{wh},$$

onde

$$\nu^{wh} = -4\sigma\lambda\sqrt{\cosh\eta - 1} (H'F(\delta, k) + HE(\delta, k))$$

a função H é definida em (4.52), F e E são funções elípticas de Jacob, enquanto que σ é a densidade Netoniana do anel e λ é a intensidade do campo homogêneo. Os argumentos das função elípticas são

$$\delta = \operatorname{sen}^{-1} \sqrt{\frac{(\cosh\eta + 1)(1 - \cos\xi)}{2(\cosh\eta - \cos\xi)}}, \quad k = \sqrt{\frac{2}{\cosh\eta + 1}}.$$

η e ξ são coordenadas toroidais. Esta solução é um exemplo de métrica com membrana, porém a presença do campo homogêneo dificulta a interpretação física de suas fontes.

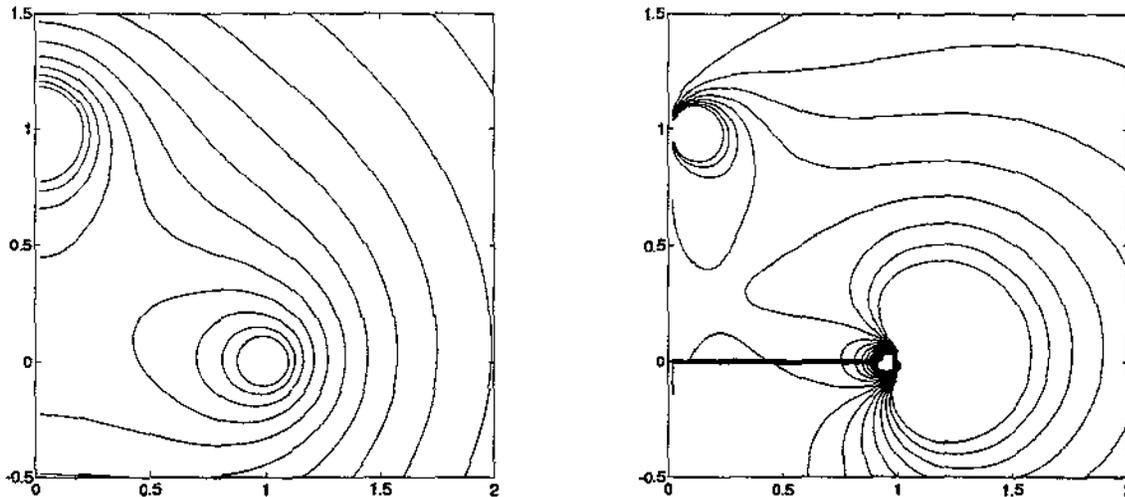


Figura 5.7: Anel de Weyl-Bach e uma partícula de Curzon.

Apresentamos agora algumas soluções, obtidas numericamente, que correspondem a sobreposições de várias métricas com o anel de Weyl-Bach.

Começamos pelo anel com uma partícula de Curzon, Figura 5.7. Este arranjo nos permite uma comparação direta com a solução analítica da sobreposição de uma partícula de Curzon com o anel de Letelier-Oliveira (5.12).

Calculamos também esta métrica com os componentes em equilíbrio. O resultado, visto na Figura 5.8, como esperávamos, não apresenta estruturas ou membranas.

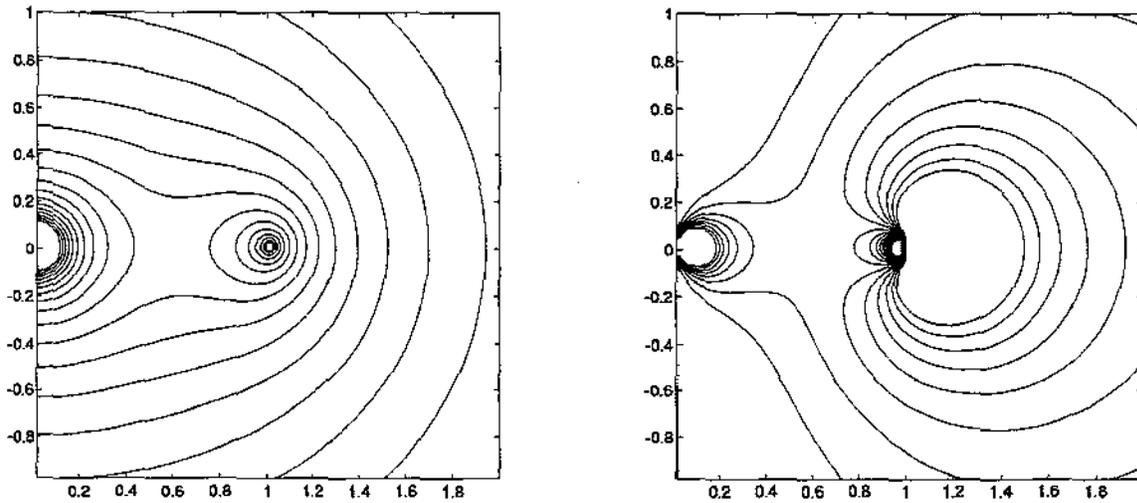


Figura 5.8: Anel de Weyl-Bach e uma partícula de Curzon em equilíbrio.

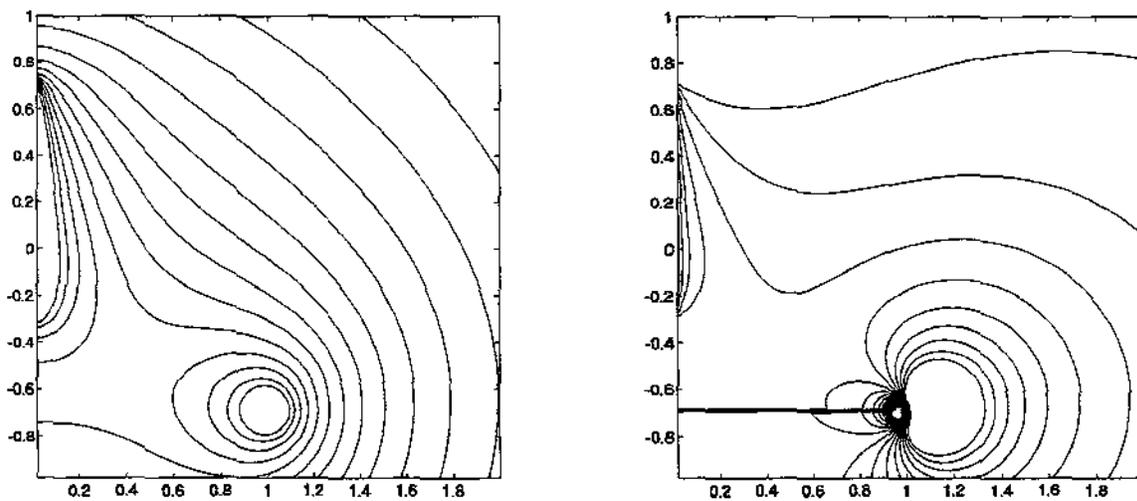


Figura 5.9: Anel de Weyl-Bach com a solução de Schwarzschild.

Uma solução fisicamente mais plausível é obtida substituindo a partícula de Curzon pela solução de Schwarzschild, embora estas soluções ainda não sejam realistas, pois o anel contém matéria negativa.

Esta nova solução foi calculada com três arranjos diferentes. Primeiro consideramos as disposições equivalentes às calculadas com a partícula de Curzon. Por fim, consideramos uma situação intermediária, que será descrita mais a frente.

A primeira disposição corresponde as soluções das Figuras 5.4 e 5.7 onde a partícula encontra-se afastada do anel, veja a Figura 5.9. Como nos casos anteriores existe uma membrana no interior do anel e uma estrutura entre o centro da membrana e a partícula.

A outra situação estudada é a solução em equilíbrio onde colocamos a barra, que representa o buraco negro, fica centrada no plano do anel. A solução resultante pode ser vista na Figura 5.10

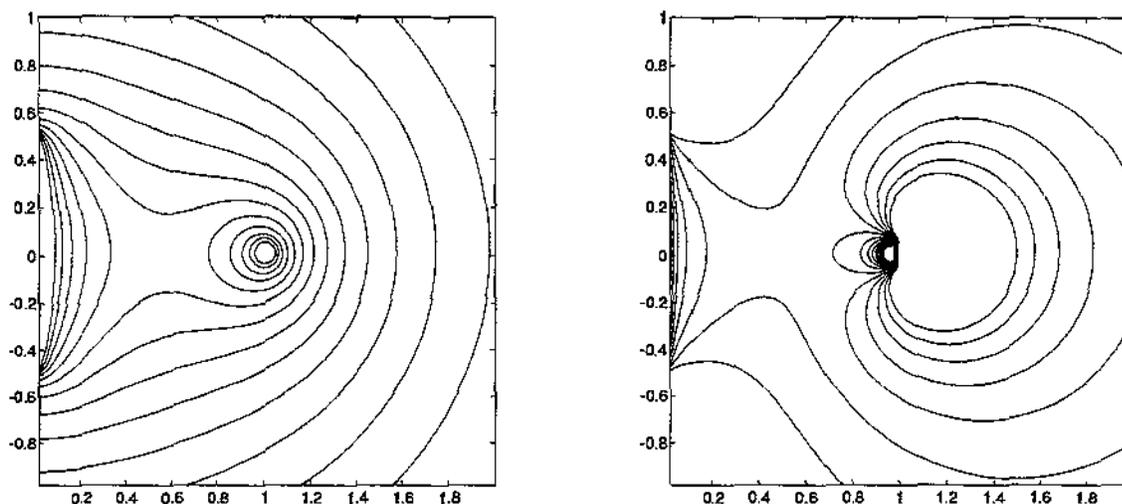


Figura 5.10: Anel de Weyl-Bach com a solução de Schwarzschild em equilíbrio.

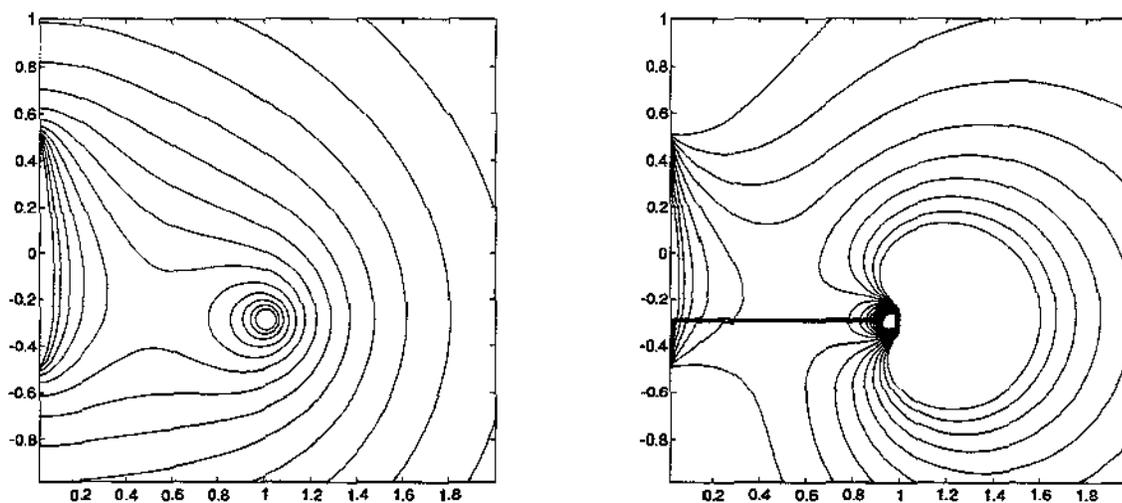


Figura 5.11: Anel de Weyl-Bach com a solução de Schwarzschild interceptando-se.

(compare com as figuras 5.5 e 5.8).

A terceira disposição não possui equivalente com as métricas envolvendo a partícula de Curzon. Nela posicionamos a barra no interior do anel, mas fora da posição de equilíbrio. Esta solução pode ser vista na Figura 5.11. Note que a estrutura não aparece enquanto que a membrana surge normalmente no interior do anel. Talvez seja lícito conjecturar que a solução estendida, isto é, com o interior do horizonte de Schwarzschild, a membrana se estendesse até o eixo de simetria e a estrutura aparecesse sobre este eixo.

Como no caso dos anéis de Letelier-Oliveira, calculamos a sobreposição de dois anéis paralelos, veja a Figura 5.12.

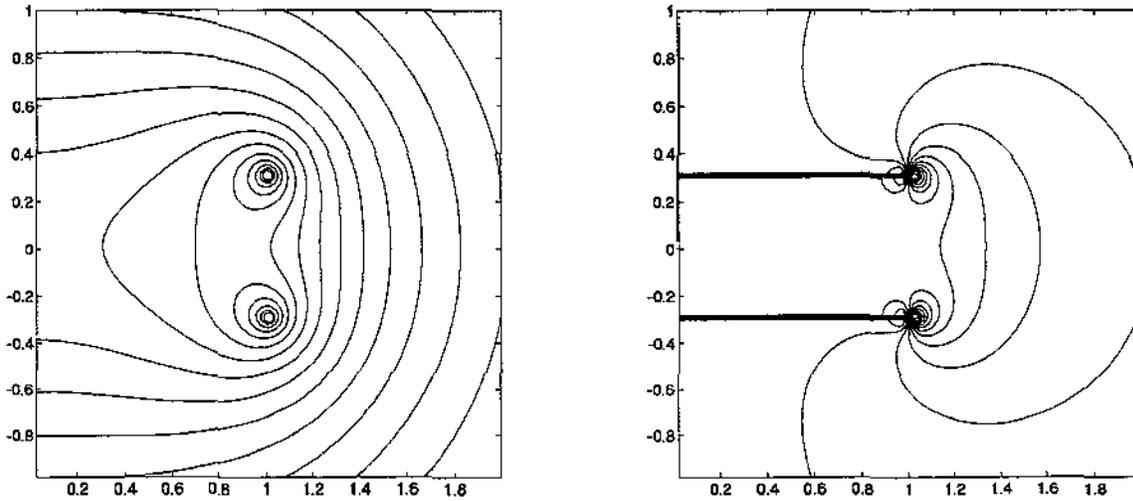


Figura 5.12: Dois anéis de Weyl-Bach

5.8 Corpos Acelerados

Como foi citado na Seção 4.7 podemos gerar soluções das equações de Einstein associadas com corpos acelerados pela sobreposição de uma solução de Weyl com uma linha semi-infinita de densidade $1/2$.

Considere que ϕ_0 seja o potencial métrico de uma linha semi-infinita (4.11) com densidade $1/2$. E que ϕ_1 seja a Imagem Newtoniana da configuração de massa que desejamos acelerar. Pela linearidade da equação de Laplace temos $\phi = \phi_0 + \phi_1$ e com ajuda da identidade (2.14) podemos escrever $\nu = \nu_0 + \nu_1 + \nu_{01}$, onde $\nu_i = \nu[\phi_i]$, e $\nu_{01} = \nu[\phi_0, \phi_1]$.

Quando aplicamos a transformação (4.12) nesta métrica os termos ϕ_0 e ν_0 desaparecem, resultando no elemento de linha

$$ds^2 = e^{2\phi_1} \frac{(\zeta d\tau - \tau d\zeta)^2}{q^2 (\zeta^2 - \tau^2)} - e^{2(\nu_1 + \nu_{01} - \phi_1)} \left[d\eta^2 + \frac{(\tau d\tau + \zeta d\zeta)^2}{\zeta^2 - \tau^2} \right] - e^{-2\phi_1} \eta^2 d\varphi^2 \quad (5.13)$$

Para que esta métrica seja regular sobre as retas $\tau^2 = \zeta^2$, portanto extensível a todo o plano $\tau - \zeta$ ela deve obedecer a condição de regularidade

$$\lim_{\tau^2 - \zeta^2 \rightarrow 0} q^2 e^{2(\nu_1 + \nu_{01})} = 1 \quad (5.14)$$

A constante q , que foi introduzida na transformação (4.12), pode ser usada para ajustar as coordenadas a esta condição. Esta métrica também é invariante as transformações citadas na Seção 4.7.

Partículas Aceleradas

Tomando a solução para a linha semi infinita, ϕ_0 e ν_0 , com $z_0 = 0$, $\epsilon = 1$ e $\sigma = 1/2$. E ϕ_i , ν_i com $i = 1, 2$ como soluções de Curzon. A sobreposição destas três métricas é dada por [7]

$$\phi = \phi_0 + \phi_1 + \phi_2, \quad (5.15)$$

$$\nu = \nu_0 + \nu_1 + \nu_2 + \nu_{12} + \nu_{012}, \quad (5.16)$$

onde ν_{12} é dada por (5.5) e

$$\nu_{012} = \frac{(R+z_1)m_1}{z_1 R_1} + \frac{(R+z_2)m_2}{z_2 R_2} + 1/2 \ln R + C. \quad (5.17)$$

$R_{1,2}$ são as distâncias Euclidianas, até os pontos $z_{1,2}$, R é a distância até a origem e C é uma constante arbitrária, cujo valor determina qual intervalo do eixo z será regular ou conterá uma singularidade estrutural.

Buracos Negros Acelerados

Para obtermos a solução das equações de Einstein para buracos negros acelerados devemos sobrepor a solução para uma linha semi-infinita (4.11) com densidade linear $\sigma = 1/2$ com a solução de Schwarzschild (4.4). Consideramos que a linha parte do ponto $z_2 = (2A)^{-1}$ e dirige-se ao infinito positivo, isso é, $\epsilon = 1$. Ao mesmo tempo que impomos que a massa do buraco negro obedeça a relação $m < (2A)^{-1}$, isso é, as imagens Newtonianas das soluções não se sobreponham. A sobreposição destas soluções é dada por [9]

$$\phi = \phi_{BH} + \phi_{SL},$$

$$\nu = 1/2 \ln \left(\frac{R_1 R_2 + r^2 + D(z-m)}{R_2 R_3 + r^2 + D(z+m)} \frac{R_1 R_3 + r^2 + z^2 - m^2}{4R_1 R_2 R_3} \frac{(R_2 - D)(R_1 + R_3 - 2m)}{R_1 + R_3 + 2m} \right),$$

onde $z_1 = m$ e $z_3 = -m$ são os extremos da barra que representa o buraco negro e $D = z - z_2$

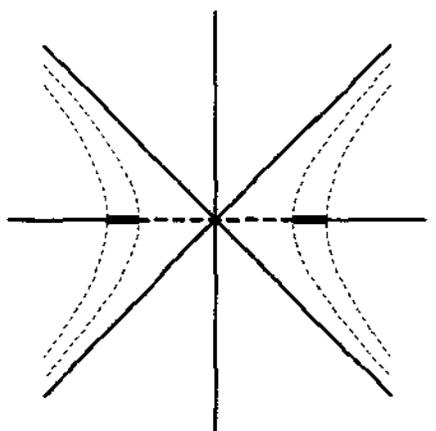


Figura 5.13: Buracos negros acelerados.

As constantes foram fixadas de modo que a singularidade cônica apareça entre a linha e a barra. De modo que, após a aplicação da transformação (4.12) teremos o espaço-tempo da Figura 5.13. Note que a barra original transformou-se em duas, ou seja, estamos lidando com dois buracos negros. A estrutura presente entre a barra e a linha aparece agora entre as duas barras, como mostra a linha tracejada, esta singularidade seria a responsável pela aceleração das partículas.

As barras partem do infinito, em $\tau = -\infty$, com a velocidade da luz. Quando chegam em $\tau = 0$, com a menor distância entre si, suas velocidades são momentaneamente nulas, então continuando o movimento acelerado atingem novamente a velocidade da luz quando $\tau = +\infty$. As linhas pontilhadas indicam as trajetórias dos seus extremos.

Como as superfícies $\tau = \pm\zeta$ são horizontes de partículas para as partícula que seguem trajetórias do tipo (4.13) jamais haverá iteração causal entre os dois buracos negros.

Buracos Negros Acelerados Sobre uma Onda Gravitacional

Partindo da solução da sub-seção anterior e introduzindo o potencial para o campo homogêneo (4.1) podemos eliminar a singularidade estrutural [9].

Pela linearidade do Laplaciano o novo ϕ é exatamente a soma dos anteriores. Enquanto a função ν torna-se

$$\nu = \frac{1}{2} \ln \left(\prod_{i=2}^3 [R_1 R_i + r^2 + D_i(z - m)] \right) \quad (5.18)$$

$$+ \frac{1}{2} \ln \left(\frac{e^{2\lambda(R_2+R_3-R_1)-\lambda z}}{R_1 + R_3 + 2m} \frac{(R_2 - D_2)(R_1 + R_3 - 2m)\lambda z \alpha^2}{R_1 R_2 R_3 [R_2 R_3 + r^2 + D_2(z + m)]} \right),$$

onde $D_i = z - z_i$, λ fixa a intensidade do campo homogêneo e α é uma constante arbitrária. Para removermos a singularidade precisamos ajustar estas constantes, impondo as equações

$$e^{-4m\lambda} = \frac{1 - 2mA}{1 + 2mA}, \quad \alpha^2 = \frac{e^{-\frac{\lambda}{A}}}{4A} \frac{1 + 2mA}{1 - 2mA}$$

quando $mA \ll 1$ e $m \neq 0$ estas expressões podem ser escritas como

$$\lambda = A \left(1 + \frac{4m^2 A^2}{3} + \mathcal{O}(m^4 A^4) \right), \quad \alpha^2 = \frac{1 + 4mA + \mathcal{O}(m^2 A^2)}{4Ae}.$$

Após a aplicação da transformação (4.12) o campo homogêneo torna-se semelhante a uma onda gravitacional onde os dois buracos negro estão imersos.

5.9 A Métrica C

Esta métrica foi descoberta por Levi-Civita [54] como a *soluzioni oblique*, caso particular dos seus espaços estáticos degenerados. Ele buscava soluções que possuíssem um vetor de Killing tipo-tempo ortogonal ao espaço tridimensional, cujo tensor de curvatura de Ricci fosse da forma

$$R^a_b = \alpha \eta^a \eta_b + \beta \delta^a_b. \quad (5.19)$$

O que é equivalente a considerar a métrica estática com tensor de Weyl tipo {22}.

Por apresentar muitas características interessantes esta métrica foi descrita, por Kinnersley e Walker [39], como exemplo para quase tudo, em contraste ao espaço de Taub-NUT, que é um contra exemplo para quase tudo.

Posteriormente Ehlers e Kundt [24] apresentaram-na como um caso particular das métricas C, daí seu nome hoje. Estas métricas podem ser representadas pelo elemento de linha

$$ds^2 = (\tilde{x} + \tilde{y})^{-2} \left[F d\tilde{t}^2 - F^{-1} d\tilde{y}^2 - G^{-1} d\tilde{x}^2 - G d\tilde{p}^2 \right] \quad (5.20)$$

com F e G representando polinômios cúbicos nas coordenadas \tilde{y} e \tilde{x} , ou seja, $F = F(\tilde{y})$ e $G = G(\tilde{x})$, tais que a expressão seja válida

$$F(\tilde{y}) = -G(-\tilde{y}).$$

Originalmente Levi-Civita escolheu os coeficientes do polinômio $G(x) = a_0 + a_1x + a_2x^2 + a_3x^3$ impondo que $a_3 = 1$ e $a_2 = 0$. Letelier e Oliveira [53] mostraram que não é necessário fixar estes coeficientes porém aqui seguiremos a notação adotada por Cornish e Uttley [20] onde o elemento de linha tem a forma

$$ds^2 = A^{-2} (x + y)^{-2} \left[K^2 F dt^2 - F^{-1} dy^2 - G^{-1} dx^2 - K^{-2} G dp^2 \right], \quad (5.21)$$

$$\text{onde } G = 1 - x^2 - 2mAx^3. \quad (5.22)$$

No caso eletricamente carregado teríamos um polinômio de quarto grau dado por $G = 1 - x^2 - 2mAx^3 - e^2A^2x^4$, mas não trataremos deste caso aqui, pois estamos interessados na métrica C apenas como um caso particular das soluções de Weyl, que são soluções para o vácuo.

Com base na nossa escolha da assinatura de um espaço-tempo devemos restringir os valores da coordenada x de modo que $G(x) > 0$, pois caso contrário teremos uma métrica com assinatura invertida $(-+++)$. Com esta restrição, se $F(y) > 0$ a coordenada t é tipo-tempo, e se $F(y) < 0$ então y é tipo-tempo.

Bonnor [8] e Cornish e Uttley [20] fazem a restrição $F > 0$ de modo que t é sempre a coordenada tipo-tempo, porém Kinnersley e Walker [39] ignoram esta restrição trabalhando com um escopo maior para a coordenada y , que desta forma permuta com t suas características tempo-espaciais. Inicialmente seguiremos os passos de Bonnor, Cornish e Uttley, pois eles apresentam explicitamente a métrica C nas coordenadas de Weyl. Ao final, apresentaremos os resultados obtidos por Kinnersley e Walker.

Vetores de Killing e Geodésicas

Uma característica importante do sistema de coordenadas em (5.21) é a adequação aos dois vetores de Killing que a métrica possui, nominalmente $\xi^a = \delta_0^a$ e $\zeta^a = \delta_3^a$, com as normas

$$\|\xi\| = \frac{F(y)}{A^2(x+y)^2}, \quad \|\zeta\| = -\frac{G(x)}{A^2(x+y)^2}. \quad (5.23)$$

Além disso ela também está adaptada ao autovetor, $\eta^a = \delta_2^a$, do tensor de Ricci tridimensional (5.19). Nos domínios em que $F(y) > 0$ e $G(x) > 0$ os vetores ξ e ζ são respectivamente tipo-tempo e tipo-espaço. Fato que justifica, ao menos nestas regiões, o estudo da métrica C como uma solução de Weyl.

As geodésicas nulas tais que $\dot{x} = \dot{p} = 0$, onde o ponto simboliza derivação pelo parâmetro afim u , são dados por

$$t(u) = \pm \frac{1}{K} \int \frac{du}{F(u)}, \quad y(u) = u. \quad (5.24)$$

A Métrica C nas Coordenadas de Weyl

Para escrevermos a métrica C nas coordenadas de Weyl devemos aplicar a transformação

$$z = \frac{1 + mAx y(x - y) + xy}{A(x + y)^2}, \quad r = \frac{\sqrt{F}\sqrt{G}}{A(x + y)^2} \quad (5.25)$$

a coordenada p é substituída por φ e t permanece inalterada. Comparando (5.21) com (2.6) concluímos que

$$e^{2\phi} = \frac{K^2 F(y)}{(x + y)^2}. \quad (5.26)$$

Esta é a transformação apresentada no artigo de Cornish e Uttley. Para que as coordenadas r , z e t sejam equiparadas com as do artigo de Bonnor elas devem ser multiplicadas por A^{-1} .

O caso $m = 0$

Neste caso a transformação para as coordenadas de Weyl (5.25) simplifica-se em

$$z = \frac{1 + xy}{A(x + y)^2}, \quad r = \frac{\sqrt{1 - x^2}\sqrt{y^2 - 1}}{A(x + y)^2} \quad (5.27)$$

e podemos efetuá-la para as regiões onde $F(y) = y^2 - 1 > 0$ e $G(x) = 1 - x^2 > 0$. Estas regiões estão expostas no primeiro gráfico da Figura 5.14, com as curvas isoparamétricas da transformação (5.27).

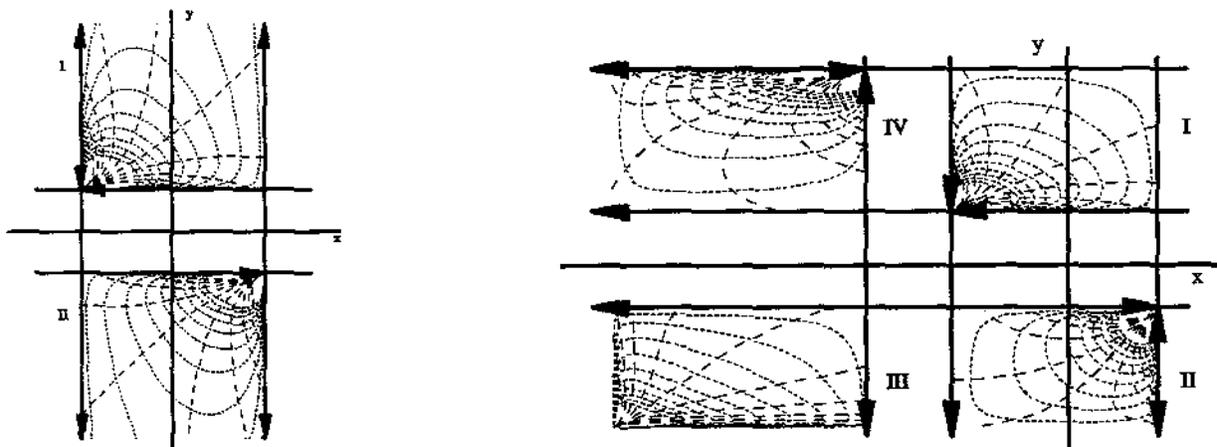


Figura 5.14: Plano xy quando $m = 0$ e quando $m \neq 0$ as linhas pontilhadas indicam curvas onde r é contante, enquanto as tracejadas z contante.

Como ambas as regiões conduzem ao mesmo resultado escolhemos a região I, de modo que teremos

$$e^{2\phi} = \frac{K^2(y^2 - 1)}{(x + y)^2}, \quad e^{2\nu} = \frac{K^2(y^2 - 1)}{y^2 - x^2}. \quad (5.28)$$

Note que $y \rightarrow 1$ implica que $\phi \rightarrow -\infty$, mas $|\phi| < \infty$ em todos os demais pontos da região I. Então a Imagem Newtoniana relacionada com ϕ é uma linha semi-infinita de densidade $1/2$ extendendo-se de $z = (2A)^{-1}$ até $z \rightarrow \infty$. De (5.27) e (5.28) podemos ver que

$$\phi = \frac{1}{2} \log \left[\sqrt{r^2 + \left(z - \frac{1}{2A}\right)^2} - \left(z - \frac{1}{2A}\right) \right] + \frac{1}{2} \log AK^2,$$

$$e^{2(\nu-\phi)} = \frac{y-x}{x+y} = 2A \sqrt{r^2 + \left(z - \frac{1}{2A}\right)^2}.$$

Com estas fórmulas podemos verificar que a condição de regularidade (3.4) é satisfeita para todo $z < (2A)^{-1}$ desde que $K^2 = 1$.

Como vimos anteriormente a solução de Weyl para uma linha semi-infinita pode ser levada em um espaço plano uniformemente acelerado pela transformação (4.12), que nesse caso será empregada com as constantes $B = A$, $a = (2A)^{-1}$ e $q = 1$. Note que com este valor de q a condição (5.14) é satisfeita trivialmente. Concluimos assim que a métrica C com $m = 0$ corresponde ao espaço de Minkowski uniformemente acelerado.

Acompanhando as transformações realizadas vemos que os horizontes $y = \pm 1$ tornam-se os planos $\zeta = \pm \tau$ que são os horizontes dos sinais luminosos emitidos por partículas uniformemente aceleradas com histórias dadas por (4.13).

O Caso $m \neq 0$ e $27m^2A^2 < 1$

A restrição realmente desejada é que o polinômio G possua três raízes reais diferentes, a expressão $27m^2A^2 < 1$ surge da escolha imposta aos coeficientes de G . Porém esta escolha não é necessária, pois podemos trabalhar com um polinômio geral [53].

Empregando a escolha dos coeficientes, descrita anteriormente, o polinômio $G(x)$ tem três raízes reais distintas x_1, x_2 e x_3 onde $x_3 < x_2 < 0 < x_1$, e as raízes de $F(y)$ são $y_i = -x_i$. Quando impomos as condições $F(y) > 0$ e $G(x) > 0$ isolamos as regiões I, II, III e IV do plano $x - y$, como mostra o segundo gráfico da Figura 5.14. Onde elas aparecem com as linhas isoparamétricas da transformação (5.25).

Pelo termo que define a coordenada r na expressão (5.25) vemos que quando $x = x_i$ ou $y = y_i$ então $r = 0$ e com o emprego das fórmulas de F e G vemos que se $x \rightarrow -\infty$ ou $y \rightarrow -\infty$, r também se anula. Portanto, as fronteiras destas regiões são mapeadas no eixo z . Agora com o uso da definição de z (5.25) vemos que estas fronteiras, após transformadas, cobrem todo o eixo z . Empregando (5.25) e as propriedades das raízes de G

$$x_i x_j x_k = (2mA)^{-1}, \quad x_i x_j + x_j x_k + x_i x_k = 0.$$

As fronteiras são mapeadas continuamente sobre o eixo z , exceto nos vértices onde $z \rightarrow \pm\infty$, pois nestes casos o valor de z depende da direção pela qual o vértice é atingido. Enquanto que o interior de cada região é mapeado em todo o plano $r - z$, mas esse mapeamento é bijetor somente nas regiões I e III.

Agora por (5.26) notamos que $e^{2\phi}$ é finita e não se anula no interior das regiões, porém nas fronteiras existem pontos onde $e^{2\phi} \rightarrow -\infty$ indicando a presença de fontes nestas regiões.

Observando primeiramente a região I os valores de z em cada um dos vértices desta região são, segundo a Figura 5.14, $z_i = -mx_i$, de modo que $z_1 < 0 < z_2 < z_3$. Quando tomamos um valor fixo para z e fazemos $r \rightarrow \infty$, no plano $r - z$ a imagem deste ponto no plano $x - y$ aproxima-se do vértice (x_2, y_2) .

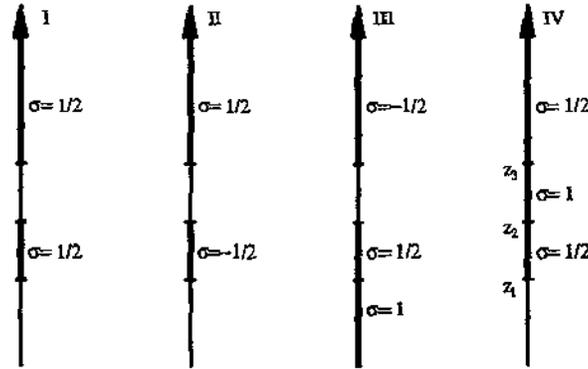


Figura 5.15: Imagem Newtoniana da métrica C.

A expressão (5.26) pode ser escrita como

$$e^{2\phi} = \frac{K^2 [R_3 - (z - z_3)] [R_1 - (z - z_1)]}{R_2 - (z - z_2)},$$

onde $R_i = \sqrt{(z - z_i)^2 + r^2}$. Esta fórmula pode ser reconhecida como a sobreposição de três linhas semi-infinitas que se estendem ao longo do eixo z até $+\infty$. Duas delas com densidade σ igual a $1/2$, cada uma partindo de z_1 ou z_3 . A outra partindo de z_2 com $\sigma = -1/2$. De modo que o resultado é uma linha semi-infinita de densidade $\sigma = 1/2$ que se estende de z_3 ao infinito e uma barra também de densidade $\sigma = 1/2$ entre z_1 e z_2 , como mostra a primeira linha da Figura 5.15, as outras regiões podem igualmente ser mapeadas no plano $r - z$ gerando, como Imagens Newtonianas as outras linhas da Figura 5.15. Porém, apenas a região I leva a um espaço tempo fisicamente aceitável, deste modo trataremos um pouco mais sobre este caso.

Vamos agora analisar a regularidade da métrica nas partes do eixo z tais que $\sigma = 0$, isto é, $-\infty < z < z_1$ e $z - 2 < z < z_3$. Cada um destes intervalos corresponde a partes da fronteira de I onde $x = x_2$ e $x = x_1$, respectivamente. Então usando o fato de que $G \rightarrow 0$ quando $x \rightarrow x_1$ ou $x \rightarrow x_2$ temos que

$$e^{-2\nu} \rightarrow \left(\frac{G_x}{2K} \right)_{x=x_1 \text{ ou } x=x_2}^2$$

então, para que (3.4) seja satisfeita, temos que impor

$$K^2 = \left(\frac{G_x}{2} \right)^2,$$

mas como $G_x(x_1) \neq G_x(x_2)$ esta condição só pode ser satisfeita para uma das regiões, ou seja, a métrica conterá uma singularidade cônica entre $-\infty$ e z_1 ou entre z_2 e z_3 . Escolhendo impor a regularidade no primeiro intervalo a expressão (5.9) pode ser reescrita como

$$K^2 = \frac{A^2}{m^2} (z_2 - z_3)^2 (z - z_1)^2.$$

Vimos que a Imagem Newtoniana desta métrica é uma linha semi-infinita de densidade $\sigma = 1/2$ e uma barra também de densidade $1/2$, ou seja, uma partícula de Schwarzschild em coordenadas de Weyl. Retomando os resultados da Seção 5.8 concluímos que estamos lidando com duas partículas esféricas aceleradas.

Aplicando a transformação (4.12) com $B = AK^2$, $a = z_3$, obtemos a métrica C na forma (5.13). Para satisfazer a condição de regularidade (5.14) basta impor $q = 1$.

Singularidades e Horizontes da Métrica C

Nas sub-seções anteriores trabalhamos com as restrições $G > 0$ e $F > 0$. Mas como foi dito Kinnersley e Walker [39], empregaram uma abordagem diferente impondo somente que $x_2 < x < x_1$ e permitindo, a priori, que y seja ilimitado. Deste modo, a região sob estudo passa a conter os horizontes $F = 0$ nas superfícies $y = y_i$.

Inicialmente definimos as coordenadas

$$Au = t + \int^y \frac{d\xi}{F(\xi)}, \quad A\rho = (x + y)^{-1}.$$

Note que u é uma coordenada nula, veja (5.24), e construímos o invariante de curvatura

$$\Psi = -\frac{1}{2} C_{abcd} l^a n^b l^c n^d = -\frac{m}{\rho^3},$$

onde l^a e n^a são os vetores nulos principais. Com base neste invariante vemos que $\rho = 0$ é uma singularidade física. Como $\rho \rightarrow \infty \Rightarrow \Psi \rightarrow 0$ o espaço é assintoticamente plano, podemos então interpretar ρ como uma coordenada radial, de modo que consideraremos $\rho > 0$. Mas ao impormos esta restrição estamos também impondo uma restrição a y , ou seja, $y > -x$. Este é o domínio com que trabalharemos.

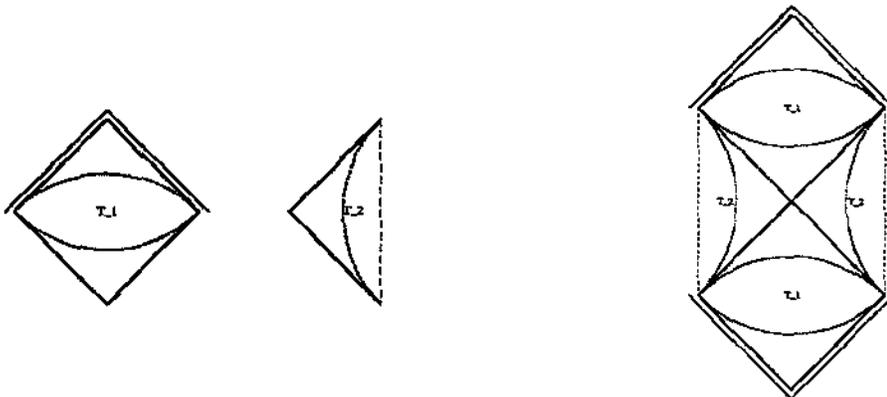


Figura 5.16: Blocos e diagrama da métrica C para o caso $m = 0$.

Tomamos agora um x fixo e desconsiderando a coordenada simétrica p , definimos uma superfície bidimensional $y - t$ cuja métrica é

$$ds^2 = A^{-2} (x + y)^{-2} [F dt^2 - F^{-1} dy^2], \quad (5.29)$$

que é conformal ao elemento de linha das coordenadas de Walker (3.6). Na verdade esta superfície possui as propriedades necessárias para que se empregue o método de blocos, exceto que ela é completamente geodésica somente para as geodésicas nulas. Se mesmo assim empregarmos o método obtemos os resultados a seguir.

Quando $m = 0$, F possui duas raízes ± 1 , mas devido as restrições em y a raiz $y = -1$ está fora do escopo considerado. Deste modo T será dividida em duas regiões $T_{1,2}$, tais que

$$\begin{aligned} T_1, & \quad -x < y < 1, \quad F < 0; \\ T_2, & \quad 1 < y < \infty, \quad F > 0. \end{aligned}$$

Observe que nos extremos destes intervalos temos os seguintes comportamentos

$$\begin{aligned} y \rightarrow -x, & \quad \rho \rightarrow \infty, & \quad \Psi \rightarrow 0; \\ y = 1, & \quad \rho = [A(x - 1)]^{-1}, & \quad \Psi = -m[A(x - 1)]^3; \\ y \rightarrow \infty, & \quad \rho \rightarrow 0, & \quad \Psi \rightarrow -\infty. \end{aligned}$$

De modo que T_1 é assintótico e T_2 é singular, como vemos na Figura 5.16. Unindo estes blocos, pelas regras do método de Walker, obtemos o diagrama da Figura 5.16.

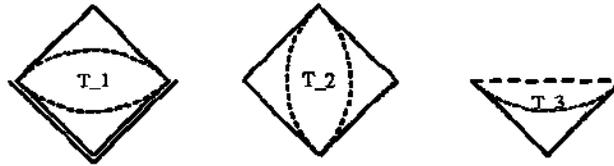


Figura 5.17: Blocos da métrica C para o caso $m \neq 0$

Consideramos agora o caso em que $m \neq 0$ e $27m^2A^2 < 1$, ou seja, quando F possui três raízes reais diferentes $y_1 < y_2 < 0 < y_3$. Novamente como estamos lidando com $y > -x$ e $-y_2 < x < -y_1$ a primeira raiz não nos interessa. Assim dividimos T em três regiões $T_{1,2,3}$ tais que

$$\begin{aligned} T_1, & \quad -x < y < y_2, \quad F < 0; \\ T_2, & \quad y_2 < y < y_3, \quad F > 0; \\ T_3, & \quad y_3 < y < \infty, \quad F < 0. \end{aligned}$$

Repetindo a análise efetuada para o caso $m = 0$, vemos que o comportamento nos extremos destes intervalos é

$$\begin{aligned} y \rightarrow -x, & \quad \rho \rightarrow \infty, & \quad \Psi \rightarrow 0; \\ y = y_2, & \quad \rho = [A(x - y_2)]^{-1}, & \quad \Psi = -m[A(x - y_2)]^3; \\ y = y_3, & \quad \rho = [A(x - y_3)]^{-1}, & \quad \Psi = -m[A(x - y_3)]^3; \\ y \rightarrow \infty, & \quad \rho \rightarrow 0, & \quad \Psi \rightarrow -\infty. \end{aligned}$$

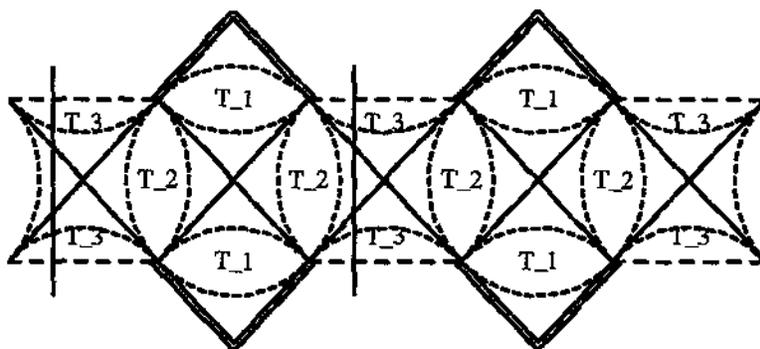


Figura 5.18: Diagrama da métrica C para o caso $m \neq 0$

Portanto, T_1 é assintótico, T_2 é regular e T_3 é singular, veja a Figura 5.17. Unindo-os chegamos ao diagrama da Figura 5.18, que é simplesmente conexo e se repete indefinidamente. Mas como podemos identificar as linhas indicadas, sem violar as condições de causalidade podemos criar um diagrama em forma de cilindro.

Conclusões

Os métodos numéricos mostraram-se adequados ao estudo das soluções de Weyl. Tanto exibindo novas soluções quanto esclarecendo o comportamento de partículas de teste na vizinhança de uma solução. Um bom exemplo das possibilidades destes métodos é o estudo sobre o anel de Weyl-Bach e suas sobreposições.

Como uma revisão este trabalho ainda não está completo. Muitas soluções não foram incluídas, algumas bastante importantes como as de Zipoy [71] e outros discos. Mesmo entre as soluções incluídas faltam dados sobre suas interpretações físicas e geométricas. Porém desejamos continuar desenvolvendo este projeto até podermos apresentar uma revisão mais abrangente.

Referências Bibliográficas

- [1] Abramowitz M. e Stegun I.A, "Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables", National Bureau of Standards, John Wiley & Sons, 10^o ed, 1972
- [2] Appel P, Quelques Remarques sur la Théorie des Potentiels Multiformes, *Ann. Math, Lpz*, 30, 155-156, 1887
- [3] Araújo M.E, Letelier P.S. e Oliveira S.R, Two Kerr Black Holes with Axisymmetric Spins: Head-on Collision and Gravitational Radiation, *Class. Quantum Grav*, 15, 3051-3060, 1998
- [4] Bateman H, "Partial Differential Equations of Mathematical Physics", Cambridge University Press, 1969
- [5] Bel L, Les sources des solutions statiques de Schwarzschild et Curzon, *Gen. Relativ. Gravit*, 1, 337-347, 1971
- [6] Birgin E.G, "Diferenciação Computacional e Aplicações", Tese de Doutorado, IMECC-UNICAMP, 1998
- [7] Bonnor W.B. e Swaminarayan N.S, An Exact Solution for Uniformly Accelerated Particle in General Relativity, *Zeitschrift für Physik*, 177, 240-256, 1964
- [8] Bonnor W.B, The Sources of the Vacuum C-Metric, *Gen. Rel. Grav*, 15, 535-551, 1983
- [9] Bonnor W.B, An exact solution of Einstein's equation for two particles falling freely in an external gravitational field, *Gen. Relativ. Gravit*, 20, 607-622, 1988
- [10] Bonnor W.B. and Martins M.A.P, The interpretation of some static vacuum metrics, *Class. Quantum Grav*, 8, 727-738, 1991
- [11] Bonnor W.B. and Davidson W, Interpreting the Levi-Civita vacuum metric, *Class. Quantum Grav*, 9, 2065-2068, 1992
- [12] Bonnor W.B, Physical interpretations of vacuum solutions of Einstein's equations-Part 1-Time-independent solutions, *Gen. Relativ. Gravit*, 24, 551-573, 1992
- [13] Carminati J. e Sarracino R.S, R-Separable Solutions of Einstein Field Equations, *J. Phys. A: Math. Gen*, 15, 2401-2413, 1982
- [14] Carter B, Killing Horizons and Orthogonally Transitive Groups in Space-Time, *J. Math. Phys.* 10, 70-81, 1969
- [15] Carot J. and Verdaguier E, Generalised soliton solutions of the Weyl class, *Class. Quantum Grav*, 6, 845-856, 1989
- [16] Chanduri S. and Banerji S, Axially symmetric solution generated by complexification technique, *Gen. Relativ. Gravit*, 16, 375-380, 1984
- [17] Chazy M.J, Sur le champ de gravitation de deux masses fixes dans la théorie de la relativité, *Bull. Soc. Math. France*, 52, 17-38, 1924
- [18] Clément G, A class of stationary axisymmetric solutions of Einstein-Maxwell scalar field theories, *Class. Quantum Grav*, 1, 283-289, 1984
- [19] Cooperstock F.I, The axially-symmetric two-body problem, *Gen. Relativ. Gravit*, 6, 91-97, 1975

- [20] Cornish F.H.J. and Uttley W.J, The interpretation of the C metric. The vacuum case, *Gen. Relativ. Gravit*, 27, 439-454, 1995
- [21] Cornish F.H.J. and Uttley W.J, The interpretation of the C metric. The charged case when $e^2 \leq m^2$, *Gen. Relativ. Gravit*, 27, 735-749, 1995
- [22] Curzon H.E.J, Cylindrical solutions of Einstein's gravitation equations, *Proc. London Math. Soc*, 23, 477-480, 1924
- [23] d'Inverno R, "Introducing Einstein's Relativity", Clarendon Press - Oxford, 1992
- [24] Ehlers J. and Kundt W, "Gravitation. An Introduction to Current Research", L. Witten, ed. (Wiley, New York), 1962
- [25] Erez G. e Rosen N, *Bull. Res. Council. of Israel*, 8F, 47, 1959
- [26] Felice F, Potential surfaces for time-like geodesics in the Curzon metric, *Gen. Relativ. Gravit*, 23, 125-149, 1991
- [27] Geroch R, Multipole moments. II. Curved space, *J. Math. Phys*, 11, 2580-2588, 1970
- [28] Gleiser R.J. and Pullin J.A, Appell rings in general relativity, *Class. Quantum Grav*, 6, 977-985, 1989
- [29] Godfrey B.B, Horizons in Weyl Metrics Exhibiting Extra Symmetries, *Gen. Relativ. Gravit*, 3, 3-16, 1972
- [30] Gutsnayeve Ts.I. and Manko V.S, On the gravitational field of a mass possessing a multipole moment, *Gen. Relativ. Gravit*, 17, 1025-1027, 1985
- [31] Hawking S.W. e Ellis G.F.R, "The Large Scale Structure of Space-Time", Cambridge, 1973
- [32] Hernández-Pastora J.L, Manko V.S. and Martín J, Some asymptotically flat generalizations of the Curzon metric, *J. Math. Phys*, 34, 4760-4774, 1993
- [33] Hernández-Pastora J.L. and Martín J, Monopole-Quadrupole static axisymmetric solutions of Einstein's fields equations, *Gen. Relativ. Gravit*, 26, 887-907, 1994
- [34] Hoenselaers C, Comment - A note in the Weyl solutions, *Class. Quantum Grav*, 5, 1045-1050, 1988
- [35] Hoenselaers C, Axially stationary vacuum solutions near a ring singularity, *Class. Quantum Grav*, 7, 581-584, 1990
- [36] Hoenselaers C, The Weyl solution for a ring in a homogeneous field, *Class. Quantum Grav*, 12, 141-148, 1995
- [37] Kahaner D, Moler C. e Nash S, "Numerical Methods and Software", Prentice Hall, New Jersey, 1989
- [38] Kinnersley W, *J. Math. Phys*, 10, 1195-, 1969
- [39] Kinnersley W. e Walker M, Uniformly Accelerating Charged Mass in General Relativity, *Phys. Rev*, D2, 1359-1370, 1970
- [40] Kinnersley W, Generation of Stationary Einstein-Maxwell Fields, *J. Math. Phys*, 14, 651-653, 1973
- [41] Kramer D, Stephani H, Herlt E. e MacCallum M, "Exact Solutions of einstein's Field Equations", Cambridge University Press, Cambridge, 1979
- [42] Kruskal M.D, Maximal Extension of Schwarzschild Metric, *Phys. Rev*, 119, 1743-1745, 1960
- [43] Ledermann W. and Vajda S, "Handbook of Applicable Mathematics", Vol IV, John Wiley & Sons, Binghamton NY, 1982
- [44] Lemos J.P.S. and Letelier P.S, Superposition of Morgan and Morgan discs with a Schwarzschild black-hole, *Class. Quantum Grav*, 10, L75-L78, 1993

- [45] Lemos J.P.S. and Letelier P.S, Topological defects with axial symmetry: a disk + a string, *Gen. Relativ. Gravit*, 25, 365-372, 1993
- [46] Lemos J.P.S. and Letelier P.S, Exact general relativistic thin disk around black holes, *Physical Review D*, 49, 5135-5143, 1994
- [47] Letelier P.S, Static and stationary multiple soliton solutions to the Einstein equations, *J. Math. Phys*, 26, 467-475, 1985
- [48] Letelier P.S. and Oliveira S.R, Exact self-gravitating disks and rings: A solitonic approach, *J. Math. Phys*, 28, 165-170, 1987
- [49] Letelier P.S. and Oliveira S.R, Superposition of Weyl solutions of the Einstein equations: cosmic strings and domain walls, *Class. Quantum Grav*, 5, L47-L51, 1988
- [50] Letelier P.S, Intermediate Singularities in the Interaction of Gravitational and Eletromagnetec Waves with Cosmic Strings, *Phys. Rev. D*, 47, 1709-1711, 1993
- [51] Letelier P.S. e OLiveira S.R, Superposition of Weyl Solutions: the Equilibrium Forces, *Class. Quantum Grav*, 15, 421-433, 1998
- [52] Letelier P.S, On the Gravitational Field of Static and Stationay Axial Symmmetric Bodies with Multi-Polar Structure, *Class. Quantum Grav*, 16, 1999
- [53] Letelier P.S. e Oliveira S.R, On Uniformly Accelerated Black Holes, pre-print
- [54] Levi-Civita T, *Rendi. Accad. Lincei*, 27, 343, 1917
- [55] Linet B, The static metrics with cylindrical symmetry describing a model of cosmic strings, *Gen. Relativ. Gravit*, 17, 1109-1115, 1985
- [56] Manko V.S, On a general static axisymmetric solution of the Einstein vacuum equations, *Gen. Relativ. Gravit*, 21, 1193-1195, 1989
- [57] Moon P. and Spencer D.E, "Field Theory Handbook", Springer-Verlag, 2nd ed, 1988
- [58] Morgan T. e Morgan L, The Gravitational Field of a Disk, *Phys. Rev*, 183, 1097-1101, 1969
- [59] Nakamura S, "Applied Numerical Methods in C", Prentice Hall International Editions, 1993
- [60] Quevedo H, On the exterior gravitational field if a mass with a multipole moment, *Gen. Relativ. Gravit*, 19, 1013-1023, 1987
- [61] Quevedo H, General static axisymmetric solution of Einstein's vacuum field equations in prolate spheroidal coordinates, *Physical Review D*, 39, 2904-2911, 1989
- [62] Robertson H.P. and Noonan T.W, "Relativity and Cosmology", W.B.Saunders Co, 2nd ed, 1969
- [63] Silva M.F.A, Herrera L, Paiva F.M. and Santos N.O, The Levi-Civita space-time, *J. Math. Phys*, 36, 3625-3631, 1995
- [64] Scott S.M. and Szekeres P, The Curzon Singularity 1: Spatial Sections, *Gen. Relativ. Gravit*, 18, 557-570, 1986
- [65] Scott S.M. and Szekeres P, The Curzon Singularity 2: Global Pictures, *Gen. Relativ. Gravit*, 18, 571-583, 1986
- [66] Synge J.L, "Relativity: The General Theory", North-Holland publishing company, Amsterdam, 1960
- [67] Thorne K.S, A toroidal solution of the vacuum Einstein field equations, *J. Math. Phys*, 16, 1860-1865, 1975
- [68] Wald R.M, "General Relativity", Univ. Chicago Press, 1984

- [69] Walker M, Block Diagrams and the Extension of Timelike Two-Surfaces, J. Math. Phys, 11, 2280-2286, 1970
- [70] Weyl H. and Bach R, Neue Lösungen der Einsteinschen Gravitationsgleichungen Math. Z, 13, 119-142, 1922
- [71] Zipoy D.M, Topology of Some Spherical Metrics, J. Math. Phys, 7, 1137-1143, 1966

Apresentamos neste apêndice, resumidamente, alguns teoremas sobre singularidades, demonstrados por Hawking e Ellis [31]. Em seu livro, Wald [68] também apresenta estes resultados com diferenças muito pequenas, em algumas definições. Não incluímos este conteúdo no corpo da dissertação pois estes teoremas se aplicam a espaço-tempos evolutivos e demonstram a existência de singularidades como o Big-Bang ou o colapso gravitacional, enquanto as métricas de Weyl, tema deste trabalho, são estáticas, não satisfazendo, portanto, as condições necessárias para a aplicação dos teoremas.

As primeiras seções deste apêndice definem uma série de conceitos que serão depois empregados para determinar com clareza o conteúdo dos teoremas. Para o caso das métricas estáticas, como as de Weyl, os resultados mais relevantes são os que caracterizam as singularidades através da incompletude geodésica.

A.1 Estrutura Causal

Apresentamos aqui o conceito de espaço-tempo temporalmente orientável e sua estabilidade. Estes conceitos serão necessários para demonstrar os teoremas sobre singularidades.

Considere M uma variedade diferenciável e g_{ab} uma métrica, com assinatura de Lorentz $(-, +, +, +)$, definida em M , então chamamos (M, g_{ab}) de um espaço-tempo e cada $p \in M$ de evento. Os vetores t^a são considerados tipo-tempo se $g_{ab}t^at^b < 0$, tipo-espaço se $g_{ab}t^at^b > 0$ e nulos se $g_{ab}t^at^b = 0$.

A.2 Orientação Temporal

Em cada ponto $p \in M$ o seu espaço tangente, V_p , é isomórfico ao espaço de Minkowski, onde podemos fazer uma escolha local da direção do futuro. Se pudermos escolher continuamente a direção do futuro, em todo M , então dizemos que o espaço é temporalmente orientável, neste caso podemos provar que

Lema 1 *Seja (M, g_{ab}) temporalmente orientável. Então existe um campo vetorial tipo-tempo, t^a , analítico e diferente do vetor nulo, em M . Em contra partida se existe um campo vetorial tipo-tempo, contínuo, então (M, g_{ab}) é temporalmente orientável.*

Em um espaço temporalmente orientável podemos definir os seguintes conceitos:

- Curva tipo-tempo orientada para o futuro, $\lambda(l)$ se para qualquer $p \in \lambda$ o vetor tangente t^a é tipo-tempo direcionado para o futuro;

- Curva causal orientada para o futuro, $\lambda(l)$ se para qualquer $p \in \lambda$ o vetor tangente t^a é tipo-tempo ou nulo direcionado para o futuro;
- Futuro cronológico, $I^+(p)$ é o conjunto de todos os pontos $q \in M$ tais que existe uma curva tipo-tempo direcionada para o futuro, λ , tal que $\lambda(0) = p$ e $\lambda(1) = q$, podemos também definir o futuro cronológico de um conjunto $S \subset M$ como $I^+(S) = \bigcup_{p \in S} I^+(p)$;
- Futuro causal, $J^+(p)$ é o conjunto de todos os pontos $q \in M$ tais que existe uma curva causal direcionada para o futuro, λ , tal que $\lambda(0) = p$ e $\lambda(1) = q$, da mesma forma para $S \subset M$ temos $J^+(S) = \bigcup_{p \in S} J^+(p)$.

Definições análogas podem ser feitas para conceitos equivalentes direcionados ao passado, λ , I^- e J^- , bastando apenas substituir o termo futuro por passado, nas definições acima. Podemos notar que $I^+(p)$ é um subconjunto aberto de M e que $p \in I^+(p)$ somente se existe uma curva tipo-tempo orientada para o futuro fechada que começa e termina em p . Em contraste, qualquer que seja $p \in M$, $p \in J^+(p)$. O mesmo sendo válido para I^- e J^- .

Definimos também uma vizinhança normal convexa de $p \in M$, que consiste em um conjunto aberto, U , com $p \in U$, tal que para todo $q, r \in U$ existe uma única geodésica conectando q e r contida totalmente em U .

Teorema 1 *Considere um espaço-tempo arbitrário (M, g_{ab}) e um ponto $p \in M$, sempre existe uma vizinhança normal convexa, U , de p . Além disso, para qualquer U , $I^+(p) \cap U$ consiste de todos os pontos alcançados por geodésicas tipo-tempo direcionadas para o futuro que partiram de p e contidas em U , enquanto $\dot{I}^+(p) \cap U$, onde \dot{I}^+ denota a fronteira de I^+ , é gerado pelas geodésicas nulas direcionadas para o futuro, contidas em U , que partiram de p .*

É importante notar que estas caracterizações de I^+ e \dot{I}^+ em termos de geodésicas podem não ser válidas quando consideramos o espaço como um todo, ou seja, em espaço-tempos gerais elas só valem localmente. Mas do teorema acima tiramos o seguinte corolário:

Corolário 1 *Se $q \in J^+(p) - I^+(p)$ então qualquer curva causal conectando p e q precisa ser uma geodésica nula.*

Outro conceito importante é a idéia de conjunto atemporal, que se define como sendo um conjunto $S \subset M$ onde não existem $p, q \in S$ tais que $q \in I^+(p)$, ou seja, $S \cap I^+(S) = \emptyset$. Agora podemos apresentar um teorema que descreve a estrutura de \dot{I}^+ .

Teorema 2 *Seja (M, g_{ab}) um espaço-tempo temporalmente orientável, e considere $S \subset M$. Então $\dot{I}^+(S)$, se não for vazio, é uma C^0 -subvariedade de M , tridimensional, imersa e atemporal.*

Uma estrutura que podemos definir nos conjuntos atemporais fechados $S \subset M$ é sua borda $\partial(S)$, que é o conjunto dos pontos $p \in S$ tais que qualquer vizinhança O de p contém $q \in I^+(p)$, $r \in I^-(p)$ e uma curva tipo-tempo conectando q e r que não intercepta S . Com esta definição concluímos que

Teorema 3 *Seja $S \neq \emptyset$ um conjunto atemporal fechado com $\partial(S) = \emptyset$. Então S é uma C^0 -subvariedade tridimensional imersa em M .*

Em algumas situações, como veremos adiante, queremos estender uma curva diferencial a partir de um ponto ou queremos poder tomar o limite de uma seqüência de curvas diferenciais, o que nem sempre pode ser feito mantendo a diferenciabilidade das curvas, assim definiremos o conceito de curvas contínuas tipo-tempo ou causais direcionadas ao futuro, ou passado.

Uma curva contínua, λ , é dita tipo-tempo, ou causal, direcionada ao futuro se para cada $p \in \lambda$ existe uma vizinhança normal convexa, U , de p tal que se $\lambda(t_1), \lambda(t_2) \in U$ com $t_1 < t_2$ então existe uma curva tipo-tempo, ou causal, diferenciável e direcionada ao futuro em U de $\lambda(t_1)$ até $\lambda(t_2)$.

A.3 Pontos Extremos e Curvas Inextensíveis

Para distinguirmos entre curvas que vão ao infinito ou terminam em uma singularidade de curvas que terminam em um ponto devido somente a sua parametrização definimos o conceito de ponto extremo futuro. Seja λ uma curva causal direcionada ao futuro, $p \in M$ é um ponto extremo futuro de λ se para qualquer vizinhança, O , de p existe t_0 tal que para todo $t > t_0$, $\lambda(t) \in O$. Observe que p não precisa pertencer a curva. A definição para ponto extremo passado é similar.

Agora podemos dizer que uma curva é inextensível para futuro, ou passado, isto é, ela já é o mais "longa" possível, não podendo ser prolongada, quando ela não possui um ponto extremo futuro, ou passado. Note que toda curva que possui um ponto extremo pode ser estendida, ao menos continuamente, a partir deste ponto.

Estas definições levam ao seguinte lema:

Lema 2 *Seja λ uma curva causal inextensível ao passado passando por p . Então através de qualquer $q \in I^+(p)$ existe uma curva tipo tempo, inextensível ao passado, γ , tal que $\gamma \subset I^+(\lambda)$.*

Expomos agora os conceitos de convergência de seqüências de curvas causais. Considere $\{\lambda_n\}$ uma seqüência de curvas causais, então podemos definir os seguintes elementos:

- Um ponto $p \in M$ é ponto de convergência de $\{\lambda_n\}$ se dada qualquer vizinhança aberta O de p , existe N tal que $\lambda_n \cap O \neq \emptyset$ para todo $n > N$;
- Uma curva λ é considerada curva de convergência de $\{\lambda_n\}$ se cada $p \in \lambda$ é um ponto de convergência de $\{\lambda_n\}$;
- Um ponto $p \in M$ é um ponto limite de $\{\lambda_n\}$ se qualquer vizinhança aberta O de p , intercepta infinitas curvas de $\{\lambda_n\}$;
- Uma curva λ é considerada curva limite de $\{\lambda_n\}$ se existe uma subseqüência $\{\lambda'_n\}$ para a qual λ é uma curva de convergência. Note que uma curva cujos pontos são pontos limite de $\{\lambda_n\}$ não é necessariamente uma curva limite.

Lema 3 *Seja $\{\lambda_n\}$ uma seqüência de curvas causais, inextensíveis, direcionadas ao futuro, que possui um ponto limite p . Então existe uma curva causal, inextensível, direcionada ao futuro, λ , que é curva limite de $\{\lambda_n\}$.*

Inicialmente definimos conjuntos que nos dizem o que um evento, ou um conjunto de eventos, pode influenciar, I^+ e J^+ , agora definiremos outros que nos digam quais os eventos que são influenciados apenas por um determinado conjunto de eventos.

O domínio de dependência futuro de um conjunto $S \subset M$, atemporal e fechado, $\mathcal{D}^+(S)$ é o conjunto dos pontos $p \in M$ tais que todas as curvas causais inextensíveis ao passado que passam por p interceptam S . Note que $S \subset \mathcal{D}^+(S) \subset J^+(S)$. Como S é atemporal então $\mathcal{D}^+(S) \cap I^-(S) = \emptyset$. A definição do domínio de dependência passado $\mathcal{D}^-(S)$ é similar. O domínio de dependência é $\mathcal{D}(S) = \mathcal{D}^+(S) \cup \mathcal{D}^-(S)$.

A.4 Conceitos de Estabilidade

Imediatamente percebemos que para que um espaço-tempo possuir uma estrutura temporal bem definida ele não pode possuir curvas tipo tempo fechadas, pois se $p \in I^+(p)$ então este evento estaria influenciando seu próprio passado. Porém, eliminar apenas os espaços onde isso ocorre não elimina todas as possibilidades de mau comportamento temporal. É possível construir espaços que estão à beira de possuírem uma curva tipo-tempo fechada, estes espaços também não podem ser considerados fisicamente aceitáveis, pois são instáveis, podendo ganhar uma curva tipo-tempo fechada com uma pequena perturbação. Exibiremos aqui três condições de estabilidade para um espaço-tempo: causalidade forte, causalidade estável e hiperbolicidade global.

Um espaço-tempo (M, g_{ab}) é considerado fortemente causal se para todo $p \in M$ e toda vizinhança O de p , existe outra vizinhança de p , $V \subset O$, tal que nenhuma curva causal intercepta V mais de uma vez.

Esta definição elimina o que intuitivamente consideramos como a beira de possuir uma curva tipo-tempo fechada, mas é possível construir espaços, que satisfazem a condição de causalidade forte, mas ainda são instáveis.

Outra tentativa para caracterizar adequadamente os espaços é dada a causalidade estável. Considere um espaço (M, g_{ab}) , tomemos agora $p \in M$ e um vetor tipo-tempo t^a em p . Se definirmos $\bar{g}_{ab} = g_{ab} - t_a t_b$, teremos uma nova métrica em p , ainda com assinatura de Lorentz, porém todos os vetores tipo-tempo ou nulos em g são tipo-tempo em \bar{g} , ou seja, esta operação "abre" o cone de luz no ponto p . Agora se tivermos um espaço-tempo à beira de possuir uma curva tipo-tempo fechada e abrirmos todos os seus cones de luz então teremos uma curva fechada no novo espaço, com a métrica \bar{g} . Assim, dizemos que um espaço é causalmente estável se existe um campo vetorial tipo-tempo, não nulo, t^a , tal que o espaço (M, \bar{g}_{ab}) não possui nenhuma curva tipo-tempo fechada.

O teorema seguinte exhibe uma importante característica de espaços causalmente estáveis.

Teorema 4 *Um espaço-tempo é causalmente estável se e, somente se, existe uma função diferenciável f em M , tal que $\nabla^a f$ é um campo vetorial tipo-tempo direcionado ao passado. A função f é chamada de "função global de tempo".*

Como corolário deste teorema temos que causalidade estável implica em causalidade forte.

Outra condição, ainda mais forte, sobre os espaço-tempos é a hiperbolicidade global. Um espaço (M, g_{ab}) é considerado globalmente hiperbólico se possui uma superfície de Cauchy. Onde a superfície de Cauchy é um conjunto atemporal, Σ , tal que $\mathcal{D}(\Sigma) = M$. O termo superfície é justificado pelo fato de que podemos aplicar o Teorema 3, que garante que Σ é uma C^0 -subvariedade tridimensional imersa em M .

A superfície de Cauchy pode ser considerada como um instante de tempo ao longo do espaço. Existem razões para supor que todo espaço-tempo fisicamente aceitável é globalmente hiperbólico. Nestes espaços temos o seguinte resultado:

Teorema 5 *Seja Σ uma superfície de Cauchy e λ uma curva causal inextensível. Então λ intercepta Σ , $I^+(\Sigma)$ e $I^-(\Sigma)$.*

Definiremos agora $C(p, q)$, para $p, q \in M$ onde (M, g_{ab}) é um espaço-tempo fortemente causal, como sendo o conjunto de todas as curvas contínuas tipo-tempo direcionadas ao futuro, que ligam p a q . Onde curvas que diferirem apenas pela parametrização são consideradas iguais. Note que se $q \notin J^+(p)$ então $C(p, q) = \emptyset$. Podemos também definir uma topologia, \mathcal{T} , em $C(p, q)$ para isso definimos que conjunto aberto em $C(p, q)$ é todo conjunto $O \subset C(p, q)$, que pode ser escrito como

$$O = \bigcup O(U),$$

onde $U \subset M$ é um conjunto aberto e $O(U) \subset C(p, q)$ é definido como

$$O(U) = \{\lambda \in C(p, q) \mid \lambda \subset U\}.$$

Com esta topologia em $C(p, q)$ podemos demonstrar o seguinte teorema:

Teorema 6 *Seja (M, g_{ab}) um espaço-tempo globalmente hiperbólico e $p, q \in M$. Então $C(p, q)$ é compacto.*

A compacidade de $C(p, q)$ implica diretamente na compacidade correspondente do conjunto $J^+(p) \cap J^-(q)$ na topologia de M . Reescrevendo temos:

Teorema 7 *Seja (M, g_{ab}) um espaço-tempo globalmente hiperbólico e $p, q \in M$. Então $J^+(p) \cap J^-(q)$ é compacto.*

Podemos também definir o espaço $C(\Sigma, p)$, para $p \in \mathcal{D}(\Sigma)$, que consiste no conjunto de todas as curvas causais contínuas direcionadas ao futuro entre Σ e p . É possível demonstrar, pelos mesmos argumentos do Teorema 6, que $C(\Sigma, p)$ é compacto.

Este é o teorema final sobre condições de estabilidade de um espaço-tempo:

Teorema 8 *Seja (M, g_{ab}) um espaço-tempo globalmente hiperbólico. Então (M, g_{ab}) é causalmente estável. A função global de tempo, f , pode ser escolhida de forma que cada superfície onde f é constante seja uma superfície de Cauchy. Assim, M pode ser foliado por superfícies de Cauchy, Σ , de forma que sua topologia seja $\mathbb{R} \times \Sigma$.*

A.5 Congruências Geodésicas

Entendemos por congruências, em um conjunto aberto $O \subset M$, uma família de curvas tais que por cada ponto $p \in O$ passe uma e somente uma curva desta família. Se tomarmos os vetores tangentes de uma congruência teremos um campo vetorial em O , e similarmente dado um campo vetorial contínuo em O podemos construir uma congruência. Com isso em mente, podemos definir congruência analítica como sendo aquela que gera um campo vetorial analítico.

Consideremos uma congruência analítica de geodésicas tipo-tempo, podemos determinar que sejam parametrizadas pelo seu tempo próprio, então teremos que todos os vetores tangentes ξ^a obedecerão a relação $\xi^a \xi_a = -1$. Assim, podemos definir o tensor

$$B_{ab} = \nabla_a \xi_b,$$

que será puramente espacial. Este tensor mede quanto o vetor de deslocamento infinitesimal entre duas geodésicas falha em ser transportado paralelamente. Para podemos analisar esta congruência definiremos uma métrica espacial

$$h_{ab} = g_{ab} + \xi_a \xi_b,$$

desta forma $h^a_b = g^{ac} h_{cb}$ é o operador de projeção sobre o subespaço do espaço tangente ortogonal a ξ^a .

Com base nestes tensores podemos definir expansão θ , cisalhamento σ_{ab} e "twist" ω_{ab} de uma congruência

$$\theta = B^{ab} h_{ab},$$

$$\sigma_{ab} = \frac{1}{2} (B_{ab} + B_{ba}) - \frac{1}{3} \theta h_{ab},$$

$$\omega_{ab} = \frac{1}{2} (B_{ab} - B_{ba}),$$

desta forma podemos decompor o tensor B_{ab} como segue

$$B_{ab} = \frac{1}{3} \theta h_{ab} + \sigma_{ab} + \omega_{ab}.$$

A partir da equação de derivação geodésica podemos deduzir equações que determinam o comportamento de θ , σ_{ab} e ω_{ab} . Entre estas equações a que nos interessa é a equação de Raychaudhuri

$$\xi^c \nabla_c \theta = \frac{d\theta}{d\tau} = -\frac{1}{3} \theta^2 - \sigma_{ab} \sigma^{ab} + \omega_{ab} \omega^{ab} - R_{ab} \xi^a \xi^b,$$

observe que

$$R_{ab} \xi^a \xi^b = 8\pi \left(T_{ab} - \frac{1}{2} T g_{ab} \right) \xi^a \xi^b = 8\pi \left(T_{ab} \xi^a \xi^b + \frac{1}{2} T \right).$$

Esta expressão pode ser considerada não negativa, pois podemos impor duas condições sobre a distribuição de massa no espaço. A condição fraca de energia

$$T_{ab}\xi^a\xi^b \geq 0$$

e a condição forte de energia

$$T_{ab}\xi^a\xi^b \geq -\frac{1}{2}T.$$

Assim, temos o seguinte resultado:

Lema 4 *Seja ξ^a o campo tangente de uma congruência de geodésicas tipo tempo, hipersuperfície ortogonal. Suponha $R_{ab}\xi^a\xi^b \geq 0$, se θ admite um valor negativo θ_0 em um ponto de uma geodésica da congruência então θ vai para $-\infty$ ao longo desta geodésica em um tempo próprio $\tau \leq 3/|\theta_0|$.*

Pelo fato de que a congruência ser hipersuperfície ortogonal $\omega_{ab} = 0$. A condição $R_{ab}\xi^a\xi^b \geq 0$ será sempre válida se a equação de Einstein for satisfeita e a matéria obedecer a condição forte de energia. O valor $-\sigma_{ab}\sigma^{ab}$ é sempre menor ou igual a zero. Então a partir da equação de Raychaudhuri temos

$$\frac{d\theta}{d\tau} + \frac{1}{3}\theta^2 \leq 0,$$

cuja solução é

$$\theta^{-1} \geq \theta_0^{-1} + \frac{\tau}{3},$$

como θ^{-1} vai a zero com tempo próprio $\tau \leq 3/|\theta_0|$, θ vai ao infinito simultaneamente.

Para trabalhar com congruências de geodésicas nulas esbarramos no problema de que não existe uma maneira natural de normalizar os vetores tangentes, k^a , destas geodésicas. Para podermos isolar o deslocamento paralelo entre duas geodésicas e o deslocamento ao longo de uma mesma, precisamos criar o espaço \hat{V}_p que contém apenas os vetores de interesse. Para construir este espaço iniciamos tomando o subespaço, \tilde{V}_p , dos vetores ortogonais a k^a em V_p . Agora definimos uma relação de equivalência, onde dois vetores, $x^a, y^a \in \tilde{V}_p$ são considerados equivalentes se existe $c \in \mathbb{R}$ tal que $x^a - y^a = ck^a$. Então \hat{V}_p é o espaço quociente de \tilde{V}_p com relação a equivalência definida acima.

Para definirmos expansão, cisalhamento e "twist" de uma congruência de geodésicas nulas trabalharemos sobre o espaço \hat{V}_p ao invés de V_p , embora não exista uma maneira natural de relacionar um vetor arbitrário $t^a \in V_p$ com $\hat{t}^a \in \hat{V}_p$ é possível fazê-lo com os vetores tais que $t^a k_a = 0$, tomando suas classes de equivalência. O mesmo sendo válido para tensores de qualquer ranque.

Após estes algebrismos podemos definir, como no caso de geodésicas tipo-tempo, expansão θ , cisalhamento $\hat{\sigma}_{ab}$ e "twist" $\hat{\omega}_{ab}$ da congruência de geodésicas nulas. Assim temos \hat{h}_{ab} e \hat{B}_{ab} como as classes de equivalência de h_{ab} e B_{ab} , respectivamente. Definindo

$$\theta = \hat{h}^{ab}\hat{B}_{ab},$$

$$\hat{\sigma}_{ab} = \frac{1}{2}(\hat{B}_{ab} + \hat{B}_{ba}) - \frac{1}{2}\theta\hat{h}_{ab},$$

$$\hat{\omega}_{ab} = \frac{1}{2}(\hat{B}_{ab} - \hat{B}_{ba}),$$

temos

$$\hat{B}_{ab} = \frac{1}{2}\theta\hat{h}_{ab} + \hat{\sigma}_{ab} + \hat{\omega}_{ab},$$

pelo mesmo processo usado para obter a equação de Raychauduri obtemos a seguinte equação, que é muito similar a anterior

$$\frac{d\theta}{d\lambda} = -\frac{1}{2}\theta^2 - \hat{\sigma}_{ab}\hat{\sigma}^{ab} + \hat{\omega}_{ab}\hat{\omega}^{ab} - R_{ab}k^ak^b,$$

que nos leva ao seguinte lema, parecido com o anterior:

Lema 5 *Seja k^a o campo tangente de uma congruência de geodésicas nulas, hipersuperfície ortogonal. Suponha que $R_{ab}k^ak^b \geq 0$. Se a expansão θ assume um valor negativo θ_0 em um ponto de uma geodésica da congruência, então θ vai a $-\infty$ ao longo desta geodésica com comprimento do parâmetro afim $\lambda \leq 2/|\theta_0|$.*

A.6 Pontos Conjugados

Considere uma espaço-tempo (M, g_{ab}) onde está definida uma conexão ∇_a e γ é uma geodésica com tangente v^a . A solução da equação de derivação geodésica μ^a

$$v^a\nabla_a(v^b\nabla_b\mu^c) = -R^c{}_{abd}\mu^bv^av^d$$

é chamada de campo de Jacobi. Dois pontos $p, q \in \gamma$ são considerados conjugados se existe um campo de Jacobi não nulo, que se anule em p e q . Isso significa que uma geodésica infinitesimalmente próxima de γ a interceptaria em p e q , embora não seja necessário que exista uma geodésica com esta propriedade, pois a definição exige apenas a existência do campo de Jacobi.

Com o conceito de congruências geodésicas podemos exibir critérios para a existência de pontos conjugados.

Lema 6 *Seja (M, g_{ab}) um espaço-tempo satisfazendo $R_{ab}\xi^a\xi^b \geq 0$ para todo vetor tipo-tempo ξ^a . Considere uma geodésica tipo-tempo γ e $p \in \gamma$. Suponha que a expansão, θ , da congruência das geodésicas tipo-tempo, direcionadas ao futuro, que emanam de p , assumam um valor negativo θ_0 em $r \in \gamma$. Então com tempo próprio $\tau \leq 3/|\theta_0|$ a partir de r ao longo de γ existe um ponto q conjugado com p , desde que γ extenda-se o suficiente.*

Este lema pode ser bastante generalizado com base na condição tipo-tempo genérica, isto é, dado (M, g_{ab}) cada geodésica tipo-tempo em M possui ao menos um ponto onde $R_{abcd}\xi^a\xi^d \neq 0$. Então temos o seguinte lema reformulado

Lema 7 *Seja (M, g_{ab}) um espaço-tempo satisfazendo a condição tipo-tempo genérica e $R_{ab}\xi^a\xi^b \geq 0$ para qualquer vetor tipo-tempo ξ^a . Então qualquer geodésica completa possui um par de pontos conjugados.*

Os pontos conjugados são importantes pois eles caracterizam as situações em que uma geodésica deixa de ser uma curva extrema, ou seja, uma geodésica tipo-tempo deixa de ser uma máximo local do tempo próprio, ou uma geodésica nula deixa de estar na fronteira do futuro de um evento, como mostra o seguinte teorema:

Teorema 9 *Seja γ uma curva analítica tipo-tempo conectando $p, q \in M$. Então uma condição necessária e suficiente para que γ maximize localmente o tempo próprio entre p e q , sobre a variação suave de um parâmetro, é que γ seja uma geodésica sem pontos conjugados entre os pontos p e q .*

Podemos estender o conceito de pontos conjugados para ponto conjugado a uma hipersuperfície tipo-espaço analítica, ou ao menos C^2 , Σ , onde hipersuperfície significa uma subvariedade tridimensional imersa em M .

Para isso definimos a curvatura extríntrica K_{ab} de Σ . Considere o campo tangente unitário, ξ^a , da congruência das geodésicas tipo-tempo ortogonais a Σ . Então podemos definir K_{ab} , sobre Σ , como

$$K_{ab} = \nabla_a \xi^b = B_{ba}.$$

O traço da curvatura extríntrica será denotado por K e definido como $K = K^a_a = h^{ab}K_{ab}$. Assim temos que $K = \theta$ onde θ é a expansão da congruência de geodésicas ortogonais a Σ . Um ponto p de uma geodésica γ , da congruência de geodésicas ortogonais a Σ , é considerado conjugado com Σ ao longo de γ se existe um campo de Jacobi que não é zero em Σ e se anula em p . Para caracterizar a existência de um ponto conjugado a Σ temos o seguinte teorema:

Teorema 10 *Seja (M, g_{ab}) um espaço-tempo satisfazendo $R_{ab}\xi^a\xi^b \geq 0$ para todo vetor tipo-tempo ξ^a . Considere uma hipersuperfície tipo espaço Σ com $K = \theta < 0$ em algum ponto $q \in \Sigma$. Então existe, com tempo próprio $\tau \leq 3/|K|$, um ponto p conjugado com Σ ao longo da geodésica γ ortogonal a Σ passando por q , assumindo que γ se estenda o suficiente.*

De forma similar ao Teorema 9 temos

Teorema 11 *Seja γ uma curva tipo-tempo analítica conectando um ponto q em uma hipersuperfície tipo-espaço analítica Σ com um ponto $p \in M$. Então uma condição necessária e suficiente para que γ maximize o tempo próprio entre p e Σ sobre a variação suave de um parâmetro é que γ seja uma geodésica ortogonal a Σ sem pontos conjugados entre Σ e p .*

Para definirmos pontos conjugados de uma geodésica nula, μ , devemos observar que a equação de derivação geodésica

$$k^c \nabla_c [k^b \nabla_b (k^a \nu_a)] = 0,$$

onde k^a é o vetor tangente a μ , implica que o campo de Jacobi, ν^a , não pode se anular em $p, q \in \mu$ a menos que $k^a \nu_a = 0$ ao longo de toda curva μ . Por outro lado, se ν^a é um campo de Jacobi então $\nu^a + (a + b\lambda)k^a$ também é, onde a e b são constantes. Portanto, os pontos p e q serão conjugados se existir uma campo de Jacobi que seja múltiplo de k^a em p e q . Ou seja, ao longo de uma geodésica nula μ os pontos $p, q \in \mu$ serão conjugados se, e apenas se, um campo $\hat{\nu}^a \in \hat{V}$ satisfizer a equação de derivação geodésica e se anule em p e q . Com estas definições temos o seguinte resultado

Lema 8 *Seja (M, g_{ab}) um espaço-tempo satisfazendo $R_{ab}k^ak^b \geq 0$ para todo vetor nulo k^a . Considere uma geodésica nula μ e $p \in \mu$. Suponha que a expansão θ da congruência composta pelas geodésicas nulas que emanam de p assumam um valor negativo θ_0 em $r \in \mu$. Então com comprimento do parâmetro afim $\lambda \leq 2/|\theta_0|$ a partir de r , existe um ponto q conjugado com p ao longo de μ , desde que μ se estenda o suficiente.*

Análogo ao caso tipo-tempo podemos definir a *condição nula genérica* como sendo a imposição de que em toda geodésica nula μ exista ao menos um ponto $r \in \mu$ onde $R_{ab}k^ak^b > 0$ ou $k_{[e}C_{a]bc[d}k^bk^c \neq 0$ onde C_{abcd} é o tensor de Weyl. Então temos

Lema 9 *Suponha que (M, g_{ab}) satisfaça a condição nula genérica e que $R_{ab}k^ak^b \geq 0$ para todo vetor nulo k^a . Então qualquer geodésica nula completa possui um par de pontos conjugados.*

Em comparação aos teoremas que determinam quando uma curva tipo-tempo é um máximo local do tempo próprio temos o seguinte teorema

Teorema 12 *Seja μ uma curva causal analítica e $p, q \in \mu$. Então não existe uma família de curvas causais λ_α conectando p e q com $\lambda_0 = \mu$ e λ_α tipo-tempo para todo $\alpha > 0$, isto é μ não pode ser suavemente deformado em uma curva tipo-tempo, se, e somente se, μ é uma geodésica nula sem pontos conjugados com p ao longo de μ entre p e q .*

Para geodésicas nulas também é possível definir ponto conjugado a uma superfície S , tipo-espaço e bidimensional. Porém, agora existem dois vetores nulos direcionados ao futuro ortogonais a S em cada ponto $q \in S$, se S é orientável então podemos fazer uma escolha contínua em S de forma a criar duas famílias de geodésicas nulas que serão referidas como famílias “ingoing” e “outgoing”. Se S não for orientável esta escolha pode ser feita localmente. Assim podemos enunciar os próximos teoremas

Teorema 13 *Seja (M, g_{ab}) um espaço-tempo satisfazendo $R_{ab}k^ak^b \geq 0$ para todo vetor nulo k^a . Considere S uma subvariedade tipo-espaço analítica e bidimensional tal que a expansão θ da congruência composta pelas geodésicas nulas, “outgoing” ou “ingoing”, assumam um valor negativo θ_0 em $q \in S$. Então com parâmetro afim $\lambda \leq 2/|\theta_0|$, existe um ponto p conjugado com S ao longo da geodésica nula, respectivamente “outgoing” ou “ingoing”, μ que passa por q .*

Teorema 14 *Seja S uma subvariedade tipo-espaço analítica bidimensional, e μ uma curva causal analítica de S até p . Então uma condição necessária e suficiente para que μ não possa ser suavemente deformada em uma curva tipo-tempo conectando S com p é que μ seja uma geodésica nula sem pontos conjugados a S entre S e p .*

Como uma consequência deste teorema e dos resultados sobre espaços globalmente hiperbólico temos

Teorema 15 *Seja (M, g_{ab}) um espaço-tempo globalmente hiperbólico e considere K uma subvariedade espacial bidimensional orientável e compacta de M . Então qualquer $p \in \dot{I}(K)$ está em uma geodésica nula direcionada ao futuro que sai ortogonalmente de K e não possui ponto conjugado com K entre K e p*

A.7 Curvas de Comprimento Máximo

Temos alguns teoremas que determinam quando uma curva maximiza localmente o tempo próprio entre dois pontos $p, q \in M$

$$\tau[\lambda] = \int_p^q \sqrt{-t^a t_a} dl,$$

onde t^a é o vetor tangente da curva causal analítica, ou pelo menos C^1 , λ . Queremos agora determinar quais as condições de existência para tais curvas. Para isso deveremos estender a função τ para todas as curvas em $C(p, q)$, para podermos usar os resultados de compacidade deste conjunto.

Inicialmente tomemos o conjunto $\tilde{C}(p, q)$ das curvas tipo-tempo analíticas, com topologia induzida por $C(p, q)$. Em $\tilde{C}(p, q)$ τ pode ser calculado pela fórmula acima, e como este conjunto é denso em $C(p, q)$, exceto por algumas geodésicas nulas, poderíamos estender a função τ tomando uma seqüência $\{\lambda_n\}$ em $\tilde{C}(p, q)$ que aproxima continuamente $\mu \in C(p, q)$ e atribuindo

$$\tau[\mu] = \lim_{n \rightarrow \infty} \tau[\lambda_n].$$

Porém, a função τ não é contínua em $\tilde{C}(p, q)$, porém segundo o próximo lema, ela é semi-contínua superior

Lema 10 *Considere (M, g_{ab}) um espaço-tempo fortemente causal e $p, q \in M$ com $q \in I^+(p)$. Então τ é semi-contínua superior em $\tilde{C}(p, q)$.*

Assim podemos estender τ como função semi-contínua superior de $\tilde{C}(p, q)$ a $C(p, q)$ tomando para $\mu \in C(p, q)$ uma vizinhança aberta $O \subset C(p, q)$ e definindo

$$T(\mu) = \sup\{\tau[\lambda] \mid \lambda \in O, \lambda \in \tilde{C}(p, q)\}$$

e então definindo

$$\tau[\mu] = \inf\{T[O] \mid O \text{ é uma vizinhança aberta de } \mu\}.$$

Os teoremas anteriores que caracterizavam a curva que maximiza τ referiam se apenas a curvas analíticas, ou seja, curvas em $\tilde{C}(p, q)$ e não em $C(p, q)$, mas eles podem ser generalizados e assim temos

Teorema 16 *Seja (M, g_{ab}) um espaço-tempo fortemente causal e $p, q \in M$ com $q \in J^+(p)$ e considere a função τ definida em $C(p, q)$. Uma condição necessária que τ assuma o valor máximo em $\gamma \in C(p, q)$ é que γ seja uma geodésica sem pontos conjugados entre p e q .*

Teorema 17 *Seja (M, g_{ab}) um espaço-tempo fortemente causal e $p \in M$ com Σ uma hipersuperfície tipo espaço, analítica e atemporal. Considere a função τ definida em $C(p, q)$. Uma condição necessária para que τ assuma o valor máximo em $\gamma \in C(p, q)$ é que γ seja uma geodésica ortogonal a Σ sem pontos conjugados com Σ entre Σ e p .*

Uma característica importante dos espaços globalmente hiperbólicos é que $C(p, q)$ é compacto, além disso sabemos que τ é semi contínua superior então podemos aplicar os teoremas de existência do máximo. O que implica diretamente nos seguintes teoremas.

Teorema 18 *Seja (M, g_{ab}) um espaço-tempo globalmente hiperbólico, com $p, q \in M$ e $q \in J^+(p)$. Então existe uma curva $\gamma \in C(p, q)$ para a qual τ assume seu valor máximo em $C(p, q)$.*

Teorema 19 *Seja (M, g_{ab}) um espaço-tempo globalmente hiperbólico, com $p \in M$ e Σ uma superfície de Cauchy. Então existe uma curva $\gamma \in C(\Sigma, p)$ para a qual τ assume seu valor máximo em $C(\Sigma, p)$.*

A.8 Teoremas Sobre Singularidades

Apresentamos aqui quatro teoremas sobre singularidades baseados na idéia de incompletude geodésica. Os dois primeiros demonstram a existência de uma singularidade na origem do universo, ou seja, a existência do “big-bang”.

Teorema 20 *Seja (M, g_{ab}) um espaço-tempo globalmente hiperbólico com $R_{ab}\xi^a\xi^b \geq 0$ para todo vetor tipo-tempo, ξ^a . Suponha que exista uma superfície de Cauchy, Σ , analítica (ou ao menos C^2) e orientada espacialmente. Para a qual o traço da curvatura extrínseca, para a congruência das geodésicas tipo-tempo, normais e orientadas para o passado, satisfaça $K \leq C < 0$, em Σ , onde C é uma constante. Então nenhuma curva tipo-tempo orientada ao passado saindo de Σ pode ter comprimento maior que $3/|C|$. Na verdade, todas as geodésicas tipo-tempo e direcionadas ao passado são incompletas.*

Demonstração: Suponha que exista uma curva tipo-tempo direcionada ao passado, λ , saindo de Σ com comprimento maior que $3/|C|$. Tome um ponto p em λ além de $3/|C|$ no passado de Σ . Pelo Teorema 19 existe uma curva de comprimento máximo γ entre p e Σ , que claramente também tem comprimento maior que $3/|C|$. Mas pelo Teorema 17 γ precisa ser uma geodésica sem ponto conjugado entre Σ e p . Mas isso contradiz o Teorema 10 que determina que deve existir um ponto conjugado entre Σ e p . Portanto a curva λ não pode existir.

Pode se considerar que na verdade este teorema ao invés de provar a existência de uma singularidade inicial provaria a inexistência da superfície de Cauchy em um espaço com estas características. Mas a hipótese da existência desta superfície pode ser enfraquecida mantendo ainda a existência de uma singularidade inicial. Como mostra o próximo teorema.

Teorema 21 *Seja (M, g_{ab}) um espaço-tempo fortemente causal com $R_{ab}\xi^a\xi^b \geq 0$ para todo vetor tipo-tempo. Suponha que exista uma hipersuperfície analítica tipo espaço, compacta, atemporal e sem borda, S . Tal que para a congruência de geodésicas tipo-tempo, normais e direcionadas para o passado saindo de S nós temos $K \leq C < 0$. Então ao menos uma geodésica inextensível tipo-tempo e direcionada para o passado saindo de S tem comprimento no máximo igual a $3/|C|$.*

Há outro contexto em que as singularidades desempenham um papel importante, chamado colapso gravitacional. Para apresentarmos um teorema neste contexto temos anteriormente que definir “trapped surface”, que consiste em uma subvariedade bidimensional, tipo-espaço, compacta e analítica, T , onde a expansão, θ , de ambos os conjuntos de geodésicas nulas direcionadas ao futuro, ortogonais a T , é sempre negativa. Por exemplo, na solução de Schwarzschild todas as esferas interiores ao horizonte de eventos, são “trapped surface”. Assim temos o seguinte teorema

Teorema 22 *Seja (M, g_{ab}) um espaço-tempo conectado e globalmente hiperbólico com uma superfície de Cauchy não compacta, que satisfaz $R_{ab}k^ak^b \geq 0$ para todo vetor nulo, k^a . Suponha ainda que M possui uma "trapped surface" T . Denote o valor máximo de θ para ambos os conjuntos de geodésicas nulas ortogonais em T por $\theta_0 < 0$. Então pelo menos uma geodésica nula, ortogonal, orientada para o futuro saindo de T possui comprimento afim menor ou igual a $2/|\theta_0|$.*

Os teoremas anteriores, sobre singularidades, estavam relacionados a situações específicas. O próximo teorema demonstra a existência de singularidades em situações muito mais gerais. O que é um forte argumento para a hipótese de que as singularidades seja características comuns aos espaço-tempos.

Teorema 23 *Suponha um espaço-tempo, (M, g_{ab}) que satisfaz estas quatro condições:*

1. $R_{ab}\xi^a\xi^b \geq 0$ para todo vetor tipo-tempo,
2. As condições genéricas tipo-tempo e nulas são satisfeitas,
3. Não existem curvas tipo-tempo fechadas e
4. Ao menos uma das três propriedades seguintes vale
 - (M, g_{ab}) possui um conjunto atemporal, compacto sem borda, isto é, (M, g_{ab}) é um universo fechado,
 - (M, g_{ab}) possui uma "trapped surface", ou
 - existe um ponto $p \in M$ tal que a expansão, das geodésicas nulas direcionadas para o futuro, ou passado, emanando de p , torne se negativa ao longo de cada geodésica desta congruência.

Então (M, g_{ab}) precisa ter ao menos uma geodésica tipo-tempo ou nula que seja incompleta.

B.1 Cilíndricas

No início quando apresentamos as equações de Weyl, nos deparamos com o Laplaciano em coordenadas cilíndricas com simetria axial (2.7). Apresentamos aqui a definição destas coordenadas em termos das coordenadas Cartesianas

$$x = r \sin \varphi, \quad y = r \cos \varphi, \quad z = z \quad (\text{B.1})$$

e com base nesta definição impomos as restrições $0 \leq r < \infty$, $0 \leq \varphi < 2\pi$ e $-\infty < z < \infty$.

O Laplaciano é definido em coordenadas Cartesianas como

$$\nabla^2 \phi = \phi_{xx} + \phi_{yy} + \phi_{zz} \quad (\text{B.2})$$

e em coordenadas cilíndricas assume a forma

$$\nabla^2 \phi = \frac{1}{r} [r \phi_r]_r + \frac{1}{r^2} \phi_{\varphi\varphi} + \phi_{zz}. \quad (\text{B.3})$$

Como estamos impondo a restrição de simetria axial e definindo as coordenadas de modo que o eixo de simetria coincida com o eixo $r = 0$, qualquer solução de interesse não dependerá da coordenada φ . Podemos então desconsiderar os termos de (B.3) que envolvam derivadas com relação a esta coordenada, temos assim exatamente a expressão (2.7). Pelo mesmo motivo, esta coordenada será omitida no restante deste apêndice.

As coordenadas cilíndricas não são as únicas adequadas a sistemas com simetria axial. Apresentamos aqui uma lista de coordenadas que podem ser úteis em determinadas ocasiões. Uma lista mais completa pode ser encontrada em [57].

Apresentaremos também as soluções por separação de variáveis para cada um dos sistemas. Em coordenadas cilíndricas considerando $\phi(r, z) = R(r)Z(z)$ temos

$$rR'' + R' + \alpha^2 rR = 0, \quad (\text{B.4})$$

$$Z' - \alpha^2 Z = 0 \quad (\text{B.5})$$

e a solução geral para este sistema é

$$\alpha = 0 \quad R = A + B \ln r, \quad (\text{B.6})$$

$$Z = C + Dz;$$

$$\alpha \neq 0 \quad R = A \mathcal{J}_0(\alpha r) + B \mathcal{Y}_0(\alpha r), \quad (\text{B.7})$$

$$Z = Ce^{\alpha z} + De^{-\alpha z}.$$

B.2 Esféricas

Definimos as coordenadas esféricas, em termos das coordenadas cilíndricas, pela transformação

$$r = R \sin \theta, \quad z = R \cos \theta \quad (\text{B.8})$$

as novas restrições são $0 \leq R < \infty$ e $0 \leq \theta \leq \pi$, e a equação de Laplace fica

$$\nabla^2 \phi = \phi_{RR} + \frac{2}{R} \phi_R + \frac{1}{R^2} \phi_{\theta\theta} + \frac{\cot \theta}{R^2} \phi_\theta. \quad (\text{B.9})$$

Separando as variáveis $\phi(R, \theta) = \mathcal{R}(R)\Theta(\theta)$ obtemos as equações

$$R^2 \mathcal{R}'' + 2R \mathcal{R}' + \alpha \mathcal{R} = 0, \quad (\text{B.10})$$

$$\sin \theta \Theta'' + \cos \theta \Theta' + \alpha \sin \theta \Theta = 0 \quad (\text{B.11})$$

e a solução geral

$$\alpha = 0 \quad \mathcal{R} = A + BR^{-1}, \quad (\text{B.12})$$

$$\Theta = C + D Q_0(\cos \theta);$$

$$\alpha = \lambda(\lambda + 1) \quad \mathcal{R} = \begin{cases} AR^{-1/2} + BR^{-1/2} \ln \rho & , \lambda = -1/2, \\ AR^\lambda + BR^{-(\lambda+1)} & , \lambda \neq -1/2, \end{cases} \quad (\text{B.13})$$

$$\Theta = C \mathcal{P}_\lambda(\cos \theta) + D \mathcal{Q}_\lambda(\cos \theta).$$

B.3 Esferoidais Prolatas

Estas coordenadas, que as vezes são chamadas de elipsoidais prolatas, são definidas por

$$r = a \sinh \eta \sin \theta, \quad z = a \cosh \eta \cos \theta \quad (\text{B.14})$$

com $0 \leq \eta < \infty$ e $0 \leq \theta \leq \pi$. O Laplaciano torna-se

$$\nabla^2 \phi = a^{-2} \left(\sinh^2 \eta + \sin^2 \theta \right)_{-1} (\phi_{\eta\eta} + \coth \eta \phi_{\eta\eta} + \phi_{\theta\theta} + \cot \theta \phi_{\theta\theta}). \quad (\text{B.15})$$

Separando variáveis $\phi(\eta, \varphi, \theta) = H(\eta)\Theta(\theta)$ temos

$$H'' + \coth \eta H' - p(p+1)H = 0, \quad (\text{B.16})$$

$$\Theta'' + \cot \theta \Theta' - p(p+1)\Theta = 0 \quad (\text{B.17})$$

então a solução geral será

$$H = A \mathcal{P}_p(\cosh \eta) + B \mathcal{Q}_p(\cosh \eta), \quad (\text{B.18})$$

$$\Theta = C \mathcal{P}_p(\cos \theta) + D \mathcal{Q}_p(\cos \theta). \quad (\text{B.19})$$

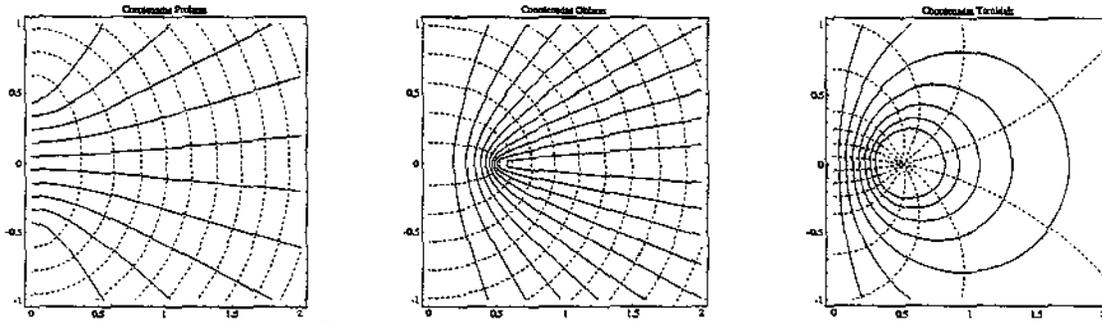


Figura B.1: Coordenadas: Prolatas, Oblatas e Toroidais.

É comum definir este sistema de coordenadas de uma forma alternativa onde tomamos

$$x = \cosh \eta, \quad y = \cos \theta \tag{B.20}$$

as restrições para estas coordenadas serão conseqüentemente dadas por $1 \leq x < \infty$ e $-1 \leq y \leq 1$. Podemos definir este sistema diretamente das coordenadas cilíndricas pela transformação

$$r^2 = a^2(x^2 - 1)(1 - y^2), \quad z = a x y. \tag{B.21}$$

Neste caso a transformação inversa é dada por

$$x = \frac{R_+ + R_-}{2a}, \quad y = \frac{R_+ - R_-}{2a}, \tag{B.22}$$

onde $R_{\pm}^2 = r^2 + (z \pm a)^2$.

B.4 Esferoidais Oblatas

Também chamado de elipsoidal oblato este sistema defini-se por

$$r = a \cosh \zeta \sin \theta, \quad z = a \sinh \zeta \cos \theta \tag{B.23}$$

com $0 \leq \zeta < \infty$ e $0 \leq \theta \leq \pi$. A equação de Laplace escreve-se como

$$\nabla^2 \phi = a^{-2} (\cosh^2 \zeta - \sin^2 \theta)^{-1} (\phi_{\zeta\zeta} + \tanh \zeta \phi_{\zeta\zeta} + \phi_{\theta\theta} + \cot \theta \phi_{\theta\theta}). \tag{B.24}$$

Separando variáveis $\phi(\zeta, \theta) = Z(\zeta)\Theta(\theta)$ temos

$$Z'' + \tanh \zeta Z' - p(p+1)Z = 0, \tag{B.25}$$

$$\Theta'' + \coth \theta \Theta' + p(p+1)\Theta = 0 \tag{B.26}$$

e solução geral

$$Z = A \mathcal{P}_p(i \sinh \zeta) + B \mathcal{Q}_p(i \sinh \zeta), \tag{B.27}$$

$$\Theta = C \mathcal{P}_p(\cos \theta) + D \mathcal{Q}_p(\cos \theta). \tag{B.28}$$

Como no caso das coordenadas esferoidais prolatas este sistema também possui uma forma alternativa, definida como

$$\xi = \sinh \zeta, \quad \eta = \cos \theta \tag{B.29}$$

com as coordenadas restritas a $-1 \leq \eta \leq 1$ e $0 \leq \xi < \infty$. Ou definindo diretamente das coordenadas cilíndricas

$$r^2 = a^2(1 + \xi^2)(1 - \eta^2), \quad z = a\xi\eta. \quad (\text{B.30})$$

Neste caso a transformação inversa é dada pelas expressões

$$\xi = \frac{\sqrt{2}z}{\sqrt{A-K}}, \quad \eta = \frac{\sqrt{A-K}}{\sqrt{2}a}, \quad (\text{B.31})$$

onde definimos $K = r^2 + z^2 - a^2$ e $A = \sqrt{K^2 + 4a^2z^2}$.

B.5 Toroidais

Podemos definir as coordenadas toroidais pela transformação real

$$r = \frac{a \sinh \eta}{\cosh \eta - \cos \xi}, \quad z = \frac{a \sin \xi}{\cosh \eta - \cos \xi} \quad (\text{B.32})$$

ou, equivalentemente, pela transformação complexa

$$z + ir = a \cot \frac{\xi + i\eta}{2}. \quad (\text{B.33})$$

Em ambos os casos teremos as restrições $0 \leq \eta < \infty$ e $-\pi < \xi \leq \pi$. A transformação inversa é dada por

$$\tan \xi = \frac{2za}{r^2 + z^2 - a^2}, \quad e^\eta = \frac{\sqrt{(r^2 + z^2 - a^2)^2 + 4z^2a^2}}{(r-a)^2 + z^2} \quad (\text{B.34})$$

e o Laplaciano escreve-se como

$$\nabla^2 \phi = \frac{(\cosh \eta - \cos \xi)^2}{a^2 \sinh \eta} \left[\left(\frac{\sinh \eta \phi_\eta}{\cosh \eta - \cos \xi} \right)_\eta + \sinh \eta \left(\frac{\phi_\xi}{\cosh \eta - \cos \xi} \right)_\xi \right]. \quad (\text{B.35})$$

Para podermos separar as variáveis neste sistema de coordenadas precisamos de uma função auxiliar, como segue $\phi(\eta, \xi) = \sqrt{\cosh \eta - \cos \xi} H(\eta) X(\xi)$, então teremos

$$H'' + \coth \eta H' + \left(\frac{1}{4} - \alpha^2 \right) H = 0, \quad (\text{B.36})$$

$$X'' + \alpha^2 X = 0 \quad (\text{B.37})$$

com soluções dadas por

$$\alpha = 0 \quad H = A \mathcal{P}_{-1/2}(\cosh \eta) + B \mathcal{Q}_{-1/2}(\cosh \eta), \quad (\text{B.38})$$

$$X = C + D\xi;$$

$$\alpha \neq 0 \quad H = A \mathcal{P}_{\alpha-1/2}(\cosh \eta) + B \mathcal{Q}_{\alpha-1/2}(\cosh \eta), \quad (\text{B.39})$$

$$X = C \sin \alpha \xi + D \cos \alpha \xi.$$

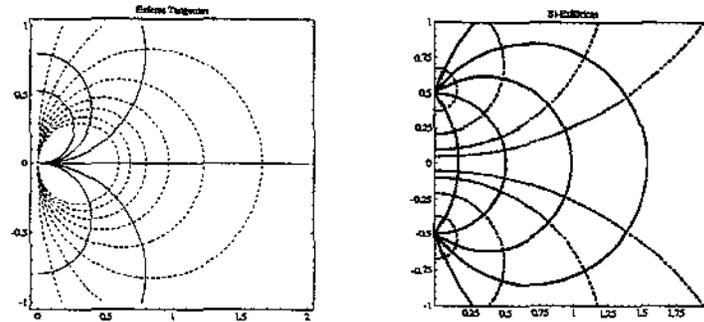


Figura B.2: Coordenadas: Esferas Tangentes e Bi-Esféricas.

B.6 Esferas Tangentes

Sistema de coordenadas que se define pela transformação

$$r = \frac{\chi}{\chi^2 + \phi^2}, \quad z = \frac{\psi}{\chi^2 + \phi^2} \quad (\text{B.40})$$

e a transformação inversa é dada, de forma bastante simétrica, pelas expressões

$$\chi = \frac{r}{r^2 + z^2}, \quad \psi = \frac{z}{r^2 + z^2}.$$

Neste sistema o Laplaciano escreve-se como

$$\nabla^2 \phi = \frac{\partial}{\partial \chi} \left(\frac{\chi \phi_\chi}{\chi^2 + \psi^2} \right) + \chi \frac{\partial}{\partial \psi} \left(\frac{\phi_\psi}{\chi^2 + \psi^2} \right). \quad (\text{B.41})$$

Como no caso das coordenadas toroidais a separação de variáveis se da segundo a fórmula $\phi(\chi, \psi) = \sqrt{\chi^2 + \psi^2} X(\chi)P(\psi)$, então teremos

$$X'' + \frac{X'}{\chi} + q^2 X = 0, \quad (\text{B.42})$$

$$P'' - q^2 P = 0 \quad (\text{B.43})$$

com soluções dadas por

$$q = 0 \quad X = A + B \log \chi, \quad (\text{B.44})$$

$$P = C + D\psi;$$

$$q \neq 0 \quad X = A \mathcal{J}_0(q\chi) + B \mathcal{Y}_0(q\chi), \quad (\text{B.45})$$

$$P = C e^{q\psi} + D e^{-q\psi}.$$

B.7 Bi-Esféricas

Podemos definir este sistema de coordenadas pelas relações

$$r = \frac{a \sin \theta}{\cosh \eta - \cos \theta}, \quad z = \frac{a \sinh \eta}{\cosh \eta - \cos \theta}. \quad (\text{B.46})$$

Nestas coordenadas o Laplaciano transforma-se em

$$\nabla^2 \phi = \sin \theta \frac{\partial}{\partial \eta} \left(\frac{\phi_\eta}{\cosh \eta - \cos \theta} \right) + \frac{\partial}{\partial \theta} \left(\frac{\sin \theta \phi_\theta}{\cosh \eta - \cos \theta} \right) \quad (\text{B.47})$$

e as variáveis separam-se pela relação

$$\phi(\eta, \theta) = \sqrt{\cosh \eta - \cos \theta} H(\eta) \Theta(\theta)$$

a equação de Laplace simplifica-se nas equações

$$H'' - \left(\frac{1}{4} + s(s+1) \right) H = 0, \quad (\text{B.48})$$

$$\Theta'' + \cot \theta \Theta' + s(s+1)\Theta = 0 \quad (\text{B.49})$$

com soluções dadas por

$$H = A e^{\eta(s+\frac{1}{2})} B e^{-\eta(s+\frac{1}{2})}, \quad (\text{B.50})$$

$$\Theta = C \mathcal{P}_s(\cos \theta) + D \mathcal{Q}_s(\cos \theta). \quad (\text{B.51})$$

Apresentamos neste apêndice alguns resultados sobre as soluções multipolares descritas na Seção 4.11, que por serem muito extensos escolhemos não incluí-los no corpo da dissertação.

C.1 Multipólos em Coordenadas Cilíndricas

Abaixo seguem as fórmulas para as soluções correspondentes aos primeiros termos da expansão em momentos de Weyl. O caso $a_0 \neq 0$ e $a_i = 0$ para $i > 0$ é a solução de Curzon descrita na seção 4.3. Quando $a_{0,1} \neq 0$ temos

$$\phi_1 = -\frac{a_0}{R} - \frac{a_1 z}{R^3},$$

$$\nu_1 = \frac{9a_1^2 r^4}{4R^8} - \frac{2a_1 r^2}{R^6} (a_1 + a_0 z) - \frac{a_0^2 r^2}{2R^4}.$$

Enquanto que quando $a_i \neq 0$ para $i = 0, 1, 2$ temos

$$\phi_2 = -\frac{a_0}{R} - \frac{a_1 z}{R^3} + \frac{a_2}{2R^5} (r^2 - 2z^2),$$

$$\nu_2 = -\frac{75a_2^2 r^6}{8R^{12}} + \frac{9a_2 r^4}{2R^{10}} (3a_2 + 2a_1 z) + \frac{3r^2}{4R^8} (3a_1^2 r^2 - 6a_2^2 + 5a_0 a_2 r^2 - 8a_1 a_2 z)$$

$$- \frac{r^2}{R^6} (2a_1^2 + 3a_0 a_2 + 2a_0 a_1 z) - \frac{a_0^2 r^2}{2R^4}.$$

Quando $a_i \neq 0$ para $i = 0, 1, 2, 3$ temos

$$\phi_3 = -\frac{a_0}{R} - \frac{a_1 z}{R^3} + \frac{a_2}{2R^5} (r^2 - 2z^2) + \frac{a_3}{2R^7} (3r^2 z - 2z^3),$$

$$\nu_3 = \frac{1225 a_3^2 r^8}{32 R^{16}} - \frac{75 a_3 r^6}{2 R^{14}} (2a_3 + a_2 z) + \frac{5 r^4}{8 R^{12}} (72 a_3^2 - 15 a_2^2 r^2 - 28 a_1 a_3 r^2 + 72 a_2 a_3 z)$$

$$- \frac{r^2}{2 R^{10}} (16 a_3^2 - 27 a_2^2 r^2 - 50 a_1 a_3 r^2 + 24 a_2 a_3 z - 18 a_1 a_2 r^2 z - 14 a_0 a_3 r^2 z)$$

$$- \frac{r^2}{4 R^8} (18 a_2^2 + 32 a_1 a_3 - 9 a_1^2 r^2 - 15 a_0 a_2 r^2 + 24 a_1 a_2 z + 16 a_0 a_3 z)$$

$$- \frac{r^2}{R^6} (2 a_1^2 + 3 a_0 a_2 + 2 a_0 a_1 z) - \frac{a_0^2 r^2}{2 R^4}.$$

C.2 Multipólos em Coordenadas Prolatas

Quando empregamos as coordenadas esferoidais prolatas (B.21) o elemento de linha das soluções de Weyl (2.6) é reescrito como

$$ds^2 = e^{2\phi} dt^2 - m^2 e^{2(\nu-\phi)} (x^2 - y^2) \left(\frac{dx^2}{x^2 - 1} + \frac{dy^2}{1 - y^2} \right) - m^2 (x^2 - 1)(1 - y^2) e^{-2\phi} d\varphi^2 \quad (\text{C.1})$$

e suas equações (2.7)–(2.8) como

$$\left[\phi_x (x^2 - 1) \right]_x + \left[\phi_y (1 - y^2) \right]_y = 0, \quad (\text{C.2})$$

$$\nu_x = \frac{1 - y^2}{x^2 + y^2} \left[x(x^2 - 1)\phi_x^2 - x(1 - y^2)\phi_y^2 - 2y(x^2 - 1)\phi_x\phi_y \right],$$

$$\nu_y = \frac{x^2 - 1}{x^2 + y^2} \left[y(x^2 - 1)\phi_x^2 - y(1 - y^2)\phi_y^2 + 2x(1 - y^2)\phi_x\phi_y \right].$$

Neste sistema de coordenadas as equações de Weyl e suas soluções obedecem aos seguintes lemas [60]

Lema 11 Se ϕ e todas as suas derivadas forem livres de singularidade em y e $1 - y^2$, isto é $\phi, \phi_x, \phi_y, \phi_{xx}, \phi_{xy}, \phi_{yy} \dots \neq 0$ para todos os valores de y e $1 - y^2$ e ν é assintoticamente plana $\lim_{x \rightarrow \infty} \nu(x, y) = 0$.

Então no eixo de simetria $\nu = 0$, ou seja, $\nu(x, \pm 1) = 0$.

Lema 12 Se ϕ é assintoticamente plana $\lim_{x \rightarrow \infty} \phi(x, y) = 0$.

Então a solução assintoticamente plana das equações de Einstein em coordenadas esferoidais prolatas pode ser calculada por

$$\nu(x, y) = (x^2 - 1) \int_{-1}^y \frac{A(x, \sigma)}{x^2 - \sigma^2} d\sigma,$$

onde $A(x, y) = y(x^2 - 1)\phi_x^2 - y(1 - y^2)\phi_y^2 + 2x(1 - y^2)\phi_x\phi_y$.

Lema 13 Quando introduzimos o sistema de coordenadas radiais

$$x = \frac{\rho}{m} - 1, \quad y = \cos \theta. \quad (\text{C.3})$$

Se as funções métricas forem assintóticas, quando tomamos o limite $m \rightarrow 0$ o elemento de linha (C.1) transforma-se na métrica de Minkowski.

A solução geral para o Laplaciano em coordenadas prolatas (C.2) é dado pela expressão integral

$$\phi(x, y) = \int_0^\infty [q_l Q_l(x) + p_l P_l(x)] [b_l Q_l(y) + c_l P_l(y)] dl$$

onde q_l, p_l, b_l e c_l dependem somente do parâmetro l , P_i e Q_i são as funções de Legendre de 1º e 2º tipos de grau i .

Impondo que ϕ e ν sejam regulares no eixo de simetria, $y = \pm 1$, e assintoticamente planas podemos simplificar a expressão acima obtendo

$$\phi(x, y) = \sum_{n=0}^{\infty} q_n Q_n(x) P_n(y) \quad (C.4)$$

onde q_n são constantes.

Se tomarmos $q_0 = 1$ e $q_n = 0$ para todo $n \geq 1$ e realizarmos a inversão $x \rightarrow -x$ obtemos a métrica de Schwarzschild, que pode ser colocada em sua forma canônica pela transformação (C.3).

Lembrando que $Q_n(-x) = (-1)^{n+1} Q_n(x)$ podemos reescrever a equação (C.4) da forma

$$\phi(x, y) = \sum_{n=0}^{\infty} (-1)^{n+1} q_n Q_n(x) P_n(y),$$

de modo que não precisaremos mais da transformação $x \rightarrow -x$, para obtermos a métrica de Schwarzschild.

Solução Geral

A partir dos resultados apresentados na seção anterior podemos apresentar a solução geral para as métricas de Weyl em coordenadas eferoidais prolatas.

$$\phi = \sum_{n=0}^{\infty} (-1)^{n+1} q_n Q_n(x) P_n(y), \quad \nu = \sum_{m,n=0}^{\infty} (-1)^{m+n} q_m q_n \Gamma^{(mn)}, \quad (C.5)$$

onde definimos

$$\begin{aligned} \Gamma^{(mn)} = & \frac{1}{2} \ln \left(\frac{x^2 - 1}{x^2 - y^2} \right) + (\epsilon_n + \epsilon_m - 2\epsilon_n \epsilon_m) \ln \left(\frac{x + y}{x - 1} \right) \\ & + (x^2 - 1) \left[x(A_{n,m} Q'_n Q_m + A_{m,n} Q'_m Q_n) - C_{n,m} Q_n Q_m + (1 - \epsilon_n) S_m + \epsilon_n S_{m+1} \right. \\ & \left. - \frac{\epsilon_n Q'_m}{m+1} [P_m - (-1)^m] \right] + (x^2 - 1)^2 \left[Q_m B_{m,n} - Q'_m A_{m,n} + \frac{1}{n+1} A_{m,n} Q'_m Q_n \right], \end{aligned}$$

$$A_{m,n} = \sum_{k=0}^{\left[\frac{n-1}{2} \right]} \left[\frac{1}{n-2k+1} + \frac{1}{n-2k} \right] A_{m,n-2k} Q'_{n-2k},$$

$$B_{m,n} = \sum_{k=0}^{\left[\frac{n-1}{2} \right]} \left[\frac{1}{n-2k} + \frac{1}{n-2k-1} \right] B_{m,n-2k-1} Q'_{n-2k-1},$$

$$A_{n,m} = \sum_{k=0}^{\left[\frac{n-1}{2} \right]} \sum_{l=0}^{\min\{n,m-2k-1\}} \frac{(2m-4k-1)K(m-2k-1, n, l)}{2(m+n) - 4(k+l) - 1} \left[P_{m+n-2(k+l)} - P_{m+n-2(k+l+1)} \right],$$

$$B_{m,n} = B_{n,m} = \sum_{j=0}^{\lfloor \frac{m-1}{2} \rfloor} \sum_{k=0}^{\lfloor \frac{n-1}{2} \rfloor} \sum_{l=0}^{\min\{m-2j-1, n-2k-1\}} (2m-4j-1) \left[\mathcal{P}_{m+n-2(j+k+l)-1} - \mathcal{P}_{m+n-2(j+k+l)-3} \right] \times$$

$$\frac{(2n-4k-1)K(m-2j-1, n-2k-1, l)}{2(m+n)-4(j+k+l)-3},$$

$$C_{m,n} = C_{n,m} = -(n+1)A_{n,m} + B_{n+1,m},$$

$$S_n = \sum_{k=0}^{\lfloor \frac{n-1}{2} \rfloor} \left(\frac{1}{n-2k} + \frac{1}{n-2k-1} \right) \left[\mathcal{P}_{n-2k-1} + (-1)^{n+1} \right] \mathcal{Q}_{n-2k-1},$$

onde $K(m, n, k)$ são os coeficientes de Clebsch-Gordan definidos como

$$K(m, n, k) = \frac{2m+2n-4k+1}{2m+2n+4k+1} \frac{\alpha_{m-k}\alpha_k\alpha_{n-k}}{\alpha_{m+n-k}}$$

com $\alpha_k = \frac{(2k-1)!!}{k!}$ e $\epsilon_k = 1$ se n for par e $\epsilon_k = 0$ se n for ímpar.

Existe também uma representação integral para esta solução [52].

Primeiras Soluções Multipolares

Listamos aqui as primeiras soluções obtidas como somas parciais da série (C.5). Quando tomamos apenas o primeiro termo geramos a solução de Schwarzschild, que é dada por

$$\phi = -\mathcal{Q}_0(x)\mathcal{P}_0(y) = \frac{1}{2} \ln \frac{x^2-1}{x^2+1}, \quad \nu = \frac{1}{2} \ln \frac{x^2-1}{x^2+y^2}.$$

Como o termo dipolar é desconsiderado, pois sempre pode ser anulado por uma transformação de coordenadas, a próxima solução é a métrica de Erez-Rozen, que corresponde a solução quadrupolar e é definida pelas funções

$$\phi = -\mathcal{Q}_0(x)\mathcal{P}_0(y) - a\mathcal{Q}_2(x)\mathcal{P}_2(y)$$

$$= \frac{1}{2} \log \frac{x-1}{x+1} - \frac{a}{4} (3y^2-1) \left(3x - \frac{3x^2-1}{2} \log \frac{x-1}{x+1} \right),$$

$$\nu = \frac{1}{64} \left[A + B \log \frac{x+1}{x-1} + C \left(\log \frac{x+1}{x-1} \right)^2 + D \log \frac{x^2-1}{x^2-y^2} \right],$$

onde definimos

$$A = 12a(y^2-1)(16+4a-3ax^2-12ay^2+27ax^2y^2),$$

$$B = 12ax(y^2-1)(8+5a-3ax^2-21ay^2+27ax^2y^2),$$

$$C = 9a^2(x^2-1)(y^2-1)(1-x^2-y^2+9x^2y^2),$$

$$D = 32(1+2a+a^2).$$

A solução octopolar é dada pela expressão

$$\begin{aligned}\phi &= -Q_0(x)P_0(y) - aQ_2(x)P_2(y) + bQ_3(x)P_3(y) \\ &= \frac{1}{2} \log \frac{x-1}{x+1} - \frac{a}{4} (3y^2 - 1) \left(3x - \frac{3x^2 - 1}{2} \log \frac{x-1}{x+1} \right) \\ &\quad - \frac{b}{4} (5y^3 - 3y) \left(\frac{15x^2 - 4}{3} + \frac{5x^3 - 3x}{2} \log \frac{x-1}{x+1} \right), \\ \nu &= \frac{1}{384} \left[A + B \log \frac{x+1}{x-1} + C \left(\log \frac{x+1}{x-1} \right)^2 + D \log \left(\frac{x^2 - 1}{x^2 - y^2} \right) \right. \\ &\quad \left. - 768b(1+a) \left(\tanh^{-1} \frac{1}{x} + \tanh^{-1} \frac{y}{x} \right) \right],\end{aligned}$$

onde

$$\begin{aligned}A &= 4(y^2 - 1) (288a + 72a^2 + 88b^2 - 54a^2x^2 - 255b^2x^2 + 225b^2x^4 - 720bxy - 1008abxy + 1080abx^3y - \\ &\quad 216a^2y^2 - 440b^2y^2 + 486a^2x^2y^2 + 3300b^2x^2y^2 - 3150b^2x^4y^2 + 2160abxy^3 - 3240abx^3y^3 + 400b^2y^4 - \\ &\quad 5025b^2x^2y^4 + 5625b^2x^4y^4),\end{aligned}$$

$$\begin{aligned}B &= 12(y+1) (32b + 32ab + 48ax + 30a^2x + 51b^2x - 18a^2x^3 - 110b^2x^3 + 75b^2x^5 + 40by + 88aby - \\ &\quad 48axy - 30a^2xy - 51b^2xy - 120bx^2y - 456abx^2y + 18a^2x^3y + 110b^2x^3y + 360abx^4y - 75b^2x^5y - \\ &\quad 40by^2 - 88aby^2 - 126a^2xy^2 - 420b^2xy^2 + 120bx^2y^2 + 456abx^2y^2 + 162a^2x^3y^2 + 1450b^2x^3y^2 - \\ &\quad 360abx^4y^2 - 1050b^2x^5y^2 - 120aby^3 + 126a^2xy^3 + 420b^2xy^3 + 1080abx^2y^3 - 162a^2x^3y^3 - 1450b^2x^3y^3 - \\ &\quad 1080abx^4y^3 + 525b^2xy^4 - 1080abx^2y^4 - 2300b^2x^3y^4 + 1080abx^4y^4 + 1875b^2x^5y^4 - 525b^2xy^5 + \\ &\quad 1050b^2x^5y^3 + 120aby^4 + 2300b^2x^3y^5 - 1875b^2x^5y^5),\end{aligned}$$

$$\begin{aligned}C &= 9(x^2 - 1)(y^2 - 1) (6a^2 + 7b^2 - 6a^2x^2 - 20b^2x^2 + 25b^2x^4 - 72abxy + 120abx^3y - 6a^2y^2 - \\ &\quad 20b^2y^2 + 54a^2x^2y^2 + 250b^2x^2y^2 - 350b^2x^4y^2 + 120abxy^3 - 360abx^3y^3 + 25b^2y^4 - 350b^2x^2y^4 \\ &\quad + 625b^2x^4y^4),\end{aligned}$$

$$D = 192(1 + 2a + a^2 + b^2).$$

E finalmente a solução 16-polar é descrita pelas funções

$$\begin{aligned}\phi &= -Q_0(x)P_0(y) - aQ_2(x)P_2(y) + bQ_3(x)P_3(y) - cQ_4(x)P_4(y) \\ &= \frac{1}{2} \log \frac{x-1}{x+1} - \frac{a}{4} (3y^2 - 1) \left(3x - \frac{3x^2 - 1}{2} \log \frac{x-1}{x+1} \right) \\ &\quad - \frac{b}{4} (5y^3 - 3y) \left(\frac{15x^2 - 4}{3} + \frac{5x^3 - 3x}{2} \log \frac{x-1}{x+1} \right) \\ &\quad - \frac{c}{64} (35y^4 - 30y^2 + 3) \left[\frac{5}{3} (11x - 21x^3) - \frac{35x^4 - 30x^2 + 3}{2} \log \frac{x-1}{x+1} \right], \\ \nu &= A + D \log \frac{x^2 - 1}{x^2 - y^2} - 2B \left(\tanh^{-1} \frac{1}{x} + \tanh^{-1} \frac{y}{x} \right) - C \log \frac{x-1}{x+1} + E \left(\log \frac{x-1}{x+1} \right)^2,\end{aligned}$$

onde definimos $B = b(1 + a + c)$, $D = \frac{1}{2}(1 + 2a + a^2 + b^2 + 2c + 2ac + c^2)$ e

$$A = \frac{y^2 - 1}{18432} \left(55296a + 13824a^2 + 16896b^2 + 76800c + 70656ac + 19200c^2 - 10368a^2x^2 - 48960b^2x^2 - 80640cx^2 - 89280acx^2 - 93465c^2x^2 + 43200b^2x^4 + 60480acx^4 + 179550c^2x^4 - 99225c^2x^6 - 138240bcxy - 193536abxy - 403200bcxy + 207360abx^3y + 1235520bcx^3y - 907200bcx^5y - 41472a^2y^2 - 84480b^2y^2 - 107520cy^2 - 168960acy^2 - 165120c^2y^2 + 93312a^2x^2y^2 + 633600b^2x^2y^2 + 403200cx^2y^2 + 1267200acx^2y^2 + 2191275c^2x^2y^2 - 604800b^2x^4y^2 - 1209600acx^4y^2 - 5377050c^2x^4y^2 + 3472875c^2x^6y^2 + 414720abxy^3 + 1812480bcxy^3 - 622080abx^3y^3 - 7476480bcx^3y^3 + 6048000bcx^5y^3 + 76800b^2y^4 + 215040acy^4 + 403200c^2y^4 - 964800b^2x^2y^4 - 1915200acx^2y^4 - 7269675c^2x^2y^4 + 1080000b^2x^4y^4 + 2116800acx^4y^4 + 20396250c^2x^4y^4 - 14056875c^2x^6y^4 - 1747200bcxy^5 + 8568000bcx^3y^5 - 7560000bcx^5y^5 - 313600c^2y^6 + 6057625c^2x^2y^6 - 18558750c^2x^4y^6 + 13505625c^2x^6y^6 \right),$$

$$C = \frac{1}{6144} \left(6144b + 6144ab + 6144bc + 9216ax + 5760a^2x + 9792b^2x + 17280cx + 23616acx + 13005c^2x - 3456a^2x^3 - 21120b^2x^3 - 13440cx^3 - 36480acx^3 - 48165c^2x^3 + 14400b^2x^5 + 20160acx^5 + 70875c^2x^5 - 33075c^2x^7 + 13824by + 23040aby + 31104bcy - 23040bx^2y - 87552abx^2y - 244800bcx^2y + 69120abx^4y + 512640bcx^4y - 302400bcx^6y - 9216axy^2 - 29952a^2xy^2 - 90432b^2xy^2 - 57600cxy^2 - 184896acxy^2 - 206100c^2xy^2 + 34560a^2x^3y^2 + 299520b^2x^3y^2 + 80640cx^3y^2 + 593280acx^3y^2 + 1273140c^2x^3y^2 - 216000b^2x^5y^2 - 423360acx^5y^2 - 2249100c^2x^5y^2 + 1190700c^2x^7y^2 - 7680by^3 - 39936aby^3 - 101760bcy^3 + 23040bx^2y^3 + 294912abx^2y^3 + 1500480bcx^2y^3 - 276480abx^4y^3 - 3676800bcx^4y^3 + 2318400bcx^6y^3 + 24192a^2xy^4 + 181440b^2xy^4 + 40320cxy^4 + 369600acxy^4 + 749070c^2xy^4 - 31104a^2x^3y^4 - 720000b^2x^3y^4 - 67200cx^3y^4 - 1430400acx^3y^4 - 5497950c^2x^3y^4 + 561600b^2x^5y^4 + 1108800acx^5y^4 + 10538850c^2x^5y^4 - 5843250c^2x^7y^4 + 23040aby^5 + 144000bcy^5 - 207360abx^2y^5 - 2566080bcx^2y^5 + 207360abx^4y^5 + 6860160bcx^4y^5 - 4536000bcx^6y^5 - 100800b^2xy^6 - 208320acxy^6 - 993300c^2xy^6 + 441600b^2x^3y^6 + 873600acx^3y^6 + 7954100c^2x^3y^6 - 360000b^2x^5y^6 - 705600acx^5y^6 - 16047500c^2x^5y^6 + 9187500c^2x^7y^6 - 67200bcy^7 + 1310400bcx^2y^7 - 3696000bcx^4y^7 + 2520000bcx^6y^7 + 437325c^2xy^8 - 3681125c^2x^3y^8 + 7686875c^2x^5y^8 - 4501875c^2x^7y^8 \right),$$

$$E = \frac{(x^2 - 1)(y^2 - 1)}{8192} \left(1152a^2 + 1344b^2 + 960ac + 1425c^2 - 1152a^2x^2 - 3840b^2x^2 - 7680acx^2 - 6675c^2x^2 + 4800b^2x^4 + 6720acx^4 + 16275c^2x^4 - 11025c^2x^6 - 13824abxy - 25920bcxy + 23040abx^3y + 103680bcx^3y - 100800bcx^5y - 1152a^2y^2 - 3840b^2y^2 - 7680acy^2 - 6675c^2y^2 + 10368a^2x^2y^2 + 48000b^2x^2y^2 + 96000acx^2y^2 + 147225c^2x^2y^2 - 67200b^2x^4y^2 - 134400acx^4y^2 - 468825c^2x^4y^2 + 385875c^2x^6y^2 + 23040abxy^3 + 103680bcxy^3 - 69120abx^3y^3 - 606720bcx^3y^3 + 672000bcx^5y^3 + 4800b^2y^4 + 6720acy^4 + 16275c^2y^4 - 67200b^2x^2y^4 - 134400acx^2y^4 - 468825c^2x^2y^4 + 120000b^2x^4y^4 + 235200acx^4y^4 + 1745625c^2x^4y^4 - 1561875c^2x^6y^4 - 100800bcxy^5 + 672000bcx^3y^5 - 840000bcx^5y^5 - 11025c^2y^6 + 385875c^2x^2y^6 - 1561875c^2x^4y^6 + 1500625c^2x^6y^6 \right).$$

Foram desenvolvidos dois programas, um para o cálculo da função ν e outro para traçar as geodésicas de uma métrica de Weyl. Nas seções seguintes descrevemos os métodos numéricos empregados, primeiro os de integração e como eles foram utilizados para calcular ν , em seguida os métodos para problemas de valor inicial, veja por exemplo [37] e [59].

D.1 Integração Numérica

Métodos numéricos para o cálculo de integrais consistem em aproximações do tipo

$$I = \int_a^b f(x)dx \sim \sum_{i=1}^n \lambda_i f(x_i) \quad (\text{D.1})$$

sendo que λ_i e x_i são constantes que dependerão do método empregado assim como o número de termos n .

O método mais simples é o método do trapézio onde fixamos $n = 2$, $\lambda_1 = \lambda_2 = (b - a)/2$, $x_1 = a$ e $x_2 = b$. Temos então a fórmula

$$I \sim \frac{b - a}{2} [f(a) + f(b)]. \quad (\text{D.2})$$

Se considerarmos, agora, a propriedade aditiva das integrais

$$I = \int_a^b f(x)dx = \sum_{i=1}^{n-1} \int_{x_i}^{x_{i+1}} f(x)dx, \quad (\text{D.3})$$

onde x_i é uma partição do intervalo $[a, b]$ de modo que $x_1 = a$ e $x_n = b$. Poderemos aplicar o método numérico independentemente em cada subintervalo obtendo um resultado mais preciso.

Existem diversas famílias de métodos de integração numérica neste trabalho escolhemos implementar os métodos de Gauss, mais precisamente os métodos de Gauss-Legendre e Gauss-Laguerre. No primeiro método, os valores de x_i são determinados pelas raízes dos polinômios de Legendre, e a região de integração deve ser o intervalo $[-1, 1]$. Quando queremos calcular uma integral em um intervalo finito arbitrário $[a, b]$ basta aplicar uma transformação na variável de integração que leve $[a, b]$ em $[-1, 1]$. O segundo método é indicado para integrais do tipo

$$\int_0^{\infty} e^{-x} f(x)dx \quad (\text{D.4})$$

e os valores de x_i são as raízes dos polinômios de Laguerre. Em ambos os casos os valores de λ_i são determinados de modo a obter a melhor precisão. Todos estes valores são tabelados, veja por exemplo [1].

A Integral de Weyl

A integral que desejamos calcular é

$$\nu = \int_{\Gamma} \tau \left[(\phi_r^2 - \phi_z^2) dr + 2\phi_r \phi_z dz \right], \quad (\text{D.5})$$

que não depende do caminho de integração, Γ , desde que este caminho esteja totalmente contido em regiões onde ϕ satisfaz a equação (2.7). Então para calcularmos ν numericamente precisamos determinar um caminho de integração que passe a distâncias seguras de qualquer singularidade de ϕ .

Como a função ν é determinada a menos de uma constante podemos fixar um ponto qualquer, p , no plano $r - z$ e determinar que $\nu(p) = 0$. Assim calculamos a integral acima (D.5) ao longo de algum caminho que leve até p , obtendo o valor de ν para qualquer ponto desejado. Relembrando a condição de regularidade sobre o eixo z , $\nu = 0$, vemos que a melhor posição para fixar p é no eixo z em uma região onde desejamos impor a regularidade.

Embora esta escolha seja matematicamente mais adequada, ela apresenta dificuldades para sua implementação, pois torna os caminhos de integração mais complexos. Escolhemos então trabalhar apenas com as funções ϕ que se anulam quando nos afastamos da origem e implementamos caminhos de integração que se dirigem ao infinito.

Como para aplicarmos os métodos de Gauss precisamos de valores tabelados, ficamos restritos a um número relativamente pequeno de pontos. Para eliminar este problema usamos a propriedade (D.3) e dividimos o caminho de integração em vários subcaminhos.

O programa que calcula os valores para ν foi chamado de *nnu*, uma contração de *Numerical Nu*. Para utilizá-lo é necessário gerar dois arquivos tipo texto, um que define os pontos onde desejamos os valores de ν e o outro com a descrição da função ϕ . Na leitura destes arquivos os espaços extras e as quebras de linha são ignorados, importando apenas a ordem dos valores e das palavras chave.

O arquivo que da malha deve conter o número de pontos. O número 2, indicando que cada ponto é descrito por dois valores e em seguida os pares r e z de cada ponto, como mostra o exemplo

```
4 2
0.0 0.0
1.0 0.0
0.0 1.0
1.0 1.0
```

um arquivo com este conteúdo determina que ν deve ser calculada nos quatro pontos $(0, 0)$, $(1, 0)$, $(0, 1)$ e $(1, 1)$. A malha, empregada para gerar as figuras contidas no texto, cobre a região $0 < r < 2$ e $-1 < z < 1$ e é composta por 10^3 pontos espaçados igualmente.

O outro arquivo é composto pela descrição da função ϕ , do caminho de integração e dos parâmetros para o método numérico. Seguindo o padrão

```
Curzon          1.0  0.0
Diagonal        0.0  1.0  1.0
IntegralParameters 100  2.0  100
```

Este arquivo determina que a função ν seja calculada a partir da função ϕ da métrica de Curzon (4.2), com $m = 1$ e $z_0 = 0$. O caminho de integração se afastará diagonalmente da origem e será dividido em 100 subcaminhos. Estes parâmetros serão melhor descritos a seguir.

O programa tem algumas soluções pré-definidas, listadas na tabela abaixo. Para calcularmos a função ν de qualquer uma destas soluções basta substituímos a palavra chave correspondente no exemplo acima.

Palavra Chave	Parâmetros	Solução	ϕ
Curzon	m z_0	Solução de Curzon	(4.2)
Rod	m z_0 l	Barra de comprimento $2l$	(4.3)
Schwarzschild	m z_0	Solução de Schwarzschild	(4.4)
Morgan	m z_0 a	Disco de Morgan-Morgan	(4.25)
DustDisk	z_0 a	Disco de Poeira	(4.32)
Appell	m z_0 a	Disco de Appell	(4.45)
Ring	m z_0 a	Anel de Letelier-Oliveira	(4.48)
WeylRing	m z_0 a	Anel Weyl-Bach	(4.50)

Como este programa foi desenvolvido sob o paradigma da Orientação Objeto a inclusão de novas soluções é muito simples, basta para isso definir uma classe, herdeira de `WeylPhi`, cuja definição está na página 134. Esta nova classe deve conter as fórmulas para ϕ , ϕ_r e ϕ_z . Então ela deve ser incluída na lista de soluções da função `InitPhi`, que pode ser vista na página 148.

A principal tarefa do programa é a função ν associada com sobreposições das soluções conhecidas. Para isso basta utilizar as palavras chave `DoublePhi`, que indica a sobreposição de duas funções ϕ , como no exemplo

```
DoublePhi
  Curzon      0.5  0.8
  WeylRing    0.5  0.0  1.0
  Diagonal    0.0  1.0  1.0
  IntegralParameters 100  2.0  100
```

ou `SuperPhi`, para sobrepor um número arbitrário de funções, neste último caso é necessário informar o número de funções ϕ que estão sendo sobrepostas, como segue

```
SuperPhi 5
  Curzon      1.0 -1.0
  Curzon      1.0  0.0
  Curzon      1.0  1.0
  DustDisk    1.0  2.0
  Ring        1.0  4.0  1.0
  AwayOf      1.0  6.0
  IntegralParameters 100  2.0  100
```

Nestas sobreposições a nova função ϕ é a soma das anteriores e ν é calculada diretamente pelos métodos numéricos.

Existem também duas escolhas pre-definidas para os caminhos de integração. Para incluir novos caminhos o processo é similar a inclusão de funções ϕ , neste caso a classe deve ser herdeira de `IntegralPath`, definida na página 139, e ser incluída na função `InitPath`, contida na página 149.

O caminho mais simples parte do ponto, onde se deseja calcular o valor da função ν , e segue em linha reta para o infinito. Para o utilizarmos, devemos incluir no arquivo de definições a linha

```
Diagonal 0.0  1.0  1.0
```

O primeiro número, que chamaremos z_0 , determina um plano $z = z_0$ onde esperamos encontrar singularidades, um anel ou disco por exemplo, de modo que os caminhos de integração serão refletidos com relação a este plano, como mostra a Figura D.1. Enquanto os dois últimos determinam a direção da linha reta e sua escala em relação ao parâmetro da curva, o primeiro fixa o incremento da coordenada r quando o parâmetro aumenta de uma unidade e o segundo é o equivalente para a coordenada z .

O segundo caminho, que chamamos `AwayOf`, recebe dois parâmetros, como neste exemplo

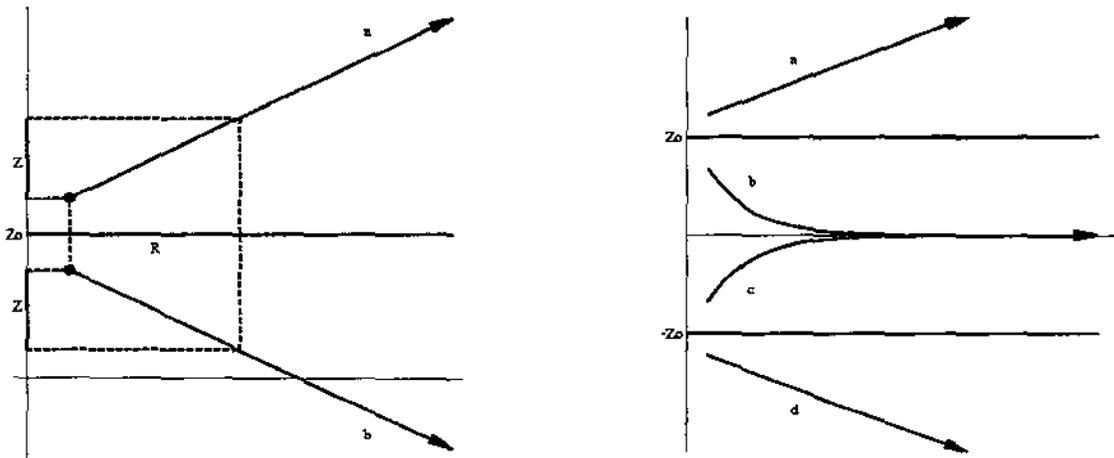


Figura D.1: Caminhos de Integração

```
AwayOf 1.0 6.0
```

O primeiro valor, z_0 , determina dois planos, $z = \pm z_0$, onde esperamos singularidades, assim os caminhos se afastarão deles. O segundo valor, v , determina com que “velocidade” os caminhos devem se afastar dos planos. Estas trajetórias podem ser vistas no segundo gráfico da Figura D.1.

As últimas informações de que o programa necessita são os parâmetros de integração, que são definidos pela linha

```
IntegralParameters 100 2.0 100
```

Para calcularmos numericamente uma integral sobre um caminho infinito, na verdade nós aproximamos seu valor por um obtido em um caminho finito. O primeiro valor determina o comprimento deste caminho, fixando o intervalo que tomaremos para o parâmetro da curva. Como este intervalo será grande para obtermos resultados mais precisos ele deve ser dividido em sub-intervalos, o número destes sub-intervalos é determinado pelo terceiro valor. Como o valor do integrando deve decrescer ao longo do caminho, o segundo valor fixa uma taxa de crescimento para os sub-intervalos.

O programa deve ser executado em linha de comando e espera por três nomes de arquivos, primeiro o que contém os parâmetros da solução, a malha e o arquivo de saída. Este arquivo de saída será tipo texto, os dois primeiros valores serão o número de linhas e o número 2, que indica que ele é composto por duas colunas. E então a lista dos valores das funções ν e ϕ em cada ponto, na mesma ordem que eles estiverem no arquivo da malha.

Como em todo resultado numérico devemos esclarecer os erros encontrados. Porém como os resultados foram empregados apenas para a geração de gráficos, onde a informação desejada é mais qualitativa do que quantitativa, consideramos desnecessário apresentar um inventário exaustivo dos erros.

Quando comparamos os resultados numéricos com as soluções exatas os erros foram da ordem de 10^{-5} , o que é suficiente para seu fim. Porém nos pontos muito próximos das singularidades de ϕ o erro aumenta drasticamente, em casos extremos chegando a valores como 10^7 , como este comportamento só foi verificado em um número muito pequeno de pontos, seus valores foram simplesmente desconsiderados. Outro comportamento patológico correu nos pontos mais afastados das singularidades, onde o erro relativo cresce, pois o valor da função ν fica muito próximo dos valores dos erros. Estes erros não tiveram nenhum tratamento especial, pois quando inseridos nos gráficos os valores de ν nestas regiões são praticamente nulos.

D.2 Problemas de Valor Inicial

Nesta seção apresentamos os métodos que foram empregados para a resolução de problemas de valor inicial para equações diferenciais ordinárias, que foram empregados para calcular as geodésicas das métricas de Weyl.

De modo geral um problema de valor inicial é descrito pelo conjunto de uma equação diferencial e uma condição inicial, como segue

$$y'(t) = f(y, t), \quad y(0) = y_0, \quad (\text{D.6})$$

onde $y(t)$ é um vetor em \mathbb{R}^n e t é um parâmetro real.

Numericamente o método mais simples, e conseqüentemente mais rústico, é método de Euler. Neste método a derivada y' é aproximada pela diferença

$$y'(t) \sim \frac{y(t+h) - y(t)}{h}, \quad (\text{D.7})$$

onde h é um número real positivo e pequeno. A solução obtida será uma amostragem y_n que aproxima $y(t_n)$ sendo que t_n é definido como $t_{n+1} = t_n + h$ e $t_0 = 0$.

Aplicando a equação (D.6) no ponto t_n temos que $y'(t_n) = f(y(t_n), t_n)$. Substituindo este resultado na aproximação (D.7) obtemos a relação de recorrência para os valores de y_n

$$y_{n+1} = y_n + hf(y_n, t_n). \quad (\text{D.8})$$

Um conjunto de métodos mais sofisticados são os métodos de Runge-Kutta. Nestes métodos obtém-se uma maior precisão empregando pontos intermediários entre t_n e t_{n+1} para calcularmos y_{n+1} . Notando que de modo geral podemos escrever a solução do problema (D.6) como

$$y(t) = y_0 + \int_0^t f(y(\tau), \tau) d\tau. \quad (\text{D.9})$$

Se buscarmos a solução apenas nos pontos de uma malha t_n , como descrito acima, teremos

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} f(y(\tau), \tau) d\tau. \quad (\text{D.10})$$

Então os métodos de Runge-Kutta são derivados da aplicação de diferentes métodos de integração para o cálculo desta integral.

O método de Runge-Kutta de segunda ordem é obtido aplicando-se a regra do trapézio para integrarmos (D.10). Porém, para que pudéssemos realmente efetuar este cálculo precisaríamos do valor de y_{n+1} , como este valor não é conhecido empregamos o método de Euler para calcular uma primeira aproximação \bar{y}_{n+1} e em seguida calculamos y_{n+1} , como mostra a fórmula abaixo

$$\begin{aligned} \bar{y}_{n+1} &= y_n + hf(y_n, t_n), \\ y_{n+1} &= y_n + \frac{h}{2} [f(y_n, t_n) + f(\bar{y}_{n+1}, t_{n+1})]. \end{aligned} \quad (\text{D.11})$$

Aplicando métodos de integração mais sofisticados obteremos métodos de Runge-Kutta mais precisos. Apresentamos a seguir os métodos de Runge-Kutta de terceira ordem

$$\begin{aligned} k_1 &= hf(y_n, t_n), & k_2 &= hf\left(y_n + \frac{k_1}{2}, t_n + \frac{h}{2}\right), & k_3 &= hf(y_n - k_1 + 2k_2, t_n + h), \\ y_{n+1} &= y_n + \frac{k_1 + 4k_2 + k_3}{6} \end{aligned} \quad (\text{D.12})$$

e quarta ordem

$$\begin{aligned}
 k_1 &= hf(y_n, t_n), & k_2 &= hf\left(y_n + \frac{k_1}{2}, t_n + \frac{h}{2}\right), \\
 k_3 &= hf\left(y_n + \frac{k_2}{2}, t_n + \frac{h}{2}\right), & k_4 &= hf(y_n - k_3, t_n + h), \\
 y_{n+1} &= y_n + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}
 \end{aligned} \tag{D.13}$$

Outro método que também é considerado como Runge-Kutta de quarta ordem é definido por

$$\begin{aligned}
 k_1 &= hf(y_n, t_n), & k_2 &= hf\left(y_n + \frac{k_1}{3}, t_n + \frac{h}{3}\right), \\
 k_3 &= hf\left(y_n + \frac{k_1}{3} + \frac{k_2}{3}, t_n + \frac{2h}{3}\right), & k_4 &= hf(y_n - k_1 - k_2 + k_3, t_n + h), \\
 y_{n+1} &= y_n + \frac{k_1 + 3k_2 + 3k_3 + k_4}{8}.
 \end{aligned} \tag{D.14}$$

Calculando as Geodésicas

O programa que calcula as geodésicas emprega o primeiro método de Runge-Kutta de quarta ordem (D.13), embora outros métodos também tenham sido implementados.

Foram implementadas as equações para as geodésicas tipo-tempo, espaço ou nulas. Restritas ao plano $\tau - z$ ou sobre todo o espaço. A trajetória pode ser calculada completa ou apenas sua projeção espacial. Porém as funções de entrada e saída do programa só foram implementadas para calcular as projeções espaciais das trajetórias tipo-tempo.

Para usarmos o programa, como no caso anterior, precisamos criar um arquivo tipo texto com as especificações necessárias. Este arquivo deve conter

1. As equações a serem resolvidas, Newtonianas ou relativísticas,
2. A condição inicial, as coordenadas r e z , e as derivadas, \dot{r} e \dot{z} , das coordenadas com relação ao parâmetro afim,
3. O tamanho do passo para os métodos numéricos, h . Nos exemplos contidos no texto foram empregados intervalos de 10^{-2} ou 10^{-3} ,
4. O número de passos que o método efetuará,
5. A função ϕ , que é definida exatamente como no caso anterior e
6. O caminho de integração que será usado para calcular o valor da função nu no ponto inicial. Para os demais pontos onde ν for necessária a própria trajetória da geodésica será usada como caminho de integração. Porém no caso de soluções para as quais o programa conhece a expressão analítica de ν este item será ignorado, podendo até mesmo ser suprimido do arquivo.

Estes itens devem ser incluídos no arquivo como mostra o exemplo

```

Newtonian
StartPoint  2.0  0.0  0.0  0.0
StepSize    0.001
NumPoints   2000
DoublePhi
  WeylRing      1.0  0.0  1.0
  Schwarzschild 1.0  1.0
  Diagonal      0.0  1.0  1.0

```

Para calcular as trajetórias relativísticas a palavra chave `Newtonian` deve ser eliminada.

Para executar este programa devemos usar o comando `weylgeo` seguido do nome do arquivo com as especificações e um nome de arquivo para a saída de dados. Este arquivo será criado de modo que possa ser executado diretamente como um arquivo do Matlab sendo da forma

```

g = [  r  z
      r  z
      r  z ];

f = [  r  z  dr  dz ];

```

A matriz `g` contém a trajetória da geodésica e `f` a posição e as derivadas do último ponto calculado, para que possamos continuar acompanhando a geodésica a partir de seu último ponto.

D.3 Listagens dos Programas

A seguir incluímos as listagens dos programas. Começando pelas classes de utilidade geral, contidas na sub-seção **Biblioteca Auxiliar**. Em seguida, apresentamos as que definem as soluções de Weyl, bem como as soluções pré-definidas descritas anteriormente. Estas classes compõem bibliotecas que são utilizadas pelos dois programas, `nnu` e `weylgeo`.

A sub-seção **Calculando ν Numericamente** engloba as funções que, usando as bibliotecas definidas nas sub-seções anteriores, calculam o valor de ν para uma lista de pontos.

Nas duas últimas sub-seções estão as classes que definem os métodos de resolução de problemas de valor inicial, para equações diferenciais ordinárias, que podem ser empregados para qualquer equação, desde que definida adequadamente. As descrições das equações para as geodésicas em espaços de Weyl e as equações de movimento Newtonianas.

Implementações Futuras

Existem vários pontos em que estes programas podem ser melhorados. O primeiro, e mais evidente, é incluir mais potenciais, tais como outros discos de Morgan-Morgan e multi-pólos. Outra possibilidade é anexar as bibliotecas de derivação automáticas [6], que permitiriam que novas funções ϕ fossem incluídas diretamente nos arquivos de especificação, sem a necessidade de incluí-las no código do programa, esta opção não foi explorada anteriormente pois os potenciais definidos no corpo do programa são avaliados mais rapidamente.

Quanto a melhorias nos métodos numéricos podemos citar duas. Uma bastante simples e já quase completamente implementada e que consiste em usar os métodos Preditor-Corretor de Adams, para os problemas de valor inicial. A outra é calcular ν empregando um caminho de integração que termine sobre o eixo z .

Também desejamos implementar funções que calculem os invariantes escalares ou as projeções do tensor de Riemann nas tetradas naturais. E os planos de Poincaré para as trajetórias das partículas de teste.

Biblioteca Auxiliar

Arquivo core.h

```

/*-----
Compatibility Tools and General Definitions
D'Afonseca L.A, DMA / IMECC / UNICAMP
September, 1998
-----*/
#ifndef CORE_INCLUDED
#define CORE_INCLUDED

#ifdef __WIN32__
typedef float      real;
typedef long int   integer;
typedef unsigned long int uinteger;
#define IntegerAbs labs
#else
typedef double     real;
typedef int        integer;
typedef unsigned int uinteger;
#define IntegerAbs abs
#endif
// Error Inquisitor -> Confiability X Speed
// #define INQUISITOR

#define Inquisition  exit(EXIT_FAILURE);

#ifdef INQUISITOR
#include <stdlib.h>
extern int InquisitorOK;
#endif
#endif

```

Arquivo core.cpp

```

/*-----
Compatibility Tools and General Definitions
D'Afonseca L.A, DMA / IMECC / UNICAMP
September, 1998
-----*/
#include "core.h"
#ifdef INQUISITOR
int InquisitorOK = 1;
#endif

```

Arquivo nreal.h

```

/*-----
Numeric Weyl Project
Definition of R^n vector
by Luis Alberto D'Afonseca - 28 October 1998
-----*/
#ifndef NREAL_INCLUDED
#define NREAL_INCLUDED

#include <math.h>
#include <iostream.h>
#include "core.h"

class NReal{
public:
    NReal();
    NReal( const uinteger n );
    NReal( const uinteger n, const real* );
    NReal( const NReal& );
    ~NReal(){ delete []ptr; };

    void    Reset( const uinteger i );
    int    Good() const { return ptr != NULL; };
    uinteger Dim () const { return nm; };

    operator real();
    NReal& operator = ( const real* ); // Only for pointers
    NReal& operator = ( const NReal& ); // of the same size.
    NReal operator + ( const NReal& ) const;
    NReal operator - ( const NReal& ) const;
    NReal& operator += ( const NReal& );
    NReal& operator -= ( const NReal& );
    NReal operator * ( const real ) const;
    real operator() ();
    real& operator() ( const uinteger ii );
    real operator() () const;
    real operator() ( const uinteger ii ) const;

```

```

friend NReal operator *( const real, const NReal& );

real Max();
real Min();
real Sum();
real NormOne();
real NormTwo();
real NormInf();

friend ostream& operator << ( ostream&, NReal& );
friend istream& operator >> ( istream&, NReal& );

protected:
    unsigned nn;
    real *ptr;
};

#ifdef INQUISITOR
#define Ptr_Test if( ptr == NULL ) Inquisition;
#define i_Test   if( ptr == NULL || i > nn ) Inquisition;
#else
#define Ptr_Test
#define i_Test
#endif

inline      NReal::operator real()      { Ptr_Test; return ptr[0]; }
inline real& NReal::operator() ( )      { Ptr_Test; return ptr[0]; }
inline real NReal::operator() ( ) const { Ptr_Test; return ptr[0]; }
inline real& NReal::operator()(const unsigned i) {i_Test; return ptr[i];}
inline real NReal::operator()(const unsigned i) const {i_Test; return ptr[i];}

#undef Ptr_Test
#undef i_Test
#endif

```

Arquivo nreal.cpp

```

/*-----
  Numeric Weyl Project
  Definition of R^n vector
  by Luis Alberto D'Afonseca - 28 October 1998
  -----*/
#include "nreal.h"
#ifdef INQUISITOR
#define nn_Test if( nn == 0 ) Inquisition;
#else
#define nn_Test
#endif

NReal::NReal(): nn ( 1 ){
    try { ptr = new real[ nn ]; }
    catch( ... ) { ptr = NULL; }
}

NReal::NReal( const unsigned n ): nn ( n ) {
    nn_Test;
    try { ptr = new real[ nn ]; }
    catch( ... ) { ptr = NULL; }
}

NReal::NReal( const unsigned n, const real* rr ): nn ( n ) {
    nn_Test;
    try {
        ptr = new real[ n ];
#ifdef INQUISITOR
        if( rr == NULL ) InquisitorOK = 0;
        else
#endif
        for( unsigned ii = 0; ii < n; ii++ )
            ptr[ii] = rr[ii];
    }catch( ... ){ ptr = NULL; }
}

NReal::NReal( const NReal& orig ): nn ( orig.nn ){
#ifdef INQUISITOR
    if( orig.ptr == NULL ) InquisitorOK = 0;
    else
#endif
    try{
        ptr = new real[ nn ];
        for( unsigned ii = 0; ii < nn; ii++ )
            ptr[ii] = orig.ptr[ii];
    }catch( ... ){ ptr = NULL; };
}

void NReal::Reset( const unsigned i ) {

```

```

    nn = i;
    nn_Test;
    try { ptr = new real[ nn ]; }
    catch( ... ) { ptr = NULL; }
}

NReal& NReal::operator = ( const real* rr ){
#ifdef INQUISITOR
    if( rr == NULL ) InquisitorOK = 0;
    else
#endif
    for( uinteger ii = 0; ii < nn; ii++ )
        ptr[ii] = rr[ii];
    return *this;
}

NReal& NReal::operator = ( const NReal& orig ){
#ifdef INQUISITOR
    if( nn != orig.nn ) Inquisition;
    if( orig.ptr == NULL ) InquisitorOK = 0;
    else
#endif
    for( uinteger ii = 0; ii < nn; ii++ )
        ptr[ii] = orig.ptr[ii];
    return *this;
}

NReal NReal::operator + ( const NReal& S ) const{
#ifdef INQUISITOR
    if( S.nn != nn ) Inquisition;
#endif
    NReal R( nn );
    for( uinteger ii = 0; ii < nn; ii++ )
        R.ptr[ii] = ptr[ii] + S.ptr[ii];
    return R;
}

NReal NReal::operator - ( const NReal& S ) const{
    NReal R( nn );
    for( uinteger ii = 0; ii < nn; ii++ )
        R.ptr[ii] = ptr[ii] - S.ptr[ii];
    return R;
}

NReal& NReal::operator += ( const NReal& S ){
    for( uinteger ii = 0; ii < nn; ii++ )
        ptr[ii] += S.ptr[ii];
    return *this;
}

NReal& NReal::operator -= ( const NReal& S ){
    for( uinteger ii = 0; ii < nn; ii++ )
        ptr[ii] -= S.ptr[ii];
    return *this;
}

NReal NReal::operator * ( const real r ) const{
    NReal R( nn );
    for( uinteger ii = 0; ii < nn; ii++ )
        R.ptr[ii] = r * ptr[ii];
    return R;
}

NReal operator * ( const real r, const NReal& P ){
    NReal R( P.nn );
    for( uinteger ii = 0; ii < P.nn; ii++ )
        R.ptr[ii] = r * P.ptr[ii];
    return R;
}

ostream& operator << ( ostream& os, NReal& R ){
#ifdef INQUISITOR
    if( R.ptr == NULL ) InquisitorOK = 0;
    else
#endif
    os << R.ptr[0];
    for( int ii = 1; ii < R.nn; ii++ )
        os << " " << R.ptr[ii];
    return os;
}

istream& operator >> ( istream& is, NReal& R ){
#ifdef INQUISITOR
    if( R.ptr == NULL ) InquisitorOK = 0;
    else
#endif
    for( int ii = 0; ii < R.nn; ii++ )
        is >> R.ptr[ii];
}

```

```

} return is;
}

real NReal::Max(){
  real max = ptr[0];
  for( int ii = 1; ii < nn; ii++ )
    if( ptr[ii] > max ) max = ptr[ii];
  return max;
}

real NReal::Min(){
  real min = ptr[0];
  for( int ii = 1; ii < nn; ii++ )
    if( ptr[ii] < min ) min = ptr[ii];
  return min;
}

real NReal::Sum(){
  real sum = ptr[0];
  for( int ii = 1; ii < nn; ii++ )
    sum += ptr[ii];
  return sum;
}

real NReal::NormOne(){
  real max = fabs( ptr[0] );
  for( int ii = 1; ii < nn; ii++ )
    if( fabs( ptr[ii] ) > max ) max = fabs( ptr[ii] );
  return max;
}

real NReal::NormTwo(){
  real sum = pow( ptr[0], 2 );
  for( int ii = 1; ii < nn; ii++ )
    sum += pow( ptr[ii], 2 );
  return sqrt( sum );
}

real NReal::NormInf(){
  real sum = fabs( ptr[0] );
  for( int ii = 1; ii < nn; ii++ )
    sum += fabs( ptr[ii] );
  return sum;
}
}

```

Arquivo function.h

```

/*-----
Numeric Weyl Project
Definition of function classes
by Luis Alberto D'Afonseca
-----*/
#ifndef FUNCTION_DEFINITION_INCLUDED
#define FUNCTION_DEFINITION_INCLUDED

#include "nreal.h"

class Function{
public:
  Function(){};
  ~Function(){};

  virtual int NumParameters(){ return 0; };
  virtual void SetParameters( const NReal& ){};
  virtual void Destructor(){};
};

class RNFunction: public Function{
public:
  RNFunction(){};
  ~RNFunction(){};

  virtual void EvaluateIn ( const NReal& In, NReal& Out, const real t ){};
  virtual int DimInput (){ return 0; };
  virtual int DimOutput (){ return 0; };
};
#endif

```

Arquivo fspecial.h

```

/*-----
Definition of Special functions
by Luis Alberto D'Afonseca
18 - November - 1998
-----*/
#ifndef SPECIAL_FUNCTIONS_INCLUDED

```

```
#define SPECIAL_FUNCTIONS_INCLUDED
#include <math.h>
#include "core.h"

// Mathematical Constants Defined
extern const int GaussNumPointers;
extern const int LaguerreNumPointers;
extern const real GaussX[];
extern const real GaussW[];
extern const real GaussX_p1[]; // = GaussX + 1
extern const real LaguerreX[];
extern const real LaguerreWxExp[];

// Special Functions Defined
real EllipticK( const real k, real ca = 0.0003 );
real DkEllipticK( const real k, const uinteger = 1 );
real cel( real qqc, real pp, real aa, real bb, real ca );
#endif
```

Arquivo fspecial.cpp

```
/*-----
Definition of Special Elliptic K function
by Luis Alberto D'Afonseca
18 - November - 1998
-----*/
#include "fspecial.h"
// Constants Definitions
const int GaussNumPointers = 10;
const real GaussX[] = { -0.973906528, -0.865063367, -0.679409568, -0.433395394,
-0.148874339, 0.148874339, 0.433395394, 0.679409568,
0.865063367, 0.973906528 },
GaussW[] = { 0.066671344, 0.149451349, 0.219086363, 0.269266719,
0.295524225, 0.295524225, 0.269266719, 0.219086363,
0.149451349, 0.066671344 },
GaussX_p1[] = { 0.026093472, 0.134936633, 0.320590432, 0.566604606,
0.851125661, 1.148874339, 1.433395394, 1.679409568,
1.865063367, 1.973906528 }; // Gauss_X + 1
const int LaguerreNumPointers = 4;
const real LaguerreX[] = { 0.322547689619392, 1.745761101158350,
4.536620296921130, 9.396070912301130 },
LaguerreWxExp[] = { 0.832739123838, 2.04810243845,
3.63114630582, 6.48714508441 };

// Function Definitions
real EllipticK( const real k, real ca ){
return cel( sqrt( 1 - k*k ), 1, 1, 1, ca );
}

real DkEllipticK( const real k, const uinteger n ){
real I = 0, k2 = k * k,
size = M_PI_2 / (real)n,
szp2 = size / 2.0;
for( uinteger tt = 0; tt < n; tt++){
real A = tt * size;
for( int ii = 0; ii < GaussNumPointers; ii++){
real s2 = pow( sin( A + GaussX_p1[ii] * szp2 ), 2 );
I += szp2 * GaussW[ii] * k * s2 * pow( 1.0 - k2*s2, -1.5 );
}
}
return I;
}

real cel( real qqc, real pp, real aa, real bb, real ca ){
if( qqc == 0 ) { return 0; /* error */ }
real f, g, q, qc = fabs( qqc ), a = aa,
b = bb, p = pp, e = qc, em = 1.0;

if( p > 0 ){
p = sqrt( p ); b = b / p;
} else {
f = qc*qc; q = 1 - f; g = 1 - p; f = f - p;
q = q * ( b - a * p ); p = sqrt( f / g );
a = ( a - b ) / g; b = -q / ( g * g * p ) + a * p;
}

while(1){
f = a; a = a + b / p; g = e / p; b = b + f * g;
b = b + b; p = g + p; g = em; em = qc + em;

if( fabs( g - qc ) > ( g + ca ) ){
qc = sqrt( e ); qc = qc + qc; e = qc * em;
}
else return M_PI_2 * ( b + a * em ) / ( em * ( em + p ) );
}
}
```

Arquivo module.h

```

-----
/*
  Definition of integer module n
  D'Afonseca L.A., DMA / IMECC / UNICAMP
  11, November, 1998
-----*/
#ifdef MODULE_INCLUDED
#define MODULE_INCLUDED
#include <math.h>
#include "core.h"
#define mm(v) IntegerAbs(v%mod)

class IntModule{
public:
  IntModule(                ): mod( 10 ), val( 0 ){ };
  IntModule( uinteger m     ): mod(  m ), val( 0 ){ };
  IntModule( uinteger m, integer v ): mod(  m ), val( mm(v) ){ };
  IntModule( const IntModule& o ): mod( o.mod ), val( o.val ){ };
  ~IntModule(){ };

  operator integer(){ return val; };

  void SetModule ( uinteger m ){ mod = m; val = mm(val); };
  uinteger GetModule () { return mod; };

  uinteger operator = ( const integer i ){val = mm( i ); return val; };
  uinteger operator += ( const integer i ){val = mm( val +i ); return val; };
  uinteger operator -= ( const integer i ){val = mm( val -i ); return val; };
  uinteger operator + ( const integer i ){ return mm( val +i ); };
  uinteger operator - ( const integer i ){ return mm( val -i ); };
  uinteger operator * ( const integer i ){ return mm( val *i ); };
  uinteger operator / ( const integer i ){ return mm( val /i ); };

  // Prefix
  uinteger operator ++ () {val = mm(++val); return val; };
  uinteger operator -- () {val = mm(--val); return val; };
  // Postfix
  uinteger operator ++ (int){uinteger old = val; val = mm(++val); return old; };
  uinteger operator -- (int){uinteger old = val; val = mm(--val); return old; };

private: uinteger mod, val;
};
#endif

```

Arquivo table.h

```

-----
/*
  General Tools Librarie - Table Class
  by Luis Alberto D'Afonseca
  begin - 24, August, 1998
  last - 04, October, 1998
-----*/
#ifdef TABLE_CLASS_INCLUDED
#define TABLE_CLASS_INCLUDED
#include <iostream.h>
#include "core.h"
#include "nreal.h"

template < class T > class TTable{
public:
  TTable();
  TTable( const uinteger );
  TTable( const uinteger, const uinteger );
  TTable( const TTable& );
  ~TTable();

  int good(){ return VoidPtr != NULL; };
  virtual int Reset ( const uinteger, const uinteger );

  uinteger GetLines () { return ll; };
  uinteger GetColumns () { return cc; };

  virtual Tk operator () ( const uinteger, const uinteger );
  Tk operator () ( const uinteger );

  friend ostream& operator << ( ostream&, TTable<T>& );
  friend istream& operator >> ( istream&, TTable<T>& );

protected:
  TTable( void*, const uinteger, const uinteger );
  void *VoidPtr;
  uinteger ll, cc; // number of lines and of columns
};

// Spetial Types
typedef TTable<real> ReTable;

```

```

typedef TTable<integer> InTable;
typedef TTable<uinteger> UiTable;

// Inline Functions
#define x_TTableIn_   template< class T > inline   TTable<T>::
#define x_TTableIn( X ) template< class T > inline X TTable<T>::
#define x_TTable_     template< class T >   TTable<T>::
#define x_TTable( X )   template< class T > X TTable<T>::
#define T_ptr ((T *)VoidPtr)

// -----
x_TTableIn_   TTable(): VoidPtr( NULL ), ll( 0 ), cc( 0 ) {}
x_TTableIn_   TTable(){ delete [] T_ptr; }
x_TTableIn( Tk ) operator()( const uinteger ii ){ return operator()(ii,0);}

x_TTableIn_   TTable( void* p, const uinteger l, const uinteger c ):
  VoidPtr( p ), ll( l ), cc( c ){

x_TTableIn( Tk ) operator()( const uinteger ii, const uinteger jj ){
  #ifdef INQUISITOR
    if( ii > ll || jj > cc || VoidPtr == NULL ) Inquisition;
  #endif
  return T_ptr[ ii * cc + jj ];
}

// Template Functions
x_TTable_   TTable( const uinteger l ): ll( l ), cc( 1 ){
  #ifdef INQUISITOR
    if( ll == 0 ) Inquisition;
  #endif
  try      { T_ptr = new T[ ll ]; }
  catch ( ... ) { T_ptr = NULL; };
}

x_TTable_   TTable( const uinteger l, const uinteger c ): ll( l ), cc( c ){
  #ifdef INQUISITOR
    if( ll == 0 || cc == 0 ) Inquisition;
  #endif
  try      { T_ptr = new T[ ll*cc ]; }
  catch ( ... ) { T_ptr = NULL; };
}

x_TTable_   TTable( const TTable& table ): ll ( table.ll ), cc ( table.cc ){
  try {
    integer size = ll*cc;
    T_ptr = new T[ size ];
    if( table.VoidPtr != NULL ){
      for( uinteger ii = 0; ii < size; ii++ )
        T_ptr[ii] = ((T *)table.VoidPtr)[ii];
    }
  }catch ( ... ) { T_ptr = NULL; };
}

x_TTable( int ) Reset ( const uinteger l , const uinteger c ){
  delete[] T_ptr;
  ll = l; cc = c;
  try      { T_ptr = new T[ ll*cc ]; }
  catch ( ... ) { T_ptr = NULL; };
  return ( T_ptr != NULL );
}

template< class T > ostream& operator << ( ostream& os, TTable<T>& tab ){
  #ifdef INQUISITOR
    if ( tab.VoidPtr == NULL ) InquisitorOK = 0;
  else
  #endif
  for( uinteger ii = 0; ii < tab.ll; ii++ ) {
    for( uinteger jj = 0; jj < tab.cc; jj++ ) os << tab(ii,jj) << " ";
    os << endl;
  }
  return os;
}

template< class T > istream& operator >> ( istream& is, TTable<T>& tab ){
  #ifdef INQUISITOR
    if ( tab.VoidPtr == NULL ) InquisitorOK = 0;
  else
  #endif
  for( uinteger ii = 0; ii < tab.ll; ii++ )
    for( uinteger jj = 0; jj < tab.cc; jj++ ) is >> tab(ii,jj);
  return is;
}

#undef T_ptr
#undef x_TTableIn_
#undef x_TTableIn
#undef x_TTable_
#undef x_TTable

```

```
#endif
```

Arquivo ctable.h

```

/*-----
General Tools Librarie - TColTable - Long Columns Table Class
by Luis Alberto D'Afonseca
begin - 16, september, 1998
last - 04, October, 1998
-----*/
#ifndef COLUMN_TABLE_INCLUDED
#define COLUMN_TABLE_INCLUDED
#include "table.h"

template < class T > class TColTable: public TTable<T>{
public:
    TColTable() {};
    TColTable( const uinteger );
    TColTable( const uinteger, const uinteger );
    TColTable( const TColTable& );
    ~TColTable();

    virtual int Reset      ( const uinteger, const uinteger );
    virtual T& operator () ( const uinteger, const uinteger );
};

typedef TColTable<real>      ReColTable;
typedef TColTable<integer>  InColTable;
typedef TColTable<uinteger> UiColTable;

#define PtrPtr      ((T**)VoidPtr)
#define PtrColumn(col) ((T**)VoidPtr)[col]
#define x_TColTableIn template< class T > inline T& TColTable<T>::

// Inline functions
x_TColTableIn operator()( const uinteger ii, const uinteger jj ){
    #ifdef INQUISITOR
        if( ii > ll || jj > cc || VoidPtr == NULL ) Inquisition;
    #endif
    return PtrColumn(jj)[ii];
}

// Template Functions
#define x_TColTable_      template< class T > TColTable<T>::
#define x_TColTable( X ) template< class T > X TColTable<T>::

x_TColTable_ TColTable( const uinteger l, const uinteger c ): TTable<T>(l,c){
    #ifdef INQUISITOR
        if( ll == 0 ) Inquisition;
    #endif
    try{
        PtrPtr      = new T*[l];
        PtrColumn(0) = new T[ll];
    }catch(...){ VoidPtr = NULL; };
}

x_TColTable_ TColTable( const uinteger l, const uinteger c ):TTable<T>(l,c){
    #ifdef INQUISITOR
        if( ll == 0 || cc == 0 ) Inquisition;
    #endif
    try{
        PtrPtr = new T*[cc];
        for( uinteger jj = 0; jj < cc; jj++ ) PtrColumn(jj) = new T[ ll ];
    }catch(...){ VoidPtr = NULL; };
}

x_TColTable_ TColTable( const TColTable& orig ){
    ll = orig.ll; cc = orig.cc;
    try{
        PtrPtr = new T*[cc];
        for( uinteger jj = 0; jj < cc; jj++ ){
            PtrColumn(jj) = new T[ ll ];
            for( int ii = 0; ii < ll; ii++ )
                PtrColumn(jj)[ii] = ((T**)orig.VoidPtr)[jj][ii];
        }
    }catch ( ... ) { VoidPtr = NULL; };
}

x_TColTable_ ~TColTable(){
    if( VoidPtr != NULL ){
        for( uinteger jj = 0; jj < cc; jj++ )
            delete [] PtrColumn(jj);
        delete [] PtrPtr;
    }
}

x_TColTable( int ) Reset ( const uinteger l , const uinteger c ){

```

```

if( VoidPtr != NULL ){
  for( uinteger jj = 0; jj < ij; jj++ )
    delete [] PtrColumn[jj];
  delete [] PtrPtr;
}
ll = l; cc = c;
try{
  PtrPtr = new T[cc];
  for( uinteger jj = 0; jj < cc; jj++ )
    PtrColumn[jj] = new T[ ll ];
} catch ( ... ) { VoidPtr = NULL; };
return ( VoidPtr != NULL );
}

#undef PtrPtr
#undef PtrColumn
#undef r_IColTable_
#undef x_IColTable
#endif

```

Definição das Soluções de Weyl

Arquivo weylfunc.h

```

/*-----
  Numeric Weyl Project
  Definition of virtual classes for implement Weyl functions
  by Luis Alberto D'Afonseca - 27 October 1998
  -----*/

#ifndef WEYL_FUNCTION_INCLUDED
#define WEYL_FUNCTION_INCLUDED
#include "function.h"

class WeylNu: public Function{
public:
  WeylNu(){};
  ~WeylNu(){};

  virtual void EvaluateIn( const real r, const real z, real& nu ){};
};

class WeylPhi: public Function{
public:
  WeylPhi(){};
  ~WeylPhi(){};

  virtual WeylNu* ExactNu ()( return NULL; );
  virtual void EvaluateIn( const real r, const real z,
    real& phi, real& drphi, real& dzphi ){};
};

WeylPhi* InitPhi( istream& is );
#endif

```

Arquivo doublephi.h

```

/*-----
  Numeric Weyl Project
  Definition of virtual classes for implement
  Two Superposed Phi Weyl functions
  by Luis Alberto D'Afonseca - 01 November 1998
  -----*/

#ifndef DOUBLE_PHI_INCLUDED
#define DOUBLE_PHI_INCLUDED
#include "weylfunc.h"

class DoublePhi: public WeylPhi{
public:
  DoublePhi(): FirstPhi( NULL ), SecondPhi( NULL ){};
  DoublePhi( WeylPhi*, WeylPhi* );
  ~DoublePhi(){};

  void SetPhi ( WeylPhi*, WeylPhi* );
  void SetFirstPhi ( WeylPhi* );
  void SetSecondPhi( WeylPhi* );
  int NumParameters ();
  void SetParameters ( const NReal& );
  void EvaluateIn( const real r, const real z,
    real& phi, real& drphi, real& dzphi );
  void Destructor();

protected: WeylPhi *FirstPhi, *SecondPhi;
};

inline DoublePhi::DoublePhi( WeylPhi* f, WeylPhi* s ):

```

```

    FirstPhi( f ), SecondPhi( s ) {}

inline void DoublePhi::SetPhi ( WeylPhi* f, WeylPhi* s ){
    FirstPhi = f; SecondPhi = s;
}

inline void DoublePhi::SetFirstPhi ( WeylPhi* f ){ FirstPhi = f; }
inline void DoublePhi::SetSecondPhi( WeylPhi* s ){ SecondPhi = s; }

inline int DoublePhi::NumParameters(){
    #ifdef INQUISITOR
        if( FirstPhi == NULL || SecondPhi == NULL ){
            InquisitorOK = 0;
            return 0;
        }
    #endif

    return FirstPhi->NumParameters() + SecondPhi->NumParameters();
}

inline void DoublePhi::EvaluateIn( const real r, const real z,
                                   real& phi, real& drphi, real& dzphi ){
    #ifdef INQUISITOR
        if( FirstPhi == NULL || SecondPhi == NULL ){
            InquisitorOK = 0;
            return;
        }
    #endif

    real p, dr, dz;

    FirstPhi->EvaluateIn ( r, z, p, dr, dz );
    SecondPhi->EvaluateIn ( r, z, phi, drphi, dzphi );

    phi += p; drphi += dr; dzphi += dz;
}

inline void DoublePhi::Destructor(){
    FirstPhi->Destructor();
    SecondPhi->Destructor();

    delete FirstPhi;
    delete SecondPhi;
}
#endif

```

Arquivo superphi.h

```

/*-----
Numeric Weyl Project
Definition of virtual classes for implement
several Superposed Phi Weyl functions
by Luis Alberto D'Afonseca - 01 November 1998
-----*/
#ifdef SUPER_PHI_INCLUDED
#define SUPER_PHI_INCLUDED
#include "weylfunc.h"

class SuperPhi: public WeylPhi{
public:
    SuperPhi( uinteger n );
    ~SuperPhi(){ delete[] Phi; };

    void SetPhi ( uinteger i, WeylPhi* );
    int NumParameters ();
    void SetParameters ( const NReal& );
    void EvaluateIn( const real r, const real z,
                    real& phi, real& drphi, real& dzphi );
    void Destructor();

protected: unsigned int nn; WeylPhi **Phi;
};

inline void SuperPhi::SetPhi ( uinteger n, WeylPhi* p ){
    #ifdef INQUISITOR
        if( n >= nn || Phi == NULL ){
            InquisitorOK = 0;
            return;
        }
    #endif

    Phi[n] = p;
}
#endif

```

Arquivo weylfunc.cpp

```

/*-----
Numeric Weyl Project
Definition of virtual classes for implement Phi Weyl functions
by Luis Alberto D'Afonseca - 01 November 1998
-----*/
#include "doubphi.h"
#include "superphi.h"

// --- Double Phi ---
void DoublePhi::SetParameters ( const NReal& P ){
#ifdef INQUISITOR
    if( FirstPhi == NULL || SecondPhi == NULL || P.Dim() != NumParameters() ){
        InquisitorOK = 0;
        return;
    }
#endif

    NReal F( FirstPhi->NumParameters() ), S( SecondPhi->NumParameters() );
    int aux = FirstPhi->NumParameters();

    for( int ii = 0; ii < aux; ii++ ) F(ii) = P(ii);
    for( ii = 0; ii < SecondPhi->NumParameters(); ii++ ) S(ii) = P( ii + aux );

    FirstPhi->SetParameters( F );
    SecondPhi->SetParameters( S );
}

// --- Superposed Phi ---
SuperPhi::SuperPhi( unsigned n ): nn( n ){
#ifdef INQUISITOR
    if( nn == 0 ){
        InquisitorOK = 0;
        Phi = NULL;
        return;
    }
#endif
    try { Phi = new WeylPhi*[nn]; }
    catch( ... ){ Phi = NULL; }
}

int SuperPhi::NumParameters(){
#ifdef INQUISITOR
    if( Phi == NULL ) {
        InquisitorOK = 0;
        return 0;
    } else {
        for( int kk = 0; kk < nn && ok; kk++ )
            if( Phi[kk] == NULL ){
                InquisitorOK = 0;
                return 0;
            }
    }
#endif
}

int num = 0;
for( int ii = 0; ii < nn; ii++ ) num += (Phi[ii]->NumParameters());
return num;
}

void SuperPhi::SetParameters ( const NReal& P ){
#ifdef INQUISITOR
    if( P.Dim() != NumParameters() ){
        InquisitorOK = 0;
        return;
    }
#endif

    NReal *Ptr;
    int cont = 0;

    for( int pp = 0; pp < nn; pp++ ){
        int aux = (Phi[pp]->NumParameters());
        Ptr = new NReal( aux );
        for( int ii = 0; ii < aux; ii++ ){
            (*Ptr)(ii) = P(cont);
            cont++;
        }

        (Phi[pp]->SetParameters( *Ptr );
        delete Ptr;
    }
}

void SuperPhi::EvaluateIn( const real x, const real z,

```

```

                                real& phi, real& drphi, real& dzphi ){
#ifdef INQUISITOR
    if( Phi == NULL ){
        InquisitorOK = 0;
        return 0;
    }else{
        for( int kk = 0; kk < mn && ok; kk++ )
            if( Phi[kk] == NULL ){
                InquisitorOK = 0;
                return 0;
            }
    }
#endif

    phi = 0; drphi = 0; dzphi = 0;

    for( int pp = 0; pp < nn; pp++ ){
        real p, dr, dz;
        (Phi[pp])->EvaluateIn ( r, z, p, dr, dz );

        phi += p; drphi += dr; dzphi += dz;
    }
}

void SuperPhi::Destructor(){
    for( int pp = 0; pp < nn; pp++ ){
        (Phi[pp])->Destructor();
        delete Phi[pp];
    }
}

```

Arquivo weylsol.h

```

/*-----
Numeric Weyl Project
Definition of virtual classes for implement Weyl Solutions
by Luis Alberto D'Afonso - 27 October 1998
-----*/
#ifndef WEYL_SOLUTION_INCLUDED
#define WEYL_SOLUTION_INCLUDED
#include "weylfunc.h"
#include "table.h"

typedef void (*EchoFunction)( unsigned );

class WeylSol: public Function{
public:
    WeylSol(          ): PtPhi( 0 ) {};
    WeylSol( WeylPhi* p );
    ~WeylSol(){};

    void SetWeylPhi   ( WeylPhi* p ){ PtPhi = p; };
    void EvaluateIn   ( const real x, const real z );
    void EvaluateForAll ( TTable<real>* Points, TTable<real>* Values,
                        EchoFunction Echo );

    real Phi () { return phi; };
    real Nu  () { return nu; };
    real DrPhi() { return drphi; };
    real DzPhi() { return dzphi; };
    real DrNu () { return drnu; };
    real DzNu () { return dznu; };
    void Destructor();

protected:
    void SetNuDerivatives ( const real x );
    void EvaluatePhiIn    ( const real x, const real z );
    virtual void EvaluateNuIn ( const real x, const real z ){};

    WeylPhi* PtPhi;
    real phi, drphi, dzphi, nu, drnu, dznu;
};

inline WeylSol::WeylSol( WeylPhi* p ): PtPhi( p ){
#ifdef INQUISITOR
    if ( p == NULL ) InquisitorOK = 0;
#endif
}

inline void WeylSol::EvaluateIn ( const real x, const real z ){
    EvaluateNuIn ( x, z );
    EvaluatePhiIn ( x, z );
}

inline void WeylSol::EvaluatePhiIn ( const real x, const real z ){
#ifdef INQUISITOR
    if( PtPhi == NULL ) InquisitorOK = 0;
    else

```

```

#endif
{
    PtPhi->EvaluateIn ( r, z, phi, drphi, dzphi );
    SetNuDerivatives ( r );
}

inline void WeylSol::SetNuDerivatives( const real r ){
    drnu = r * ( drphi * drphi - dzphi * dzphi );
    dznu = 2.0 * r * drphi * dzphi;
}

inline void WeylSol::Destructor(){
    PtPhi->Destructor();
    delete PtPhi;
}

class ExactWeylSol: public WeylSol{
public:
    ExactWeylSol(): WeylSol( 0 ), PtNu( 0 ) {};
    ExactWeylSol( WeylPhi* p, WeylNu* n ): WeylSol( p ), PtNu( n ) {};
    ~ExactWeylSol(){};

    void SetWeylNu ( WeylNu* n ){ PtNu = n; };
    void Destructor();

protected:
    void EvaluateNuIn ( const real r, const real z );
    WeylNu* PtNu;
};

inline void ExactWeylSol::EvaluateNuIn ( const real r, const real z ){
#ifdef INQUISITOR
    if( PtNu == NULL ) InquisitorOK = 0;
    else
#endif
    PtNu->EvaluateIn( r, z, nu );
}

inline void ExactWeylSol::Destructor(){
    WeylSol::Destructor();
    PtNu->Destructor();
    delete PtNu;
}
#endif

```

Arquivo weylsol.cpp

```

/*-----
Numeric Weyl Project
Definition of virtual classes for implement Weyl Solutions
by Luis Alberto D'Afonseca - 28 October 1998
-----*/
#include "weylsol.h"
#define R(i) (*Points)(i,0)
#define Z(i) (*Points)(i,1)
#define NU(i) (*Values)(i,0)
#define PHI(i) (*Values)(i,1)

void WeylSol::EvaluateForAll ( TTable<real>* Points, TTable<real>* Values,
const unsigned EchoStep, EchoFunction Echo ){
#ifdef INQUISITOR
    if( Points == NULL || Values == NULL || Echo == NULL || EchoStep == 0 ){
        InquisitorOK = 0;
        return;
    }else if( Points->GetLines() != Values->GetLines() ||
        Points->GetLines() == 0 ||
        Points->GetColumns() != 2 ||
        Values->GetColumns() != 6 ){
        InquisitorOK = 0;
        return;
    }
#endif
    unsigned size = Points->GetLines(), ii = 0;
    do{
        unsigned stop = ( ii + EchoStep > size)? size: ii + EchoStep;

        while( ii < stop ){
            EvaluateIn( R( ii ), Z( ii ) );

            NU( ii ) = nu; PHI(ii) = phi; ii++;
        }
        Echo( ii );
    }
}

```

```

} while( ii < size );
}

```

Arquivo intepath.h

```

/*-----
Numeric Weyl Project
Definition of Integration Path classes
by Luis Alberto D'Afonseca - 27 October 1998
-----*/
#ifndef INTEGRATION_PATHS_INCLUDED
#define NUMERIC_WEYL_SOLUTION_INCLUDED
#include <iostream.h>
#include "function.h"

class IntegrationPath: public Function {
public:
    IntegrationPath(){};
    ~IntegrationPath(){};

    virtual void StartPath ( const real r, const real z );
    virtual void EvaluateIn ( const real t, real& r, real& z ){};

protected: real r_init, z_init;
};

IntegrationPath* InitPath( istream& is );

class DiagonalPath: public IntegrationPath{
public:
    DiagonalPath();
    DiagonalPath( const real Zo, const real rs, const real zs );
    ~DiagonalPath(){};

    int NumParameters(){ return 3; };
    void SetParameters( const NReal& ); // zo, r_step and z_step_abs
    void StartPath ( const real r, const real z );
    void EvaluateIn ( const real t, real& r, real& z );

protected: real r_step, z_step, z_abs, zo;
};

class AwayOfPath: public IntegrationPath{
public:
    AwayOfPath(
        ): AwayOf( 1.0 ), Speed( -0.01 ) {};
    AwayOfPath( const real a, const real s ): AwayOf( a ), Speed( -s ) {};
    ~AwayOfPath(){};

    int NumParameters(){ return 2; };
    void SetParameters( const NReal& ); // AwayOf and Speed
    void StartPath ( const real r, const real z );
    void EvaluateIn ( const real t, real& r, real& z );

protected: real AwayOf, Speed, Ce, Cl, S;
};

inline void IntegrationPath::StartPath( const real r, const real z ){
    r_init = r; z_init = z;
}

inline DiagonalPath::DiagonalPath(): r_step(1.0), z_abs(1.0), zo(0.0){}

inline
DiagonalPath::DiagonalPath( const real Zo, const real rs, const real zs ):
    r_step( rs ), z_abs ( zs ), zo( Zo ) {}

inline void DiagonalPath::SetParameters( const NReal& P ){
    #ifdef INQUISITOR
        if( P.Dim() != 2 ){
            InquisitorOK = 0;
            return;
        }
    #endif
    zo = P(0); r_step = P(1); z_abs = P(2);
}

inline void DiagonalPath::StartPath( const real r, const real z ){
    r_init = r; z_init = z;
    z_step = z_abs * (real)((z > zo)? 1: -1);
}

inline void DiagonalPath::EvaluateIn( const real t, real& r, real& z ){
    r = r_init + r_step * t; z = z_init + z_step * t;
}

inline void AwayOfPath::SetParameters( const NReal& P ){

```

```

#ifdef INQUISITOR
  if( P.Dim() != 2 ){
    InquisitorOK = 0;
    return;
  }
#endif
AwayOf = P(0); Speed = - P(1);
}

inline void AwayOfPath::StartPath( const real r, const real z ){
  r_init = r; z_init = z;
  if( fabs(z) < AwayOf ){
    Ce = z_init; Cl = 0.0;
  }else{
    Ce = 0.0; Cl = 1.0; S = (z < 0)? -1: 1;
  }
}

inline void AwayOfPath::EvaluateIn( const real t, real& r, real& z ){
  r = r_init + t;
  z = Ce * exp( Speed * t ) + Cl * ( z_init + S * t );
}
#endif

```

Arquivo nweylsol.h

```

/*-----
  Numeric Weyl Project
  Definition of virtual classes for implement Numeric Weyl Solutions
  by Luis Alberto D'Afonseca - 27 October 1998
  -----*/

#ifdef NUMERIC_WEYL_SOLUTION_INCLUDED
#define NUMERIC_WEYL_SOLUTION_INCLUDED
#include "weylsol.h"
#include "intepath.h"

class NWeylSol: public WeylSol{
public:
  NWeylSol( ): WeylSol( ) {};
  NWeylSol( WeylPhi* p ): WeylSol( p ) {};
  NWeylSol( WeylPhi* p, const real ro, const real zo, const real no );
  ~NWeylSol(){};

  virtual int NumParameters(){ return 3; };
  virtual void SetParameters( const MReal& ); // Ro, Zo and No

protected:
  void EvaluateNuIn ( const real r, const real z );
  real Ro, Zo, No;
};

class JumpNWeylSol: public NWeylSol{
public:
  JumpNWeylSol(){};
  JumpNWeylSol( WeylPhi* p ): NWeylSol( p ) {};
  JumpNWeylSol( WeylPhi* p, const real ro, const real zo, const real no );
  ~JumpNWeylSol(){};

protected:
  void EvaluateNuIn ( const real r, const real z );
};

class AssintNWeylSol: public JumpNWeylSol{
public:
  AssintNWeylSol(): InPath( 0 ) {};
  AssintNWeylSol( WeylPhi*, IntegrationPath* );
  AssintNWeylSol( WeylPhi*, IntegrationPath*,
                 const real s, const real f, const uinteger n );
  ~AssintNWeylSol(){};

  virtual int NumParameters(){ return 3; };
  virtual void SetParameters( const MReal& ); // Size, Factor and Ndiv

  void SetIntegrationPath ( IntegrationPath* I ){ InPath = I; };
  void Destructor();

protected:
  void EvaluateNuIn ( const real r, const real z );
  real step, factor;
  uinteger ndiv;
  IntegrationPath* InPath;
};

// NWeylSol class
inline NWeylSol::NWeylSol( WeylPhi* p, const real ro, const real zo,
                          const real no ):
  WeylSol( p ), Ro ( ro ), Zo ( zo ), No ( no ){}

```

```

inline void NWeylSol::SetParameters( const NReal& P ){
#ifdef INQUISITOR
    if( P.Dim() != 3 ){
        InquisitorOK = 0;
        return;
    }
#endif

    Ro = P(0); Zo = P(1); No = P(2);
}

// JumpNWeylSol class
inline JumpNWeylSol::JumpNWeylSol( WeylPhi* p, const real ro,
                                   const real zo, const real no ):
    NWeylSol( p, ro, zo, no ){

inline void JumpNWeylSol::EvaluateNuIn ( const real r, const real z ){
    NWeylSol::EvaluateNuIn( r, z );
    Ro = r; Zo = z; No = nu;
}

// AssintNWeylSol class
inline AssintNWeylSol::AssintNWeylSol( WeylPhi* p, IntegrationPath* I ):
    JumpNWeylSol( p ), InPath( I ){

inline AssintNWeylSol::AssintNWeylSol( WeylPhi* p, IntegrationPath* I,
                                       const real s, const real f, const uinteger n ):
    JumpNWeylSol( p ), InPath( I ), factor( f ), ndiv( n ),
    step( pow( s, 1.0/f ) / (real)n ){

inline void AssintNWeylSol::SetParameters( const NReal& P ){
#ifdef INQUISITOR
    if( P.Dim() != 3 ){
        InquisitorOK = 0;
        return;
    }
#endif

    factor = P(1);
    ndiv = (integer)P(2);
    step = pow( P(0), 1.0/factor ) / (real)ndiv;
}

inline void AssintNWeylSol::Destructor(){
    JumpNWeylSol::Destructor();
    InPath->Destructor();
    delete InPath;
}
#endif

```

Arquivo nweylsol.cpp

```

/*-----
Numeric Weyl Project
Definition of virtual classes for implement Numeric Weyl Solutions
by Luis Alberto D'Afonseca - 27 October 1998
-----*/
#include "nweylsol.h"
#include "fspecial.h"

void NWeylSol::EvaluateNuIn( const real r, const real z ){
    real I = 0, A = ( r - Ro )/2.0, B = ( z - Zo )/2.0;

    for( int ii = 0; ii < GaussNumPointers; ii++ ){
        real R = Ro + A * GaussI_pi[ii],
            Z = Zo + B * GaussI_pi[ii];
        EvaluatePhiIn ( R, Z );
        I += ( A * drnu + B * dznu ) * GaussW[ii];
    }
    nu = No + I;
}

void AssintNWeylSol::EvaluateNuIn ( const real r, const real z ){
    Ro = r; Zo = z; No = 0;
    real R, Z;

    InPath->StartPath( r, z );
    for( int ii = 0; ii < ndiv; ii++ ){
        InPath->EvaluateIn( pow( step*(real)ii, factor ), R, Z );
        JumpNWeylSol::EvaluateNuIn( R, Z );
    }

    // --- Gauss Laguerre Method ---
    InPath->EvaluateIn( pow( step*(real)(ii-1), factor ) + 1, R, Z );

    real I = 0, A = R - Ro, B = Z - Zo;

```

```

for( ii = 0; ii < LaguerreNumPointers; ii++ ){
  R = Ro + A * LaguerreX[ii];
  Z = Zo + B * LaguerreX[ii];

  EvaluatePhiLn ( R, Z );
  I += ( A * drnu + B * dznu ) * LaguerreWxExp[ii];
}
nu = I - nu;
}

```

Arquivo appell.h

```

/*-----
Numeric Weyl Project
Definition of Appell Solution
by Luis Alberto D'Afonseca - 30 October 1998
-----*/
#ifndef APPELL_SOLUTION_INCLUDED
#define APPELL_SOLUTION_INCLUDED
#include "weylfunc.h"

class AppellPhi: public WeylPhi{
public:
  AppellPhi(): m( 0 ), zo( 0 ), a2( 0 ) {};
  AppellPhi( const real M, const real Zo, const real A );
  ~AppellPhi(){};

  int NumParameters(){ return 3; };
  void SetParameters( const NReal& ); // M, Zo and A
  void EvaluateLn( const real r, const real z,
                  real& phi, real& drphi, real& dzphi );

protected: real m, zo, a2;
};

inline AppellPhi::AppellPhi( const real M, const real Zo, const real A ):
  m ( M / M_SQRT2 ), zo( Zo ), a2( A * A ) {};

inline void AppellPhi::SetParameters( const NReal& P ){
  #ifdef INQUISITOR
  if( P.Dim() != 3 ) Inquisition;
  #endif
  m = P(0) / M_SQRT2; zo = P(1); a2 = P(2) * P(2);
}

inline void AppellPhi::EvaluateLn( const real r, const real z,
  real& phi, real& drphi, real& dzphi ){
  real x2 = r * r, Dz = z - zo, Dz2 = Dz * Dz,
  R = r2 + Dz2 - a2, B = R * R + 4.0 * a2 * Dz2,
  sB = sqrt( B ), A = sB + R,
  mAB = m / ( 2.0 * pow( A, 0.5 ) * pow( B, 1.5 ) ),
  Ar = 2.0 * r * ( 1.0 + R / sB ),
  Az = 2.0 * Dz * ( 1.0 + ( x2 + Dz2 + a2 ) / sB ),
  Br = 4.0 * r * R,
  Bz = 4.0 * Dz * ( 2.0 * a2 + R );
  phi = - m * sqrt( A / B );
  drphi = mAB * ( A * Ar - B * Br );
  dzphi = mAB * ( A * Az - B * Bz );
}
#endif

```

Arquivo bkhole.h

```

/*-----
Numeric Weyl Project
Definition of Schwarzschild Solution
by Luis Alberto D'Afonseca - 30 October 1998
-----*/
#ifndef SCHWARZSCHILD_SOLUTION_INCLUDED
#define SCHWARZSCHILD_SOLUTION_INCLUDED
#include "weylfunc.h"

class SchwarzschildNu: public WeylNu{
public:
  SchwarzschildNu(){};
  SchwarzschildNu( const real M, const real Zo );
  ~SchwarzschildNu(){};

  int NumParameters(){ return 2; };
  void SetParameters( const NReal& ); // M and Zo
  void EvaluateLn( const real r, const real z, real& nu );

protected: real m1, zp, zm;
};

class SchwarzschildPhi: public WeylPhi{

```

```

public:
  SchwarzschildPhi(): m2( 0 ), m4( 0 ), zp( 0 ), zm( 0 ){};
  SchwarzschildPhi( const real M, const real Zo );
  ~SchwarzschildPhi(){};

  int NumParameters(){ return 2; };
  void SetParameters( const NReal& ); // M and Zo
  WeylNu* ExactNu() { return new SchwarzschildNu( m, zo ); };
  void EvaluateIn( const real r, const real z,
                  real& phi, real& drphi, real& dzphi );

protected: real m, zo, m2, m4, zp, zm;
};

inline SchwarzschildPhi::SchwarzschildPhi( const real M, const real Zo ):
  m( M ), zo( Zo ), m2( 2.0 * M ), m4( 4.0 * M * M ),
  zp( Zo - M ), zm( Zo + M ) {};

inline void SchwarzschildPhi::SetParameters( const NReal& P ){
  #ifdef INQUISITOR
    if( P.Dim() != 2 ) Inquisition;
  #endif

  real m = P(0), zo = P(1);
  m2 = 2.0 * m; m4 = 4.0 * m * m; zp = zo - m; zm = zo + m;
}

inline void SchwarzschildPhi::EvaluateIn( const real r, const real z,
                                          real& phi, real& drphi, real& dzphi ){
  real r2 = r*r, Dzp = z - zp, Dzm = z - zm,
  Rp = sqrt( r2 + Dzp*Dzp ),
  Rm = sqrt( r2 + Dzm*Dzm ),
  A = m2 / ( pow( Rm + Rp, 2.0 ) - m4 );
  phi = 0.5 * log( ( Rm + Rp - m2 ) / ( Rm + Rp + m2 ) );
  drphi = A * r * ( 1.0/Rp + 1.0/Rm );
  dzphi = A * ( Dzp/Rp + Dzm/Rm );
}

// SchwarzschildNu inline functions
inline SchwarzschildNu::SchwarzschildNu( const real M, const real Zo ):
  m4( 4.0*M*M ), zp( Zo - M ), zm( Zo + M ) {}

inline void SchwarzschildNu::SetParameters( const NReal& P ){
  #ifdef INQUISITOR
    if( P.Dim() != 2 ) Inquisition;
  #endif

  real M = P(0), Zo = P(1);
  m4 = 4.0 * M * M; zp = Zo - M; zm = Zo + M;
}

inline void SchwarzschildNu::EvaluateIn( const real r, const real z, real& nu ){
  real r2 = r*r, Dzp = z - zp, Dzm = z - zm,
  Rp = sqrt( r2 + Dzp*Dzp ),
  Rm = sqrt( r2 + Dzm*Dzm );
  nu = 0.5 * log( ( pow( Rm + Rp, 2.0 ) - m4 ) / ( 4.0 * Rm * Rp ) );
}
#endif

```

Arquivo curzon.h

```

/*-----
  Numeric Weyl Project
  Definition of Curzon Solution
  by Luis Alberto D'Afonseca - 28 October 1998
  -----*/
#ifdef CURZON_SOLUTION_INCLUDED
#define CURZON_SOLUTION_INCLUDED
#include "weylfunc.h"

class CurzonNu: public WeylNu{
public:
  CurzonNu(){};
  CurzonNu( const real m, const real zo ): M2( m*m ), Zo( zo ) {};
  ~CurzonNu(){};

  int NumParameters(){ return 2; };
  void SetParameters( const NReal& ); // M and Zo
  void EvaluateIn( const real r, const real z, real& nu );

protected: real M2, Zo;
};

class CurzonPhi: public WeylPhi{
public:
  CurzonPhi(): M( 0 ), Zo( 0 ) {};
  CurzonPhi( const real m, const real zo ): M( m ), Zo( zo ) {};

```

```

~CurzonPhi(){};

int NumParameters(){ return 2; };
void SetParameters( const NReal& ); // M and Zo
WeylNu* ExactNu() { return new CurzonNu( M, Zo ); };
void EvaluateIn( const real r, const real z,
                real& phi, real& drphi, real& dzphi );

protected: real M, Zo;
};

inline void CurzonPhi::SetParameters( const NReal& P ){
#ifdef INQUISITOR
    if( P.Dim() != 2 ) Inquisition;
#endif
    M = P(0); Zo = P(1);
}

inline void CurzonPhi::EvaluateIn( const real r, const real z,
                                real& phi, real& drphi, real& dzphi ){
    real Dz = z - Zo, R2 = r*r + Dz*Dz,
        Aux = M * pow( R2, -1.5 );

    phi = -M / sqrt( R2 ); drphi = r * Aux; dzphi = Dz * Aux;
}

// CurzonNu inline functions
inline void CurzonNu::SetParameters( const NReal& P ){
#ifdef INQUISITOR
    if( P.Dim() != 2 ) Inquisition;
#endif
    M2 = P(0)*P(0); Zo = P(1);
}

inline void CurzonNu::EvaluateIn( const real r, const real z, real& nu ){
    real Dz = z - Zo, r2 = r*r, R2 = r2 + Dz*Dz;
    nu = - ( M2*r2 ) / ( 2.0*R2*R2 );
}
#endif

```

Arquivo dustdisk.h

```

/*-----
Numeric Weyl Project
Definition of DustDisk Solution
by Luis Alberto D'Afonseca - 28 October 1998
-----*/
#ifdef DUSTDISK_SOLUTION_INCLUDED
#define DUSTDISK_SOLUTION_INCLUDED
#include "weylfunc.h"

class DustDiskNu: public WeylNu{
public:
    DustDiskNu(){};
    DustDiskNu( const real zz, const real aa ): zo(zz), a2(aa*aa/16.0) {};
    ~DustDiskNu(){};

    int NumParameters(){ return 2; };
    void SetParameters( const NReal& );
    void EvaluateIn( const real r, const real z, real& nu );

protected: real zo, a2;
};

class DustDiskPhi: public WeylPhi{
public:
    DustDiskPhi(): zo( 0 ), a( 0 ) {};
    DustDiskPhi( const real zz, const real aa ): zo( zz ), a( aa / 2.0 ) {};
    ~DustDiskPhi(){};

    int NumParameters(){ return 2; };
    void SetParameters( const NReal& ); // zo and a
    WeylNu* ExactNu() { return new DustDiskNu( zo, a ); };
    void EvaluateIn( const real r, const real z,
                    real& phi, real& drphi, real& dzphi );

protected: real zo, a;
};

inline void DustDiskPhi::SetParameters( const NReal& P ){
#ifdef INQUISITOR
    if( P.Dim() != 2 ) Inquisition;
#endif

```

```

zo = P(0); a = P(1) / 2.0;
}

inline void DustDiskPhi::EvaluateIn( const real r, const real z,
                                     real& phi, real& drphi, real& dzphi ){
  real r2 = r * r, Dz = z - zo, Dz2 = Dz * Dz,
  aux = (real)(z < zo)? 1: -1) * a * pow( r2 + Dz2, -2.5 );

  phi = aux * Dz * ( r2 + Dz2 );
  drphi = aux * (-3.0) * r * Dz;
  dzphi = aux * ( r2 - 2.0 * Dz2 );
}

// DustDiskNu inline functions
inline void DustDiskNu::SetParameters( const NReal& P ){
#ifdef INQUISITOR
  if( P.Dim() != 2 ) Inquisition;
#endif

zo = P(0); a2 = P(1)*P(1) / 16.0;
}

inline void DustDiskNu::EvaluateIn( const real r, const real z, real& nu ){
  real r2 = r * r, Dz2 = pow( z - zo, 2 );
  nu = ( a2 * r2*r2 / pow(Dz2 + r2, 4) ) * ( 1.0 - 8.0 * Dz2 / r2 );
}
#endif

```

Arquivo morgan.h

```

/*-----
Numeric Weyl Project
Definition of Morgan-Morgan Disk
by Luis Alberto D'Afonseca - 31 October 1998
-----*/

#ifdef MORGAN_SOLUTION_INCLUDED
#define MORGAN_SOLUTION_INCLUDED
#include "weylfunc.h"

class MorganPhi: public WeylPhi{
public:
  MorganPhi(): zo( 0 ), a2( 0 ), aa( 0 ), mxa( 0 ), mda( 0 ) {};
  MorganPhi( const real M, const real Zo, const real A );
  ~MorganPhi(){};

  int NumParameters(){ return 3; };
  void SetParameters( const NReal& ); // M, Zo and A
  void EvaluateIn( const real r, const real z,
                  real& phi, real& drphi, real& dzphi );

protected: real zo, a2, aa, mxa, mda;
};

inline MorganPhi::MorganPhi( const real M, const real Zo, const real A ):
  zo(Zo), a2(A*A), aa(M_SQRT2*fabs(A)),mxa(M*A), mda(-M/A) {};

inline void MorganPhi::SetParameters( const NReal& P ){
#ifdef INQUISITOR
  if( P.Dim() != 3 ) Inquisition;
#endif

  real A = P(2);

  zo = P(1); a2 = A*A; aa = M_SQRT2*fabs(A); mxa = P(0)*A; mda = -P(0)/A;
}

inline void MorganPhi::EvaluateIn( const real r, const real z,
                                   real& phi, real& drphi, real& dzphi ){
  real r2 = r * r, Dz = z - zo, Dz2 = Dz * Dz,
  R = r2 + Dz2 - a2,
  sB = sqrt( R*R + 4.0 * Dz2 * a2 ),
  A = R + sB,
  Ar = 2.0 * r * ( 1.0 + R / sB ),
  Az = 2.0 * Dz * ( 1.0 + ( r2 + Dz2 + a2 ) / sB ),
  mA = mxa / ( sqrt( A ) * aa * ( 2.0 * a2 + A ));

  phi = mda * atan( sqrt( ( 2.0 * a2 ) / A ));
  drphi = Ar * mA; dzphi = Az * mA;
}
#endif

```

Arquivo realring.h

```

/*-----
Numeric Weyl Project
Definition of True Ring
-----*/

```

```

by Luis Alberto D'Afonseca - 31 October 1998
-----*/
#ifndef REAL_RING_SOLUTION_INCLUDED
#define REAL_RING_SOLUTION_INCLUDED
#include "weylfunc.h"
#include "fspecial.h"

class RealRingPhi: public WeylPhi{
public:
    RealRingPhi() {};
    RealRingPhi( const real M, const real Zo, const real A );
    ~RealRingPhi(){};

    int NumParameters(){ return 3; };
    void SetParameters( const NReal& ); // M, Zo and A
    void EvaluateIn( const real r, const real z,
                    real& phi, real& drphi, real& dzphi );

protected: real a, zo, am4, a4, a2;
};

inline RealRingPhi::RealRingPhi( const real M, const real Zo, const real A ):
    a ( A ), zo ( Zo ), am4( 4.0*A*M ), a4 ( 4.0*A ), a2 ( 2.0*A ) {};

inline void RealRingPhi::SetParameters( const NReal& P ){
    #ifdef INQUISITOR
        if( P.Dim() != 3 ) Inquisition;
    #endif
    a = P(2); zo = P(1); am4 = 4.0*a*P(0); a4 = 4.0*a; a2 = 2.0*a;
}

inline void RealRingPhi::EvaluateIn( const real r, const real z,
                                    real& phi, real& drphi, real& dzphi ){
    real a_r = a + r,
        r2 = 2.0 * r,
        z_zo = z - zo,
        R = a_r*a_r + z_zo*z_zo,
        sR = sqrt( R ),
        R_3 = am4 * pow( R, -1.5 ),
        A = - am4 / sR,
        Ar = a_r * R_3,
        Az = z_zo * R_3,
        B = sqrt( a4 * r ) / sR,
        BR2 = B * R * R,
        Br = a2 * ( z_zo*z_zo + a*a - r*r ) / BR2,
        Bz = - a4 * r * z_zo / BR2,
        K_B = EllipticK( B ),
        DK_B = A * DkEllipticK( B );

    phi = A * K_B;
    drphi = Ar * K_B + DK_B * Br;
    dzphi = Az * K_B + DK_B * Bz;
}
#endif

```

Arquivo ring.h

```

/*-----*/
Numeric Weyl Project
Definition of True Ring
by Luis Alberto D'Afonseca - 31 October 1998
-----*/
#ifndef RING_SOLUTION_INCLUDED
#define RING_SOLUTION_INCLUDED
#include "weylfunc.h"

class RingPhi: public WeylPhi{
public:
    RingPhi(): zo( 0 ), ms2( 0 ), ma( 0 ), a2( 0 ), a22( 0 ), sa( 0 ) {};
    RingPhi( const real M, const real Zo, const real A );
    ~RingPhi(){};

    int NumParameters(){ return 3; };
    void SetParameters( const NReal& ); // M, Zo and A
    void EvaluateIn( const real r, const real z,
                    real& phi, real& drphi, real& dzphi );

protected: real zo, ms2, ma, a2, a22, sa;
};

//-----*/
inline RingPhi::RingPhi( const real M, const real Zo, const real A ):
    zo ( Zo ),
    ms2( M / ( 2.0 * M_SQRT2 ) ),
    ma ( M / ( 2.0 * A ) ),
    a2 ( A * A ),

```

```

a22( 2.0 * a2 ),
sa ( A / fabs(A) )};

inline void RingPhi::SetParameters( const NReal& P ){
#ifdef INQUISITOR
    if( P.Dim() != 3 ) Inquisition;
#endif

    real M = P(0), A = P(2);

    zo = P(1);
    ms2 = M / ( 2.0 * M_SQRT2 );
    ma = M / ( 2.0 * A );
    a2 = A * A;
    a22 = 2.0 * a2;
    sa = A / fabs(A);
}

inline void RingPhi::EvaluateIn( const real r, const real z,
                                real& phi, real& drphi, real& dzphi ){
    real r2 = r * r,
        Dz = z - zo,
        Dz2 = Dz * Dz,
        R = r2 + Dz2 - a2,
        B = R * R + 2.0 * a22 * Dz2,
        sB = sqrt( B ),
        A = sB + R,
        sA = sqrt( A ),
        Ar = r * ( 1.0 + R / sB ), // = Ar / 2.0
        Az = Dz * ( 1.0 + (r2 + Dz2 + a2) / sB ), // = Az / 2.0
        Br = r * R, // = Br / 2.0
        Bz = Dz * ( a22 + R ), // = Bz / 2.0
        mA = ( 2.0 * sa / ( a22 + A ) - 1.0 / sB ) / sA,
        mB = 2.0 * A * pow( B, -1.5 ) / sA;

    phi = -ms2 * sA / sB - ma * atan( sqrt( a22 ) / sA );
    drphi = ms2 * ( Ar * mA + Br * mB );
    dzphi = ms2 * ( Az * mA + Bz * mB );
}
#endif

```

Arquivo rod.h

```

/*-----
Numeric Weyl Project
Definition of Schwarzschild Solution
by Luis Alberto D'Afonseca - 30 October 1998
-----*/
#ifdef ROD_SOLUTION_INCLUDED
#define ROD_SOLUTION_INCLUDED
#include "weylfunc.h"

class RodNu: public WeylNu{
public:
    RodNu(){};
    RodNu( const real M, const real Zo, const real L );
    ~RodNu(){};

    int NumParameters(){ return 3; };
    void SetParameters( const NReal& ); // M, Zo and L
    void EvaluateIn( const real r, const real z, real& nu );

protected: real ml, l4, zp, zm;
};

class RodPhi: public WeylPhi{
public:
    RodPhi(){};
    RodPhi( const real M, const real Zo, const real L );
    ~RodPhi(){};

    int NumParameters(){ return 3; };
    void SetParameters( const NReal& ); // M, Zo and L
    WeylNu* ExactNu() { return new RodNu( m, zo, l ); };
    void EvaluateIn( const real r, const real z,
                    real& phi, real& drphi, real& dzphi );

protected: real m, l, zo, ml, l2, m2, zp, zm;
};

//-----
inline RodPhi::RodPhi( const real M, const real Zo, const real L ):
    m( M ), zo( Zo ), l( L ), ml( M / ( 2.0 * L ) ), l2( 2.0 * L ),
    m2( 2.0 * M ), zp( Zo - L ), zm( Zo + L ) {};

inline void RodPhi::SetParameters( const NReal& P ){
#ifdef INQUISITOR

```

```

    if( P.Dim() != 2 ) Inquisition;
#endif

    m = P(0); zo = P(1); l = P(2); ml = m / ( 2.0 * l );
    l2 = 2.0 * l; m2 = 2.0 * m; zp = zo - l; zm = zo + l;
}

inline void RodPhi::EvaluateIn( const real r, const real z,
                               real& phi, real& drphi, real& dzphi ){
    real r2 = r * r, Dzp = z - zp, Dzm = z - zm,
    Rp = sqrt( r2 + Dzp*Dzp ), Rm = sqrt( r2 + Dzm*Dzm ),
    A = Rm + Rp - l2, B = Rm + Rp + l2,
    C = m2 / ( A * B );

    phi = ml * log(A/B);
    drphi = C * r * ( 1.0/Rp + 1.0/Rm );
    dzphi = C * ( Dzp/Rp + Dzm/Rm );
}

// RodNu inline functions
inline RodNu::RodNu( const real M, const real Zo, const real L ){
    ml( M*M / ( 2.0 * L*L ) ), l4( 4.0 * L * L ),
    zp( Zo - L ), zm( Zo + L ) {}

inline void RodNu::SetParameters( const NReal& P ){
    #ifdef INQUISITOR
    if( P.Dim() != 2 ) Inquisition;
    #endif

    ml = P(0) * P(0) / ( 2.0 * P(2) * P(2) );
    l4 = 4.0 * P(2) * P(2);
    zp = P(1) - P(2); zm = P(1) + P(2);
}

inline void RodNu::EvaluateIn( const real r, const real z, real& nu ){
    real r2 = r * r, Dzp = z - zp, Dzm = z - zm,
    Rp = sqrt( r2 + Dzp*Dzp ), Rm = sqrt( r2 + Dzm*Dzm );

    nu = ml * log( ( pow( Rm + Rp, 2.0 ) - l4 ) / ( 4.0 * Rm * Rp ) );
}
#endif

```

Arquivo initphi.cpp

```

/*-----
Numeric Weyl Project
Definition of Initializator function for WeylPhi Class
by Luis Alberto D'Afonseca - 27 October 1998
-----*/

#include <fstream.h>
#include <string.h>
#include "weylfunc.h"
#include "doubphi.h"
#include "superphi.h"
#include "appell.h"
#include "bikhole.h"
#include "curzon.h"
#include "dustdisk.h"
#include "morgan.h"
#include "realring.h"
#include "ring.h"
#include "rod.h"
#define SwitchClass(name,class) if(strcmp(str,name)==0) return new class;

WeylPhi* SetPhiClass( char *str ){
    SwitchClass( "Appell", AppellPhi );
    SwitchClass( "Schwarzschild", SchwarzschildPhi );
    SwitchClass( "Curzon", CurzonPhi );
    SwitchClass( "DustDisk", DustDiskPhi );
    SwitchClass( "Morgan", MorganPhi );
    SwitchClass( "WeylRing", RealRingPhi );
    SwitchClass( "Ring", RingPhi );
    SwitchClass( "Rod", RodPhi );
    throw str;
}

void SetParameters( istream& is, WeylPhi* phi ){
    if( phi->NumParameters() != 0 ){
        NReal P( phi->NumParameters() );
        is >> P; cout << " " << P << endl;
        phi->SetParameters( P );
    }
}

WeylPhi* InitPhi( istream& is ){
    WeylPhi* phi;
    char str[50];

```

```

try{
  is >> str; cout << str;

  // ---- Double Phi ----
  if( strcmp( str, "DoublePhi" ) == 0 ){
    is >> str; cout << endl << str;
    WeylPhi *first = SetPhiClass ( str );
    SetParameters ( is, first );

    is >> str; cout << str;
    WeylPhi *second = SetPhiClass ( str );
    SetParameters ( is, second );
    phi = new DoublePhi( first, second );
  }else{ // ---- Superposed Phi ----
    if( strcmp( str, "SuperPhi" ) == 0 ){
      integer nn;

      is >> nn; cout << " " << nn << endl;
      phi = new SuperPhi( nn );
      for( integer ii = 0; ii < nn; ii++){
        is >> str; cout << str;
        WeylPhi *phi_aux = SetPhiClass ( str );
        SetParameters ( is, phi_aux );
        ((SuperPhi*)phi)->SetPhi( ii, phi_aux );
      }
    }else{ // ---- Single Phi ----
      phi = SetPhiClass( str );
      SetParameters ( is, phi );
    }
  }
}catch( char* str_error ){
  cout << "Error:" << str_error << " is wrong" << endl;
  phi = NULL;
}catch( ... ){
  cout << "Error: Initialization phi error" << endl;
  phi = NULL;
}
return phi;
}

```

Arquivo initpath.cpp

```

/*-----
Numeric Weyl Project
Definition of Initializer function for IntegrationPath Class
by Luis Alberto D'Afonseca - 21 November 1998
-----*/

#include <fstream.h>
#include <string.h>
#include "nweylsol.h"
#define SwitchClass(name,class) if(strcmp(str,name)==0) return new class;

IntegrationPath* SetPathClass( char *str ){
  SwitchClass( "Diagonal", DiagonalPath );
  SwitchClass( "AwayOf", AwayOfPath );
  throw str;
}

void SetParameters( istream& is, IntegrationPath* path ){
  if( path->NumParameters() != 0 ){
    NReal P( path->NumParameters() );
    is >> P; cout << " " << P << endl;
    path->SetParameters( P );
  }
}

IntegrationPath* InitPath( istream& is ){
  IntegrationPath* path;
  char str[50];
  try{
    is >> str; cout << str;
    path = SetPathClass( str );
    SetParameters ( is, path );
  }catch( char* str_error ){
    cout << "Error:" << str_error << " is wrong" << endl;
    path = NULL;
  }catch( ... ){
    cout << "Error: Initialization path error" << endl;
    path = NULL;
  }
  return path;
}

```

Calculando ν Numericamente

Arquivo points.cpp

```

/*-----\\
Numeric Weyl Program
developed by Luis Alberto D'Afonseca - DMA / IMECC / UNICAMP
September, 1998
//-----*/
#include <fstream.h>
#include "CTable.h"

int ReadPoints ( char *filename, ReColTable& Points ){
    int ok = 1;
    ifstream in_file( filename, ios::nocreate || ios::in );

    if( in_file.bad() ){
        ok = 0;
        cout << "Error 02: File " << filename << " doesn't exist" << endl;
    }else{
        // --- Reading Number of Points ---
        uinteger lin, col;
        in_file >> lin >> col;

        if( lin < 1 || col != 2 ){
            ok = 0;
            cout << "Error 03: File " << filename << " is bad" << endl;
        }else{
            cout << "Number of points = " << lin << endl << endl;
            Points.Reset( lin, 2 );

            if( !Points.good() ){
                ok = 0;
                cout << "Error 06: Not enough memory" << endl;
            }else{ // --- Reading the Points ---
                in_file >> Points;
            }
        }
    }
    in_file.close();
    return ok;
}

```

Arquivo param.cpp

```

/*-----\\
Numeric Weyl Program
developed by Luis Alberto D'Afonseca - DMA / IMECC / UNICAMP
September, 1998
//-----*/
#include <fstream.h>
#include <string.h>
#include "nweylsol.h"

WeylSol* Parameters ( char *filename ){
    WeylSol *solution = NULL;
    ifstream op_file( filename, ios::nocreate || ios::in );

    if( op_file.bad() ){
        cout << "Error 10: File " << filename << " doesn't exist" << endl;
    }else{
        WeylPhi *phi = InitPhi ( op_file );
        IntegrationPath *path = InitPath( op_file );

        if( phi != NULL && path != NULL ){
            char str[50];
            op_file >> str;

            if( strcmp( "IntegralParameters", str ) == 0 ){
                real Size, Factor, Ndiv;
                op_file >> Size >> Factor >> Ndiv;
                solution = new AssintWeylSol( phi, path, Size, Factor, Ndiv );
            }else{
                cout << "Error 11: Integral Parameters not found" << endl;
            }
        }
    }
    op_file.close();
    return solution;
}

```

Arquivo nnu.cpp

```

/*-----\\
Numeric Weyl Program - September, 1998

```

```

developed by Luis Alberto D'Afonseca - DMA / IMECC / UNICAMP
//-----*/
#include <fstream.h>
#include <time.h>
#include <dir.h>
#include "Table.h"
#include "nweylsol.h"
#define SendBeep

int Wellcome ( int argc );
WeylSol* Parameters ( char *filename );
int ReadPoints ( char *filename, ReColTable& Points );
void SaveValues ( ofstream& out_file, ReTable& Values );
void Echo ( uinteger );

int main( int argc, char *argv[] ){
  if( !Wellcome( argc ) ) return -1;
  WeylSol *solution = Parameters( argv[1] );
  if( solution == NULL ) return -1;
  ReColTable Points;

  if( !ReadPoints( argv[2], Points ) ){
    solution->Destructor();
    return -1;
  }

  fblk temp;
  if ( findfirst(argv[3],&temp,0) == 0 ){
    cout << "File " << argv[3] << " already exist. Overwrite (y/n)? ";
    char cc; cin >> cc;

    if ( cc != 'y' && cc != 'Y' ){
      solution->Destructor();
      return -1;
    }
  }

  ofstream out_file( argv[3] );
  if( out_file.bad() ){
    cout << "Erro 04: File " << argv[3] << " can't be created" << endl;
  }else{
    ReTable Values( Points.GetLines(), 2 );
    if( !Values.good() ){
      cout << "Error 06: Not enough memory" << endl;
    }else{
      time_t timerI = time(NULL);
      struct tm *tblock = localtime(&timerI);
      cout << "Begin - " << asctime(tblock);

      solution->EvaluateForAll( &Points, &Values, 1000, Echo );

      time_t timerF = time(NULL);
      tblock = localtime(&timerF);
      cout << "End - " << asctime(tblock) << "Time elapsed = "
        << (timerF - timerI) << " seconds" << endl;

      SaveValues( out_file, Values );
      cout << endl << "Numeric Weyl concluded " << endl << endl;
      #ifdef SendBeep
        cout << '\a';
      #endif
    }
  }
  out_file.close();
  solution->Destructor();
  delete solution;
  return 0;
}

int Wellcome( int argc ){
  int ok = 1;
  cout << endl << "Numeric Weyl Suit - Program Numeric Nu"
    << endl << "developed by D'Afonseca L.A, DMA/IMECC/UNICAMP" << endl << endl;

  // --- Test the Number of Arguments ---
  if ( argc != 4 ){
    ok = 0;
    cout << endl << "Error 01: The number of parameters is wrong"
      << endl << "This program waits:"
      << endl << " the name of the file with solution parameters"
      << endl << " the name of the file with list of (x,z) points and"
      << endl << " the file name for save output data" << endl << endl;
  }
  return ok;
}

void SaveValues ( ofstream& out_file, ITable<real>& Values ){
  out_file << Values.GetLines() << endl << 2 << endl << endl;
}

```

```

    out_file.setf( ios::right | ios::scientific );
    out_file << Values;
}

void Echo ( uinteger ii ){
    cout << " " << ii << " points done" << endl;
}

```

Resolução de EDOS

Arquivo edosol.h

```

/*-----
Numeric Weyl Project
Definition of EDO initial values Solver
by Luis Alberto D'Afonseca
-----*/

#ifndef EDO_SOLVER_INCLUDED
#define EDO_SOLVER_INCLUDED
#include "function.h"
#include "table.h"

class EDOSolver: public RNFunction{ // Method of Euler
public:
    EDOSolver( RNFunction* );
    ~EDOSolver(){};

    void Reset( RNFunction* );
    void SetStep( const real );
    real GetStep() { return step; };
    void StartPoint( const real t, const NReal& P );

    virtual void Run( NReal& );
    virtual void Run( const uinteger nn, NReal& );
    virtual void Run( const uinteger nn, ReTable *Hist );

    int Dim      () { return F->DimOutput(); };
    int DimInput () { return F->DimOutput(); };
    int DimOutput () { return F->DimOutput(); };

protected:
    virtual void EvaluateIn ( const NReal&, NReal&, const real );
    RNFunction *F;

    real time, step;
    NReal point, F_aux;
};

inline EDOSolver::EDOSolver( RNFunction* f ): F( f ),
#ifdef INQUISITOR
    point(),
    F_aux()
{
    if( F == NULL ) Inquisition;
    point.Reset( F->DimOutput() );
    F_aux.Reset( F->DimOutput() );
}
#else
    point( F->DimOutput() ),
    F_aux( F->DimOutput() )
{}
#endif

inline void EDOSolver::Reset( RNFunction* f ){
#ifdef INQUISITOR
    if( f == NULL ) Inquisition;
#endif
    F = f;
    point = Dim();
    F_aux = Dim();
}

inline void EDOSolver::StartPoint( const real t, const NReal& P ){
    time = t; point = P;
}

inline void EDOSolver::SetStep( const real newstep ){
#ifdef INQUISITOR
    if( newstep <= 0 ) Inquisition;
    else
#endif
    step = newstep;
}

inline void EDOSolver::EvaluateIn ( const NReal& P, NReal& newP, const real t ){
    F->EvaluateIn( P, newP, t ); // newP = P' temporarily
}

```

```

newP = P + step * newP;      // newP = P_n+1
}

inline void EDOSolver::Run( NReal& newpoint ){
  EvaluateIn( point, newpoint, time );
  point = newpoint;
  time += step;
}
#endif

```

Arquivo runge4.h

```

/*-----
  Numeric Weyl Project
  Definition of EDO initial values Solver
  Fourth Order Ringe-Kutta Method
  by Luis Alberto D'Afonseca
  -----*/

#ifndef RUNGE4_INCLUDED
#define RUNGE4_INCLUDED
#include "edosol.h"

class Runge4: public EDOSolver{
public:
  Runge4( RFunction* f ): EDOSolver( f ){
    ~Runge4();
};

protected:
  void EvaluateIn ( const NReal&, NReal&, const real );
};
#endif

```

Arquivo edosol.cpp

```

/*-----
  Numeric Weyl Project
  Definition of EDO initial values Solver
  by Luis Alberto D'Afonseca
  -----*/

#include "runge4.h" // runge4.h already include edosol.h

void EDOSolver::Run( const uinteger nn, NReal& newP ){
  for( uinteger ii = 0; ii < nn; ii++ ){
    EvaluateIn( point, newP, time );
    point = newP;
    time += step;
  }
}

void EDOSolver::Run( const uinteger nn, ReTable *Hist ){
  #ifdef INQUISITOR
    if( Hist->GetLines() != nn || Hist->GetColumns() != 1 ) Inquisition;
  #endif

  NReal aux( Dim() );
  uinteger dd = Dim();
  for( uinteger ii = 0; ii < nn; ii++ ){
    EvaluateIn( point, aux, time );
    for( uinteger jj = 0; jj < dd; jj++ ) (*Hist)(ii,jj) = aux(jj);
    point = aux;
    time += step;
  }
}

void Runge4::EvaluateIn ( const NReal& P1, NReal& P2, const real t ){
  #ifdef INQUISITOR
    if( P1.Dim() != P2.Dim() || P1.Dim() == 0 ) Inquisition;
  #endif
  real aux = t + step / 2.0;
  F->EvaluateIn( P1, F_aux, t );
  NReal K1 = step * F_aux;
  F->EvaluateIn( P1 + K1 * 0.5, F_aux, aux );
  NReal K2 = step * F_aux;
  F->EvaluateIn( P1 + K2 * 0.5, F_aux, aux );
  NReal K3 = step * F_aux;
  F->EvaluateIn( P1 + K3, F_aux, t + step );
  NReal K4 = step * F_aux;
  P2 = P1 + ( 1.0/6.0 )*( K1 + 2.0*K2 + 2.0*K3 + K4 );
}

```

Cálculo das Geodésicas

Arquivo weylgeo.h

```

/*-----

```

```

Numeric Weyl Project
Definition of Geodesic Equations for Weyl Spaces
by Luis Alberto D'Afonseca
begin - 05 November 1998
-----*/
#ifndef WEYL_GEODESICS_INCLUDED
#define WEYL_GEODESICS_INCLUDED
#include "function.h"
#include "weylsol.h"

class WeylGeodesic: public RFunction{
public:
    WeylGeodesic( WeylSol* );
    ~WeylGeodesic(){};

    int NumParameters(){ return 1; };
    void SetParameters( const NReal& P ){ alpha = P(0); };
    void TimeGeodesic(){ alpha = 1; };
    void NullGeodesic(){ alpha = 0; };
    void SetTRZPMode (); // P(i) = { t, r, z, p, t', r', z', p' }
    void SetTRZMode (); // P(i) = { t, r, z, t', r', z' }
    void SetRZMode (); // P(i) = { r, z, r', z' }
    void EvaluateIn ( const NReal&, NReal&, const real );
    int DimInput ();{ return dim; };
    int DimOutput ();{ return dim; };
    void Destructor();

protected:
    void Evaluate_TRZP ( const NReal&, NReal& );
    void Evaluate_TRZ ( const NReal&, NReal& );
    void Evaluate_RZ ( const NReal&, NReal& );
    void (WeylGeodesic::*EvaluateFunction)( const NReal&, NReal& );

    WeylSol* W;
    real alpha;
    int dim;
};

// inline functions
inline WeylGeodesic::WeylGeodesic( WeylSol* w ):
    EvaluateFunction( &WeylGeodesic::Evaluate_TRZP ),
    W ( w ), alpha ( 1 ), dim ( 8 ) {}

inline void WeylGeodesic::EvaluateIn ( const NReal& P1, NReal& P2, const real ){
    (this->*EvaluateFunction)( P1, P2 );}

inline void WeylGeodesic::SetTRZPMode(){
    dim = 8;
    EvaluateFunction = &WeylGeodesic::Evaluate_TRZP;
}

inline void WeylGeodesic::SetTRZMode(){
    dim = 4;
    EvaluateFunction = &WeylGeodesic::Evaluate_TRZ;
}

inline void WeylGeodesic::SetRZMode(){
    dim = 4;
    EvaluateFunction = &WeylGeodesic::Evaluate_RZ;
}

inline void WeylGeodesic::Destructor(){
    W->Destructor();
    delete W;
}
#endif

```

Arquivo weylgeo.cpp

```

/*-----*/
Numeric Weyl Project
Definition of Geodesic Equations for Weyl Spaces
by Luis Alberto D'Afonseca
begin - 05 November 1998
-----*/
#include "weylgeo.h"

// General Macros
#define w_mu W->Wu()
#define w_drnu W->DrWu()
#define w_dznu W->DzWu()
#define w_phi W->Phi()
#define w_drphi W->DrPhi()
#define w_dzphi W->DzPhi()

void WeylGeodesic::Evaluate_TRZP ( const NReal& P1, NReal& P2 ){
    W->EvaluateIn( P1(1), P1(2) );
}

```

```

real DuPhi = w_drphi * P1(5) + w_dzphi * P1(6),
  expNu = P1(1) * exp( -2.0 * w_nu ),
  a_exp = - alpha * exp( -2.0 * ( w_nu - w_phi ) ),
  dr2 = P1(5) * P1(5), dz2 = P1(6) * P1(6),
  dp2 = P1(7) * P1(7), drz = 2.0 * P1(5) * P1(6),
  rx2 = 2.0 * P1(1);

P2(4) = -2.0*P1(4)*DuPhi; // time equation
P2(7) = 2.0*P1(7)*( DuPhi - 1.0/P1(1) ); // angular equation
P2(5) = a_exp*w_drphi + dp2*expNu*( rx2*w_drphi + 1.0 ) // equation of r
  - drz*( w_dznu - w_dzphi ) - dr2*w_drnu + dz2*( w_drnu - 2.0*w_drphi );
P2(6) = a_exp*w_dzphi + dp2*rx2*expNu - drz*( w_drnu - w_drphi ) // equation of z
  - dz2*w_dznu + dr2*( w_dznu - 2.0*w_dzphi );

P2(0) = P1(4); P2(1) = P1(5); P2(2) = P1(6); P2(3) = P1(7);
}

void WeylGeodesic::Evaluate_TRZ ( const NReal& P1, NReal& P2 ){
  W->EvaluateIn( P1(1), P1(2) );
  real DuPhi = w_drphi * P1(4) + w_dzphi * P1(5),
    a_exp = - alpha * exp( -2.0 * ( w_nu - w_phi ) ),
    dr2 = P1(4) * P1(4), dz2 = P1(5) * P1(5), drz = 2.0 * P1(4) * P1(5);

  P2(3) = -2.0 * P1(3) * DuPhi; // time equation
  P2(4) = a_exp*w_drphi - drz*( w_dznu - w_dzphi ) - dr2*w_drnu // equation of r
    + dz2*( w_drnu - 2.0 * w_drphi );
  P2(5) = a_exp*w_dzphi - drz*( w_drnu - w_drphi ) - dz2*w_dznu // equation of z
    + dr2*( w_dznu - 2.0 * w_dzphi );

  P2(0) = P1(3); P2(1) = P1(4); P2(2) = P1(5);
}

void WeylGeodesic::Evaluate_RZ ( const NReal& P1, NReal& P2 ){
  W->EvaluateIn( P1(0), P1(1) );
  real a_exp = - alpha * exp( -2.0 * ( w_nu - w_phi ) ),
    dr2 = P1(0) * P1(0), dz2 = P1(1) * P1(1), drz = 2.0 * P1(0) * P1(1);

  P2(2) = a_exp*w_drphi - drz*( w_dznu - w_dzphi ) - dr2*w_drnu // equation of r
    + dz2*( w_drnu - 2.0 * w_drphi );
  P2(3) = a_exp*w_dzphi - drz*( w_drnu - w_drphi ) - dz2*w_dznu // equation of z
    + dr2*( w_dznu - 2.0 * w_dzphi );

  P2(0) = dr; P2(1) = dz;
}

```

Arquivo newfall.h

```

/*-----
Numeric Weyl Project
Definition of Newtonian Movement Equations for Weyl Spaces
by Luis Alberto D'Afonseca
begin - 01 March 1998
-----*/

#ifdef NEWTONIAN_FALL_INCLUDED
#define NEWTONIAN_FALL_INCLUDED
#include "function.h"
#include "weylfunc.h"

class NewtonianFall: public RNFunction{
public:
  NewtonianFall( WeylPhi* );
  ~NewtonianFall(){};

  int NumParameters(){ return 0; };
  void SetParameters( const NReal& ){};
  void SetRZPMODE (); // P(i) = { r, z, p, r', z', p' }
  void SetRZMODE (); // P(i) = { r, z, r', z' }
  void EvaluateIn ( const NReal&, NReal&, const real );
  int DimInput () { return dim; };
  int DimOutput () { return dim; };
  void Destructor ();

protected:
  void Evaluate_RZP ( const NReal&, NReal& );
  void Evaluate_RZ ( const NReal&, NReal& );
  void (NewtonianFall::*EvaluateFunction)( const NReal&, NReal& );

  WeylPhi* P;
  int dim;
};

inline NewtonianFall::NewtonianFall( WeylPhi* pp ):
  EvaluateFunction( &NewtonianFall::Evaluate_RZP ),
  P( pp ), dim( 6 ) {}

inline void NewtonianFall::EvaluateIn(const NReal& P1, NReal& P2, const real)
{ (this->*EvaluateFunction)( P1, P2 ); }

```

```

inline void NewtonianFall::SetRZPMode(){
  dim = 6; EvaluateFunction = &NewtonianFall::Evaluate_RZP;
}

inline void NewtonianFall::SetRZMode(){
  dim = 4; EvaluateFunction = &NewtonianFall::Evaluate_RZ;
}

inline void NewtonianFall::Destructor(){
  P->Destructor(); delete P;
}
#endif

```

Arquivo newfall.cpp

```

/*-----*/
  Numeric Weyl Project
  Definition of Newtonian Movement Equations for Weyl Spaces
  by Luis Alberto D'Afonseca
  begin - 01 March 1998
/*-----*/
#include "newfall.h"

void NewtonianFall::Evaluate_RZP ( const NReal& P1, NReal& P2 ){
  real phi, drphi, dzphi,
    rr_abs = fabs(P1(0)), rr_sig = (real)((P1(0) > 0)? 1: -1);

  P->EvaluateIn( rr_abs, P1(1), phi, drphi, dzphi );

  P2(0) = P1(3); P2(1) = P1(4); P2(2) = P1(5);
  P2(3) = rr_abs*P1(5)*P1(5) - drphi * rr_sig;
  P2(4) = - dzphi;
  P2(5) = - 2.0*P1(3)*P1(5) / rr_abs;
}

void NewtonianFall::Evaluate_RZ ( const NReal& P1, NReal& P2 ){
  real phi, drphi, dzphi,
    rr_abs = fabs(P1(0)), rr_sig = (real)((P1(0) > 0)? 1: -1);

  P->EvaluateIn( rr_abs, P1(1), phi, drphi, dzphi );

  P2(0) = P1(2);      P2(1) = P1(3);
  P2(2) = -drphi*rr_sig;  P2(3) = -dzphi;
}

```

Arquivo main.cpp

```

/*-----*/
  Calculo das Geodesicas nas metricas de Weyl
  por Luis Alberto D'Afonseca - Novembro, 1998
/*-----*/
#include <fstream.h>
#include <string.h>
#include <dir.h>
#include "nweylsol.h"
#include "weylgeo.h"
#include "newfall.h"
#include "runge4.h"
#include "ctable.h"

RNFunction* InitEquation( istream& is, NReal P, int theory ){
  WeylPhi *phi = InitPhi ( is ); if( phi == NULL ) return NULL;
  RNFunction* equations;
  if( theory == 1 ){ // Newtonian
    NewtonianFall* nf = new NewtonianFall( phi );
    nf->SetRZMode();
    equations = nf;
  }else{ // Relativity
    WeylNu *nu = phi->ExactNu();
    WeylSol *solution;

    if( nu != NULL ){ solution = new ExactWeylSol( phi, nu );
    }else{
      IntegrationPath *path = InitPath( is ); if( path == NULL ) return NULL;
      AssintNWeylSol sol_aux ( phi, path, 200, 2, 500 );
      sol_aux.EvaluateIn ( P(0), P(1) );
      real aux = sol_aux.Nu();
      path->Destructor();
      delete path;
      solution = new JumpNWeylSol( phi, P(0), P(1), aux );
    }
    WeylGeodesic *geo = new WeylGeodesic ( solution );
    geo->TimeGeodesic ();
    geo->SetRZMode ();
    equations = geo;
  }
}

```

```

return equations;
}

int main( int argc, char *argv[] ){
  if( argc != 3 ){
    cout << "Error: Number of parameters wrong" << endl;
    return -1;
  }

  ifstream temp;
  if ( findfirst(argv[2],&temp,0) == 0 ){
    cout << "File " << argv[2] << " already exist. Overwrite (y/n)? ";
    char cc; cin >> cc;
    if ( cc != 'y' && cc != 'Y' ) return -1;
  }

  ifstream in_file( argv[1] );
  if( in_file.bad() ){
    cout << "Error: File " << argv[1] << " not found" << endl;
    return -1;
  }

  // --- Newtonian or Relativistic ---
  int theory = 0; // 0 = Relativity or 1 = Newtonian

  char str[50]; in_file >> str;
  if( strcmp( "Newtonian", str ) == 0 ){
    theory = 1;
    in_file >> str;
  }

  // --- Start Point ---
  if( strcmp( "StartPoint", str ) != 0 ){
    in_file.close();
    cout << "Error: File " << argv[1] << " is bad" << endl;
    return -1;
  }
  NReal start( 4 ); in_file >> start;
  cout << "Start Point = [ " << start << ']' << endl;

  // --- Step Size ---
  in_file >> str;
  if( strcmp( "StepSize", str ) != 0 ){
    in_file.close();
    cout << "Error: File " << argv[1] << " is bad" << endl;
    return -1;
  }
  real step; in_file >> step;
  cout << "Step size = " << step << endl;

  // --- Number of Points ---
  in_file >> str;
  if( strcmp( "NumPoints", str ) != 0 ){
    in_file.close();
    cout << "Error: File " << argv[1] << " is bad" << endl;
    return -1;
  }
  uinteger NumP; in_file >> NumP;
  cout << "Number of Points = " << NumP << endl;

  // --- Equation ---
  RNFFunction *equations = InitEquation( in_file, start, theory );
  if( equations == NULL ){
    in_file.close();
    cout << "Error: File " << argv[1] << " is bad" << endl;
    return -1;
  }
  Runge4 edo ( equations );
    edo.StartPoint ( 0, start );
    edo.SetStep ( step );

  cout << " calculando " << endl;
  ReColTable table( NumP, edo.Dim() );
  edo.Run( NumP, &table );

  cout << " salvando " << endl;
  ofstream out_file ( argv[2] );
  if( out_file.bad() ){
    in_file.close();
    cout << "Error: File " << argv[2] << " can't by created" << endl;
    return -1;
  }

  out_file.setf( ios::right | ios::scientific );
  out_file << "g = [ ... " << endl;
  uinteger nn = table.GetLines();

  for( uinteger ii = 0; ii < nn; ii++ )

```

```
    out_file << table(ii,0) << " " << table(ii,1) << endl;
out_file << " ];" << endl << endl << "f = [ ";
for( ii = 0; ii < table.GetColumns(); ii++ )
    out_file << table(nn-1,ii) << " ";

out_file << " ];" << endl;
out_file.close();

cout << " fim " << endl;

equations->Destructor();
delete equations;

return 0;
}
```