Projeto 4 - Movimento Oscilatório

Instituto de Física de São Carlos
Universidade de São Paulo

Vinícius Bastos Marcos (12556715)

Introdução à Física Computacional
Prof. Francisco Castilho Alcaraz

Novembro, 2022

Tarefa A

Essa tarefa trata-se da descrição de um dos movimentos oscilatórios mais simples: o pêndulo na aproximação para ângulos pequenos. Para isso, nos é apresentado dois métodos: Euler e Euler-Cromer. Veremos que o primeiro possui deficiências e que o segundo é uma pequena correção que faz toda diferença.

A equação diferencial de segunda ordem que descreve o pêndulo simples, na aproximação $sen\theta \approx \theta$, é:

$$\frac{d^2\theta}{dt^2} = -\frac{g}{l}\theta$$

O método de Euler para o problema é:

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

$$\theta_{i+1} = \theta_i + \omega_i \Delta t \Rightarrow \theta_{i+1} = \theta_i + \Delta t \cdot (\omega_{i+1} + \frac{g}{l}\theta_i \Delta t) \tag{2}$$

Perceba que manipulei as equações da esquerda, as "originais", para que na implementação do código não seja necessária a criação de variáveis em demasia, consigo calcular θ_{i+1} a partir do ω_{i+1}.

O método de Euler-Cromer por sua vez, modifica apenas a equação (2) do método de Euler. Assim,

$$\omega_{i+1} = \omega_i - \frac{g}{l}\theta_i \Delta t \tag{3}$$

$$\theta_{i+1} = \theta_i + \omega_{i+1} \Delta t \tag{4}$$

Veja que neste caso a manipulação realizada no método de Euler não se faz necessária.

Com isso, iremos calcular a velocidade angular, ω, e a posição angular, θ. Com isso, será possível plotar um gráfico de θ versus *tempo*, que deve ser uma oscilação senoide com amplitude constante.

Calcularemos também a Energia total da massa em cada instante de tempo, que pelo princípio da conservação de energia deve ser a mesma, constante, da seguinte maneira:

$$E = mgl(1 - cos\theta) + \frac{1}{2}m\omega^2 l^2, \tag{5}$$

plotaremos esse gráfico da energia versus tempo.

O código utilizado para implementar o que foi descrito está disposto logo abaixo.

```
c      tarefa a
       implicit real*8(a-h, o-z)
       parameter(tempo = 100.d0) !tempo total do problema
       parameter(i = 1000) !número de passos
       parameter(g = 9.8d0) !aceleração gravitacional
       parameter(comp = 9.8d0) !comprimento do pêndulo
       parameter(pi = dacos(-1.d0))
       parameter(rm = 1.d0) !massa

c      arquivos de saída para graficar
       iout10 = 10 !unidade arquivo de saída
```

```
      iout11 = 11 !unidade arquivo de saída
      open(unit=iout10, FILE='saida-a-thetaeuler.dat')
      open(unit=iout11, FILE='saida-a-energiaeuler.dat')
      iout20 = 20 !unidade arquivo de saída
      iout21 = 21 !unidade arquivo de saída
      open(unit=iout20, FILE='saida-a-thetaeulercromer.dat')
      open(unit=iout21, FILE='saida-a-energiaeulercromer.dat')

      omega0 = 0.d0 !iniciando a velocidade angular
      deltaT = tempo/i !o valor de cada delta t
      theta0 = (15.d0/180.d0)*pi !15 graus em radianos

c     Método de Euler
      omega = omega0
      theta = theta0
      E = 0.d0
      do j = 1, i
      !cálculo da energia total
      E = rm*g*comp*(1.d0 - dcos(theta))+(0.5d0)*rm*(comp*omega)**2
      !cálculo do omega
      omega = omega - (g/comp)*theta*deltaT
      !cálculo do theta
      theta = theta + deltaT*(omega + (g/comp)*theta*deltaT)
      write(iout10,*) j*deltaT, mod(theta, 2*pi)
      write(iout11,*) j*deltaT, E
      end do

c     Método de Euler-Cromer
      omega = omega0
      theta = theta0
      E = 0.d0
      do j = 1, i
      !cálculo da energia total
      E = rm*g*comp*(1.d0 - dcos(theta))+(0.5d0)*rm*(comp*omega)**2
      !cálculo do omega
      omega = omega - (g/comp)*theta*deltaT
      !cálculo do theta
      theta = theta + deltaT*omega
      write(iout20,*) j*deltaT, mod(theta, 2*pi)
      write(iout21,*) j*deltaT, E
      end do

      close(iout10)
      close(iout11)
      close(iout20)
      close(iout21)

      end
```

Algoritmo 1: código para resolução da tarefa A

Rodando o código e plotando os gráficos chegamos aos seguintes resultados, que nos leva as seguintes conclusões.

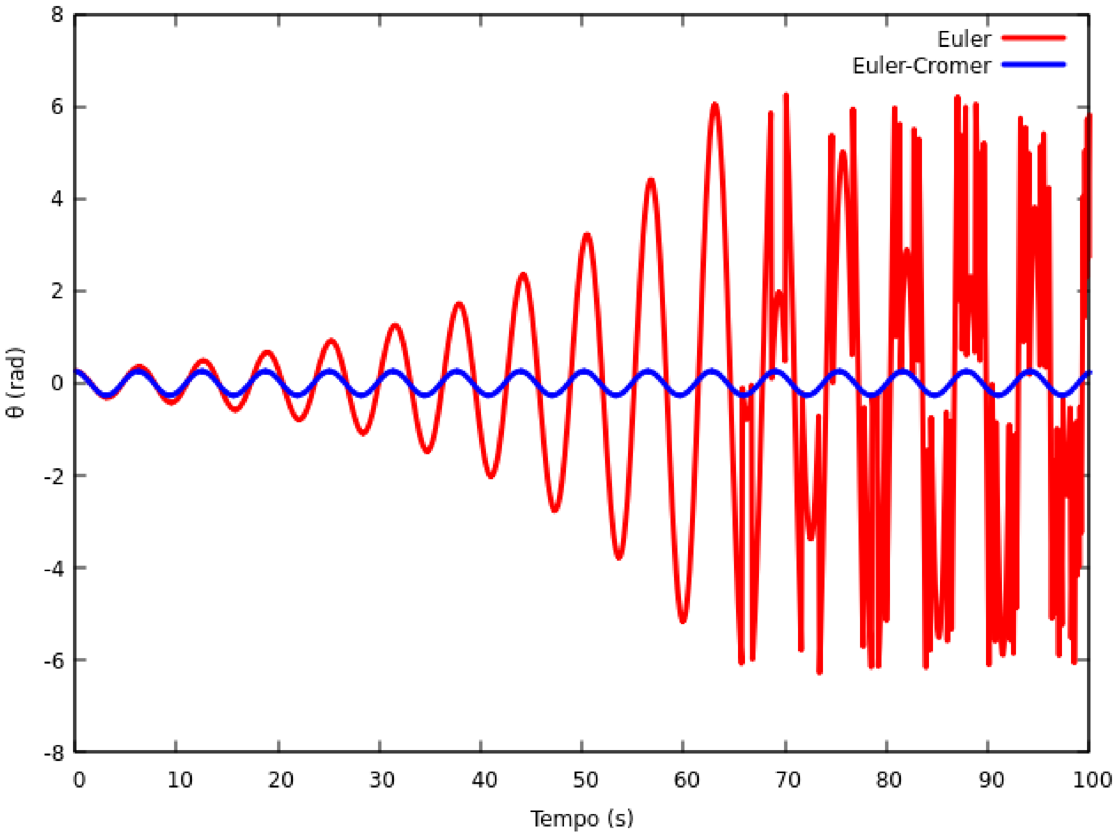

Figura 1: gráfico de $\theta(t)$ para os métodos de Euler e Euler-Cromer

Era esperado que a função $\theta(t)$ fosse uma senoide, limitada inferior e superiormente. Vemos claramente que o método de Euler não nos leva na descrição correta do movimento, visto que a amplitude de movimento aumenta de maneira que não faz nem sentido.

O método de Euler-Cromer, por sua vez, nos mostra exatamente o que esperávamos, sendo portanto o melhor método.

O gráfico de de $E(t)$ abaixo nos confirma o que foi discutido acima: o método de Euler nos leva a absurdos, como a não conservação da energia total do sistema. O método de Euler-Cromer, mais uma vez nos fornece o que esperávamos: um gráfico constante, que pela escala do gráfico para estar no zero, mas ao consultar o arquivo com os dados, vemos que está em torno de $3J$.

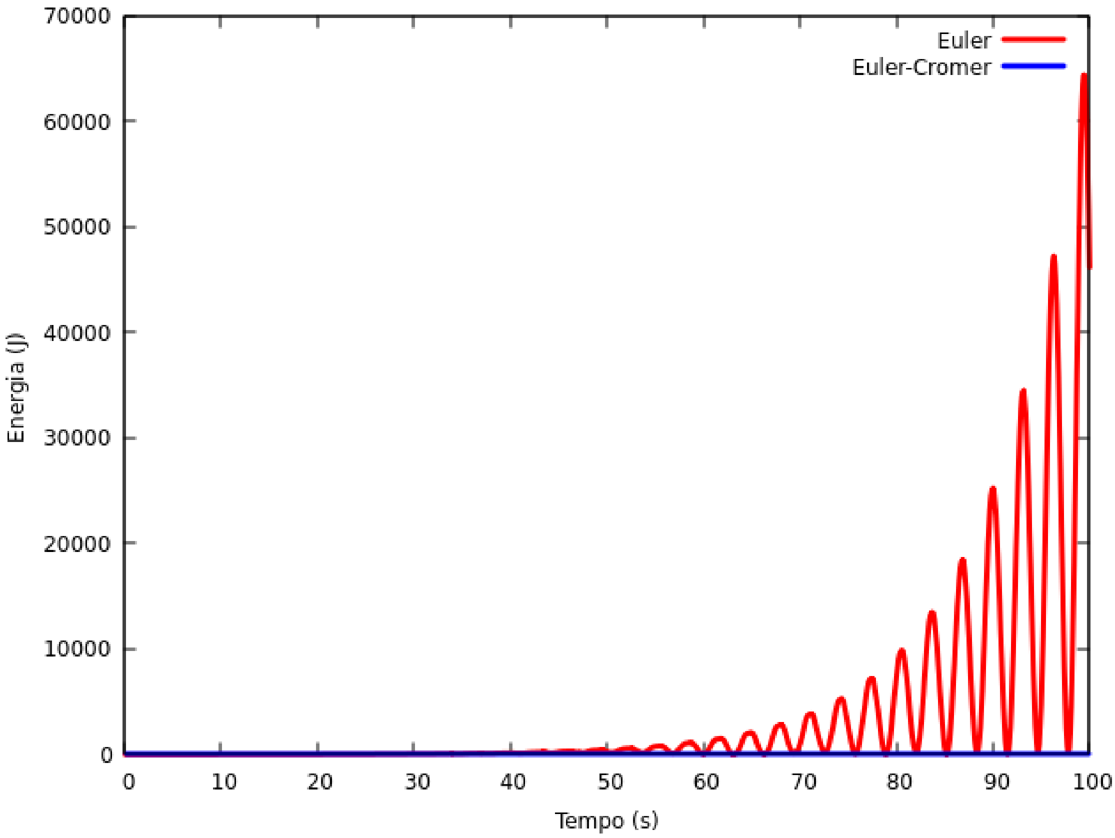

Figura 2: gráfico de $E(t)$ para os métodos de Euler e Euler-Cromer

Tarefa B

Essa tarefa, assim como todas as posteriores (C, D e E), são sobre um movimento mais geral de um pêndulo. Temos agora amortecimento (γ) e uma força senoidal imposta (F_0). A equação diferencial de segunda ordem que descreve o movimento da massa é:

$$\frac{d\omega}{dt} = -\frac{g}{l}sen(\theta) - \gamma\frac{d\theta}{dt} + F_0 sen(\Omega t)$$

que tornando-a discreta (adaptando o método de Euler-Cromer para o problemas mais geral) e usando o fato de que $\frac{d\theta}{dt} = \omega$, temos

$$\omega_{i+1} = \omega_i - \frac{g}{l}sen(\theta_i)\Delta t - \gamma\omega_i\Delta t + \Delta t F_0 sen(\Omega t) \tag{6}$$

e para o θ_{i+1} utilizaremos a mesma equação (4).

Tarefas B1 e B2

As tarefa B1 e B2 são sobre o cálculo do período do movimento descrito acima, recaindo no caso na tarefa A, pois $\gamma = F_0 = 0$. Devemos calculá-lo para alguns valores de θ_0, mostrando que depende da condição inicial.

O cálculo desse período deve ocorrer de três maneiras:

- Com a descrição do movimento, distância temporal entre cristas, que no caso é quando $\theta = \theta_0 t$, pois pela conservação de energia e como não há dissipação da mesma, esse é o máximo. Mas no código isso vai se traduzir em mudança de sinal do ângulo;

- Pelo cálculo numérico da integral elíptica a seguir

$$T = \sqrt{\frac{2l}{g}} \int_{\theta_0}^{-\theta_0} \frac{d\theta}{\sqrt{cos\theta - cos\theta_0}} \tag{7}$$

utilizando a Regra de Boole e somando um fator(método já explorado e explicado no projeto 3), sendo ele:

$$2 \cdot \sqrt{\frac{2l}{g} \frac{\epsilon}{sen\theta_0}};$$

- Pela aproximação em ângulos pequenos (verificar se realmente vale):

$$T \approx 2\pi\sqrt{\frac{l}{g}}(1 + \frac{\theta_0^2}{16}) \tag{8}$$

O código utilizado para isso foi o seguinte:

```fortran
c     tarefa b1
      implicit real*8(a-h, o-z)
      parameter(tempo = 100.d0) !tempo total do problema
      parameter(i = 1000) !número de passos
      parameter(g = 9.8d0) !aceleração gravitacional
      parameter(comp = 9.8d0) !comprimento do pêndulo
      parameter(pi = dacos(-1.d0))
      parameter(rm = 1.d0) !massa
      parameter(gamma = 0.d0)
      parameter(F0 = 0.d0)

c     arquivos de saída para graficar
      iout10 = 10 !unidade arquivo de saída
      open(unit=iout10, FILE='saida-b1eb2.dat')
      write(iout10, *) 'Theta0    ',
     +'                    Busca Direta   ',
     +'             Integral Elíptica    ',
     +'       Aproximação ângulos pequenos    '

      omega0 = 0.d0 !iniciando a velocidade angular
      deltaT = tempo/i !o valor de cada delta t
      theta0 = (5.d0/180.d0)*pi !5 graus em radianos

      do k = 1, 15
c     Busca direta do Período
c     Método de Euler-Cromer
      omega = omega0
      theta = theta0
```

```
      Tdireto = 0.d0
      n = 0
      do while (Tdireto == 0.d0)
        do j = 1, i
        !cálculo do omega
        t = j*deltaT
        omega = omega - (g/comp)*dsin(theta)*deltaT - gamma*omega*delt
     +aT +F0*deltaT*dsin(omegaF*t)
        !cálculo do theta
        theta1 = theta !para comparar
        theta = theta + deltaT*omega
        if (theta1 * theta < 0) then
              n = n + 1
              if (n == 1) then
                    Tesq = t
              else if (n == 3) then
                    Tdireto = t - Tesq
              end if
        end if
        end do
      end do

c     Integral elíptica - Regra de Boole
      h = theta0/i
      eps = 1.d-7 !arbitrário
      Tint = 0.d0
      do j = 1, (i/4)-1
          x0 = -theta0 + 4.d0*j*h + eps
          Tint = Tint + 7.d0*f(x0-4.d0*h, theta0) + 32.d0*f(x0-3.d0*h,
     +theta0) + 12.d0*f(x0-2.d0*h, theta0) + 32.d0*f(x0-1.d0*h, theta0)
     ++ 7.d0*f(x0, theta0)
      end do
      Tint = Tint * ((2.d0*h)/45.d0)
      Tint = Tint + 2.d0*dsqrt((2.d0*comp*eps)/(g*dsin(theta0)))

c     Aproximação de ângulos pequenos
      Tpeq = 2.d0 * pi * dsqrt(comp/g) * (1.d0 + ((theta0**2)/16))

      write(iout10,*) theta0, Tdireto, Tint, Tpeq

      theta0 = theta0 + (5.d0/180.d0)*pi !vou de 5 em 5 graus
      end do

      close(iout10)

      end

      function f(x, theta0)
```

```
            implicit real*8(a-h, o-z)
            f = 1.d0/dsqrt(dcos(x) - dcos(theta0))
      return
      end function
```

Algoritmo 2: código para resolução das tarefas B1 e B2

Assim, obtive o seguinte resultado

Theta0	Busca Direta	Integral Elíptica	Aproximação ângulos pequenos
8.7266462599716474E-002	6.3000000000000007	5.8719025654995560	6.2861758817050779
0.17453292519943295	6.3000000000000007	5.9953882422420390	6.2951476052815529
0.26179938779914941	6.3000000000000007	6.0944095441157771	6.3101004779090131
0.34906585039886590	6.3000000000000007	6.1824212066234177	6.3310344995874566
0.43633231299858238	6.3000000000000007	6.2648281189161805	6.3579496703168825
0.52359877559829882	6.4000000000000004	6.3445391737864627	6.3908459900972945
0.61086523819801530	6.4000000000000004	6.4234033509347892	6.4297234589286889
0.69813170079773179	6.5000000000000000	6.5027417703838761	6.4745820768110658
0.78539816339744828	6.6000000000000014	6.5835893651744630	6.5254218437444287
0.87266462599716477	6.6000000000000014	6.6668213887204546	6.5822427597287749
0.95993108859688125	6.6000000000000005	6.7532277040250781	6.6450448247641036
1.0471975511965976	6.7000000000000002	6.8435613056967757	6.7138280388504166
1.1344640137963140	6.7999999999999998	6.9385736625262897	6.7885924019877129
1.2217304763960304	7.0000000000000009	7.0390435627041175	6.8693379141759943
1.3089969389957468	7.0000000000000009	7.1458034851946532	6.9560645754152590

Figura 3: valores obtidos para a resolução das tarefas B1 e B2

Logo, conseguimos concluir que o método da integral elíptica é muito bom e próximo do encontrado diretamente e que o método que serve apenas para ângulos pequenos é realmente muito preciso para tal tarefa.

Tarefa B3

Essa tarefa nos pede para considerar que $\gamma = 0,5$ e $F_0 = 0$, e plotar seu gráfico $\theta(t)$. Assim, devemos classificar o tipo de amortecimento, pela "cara"do gráfico.

O código para implementar o que foi proposto está disposto logo abaixo.

```
c     tarefa b3
      implicit real*8(a-h, o-z)
      parameter(tempo = 100.d0) !tempo total do problema
      parameter(i = 1000) !número de passos
      parameter(g = 9.8d0) !aceleração gravitacional
      parameter(comp = 9.8d0) !comprimento do pêndulo
      parameter(pi = dacos(-1.d0))
      parameter(rm = 1.d0) !massa
      parameter(gamma = 0.5d0)
      parameter(F0 = 0.d0)

c     arquivos de saída para graficar
      iout10 = 10 !unidade arquivo de saída
      open(unit=iout10, FILE='saida-b3.dat')

      omega0 = 0.d0 !iniciando a velocidade angular
      deltaT = tempo/i !o valor de cada delta t
```

```
      theta0 = (15.d0/180.d0)*pi !15 graus em radianos

c     Método de Euler-Cromer
      omega = omega0
      theta = theta0
      do j = 1, i
      !cálculo do omega
      t = j*deltaT
      omega = omega - (g/comp)*dsin(theta)*deltaT - gamma*omega*delt
     +aT +F0*deltaT*dsin(omegaF*t)
      !cálculo do theta
      theta = theta + deltaT*omega
      write(iout10,*) t, theta
      end do

      close(iout10)

      end
```

Algoritmo 3: código para resolução da tarefa B3

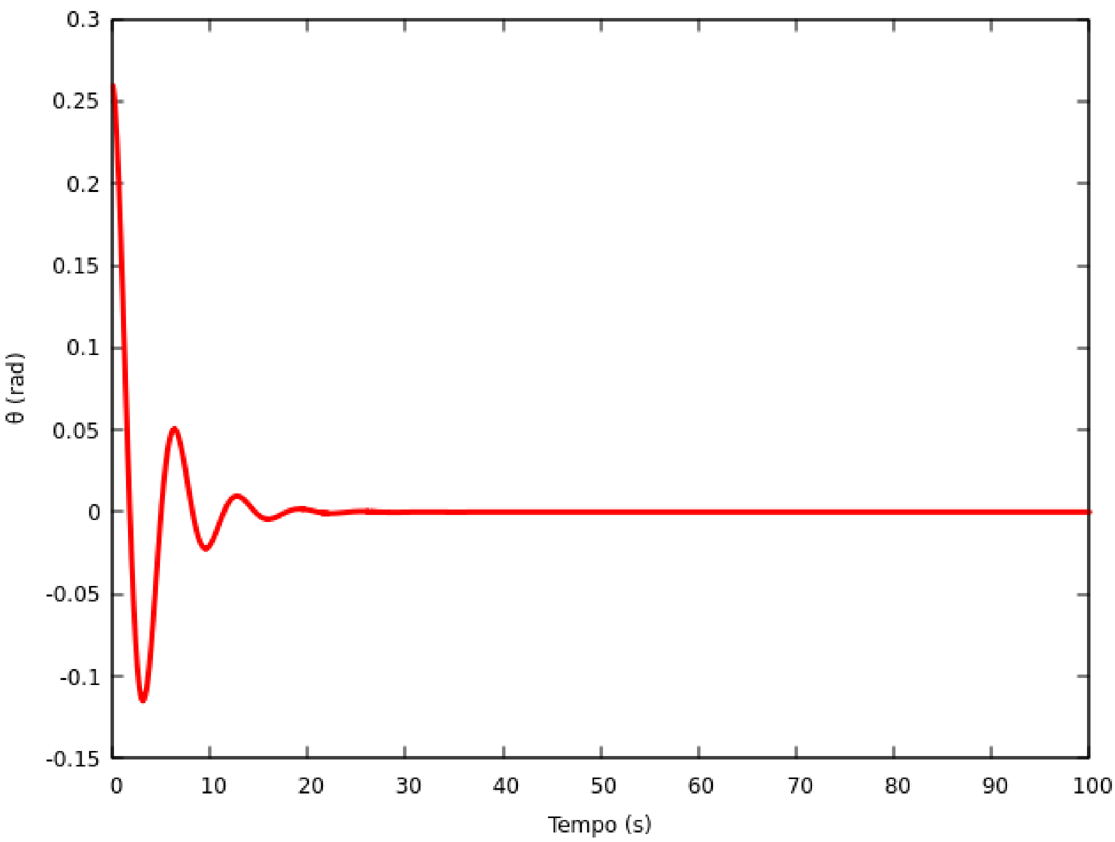

Figura 4: $\theta(t)$ para a tarefa B3

Assim, é evidente que temos um amortecimento <u>subcrítico</u>, em que vemos uma função senoide modulada por uma exponencial decrescente.

Tarefa B4

Esta tarefa nos pede para considerar o seguinte caso: $\gamma = 0,5$, $\Omega = 2/3$, $\Delta t = 0,03$, e os seguintes casos $F_0 = 0$, $F_0 = 0,5$,$F_0 = 1,2$. Devemos plotar os gráficos de $\theta(t)$ e $\omega(t)$ para os três casos de F_0, afim de compará-los. Como o método já foi discutido acima, devo apenas colocar o código implemntado abaixo, já que pelas características das tarefas o método é o mesmo.

```fortran
c     tarefa b4
      implicit real*8(a-h, o-z)
      parameter(tempo = 30.d0) !tempo total do problema
      parameter(i = 1000) !número de passos
      parameter(g = 9.8d0) !aceleração gravitacional
      parameter(comp = 9.8d0) !comprimento do pêndulo
      parameter(pi = dacos(-1.d0))
      parameter(rm = 1.d0) !massa
      parameter(gamma = 0.5d0)
      parameter(F0 = 1.2d0) !força que vou madar de acordo com o caso
      parameter(omegaF = 2.d0/3.d0)

c     arquivos de saída para graficar
      iout10 = 10 !unidade arquivo de saída
      open(unit=iout10, FILE='saida-b4-f12.dat') !vou mudando o nome do arquivo de acordo com o F0

      omega0 = 0.d0 !iniciando a velocidade angular
      deltaT = tempo/i !o valor de cada delta t
      theta0 = (15.d0/180.d0)*pi !15 graus em radianos

c     Método de Euler-Cromer
      omega = omega0
      theta = theta0
      do j = 1, i
      !cálculo do omega
      t = j*deltaT
      omega = omega - (g/comp)*dsin(theta)*deltaT - gamma*omega*delt
     +aT +F0*deltaT*dsin(omegaF*t)
      !cálculo do theta
      theta = theta + deltaT*omega
      write(iout10,*) t, mod(theta, 2*pi), omega
      end do

      close(iout10)

      end
```

Algoritmo 4: código para resolução da tarefa B4

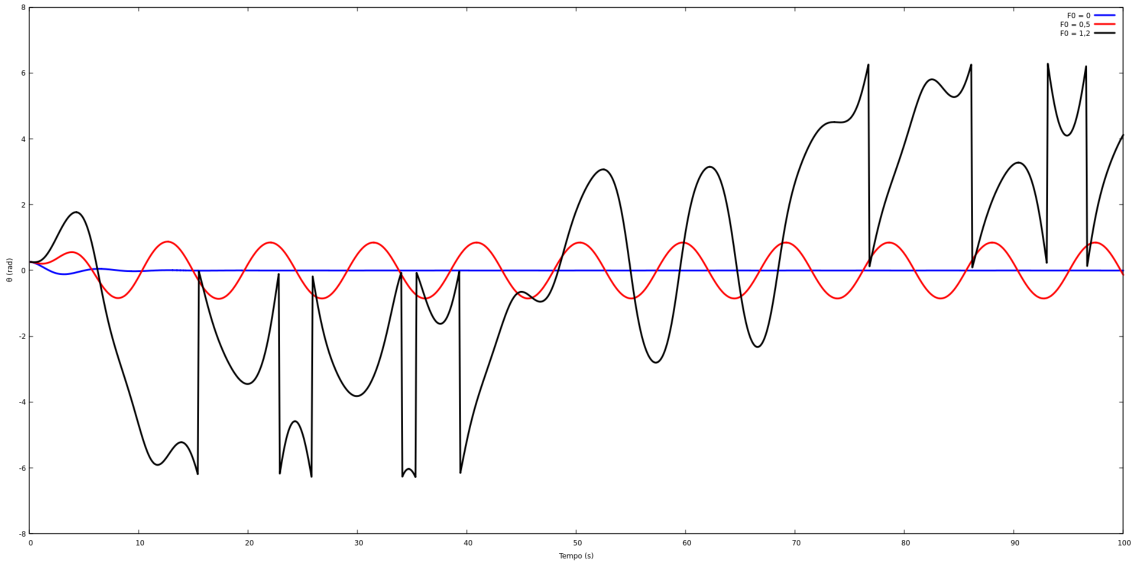

Figura 5: $\theta(t)$ para os três casos juntos

Como podemos observar na figura 4 acima, temos que o caso $F_0 = 0$ acaba caindo no caso da tarefa C, um amortecimento subcrítico, portanto um movimento periódico.

O caso $F_0 = 0,5$, como conseguimos notar, é um exemplo clássico de movimento oscilatório forçado, que no começo temos o movimento tentando se estabilizar e a partir do quarto pico, no caso, começamos um estado periódico com velocidade angular proveniente de Ω.

Já no caso $F_0 = 1,2$, esse estado de movimento não é alcançado. Não conseguimos de maneira trivial prever seu movimento, se caracterizando como um movimento não periódico, no caso, caótico.

O que foi explorado acima serve, também, para o gráfico abaixo, da velocidade angular em função do tempo. complementa o que foi observado de movimento periódico previsível e caótico não previsível.

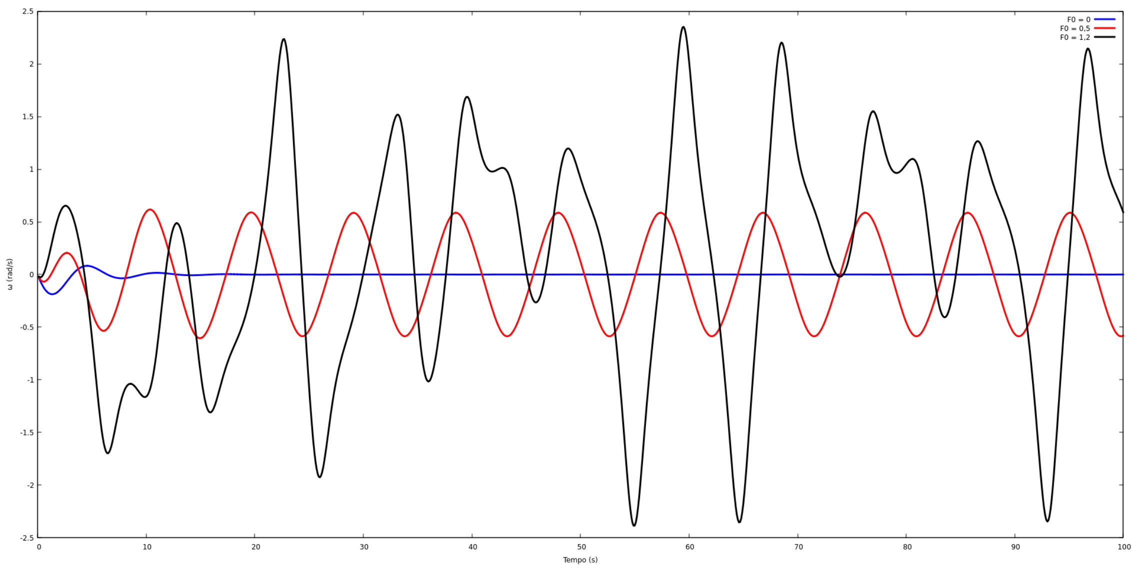

Figura 6: $\omega(t)$ para os três casos juntos

Tarefa C

A tarefa C ainda explora os casos acima em que $F_0 = 0,5$ e $F_0 = 1,2$. Só que agora devemos tratar, para cada caso, dois pêndulos forçados mas com ângulo inicial, θ_0, diferentes, ou seja, condições iniciais diferentes.

Estudaremos como que a diferença entre a posição angular deles, $\Delta\theta$ evolui em relação a passagem do tempo, de forma que temos a relação de Liapunov, com seu expoente λ:

$$\Delta\theta \approx \exp \lambda t \tag{9}$$

aplicando o logritmo natural em ambos os lados da equação:

$$ln(\Delta\theta) = \lambda t \tag{10}$$

uma função linear.

Segundo Liapunov, se $\lambda > 0$, temos um movimento caótico e o contrário, $\lambda < 0$, temos um movimento não caótico. Isso pode ser interpretado como conforme o tempo avança, se os movimentos de aproximam ou afastam, drasticamente. Para isso, vou utilizar a diferença inicial da posição angular de 0,001 rad.

Logo, irei plotar, para cada caso, o gráfico de , para encontrar em um *fitting* linear o valor aproximado de λ e observar o comportamento. o código implementado está logo abaixo.

```
c     tarefa c
      implicit real*8(a-h, o-z)
      parameter(tempo = 30.d0) !tempo total do problema
      parameter(i = 1000) !número de passos
      parameter(g = 9.8d0) !aceleração gravitacional
      parameter(comp = 9.8d0) !comprimento do pêndulo
      parameter(pi = dacos(-1.d0))
      parameter(rm = 1.d0) !massa
      parameter(gamma = 0.5d0)
      parameter(F0 = 1.2d0) !força que vou alternando para os casos
      parameter(omegaF = 2.d0/3.d0)
      parameter(theta01 = 0.221d0)
      parameter(theta02 = 0.222d0)

c     arquivos de saída para graficar
      iout10 = 10 !unidade arquivo de saída
      open(unit=iout10, FILE='saida-c-f12.dat')

      omega01 = 0.d0
      omega02 = 0.d0
      deltaT = tempo/i !o valor de cada delta t
      deltaTheta = theta02 - theta01 !diferença de ângulos inicial

c     Método de Euler-Cromer
      omega1 = omega01
      omega2 = omega02
      theta1 = theta01
```

```
      theta2 = theta02
      do j = 1, i
      !cálculo do omega
      t = j*deltaT
      omega1 = omega1 - (g/comp)*dsin(theta1)*deltaT - gamma*omega1*del
     +taT +F0*deltaT*dsin(omegaF*t)
      omega2 = omega2 - (g/comp)*dsin(theta2)*deltaT - gamma*omega2*del
     +taT +F0*deltaT*dsin(omegaF*t)
      !cálculo do theta
      theta1 = theta1 + deltaT*omega1
      theta2 = theta2 + deltaT*omega2

      deltaTheta = theta2 - theta1

      write(iout10,*) t, dlog(abs(deltaTheta))
      end do

      close(iout10)

      end
```

Algoritmo 5: código para resolução da tarefa C

Os resultados obtidos estão na figura a seguir. Vemos claramente que no caso de $F_0 = 0,5$, temos os pêndulos convergindo para uma mesma posição, pois o gráfico linear é decrescente. O valor de λ encontrado foi de -0,25, aproximadamente. Logo, o movimento não é caótico, como vimos na tarefa anterior. Já no caso $F_0 = 1,2$, temos mais uma confirmação, o movimento é realmente caótico. O gráfico linear que melhor descreve é crescente e tem coeficiente angular, λ, de valor aproximado a 0,08.

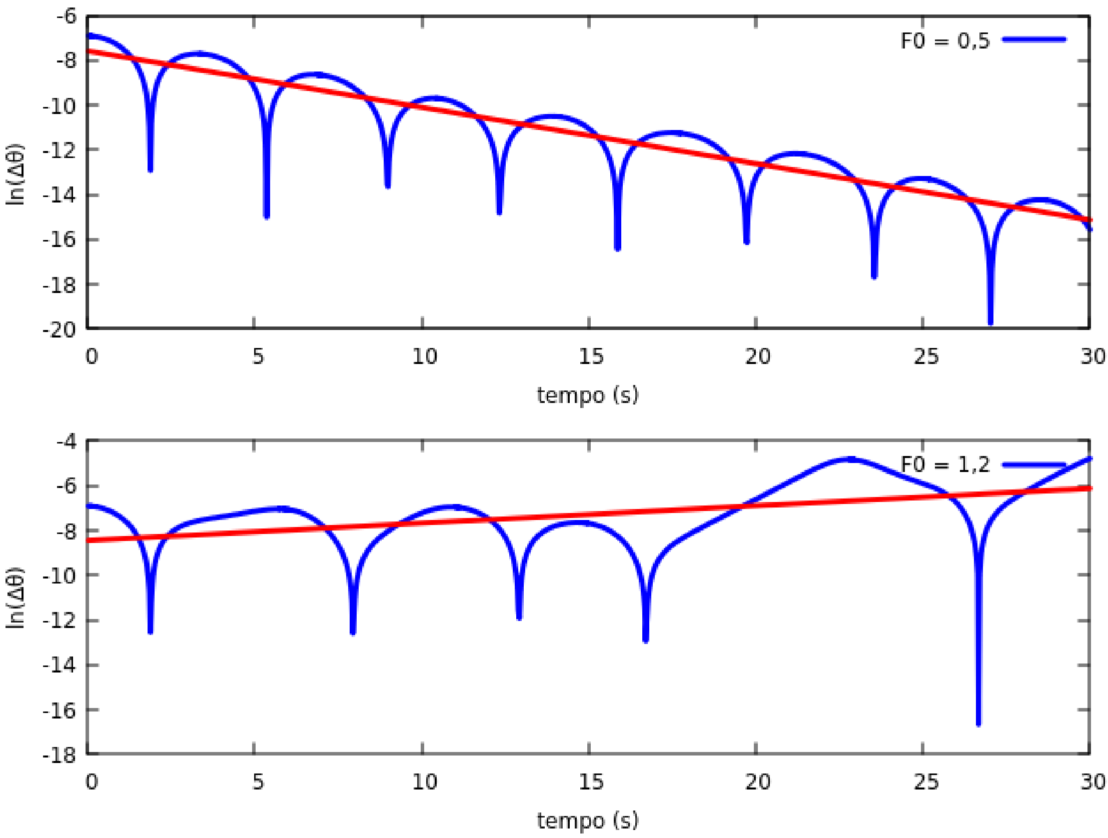

Figura 7: $ln(\Delta\theta)$ versus *tempo* para os dois casos

Tarefa D

A tarefa D complementa a anterior. Agora iremos visualizar para cada caso o gráfico de $\omega(\theta)$. O código implementado, que não traz novidades, está disposto logo abaixo.

```
c     tarefa d
      implicit real*8(a-h, o-z)
      parameter(tempo = 100.d0) !tempo total do problema
      parameter(i = 1000) !número de passos
      parameter(g = 9.8d0) !aceleração gravitacional
      parameter(comp = 9.8d0) !comprimento do pêndulo
      parameter(pi = dacos(-1.d0))
      parameter(rm = 1.d0) !massa
      parameter(gamma = 0.5d0)
      parameter(F0 = 1.2d0) !força que vou alternando para os casos
      parameter(omegaF = 2.d0/3.d0)
      parameter(theta01 = 0.221d0)
      parameter(theta02 = 0.222d0)

c     arquivos de saída para graficar
      iout10 = 10 !unidade arquivo de saída
      open(unit=iout10, FILE='saida-d-f12-p1.dat')
      iout11 = 11 !unidade arquivo de saída
```

```fortran
      open(unit=iout11, FILE='saida-d-f12-p2.dat')

      omega01 = 0.d0
      omega02 = 0.d0
      deltaT = tempo/i !o valor de cada delta t
      deltaTheta = theta02 - theta01 !diferença de ângulos inicial

c     Método de Euler-Cromer
      omega1 = omega01
      omega2 = omega02
      theta1 = theta01
      theta2 = theta02
      do j = 1, i
      !cálculo do omega
      t = j*deltaT
      omega1 = omega1 - (g/comp)*dsin(theta1)*deltaT - gamma*omega1*del
     +taT +F0*deltaT*dsin(omegaF*t)
      omega2 = omega2 - (g/comp)*dsin(theta2)*deltaT - gamma*omega2*del
     +taT +F0*deltaT*dsin(omegaF*t)
      !cálculo do theta
      theta1 = theta1 + deltaT*omega1
      theta2 = theta2 + deltaT*omega2

      write(iout10,*) theta1, omega1
      write(iout11,*) theta2, omega2
      end do

      close(iout10)
      close(iout11)

      end
```

Algoritmo 6: código para resolução da tarefa D

Os gráficos obtidos são os abaixo.

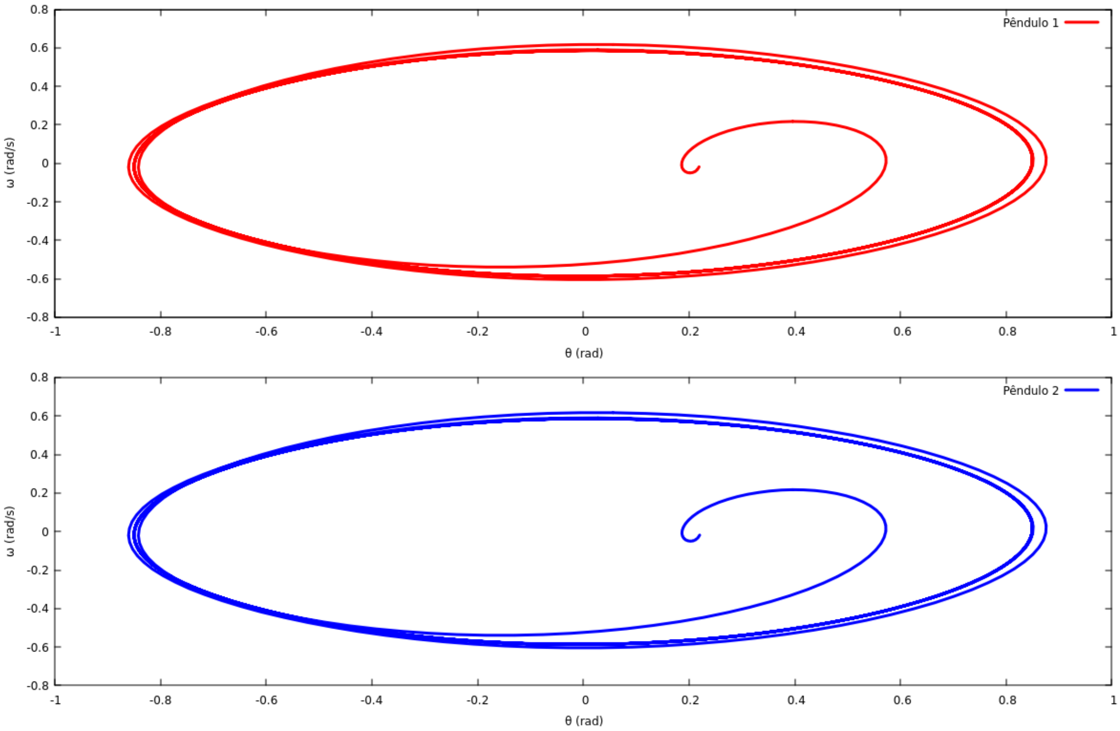

Figura 8: $\omega(\theta)$ para $F_0 = 0,5$, separados

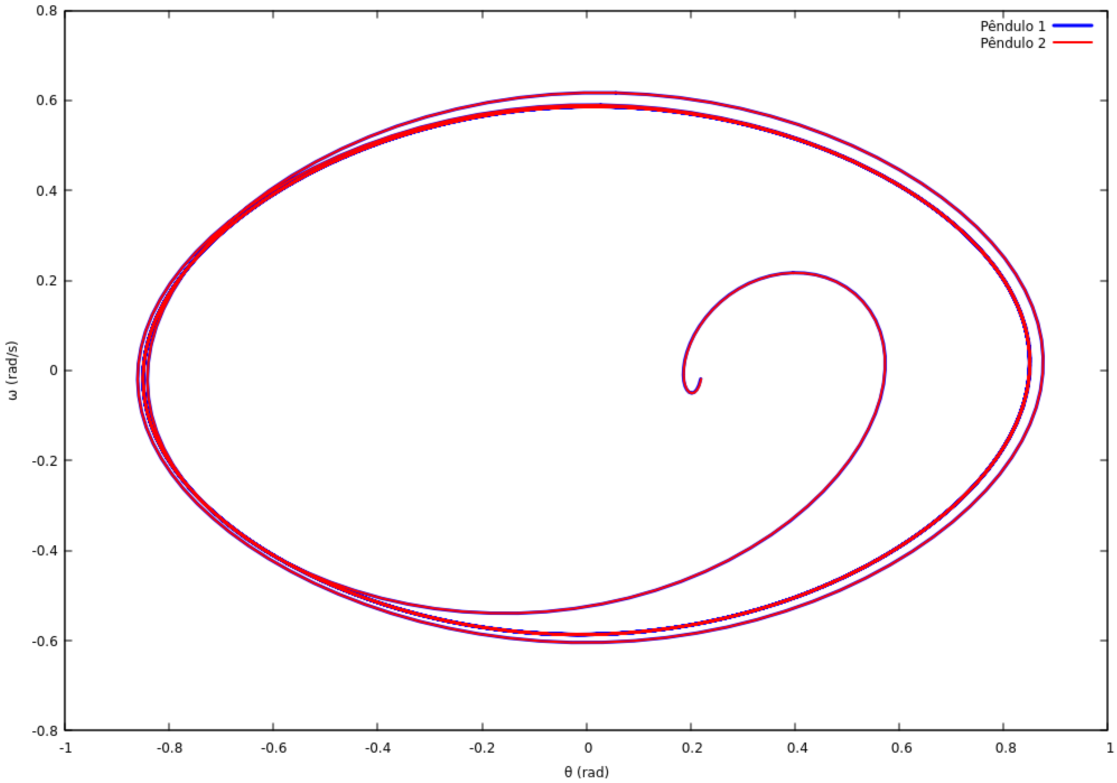

Figura 9: $\omega(\theta)$ para $F_0 = 0,5$, juntos

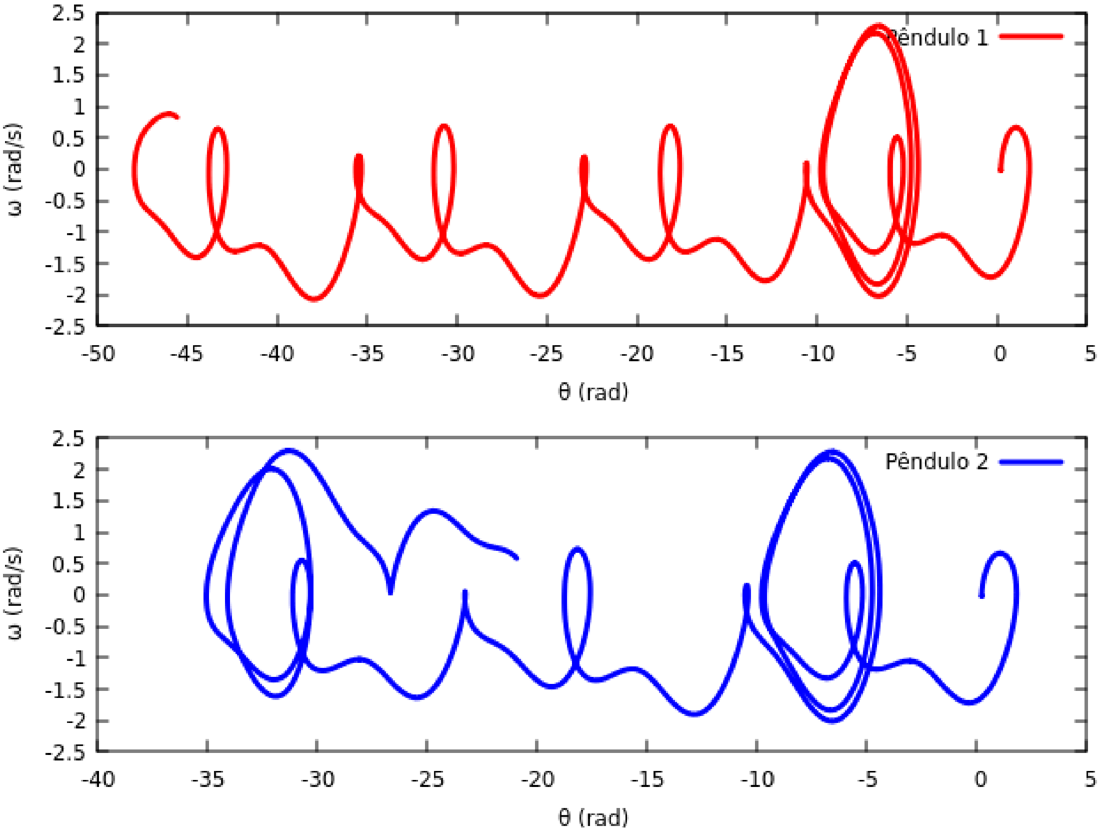

Figura 10: $\omega(\theta)$ para $F_0 = 1,2$, separados

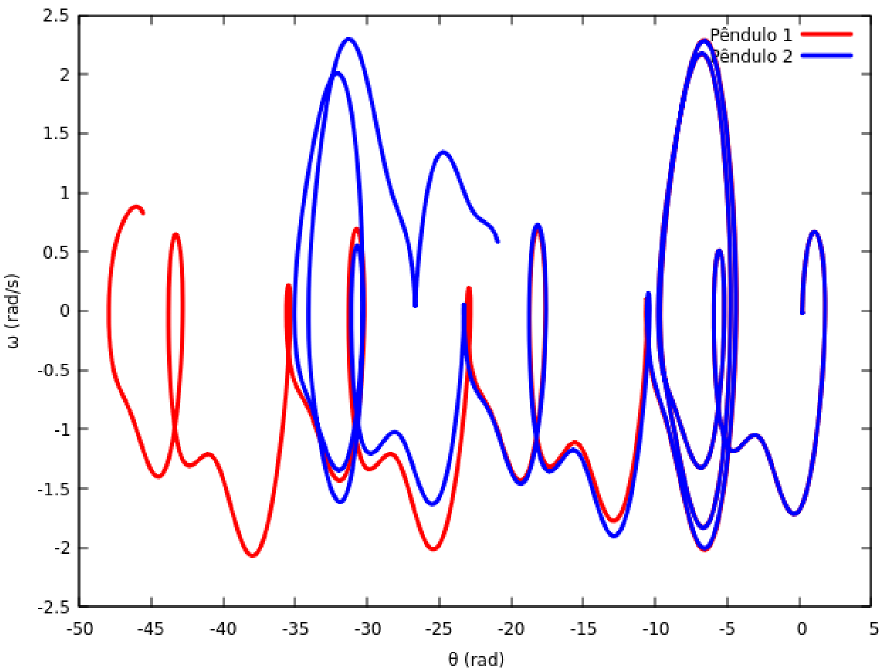

Figura 11: $\omega(\theta)$ para $F_0 = 0,5$, juntos

Assim conseguimos visualizar com clareza como o movimento no caso $F_0 = 0,5$ converge e o do caso $F_0 = 1,2$ diverge. Decidi colocar, para cada um dos casos, uma plotagem separada de cada pêndulo e depois juntos. Dessa forma conseguimos observar como eles evoluem separadamente e qual as diferenças entre eles.

No caso $F_0 = 0,5$, vemos que os gráficos começam pouco separados e depois são iguais, quase não conseguimos diferenciá-los juntos. Já no outro caso, temos um movimento claramente caótico que diverge, pois começam, como no primeiro caso, pouco separados e logo ficam completamente diferentes.

Tarefa E

A tarefa E serve para visualizarmos o que foi estudo na tarefa anterior com mais clareza. Neste caso, vamos utilizar a secção de Poincaré, em que plotamos no gráfico de $\omega(\theta)$ apenas os pontos em que a seguinte igualdade é verdadeira, para n inteiro:

$$\Omega t = n\pi$$

ou seja, quando Ωt é múltiplo de π, que foi a lógica utilizada no *if* do código implementado (coloquei que *mod(omegaF*t, pi) .lt. 0.000000001* e não 0 para levar em conta erros intrínsecos a limitação de precisão do computador e da linguagem). O código ultilzado está disposto logo abaixo.

```
c      tarefa e
       implicit real*8(a-h, o-z)
       parameter(tempo = 100.d0) !tempo total do problema
       parameter(i = 1000) !número de passos
       parameter(g = 9.8d0) !aceleração gravitacional
       parameter(comp = 9.8d0) !comprimento do pêndulo
       parameter(pi = dacos(-1.d0))
       parameter(rm = 1.d0) !massa
       parameter(gamma = 0.5d0)
       parameter(F0 = 1.2d0) !força que vou alternando para os casos
       parameter(omegaF = 2.d0/3.d0)
       parameter(theta01 = 1.2d0)
       parameter(theta02 = 2.4d0)

c      arquivos de saída para graficar
       iout10 = 10 !unidade arquivo de saída
       open(unit=iout10, FILE='saida-e-f12-p1.dat')
       iout11 = 11 !unidade arquivo de saída
       open(unit=iout11, FILE='saida-e-f12-p2.dat')

       omega01 = 0.d0
       omega02 = 0.d0
       deltaT = tempo/i !o valor de cada delta t

c      Método de Euler-Cromer
       omega1 = omega01
       omega2 = omega02
       theta1 = theta01
       theta2 = theta02
```

```
  do j = 1, i
  !cálculo do omega
  t = j*deltaT
  omega1 = omega1 - (g/comp)*dsin(theta1)*deltaT - gamma*omega1*del
 +taT +F0*deltaT*dsin(omegaF*t)
  omega2 = omega2 - (g/comp)*dsin(theta2)*deltaT - gamma*omega2*del
 +taT +F0*deltaT*dsin(omegaF*t)
  !cálculo do theta
  theta1 = theta1 + deltaT*omega1
  theta2 = theta2 + deltaT*omega2
  theta1 = mod(theta, 2*pi)
  theta2 = mod(theta, 2*pi)

  if (mod(omegaF*t, pi) .lt. 0.000000001) then
      write(iout10,*) theta1, omega1
      write(iout11,*) theta2, omega2
  end if

  end do

  close(iout10)
  close(iout11)

  end
```

Algoritmo 7: código para resolução da tarefa E

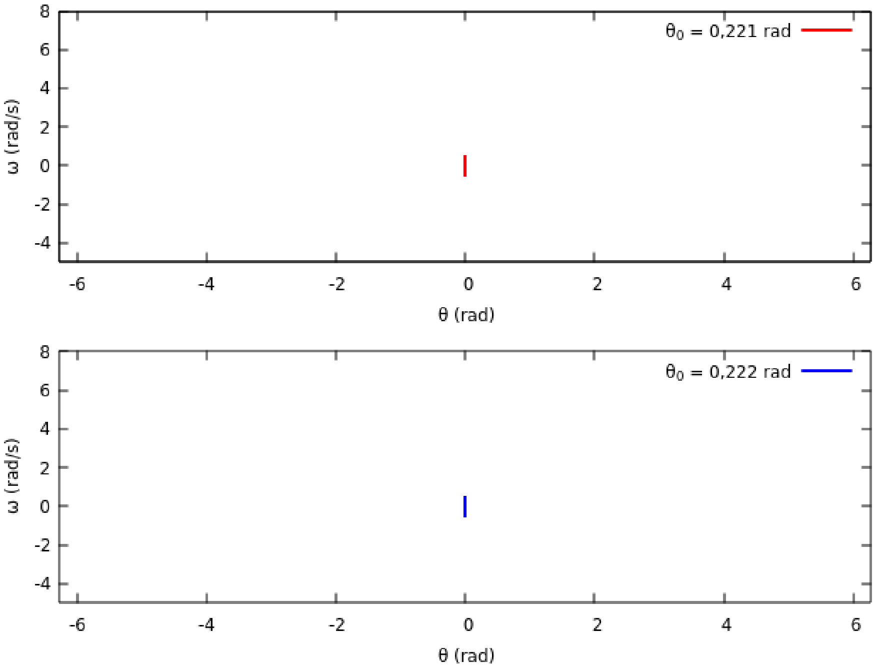

Figura 12: $\omega(\theta)$ para $F_0 = 0,5$ na condição inicial descrita no gráfico

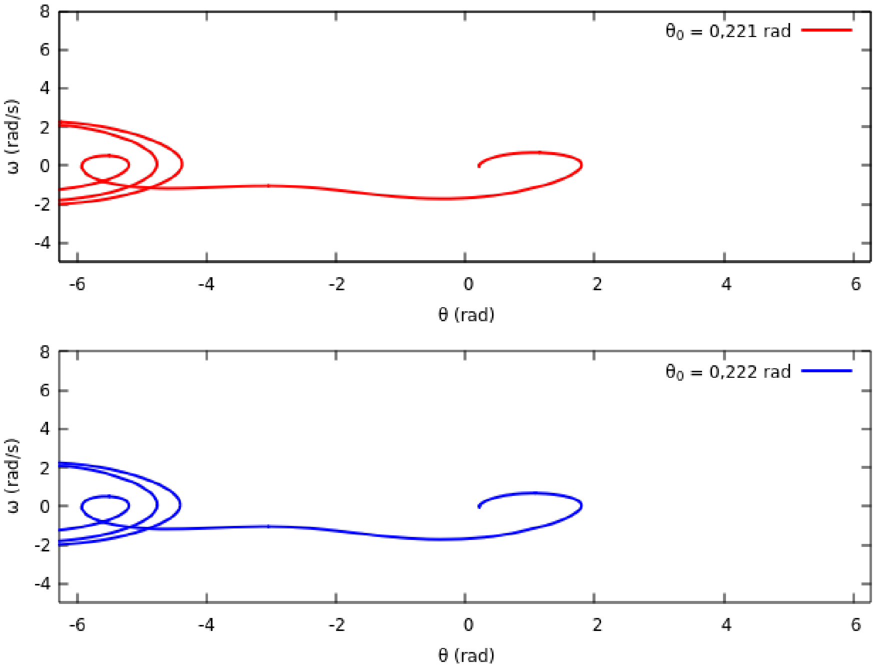

Figura 13: $\omega(\theta)$ para $F_0 = 1,2$ na condição inicial descrita no gráfico

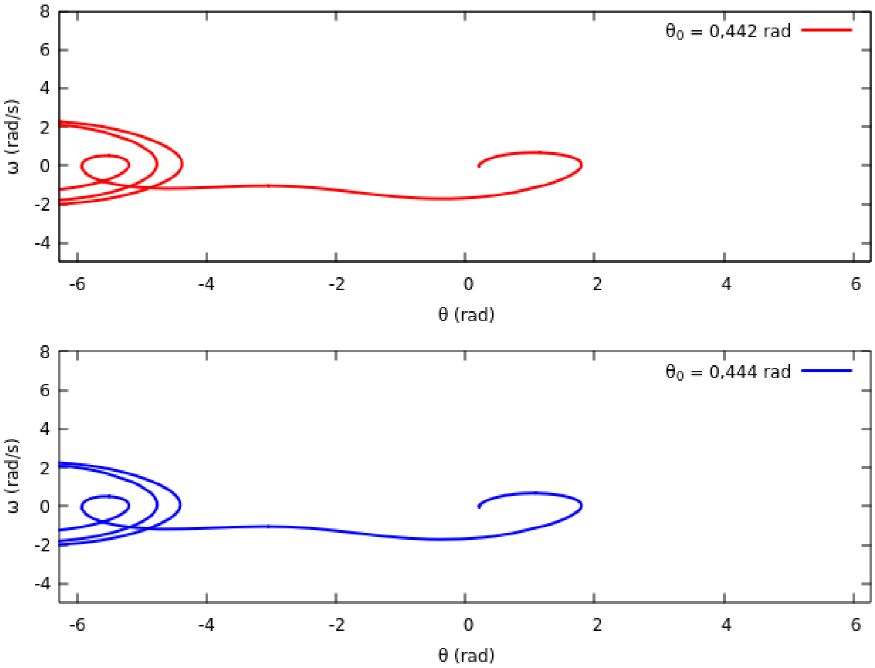

Figura 14: $\omega(\theta)$ para $F_0 = 1,2$ na condição inicial descrita no gráfico

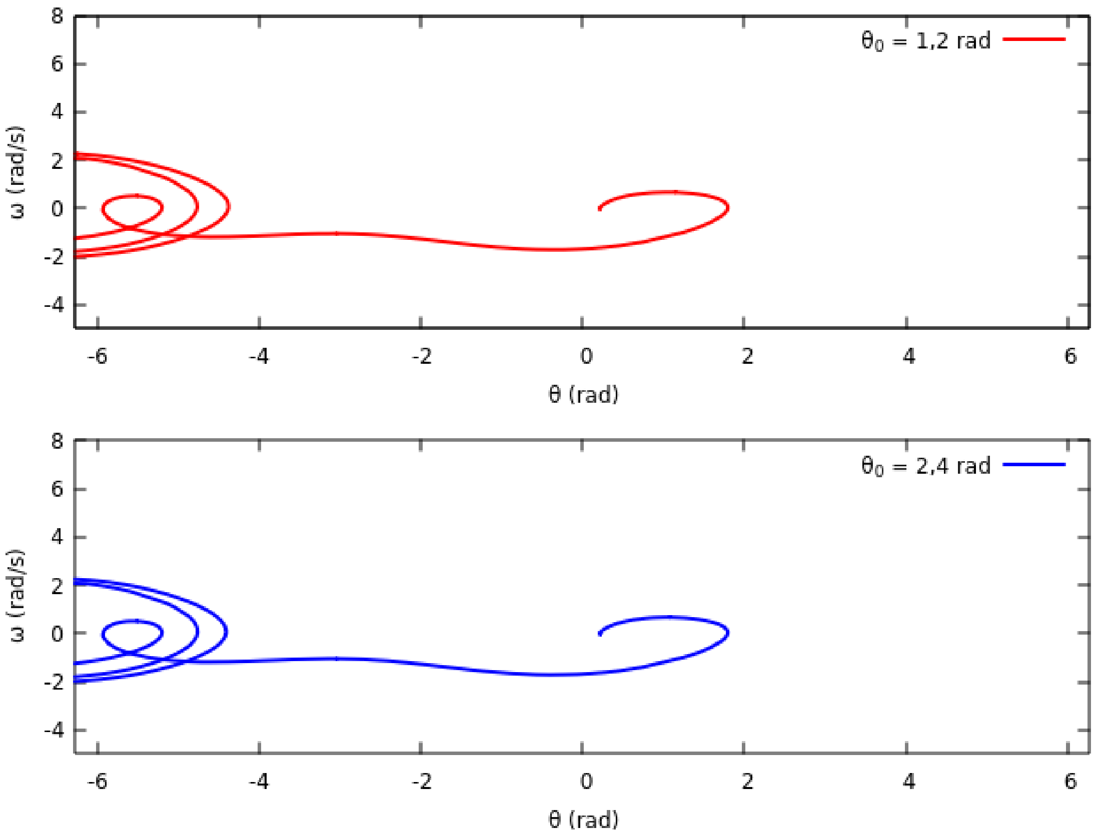

Figura 15: $\omega(\theta)$ para $F_0 = 1,2$ na condição inicial descrita no gráfico

Assim, temos as seguintes conclusões para o que foi visto nos gráficos acima.

- As figuras não se alteram se mudamos as condições iniciais, visto que as testadas foram complemente diferentes entre si, demonstrando a universalidade do caos;
- As figuras para movimentos não caóticos (determinísticos) são realmente um ponto, visto que por alguma limitação de precisão intrínseca a linguagem e ao computador vemos uma pequena reta, quase um ponto;
- As figuras que mostram os movimentos caóticos, os atratores estranhos podem ser visualizadas acima.