Projeto 4 - Movimento Oscilatório

Instituto de Física de São Carlos
Universidade de São Paulo

Gabriel Lima Alves (12558547)

Introdução à Física Computacional
Prof. Francisco Castilho Alcaraz

Dezembro, 2022

Tarefa A

Na tarefa A é pedido para resolver numericamente o movimento de um pêndulo simples, sem forças externas. Para isso foi utilizado duas discretizações diferentes das equações do movimento de um pêndulo. O primeiro é método de Euler que esta demonstrado abaixo:

$$\omega_{i+1} = \omega_i - \frac{g}{l}\theta_i \Delta t$$

$$\theta_{i+1} = \theta_i + \omega_i \Delta t$$

O segundo modo de calculo utilizado é o método de Euler-Cromer, muito similar ao anterior, porém com uma pequena correção:

$$\omega_{i+1} = \omega_i - \frac{g}{l}\theta_i \Delta t$$

$$\theta_{i+1} = \theta_i + \omega_{i+1} \Delta t$$

A partir, dos dados obtidos dos dois métodos foi feito um gráfico que descreve a oscilação de θ em função do tempo e a energia do sistema ao longo do movimento.

O calculo da energia foi feito utilizando a seguinte equação:

$$E = \frac{1}{2}mv^2l^2 + mgl \cdot \cos\theta$$

O algoritmo utilizado para a encontrar a solução do movimento do pêndulo pode ser visto a seguir.

```fortran
c     program tarefa-A
      implicit real*8 (a-h,o-z)
      Parameter(nite = 1000d0)
      Parameter(tempo = 50d0)!segundos
      Parameter(rpi = 2d0*dacos(-1d0))
      Parameter(rg = 9.8d0)
      Parameter(rl = 9.8d0)
      Parameter(m = 1.0d0)

      Parameter(isaid1 = 10)
      Parameter(isaid2 = 20)
      open(unit=isaid1,file='tarefa-1-saida-1-12558547.dat')
      open(unit=isaid2,file='tarefa-1-saida-2-12558547.dat')

      deltat = (tempo*1d0)/(nite*1d0)

      theta1 = rpi/48.0d0
      omega1 = 0.0d0

      theta2 = theta1
      omega2 = omega1

```

```
      ener1 = -m*rg*rl*dcos(theta1)
      ener2 = ener1

      do i = 1, nite

c     calculo de omega \ Método de Euler
      omega1_p1 = omega1 - (rg/rl)*theta1*deltat
c     calculo de theta \ Método de Euler
      theta1_p1 = theta1 + omega1*deltat

c     calculo de omega \ Método de Euler-Cromer
      omega2_p1 = omega2 - (rg/rl)*theta2*deltat
c     calculo de theta \ Método de Euler-Cromer
      theta2_p1 = theta2 + omega2_p1*deltat

      ener1 = m*rg*rl*dcos(theta1_p1)
     ++(1d0/2d0)*m*(omega1_p1**2d0)*(rl**2d0)
      ener2 = m*rg*rl*dcos(theta2_p1)
     ++(1d0/2d0)*m*(omega2_p1**2d0)*(rl**2d0)

      write(isaid1,*) i*deltat , theta1_p1, theta2_p1
      write(isaid2,*) i*deltat , ener1, ener2

      theta1 = theta1_p1
      omega1 = omega1_p1
      theta2 = theta2_p1
      omega2 = omega2_p1

      end do
      close(isaid1)
      close(isaid2)
      end
```

Algoritmo 1: Código para resolução da tarefa A

Resultados

A seguir está o gráfico de θ (rad) em função do tempo gerado a partir da solução encontrada pelo algoritmo.

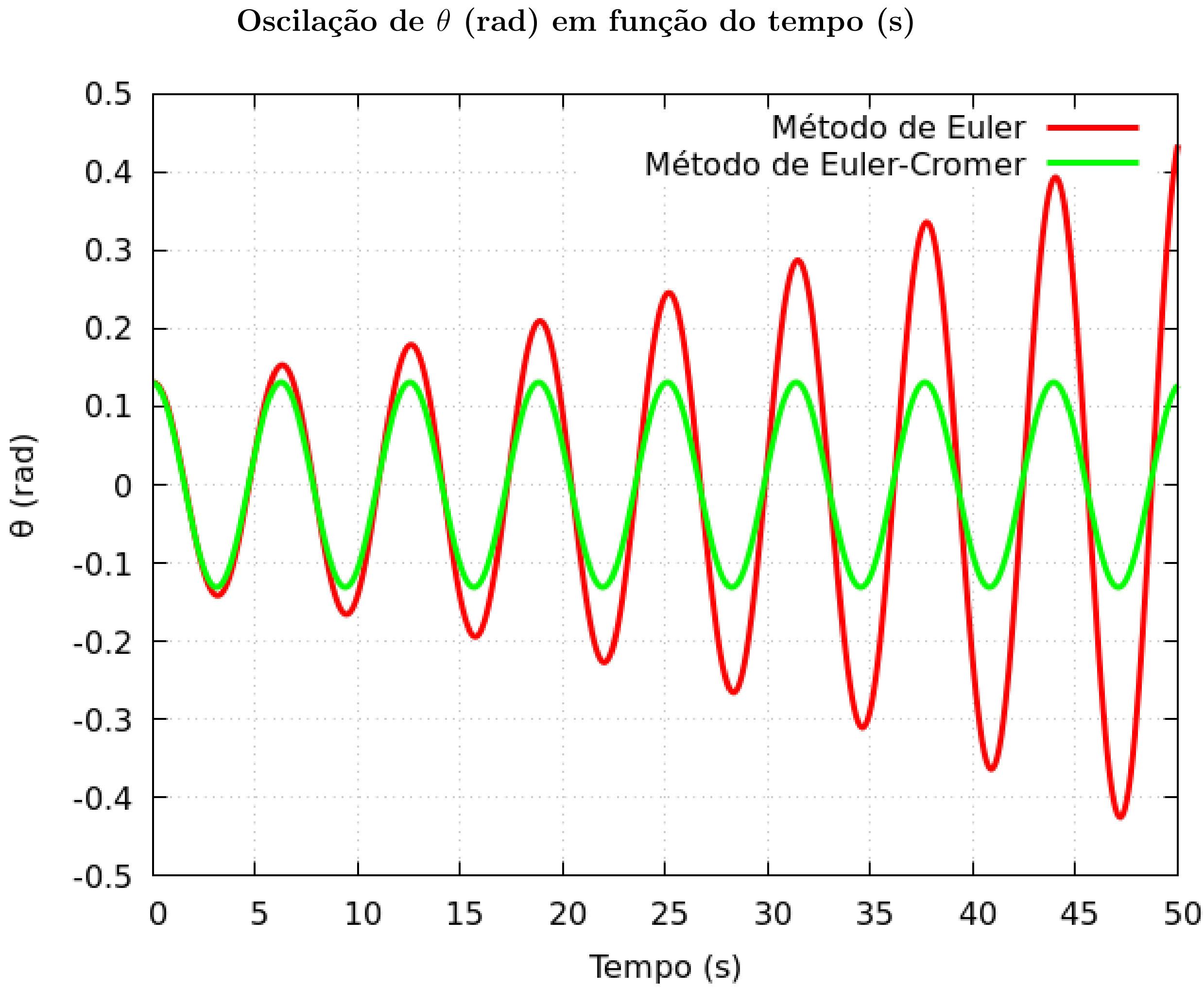

Figura 1: Valores de θ em função do tempo t utilizando os métodos de Euler e Euler-Cromer

Observando o gráfico acima é possível perceber uma inconsistência, visto que no método de Euler a amplitude de variação de θ aumenta ao longo do tempo, o que não deveria acontecer em um sistema isolado. Em períodos de tempo menores o método não apresenta um erro grande, mas em períodos maiores fica facilmente aparente. Esse erro é insignificante no método Euler-Cromer, pois esse método corrige o valor a cada oscilação mantendo a consistência para períodos maiores.

Logo em seguida, foi feito o gráfico da energia do sistema em cada ponto do comento do pêndulo.

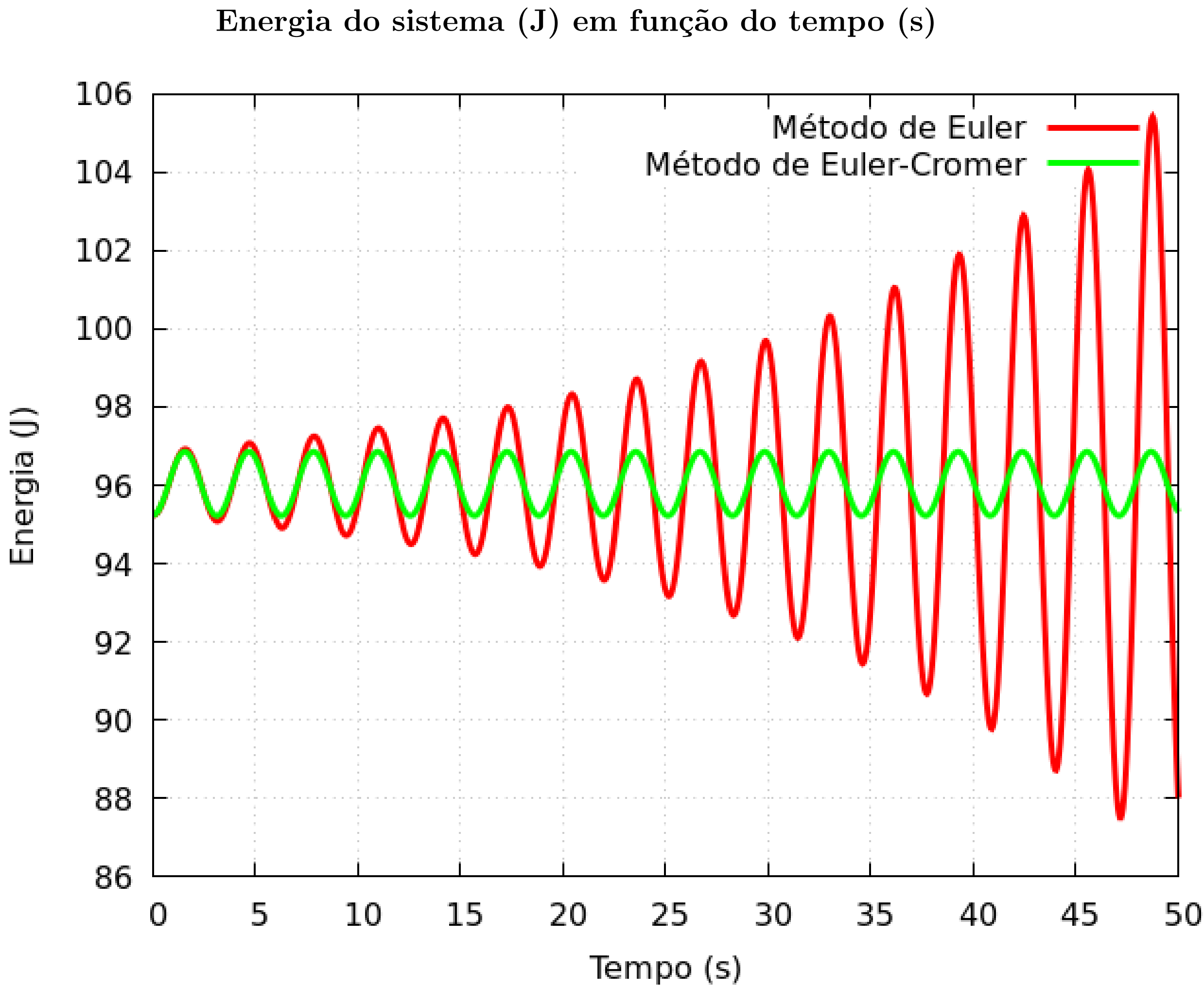

Figura 2: Valores da energia do sistema em cada ponto utilizando os métodos de Euler e Euler-Cromer

Analisando o gráfico das energias do sistema de cada método é possível, novamente, notar uma inconsistência na energia calculada pelo método Euler. Pois este, ao contrário da energia do método de Euler-Cromer que se mantém constante, é crescente ao longo do tempo.

Assim, é possível concluir que o método de Euler-Cromer é mais preciso e constante que o método de Euler. E que o método de Euler não é adequado para o estudo de sistemas oscilatórios.

Tarefa B

Tarefa B1 e B2

O objetivo da tarefa B1 é comparar o período do movimento de um pêndulo simples calculado a partir do método de Euler-Cromer com a resposta calculada numericamente da seguinte integral:

$$T = \sqrt{\frac{2l}{g}} \int_{-\theta_0}^{\theta_0} \frac{d\theta}{\sqrt{\cos\theta - \cos\theta_0}}$$

Para resolver essa integral é preciso adicionar um erro ϵ para impedir o denominador de ficar muito próximo de zero, assim depois de adicionar esse erro e manipulando a equação chegamos na seguinte expressão:

$$T = \sqrt{\frac{2l}{g}} \int_{-\theta_0}^{\theta_0+\epsilon} \frac{d\theta}{\sqrt{\cos\theta - \cos\theta_0}} + 2\sqrt{\frac{2l}{g}}\sqrt{\frac{\epsilon}{\sin\theta_0}}$$

A resolução dessa expressão sera dadá pelo regra de Boole utilizado no projeto anterior e descrito a seguir.

$$\int_{x_0}^{x_4} f(x)dx = \frac{2h}{45} \cdot (7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4) + O(h^7)$$

Além disso, o calculo do período utilizando os dados do método de Euler-Cromer será feito utilizando o método da busca direta também utilizado no projeto anterior.

Por ultimo, a tarefa B2 é pedido para comparar os resultados obtidos na tarefa B1 com o calculo analítico do período definido pela equação abaixo. Como as tarefas são muito similares, elas foram resolvidas no mesmo algoritmo.

$$T \approx 2\pi\sqrt{\frac{l}{g}} \cdot (1 + \frac{\theta_0^2}{16})$$

O código com as implementações descritas pode ser visto a seguir.

```
c      program tarefa-B
       implicit real*8 (a-h,o-z)
       Parameter(nite = 10000d0)!numero de interações
       Parameter(tempo = 500d0)!segundos
       Parameter(rpi = 2d0*dacos(-1d0))!2pi
       Parameter(rg = 9.8d0)!gravidade
       Parameter(rl = 9.8d0)!comprimento do pendulo
       Parameter(m = 1.0d0)!massa do pendulo
       Parameter(in = 10d0)
       dimension rlist_tetha(6)

       Parameter(isaid1 = 10)
       open(unit=isaid1,file='tarefa-2-saida-1-12558547.dat')

       rlist_tetha = (/(rpi/96.0d0),(rpi/48.0d0),(rpi/24.0d0),
      +(rpi/16.0d0),(rpi/8.0d0),(rpi/4.0d0)/)
       deltat = (tempo*1d0)/(nite*1d0)

       do l = 1,6
       tetha_0 = rlist_tetha(l)

       theta = tetha_0
```

```fortran
      omega = 0.0d0

      iter = 0

      rboole = 0d0
      epson = 1d-7
      h = (2d0*tetha_0)/nite
      do i = 1, nite

c     calculo de omega
      omega_p1 = omega - (rg/rl)*dsin(theta)*deltat
c     calculo de theta
      theta_p1 = theta + omega_p1*deltat

      if(theta_p1*theta<0) then
            iter = iter +1
      end if

      theta = theta_p1
      omega = omega_p1
      end do

      !regra de boole
      do j = 1, (nite/4)-1
            xj = -tetha_0 + 4d0*j*h + epson
            rboole = rboole +7d0*fx(xj,-4d0,h,tetha_0)
     ++32d0*fx(xj,-3d0,h,tetha_0) +12d0*fx(xj,-2d0,h,tetha_0)
     ++32d0*fx(xj,-1d0,h,tetha_0) +7d0*fx(xj,0d0,h,tetha_0)
      end do
      rboole =rboole*(2*h/45)

      rboole =rboole*dsqrt((2d0*rl)/rg)
      rboole =rboole +2d0*dsqrt((2d0*rl)/rg)*dsqrt(epson/dsin(tetha_0))

      analitico = rpi*dsqrt(rl/rg)*(1 + tetha_0**2/16)
      write(isaid1,*) rlist_tetha(l), (2d0*tempo)/iter, rboole,
     +analitico

      end do

      close(isaid1)
      end

      function fx(x,dn,h,tetha_0)
            implicit real*8 (a-h,o-z)
            arg = x + dn*h
            fx = 1/dsqrt(dcos(arg)-dcos(tetha_0))
      end function
```

Algoritmo 2: Código para resolução da tarefa B1 e B2

Resultados

A saída do algoritmo pode ser vista a seguir, em que cada coluna representa, respectivamente, θ_0, período do método de Euler-Cromer, período calculado pela integral e o período calculado analiticamente.

6.5449846949787352E-002	6.2893081761006293	6.2549122243565112	6.2848675053501752
0.13089969389957470	6.2893081761006293	6.2885101744030631	6.2899140998619432
0.26179938779914941	6.3291139240506329	6.3500398318991760	6.3101004779090131
0.39269908169872414	6.3291139240506329	6.4165081873014200	6.3437444413207968
0.78539816339744828	6.5359477124183005	6.6851813494192509	6.5254218437444287
1.5707963267948966	7.4074074074074074	7.7233400939005739	7.2521314534389560

Figura 3: saída do algoritmo

É possível concluir, analisando os resultados, que o período do método de Euler-Cromer e o período calculado pela integral são próximos, sendo que o período do método de Euler-Cromer é relativamente mais preciso quando comparado com o método analítico.

Além disso, quando comparado o período do método de Euler-Cromer e o período calculado analiticamente para ângulos menores que $\theta_0 < \frac{\pi}{12}$ podemos perceber que eles se mantém iguais. Por isso, pode se concluir que para ângulos pequenos o período independe do valor de θ_0 e pode ser dado pela seguinte equação.

$$T \approx 2\pi\sqrt{\frac{l}{g}}$$

Tarefa B3

A tarefa B3 é muito similar as anteriores, objetivo dela é resolução numericamente do movimento de um pêndulo amortecido com $\gamma = 0.5$. A discretização utilizada no algortimo pode ser vista abaixo:

$$\omega_{i+1} = \omega_i - \frac{g}{l}\sin{(\theta_i)}\Delta t - \gamma\omega_i\Delta t$$

$$\theta_{i+1} = \theta_i + \omega_{i+1}\Delta t$$

O algoritmo com a solução pode ser visto abaixo.

```
c      program tarefa-A
       implicit real*8 (a-h,o-z)
       Parameter(nite = 1000d0)
       Parameter(tempo = 50d0)!segundos
       Parameter(rpi = 2d0*dacos(-1d0))
       Parameter(rg = 9.8d0)
       Parameter(rl = 9.8d0)
       Parameter(m = 1.0d0)
       Parameter(gamma = 0.5d0)

```

```
      Parameter(isaid1 = 10)
      open(unit=isaid1,file='tarefa-3-saida-1-12558547.dat')

      deltat = (tempo*1d0)/(nite*1d0)

      theta = rpi/48.0d0
      omega = 0.0d0

      do i = 1, nite

c     calculo de omega
      omega_p1 = omega -(rg/rl)*dsin(theta)*deltat -gamma*omega*deltat
c     calculo de theta
      theta_p1 = theta + omega_p1*deltat

      write(isaid1,*) i*deltat , theta_p1

      theta = theta_p1
      omega = omega_p1

      end do
      close(isaid1)
      end
```

Algoritmo 3: Código para resolução da tarefa B3

Resultados

A seguir está o gráfico de θ (rad) em função do tempo gerado a partir da solução encontrada pelo algoritmo.

Oscilação de θ (rad) em função do tempo (s)

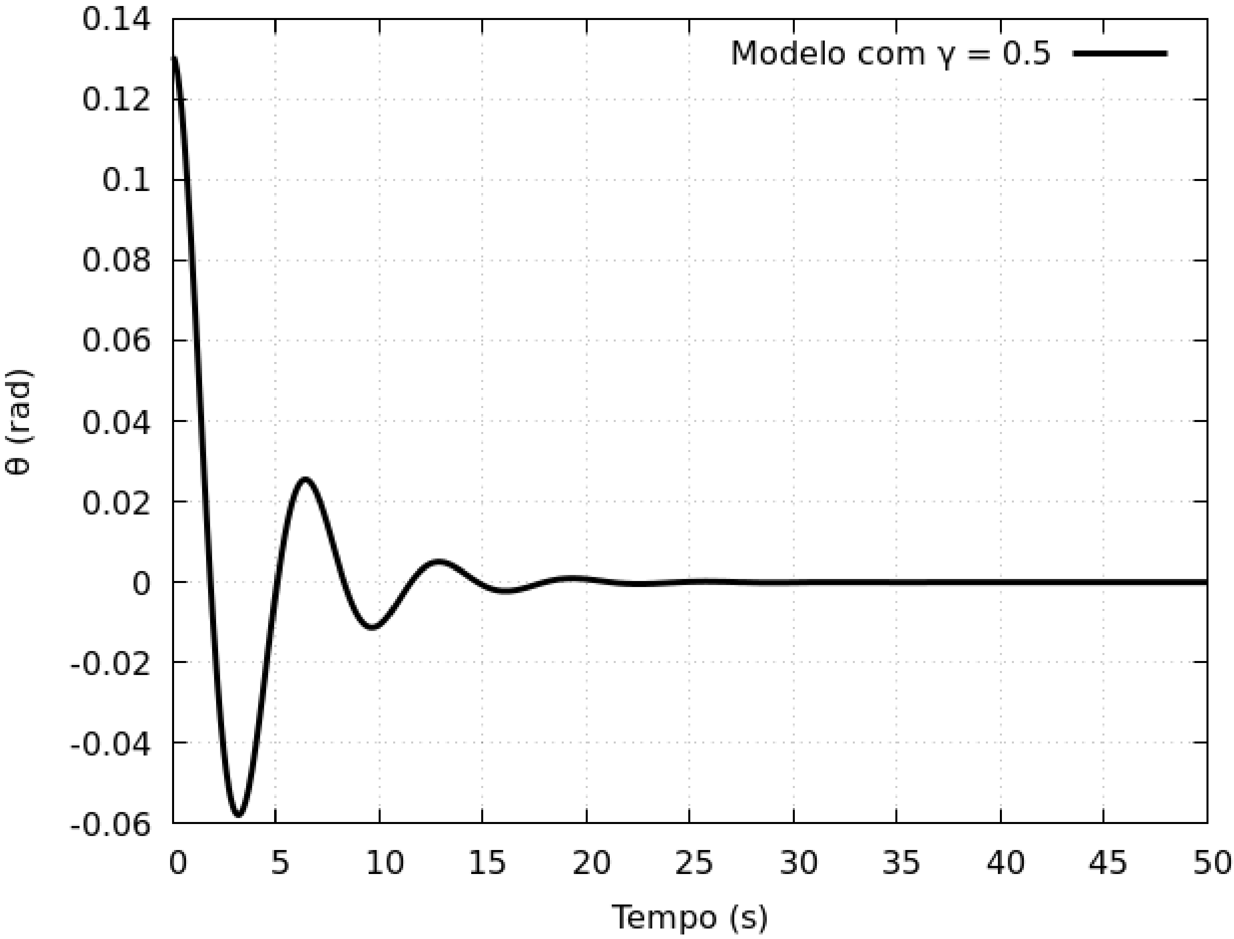

Figura 4: Valores de θ em função do tempo t em um sistema amortecido

Analisando o gráfico, é possível concluir que o amortecimento é subcrítico, como esperado.

Tarefa B4

A tarefa B4 é também muito similar as anteriores, objetivo dela é resolução numericamente do movimento de um pêndulo amortecido forçado com $\gamma = 0.5$, $\Omega = \frac{2}{3}$ e $F_0 = 0; 0.5; 1.2$. A discretização utilizada no algoritmo pode ser vista abaixo:

$$\omega_{i+1} = \omega_i - \frac{g}{l} \sin{(\theta_i)} \Delta t - \gamma \omega_i \Delta t + F_0 \sin{(\omega \theta i)} \Delta t$$

$$\theta_{i+1} = \theta_i + \omega_{i+1} \Delta t$$

O algoritmo com a solução pode ser visto abaixo.

```
c      program tarefa-A
       implicit real*8 (a-h,o-z)
       Parameter(nite = 1000d0)
       Parameter(tempo = 50d0)!segundos
```

```fortran
      Parameter(rpi = 2d0*dacos(-1d0))
      Parameter(rg = 9.8d0)
      Parameter(rl = 9.8d0)
      Parameter(m = 1.0d0)
      Parameter(gamma = 0.5d0)
      Parameter(omm = 0.67d0)
      dimension f_0(3)

      Parameter(isaid1 = 10)
      Parameter(isaid2 = 20)
      open(unit=isaid1,file='tarefa-4-saida-1-12558547.dat')
      open(unit=isaid2,file='tarefa-4-saida-2-12558547.dat')

      f_0 = (/0d0,0.5d0,1.2d0/)

      deltat = 0.03d0

      theta = rpi/48d0
      omega = 0.0d0

      theta2 = rpi/48d0
      omega2 = 0.0d0

      theta3 = rpi/48d0
      omega3 = 0.0d0

      iter = 0

      do i = 1, nite

c     calculo de omega1
      omega_p1 = omega -(rg/rl)*dsin(theta)*deltat
     +-gamma*omega*deltat+f_0(1)*dsin(omm*deltat*i)*deltat

c     calculo de theta1
      theta_p1 = theta + omega_p1*deltat

c     calculo de omega2
      omega2_p1 = omega2 -(rg/rl)*dsin(theta2)*deltat
     +-gamma*omega2*deltat+f_0(2)*dsin(omm*deltat*i)*deltat

c     calculo de theta2
      theta2_p1 = theta2 + omega2_p1*deltat

c     calculo de omega3
      omega3_p1 = omega3 -(rg/rl)*dsin(theta3)*deltat
     +-gamma*omega3*deltat+f_0(3)*dsin(omm*deltat*i)*deltat

```

```
c      calculo de theta3
       theta3_p1 = theta3 + omega3_p1*deltat

       if(theta_p1*theta<0) then
             iter = iter +1
       end if

       write(isaid1,*) i*deltat, mod(50*rpi+theta_p1,-rpi),
      +mod(50*rpi+theta2_p1,-rpi),mod(50*rpi+theta3_p1,-rpi)

       write(isaid2,*) i*deltat, omega_p1,omega2_p1,omega3_p1

       theta = theta_p1
       omega = omega_p1

       theta2 = theta2_p1
       omega2 = omega2_p1

       theta3 = theta3_p1
       omega3 = omega3_p1

       end do

       freq = (2d0*tempo)/iter

       freq =  1d0/freq

       write(*,*) freq

       close(isaid1)
       close(isaid2)
       end
```

Algoritmo 4: Código para resolução da tarefa B4

Resultados

A seguir estão os gráficos de θ (rad) em função do tempo e ω (rad/s) em função do tempo.

Oscilação de θ (rad) em função do tempo (s)

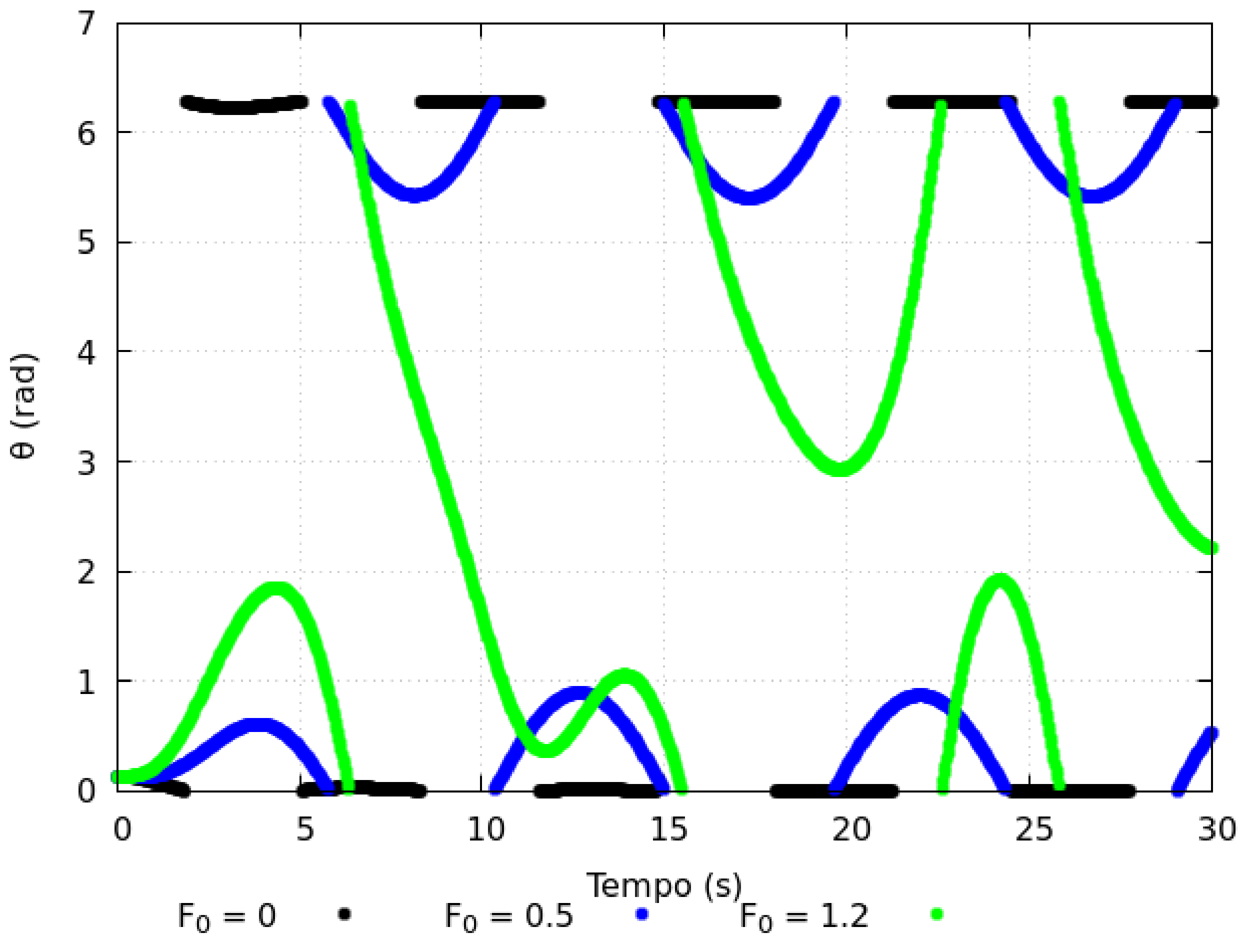

Figura 5: Valores de θ em função do tempo t em um sistema amortecido forçado

Oscilação de ω (rad/s) em função do tempo (s)

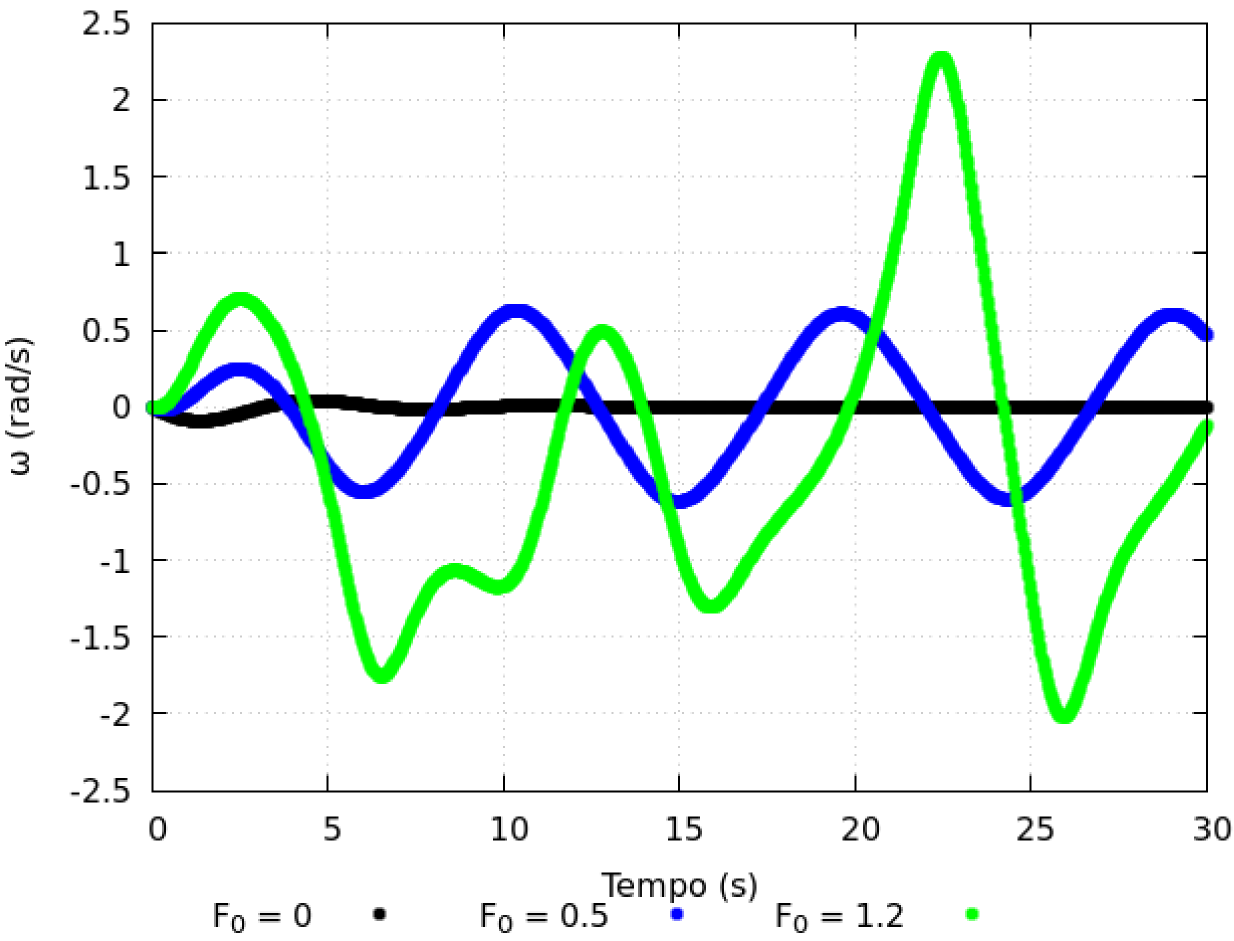

Figura 6: Valores de ω em função do tempo t em um sistema amortecido forçado

O resultado obtido para a frequência de $F_0 = 0$ foi de f = 8.9999999999999997E-002, esse é o caso do pendulo com apenas forças dissipativas, por isso sua frequência tende a 0 visto que o sistema amortecido tem poucas oscilações. Dessa forma, pode se afirmar que o resultado é coerente com o esperado.

No caso do pendulo com $F_0 = 0.5$ suas forças externas praticamente se anulam, desa forma seu comportamento é muito similar ao pendulo simples (sem forças externas), que foi visto anteriormente.

Por ultimo, o pendulo com $F_0 = 1.2$ tem um comportamento caótico, sendo não periódico e sem frequência. Pequenas variações de θ_0 nesse pendulo muda completamente seu comportamento.

Tarefa C

A tarefa C tem como objetivo analisar mais profundamente o caos presente no movimento de um pêndulo amortecido forçado. As condições iniciais do sistema são iguais à tarefa B4, com a diferença que $F_0 = 0.5; 1.2$.

Além disso, agora o sistema ira simular dois pêndulos ao mesmo tempo. A diferença entre os θ dos dois pêndulos soltos simultaneamente vai ser de tal forma que $\Delta\theta = 10^{-3}$.

Para determinar a existência ou não do regime caótico será utilizado a expressão de Liapunov. Se os θ_0 dos pêndulos se afastarem exponencialmente isso significa que o sistema é caótico, e pelo contrario, se os θ_0 dos pêndulos se aproximarem exponencialmente isso significa que o sistema é não caótico.

Sabendo disso, para saber se os θ_0 dos pêndulos se afastam ou se aproximam é utilizado a seguinte equação:

$$ln(|\Delta\theta|) = \lambda t$$

se $\lambda > 0$ o sistema é caótico, se $\lambda < 0$ o sistema não é caótico.

O algoritmo com a solução pode ser visto abaixo.

```
c       program tarefa-C
        implicit real*8 (a-h,o-z)
        Parameter(nite = 1000d0)
        Parameter(tempo = 30d0)!segundos
        Parameter(rpi = 2d0*dacos(-1d0))
        Parameter(rg = 9.8d0)
        Parameter(rl = 9.8d0)
        Parameter(m = 1.0d0)
        Parameter(gamma = 0.5d0)
        Parameter(omm = 0.67d0)
        dimension f_0(2)

        Parameter(isaid1 = 10)
        Parameter(isaid2 = 20)
        open(unit=isaid1,file='tarefa-5-saida-1-12558547.dat')
        open(unit=isaid2,file='tarefa-5-saida-2-12558547.dat')

        f_0 = (/0.5d0,1.2d0/)

        deltat = (tempo*1d0)/(nite*1d0)

        do j = 1, 2

        theta1 = rpi/48d0
        omega1 = 0.0d0

        theta2 = theta1 + 0.001d0
        omega2 = omega1

        do i = 1, nite

c       calculo de omega1
        omega1_p1 = omega1 -(rg/rl)*dsin(theta1)*deltat
       +-gamma*omega1*deltat+f_0(j)*dsin(omm*deltat*i)*deltat

c       calculo de theta1
        theta1_p1 = theta1 + omega1_p1*deltat
```

```
c     calculo de omega2
      omega2_p1 = omega2 -(rg/rl)*dsin(theta2)*deltat
     +-gamma*omega2*deltat+f_0(j)*dsin(omm*deltat*i)*deltat

c     calculo de theta2
      theta2_p1 = theta2 + omega2_p1*deltat

      write(10*j,*) i*deltat,
 ↪  dlog(abs(theta2_p1-theta1_p1))!mod(50*rpi+theta1_p1-theta2_p1,-rpi)

      theta1 = theta1_p1
      omega1 = omega1_p1

      theta2 = theta2_p1
      omega2 = omega2_p1

      end do
      end do
      close(isaid1)
      close(isaid2)
      end
```

Algoritmo 5: Código para resolução da tarefa C

Resultados

A seguir estão os gráficos de $ln(\Delta\theta)$ (rad) em função do tempo para $F_0 = 0.5$ e $F_0 = 1.2$.

$ln(\Delta\theta)$ **(rad) em função do tempo (s) com** $F_0 = 0.5$

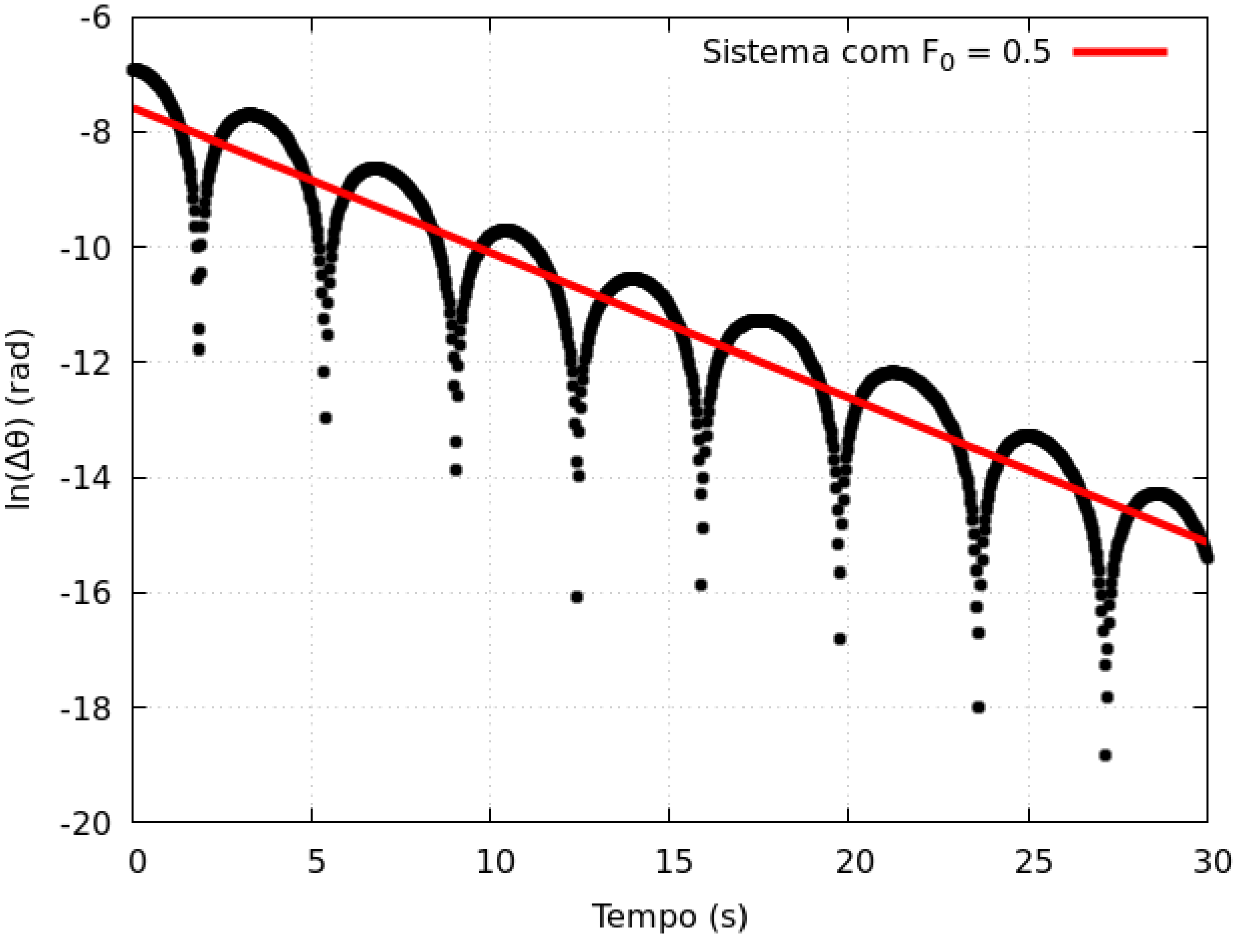

Figura 7: Valores de $\Delta\theta$ em função do tempo t em um sistema amortecido forçado

$ln(\Delta\theta)$ **(rad) em função do tempo (s) com** $F_0 = 1.2$

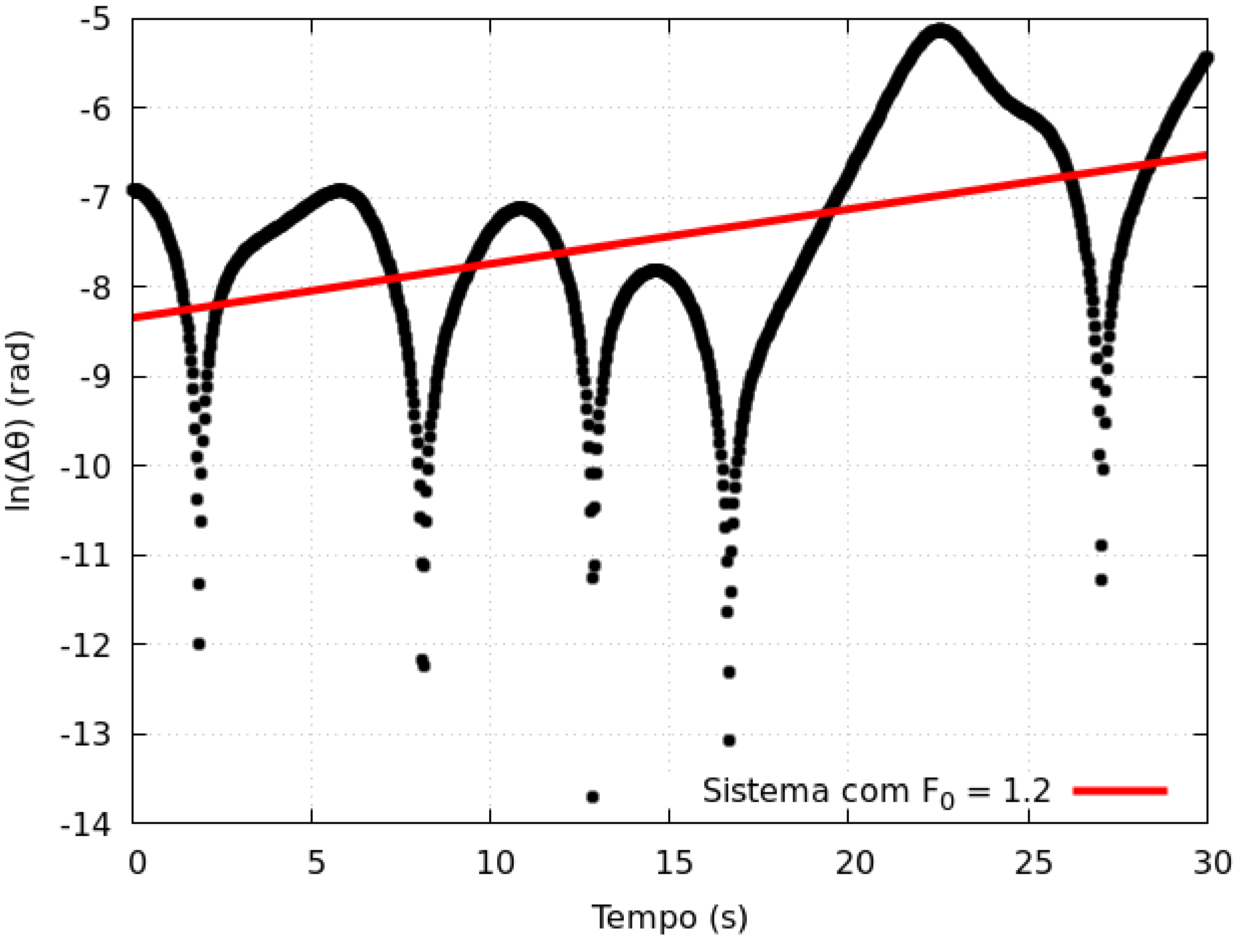

Figura 8: Valores de $\Delta\theta$ em função do tempo t em um sistema amortecido forçado

Fazendo o ajuste linear dos dados obtidos do algoritmo, foram obtidos os seguintes expoente de Liapunov. Para $F_0 = 0.5$, $\lambda = -0.252 \pm 0.003$; e para $F_0 = 1.2$, $\lambda = 0.061 \pm 0.004$. Dessa forma, é possível concluir que o sistema com $F_0 = 1.2$ é caótico, pois seu expoente de Liapunov é maior que 0. Esse resultado confirma os dados encontrados na tarefa anterior.

Tarefa D

A tarefa D tem os mesmos parâmetros iniciais da tarefa C e utiliza do mesmo logica do anterior. A seguir esta o algoritmo para a resolução da tarefa D.

```
c      program tarefa-C
       implicit real*8 (a-h,o-z)
       Parameter(nite = 10000d0)
       Parameter(tempo = 300d0)!segundos
       Parameter(rpi = 2d0*dacos(-1d0))
       Parameter(rg = 9.8d0)
       Parameter(rl = 9.8d0)
```

```
      Parameter(m = 1.0d0)
      Parameter(gamma = 0.5d0)
      Parameter(omm = 0.67d0)
      dimension f_0(2)

      Parameter(isaid1 = 10)
      Parameter(isaid2 = 20)
      open(unit=isaid1,file='tarefa-6-saida-1-12558547.dat')
      open(unit=isaid2,file='tarefa-6-saida-2-12558547.dat')

      f_0 = (/0.5d0,1.2d0/)

      deltat = (tempo*1d0)/(nite*1d0)

      do j = 1, 2

      theta1 = rpi/48d0
      omega1 = 0.0d0

      do i = 1, nite

c     calculo de omega1
      omega1_p1 = omega1 -(rg/rl)*dsin(theta1)*deltat
     +-gamma*omega1*deltat+f_0(j)*dsin(omm*deltat*i)*deltat

c     calculo de theta1
      theta1_p1 = theta1 + omega1_p1*deltat

      write(10*j,*) mod(50*rpi+theta1_p1,-rpi), omega1_p1

      theta1 = theta1_p1
      omega1 = omega1_p1

      end do
      end do
      close(isaid1)
      close(isaid2)
      end
```

Algoritmo 6: Código para resolução da tarefa D

Resultados

Logo em seguida, estão os gráficos de ω (rad/s) em função de θ (rad) com $F_0 = 0.5; 1.2$ e $\theta_0 = 0.1309; 0.1609; 0.1009$.

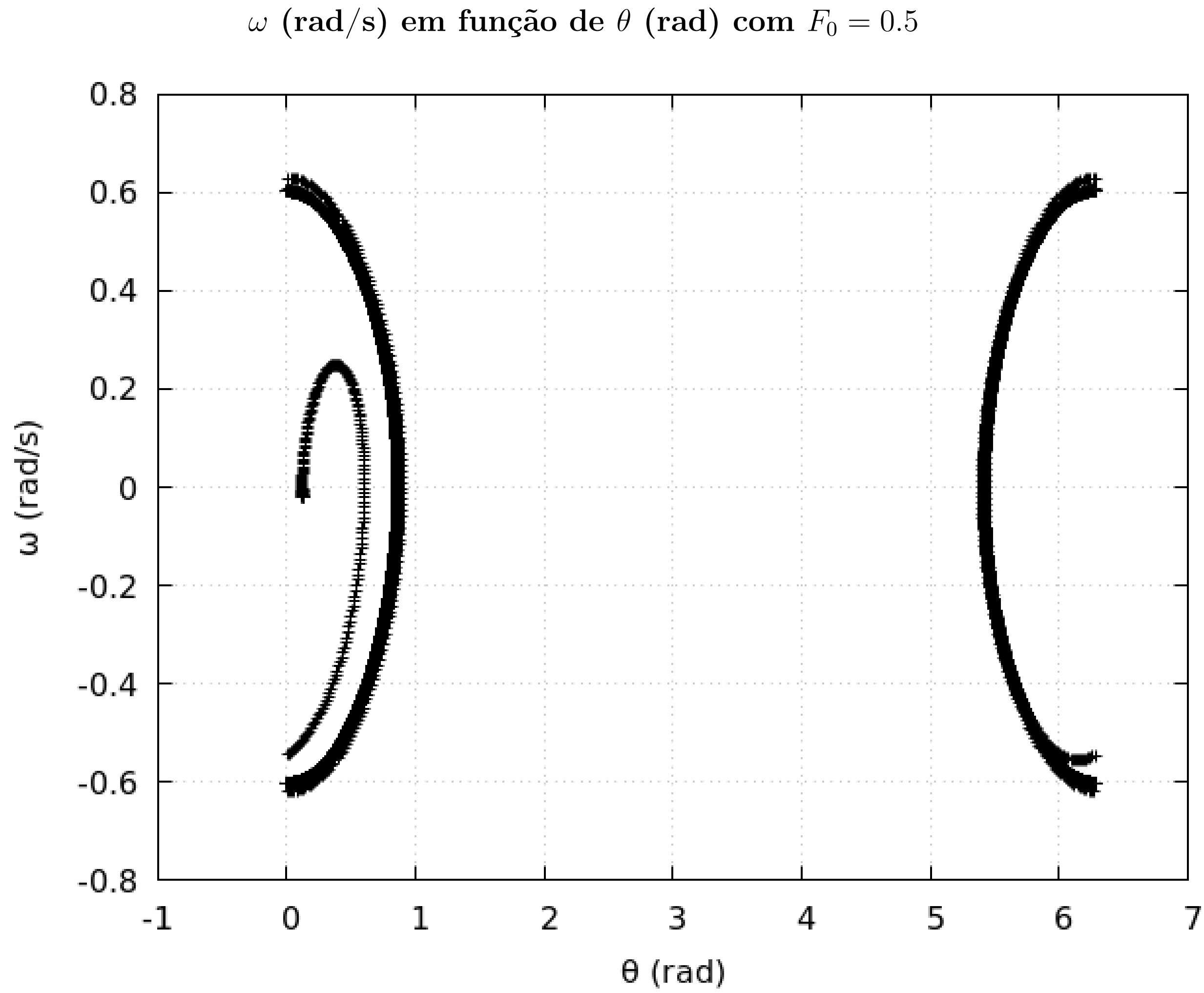

Figura 9: Valores de ω função de θ (rad) em um sistema amortecido forçado com $\theta_0 = 0.1309$ (rad)

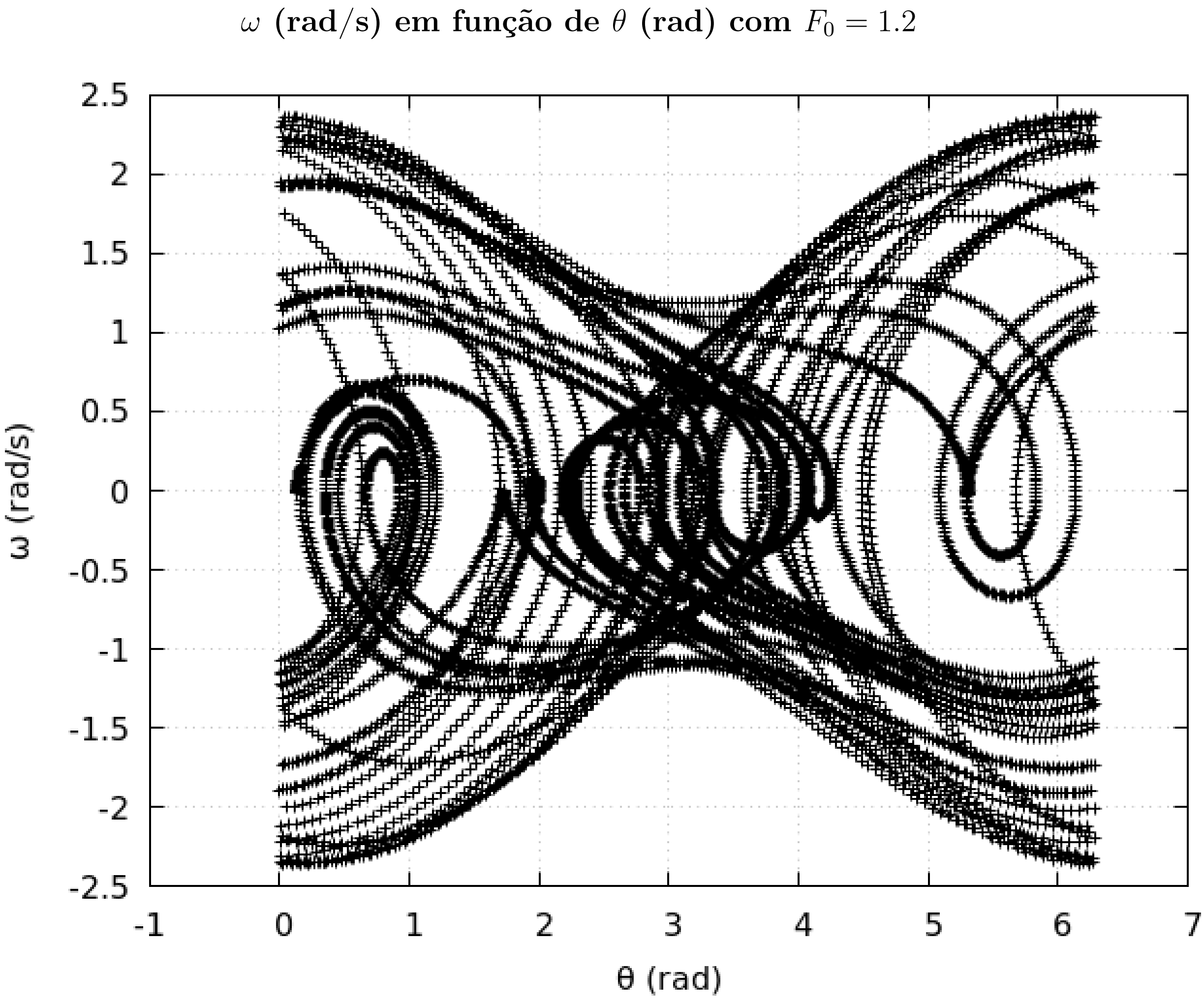

Figura 10: Valores de ω função de θ (rad) em um sistema amortecido forçado com $\theta_0 = 0.1309$ (rad)

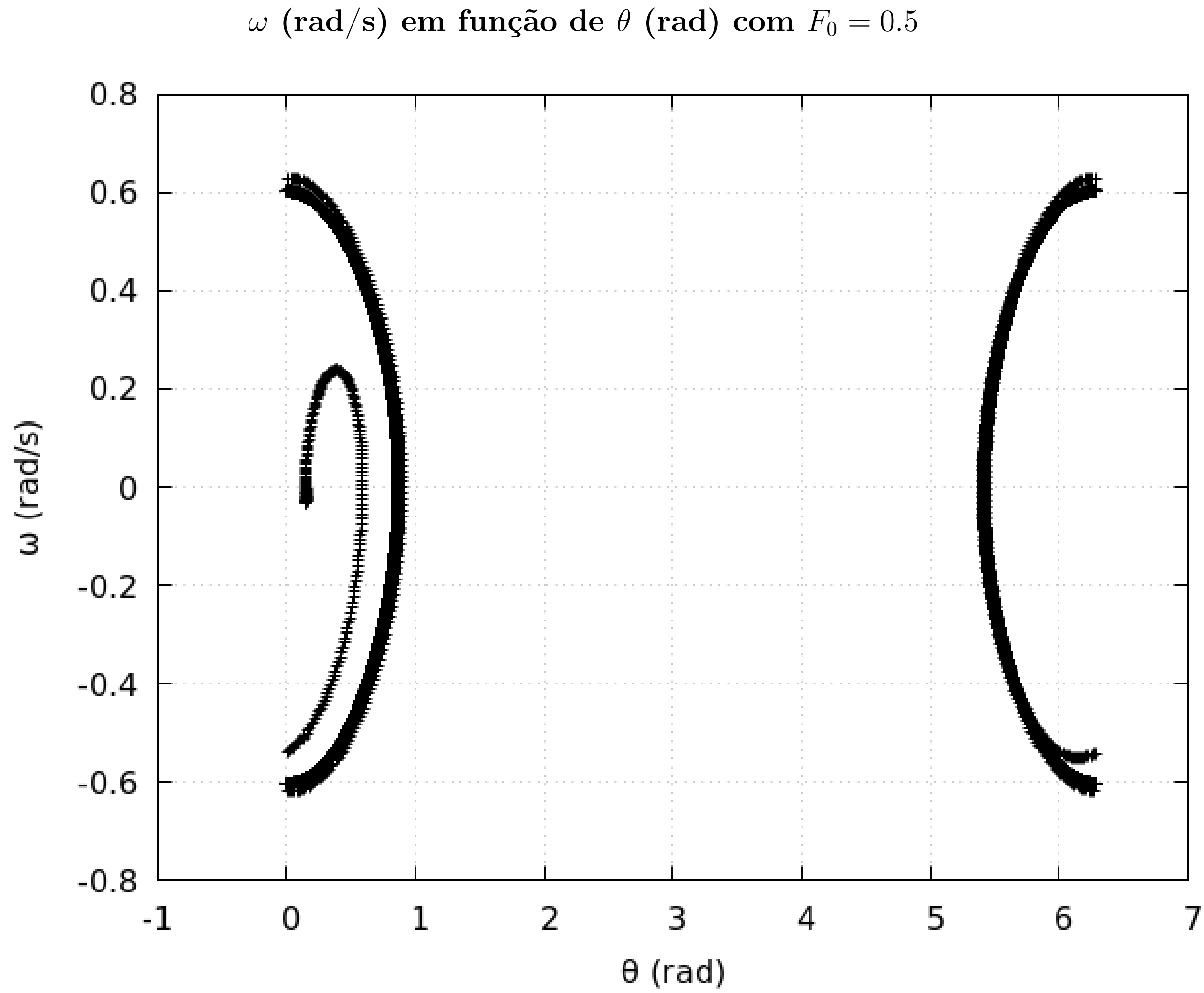

Figura 11: Valores de ω função de θ (rad) em um sistema amortecido forçado com $\theta_0 = 0.1609$ (rad)

ω **(rad/s) em função de** θ **(rad) com** $F_0 = 1.2$

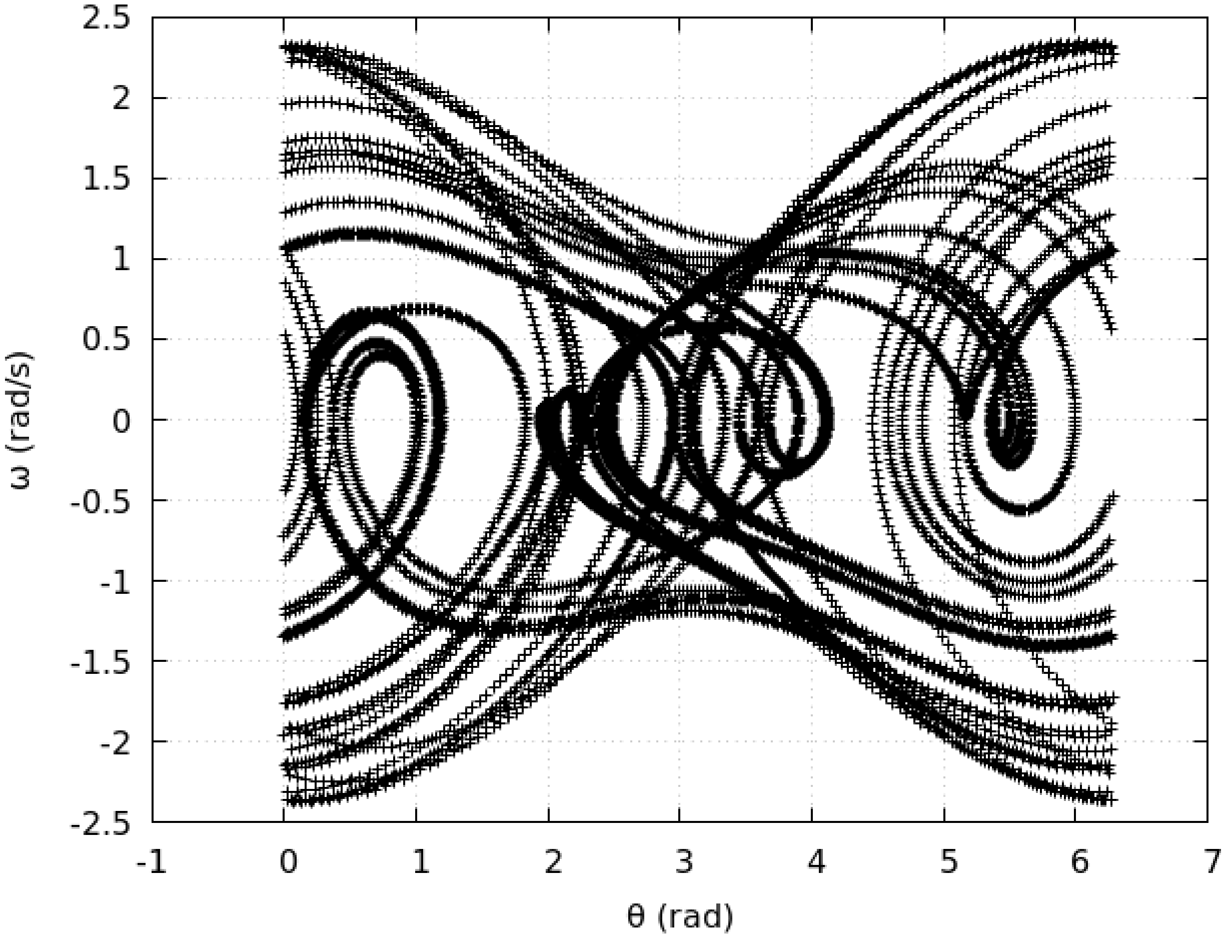

Figura 12: Valores de ω função de θ (rad) em um sistema amortecido forçado com $\theta_0 = 0.1609$ (rad)

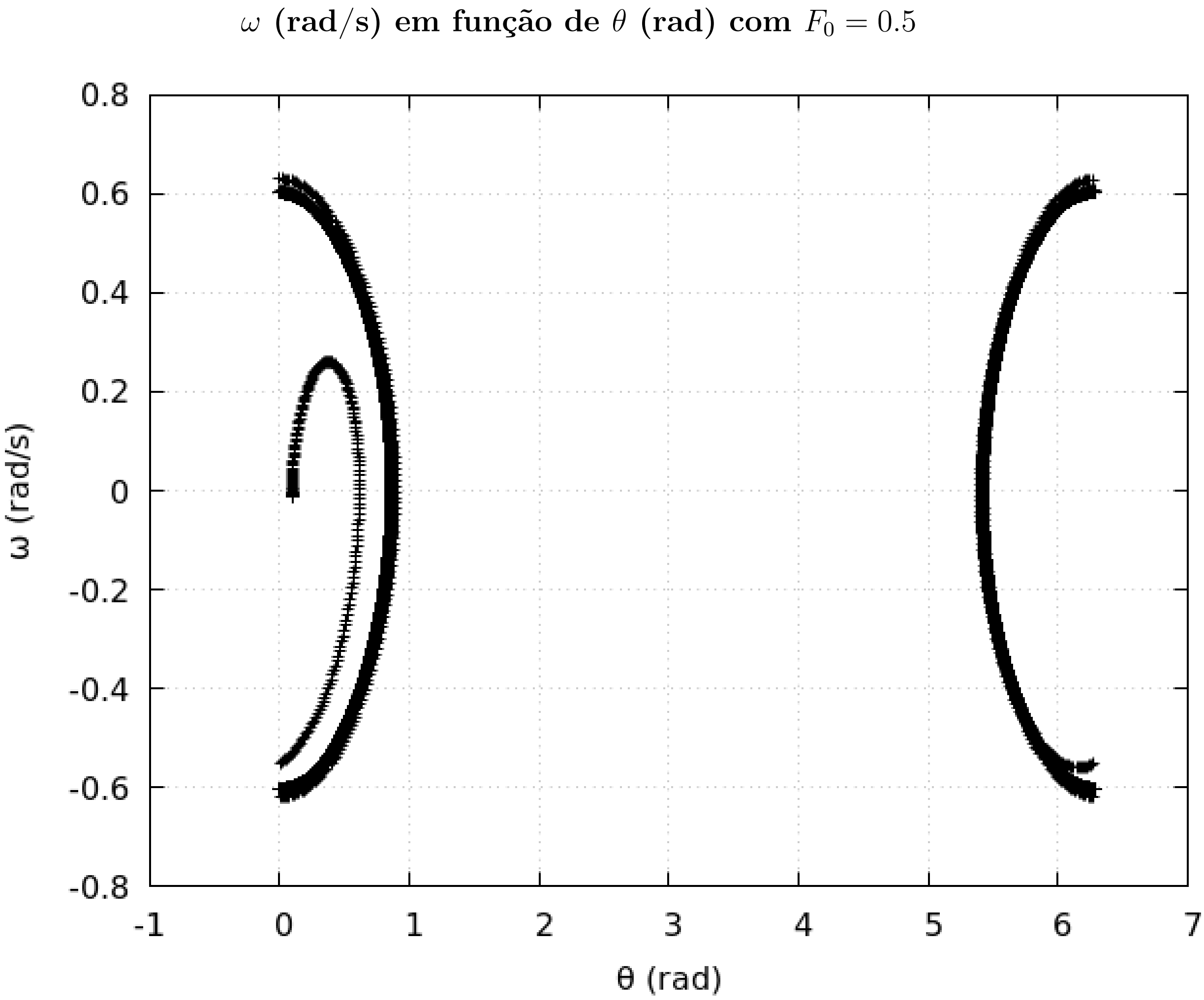

Figura 13: Valores de ω função de θ (rad) em um sistema amortecido forçado com $\theta_0 = 0.1009$ (rad)

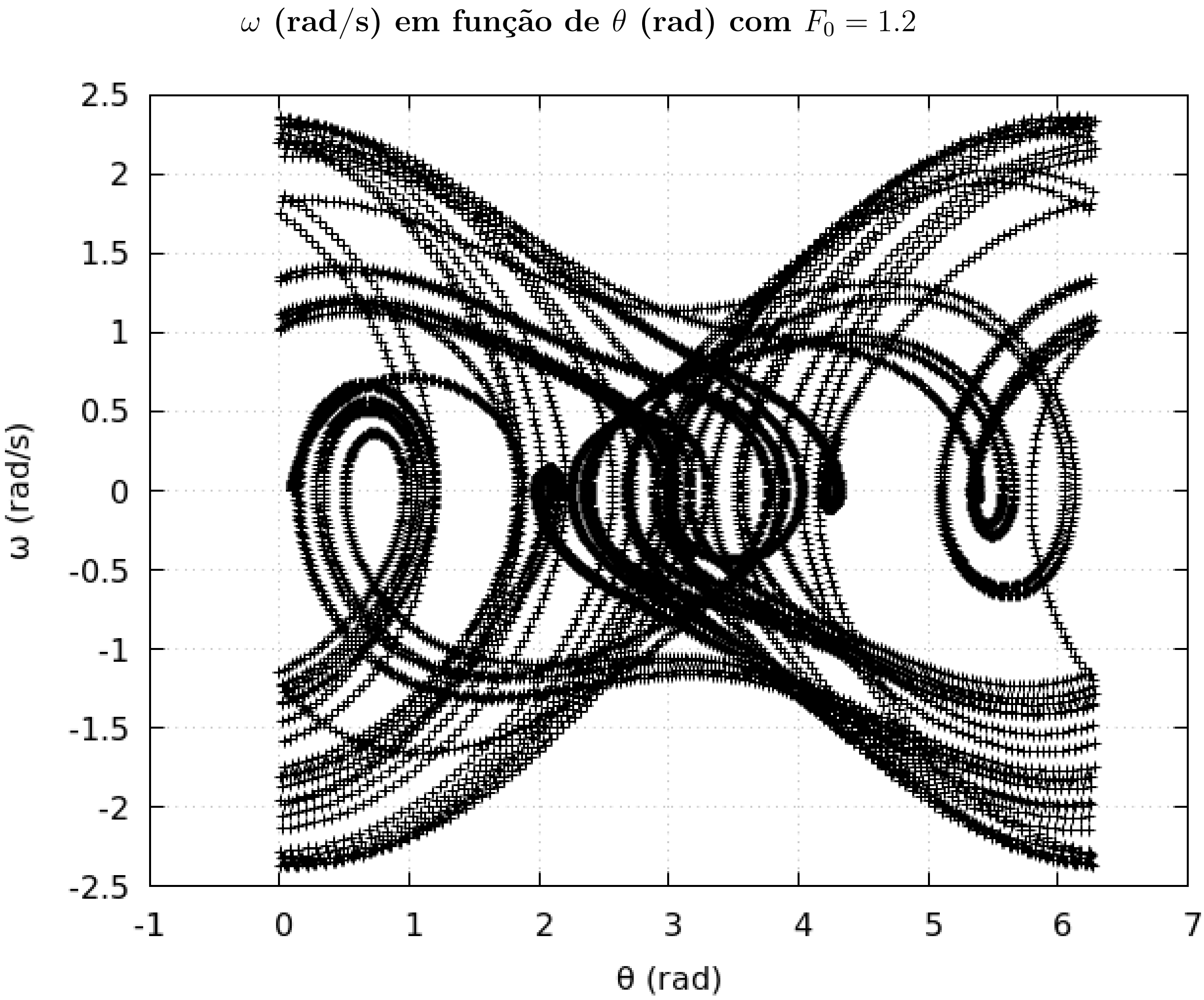

Figura 14: Valores de ω função de θ (rad) em um sistema amortecido forçado com $\theta_0 = 0.1009$ (rad)

Analisando os gráficos, é possível perceber que para os gráficos com $F_0 = 0.5$ não se consegue perceber a diferença entre eles. Isso não ocorre para os gráficos com $F_0 = 1.2$ que apesar de pequenas são perceptíveis as diferenças, a pequena diferença entre os ângulos iniciais causa uma mudança perceptível no padrão presente nos gráficos.

Tarefa E

O objetivo da tarefa E é para padronizar a representação dos movimentos vistos anteriormente. Com isso em mente, vai ser utilizado a secção de Poincaré no gráfico de $\omega(\theta)$. Dessa forma, apenas os pontos em que a seguinte condição for verdadeira serão considerados.

$$|t - \frac{n\pi}{\Omega}| < \frac{\Delta t}{2}$$

O código utilizado esta abaixo.

```
c       program tarefa-C
        implicit real*8 (a-h,o-z)
```

```
      Parameter(nite = 500000d0)
      Parameter(tempo = 150000d0)!segundos
      Parameter(rpi = 2d0*dacos(-1d0))
      Parameter(rg = 9.8d0)
      Parameter(rl = 9.8d0)
      Parameter(m = 1.0d0)
      Parameter(gamma = 0.5d0)
      Parameter(omm = 0.67d0)
      dimension f_0(2)

      Parameter(isaid1 = 10)
      Parameter(isaid2 = 20)
      open(unit=isaid1,file='tarefa-7-saida-1-12558547.dat')
      open(unit=isaid2,file='tarefa-7-saida-2-12558547.dat')

      f_0 = (/0.5d0,1.2d0/)

      deltat = (tempo*1d0)/(nite*1d0)

      do j = 1, 2

      theta1 = rpi/48d0 - 0.03d0
      omega1 = 0.0d0

      do i = 1, nite

c     calculo de omega1
      omega1_p1 = omega1 -(rg/rl)*dsin(theta1)*deltat
     +-gamma*omega1*deltat+f_0(j)*dsin(omm*deltat*i)*deltat

c     calculo de theta1
      theta1_p1 = theta1 + omega1_p1*deltat

      if (mod(omm*deltat*i, (rpi/2d0)) < deltat/2d0) then
            write(10*j, *)  mod(50*rpi+theta1_p1,-rpi), omega1_p1
      end if

      theta1 = theta1_p1
      omega1 = omega1_p1

      end do
      end do
      close(isaid1)
      close(isaid2)
      end
```

Algoritmo 7: Código para resolução da tarefa D

Resultados

Logo em seguida, estão os gráficos de ω (rad/s) em função de θ (rad) com $F_0 = 0.5; 1.2$ e $\theta_0 = 0.1309; 0.1609; 0.1009$ na secção de Poincaré $\Omega t = n\pi$.

ω (rad/s) em função de θ (rad) com $F_0 = 0.5$

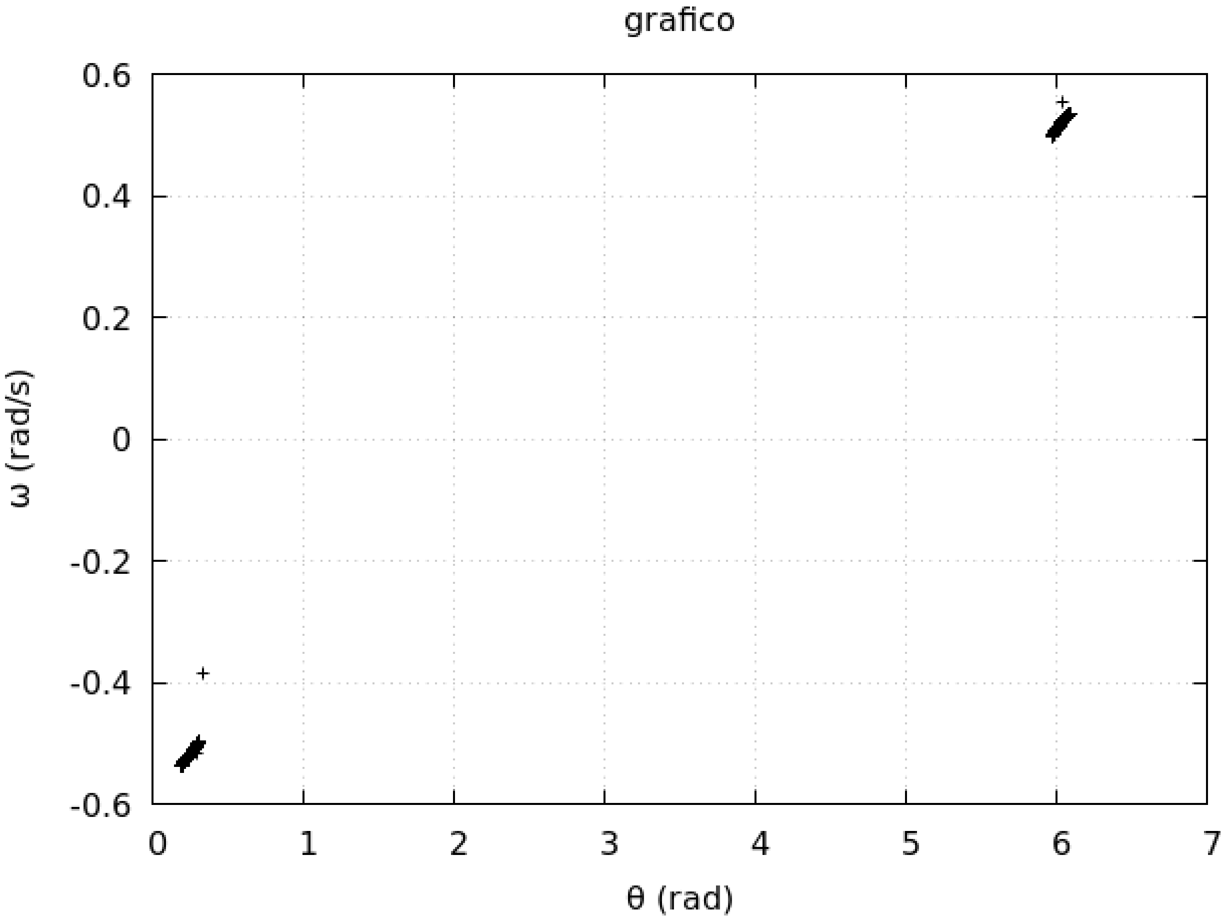

Figura 15: Valores de ω função de θ (rad) em um sistema amortecido forçado com $\theta_0 = 0.1309$ (rad)

ω **(rad/s) em função de** θ **(rad) com** $F_0 = 1.2$

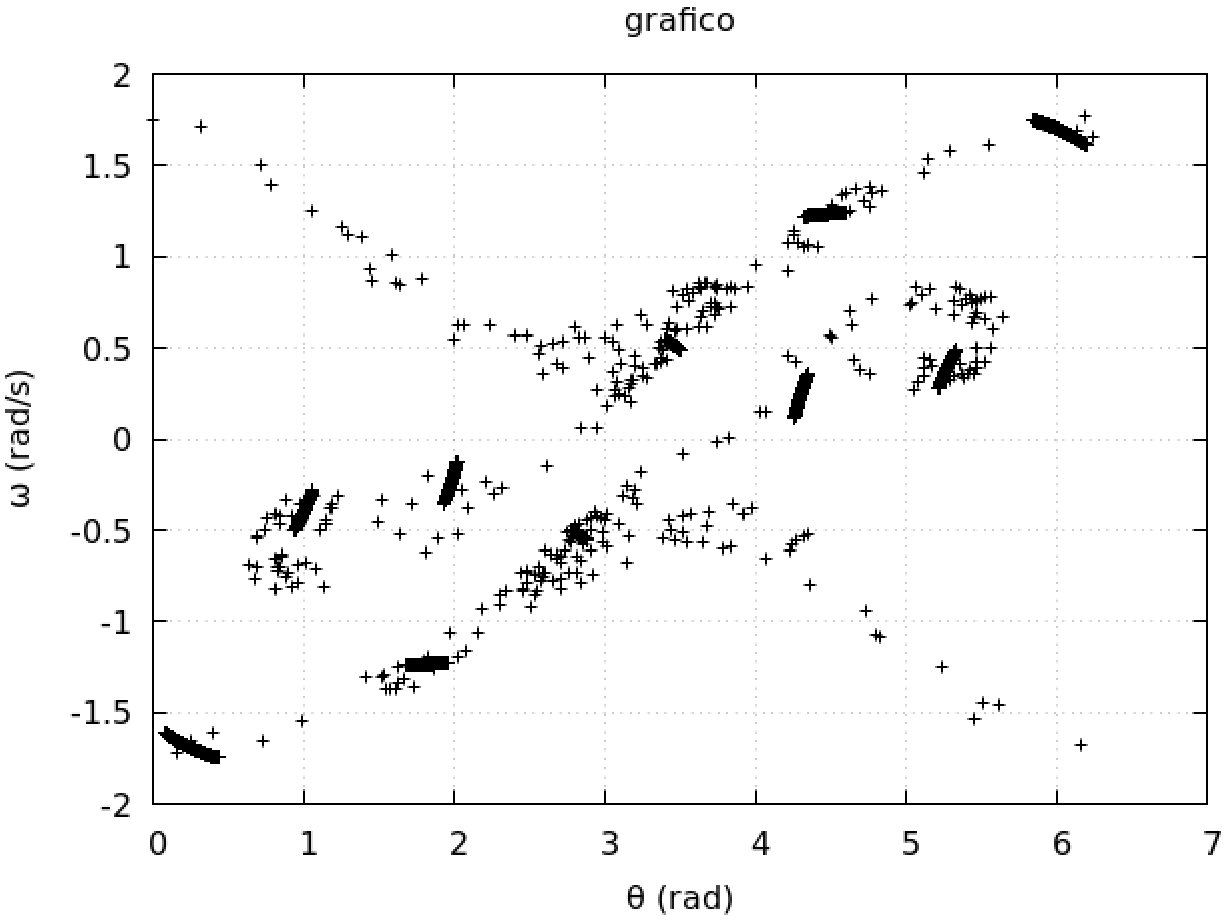

Figura 16: Valores de ω função de θ (rad) em um sistema amortecido forçado com $\theta_0 = 0.1309$ (rad)

ω **(rad/s) em função de** θ **(rad) com** $F_0 = 0.5$

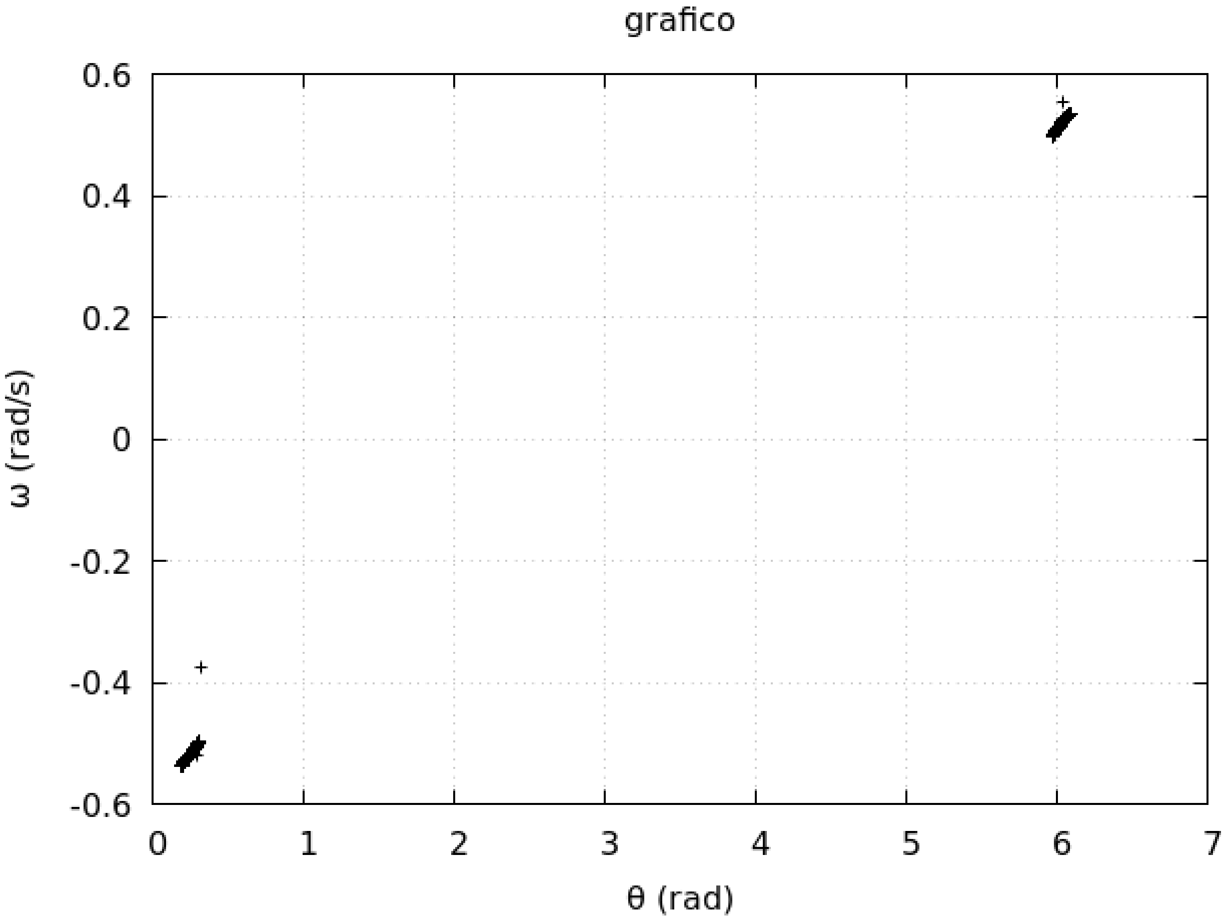

Figura 17: Valores de ω função de θ (rad) em um sistema amortecido forçado com $\theta_0 = 0.1609$ (rad)

ω (rad/s) em função de θ (rad) com $F_0 = 1.2$

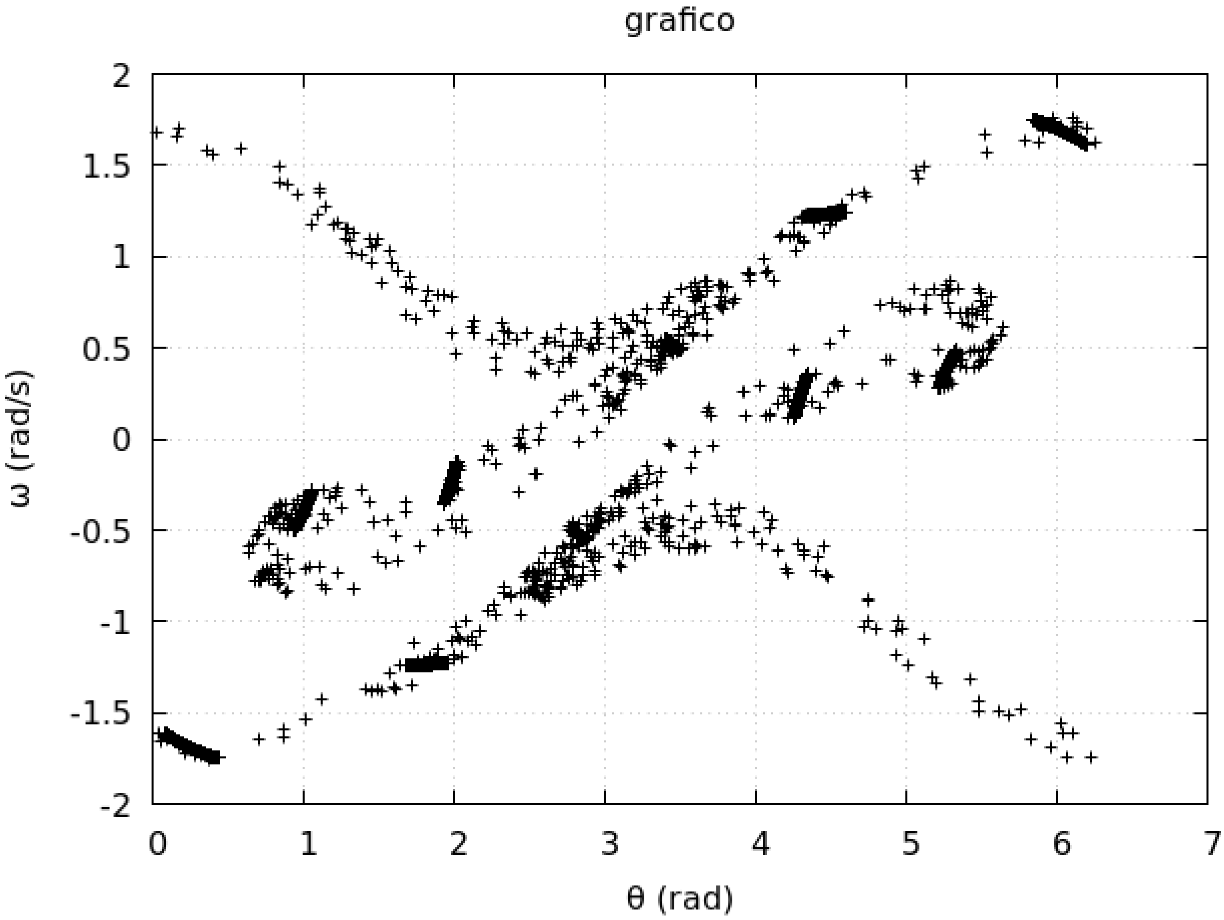

Figura 18: Valores de ω função de θ (rad) em um sistema amortecido forçado com $\theta_0 = 0.1609$ (rad)

ω (rad/s) em função de θ (rad) com $F_0 = 0.5$

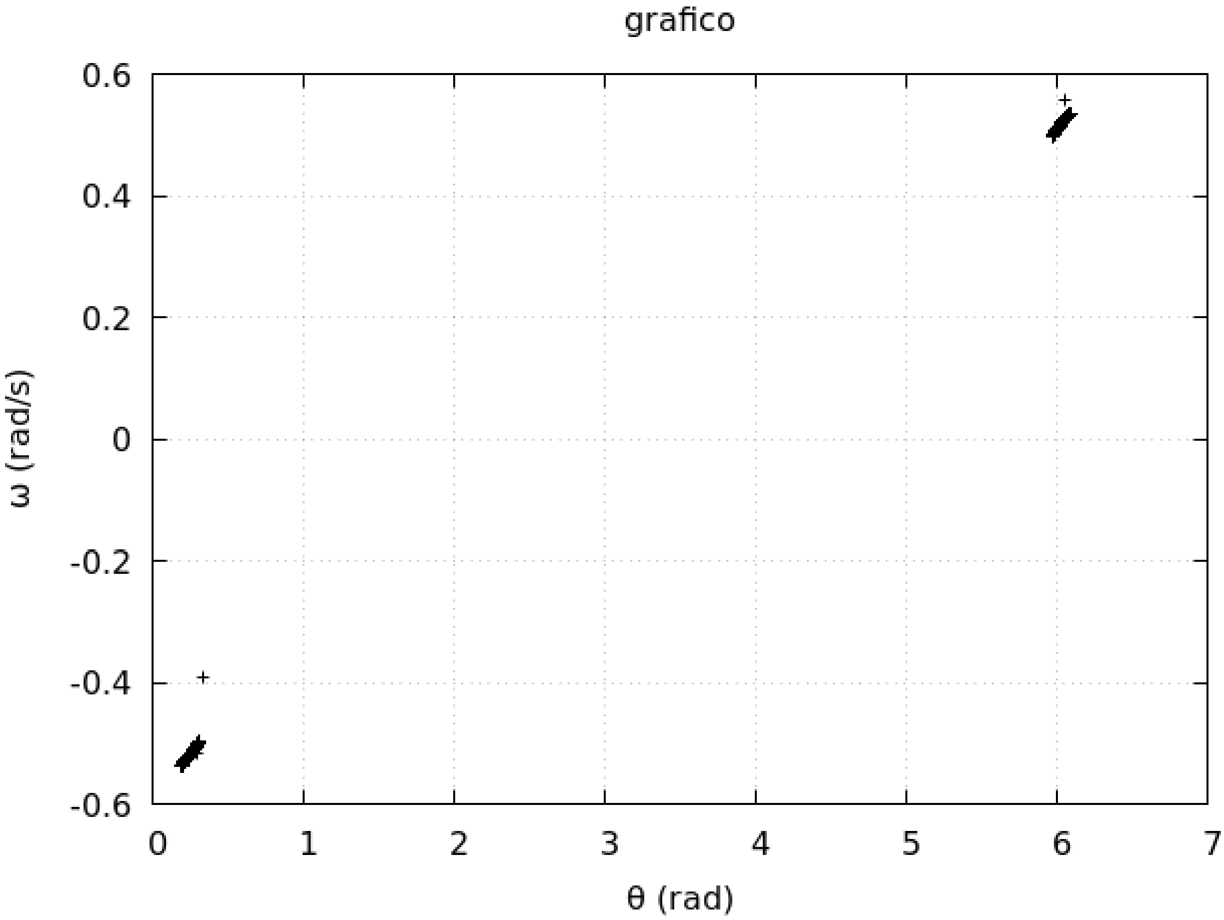

Figura 19: Valores de ω função de θ (rad) em um sistema amortecido forçado com $\theta_0 = 0.1009$ (rad)

ω (rad/s) em função de θ (rad) com $F_0 = 1.2$

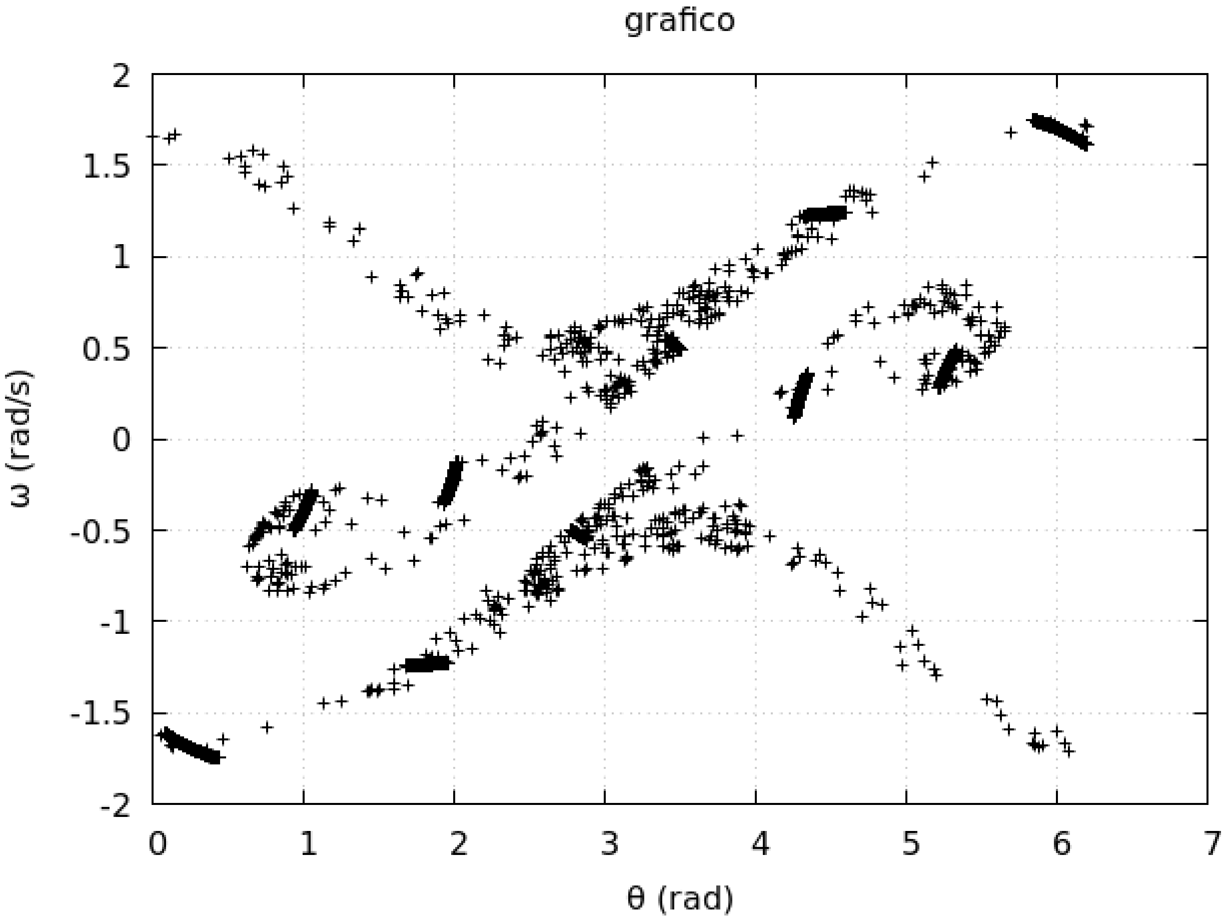

Figura 20: Valores de ω função de θ (rad) em um sistema amortecido forçado com $\theta_0 = 0.1009$ (rad)

Dessa forma, pode se concluir que as figuras não se alteram caso as condições iniciais mudem demonstrando assim a universalidade do caos, como exemplificado pelos gráficos acima. Além disso, caso determinístico pode ser observar que o resultado é bem próximo de um ponto.