Projeto 05: O problema de Kepler

7600017 - Introdução à Física Computacional - 2023/02
13/12/2023

Prof. Dr. José Abel Hoyos
Gabriel de Freitas de Azeredo (11810964)

Resumo

Um dos maiores sucessos da mecânica clássica é explicar o movimento e toda a dinâmica dos corpos celestes do nosso sistema solar. Neste projeto, o objetivo é utilizar todo arcabouço de métodos numéricos desenvolvidos nos projetos anteriores para aplicá-los diretamente no problema de Kepler. Primeiramente, implementando para órbitas circulares, passando para as Leis de Kepler e depois para o movimento de asteroides.

1 Tarefa 1

1.1 Estratégias de resolução

Para este projeto, ao invés de utilizarmos as coordenadas padrões (x, y) para os cálculos, utilizamos a grandeza adimensional $\vec{\rho} = \vec{r}/(UA)$, então a lei de Newton para uma força central gravitacional é

$$\frac{d^2\vec{\rho}}{d\tau^2} = -\frac{\vec{\rho}}{\rho^3} \tag{1}$$

1.1.1 Tarefa a) Velocidades iniciais para órbitas circulares

Para uma órbita circular, temos que a força gravitacional centrípeta é constante em módulo e escrita da forma

$$\frac{M_s v^2}{r} = \frac{GM_s M_p}{r^2}, \tag{2}$$

mas para nossos parâmetros normalizados $v \to \frac{v}{\frac{2\pi(AU)}{ano}}$, então

$$\frac{d\rho}{d\tau} = \frac{ano}{(AU)2\pi}\sqrt{\frac{GM_s}{(AU)\rho}} = \frac{ano}{(AU)2\pi}\sqrt{\frac{4\pi^2(AU)^3}{ano^2(AU)\rho}} = \frac{1}{\sqrt{\rho}} \tag{3}$$

como a velocidade descrita por (3) está em módulo e nossa condição inicial é $\rho_0 = (a, 0)$, temos que a velocidade inicial deve ser radial de forma $\dot{\rho}_0 = (0, 1/\sqrt{a})$.

a	$1/\sqrt{a}$
0.39	1.6
0.72	1.2
1.0	1.0
1.52	0.81
5.2	0.44
9.58	0.32
19.2	0.23
30.1	0.18

Tabela 1: Velocidades iniciais para a tarefa 1a.

A tabela 1 mostra o resultado das velocidades com duas casas decimais.

1.1.2 Tarefa b) Solução numérica para as órbitas

As séries de Taylor ao redor de um ponto t, para frente e para trás são

$$z(t \pm \Delta t) = z(t) \pm \dot{z}(t)\Delta t + \frac{1}{2}\ddot{z}(t)(\Delta t)^2 \pm \frac{1}{6}\dddot{z}(t)(\Delta t)^3 + ... \tag{4}$$

somando as duas equação obtemos $z(t+\Delta t) + z(t-\Delta t) = 2z(t) + \ddot{z}(t)(\Delta t)^2 + O((\Delta t)^4)$, ou seja, para os parâmetros adimensionais, temos que $\ddot{z}(t)$ segue (1). Discretizando a equação e utilizando esse fato, obtém-se para $\vec{\rho} = (x, y)$:

$$x_{i+1} = 2x_i - x_{i-1} - \frac{x}{\rho^3}(\Delta\tau)^2 \tag{5}$$

da mesma forma para y

$$y_{i+1} = 2y_i - y_{i-1} - \frac{y}{\rho^3}(\Delta\tau)^2. \tag{6}$$

Como são necessárias duas posições para o inicio do método, uma iteração com o método de Euler-Cromer será feita antes da iteração por Velvet. As equações para o método de Euler-Cromer já foram dadas e serão implementadas no código diretamente, seguindo o projeto anterior:

$$x_{x,i+1} = v_{x,i} - \frac{\rho_{x,i}}{\rho_i^3}\Delta\tau, \tag{7}$$

$$x_{y,i+1} = v_{y,i} - \frac{\rho_{y,i}}{\rho_i^3}\Delta\tau, \tag{8}$$

para as velocidade. Agora, para as posições temos:

$$\rho_{x,i+1} = \rho_{x,i} + v_{x,i+1}\Delta\tau, \tag{9}$$

$$\rho_{y,i+1} = \rho_{y,i} + v_{y,i+1}\Delta\tau, \tag{10}$$

1.1.3 Tarefa c) Variação do tamanho do passo

A ideia é começar com um passo $\Delta\tau$ inicial e resolver as órbitas com os dois métodos calculando $\delta = \rho_{max}/\rho_{min} - 1$ ao final de cada solução, variando o passo até que o desvio seja menor que 1×10^{-3}. Pelas aproximações feitas em cada método numérico já é possível inferir que o erro cai com $\Delta\tau$ quadraticamente no caso de Velvet e linearmente no caso de Euler-Cromer.

1.1.4 Tarefa d) Energia total em função do tempo

A energia total do sistema é a soma do potencial gravitacional com a energia cinética do planeta (supondo o sol parado). Então

$$E = \frac{1}{2}mv^2 - \frac{GM_s m}{r} \tag{11}$$

para as variáveis adimensionais consideradas nesse projeto temos, novamente, $v = \frac{2\pi(UA)}{ano}\dot{\rho}$ e $r = \rho(UA)$. Substituindo e deixando as constantes $\frac{GM_s m}{(AU)}$ como evidência

$$E = \left(\frac{\dot{\rho}^2}{2} - \frac{1}{\rho}\right)\frac{GM_s m}{(AU)}. \tag{12}$$

Dessa forma obtemos uma expressão para a energia em função das variáveis adimensionais adotadas.

1.1.5 Desenvolvimento do código

Foram implementadas algumas sub-rotinas que aplicam os métodos dadas as condições de iteração máxima (τ_m), tamanho do passo ($\Delta\tau$) e posição inicial do ρ_x, além do número do arquivo de saída. Para plotar as velocidades e as posições apenas foram utilizadas as saídas do programa, após rodar as sub-rotinas em um loop para cada condição inicial em *tarefa-1b.f*.

```
!       Subrotina que implementa o método de velvet.
!
!       nfile = número do arquivo de saída
!       x0 = posição x inicial do planeta
!       dtau = passo de iteração
!       tau_m = tempo máximo de iterações

subroutine ecromer_method(nfile, x0, dtau, tau_m)
implicit real*8(a-h, o-z)

write(nfile,*) "tau x y rho vx vy v"

tau = 0.d0

!       Condições inicias
!
!       rho = (a, 0),
```

```fortran
!       v = (0, 1 / sqrt(a)).

x = x0
y = 0.d0

rho = vec_mod(x, y)

vx = 0.d0
vy = 1.d0 / dsqrt(x)

v = vec_mod(vx, vy)

!       Escreve as condições inicias.

write(nfile,*) tau, x, y, rho, vx, vy, v

!       Iteração no método de Euler-Cromer.

do while (tau .lt. tau_m)

   tau = tau + dtau

   vx = vx - dtau * x / rho**(3.d0)
   x = x + dtau * vx

   vy = vy - dtau * y / rho**(3.d0)
   y = y + dtau * vy

   rho = vec_mod(x, y)
   v = vec_mod(vx, vy)

   write(nfile,*) tau, x, y, rho, vx, vy, v
end do

return
end subroutine ecromer_method
```

E uma subrotina também foi escrita para o método de Velvet.

```fortran
!       Subrotina que implementa o método de velvet.
!
!       nfile = número do arquivo de saída
!       x0 = posição x inicial do planeta
!       dtau = passo de iteração
!       tau_m = tempo máximo de iterações

subroutine velvet_method(nfile, x0, dtau, tau_m)
implicit real*8(a-h, o-z)
```

```fortran
write(nfile,*) "tau x y rho vx vy v"

tau = 0.d0

!       Condições inicias
!
!       rho = (a, 0),
!       v = (0, 1 / sqrt(a)).

x_aux = x0
y_aux = 0.d0

rho = vec_mod(x_aux, y_aux)

vx = 0.d0
vy = 1.d0 / dsqrt(x_aux)

v = vec_mod(vx, vy)

!       Escreve as condições inicias.

write(nfile,*) tau, x_aux, y_aux, rho, vx, vy, v

!       Realiza uma iteração com o método de Euler-Cromer.

tau = tau + dtau

vx = vx - dtau * x_aux / rho**(3.d0)
x = x_aux + dtau * vx

vy = vy - dtau * y_aux / rho**(3.d0)
y = y_aux + dtau * vy

rho = vec_mod(x, y)
v = vec_mod(vx, vy)

write(nfile,*) tau, x, y, rho, vx, vy, v

!       Itera o método de Verlet

do while (tau .lt. tau_m)

   tau = tau + dtau

   x_aux = 2.d0 * x - x_aux - dtau**2.d0 * x / rho**(3.d0)
```

```
        y_aux = 2.d0 * y - y_aux - dtau**2.d0 * y / rho**(3.d0)

        rho = vec_mod(x_aux, y_aux)

        vx = (x_aux - x) / dtau
        vy = (y_aux - y) / dtau

        v = vec_mod(vx, vy)

        write(nfile,*) tau, x_aux, y_aux, rho, vx, vy, v

        aux = x_aux
        x_aux = x
        x = aux

        aux = y_aux
        y_aux = y
        y = aux

    end do

    return
    end subroutine velvet_method
```

Para calcular a energia, implementei a equação (12) em uma função. As mesmas sub rotinas dos métodos foram utilizadas para iterar. Função que calcula a energia do arquivo *tarefa-1d*:

```
    function energy(rho, v)
    implicit real*8(a-h, o-z)

    energy = 0.5d0 * v**2.d0 - 1.d0 / rho
    return
    end function energy
```

Para implementar o gráfico de δ por $\Delta\tau$, implementei um *loop* dentro dos métodos com $\Delta\tau$ sendo dividido por 1.05 em cada iteração, até $\delta < 1 \times 10^{-}3$. O código está em *tarefa-1c.f*

1.2 Resultados

Com as condições iniciais calculadas na *tarefa 1b*, executei o código da *tarefa 2b* para obter as velocidades e órbitas para cada planeta do sistema solar. Com passo de 1×10^{-4} os dois métodos convergiram bem e as velocidades e distâncias do sol se mantiveram constantes como o esperado.

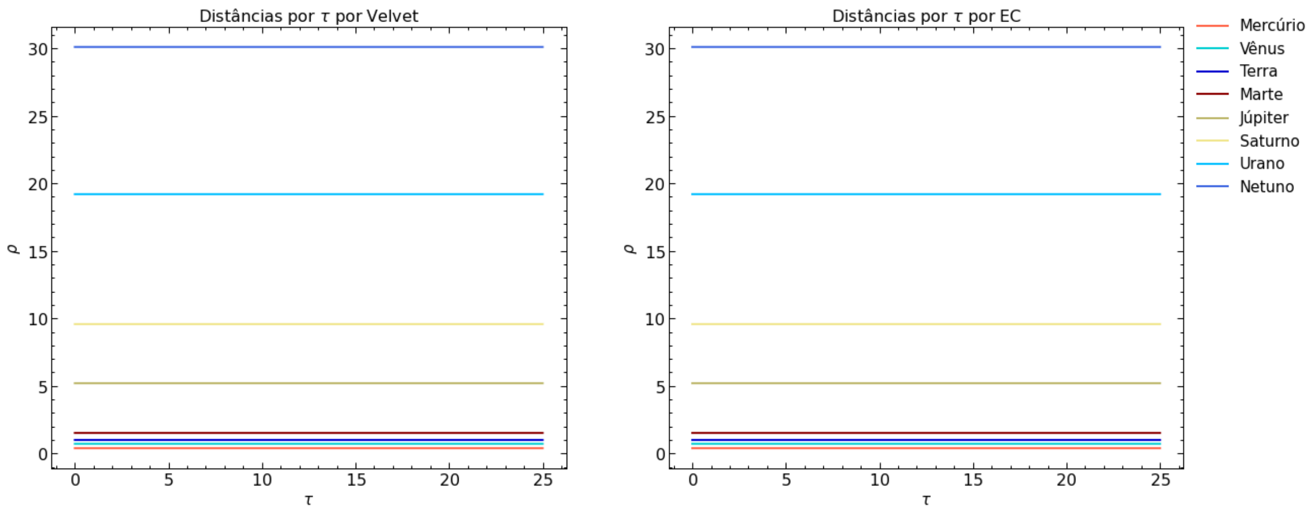

Figura 1: Raio das órbitas circulares para os planetas do sistema solar.

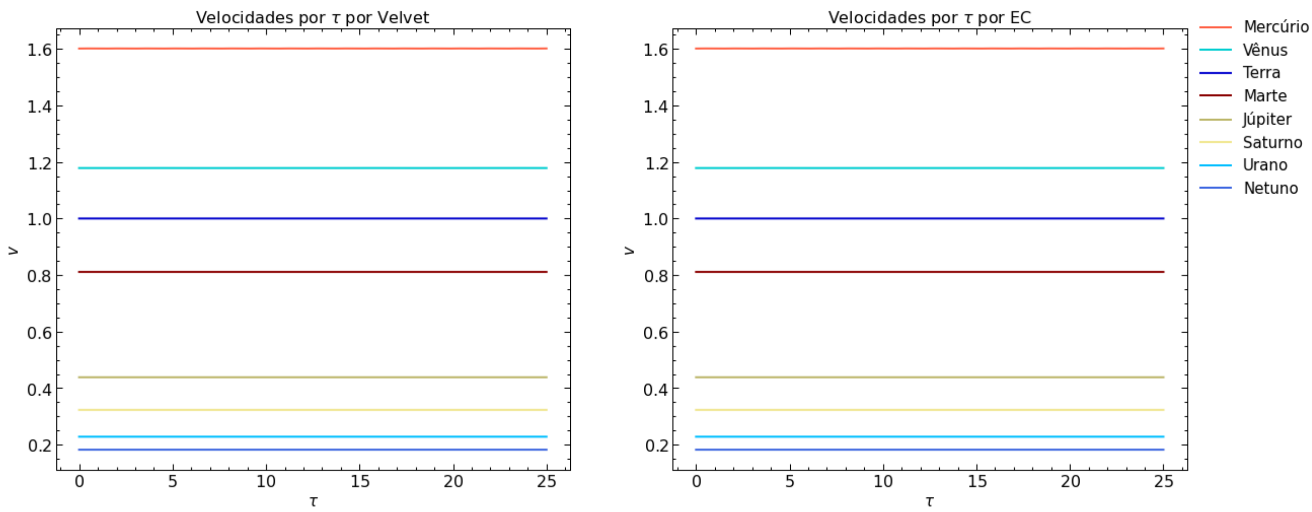

Figura 2: Raio das órbitas circulares para os planetas do sistema solar.

Para as energias, utilizando o mesmo passo e tempo máximo de simulação, obtive as energias. Como esperado, a energia se conserva para os dois métodos nesses parâmetros. A energia negativa é condição para órbitas fechadas, então também está correta. O código executado foi *tarefa-1d.f*.

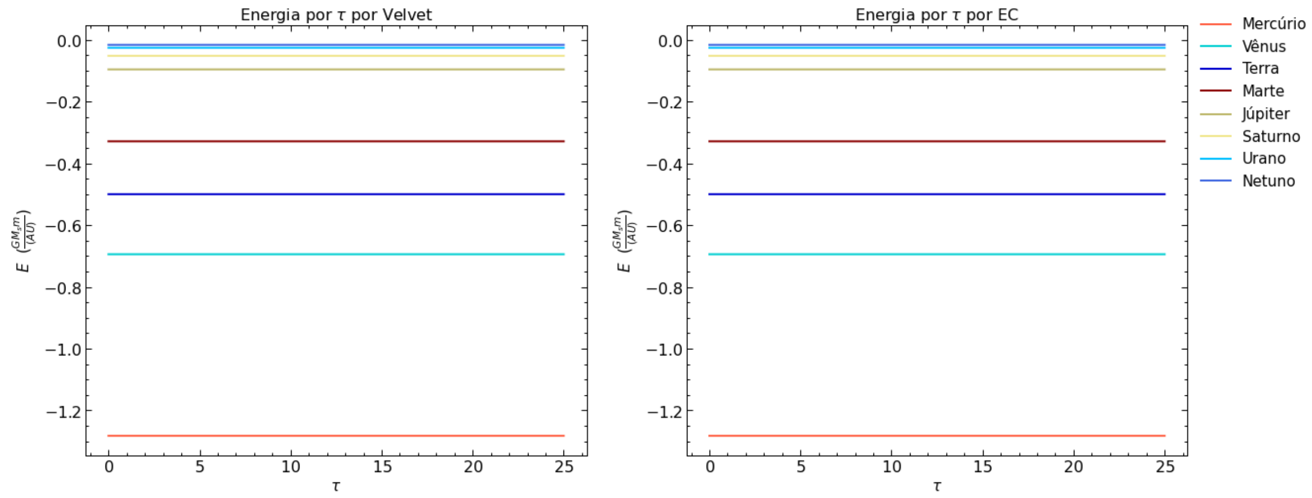

Figura 3: Energia dos planetas do sistema solar.

A curva para δ em função de $\Delta\tau$ para os métodos está abaixo. As duas curvas parecem estranhamente idênticas, mas não consegui encontrar o possível erro a tempo da entrega deste projeto.

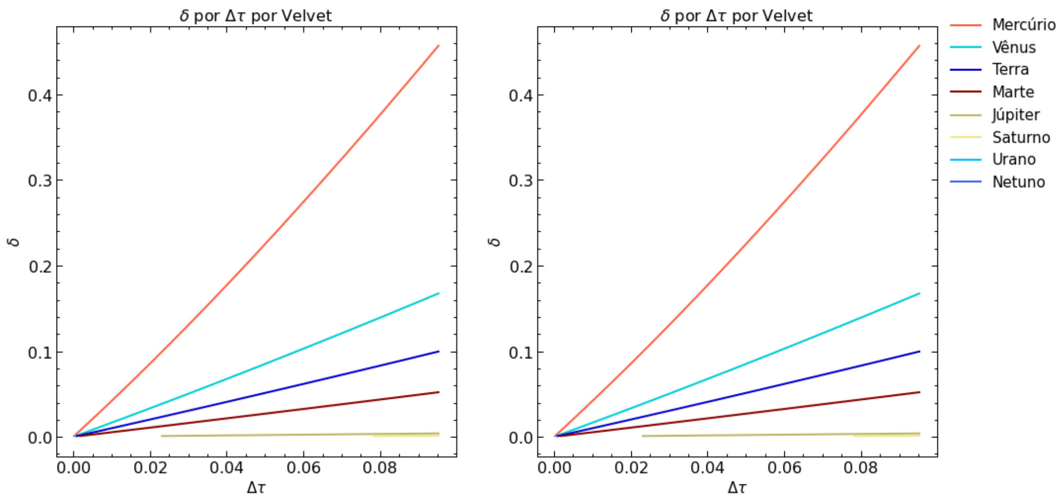

Figura 4: Curva δ em função de $\Delta\tau$.

2 Tarefa 2

2.1 Estratégias de resolução

2.1.1 Tarefa a) Órbitas fechadas

Para as órbitas serem fechadas a energia deve ser negativa. Então a condição que estamos buscando vem da equação (12).

$$E = \left(\frac{\dot{\rho}^2}{2} - \frac{1}{\rho}\right)\frac{GM_s m}{(AU)} < 0 \Rightarrow \rho^2 < \frac{2}{\rho} \tag{13}$$

Ou seja, podemos rodar o programa com velocidade inicial com módulo um pouco maior que a desigualdade acima e as órbitas serão abertas.

2.1.2 Tarefa b) Primeira lei de Kepler

Da definição da elipse tiramos que $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$. Como o sol está no centro do sistema de coordenadas, as órbitas são uma elipse deslocada em x de forma que

$$\frac{(x-c)^2}{a^2} + \frac{y^2}{b^2} = 1, \tag{14}$$

com a e b obtidos, caso a equação acima for respeitada, a órbita é uma elipse.

2.1.3 Tarefa c) Segunda lei de Kepler

A área do triangulo formado entre o sol e dois pontos que o planeta percorre na sua trajetória com diferença de tempo dt é

$$dA = \frac{1}{2}|\vec{r}(t) \times \vec{v}(t)dt|, \tag{15}$$

então a variação da área no tempo é

$$\frac{dA}{dt} = \frac{1}{2}|\vec{r}(t) \times \vec{v}(t)|, \tag{16}$$

utilizando as definições anteriores para ρ, $\dot{\rho}$ e τ, além disso, adotando $A' = A/(AU)^2$, chegamos na relação para os parâmetros adimensionais:

$$\frac{(AU)^2}{\frac{ano}{2\pi}}\frac{dA'}{d\tau} = \frac{2\pi(AU)}{ano}(AU)\frac{1}{2}|\rho_x\dot{\rho}_y - \rho_y\dot{\rho}_x| \Rightarrow \frac{dA'}{d\tau} = \frac{1}{2}|\rho_x\dot{\rho}_y - \rho_y\dot{\rho}_x|. \tag{17}$$

Ou seja, para a segunda lei de Kepler, $\frac{dA'}{d\tau} = cte$. Para nossos valores iniciais temos que o valor esperado para essas constantes são como a equação

$$\frac{dA'}{d\tau} = \frac{1}{2}|\frac{a}{\sqrt{a}}|, \tag{18}$$

e podem ser visualizados para cada planeta na tabela 2.

Planeta	a	$\frac{dA'}{dt}$
Mercúrio	0.39	0.31
Vênus	0.72	0.42
Terra	1.0	0.50
Marte	1.52	0.62
Júpiter	5.2	1.14
Saturno	9.58	1.55
Urano	19.2	2.19
Netuno	30.1	2.74

Tabela 2: Taxa de variação de A' para os valores de condição inicial.

2.1.4 Tarefa d) Terceira lei de Kepler

A terceira lei de Kepler fala que dado o semi-eixo maior da órbita a, a razão $\frac{T^2}{a^3}$ é constante para todos os planetas. O semi-eixo já será obtido para as tarefas anteriores, então o desafio é encontrar os períodos para os planetas.

2.1.5 Desenvolvimento do código

Os códigos em grande parte já foram desenvolvidos na tarefa anterior, a sub rotina do método de Velvet foi inteiramente aproveitada. A função que implementa $\frac{dA'}{d\tau}$ no arquivo *tarefa-2b.f* é:

```
function dA_dt(x, y, vx, vy)

real*8 :: x, y, vx, vy, dA_dt

dA_dt = 0.5d0 * abs(x * vy - vx * y)

return
end function dA_dt
```

Para calcular os períodos utilizei a técnica de salvar t_n tal que y é 0. Então a relação é $\frac{T}{2} = t_n$. Não foi realizada a média de períodos conforme os planetas transladam, pois planetas com órbitas maiores exigiriam ainda mais tempo de simulação para obterem o mesmo número de períodos dos menores. Para achar o semi-eixo maior a, uma relação semelhante foi implementada tal que $a = (x - x_0)/2$, sendo x o valor tal que y respectivo é 0 (da outra extremidade da elipse). Por fim, b é o maior y obtido no código. A condição que foi implementada também no arquivo *tarefa-2b.f* é:

```
      if (y_aux .gt. b) then
         b = y_aux
      end if

      if ((abs(y_aux) .lt. 1.d-4).and.(x_aux .lt. 0.d0).and.
     &   (a .eq. 0.d0)) then

         tn = tau
         a = (x0 - x_aux) / 2.d0

      end if
```

2.2 Resultados

Sobre órbitas abertas e fechadas, rodei basicamente o código da tarefa anterior, porém utilizando como condição inicial para $\dot{\rho}_y = \sqrt{2/\rho} + 1 \times 10^{-4} > \sqrt{2/\rho}$. Ou seja, não respeita a desigualdade para órbitas fechadas deduzida. Dessa forma, obtive as seguintes trajetórias para alguns planetas do sistema solar:

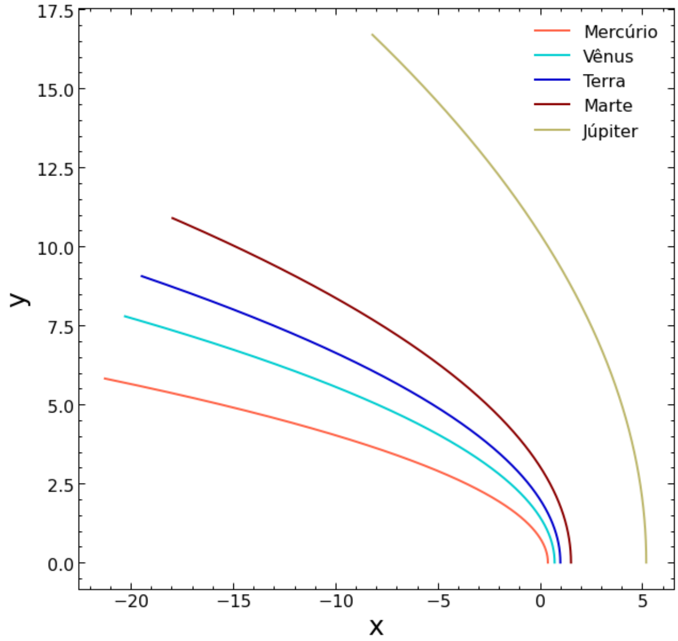

Figura 5: Órbitas abertas para os planetas do sistema solar.

Para a primeira lei, os resultados foram próximos do esperado para órbitas elípticas para todos os planetas:

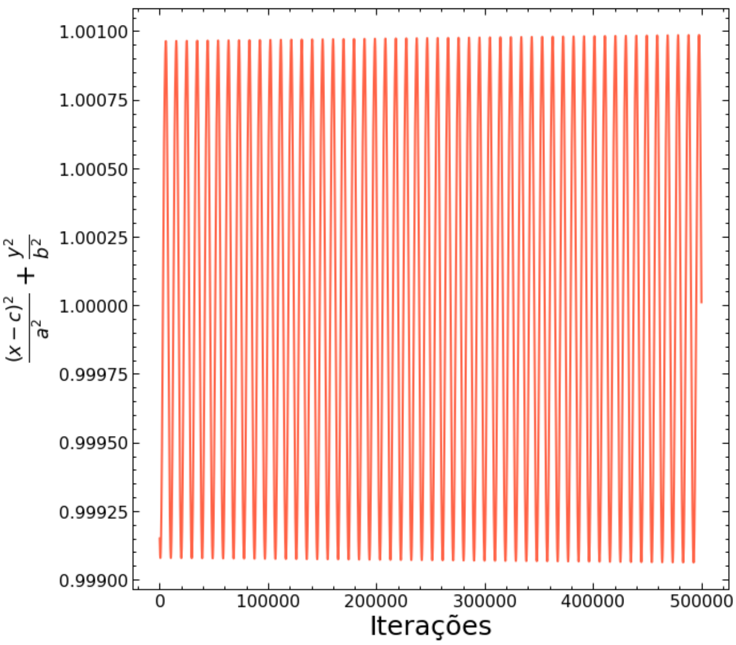

Figura 6: Equação da elipse para os parâmetros de Mercúrio.

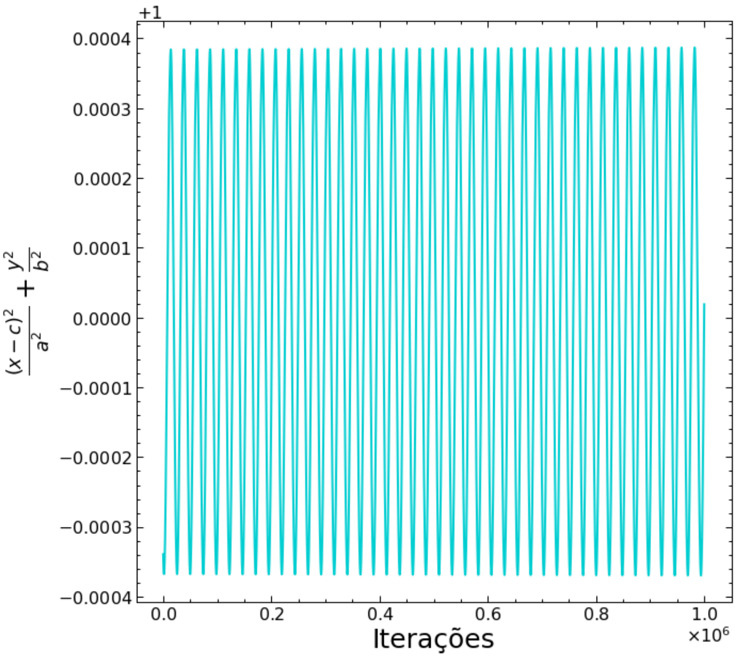

Figura 7: Equação da elipse para os parâmetros de Vênus.

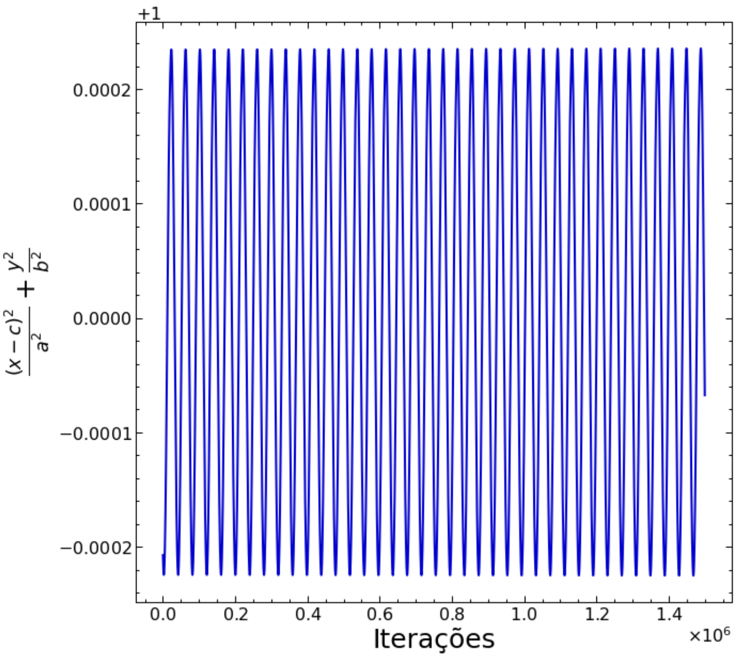

Figura 8: Equação da elipse para os parâmetros da Terra.

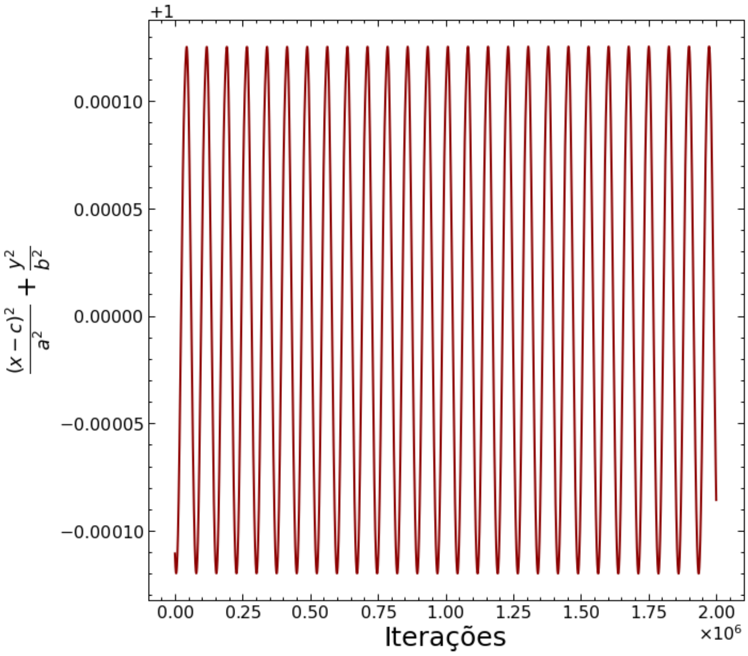

Figura 9: Equação da elipse para os parâmetros de Marte.

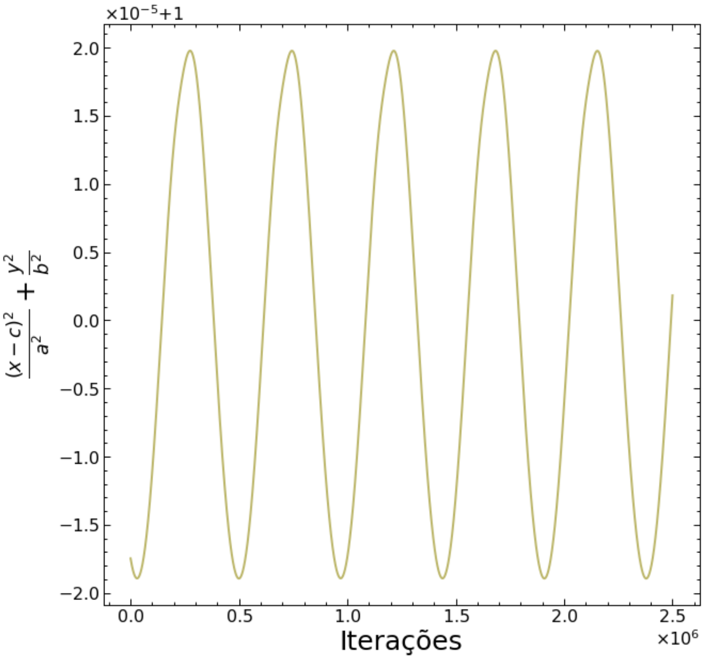

Figura 10: Equação da elipse para os parâmetros de Júpiter.

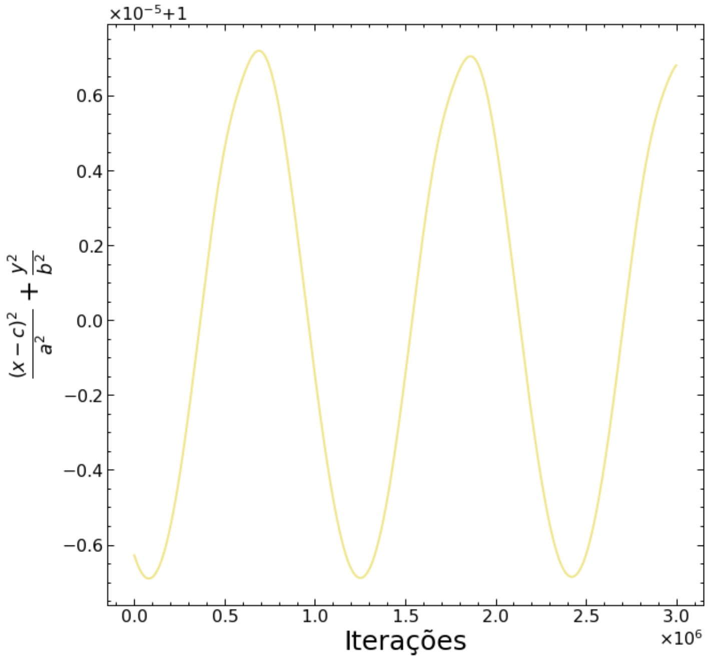

Figura 11: Equação da elipse para os parâmetros de Saturno.

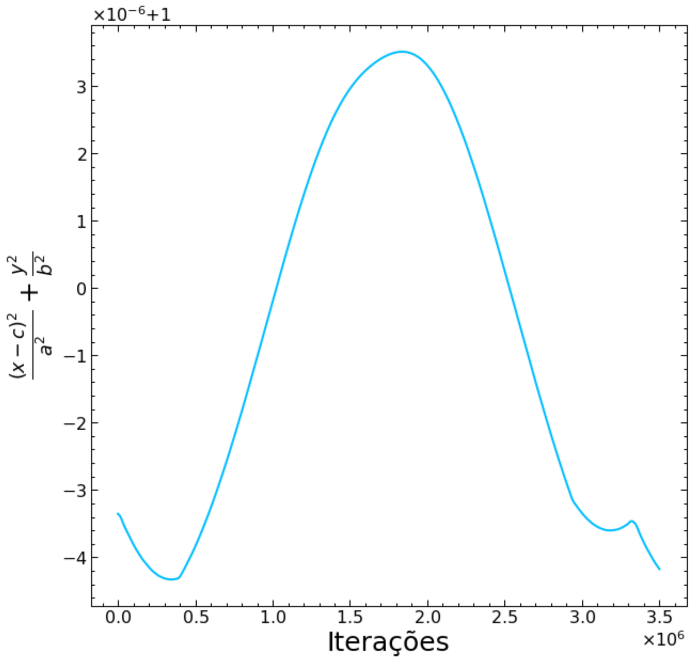

Figura 12: Equação da elipse para os parâmetros de Urano.

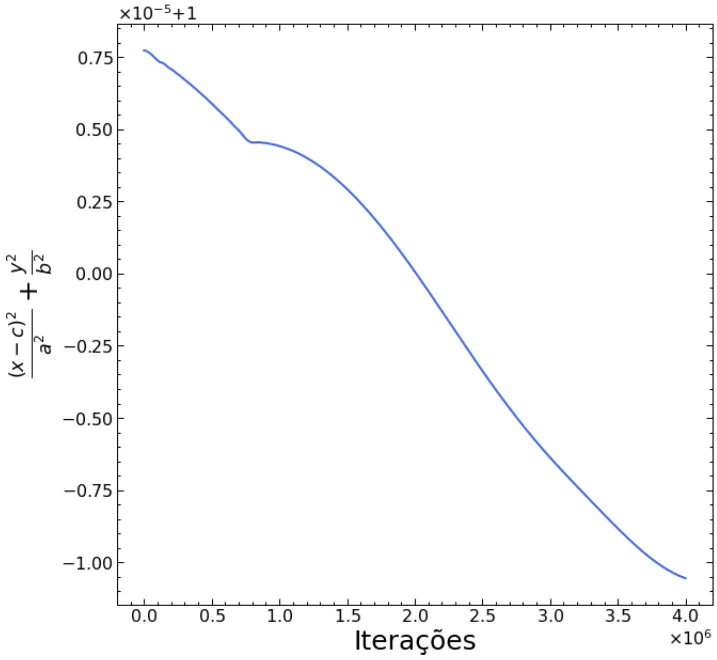

Figura 13: Equação da elipse para os parâmetros de Netuno.

Mesmo oscilatório, todos os estão muito próximos de 1, as órbitas são elípticas. Para a segunda lei, obtive resultados consonantes com o esperado na tabela 2. A taxa de variação de área varrida com tempo é constante.

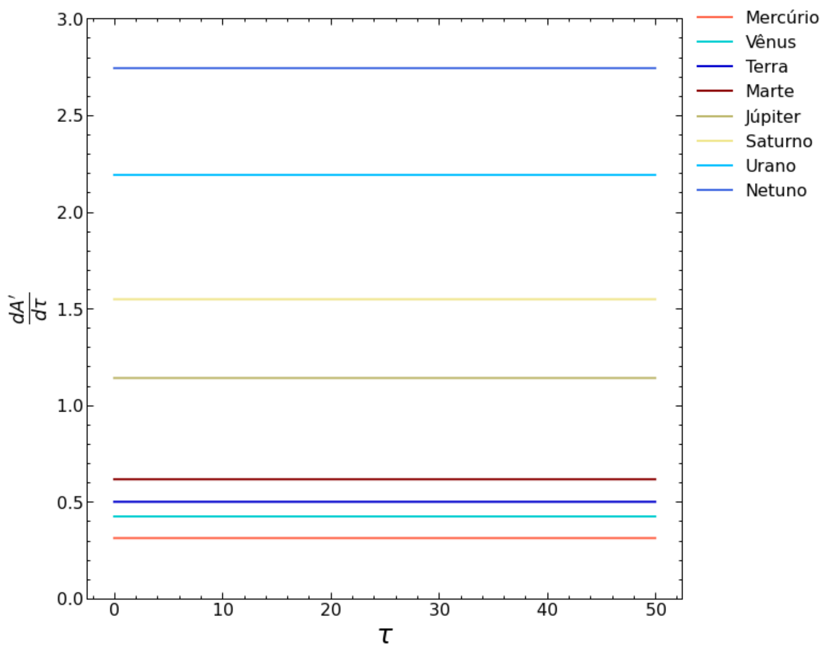

Figura 14: $\frac{dA'}{d\tau}$ para os planetas do sistema solar.

Os períodos obtidos do terminal para os planetas foram expostos na tabela abaixo. A relação de Kepler é respeitada para todos os planetas dando aproximadamente $T^2/a^3 = 15,69$.

T	a
0.9646	0.39
2.42	0.72
3.961	1.0
7.424	1.52
46.98	5.2
117.5	9.58
333.3	19.2
654.2	30.1

Tabela 3: Períodos obtidos com os respectivos semi-eixos maiores.

3 Tarefa 3

3.1 Estratégias de resolução

3.1.1 Tarefa a)

Das equações para $\ddot{\rho}$ dadas no projeto, utilizando a definição do método de Velvet expostas nas tarefas anteriores, para a Terra:

$$\rho_{x,i+1} = 2\rho_{x,i} - \rho_{x,i+1} + \left(a_{T,S,x} + \frac{M_J}{M_s} a_{T,J,x}\right)(\Delta\tau)^2, \tag{19}$$

$$\rho_{y,i+1} = 2\rho_{y,i} - \rho_{y,i+1} + \left(a_{T,S,y} + \frac{M_J}{M_s} a_{T,J,y}\right)(\Delta\tau)^2 \tag{20}$$

e para Júpiter

$$\rho_{x,i+1} = 2\rho_{x,i} - \rho_{x,i+1} + \left(a_{J,S,x} - \frac{M_T}{M_s} a_{T,J,x}\right)(\Delta\tau)^2, \tag{21}$$

$$\rho_{y,i+1} = 2\rho_{y,i} - \rho_{y,i+1} + \left(a_{J,S,y} - \frac{M_T}{M_s} a_{T,J,y}\right)(\Delta\tau)^2, \tag{22}$$

onde $a_{1,2,\alpha} = -\frac{(\rho_{1,\alpha} - \rho_{2,\alpha})}{|\rho_1 - \rho_2|^3}$