

UNIVERSIDADE FEDERAL DO ABC  
UFABC

Carla das Virgens Pequini

Estudo numérico da gravitação e o problema  
dos  $n$  corpos

BACHARELADO DE CIÊNCIAS & TECNOLOGIA

SÃO PAULO  
2015

UNIVERSIDADE FEDERAL DO ABC  
UFABC

Carla das Virgens Pequini

# Estudo numérico da gravitação e o problema dos $n$ corpos

BACHARELADO DE CIÊNCIAS & TECNOLOGIA

Relatório submetido a Universidade Federal do ABC, com informações finais sobre o andamento do projeto de iniciação científica, sob a orientação da Prof. Dra. Cecília Chirenti.

SÃO PAULO  
2015

## Resumo

O presente trabalho tem como intuito mostrar como a lei da gravitação universal de Newton atua no problema de Kepler, dois corpos, três corpos e  $n$  corpos através do uso do programa em Java Triana, que por sua vez atua resolvendo equações de movimento, explorando a dinâmica e evolução de diferentes sistemas gravitacionais. No intuito de resolver o problema de  $n$  corpos, para  $n$  muito grande, foi feito o estudo do Algoritmo de Barnes-Hut.

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Objetivos</b>	<b>4</b>
<b>3</b>	<b>Métodos</b>	<b>5</b>
3.1	Linguagens Computacionais . . . . .	5
3.2	Métodos Computacionais . . . . .	5
<b>4</b>	<b>Desenvolvimento</b>	<b>7</b>
4.1	Método Numérico . . . . .	7
4.1.1	Orbit . . . . .	7
4.1.2	Binary . . . . .	9
4.1.3	MercPert . . . . .	10
4.1.4	Multiple . . . . .	10
4.1.5	Algoritmo de Barnes-Hut . . . . .	10
4.2	Método Analítico . . . . .	11
<b>5</b>	<b>Resultados</b>	<b>16</b>
5.1	Orbit . . . . .	16
5.2	Binary . . . . .	22
5.3	MercPert . . . . .	24
5.4	Multiple . . . . .	25
5.5	Algoritmo de Barnes-Hut . . . . .	27
<b>6</b>	<b>Cronograma</b>	<b>28</b>
<b>7</b>	<b>Conclusão</b>	<b>29</b>

# 1 Introdução

O presente trabalho tem o intuito de introduzir o problema de força central, dois e três corpos, para então entrar no problema de  $n$  corpos.

Segundo Newton o problema de  $n$  corpos pode ser formulado como:

"Conhecidas as massas de  $n$  corpos que se movem governados pela gravitação mútua e dadas suas respectivas posições e velocidades num determinado instante, calcular suas posições e velocidades num instante futuro" [1].

O caso de força central (quando  $n= 1$ ) apresenta a força gravitacional que um corpo central realiza em outro. Nesse caso é desconsiderada a terceira lei de Newton, pois considera-se nula a força que o corpo em órbita exerce sob o corpo central. Esse tipo de sistema é desenvolvido pelo código em Java Orbit [2]. Neste caso são mais facilmente observadas as três leis de Kepler.

- A primeira lei afirma: "As órbitas descritas pelos planetas em redor do Sol são elipses, com o Sol num dos focos" [3]. Os cálculos utilizados no programa Triana [4] geram órbitas que comprovam essa lei e serão apresentadas no decorrer do trabalho.
- A segunda lei afirma: "O raio vetor que liga um planeta ao Sol descreve áreas iguais em tempos iguais" [3]. A observação dessa lei também é apresentada neste trabalho através de alterações no código em Java do Triana, que geram um gráfico de uma constante, representando essa relação área-tempo.
- A terceira lei afirma: "O quadrado do período de revolução é proporcional ao cubo da média da distância ao Sol" [5]. A observação dessa lei é feita através da análise de gráficos e alguns cálculos.

O caso de dois corpos considera o que o anterior desconsidera, mostrando as forças gravitacionais exercidas por cada corpo envolvido no sistema. Nesse caso utiliza-se o Binary [6]. Com alterações no código desse programa é possível observar como o centro de massa do sistema se comporta, verificando o Teorema da Conservação do Momento Linear, que afirma que se a força externa total é zero, o momento linear total do sistema de partículas é conservado [7].

O caso de três corpos não possui solução analítica geral, apenas algumas soluções planares [8]. Porém é possível resolvê-lo numericamente através do MercPert [9] e do Multiple [10].

Para  $n>3$  não há soluções analíticas. Para resolver o problema de  $n$  corpos então é possível também utilizar-se o Multiple, porém o gasto computacional aumenta juntamente com o número de corpos [11]. Assim, chega-se um momento em que se torna inviável utilizar esse método de soma direta da força gravitacional. Devido a isso foram desenvolvidos outros métodos computacionais no intuito de resolver o problema de  $n$  corpos, dentre eles o Algoritmo de Barnes-Hut.

## 2 Objetivos

Os objetivos deste trabalho estão relacionados principalmente com a integração numérica das equações de movimento. Porém também espera-se cumprir os seguintes objetivos:

- Implementar um programa que resolva o problema *dencorpos* de forma eficiente e interativa com o usuário.
- Estudar o problema de Kepler e a dinâmica de sistemas binários.
- Explorar soluções caóticas e órbitas exóticas no problema de 3 corpos.
- Entender a interação de um sistema binário com um terceiro corpo.
- Realizar simulações de sistemas *dencorpos*, descrevendo a evolução de sistemas astrofísicos e cosmológicos.

Além disso pode-se enfatizar o reforço ao conteúdo aprendido na própria universidade [12], em disciplinas tais como Bases Computacionais da Ciência no uso do raciocínio lógico, Processamento da Informação no uso do Java, Fenômenos Mecânicos no uso da física básica, Mecânica Clássica ao tratar da Gravitação, Física Computacional ao tratar dos métodos computacionais, Cálculo Numérico ao tratar dos métodos numéricos, e todas as respectivas disciplinas obrigatórias do eixo da matemática [13] no desenvolvimentos dos cálculos utilizados no decorrer do trabalho. Também pode-se dizer que houve uma introdução em disciplinas não cursadas, tais como Cálculo Vetorial e Tensorial, Mecânica Analítica e Noções de Astronomia e Cosmologia.

## 3 Métodos

Este projeto utiliza conceitos físicos obtidos através da revisão da literatura, e métodos computacionais e numéricos utilizados na integração através de linguagens computacionais.

### 3.1 Linguagens Computacionais

- **Java:**

O programa utilizado Triana [4], desenvolvido pela Universidade Cardiff, foi utilizado no desenvolvimentos dos gráficos de posição, energia e velocidade. Foram feitas algumas modificações no código no intuito de obter mais informações.

- **ShareLatex:**

O editor de texto online ShareLatex [14] foi utilizado na elaboração do presente relatório devido à melhor aparência, principalmente no uso de fórmulas.

- **Gnuplot:**

O programa de linha de comando Gnuplot [15] foi utilizado na plotagem de gráficos em duas dimensões e em animações em gif.

### 3.2 Métodos Computacionais

Os códigos em Java do Triana realizam verificações dos resultados de cada iteração para garantir uma trajetória mais correta. Isso é necessário pois o programa possui sua precisão determinada pelo tamanho do passo temporal inserido pelo usuário. Quanto menor o passo, mais precisa é a órbita gerada, pois para gerá-la considera-se um sistema MRUV, o que não é apropriado, uma vez que as órbitas são elípticas. Porém seus resultados podem ser bem próximos da realidade usando-se passos temporais pequenos o suficiente. São três verificações:

- **Análise do passo temporal:** Se o passo temporal for pequeno suficiente, a variação da velocidade, e conseqüentemente do espaço do corpo em estudo não será muito grande, então a nova aceleração também não deve ser. O usuário insere um valor máximo para o erro relativo do passo temporal, e o programa analisa se o erro de cada iteração realmente é menor do que esse valor inicial. Se não for, o valor  $dt$  do passo tempo será dividido pela metade até que o seja.
- **Método preditor-corretor:** Como a aceleração é considerada constante durante toda a trajetória, uma maneira de garantir que os resultados equivalham a realidade é verificar se a variação de espaço devido a essa consideração está dentro do valor máximo da precisão relativa dada pelo usuário. Para tanto verificamos a variação no espaço considerado MRUV, e se ela for maior do que o erro inicial, então seu cálculo é refeito utilizando uma nova aceleração, calculada entre a média aritmética das acelerações utilizadas para calcular as posições verificadas.
- **Análise angular da órbita:** Para impedir que o programa fique calculando a mesma órbita infinitas vezes, é necessário estabelecer algum método de parada. Este, por sua vez, é feito através do cálculo da tangente do ângulo que a órbita anda em cada iteração. Quando esse valor mudar de sinal, significa que chegou-se a metade da

órbita. Então subtrai-se o equivalente a  $180^\circ$ , e continua-se calculando a tangente do ângulo de cada iteração até que esse mude novamente de sinal. Ao mudar pela segunda vez, significa que completou-se novamente outra semi-órbita, ou seja, que a órbita está completa. Então as iterações quebram, finalizando a execução do programa.

Através do código em Java mencionado acima nota-se também o uso de processos iterativos, ou seja, a repetição de um conjunto de equações através do uso de laços (loops). E com o uso dos mesmos há propagação de erro, sendo assim mostrado o Conceito de Erro nas Análises do passo temporal e no Método preditor-corretor. Neles utiliza-se um valor previamente estipulado pelo usuário para realização dos cálculos abaixo

$$\frac{|x - x_0| + |y - y_0|}{|x_0| + |y_0|} > \text{erro estipulado.} \quad (1)$$



## 4 Desenvolvimento

### 4.1 Método Numérico

Computacionalmente pode-se fazer aproximações suficientemente boas para que seja possível chegar no resultado desejado, principalmente porque não há soluções analíticas para o problema *dencorpos* quando  $n > 3$ , exceto o caso em que há uma rede quadrada infinita em que os corpos, todos com a mesma massa, estão localizados nos vértices dessa rede. As equações de movimento para o problema *dencorpos* são então, no programa Triana, equações do MRUV, cuja aceleração vêm da Lei da Gravitação Universal de Newton - que considera a força que o  $j$ -ésimo corpo de massa  $m_j$  e posição  $r_j$  exerce sobre o  $i$ -ésimo corpo de massa  $m_i$  e posição  $r_i$  -, dada pela equação:

$$\vec{F}_{ij} = - \sum_{j=1}^n Gm_i m_j \frac{\vec{r}_i - \vec{r}_j}{|\vec{r}_i - \vec{r}_j|^3} \quad j \neq i. \quad (2)$$

E pela terceira lei de Newton tem-se que a força que o  $i$ -ésimo corpo exerce no  $j$ -ésimo corpo é  $-\vec{F}_{ij}$ . Essa fórmula é aplicada na sua forma mais simples quando tratada no problema de um e dois corpos, desenvolvidos através do Orbit [2] e do Binary [6], respectivamente, no ambiente Triana. E através do seu desenvolvimento e do uso de equações de movimento uniforme e uniformemente acelerado obtém-se os dados necessários para se desenvolver os diferentes sistemas. No desenvolvimento do problema de 3 àncorpos, a mesma fórmula é utilizada, porém nesses casos o somatório se torna essencial, uma vez que a força que um corpo está sentindo não é mais determinada por apenas um outro corpo, e sim por um conjunto de corpos.

#### 4.1.1 Orbit

Através da igualdade entre a segunda lei de Newton,  $\vec{F}_i = m_i a$ , com sua Lei da Gravitação Universal (equação 2), considerando que o  $i$ -ésimo corpo está localizado no centro do eixo cartesiano ( $r_i = 0$ ), é possível obter a aceleração sofrida pelo  $j$ -ésimo corpo no  $k$ -ésimo passo temporal  $dt$ :

$$\vec{a}_k = - \frac{Gm_j}{r^3} \vec{r}_k. \quad (3)$$

Como no Orbit só há dois corpos, há apenas a força que o corpo central (de maior massa  $m_i$ , sendo portanto o  $i$ -ésimo corpo) exerce no outro corpo em estudo (de menor massa  $m_j$ , sendo portanto o  $j$ -ésimo).

Considerando os parâmetros iniciais inseridos pelo usuário, tais como massa de ambos corpos em estudo, posição e velocidade do  $j$ -ésimo corpo e passo temporal pequeno, é obtida a sua aceleração inicial, e com ela é possível obter a variação da velocidade inicial, calculada através da equação de movimento uniformemente variado  $dv = a dt$ . Assim, em cada  $k$ -ésimo passo temporal  $dt$  é calculada a próxima velocidade  $k+1$  através da  $k$ -ésima aceleração e velocidade, como apresentado na equação abaixo:

$$v_{k+1} = v_k + a_k dt. \quad (4)$$

Tendo o valor da velocidade  $k+1$  é possível obter a nova posição  $k+1$  do corpo. Para tanto primeiro considera-se que o corpo se move em movimento uniforme, assim calcula-se a variação da sua posição pela fórmula  $v dt$ . E então considera-se a variação quando

o corpo move-se em movimento uniformemente acelerado, calculada pela fórmula  $dt \frac{dv}{2}$ . A soma de ambas fórmulas gera a variação da posição sofrida pelo corpo devido a força gravitacional do  $i$ -ésimo corpo, conforme a equação

$$x_{k+1} = x_k + v_{k+1}dt + \frac{a_k dt^2}{2}. \quad (5)$$

Com a nova posição tem-se uma nova aceleração, a aceleração  $k+1$ , e assim uma nova velocidade e mais uma vez uma nova posição, e assim segue a iteração até terminar a órbita. Sabendo todos esses parâmetros também é possível verificar outras características de uma órbita resultante de uma força central, tal como sua excentricidade, que pode ser obtida através de relações simples obtidas da observação da figura 1.

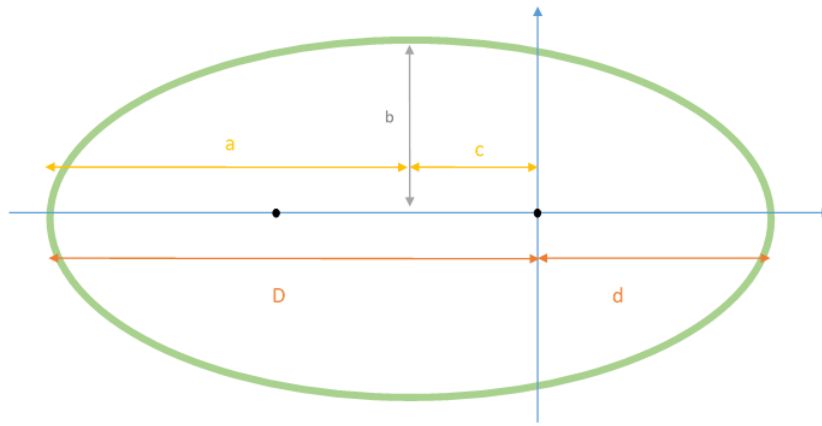


Figura 1: Representação das distâncias  $a$ ,  $b$ ,  $c$ ,  $d$  e  $D$  utilizadas na verificação da excentricidade da órbita. Os pontos pretos na imagem representam os focos da elipse.

- Definindo como  $a$  a distância do centro dos focos das elipse até sua extremidade, distância chama de semi-eixo maior, e  $c$  como a distância do centro dos focos da elipse até um dos focos, temos que a excentricidade  $e$  da órbita vale:

$$e = \frac{c}{a}. \quad (6)$$

- Definindo como  $d$  a distância inicial dada pelo usuário, distância mínima pois ocorre no perihélio (ponto em que o corpo está mais perto do foco em que está localizado o corpo que gera a força), e  $D$  como a distância do centro do eixo até o ponto mais distante dele, distância máxima pois ocorre no afélio (ponto em que o corpo está mais distante do foco), então:

$$D = a + c. \quad (7)$$

- Observando que o comprimento total da órbita equivale a soma das distâncias máxima e mínima da mesma maneira que equivale ao dobro do semi-eixo maior, temos:

$$a = \frac{D + d}{2} \quad (8)$$

Com algumas modificações no código original do Orbit é possível fazê-lo mostrar a excentricidade de cada órbita gerada por ele utilizando as equações 6 e 7.

Outra característica interessante a se obter dessas órbitas é a velocidade que um corpo deve estar para que passe de uma órbita elíptica para uma órbita parabólica, e então hiperbólica. Ao mudar a velocidade de um corpo, mantendo constante todos os seus outros parâmetros, de modo que a energia potencial seja a mesma, o que se altera é a energia total do sistema. Como apresentado pela equação 21, ao alterar a energia do sistema, ocorre a alteração do movimento do corpo sujeito a força central.

É chamada de velocidade orbital a velocidade de um corpo sujeito a força gravitacional (equação 2) cuja órbita é circular, e que portanto está sujeito a uma força centrípeta também. Assim:

$$\begin{aligned}
 F_{centripeta} &= F_{gravitacional} \\
 m \frac{v^2}{r} &= \frac{GMm}{r^2} \\
 v = v_{orbital} &= \sqrt{\frac{GM}{r}}
 \end{aligned} \tag{9}$$

É chamada de velocidade de escape a velocidade com a qual o corpo sai de sua órbita fechada (elíptica) para uma órbita aberta (parabólica), e ela pode ser obtida pela Conservação de Energia do Sistema, como apresenta-se a seguir:

$$\begin{aligned}
 E_{Inicial} &= E_{Final} \\
 U_I + K_I &= U_F + K_F \\
 \frac{-GMm}{r_I} + \frac{mv_I^2}{2} &= \frac{-GMm}{r_F} + \frac{mv_F^2}{2}.
 \end{aligned}$$

Como  $r_F \rightarrow \infty$ , então  $\frac{-GMm}{r_F} \rightarrow 0$ . E conforme  $r_F \rightarrow \infty$ ,  $v_F \rightarrow 0$ . Logo:

$$\begin{aligned}
 \frac{mv_I^2}{2} &= \frac{GMm}{r_I} \\
 v_I &= \sqrt{\frac{2GM}{r_I}} \\
 v_{escape} &= \sqrt{2}v_{orbital}.
 \end{aligned} \tag{10}$$

#### 4.1.2 Binary

Utilizando novamente a mesma equação para a aceleração do Orbit, obtemos não só a aceleração do j-ésimo corpo no k-ésimo passo temporal, mas do i-ésimo também, uma vez que o i-ésimo corpo não está mais fixado no centro do eixo cartesiano. E assim seguimos a mesma lógica anterior, com a k-ésima aceleração calculamos a variação da k-ésima velocidade, considerada uniformemente variada, de ambos corpos. Com a soma da k-ésima velocidade com essa variação temos a velocidade k+1, que gera uma nova posição k+1 para ambos os corpos, calculada pela soma da variação da posição quando considera-se a velocidade uniforme e uniformemente acelerada. Com a nova posição k+1 calcula-se a nova aceleração k+1 e assim segue a iteração até que a órbita dos dois corpos esteja completa.

Com algumas modificações no código original do Binary é possível fazê-lo mostrar como o centro de massa do sistema de dois corpos em estudo se move, através do cálculo de sua posição  $R$  em cada iteração realizada pelo programa, usando a equação:

$$R_{k+1} = \frac{m_i r_i + m_j r_j}{m_i + m_j}. \quad (11)$$

Em que  $m$  representa a massa e  $r$  a posição do  $i$ -ésimo corpo no passo temporal  $k+1$ .

#### 4.1.3 MercPert

Funciona de maneira distinta ao Orbit e ao Binary, pois considera a órbita de dois corpos como círculos para testar o efeito que a força gravitacional de ambos gera no movimento de um terceiro corpo. Isso é chamado de Problema de Três Corpos restrito-plano-circular [1].

#### 4.1.4 Multiple

Funciona de maneira semelhante ao Binary, porém a força que gera o movimento de um dado corpo agora corresponde a soma de força que cada corpo gera nele. É uma força calculada em pares, por isso assemelha-se ao Binary, porém utilizado várias vezes para cada nova dupla de corpos. O nome desse método é Método da Força Bruta [11].

#### 4.1.5 Algoritmo de Barnes-Hut

Esse método é chamado de Método da Árvore, devido a maneira que os corpos são distribuídos no espaço. Utiliza-se na sua execução a posição e velocidade iniciais, além da massa de cada corpo, e o passo temporal. Cada corpo é colocado em seu lugar no espaço, com seus devidos parâmetros, um por um, e cada vez que um novo corpo é inserido, o espaço é dividido em quadrantes (quando se trata de um espaço em duas dimensões) ou em octantes (quando se trata de um espaço em três dimensões), de modo a garantir que cada quadrante/octante contenha no máximo um corpo, podendo haver espaços vazios. De acordo com essa divisão tem-se então a chamada "árvore", que nada mais é do que ramificações nas quais os nós representam os quadrantes. Nós internos são aqueles quadrantes/octantes que possuem mais de um corpo e tiveram que ser subdivididos de modo a garantir que houvesse no máximo um, e cada nó interno armazena o centro de massa dos corpos que nele estão contidos. Nós externos representam os quadrantes/octantes que não possuem ou que possuem apenas um corpo. Para calcular a força em cada corpo é utilizado um parâmetro de aceitação definido pelo usuário, em que, se a razão entre o tamanho do quadrante/octante, representado por um nó interno, e a distância do centro de massa desse quadrante até o corpo que está sentindo a força for menor do que esse parâmetro, é possível considerar o aglomerado de corpos no quadrante/octante como apenas um cuja massa é dada pela soma de todas as massas ali contidas e a posição é dada pela posição do centro de massa desses corpos. Esse cálculo é repetido para  $osncorpos$  colocados inicialmente no espaço, e após calculada a força em um corpo é possível calcular os parâmetros de interesse, tais como as novas posições e velocidades de cada corpo, assim como a energia do sistema. Esses cálculos são repetidos após cada passo temporal, até o encerramento do programa [16] [11].

A diferença mais notável entre o método da Força Bruta (utilizado no Multiple, por exemplo) e o Algoritmo de Barnes-Hut é a diferença no gasto computacional. O primeiro realiza  $n(n-1)$  equações para resolver o problema de  $n$  corpos, e quando  $n \gg 1$ ,

temos aproximadamente  $n^2$  equações sendo feitas a cada passo temporal. O Algoritmo de Barnes-Hut consegue diminuir esse número para  $n \log n$ . A diferença do gasto computacional fica mais obvia ao observar a figura 2, que mostra o que ocorre com ambos casos conforme  $n$  fica maior.

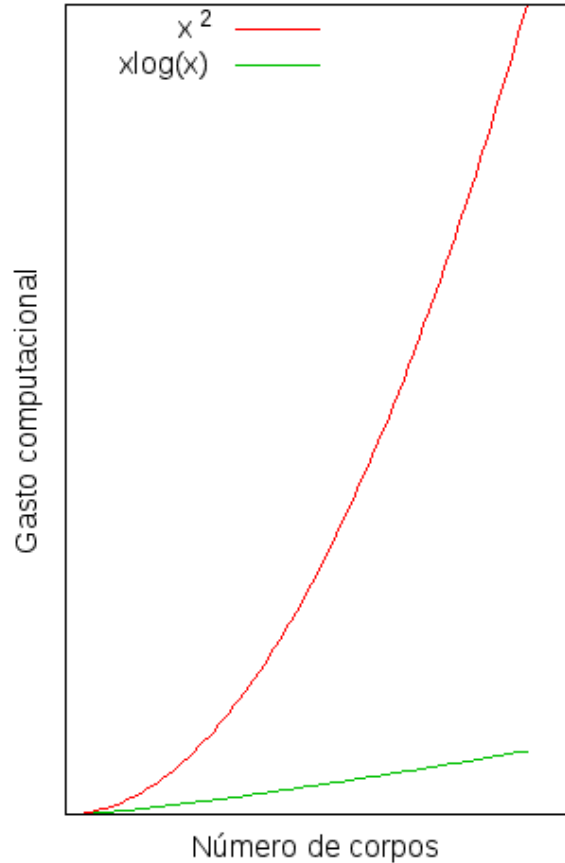


Figura 2: Representação do gasto computacional do método da Força Bruta (cujo gasto é  $n^2$ ) e Algoritmo de Barnes-Hut (cujo gasto é  $n \log n$ ).

## 4.2 Método Analítico

A força gravitacional Newtoniana é uma força atrativa do tipo

$$F(r) = \frac{k}{r^2}, \quad k < 0. \quad (12)$$

No qual  $k = -GMm$ , onde  $M$  e  $m$  são as massas dos corpos considerados e  $G = 6,67e - 11 \frac{Nm^2}{kg^2}$ . Essa força é conservativa, pois seu rotacional é zero ( $\nabla \wedge F(r) = 0$ ) e portanto há uma energia potencial  $V(r)$  associada a ela através da relação

$$F(r) = -\nabla V(r). \quad (13)$$

Onde  $\nabla$  é operado diferencial  $\left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)$ . Assim, a energia potencial obtida é do tipo

$$V(r) = -\frac{k}{r}, \quad k < 0. \quad (14)$$

Utilizando as coordenadas polares e as relações resultantes dessa consideração [17], tais como as da força conservativa

$$\begin{cases} m\ddot{r} = mr\ddot{\theta} - \frac{\partial V(r)}{\partial r} \\ \frac{\partial mr^2\dot{\theta}}{\partial t} = \frac{\partial V(r)}{\partial \theta} \end{cases}, \quad (15)$$

e as relações resultantes da Conservação de Energia do sistema

$$E = \frac{1}{2}m\dot{r}^2 + \left[ \frac{J^2}{2mr^2} + V(r) \right] = T + U_{ef}, \quad (16)$$

onde  $T$  equivale a energia cinética do sistema e  $U_{ef}$  equivale ao potencial efetivo do sistema do seguinte modo:

$$U_{ef} = \frac{J^2}{2mr^2} + V(r) \quad (17)$$

E sabendo que  $\frac{\partial V(r)}{\partial \theta} = 0$ , temos que

$$mr^2\ddot{\theta} = \text{constante}. \quad (18)$$

Essa constante é chamada de Momento Angular  $J$ , e é dessa relação que vem não só a conservação do momento angular para forças centrais  $F(r)$ , mas também a Lei das Áreas de Kepler

$$\begin{aligned} dA &= \frac{rd\theta r}{2} \\ \frac{dA}{dt} &= \frac{r^2\dot{\theta}}{2} = \frac{J}{2m}. \end{aligned} \quad (19)$$

E é através das equações 15 que é possível descobrir  $r(\theta)$ , utilizando a substituição  $u(\theta) = \frac{1}{r(\theta)}$ , que nos leva a equação da trajetória [18]

$$r(\theta) = \frac{\alpha}{\varepsilon \cos(\theta) + 1}, \quad (20)$$

onde  $\alpha = \frac{J^2}{km}$  e

$$\varepsilon = \left( 1 + \frac{2EJ^2}{k^2m} \right)^{\frac{1}{2}}. \quad (21)$$

Integrando a equação 19 tem-se o período de uma órbita elíptica, cuja área é dada por  $A = \pi ab$  onde  $a$  é o semi-eixo maior e  $b$  é o semi-eixo menor. Utilizando a relação  $\sqrt{a}\sqrt{\alpha} = b$  temos:

$$T^2 = \frac{4m\pi^2 a^3}{k} = \frac{4\pi^2 a^3}{GM}, \quad (22)$$

que corresponde a Terceira Lei de Kepler.

Analisando a equação 21, que representa a excentricidade  $\varepsilon$  de uma dada órbita obtida pela equação de trajetória  $r(\theta)$ , temos os possíveis movimentos do corpo [18]:

- Se a Energia do Sistema é positiva ( $E > 0$ ), a órbita possui trajetória hiperbólica, e portanto sua excentricidade é maior do que 1 ( $\varepsilon > 1$ );
- Se a Energia do Sistema é nula ( $E = 0$ ), a órbita possui trajetória parabólica, e portanto sua excentricidade é igual a 1 ( $\varepsilon = 1$ );

- Se a Energia do Sistema está entre  $-\frac{k^2m}{2J^2}$  e 0  $\left(0 < E < -\frac{k^2m}{2J^2}\right)$ , então a órbita possui trajetória elíptica e portanto sua excentricidade está compreendida entre 0 e 1  $(0 < \varepsilon < 1)$ ;
- Se a Energia do Sistema equivale a  $-\frac{k^2m}{2J^2}$   $\left(E = -\frac{k^2m}{2J^2}\right)$ , então a órbita possui trajetória circular e portanto sua excentricidade é nula  $(\varepsilon = 0)$ .

É possível chegar as mesmas conclusões através da observação de um gráfico do Potencial Efetivo versus a posição do corpo, como apresentado na figura 3.

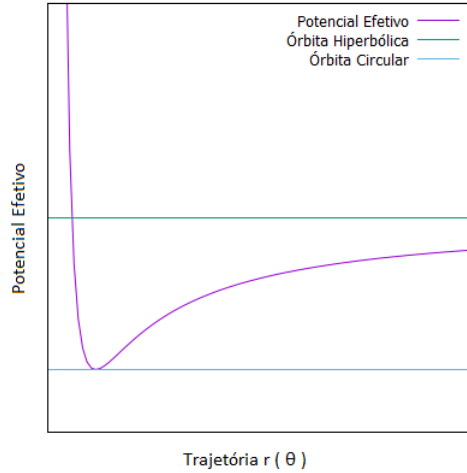


Figura 3: Gráfico do Potencial Efetivo (equação 17) pela posição do corpo.

Também é possível obter a equação de trajetória 20 através das Equações de Lagrange

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_k} \right) - \frac{\partial L}{\partial q_k} = 0 \quad k = 1, \dots, n, \quad (23)$$

onde  $L$  é a lagrangiana ( $L = T - V$  onde  $T$  é a energia cinética do sistema e  $V$  é a energia potencial) e  $q_k$  é a  $k$ -ésima coordenada generalizada [7].

O Problema de Kepler é o estudo do movimento de um corpo devido a um potencial gravitacional, como o movimento de um corpo do sistema solar devido ao Sol, ou o movimento de um satélite devido a Terra [8]. Portanto, os cálculos desenvolvidos até o presente momento correspondem ao Problema de Kepler.

O problema de 2 corpos pode ser obtido através do Problema de Kepler, e portanto as órbitas geradas por ele são chamadas de órbitas keplerianas. Para tanto é necessário utilizar a massa reduzida  $\mu$  no lugar da massa  $m$

$$\mu = \frac{m_1 m_2}{r^2}, \quad (24)$$

onde  $m_1$  pode se referir à massa do menor corpo  $m$  e  $m_2$  à massa do maior  $M$  [19]. Assim, teremos duas equações de trajetória  $r_1(\theta)$  e  $r_2(\theta)$  obtidas da equação 20 com a devida substituição de  $\mu$ .

Como já dito, para o problema de 3 corpos não há solução geral, porém há soluções planares, dentre as quais tem-se a solução de Euler e a solução de Lagrange [8]. Primeiramente, deve-se considerar que as equações de movimento do problema de 3 corpos correspondem a:

$$\begin{cases} m_1 \ddot{r}_1 = F_{12} + F_{13} \\ m_2 \ddot{r}_2 = -F_{12} + F_{23} \\ m_3 \ddot{r}_3 = -F_{23} - F_{13} \end{cases}, \quad (25)$$

onde  $F_{ij}$  é a força gravitacional universal apresentada em 2. Considerando novas variáveis  $s_1 = r_3 - r_2$ ,  $s_2 = r_1 - r_3$  e  $s_3 = r_2 - r_1$ , têm-se:

$$\begin{cases} \ddot{s}_1 = -GM \frac{\vec{s}_1}{s_1^3} + Gm_1 \vec{s}_T \\ \ddot{s}_2 = -GM \frac{\vec{s}_2}{s_2^3} + Gm_2 \vec{s}_T \\ \ddot{s}_3 = -GM \frac{\vec{s}_3}{s_3^3} + Gm_3 \vec{s}_T \end{cases}, \quad (26)$$

onde  $M = m_1 + m_2 + m_3$  e  $\vec{s}_T = \frac{\vec{s}_1}{s_1^3} + \frac{\vec{s}_2}{s_2^3} + \frac{\vec{s}_3}{s_3^3}$ .

É chamada de Solução de Euler aquela em que considera-se que as soluções para as equações de movimento de 3 corpos planas (equações 26) são todas colineares. Essa consideração leva à soluções do tipo  $s_1(t)$ ,  $s_2(t) = \lambda(t)s_1(t)$ ,  $s_3(t) = -(1 + \lambda)s_1(t)$ , onde  $\lambda$  é uma função positiva de  $t$ , e  $\dot{\lambda} = 0$ , o que, através da sua derivação duas vezes e substituição na equação 26, o leva a ser uma raiz positiva de um polinômio de quinta ordem. E o rearranjo das equações 26 leva a equação diferencial

$$\ddot{s}_1 = -GM \frac{m_1 + m_3(1 + \lambda)^2}{(1 + \lambda)^2(m_3 + m_1(1 + \lambda))} \frac{\vec{s}_1}{s_1^3}, \quad (27)$$

cuja solução é semelhante a equação da trajetória  $r(\theta)$  (equação 20) do Problema de Kepler [20]

$$\begin{cases} \vec{r}_1 = \frac{-\alpha[m_2 + \lambda(m_2 + m_1)]}{M(1 + \varepsilon \cos \theta)} \\ \vec{r}_2 = \frac{\alpha[m_3 + m_1(1 + \lambda)]}{M(1 + \varepsilon \cos \theta)} \\ \vec{r}_3 = \frac{\alpha[m_1 \lambda - m_2]}{M(1 + \varepsilon \cos \theta)} \end{cases}, \quad (28)$$

onde

$$\alpha = \frac{J^2}{\left\{ \frac{GM[m_1 + m_3(1 + \lambda)^2]}{(1 + \lambda)^2[m_3 + m_1(1 + \lambda)]} \right\}}$$

e

$$\varepsilon = \sqrt{\frac{2EJ^2}{\left\{ \frac{GM[m_1 + m_3(1 + \lambda)^2]}{(1 + \lambda)^2[m_3 + m_1(1 + \lambda)]} \right\}}}.$$

É chamada de Solução de Lagrange aquelas em que considera-se que as soluções para as equações de movimento de 3 corpos planas (equações 26) são todas iguais em qualquer instante  $t$ . Essa consideração ( $s_1(t) = s_2(t) = s_3(t)$ ) implica que os corpos devem estar sempre nos vértices de um triângulo equilátero, e sua solução novamente é semelhante a equação da trajetórias  $r(\theta)$  (equação 20) do Problema de Kepler. Neste caso, temos que a solução [21]

$$\begin{cases} \vec{r}_1 = \frac{J^2}{2Gm_1^2 M_1 \left( 1 + \sqrt{1 + \frac{2EJ^2}{G^2 m_1^2 M_1^2} \cos \theta} \right)} \\ \vec{r}_2 = \frac{J^2}{2Gm_2^2 M_2 \left( 1 + \sqrt{1 + \frac{2EJ^2}{G^2 m_2^2 M_2^2} \cos \theta} \right)} \\ \vec{r}_3 = \frac{J^2}{2Gm_3^2 M_3 \left( 1 + \sqrt{1 + \frac{2EJ^2}{G^2 m_3^2 M_3^2} \cos \theta} \right)} \end{cases}, \quad (29)$$



onde

$$M_1 = \frac{(m_2^2 + m_3^2 + m_2 m_3)^{\frac{3}{2}}}{M^2}$$

$$M_2 = \frac{(m_1^2 + m_3^2 + m_1 m_3)^{\frac{3}{2}}}{M^2}$$

$$M_3 = \frac{(m_2^2 + m_1^2 + m_2 m_1)^{\frac{3}{2}}}{M^2}.$$

## 5 Resultados

### 5.1 Orbit

Para comprovar as Leis de Kepler foram feitas modificações no código do Orbit da seguinte maneira:

```
// declaração de variáveis
double l1, l2, l3;
    //distâncias
double alfa, seno;
    // angulo entre l1 e l2 e seno desse angulo
double Area, KAl;
    // area contida entre l1, l2 e l3 e taxa de variação dessa area
double [] KAlCoordinate = new double[maxSteps];
    // vetor para armazenar os valores da taxa de variação da area
double max, min;
    // distancia máxima e mínima até o foco
double a, c, e;
    // semi-eixo maior, distancia focal e excentricidade
double p2, R3, K3l;
    // quadrado do periodo, cubo do semi-eixo maior e terceira lei de kepler

for (j = 1; (!fullOrbit && (j < maxSteps)); j++) {
//inicia-se o loop

    ...

    l1 = Math.sqrt((x0*x0)+(y0*y0));
        //distância entre o ponto central e o ponto (x_{k}, y_{k})
    x1 = x0 + dx + ddx1;
        //definição do ponto x_{k+1}
    y1 = y0 + dy + ddy1;
        //definição do ponto y_{k+1}
    l2 = Math.sqrt((x1*x1)+(y1*y1));
        //distância entre o ponto central e o ponto (x_{k+1}, y_{k+1})
    l3 = Math.sqrt(((x0-x1)*(x0-x1))+((y0-y1)*(y0-y1)));
        //distância entre o ponto (x_{k}, y_{k}) e o ponto (x_{k+1}, y_{k+1})
    alfa = Math.acos(((l3*l3) - (l1*l1) - (l2*l2))/(-2*l1*l2));
        // ângulo entre as distâncias l1 e l2
    seno = Math.sin(alfa);
        //seno do ângulo entre as distâncias l1 e l2
    Area = 0.5*l1*l2*seno;
        //área contida entre as distâncias l1, l2 e l3
    KAl = Area/dt1;
        //variação da área contida entre as distâncias l1, l2 e l3 com o tempo
    KAlCoordinate[j] = KAl;
        // armazenamento do valor obtido para variação da área em um vetor
```

```

    if(x1>max){
        max = x1;    // obtenção da distância máxima (afélio)
    }
    if(x1<min){
        min = x1;    // obtenção da distância mínima (perihélio)
    }
    ...

//finaliza-se o loop
}

min = Math.abs(min);
a = (min + max)/2; // definição do semi-eixo maior
c = min - a;      // definição do centro dos focos da elipse
e = c/a;         // calculo da excentricidade

p2 = t*t;        // quadrado do período
R3 = a*a*a;      // cubo do semi-eixo maior
K3l = p2/R3;
// terceira lei de kepler

```

Assim, para comprovar a primeira lei de Kepler o que foi feito foi utilizar as equações 6, 7 e 8. Os resultados obtidos para a excentricidade são apresentados na tabela 1.

Tabela 1: Comparação das excentricidades obtidas através do Orbit com as excentricidades encontradas na literatura dos planetas do Sistema Solar.

Planeta	Excentricidade Numérica	Excentricidade Real
Mercúrio	0.207	0.206
Vênus	0.005	0.007
Terra	0.009	0.017
Marte	0.065	0.093
Júpiter	0.014	0.049
Saturno	0.041	0.053
Urano	0.034	0.046
Netuno	0.007	0.012

Apesar de haver uma grande diferença nos resultados apresentados na tabela 1, é preciso perceber que a excentricidade é um parâmetro sensível. Conforme aumenta-se a ordem de grandeza da excentricidade, mais elípticas são as órbitas, e mais significante se tornam essas mudanças na trajetória do corpo. Portanto, apesar da discrepância dos resultados ser maior na órbita de Júpiter, será mais notável na órbita de Mercúrio, cuja discrepância é menor. As órbitas obtidas foram divididas entre os planetas rochosos e gasosos, de modo a se ter uma melhor visualização. Elas podem ser vistas na figura 4.

Porém ainda é possível visualizar todos os planetas juntos, como mostrado na figura 5.

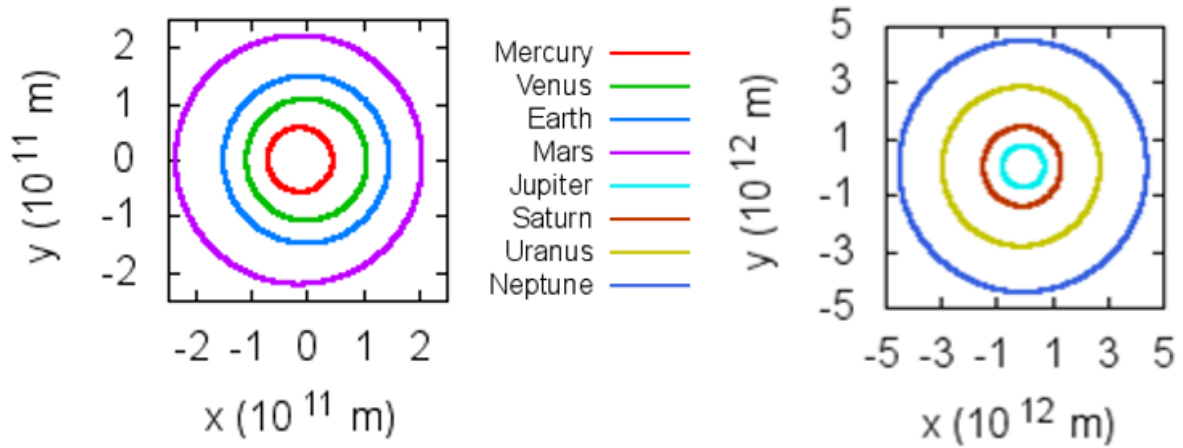


Figura 4: Representação do Sistema Solar dividido entre os planetas rochosos (à esquerda) e gasosos (à direita).

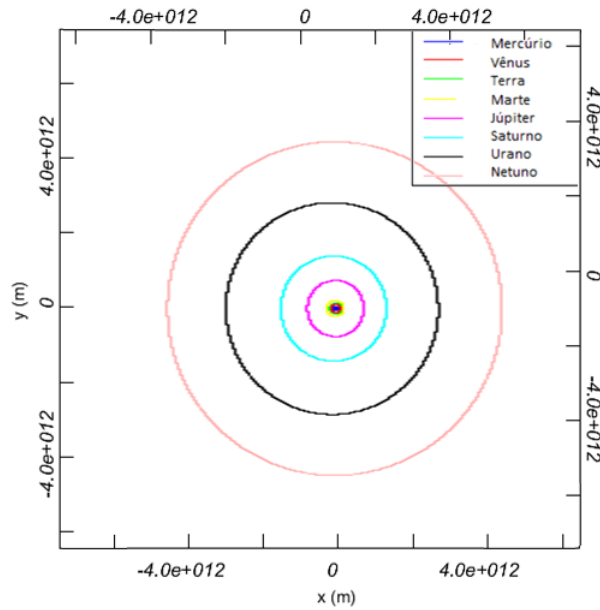


Figura 5: Representação do sistema solar feita através do Orbit.

Na segunda lei de Kepler é preciso calcular a área contida entre pontos consecutivos  $(x_k, y_k)$  e  $(x_{k+1}, y_{k+1})$ , que pode ser aproximada a de um triangulo cujos lado  $a$  e  $b$  correspondem as distâncias destes pontos ao centro, e o ângulo  $\theta$  contido entre eles pode ser obtido através da lei dos cossenos. Assim segue a equação:

$$\begin{aligned}
 A &= \frac{1}{2}ab \sin(\theta) = \\
 &= \frac{1}{2}\sqrt{x_0^2 + y_0^2}\sqrt{x_1^2 + y_1^2} \sin \left( \arccos \left( \frac{((x_k - x_{k+1})^2 + (y_k - y_{k+1})^2) - (x_k^2 + y_k^2) - (x_{k+1}^2 + y_{k+1}^2)}{-2\sqrt{x_k^2 + y_k^2}\sqrt{x_{k+1}^2 + y_{k+1}^2}} \right) \right).
 \end{aligned} \tag{30}$$

A área obtida, em função do tempo percorrido para gerar a órbita de Mercúrio é apresentada na figura 6.

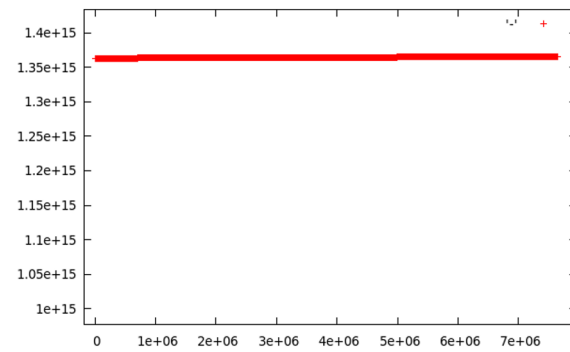


Figura 6: Representação da segunda lei de Kepler em Mercúrio, que afirma que a relação área-tempo permanece constante conforme o planeta percorre sua órbita.

A terceira lei de Kepler é obtida pela divisão do quadrado o período obtido no Orbit pelo cubo do semi-eixo maior também obtido no Orbit. O resultado é apresentado na tabela 2, onde utiliza-se o Erro Relativo Percentual

$$\text{Erro Relativo Percentual} = 100 \frac{|\text{Valor Aproximado} - \text{Valor Exato}|}{\text{Valor exato}}. \quad (31)$$

Tabela 2: Comparação das razões obtidas através do Orbit com as razões encontradas na literatura, relativa a terceira lei de Kepler, dos planetas do Sistema Solar.

Planeta	$\frac{p^2}{a^3}$ Numérico ( $10^{-10} s^2 km^{-3}$ )	$\frac{p^2}{a^3}$ Real ( $10^{-10} s^2 km^{-3}$ )	Erro Relativo Percentual
Mercúrio	2.95	2.98	1 %
Vênus	2.95	2.97	0.7 %
Terra	2.95	2.97	0.7 %
Marte	2.95	2.97	0.7 %
Júpiter	2.96	2.97	0.3 %
Saturno	2.96	2.99	1 %
Urano	2.96	2.98	0.7 %
Netuno	2.96	2.97	0.3 %

E finalmente pode-se alterar a velocidade do corpo a fim de obter órbitas elípticas, parabólicas e hiperbólicas. Órbitas elípticas são todas aquelas cuja velocidade do corpo é menor do que a velocidade de escape (equação 10) e já foram representadas na figura 4 e 5. Órbitas parabólicas são aquelas cuja velocidade do corpo corresponde a velocidade de escape. E órbitas hiperbólicas são as que o corpo se move com velocidade acima da velocidade de escape. Podemos ver cada um dos casos na figura 7, onde também é mostrado como se comporta a energia do sistema em ambos casos.

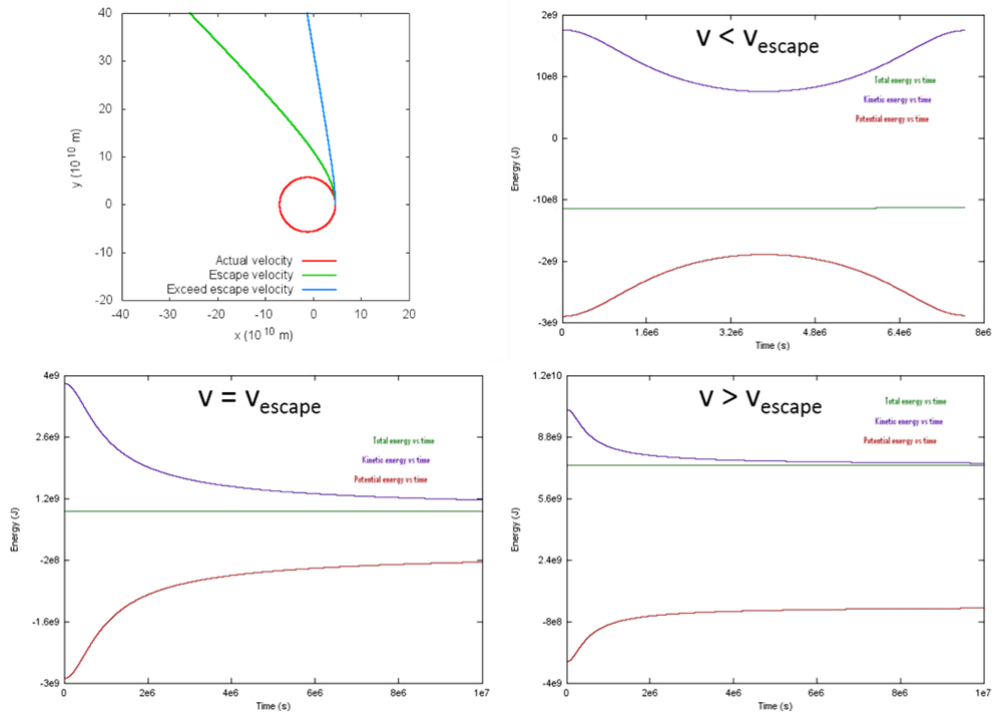


Figura 7: Representação de órbitas elípticas, parabólicas e hiperbólicas, juntamente com suas respectivas energias.

Deve-se notar que, ao aumentar a velocidade de um corpo, mantendo-se constante a força gravitacional que atua nele, aumenta-se a energia do sistema, e portanto a excentricidade da órbita, de acordo com a equação 21. Isso é mostrado nos gráficos da figura 7, onde a energia do sistema é representada pela reta verde, que permanece constante, como esperado para uma força central conservativa, e que aumenta conforme a velocidade inicial do corpo em movimento também aumenta. Além disso, deve-se notar que a velocidade de escape é aquela que leva o corpo até o infinito, ou seja, cuja energia potencial gravitacional tende a zero, como é verificado nos gráficos em que  $v \geq v_{escape}$ , em que a energia potencial é representada por uma linha vermelha, e cujo energia cinética tende a zero, pois a velocidade no infinito também tende a zero, representado pela linha azul nos mesmos gráficos.

No intuito de comparação, também pode-se plotar a equação 20 da órbita de Júpiter, que foi a que teve o resultado mais divergente do esperado, com a encontrada pelo programa Orbit. Essa comparação é mostrada na figura 9. Como já dito acima, a diferença não é muito grande, pois a excentricidade é sensível a ordem de grandeza, e assim sendo, ambas órbitas de excentricidade 0,014 e 0,049 são próximas uma da outra devido a ambas serem aproximadamente círculos.

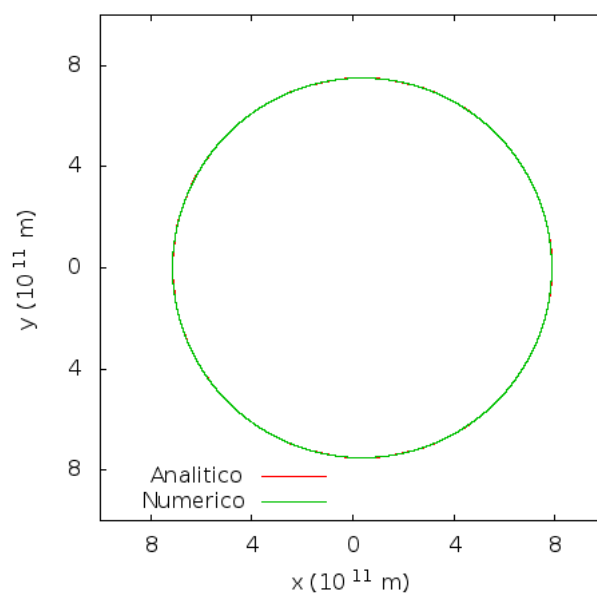


Figura 8: Comparação da órbita de Júpiter obtida pela equação 20 (analítico) com a obtida pelo Orbit (numérico).

Ao verificar a mesma comparação para a órbita de Mercúrio, o que se vê é uma discrepância muito maior, apesar das excentricidades serem muito próximas. O que ocorre é que as órbitas com excentricidade 0,206 e 0,207 são elípticas, e quanto mais elípticas, maiores se verão as diferenças em suas curvas.

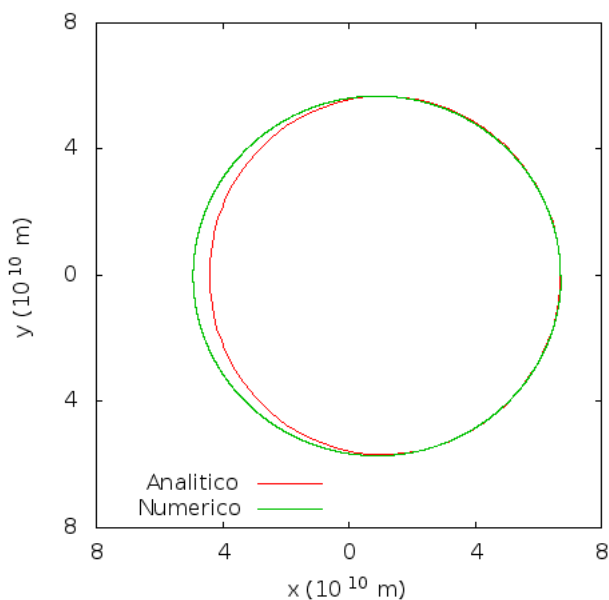


Figura 9: Comparação da órbita de Mercúrio obtida pela equação 20 (analítico) com a obtida pelo Orbit (numérico).

## 5.2 Binary

Através do Binary é possível não só observar o comportamento de 2 corpos, como também como o centro de massa entre eles se comporta. De acordo com o Teorema da Conservação do Momento Linear, quando a força externa total ( $F_{externa}$ ) é zero (válido neste caso, uma vez que a única força atuando no sistema é a força gravitacional), o momento linear total (P) é conservado (P = constante). Uma vez que o centro de massa se move como se a força externa total estivesse agindo na massa inteira do sistema centrada no centro de massa (M), temos:

$$P = \sum_i m_i v_i = \sum_i m_i \frac{dr_i}{dt} = M \frac{dR}{dt}; \quad (32)$$

$$F_{externa} = M \frac{d^2 R}{dt^2} = \frac{dP}{dt} = 0. \quad (33)$$

Conclui-se então que o centro de massa se move com velocidade  $\frac{dR}{dt} = constante$  ou  $\frac{dR}{dt} = 0$  sempre que  $F_{externa} = 0$ , comprovado na figura 10, onde ambos, à esquerda, corpos possuem a mesma massa e mesma velocidade, e o momento resultante é nulo, e, à direita, os corpos possuem massas diferentes e velocidades diferentes, porém o momento resultante ainda permanece nulo, fazendo com que o centro de massa fique parado em um ponto fixo. E comprovado na figura 11, onde deixamos as velocidades e/ou massas diferentes de modo que o momento resultante seja diferente de zero, e o centro de massa se move em MRU (Movimento Retilíneo Uniforme). Para ser possível a observação do centro de massa houve modificação no código da seguinte maneira:

```
// declaração de variáveis
double CMx0 = ((xA0*MA) + (xB0*MB))/(MA+MB);
//definição do centro de massa inicial no eixo x
double CMy0 = ((yA0*MA) + (yB0*MB))/(MA+MB);
// definição do centro de massa inicial no eixo y
double [] CMxCoordinate = new double[maxSteps];
// vetor para armazenar as posições do centro de massa no eixo x
double [] CMyCoordinate = new double[maxSteps];
// vetor para armazenar as posições do centro de massa no eixo x
CMxCoordinate[0] = CMx0;
// armazenamento do centro de massa inicial (eixo x) no vetor
CMyCoordinate[0] = CMy0;
// armazenamento do centro de massa inicial (eixo y) no vetor

...

for (j = 1; (!fullOrbit && (j < maxSteps)); j++) {
// inicia-se o loop

...

CMx0 = ((xA0*MA) + (xB0*MB))/(MA+MB);
// centro de massa do ponto x_{k}
CMy0 = ((yA0*MA) + (yB0*MB))/(MA+MB);
// centro de massa do ponto y_{k}
```



```

...

CMxCoordinate[j] = CMx0;
    // armazenamento do centro de massa do ponto x_{k} no vetor
CMyCoordinate[j] = CMy0;
    // armazenamento do centro de massa do ponto y_{k} no vetor
...

// finalizou-se o loop
}

```

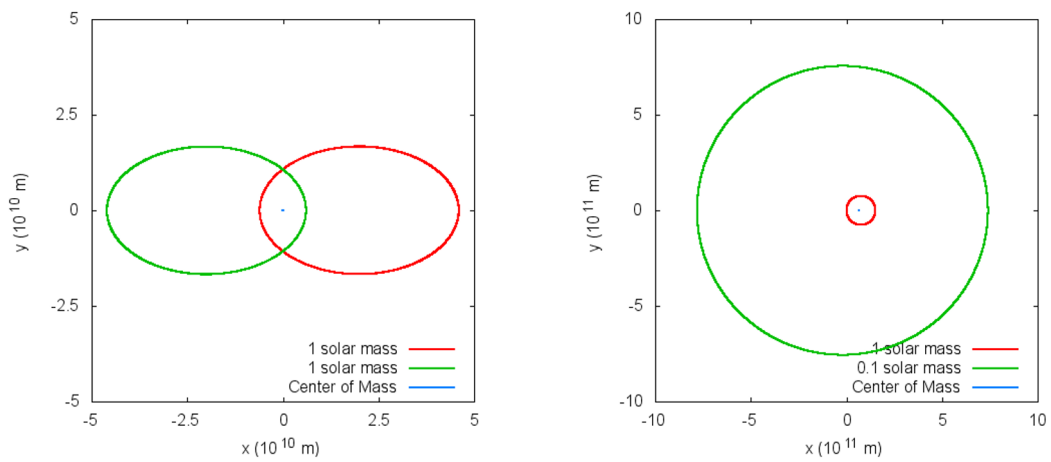


Figura 10: Representação da órbita de dois corpos com massas iguais e velocidades iguais, de modo que o momento resultante seja nulo.

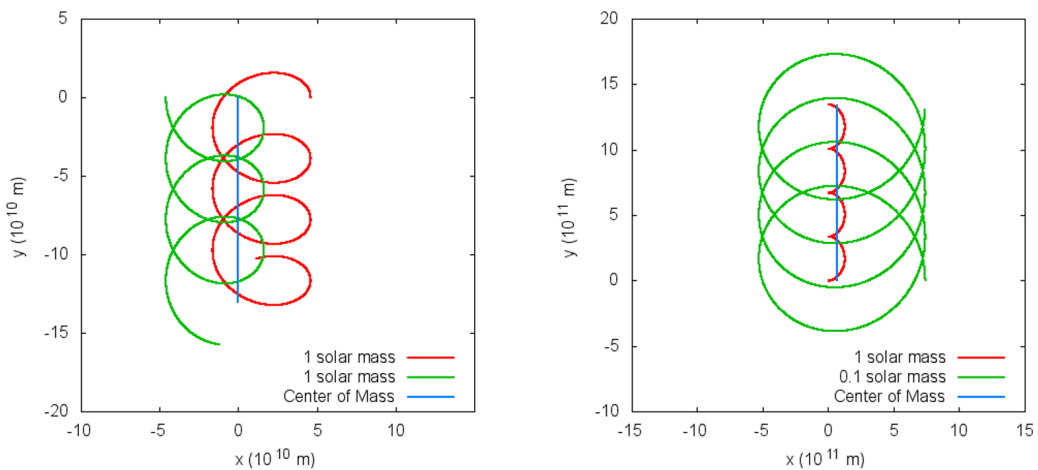


Figura 11: Representação da órbita de dois corpos com massas diferentes e velocidades diferentes, de modo que o momento resultante seja não nulo.

Podemos também verificar um sistema binário no qual um ou ambos planetas usam suas velocidades de escape. No primeiro caso, utilizando dois corpos com massas equi-

valentes à massa do sol ( $2e30$  kg), e colocando como velocidade inicial para o corpo A a sua velocidade de escape ( $v_{escape} = 53852$  m/s) e o corpo B com velocidade de  $13000$  m/s, temos um sistema como representado à esquerda da figura 12, em que o corpo A irá continuar a interagir com o corpo B, porém seguirá uma trajetória ondulatória agora, em vez da "mola" que é apresentada na figura 11. Isso ocorre porque, apesar de um dos corpos possuir velocidade de escape, o outro não possui, e portanto ambos planetas ainda continuam orbitando um ao outro. Se fizermos algo similar, porém colocarmos para ambos os corpos a mesma velocidade de escape, obtemos um sistema como o representado à direita da figura 12, onde ambos corpos não chegam a interagir ao entrarem em contato com o campo gravitacional um do outro.

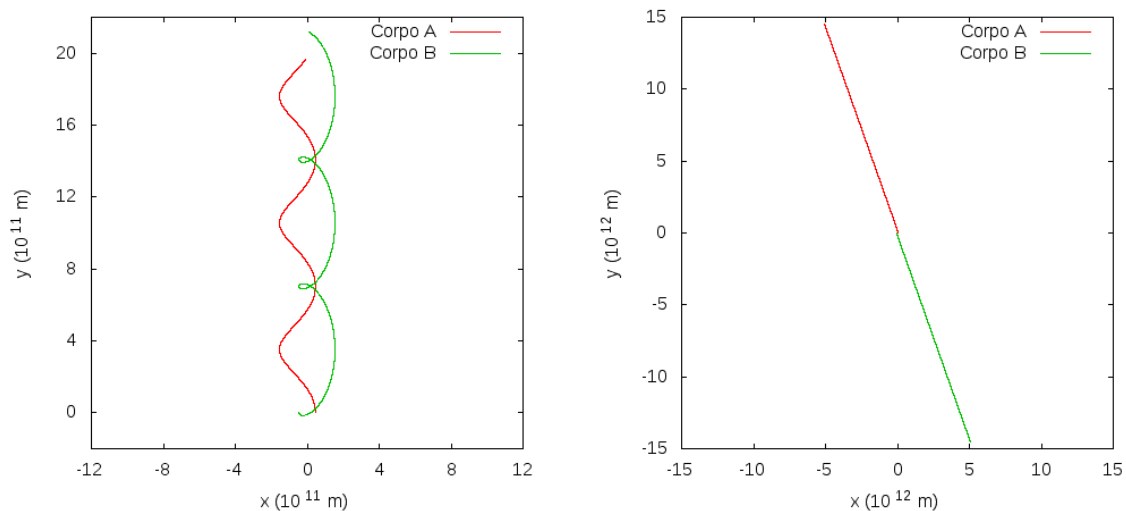


Figura 12: Representação da órbita de dois corpos com massas iguais, onde na primeira um dos corpos assume a velocidade de escape, e na segunda ambos a assumem.

### 5.3 MercPert

O problema de 3 corpos restritos tem muitos usos astrofísicos, e portanto os resultados obtidos por ele são mais comuns do que as outras soluções existentes para o problema de 3 corpos. Tais usos abrangem o movimento de cometas e asteroides no sistema solar e a trajetória de veículos espaciais e sondas lunares, por exemplo. Nos primeiros casos tem-se o Sol e Júpiter como sendo os corpos de órbitas circulares, e no último tem-se a Terra e a Lua [1].

Com o uso do MercPert é possível explorar várias trajetórias randomicamente, como as apresentadas na figura 13, ou mesmo testar trajetórias reais, como a do cometa Halley, apresentada na figura 14 (as condições iniciais utilizadas foram obtidas em [22]). Vale comentar, a partir deste ponto, que o gasto computacional começou a ficar notável, uma vez que cada gráfico gerado leva de 15 a 30 minutos.

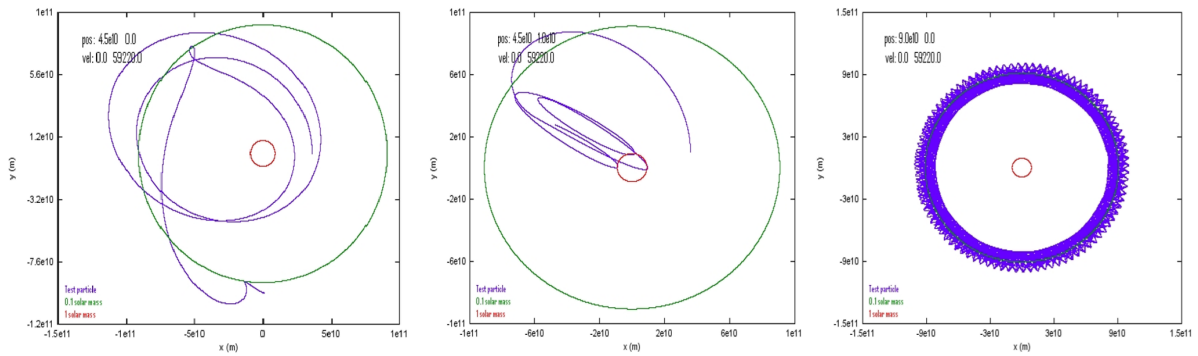


Figura 13: Trajetórias randômicas de um corpo sujeito ao potencial gravitacional de outros dois presos a órbitas circulares.

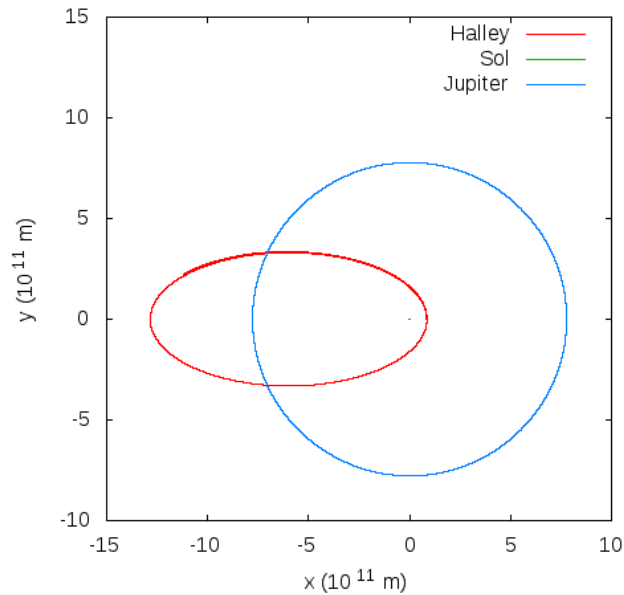


Figura 14: Representação da órbita do cometa Halley feita através do problema de 3 corpos restrito-plano-circular.

## 5.4 Multiple

O Multiple permite não apenas a resolução do problema de 3 corpos, como do problema de  $n$  corpos. Para 3 corpos é possível ver casos em que um sistema binário é formado e o terceiro corpo é ejetado, como apresentado na figura 15. Isso ocorre porque os dois corpos menores liberam energia quando se aproximam um do outro, e essa energia é recebida pelo terceiro corpo mais massivo, que então é expelido [5].

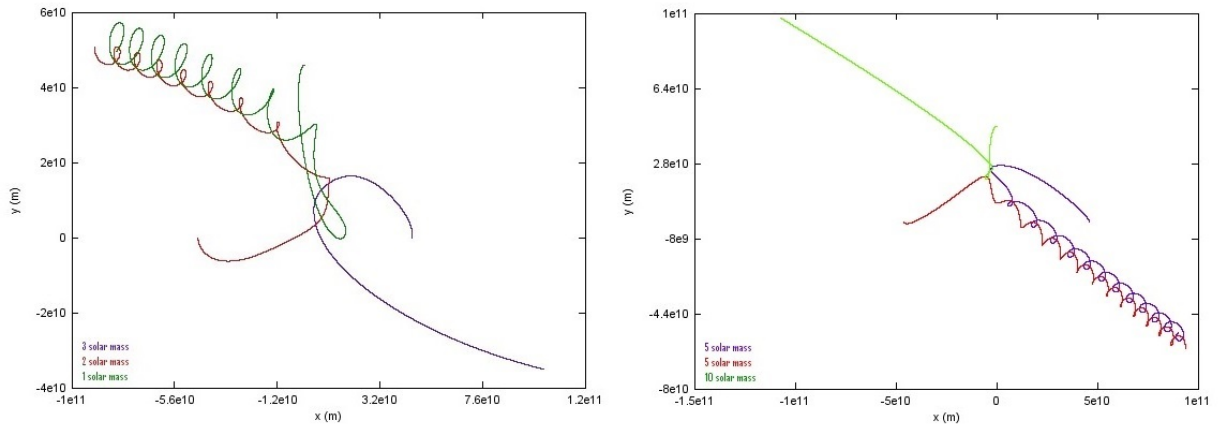


Figura 15: Representação da interação de três corpos onde há a ejeção de um corpo e formação de um sistema binário.

Porém também é possível ver casos em que os corpos fazem coreografias, ou seja, movimentos que se tornam repetitivos conforme o passar do tempo. Para tanto utilizou-se a equação 28 para ver a coreografia obtidas com as Soluções de Euler. As órbitas obtidas são mostradas na figura 16, onde as excentricidades são de 0, 0.8 e 0.8 novamente, com as proporções das massas de 1:2:3 para as duas primeiras, onde  $\lambda \approx 0.89$  [8], e 1:1:1 para a última, onde  $\lambda \approx 1$ , da esquerda para a direita.

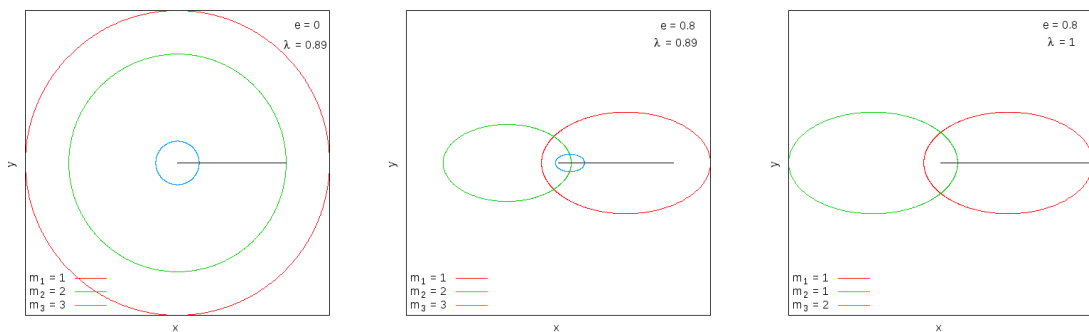


Figura 16: Representação das Soluções de Euler para o problema de 3 corpos planares.

Para as coreografias obtidas com as Soluções de Lagrange utilizou-se a equação 29. As órbitas obtidas são mostradas na figura 17, onde as excentricidades são de 0.8, 0 e 0.8 novamente, com as proporções das massas de 1:2:3 para a primeira e 1:1:1 para as duas últimas, da esquerda para a direita.

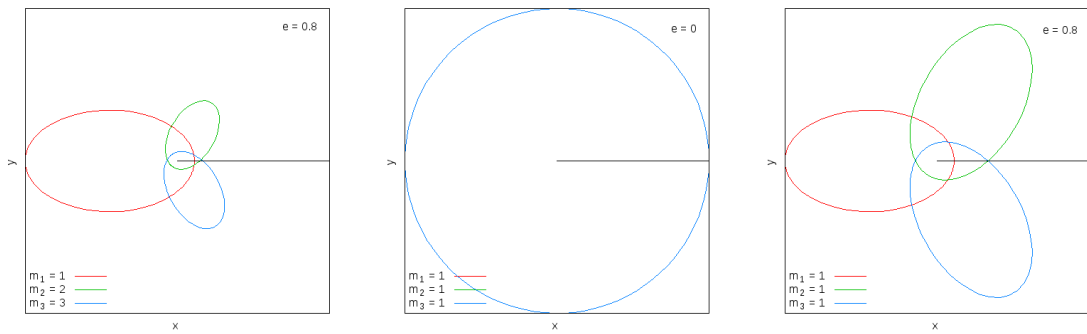


Figura 17: Representação das Soluções de Lagrange para o problema de 3 corpos planares.

## 5.5 Algoritmo de Barnes-Hut

O algoritmo de Barnes-Hut pode ser simulado em [23], uma página online que executa o código de uma maneira bem didática. Para adicionar corpos basta clicar na tela, definindo assim sua localização, e sua massa é definida pelo tamanho da esfera que é adicionada (a tecla e do teclado aumenta a esfera e a letra d diminui). O tamanho do passo temporal também pode ser definido através de uma barra lateral. Além de tudo isso é possível observar, durante a execução do algoritmo, o número de cálculos que seria feito pelo método da força bruta, e o número de cálculos que é feito pelo algoritmo, vendo assim a diferença no tempo computacional de cada um.

Durante a execução desse código várias coisas devem ser notadas, como, por exemplo, o número de cálculos aumenta significativamente no método da força bruta quando o número de corpos aumenta. Ou como, ao adicionar incontáveis corpos com velocidade inicial nula eles sempre se atraem, mostrando a presença apenas da força gravitacional atuando sobre eles, que por sua vez é uma força atrativa. Ou como a influência das condições iniciais que se define para cada corpo, pois o resultado obtido para cada simulação, ao adicionar corpos com diferentes posições e velocidades iniciais, é caótico, porém pode ser obtido novamente ao adicionar as mesmas condições iniciais adicionadas anteriormente. Assim, apesar do movimento dos corpos ser caótico, não é imprevisível.

## 6 Cronograma

Este projeto possui a duração de 12 meses, de 01/08/2014 a 31/07/2015. O projeto deverá ser desenvolvido de acordo com o seguinte cronograma:

- 01/08/2014 a 20/09/2014: Estudo analítico e revisão da literatura
- 01/10/2014 a 30/11/2014: Estudo dos métodos numéricos e elaboração do programa
- 01/12/2014 a 31/01/2015: Testes numéricos iniciais: o problema de Kepler, sistemas binários. Elaboração do Relatório Parcial.
- 01/02/2015 a 31/03/2015: Estudos numéricos do problema de 3 corpos.
- 01/04/2015 a 31/05/2015: Aplicações astrofísicas: interação de sistemas binários com um terceiro corpo
- 01/06/2015 a 3/07/2015: Generalizar os resultados para o problema de n-corpos. Visualização dos resultados, elaboração de vídeos. Relatório Final.

## 7 Conclusão

O estudo de soluções analíticas para o problema de  $n$  corpos exigiu maior atenção, devido aos cálculos envolvidos. Porém, uma vez atingindo o resultado final, a montagem dos gráficos exigiu menos trabalho do que os gráficos feitos através dos métodos numéricos.

O estudo da gravitação mostrou diversas aplicações físicas, desde as mais simples, como órbitas de corpos e satélites presas a uma força central, até casos em que três corpos se separam em dois sistemas. A principal característica que se deve ressaltar em tudo isso é a grande dependência das condições iniciais de cada corpo. É muito fácil cair em órbitas caóticas ou não satisfatórias (devido a um passo temporal muito grande), e o gasto computacional cada vez que isso ocorre é considerável.

Disciplinas como Mecânica Clássica, Física Computacional, e até mesmo Fenômenos Mecânicos, foram essenciais para o desenvolvimento desse projeto. Como a aluna não apresentava conhecimento básico em programação, desenvolveu-se a habilidade de não só editar um código já pronto, como também de utilizar o Gnuplot, um programa de linha de comando, para gerar os gráficos que foram apresentados aqui.

## Referências

- [1] Antonio Fernando Bertachini de Almeida Prado. *Trajelórias Espaciais e Manobras Assistidas por Gravidade*. Instituto Nacional de Pesquisas Espaciais - INPE, 2001.
- [2] Gravity from the ground up: Computer programs orbit. <http://www.gravityfromthegroundup.org/programs/Orbit.html>. Acessado: 2015-01-17.
- [3] Herch Moysés Nussenzveig. *Curso de Física Básica 1 Mecânica*. Editora Edgard Blucher LTDA, 2002.
- [4] Triana: The open source environment. <http://www.trianacode.org/>. Acessado: 2015-01-17.
- [5] Bernard Schutz. *Gravity from the Ground Up*. Press Syndicate of the University of Cambridge, 2005.
- [6] Gravity from the ground up: Computer programs binary. <http://www.gravityfromthegroundup.org/programs/Binary.html>. Acessado: 2015-01-17.
- [7] Nivaldo A. Lemos. *Mecânica Analítica*. Editora Livraria da Física, 2007.
- [8] O problema dos três corpos. <http://cmup.fc.up.pt/cmup/relatividade/3Corpos/3corpos.html>. Acessado: 2015-08-26.
- [9] Gravity from the ground up: Computer programs mercpert. <http://www.gravityfromthegroundup.org/programs/MercPert.html>. Acessado: 2015-08-20.
- [10] Gravity from the ground up: Computer programs multiple. <http://www.gravityfromthegroundup.org/programs/Multiple.html>. Acessado: 2015-02-26.
- [11] Fast hierarchical methods for the n-body problem, part 1. <http://http.cs.berkeley.edu/~demmel/cs267/lecture26/lecture26.html>. Acessado: 2015-08-20.
- [12] Catálogo de disciplinas de graduação edição 2012. [http://prograd.ufabc.edu.br/images/pdf/catalogo\\_de\\_disciplinas\\_2012.pdf](http://prograd.ufabc.edu.br/images/pdf/catalogo_de_disciplinas_2012.pdf). Acessado: 2015-02-26.
- [13] Página de disciplinas de matemática da ufabc. <http://gradmat.ufabc.edu.br/disciplinas/>. Acessado: 2015-02-16.
- [14] Sharelatex. <https://www.sharelatex.com/>. Acessado: 2015-01-17.
- [15] Gnuplot homepage. <http://www.gnuplot.info/>. Acessado: 2015-02-26.
- [16] The barnes-hut algorithm. <http://arborjs.org/docs/barnes-hut>. Acessado: 2015-08-20.
- [17] Cinemática da partícula. [http://www.fis.ita.br/fis14/material\\_didatico/cinematica/principal.htm](http://www.fis.ita.br/fis14/material_didatico/cinematica/principal.htm). Acessado: 2015-08-27.
- [18] Tom W.B. Kibble and Frank H. Berkshire. *Classical Mechanics*. Imperial College Press, 2004.



- [19] Columbia Goldstein, Charles Poole, and John Safko. *Classical Mechanics*. Addison Wesley Publishing Company.
- [20] Introdução ao estudo do problema de n-corpos. [http://www.impa.br/opencms/pt/eventos/downloads/jornadas\\_2006/trabalhos/jornadas\\_antonio\\_fernandes.pdf](http://www.impa.br/opencms/pt/eventos/downloads/jornadas_2006/trabalhos/jornadas_antonio_fernandes.pdf). Acessado: 2015-08-31.
- [21] Three body problem-a newtonian approach. [http://www.academia.edu/3308113/Three\\_Body\\_Problem\\_-\\_A\\_Newtonian\\_Approach](http://www.academia.edu/3308113/Three_Body_Problem_-_A_Newtonian_Approach). Acessado: 2015-08-31.
- [22] Debora A. Katz. *Physics for Scientists and Engineers*. Brooks Cole, 2014.
- [23] Barnes-hut implementation in html/javascript. <http://www.andrew.cmu.edu/user/sameera/demos/BNtree/>. Acessado: 2015-08-28.