



Simulação da primeira aproximação post-Newtoniana em sistemas de vários corpos e sistema binário com perda de energia sob forma de ondas gravitacionais

Departamento de Engenharia
Universidade do Minho
Física Computacional

Irving L. Reascos V. A89154

Professor responsável
Nuno Filipe Silva Fernandes Castro

Braga - 2021



Índice

1	Introdução	1
2	Estudo Teórico	1
2.1	Ondas gravitacionais	1
2.2	<i>Post-Newtonian Approximation</i>	2
2.3	Dinâmica de dois corpos	3
2.3.1	Equações de Movimento	3
2.3.2	Evolução da órbita	4
2.3.3	Ondas gravitacionais	4
3	Resolução numérica	5
3.1	Sistema Solar	6
3.1.1	Simulação	6
3.1.2	Resultados	7
3.2	Sistema Binário	8
3.2.1	Simulação	9
3.2.2	Resultados	10
4	Conclusões	13
5	Bibliografia	14

1 Introdução

As observações das ondas gravitacionais são importantes para aprender mais sobre o nosso Universo e conhecer mais sobre os acontecimentos que as produzem [1]. Para isso existem vários detetores no mundo que procuram detectar estes acontecimentos que podem trazer informação relevante ao contexto da astrofísica, sendo as simulações uma parte importante no entendimento de estas informações.

O seguinte relatório surge no contexto da disciplina Física Computacional para simular estes fenómenos previstos pela teoria da relatividade de Einstein com o objetivo de obter um método que permita prever para diferentes sistemas gravitacionais, a sua evolução ao longo do tempo, para isto procura-se simular a dinâmica de um sistema de vários corpos como pode ser o caso do sistema solar e finalmente um sistema binário que perde energia sob a forma de ondas gravitacionais até formar um único corpo.

A metodologia proposta para atingir os objetivos descritos tem com base a aplicação dos conhecimentos adquiridos durante a unidade curricular ao longo do semestre, fazendo uso principalmente da resolução numérica de equações diferenciais e do estudo teórico da relatividade geral, focando todo o trabalho em volta da primeira aproximação post-Newtoniana.

2 Estudo Teórico

2.1 Ondas gravitacionais

Com as equações de campo de Einstein linearizadas numa *gauge* de Lorentz mostra que as perturbações num espaço tempo plano são equações de onda cuja fonte é o tensor de matéria stress e energia, *matter stress-energy tensor*, que com matéria presente no vácuo gera ondas gravitacionais, *gravitational waves*.

Estas ondas ao propagam-se à velocidade da luz e deformam o espaço tempo. É possível diferenciar estas perturbações do espaço tempo em duas polarizações das ondas, h_+ e h_\times , *plus-polarized gravitational wave* e *cross-polarized gravitational wave* respetivamente, cuja deformação do espaço pode ser vista na **Figura 1** [2][3].

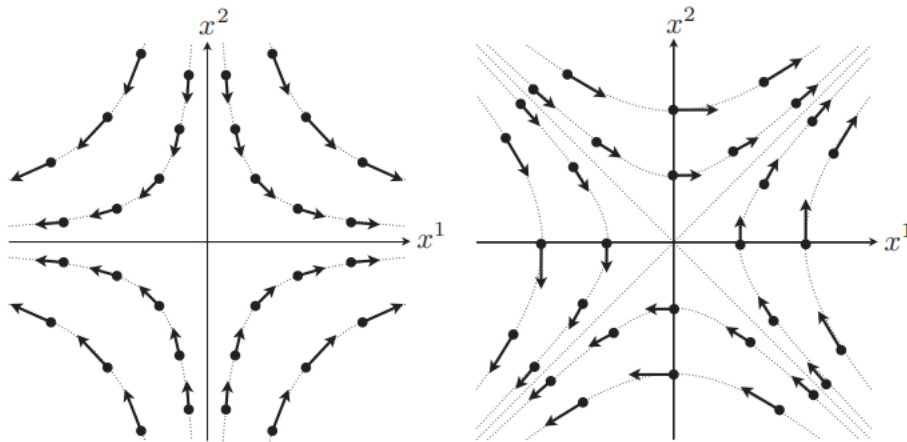


Figura 1: Os graficos mostra a deformação do espaço produzida por uma onda gravitacional a propagar-se em x^3 , para a qual o primeiro gráfico apresenta a polarização (h_+) *plus* e no segundo a (h_\times) *cross*[2].

2.2 Post-Newtonian Approximation

Para simular a dinâmica de vários corpos num sistema gravitacional que tenha em conta os efeitos da relatividade geral de Einstein optou-se por seguir a aproximação post-Newtoniana [2][4]. Portanto, para um sistema de N partículas pontuais temos o seguinte *matter-stress-energy tensor*:

$$\begin{aligned} T^{00} &= \sum_A m_A \left\{ 1 + \frac{1}{2} \frac{v_A^2}{c^2} - \sum_{B \neq A} \frac{Gm_B}{c^2 r_{AB}} \right\} \delta^3(r - r_A(t)) \\ T^{0j} &= \sum_A m_A v_A^j \delta^3(r - r_A(t)) \\ T^{ij} &= \sum_A m_A v_A^i v_A^j \delta^3(r - r_A(t)) \end{aligned} \quad (1)$$

Onde m_A é, r_A , v_A , são a massa, posição e velocidade da partícula A respetivamente, e r_{AB} é a distância entre a partícula A e B . O *stress-energy tensor* efetivo é:

$$\begin{aligned} \tau^{00} &= \sum_A m_A \left\{ 1 + \frac{1}{2} \frac{v_A^2}{c^2} - \sum_{B \neq A} \frac{Gm_B}{c^2 r_{AB}} \right\} \delta^3(r - r_A(t)) \\ &\quad - \frac{1}{8\pi G c^2} [4\Phi \nabla^2 \Phi + 3(\nabla \Phi) \cdot (\nabla \Phi)] \\ \tau^{0j} &= \sum_A m_A v_A^j \delta^3(r - r_A(t)) \\ \tau^{ij} &= \sum_A m_A v_A^i v_A^j \delta^3(r - r_A(t)) \\ &\quad + \frac{1}{8\pi G} \left\{ -2 \frac{\partial \Phi}{\partial r_i} \frac{\partial \Phi}{\partial r_j} - 4\Phi \frac{\partial^2 \Phi}{\partial r_i \partial r_j} \right. \\ &\quad \left. + \delta^{ij} [4\Phi \nabla^2 \Phi + 3(\nabla \Phi) \cdot (\nabla \Phi)] \right\} \end{aligned} \quad (2)$$

onde,

$$\Phi(r, t) = - \sum_B \frac{Gm_B}{\|r - r_B(t)\|} \quad (3)$$

Definimos ainda os seguintes potenciais que são necessários para obter a métrica da primeira ordem post-Newtoniana no *standard gauge*.

$$\Psi(r, t) = \frac{1}{4} \sum_A \frac{Gm_A}{\|r - r_A(t)\|} \left(3 \frac{v_A^2}{c^2} - 2 \sum_{B \neq A} \frac{Gm_B}{c^2 r_{AB}} \right) \quad (4)$$

$$\chi(r, t) = -G \sum m_A \|r - r_A(t)\| \quad (5)$$

Assim, obtemos a métrica

$$\begin{aligned} g_{00} &= -c^2 + 2 \sum_A \frac{Gm_A}{R_A} - 2 \frac{1}{c^2} \left(\sum_A \frac{Gm_A}{R_A} \right)^2 \\ &\quad + 3 \sum_A \frac{Gm_A}{R_A} \frac{v_A^2}{c^2} - 2 \sum_A \sum_{B \neq A} \frac{Gm_A}{R_A} \frac{Gm_B}{c^2 r_{AB}} + O(\varepsilon^6) \\ g_{0i} &= -\frac{7}{2} \sum_A \frac{Gm_A}{c^2 R_A} (v_A)_i - \frac{1}{2} \sum_A \frac{Gm_A (\mathbf{v}_A \cdot \mathbf{R}_A)}{c^2 R_A^3} (R_A)_i + O(\varepsilon^5) \\ g_{ij} &= \delta_{ij} \left(1 + 2 \sum_A \frac{Gm_A}{c^2 R_A} \right) + O(\varepsilon^6) \end{aligned} \quad (6)$$

A partir da qual podemos obter para o dado sistema o lagrangiano

$$\begin{aligned}
\mathcal{L} &:= - \sum_A m_A c^2 \frac{d\tau}{dt} = - \sum_A m_A c \sqrt{-g_{00} - 2g_{0i}v_A^i - g_{ij}v_A^i v_A^j} \\
&= - \sum_A m_A \left(c^2 - \frac{1}{2}v_A^2 - \frac{1}{8}\frac{v_A^4}{c^2} \right) \\
&\quad + \frac{1}{2} \sum_A \sum_{B \neq A} \frac{Gm_A m_B}{r_{AB}} \left[1 + 3\frac{v_A^2}{c^2} - \sum_{C \neq A} \frac{Gm_C}{c^2 r_{AC}} - \frac{7}{2} \frac{\mathbf{v}_A \cdot \mathbf{v}_B}{c^2} \right. \\
&\quad \left. - \frac{1}{2} \frac{(\mathbf{v}_A \cdot \mathbf{r}_{AB})(\mathbf{v}_B \cdot \mathbf{r}_{AB})}{c^2 r_{AB}^2} \right]
\end{aligned} \tag{7}$$

que de acordo com as equações de *Euler-Lagrange*, permite-nos obter as equações de movimento chamadas *Einstein-Infeld-Hoffman equations*

$$\begin{aligned}
\frac{d^2 \mathbf{r}_A}{dt^2} &= - \sum_{B \neq A} \frac{Gm_B \mathbf{r}_{AB}}{r_{AB}^3} \left[1 - 4 \sum_{C \neq A} \frac{Gm_C}{c^2 r_{AC}} \right. \\
&\quad + \sum_{C \neq A, B} \left(-\frac{Gm_C}{c^2 r_{BC}} + \frac{1}{2} \frac{Gm_C (\mathbf{r}_{AB} \cdot \mathbf{r}_{BC})}{c^2 r_{BC}^3} \right) \\
&\quad - 5 \frac{Gm_A}{c^2 r_{AB}} + \frac{v_A^2}{c^2} - 4 \frac{\mathbf{v}_A \cdot \mathbf{v}_B}{c^2} + 2 \frac{v_B^2}{c^2} - \frac{3}{2} \left(\frac{\mathbf{v}_B \cdot \mathbf{r}_{AB}}{c r_{AB}} \right)^2 \Big] \\
&\quad - \frac{7}{2} \sum_{B \neq A} \frac{Gm_B}{r_{AB}} \sum_{C \neq A, B} \frac{Gm_C \mathbf{r}_{BC}}{c^2 r_{BC}^3} \\
&\quad + \sum_{B \neq A} \frac{Gm_B}{r_{AB}^3} \frac{\mathbf{r}_{AB} \cdot (4\mathbf{v}_A - 3\mathbf{v}_B)}{c} \frac{\mathbf{v}_A - \mathbf{v}_B}{c}
\end{aligned} \tag{8}$$

2.3 Dinâmica de dois corpos

2.3.1 Equações de Movimento

Partindo do lagrangiano da equação 7, para dois corpos reduz-se a

$$\begin{aligned}
\mathcal{L} &= - (m_1 + m_2) c^2 + \frac{1}{2} m_1 v_1^2 + \frac{1}{2} m_2 v_2^2 + \frac{1}{8} m_1 \frac{v_1^4}{c^2} + \frac{1}{8} m_2 \frac{v_2^4}{c^2} \\
&\quad + \frac{Gm_1 m_2}{r_{12}} \left[1 + \frac{3}{2} \frac{v_1^2 + v_2^2}{c^2} - \frac{1}{2} \frac{G(m_1 + m_2)}{c^2 r_{12}} - \frac{7}{2} \frac{\mathbf{v}_1 \cdot \mathbf{v}_2}{c} \right. \\
&\quad \left. - \frac{1}{2} \frac{(\mathbf{v}_1 \cdot \mathbf{r}_{12})(\mathbf{v}_2 \cdot \mathbf{r}_{12})}{c^2 r_{12}^2} \right]
\end{aligned} \tag{9}$$

Que no sistema de coordenadas do centro de massa origina a equação de movimento seguinte

$$\begin{aligned}
\frac{d^2 \mathbf{r}}{dt^2} &= - \frac{GM}{r^3} \mathbf{r} \left[1 - (4 + 2\eta) \frac{GM}{c^2 r} + (1 + 3\eta) \frac{v^2}{c^2} - \frac{3}{2} \eta \frac{(\mathbf{v} \cdot \mathbf{r})^2}{c^2 r^2} \right] \\
&\quad + (4 - 2\eta) \frac{GM}{c^2 r^3} (\mathbf{v} \cdot \mathbf{r}) \mathbf{v}
\end{aligned} \tag{10}$$

onde as posições e velocidades do corpo 1 e 2 ficam definidas da seguinte maneira nas coordenadas do centro de massa,

$$\mathbf{r}_1 = -\frac{m_2}{M} \mathbf{r}, \quad \mathbf{r}_2 = \frac{m_1}{M} \mathbf{r}, \quad \mathbf{v}_1 = -\frac{m_2}{M} \mathbf{v}, \quad \mathbf{v}_2 = \frac{m_1}{M} \mathbf{v} \tag{11}$$

onde,

$$M = m_1 + m_2, \quad \mathbf{r} := \mathbf{r}_{12}, \quad \mu := \frac{m_1 m_2}{M}, \quad \eta := \frac{\mu}{M} \quad (12)$$

2.3.2 Evolução da órbita

O sistema binário irá perturbar a métrica no espaço no qual se encontra, esta perturbação chamada de onda gravitacional faz o sistema perder energia, originando uma órbita dinâmica. Este dinamismo da órbita pode ser expresso em ordem a um parâmetro post-Newtoniano $\nu := (GM\omega/c^3)^{2/3}$, que se relaciona com as componentes do sistema binário da seguinte maneira.[2]

$$\nu^{3/2} = \frac{GM\omega}{c^3} = \frac{GMv}{c^3 r} \quad (13)$$

que para $\left(\frac{v^2}{c^2}\right)^2 \ll 1$ podemos reescrever e obter o seguinte,

$$\nu = \frac{v^2}{c^2} \left[1 + \frac{2}{3} (3 - \eta) \frac{v^2}{c^2} \right] \quad (14)$$

escrevendo a velocidade em função do parâmetro post-Newtoniano temos que

$$v^2 = \nu c^2 \left[1 - \frac{2}{3} (3 - \eta) \nu \right] \quad (15)$$

Finalmente, a evolução temporal deste parâmetro, junto com a fase orbital φ é descrita pelo método de *TaylorT4* da seguinte maneira

$$\begin{aligned} \frac{d\nu}{dt} = & \frac{64}{5} \eta \frac{c^3}{GM} x^5 \left\{ 1 - \left(\frac{743}{336} + \frac{11}{4} \eta \right) x + 4\pi x^{3/2} \right. \\ & + \left(\frac{34103}{18144} + \frac{13661}{2016} \eta + \frac{59}{18} \eta^2 \right) x^2 \\ & - \left(\frac{4159}{672} + \frac{189}{8} \eta \right) \pi x^{5/2} \\ & + \left[\frac{16447322263}{139708800} + \frac{16}{3} \pi^2 - \frac{1712}{105} \gamma_E - \frac{856}{105} \ln(16x) \right] x^3 \\ & \left. - \left(\frac{4415}{4032} - \frac{358675}{6048} \eta - \frac{91495}{1512} \eta^2 \right) \pi x^{7/2} \right\} \end{aligned} \quad (16)$$

$$\frac{d\varphi}{dt} = \frac{c^3}{GM} x^{3/2}$$

Assim, juntamente com as relações anteriormente descritas nas equações 13 a 15 podemos prever como se vai comportar o sistema no tempo.

2.3.3 Ondas gravitacionais

Como visto anteriormente o sistema perde energia sob a forma de ondas gravitacionais, cuja forma da onda h_{ij}^{TT} para um observador na posição (r, θ, ϕ) relativamente ao sistema de coordenadas pode ser descrito em termos de duas componentes, que correspondem aos dois tipos de polarização das ondas, h_+ e h_\times .

Estas polarizações podem ser escritas como uma expansão em ordem dos *spin-2-weighted spherical harmonics* [2],

$$h_+ + ih_\times = \sum_{l=2}^{\infty} \sum_{-l}^l -2Y_{lm}(\theta, \phi) h_{lm} \quad (17)$$

Para a aproximação post-Newtoniana de primeira ordem, os modos relevantes são: h_{22} , h_{21} , h_{33} , h_{32} , h_{31} , h_{44} e h_{42} (Anexo 1)[Livro1].

3 Resolução numérica

Para simular os diferentes sistemas caracterizados pelas equações diferenciais definidas no estudo teórico, separou-se estas em várias equações diferenciais de primeira ordem. Para as equações diferenciais de segunda ordem optou-se por se separar em duas equações diferenciais de primeira ordem da seguinte maneira,

$$\frac{d^2 r}{dt^2} = f(r, t) \quad (18)$$

$$\begin{aligned} \frac{dr}{dt} &= v(t) \\ \frac{dv}{dt} &= f(r, t) \end{aligned} \quad (19)$$

Para cada caso em concreto de forma a facilitar e simplificar a implementação da solução do problema, resolveu-se fazer uso dos objetos em *python*, criando dois objetos principais que são: *Particle* e *System*.

As partículas, *Particle*, têm como atributos todos as características essenciais e não essenciais que as definem, como posição, velocidade, massa, etc. No caso do sistema, *System*, como o seu nome indica, é o sistema de várias partículas cujas dinâmicas são caracterizadas pelas equações de movimento anteriormente descritas.

É neste objeto, *System*, que se encontra a resolução numérica das equações diferenciais tendo em conta as interações entre cada um dos corpos que conformam o sistema. Para resolver em cada intervalo de tempo h , ou seja o passo temporal para o qual será calculada a dinâmica do sistema, foi usado o método de *Runge-Kutta* de ordem 4.

```
def Runge_Kutta(f, r0, t0, tf, h, *args):
    T = np.arange(t0, tf, h)
    r = [np.array(r0)]
    for i, t in enumerate(T[:-1]):
        k0 = h*f(r[i], t, *args)
        k1 = h*f(r[i] + k0/2, t + h/2, *args)
        k2 = h*f(r[i] + k1/2, t + h/2, *args)
        k3 = h*f(r[i] + k2, t + h, *args)
        r.append(r[i] + 1/6*(k0 + 2*k1 + 2*k2 + k3))
    return np.array(r)
```

Este método foi modificado de forma a ser implementado com o resto das necessidades que problema implica.

3.1 Sistema Solar

Como primeira abordagem ao problema principal foram implementadas as equações do movimento para um sistema de N corpos no qual não se tem em conta a perda de energia por ondas gravitacionais. Assim resolveu-se simular o sistema solar, devido ao fácil acesso às condições iniciais dos vários corpos que o conformam e que facilmente se pode verificar, visualizando o período da sua órbita, se os resultados obtidos se encontram em concordância com a realidade.

Para isto, como anteriormente descrito foi usada a dinâmica do sistema para N corpos, resolvendo as equações da primeira aproximação post-Newtoniana que se encontra na equação 8. Assim a sua resolução pelo método de *Runge-Kutta* pode ser visto no ficheiro *system-Post-Newtonian.py*. De forma a implementar a equação diferencial de segunda ordem em duas de primeira ordem para os N corpos do sistema optou-se por resolver em cada intervalo de tempo as equações para todos os corpos, sendo a sua implementação uma função do objeto *System*, *calc_MotionEquation*.

Assim, a simulação do sistema solar vista no *Jupyter Notebook*, *SystemBodies* usa unicamente os dois objetos presentes no ficheiro anteriormente mencionado.

3.1.1 Simulação

A primeira parte consiste em definir as partículas, *Particles*, que irão conformar o sistema, para isto definiu-se os planetas do sistema solar com as suas respetivas condições iniciais como posição e velocidade e as suas características como massa [6], nome e cor, tendo as últimas o propósito da visualização dos resultados e não dos cálculos.

```
Sun = Particle(1.9891e30, name='Sun', color='#FFBD00')
Mercury = Particle(3.3011e23, [69816900e3, 0, 0], [0, 47.362e3, 0],
                    name='Mercury', color='#6C3417')
Venus = Particle(4.8675e24, [108939000e3, 0, 0], [0, 35e3, 0],
                      name='Venus', color='#549581')
Earth = Particle(5.9722e24, np.array([1.4710e11, 0, 0]), np.array([0, 3.0297e4, 0]),
                 name='Earth', color='#1BB46E')
Mars = Particle(6.4185e23, [249209300e3, 0, 0], np.array([0, 24.077e3, 0]),
                 name='Mars', color='#BA3405')
Jupiter = Particle(1.8982e27, [740.52e9, 0, 0], [0, 13.07e3, 0],
                       name='Jupiter', color='#92A9B7')
Saturn = Particle(5.6834e26, [1352.55e9, 0, 0], [0, 9.68e3, 0],
                   name='Saturn', color='#B09F59')
Uranus = Particle(8.6810e25, [2.742e12, 0, 0], [0, 6.80e3, 0],
                     name='Uranus', color='#8ED8F3')
Neptune = Particle(1.02413e26, [4.46e12, 0, 0], [0, 5.43e3, 0],
                    name='Neptune', color='#257A99')

Bodies = [Sun, Mercury, Venus, Earth, Mars, Jupiter, Saturn, Neptune, Uranus]
```

Posteriormente cria-se o sistema e adiciona-se-lhe cada corpo, neste caso cada planeta.

```
SOLAR_SYSTEM = System()
for body in Bodies:
    SOLAR_SYSTEM.add_Particle(body)
```

Finalmente, faz-se o ciclo no qual se atualiza em cada tempo t a posição e velocidade de cada corpo

```
while t <= tf:
    sys.stdout.write(f'\r{np.round(t/tf*100, 2)}/{100}%')
    sys.stdout.flush()
    r = SOLAR_SYSTEM.evolution_dt(h)
    SOLAR_SYSTEM.modify_R(r)
```


$$t += h$$

3.1.2 Resultados

Com o propósito de simular o sistema solar, e obter uma trajetória aproximada à realidade de todos os planetas, de pelo menos uma volta ao sol, o tempo total de simulação teria de ser de pelo menos 164 anos terrestres aproximadamente pois é o período máximo de todos os corpos que conformam o sistema, correspondendo ao planeta Neptuno. Assim para termos um passo relativamente grande de forma a não ter tempos de simulação muito grandes mas que permita ainda simular as órbitas de corpos com períodos pequenos como por exemplo o da Terra, foi usado um passo temporal h de 4 dias, que correspondem a 345600 segundos.

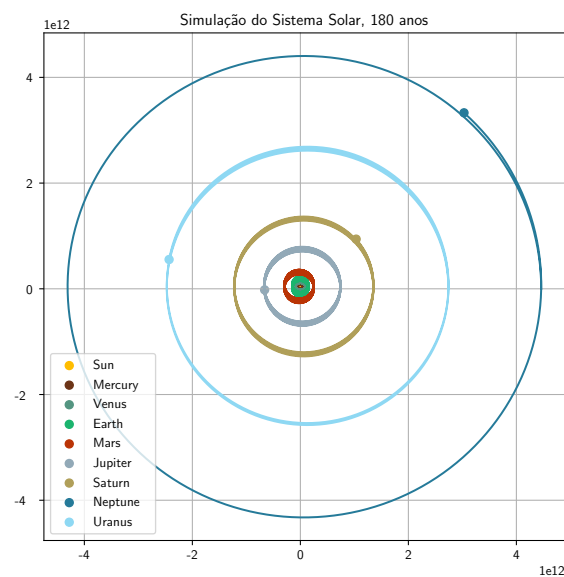


Figura 2: Simulação de 180 anos com passo temporal de 4 dias do Sistema Solar

É possível observar que para os corpos com um período orbital grande o suficiente o erro na simulação é pequeno enquanto para corpos com períodos pequenos como Mercúrio, Vênus, Terra ou Marte, o erro da sua órbita é maior, isto deve-se ao passo temporal ser grande demais, pois a sua dinâmica varia mais rapidamente no tempo do que os outros corpos.

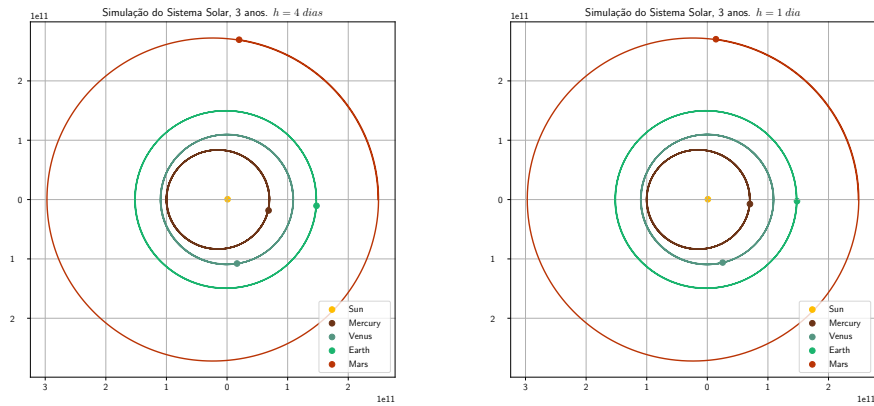


Figura 3: Gráfico de 5 corpos da simulação de 3 anos do Sistema Solar, para passo temporal de 4 e 1 dia.

O erro evidente é na diferença angular das posições entre ambas as simulações sendo os valores mais realistas quando o passo temporal h é mais pequeno algo que é evidente no gráfico seguinte, tomando como exemplo a última posição de cada corpo simulado ao fim dos 3 anos.

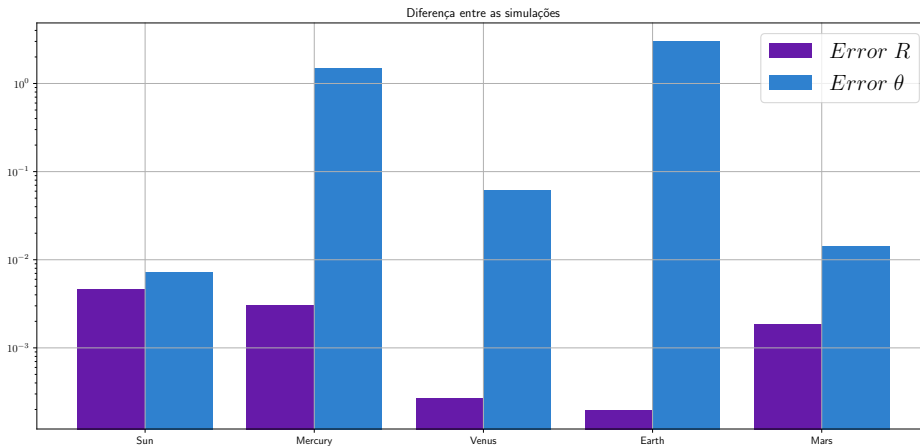


Figura 4: Diferença entre as posições finais entre as simulações com passo $h = 4$ dias e $h = 1$ dia

O erro é proporcional ao tempo de simulação, portanto, para a simulação de 180 anos é evidente que as posições dos corpos com períodos pequenos vão ser menos fiéis à realidade, ou seja, apresentaram um erro maior em relação aos corpos cuja dinâmica é mais lenta.

3.2 Sistema Binário

Com o aprendido no problema anteriormente abordado e tendo verificado a fiabilidade do método, abordamos o problema principal em questão, simular um sistema binário que perde energia em forma de ondas gravitacionais. Para isto tendo em conta o mesmo método proposto foram resolvidas as equações diferenciais tanto da dinâmica do sistema como dos parâmetros post-Newtonianos que caracterizam o mesmo.

Em termos do programa, temos de igual maneira no ficheiro *system_dinamics.py* dois objetos,

um para as partículas e outro para o sistema binário. O objeto das partículas é semelhante ao que foi visto no problema anterior, enquanto o objeto que define o sistema tem componentes extras pois não só se encarga de resolver numericamente as equações de movimento dos dois corpos mas também da perda de energia em forma de ondas gravitacionais, cujos parâmetros post-Newtonianos da equação 16 são também caracterizados por equações diferenciais e posteriormente relacionar ambos os resultados para ter a simulação completa.

3.2.1 Simulação

Para simular o sistema primeiramente foi necessário definir as condições iniciais do mesmo, sendo as massas dos corpos, velocidades e/ou posições. Isto deve-se ao facto de que com as equações 13 a 15 podemos relacionar a posição inicial no caso de querer ter como condição inicial apenas a velocidade de ambos os corpos.

O cálculo das equações de movimento encontra-se definido pela função *calc_MotionEquation*, a evolução temporal dos parâmetros post-Newtonianos por *calc_Args* e finalmente a relação entre os seus resultados está descrita na função *calc_vectors*. A cada passo temporal h os valores das grandezas são atualizados pela função *evolution_dt* que corresponde à implementação do método de *Runge-Kutta* de ordem 4.

Posteriormente com o sistema definido resolveu-se as equações diferenciais para obter as posições, velocidades e valores dos parâmetros post-Newtonianos para cada passo temporal h usando a função *export_data* do ficheiro *two_bodies_simulation.py*, o qual nos permitirá salvar os resultados finais para posteriores análises.

```
while t <= tf:
    sys.stdout.write(f'\r{np.round(t/tf*100,2)}/{100}%')
    sys.stdout.flush()
    r,args = System.evolution_dt(h)
    if np.isnan(r).any():
        print('NAN ERROR',t/tf)
        break
    System.calc_vectors(r,args)
    t += h
```

Com os resultados obtidos da simulação do sistema binário é possível obter a partir da equação 17, a forma da onda gravitacional para o sistema. Para o cálculo das componentes associadas às polarizações *plus* e *cross* como visto na equação 17 é preciso calcular os *spin-2-weightedspherical harmonics* para os quais foi usado um código de *python* já desenvolvido que se encontra no ficheiro *spin_weighted_Yml.py*[5]. Assim, finalmente, com a função *calc_WaveForm* do ficheiro *calc_gravitational_waves.py* consegue-se obter o módulo H da forma de onda.

$$H = h_+ + ih_\times \quad (20)$$

```
def calc_WaveForm(r,theta,phi,psi,x,x0,m1,m2):
    M = m1+m2
    u = m1*m2/M
    n = u/M
    dm = m1-m2

    ls = [2,3,4]

    H = 0+0j
```

```

h22 = -sqrt(pi/5)*G*u/(c**2*r)*np.exp(-2j*psi)*x*(\
    1-(107/42-55/42*n)*x + (2*pi+6j*np.log(x/x0))*x**1.5\
    - (2173/1512+1096/216*n-2047/1512*n**2)*x**2\
    -((107/21-34/21*n)*pi+24j*pi+1j*(107/7-34/7*n)*np.log(x/x0))*x**2.5\
    \
    +(27027409/646800-856/105*gammaE+2/3*pi**2-1712/105*np.log(2)-428/\
    105*np.log(x)\
    -18*(np.log(x/x0))**2-(278185/33264-41/96*pi**2)*n-20261/2772*n**2\
    +114635/99792*n**3\
    +1j*428/105*pi+12j*pi*np.log(x/x0))*x**3)
h21 = -1j*8/3*sqrt(pi/5)*G*u/(c**2*r)*dm/M*np.exp(-1j*psi*x**1.5)
h33 = 3j*sqrt(pi/7)*G*u*dm/(c**2*r*M)*np.exp(-3j*psi)*x**1.5
h32 = -8/3*sqrt(pi/7)*G*u/(c**2*r)*np.exp(-2j*psi)*(1-3*n)*x**2
h31 = -1j/3*sqrt(2*pi/35)*G*u*dm/(c**2*r*M)*np.exp(-1j*psi)*x**1.5
h44 = 64/9*sqrt(pi/7)*G*u/(c**2*r)*np.exp(-4j*psi)*(1-3*n)*x**2
h42 = -8/63*sqrt(pi)*G*u/(c**2*r)*np.exp(-2j*psi)*(1-3*n)*x**2

hs = {'h21':h21, 'h22':h22, 'h31':h31, 'h32':h32, 'h33':h33, 'h44':h44, 'h42':h42
      }

for l in ls:
    if l != 4:
        for m in range(1,l+1):
            h = hs['h'+str(l)+str(m)]
            H += sYlm(-2,l,m,theta,phi)*h + sYlm(-2,l,-m,theta,phi)*(-1)**(
                -1)*np.conjugate(h)
        else:
            h = hs['h'+str(l)+'2']
            m = 2
            H += sYlm(-2,l,m,theta,phi)*h + sYlm(-2,l,-m,theta,phi)*(-1)**(-1)*
                np.conjugate(h)
            h = hs['h'+str(l)+'4']
            m = 4
            H += sYlm(-2,l,m,theta,phi)*h + sYlm(-2,l,-m,theta,phi)*(-1)**(-1)*
                np.conjugate(h)

return H

```

3.2.2 Resultados

Definiu-se um sistema binário definido por dois corpos de 8 e 5 massas solares a orbitar inicialmente com uma velocidade de módulo de 0.1 vezes a velocidade da luz. Usando como já referido como condição inicial, a velocidade, podemos obter para um tempo de $t_c = 1.5184$ segundos com passo temporal $h = 0.8 \mu s$ a dinâmica do sistema.

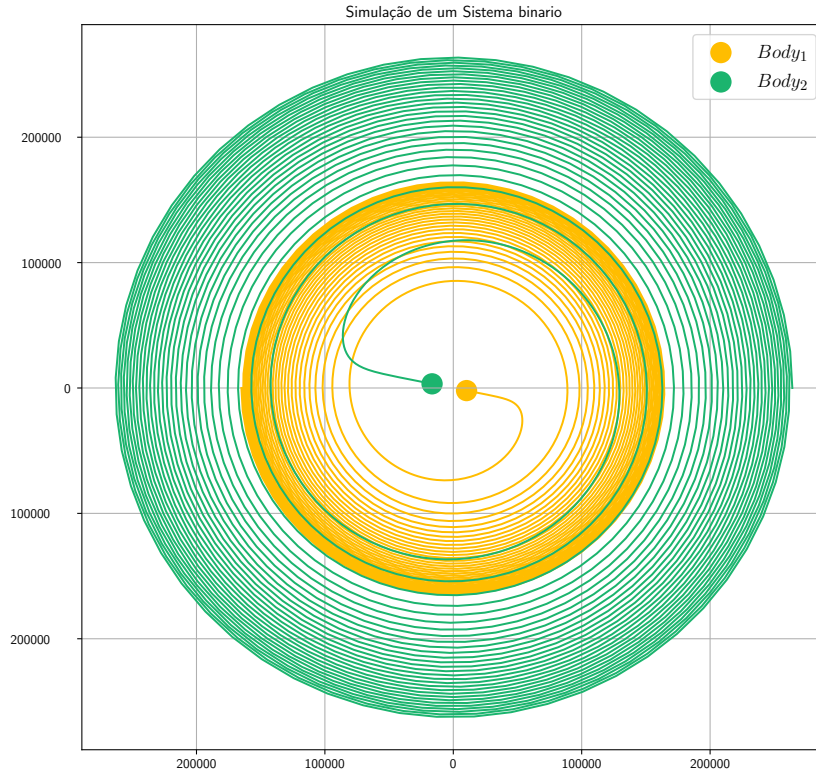


Figura 5: Simulação de 1.5184 s do sistema binário de corpos de 8 e 5 massas solares com passo temporal de $h = 0.8 \mu s$.

Na **Figura 5** é possível ver que os corpos orbitam o centro do sistema de massa e acabam "caindo" para o mesmo num tempo t_c , tempo no qual ambos corpos vão se juntar e formar um único corpo. Esta queda para o centro de massa do sistema como anteriormente discutido no estudo teórico deve-se a que o sistema perde energia sob a forma de ondas gravitacionais, para este sistema o módulo da onda gravitacional para uma posição $r = 5 \text{ km}$ no mesmo plano de órbita cresce no tempo, ou seja, o seu maximo será no momento em que ambos os corpos se juntam.

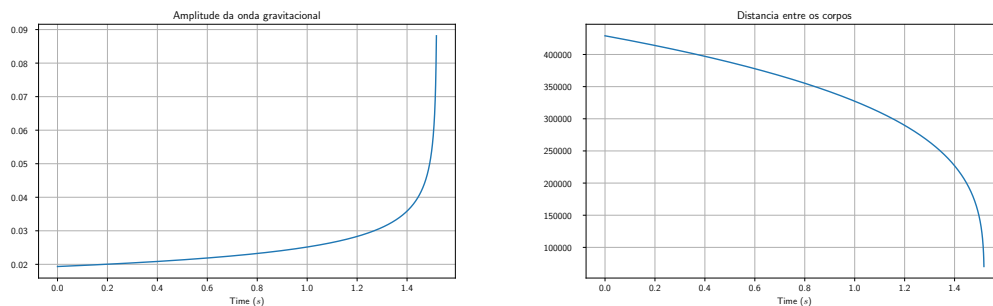


Figura 6: Gráfico da evolução temporal da distância entre os corpos e do módulo da onda H .

Com os resultados anteriormente vistos podemos visualizar a evolução da órbita, tendo como

condição inicial a velocidade dos corpos. Usando como condição inicial a posição e a velocidade os resultados da evolução temporal parecem não se adequar à realidade mostrando os pontos fracos da metodologia usada para resolver o problema. No gráfico da figura 7 vê-se a simulação de 5×10^{11} s com um passo temporal $h = 8 \times 10^6$ s, para os mesmos corpos de 8 e 5 massas solares.

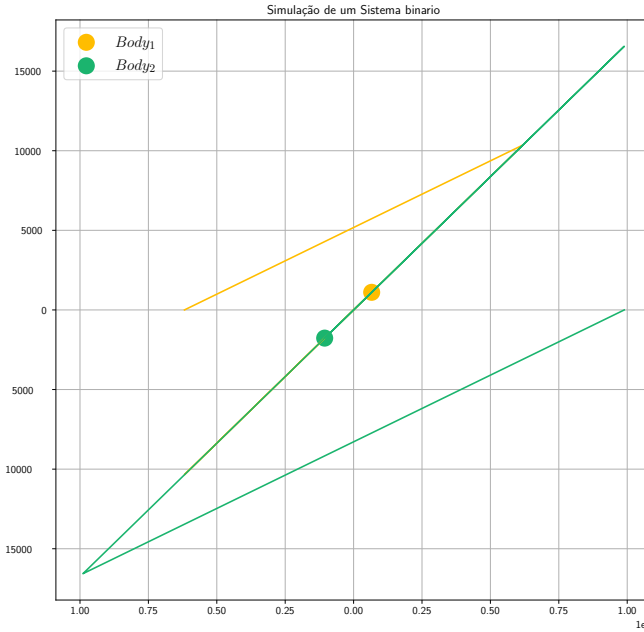


Figura 7: Simulação de um Sistema binário de dois corpos de 8 e 5 massas solares com posições iniciais de 0.1 Unidades Astronómicas.

A evolução no tempo da onda gravitacional gerada pelo sistema e a distância entre os corpos é visualizada na figura 8

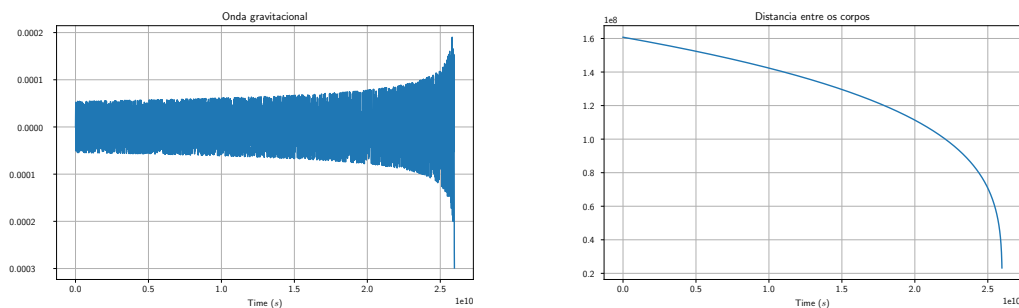


Figura 8: Gráfico da evolução temporal da distância entre os corpos e da onda H

É possível verificar que a dinâmica de aproximação dos corpos é mantida e que conforme vão-se aproximando, a amplitude da onda gravitacional gerada pelo sistema aumenta e a distância entre os corpos decai até formar um único corpo. No entanto, no gráfico da órbita observado podemos pensar que existe algum erro na simulação que pode ter a ver com o passo temporal utilizado ou um possível erro na própria implementação da simulação.

4 Conclusões

Para atingir os objetivos propostos no início do trabalho, começou-se por estudar de forma sucinta a teoria de relatividade geral de Einstein fazendo uso de dois livros, os quais formam parte da bibliografia deste trabalho. A compreensão da teoria foi um dos principais obstáculos, visto que a sua formulação não só aborda conceitos contraintuitivos mas também o facto de que esta apresenta a sua própria notação. Ficando a teoria minimamente compreendida, o foco foi posto só no caso concreto das ondas gravitacionais e na dinâmica dos corpos, para o qual como foi visto no trabalho decidiu-se implementar a primeira aproximação post-Newtoniana, pois para um primeiro contacto com a teoria de relatividade de Einstein era a que mais facilmente se poderia compreender.

Desta forma é possível concluir que a simulação da dinâmica de vários, corpos como no caso abordado do sistema solar, foi bem conseguida, pois facilmente nos resultados apresentados se conseguiu observar o seu correto funcionamento tendo em atenção as próprias limitações do modelo teórico e da própria implementação do problema. No caso da simulação do sistema binário, como foi visto parece atingir os resultados esperados, mas que ao não ter um ponto de comparação não é possível concluir que o obtido é uma boa aproximação da realidade, isto deve-se a que sistemas binários que foram detectados, por exemplo pelo LIGO, vêm acompanhados das suas características e no caso do presente, simulá-las vai além das capacidades do modelo programado dando erros de números infinitos principalmente no momento de relacionar os parâmetros post-Newtonianos com a velocidade do sistema. Portanto, conclui-se que se atingiu o objetivo proposto de simular este tipo de sistemas astronómicos mas com as suas limitações que podem ser para trabalhos futuros melhoradas e corrigidas.

Finalmente fica um trabalho que desde a minha perspectiva não se encontra finalizado havendo ainda possíveis melhoramentos para resolver os problemas e limitações do modelo teórico e da simulação, como abordar o problema com uma aproximação post-Newtoniana de ordem superior e em termos de simulação, resolver a problemática de números muito grandes fazendo uso de outro sistema de unidades que permitam trabalhar para grandes velocidades e distâncias mas com números mais pequenos, evitando problemas de infinitos na máquina.

5 Bibliografia

- [1] B. P. Abbott et al., A guide to LIGO–Virgo detector noise and extraction of transient gravitational-wave signals, *Class. Quantum Grav.*, vol. 37, n. 5, p. 055002, Fev. 2020, doi: 10.1088/1361-6382/ab685e.
- [2] J. Creighton e W. Anderson, *Gravitational-Wave Physics and Astronomy: An Introduction to Theory, Experiment and Data Analysis*, Gravitational-Wave Physics and Astronomy: An Introduction to Theory, Experiment and Data Analysis, Set. 2011, doi: 10.1002/9783527636037.
- [3] R. d’Inverno, *Introducing Einstein’s Relativity*. Oxford, New York: Oxford University Press, 1992.
- [4] R. V. Wagoner e C. M. Will, Post-Newtonian gravitational radiation from orbiting point masses, *The Astrophysical Journal*, vol. 210, pp. 764–775, Dez. 1976, doi: 10.1086/154886.
- [5] Christian Reisswig (2009-2011),
<https://svn.einsteintoolkit.org/pyGWAnalysis/trunk/DataAnalysis/harmonics.py>
- [6] https://en.wikipedia.org/wiki/Solar_SystemJupiter

Anexo A - Modos relevantes (Kidder, 2008)[2]

$$\begin{aligned}
h_{22} &= -8\sqrt{\frac{\pi}{5}}\frac{G\mu}{c^2r}e^{-2i\varphi}\nu\left[1-\left(\frac{107}{42}-\frac{55}{42}\eta\right)\nu\right]+O(\varepsilon^5) \\
h_{21} &= -i\frac{8}{3}\sqrt{\frac{\pi}{5}}\frac{G\mu}{c^2r}\frac{\delta m}{M}e^{-i\varphi}\nu^{3/2}+O(\varepsilon^5) \\
h_{33} &= 3i\sqrt{\frac{6\pi}{7}}\frac{G\mu}{c^2r}\frac{\delta m}{M}e^{-3i\varphi}\nu^{3/2}+O(\varepsilon^5) \\
h_{32} &= -\frac{8}{3}\sqrt{\frac{\pi}{7}}\frac{G\mu}{c^2r}e^{-2i\varphi}(1-3\eta)\nu^2+O(\varepsilon^6) \\
h_{31} &= -\frac{i}{3}\sqrt{\frac{2\pi}{35}}\frac{G\mu}{c^2r}\frac{\delta m}{M}e^{-i\varphi}\nu^{3/2}+O(\varepsilon^5) \\
h_{44} &= \frac{64}{9}\sqrt{\frac{\pi}{7}}\frac{G\mu}{c^2r}e^{-4i\varphi}(1-3\eta)\nu^2+O(\varepsilon^6) \\
h_{42} &= -\frac{8}{63}\sqrt{\pi}\frac{G\mu}{c^2r}e^{-2i\varphi}(1-3\eta)\nu^2+O(\varepsilon^6)
\end{aligned} \tag{21}$$

onde $\delta_m := m_1 - m_2$