Universidade de São Paulo

Introdução à Física Computacional - Projeto 4

João Victor Dell Agli Floriano - 10799783

1. Tarefa A

Este programa tem como objetivo simular numericamente o movimento de um pêndulo simples, por meio de sua coordenada θ, a partir do método de euler, que consiste no seguinte procedimento para o cálculo da posição e velocidade angular:

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

$$\theta_{i+1} = \theta_i + \omega_i \Delta t \quad (2)$$

Este método, realizado iterativamente em n passos, é uma maneira inicialmente interessante e simples de realizar essa simulação. Com os valores de $l = 9,8\,m$, $g = 9,8\,m/s^2$, $m = 1\,kg$, $\omega_0 = 0.0$, $\theta_0 = \pi/8$ $\Delta t = 0.01s$, $n = 55000$, calcularemos a energia do pêndulo em função do tempo, assim como sua posição angular. O código que implementa esse algoritmo é:

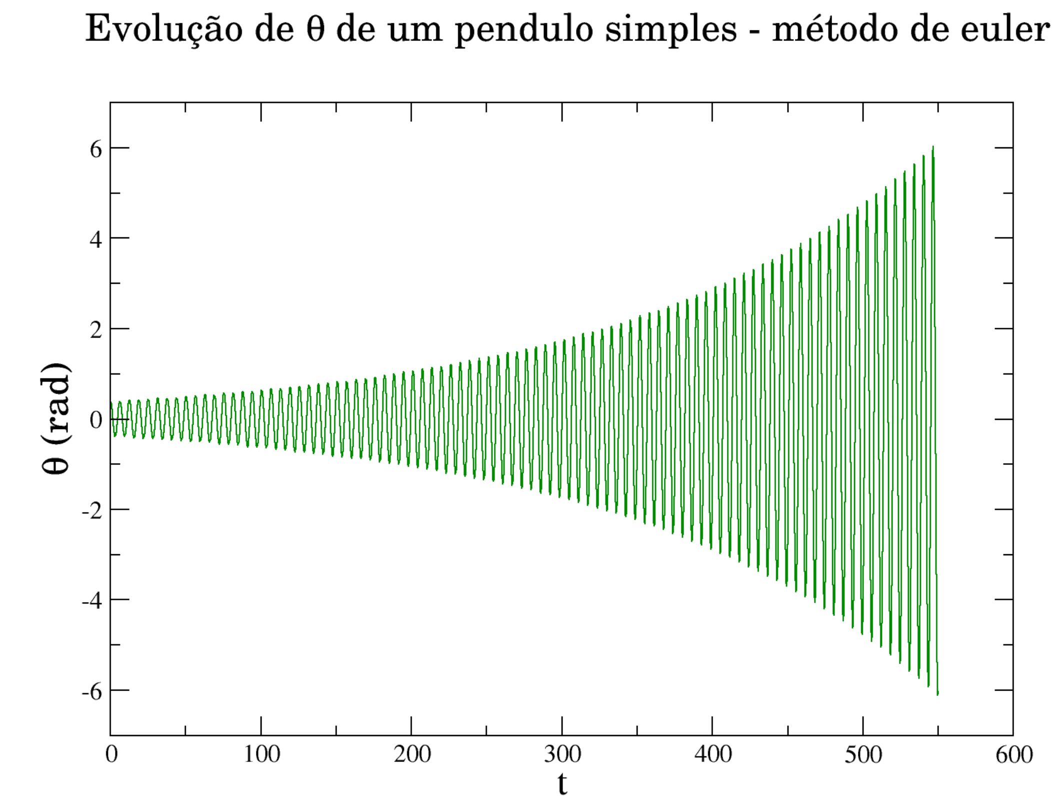
```
program main
      implicit double precision (a-h, o-z)





      g = 98d-1
      pi = 4d0*datan(1d0)

      rl = 98d-1
      m = 1d0
      theta = pi/8d0
      w = 0d0
      e = 0d0

      dt = 1d-2






      open(20, file = "saida-A-1-10799783.dat")
      open(21, file = "saida-A-2-10799783.dat")
```

```
      do i = 1, 55000

            w0 = w
            w = w - (g/rl)*theta*dt
            theta = theta + w0*dt

            !atualizacao do theta para que não passe 2*pi
            theta = dmod(theta, 2d0*pi)



            e = (m*(rl**2)*w**2)/2d0 + m*g*rl*(theta**2)/2d0

            write(20,*) i*dt, theta
            write(21,*) i*dt, e


      end do

      close(20)
      close(21)


     stop
end program main
```

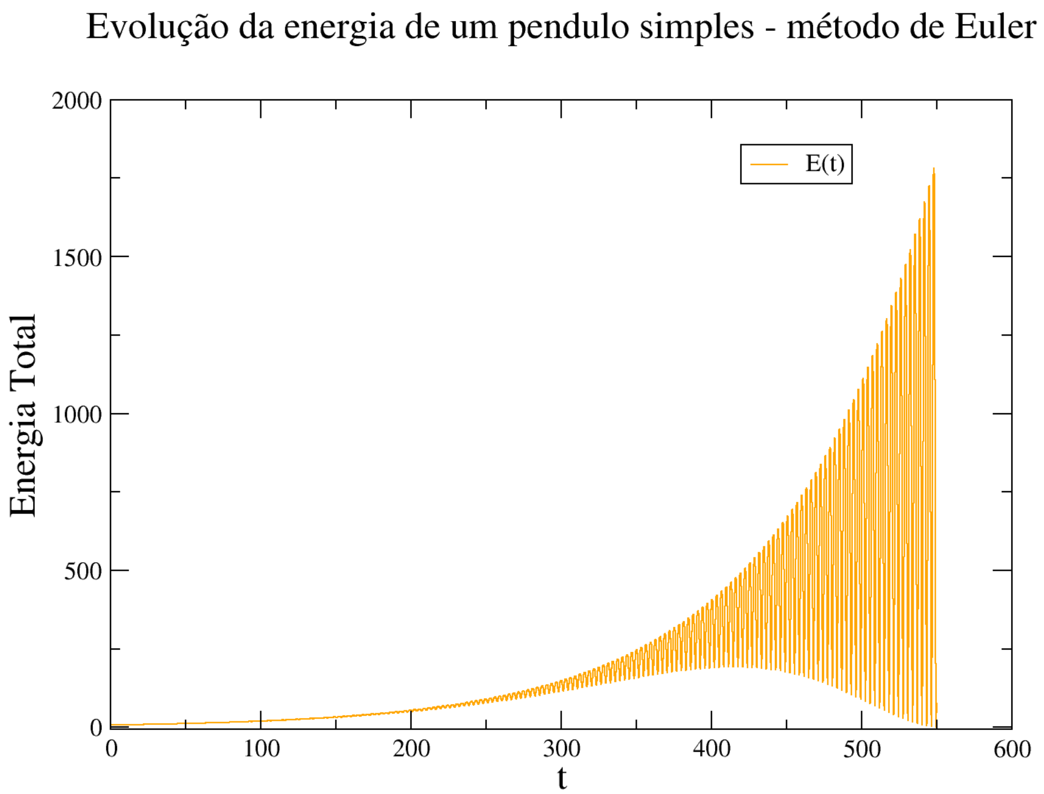
Que, ao ser executado:

```
joao@joao-Inspiron-15-7000-Gaming: ~/Área de Trabalho/Intro a Fiscomp/Quarto P...
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
o Projeto$ cd tarefaA
(base) joao@joao-Inspiron-15-7000-Gaming:~/Área de Trabalho/Intro a Fiscomp/Quar
o Projeto/tarefaA$ f77 A.f -o A
(base) joao@joao-Inspiron-15-7000-Gaming:~/Área de Trabalho/Intro a Fiscomp/Quar
o Projeto/tarefaA$ ./A
(base) joao@joao-Inspiron-15-7000-Gaming:~/Área de Trabalho/Intro a Fiscomp/Quar
o Projeto/tarefaA$
```

Demonstra um resultado um tanto preocupante para o ângulo e a energia:



Evolução de θ de um pendulo simples - método de euler
θ (rad)
6
4
2
0
-2
-4
-6
0
100
200
300
400
500
600
t



Evolução da energia de um pendulo simples - método de Euler
Energia Total
2000
1500
1000
500
0
E(t)
0
100
200
300
400
500
600
t

Por este método, é verificado um aumento na amplitude do pêndulo, o que implica em um aumento da energia do sistema, porém, conhecendo a disposição e todas as variáveis que envolvem o problema do pêndulo simples, sujeito apenas à ação da gravidade, uma força conservativa, e de nenhuma força externa mais, é evidente que esse resultado está errado, pois a energia deveria ter se conservado ao longo de todo o problema, tendo como valor máximo seu valor inicial. Ou seja: este método não garante a coonservação de energia!

Uma correção para o método de Euler é viável pelo método derivado deste, de Euler Cromer, que aplica apenas uma pequena alteração na equação do θ :

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

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

Com as mesmas condições iniciais anteriores, aplicando agora esta alteração, temos:

```
program main
       implicit double precision (a-h, o-z)
       double precision coord(2)



       !conserto ao método anterior
       g = 98d-1
       pi = 4d0*datan(1d0)

       rl = 98d-1
       m = 1d0
       theta = pi/8d0
       w = 0d0
       e = 0d0

       dt = 1d-2
```

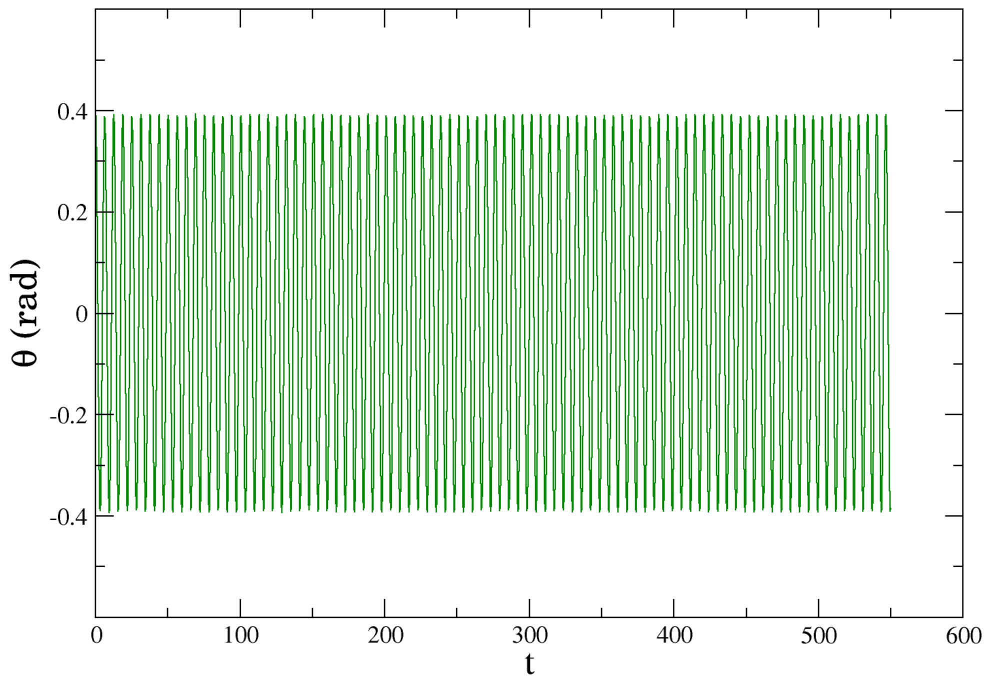
```
      open(22, file = "saida-A-3-10799783.dat")
      open(23, file = "saida-A-4-10799783.dat")


      do i = 1, 55000

             w = w - (g/rl)*theta*dt
             theta = theta + w*dt



             e = (m*(rl**2)*w**2)/2d0 + m*g*rl*(theta**2)/2d0

             write(22,*) i*dt, theta
             write(23,*) i*dt, e


      end do

      close(22)
      close(23)



      stop
end program main
```

Que, ao ser executado:

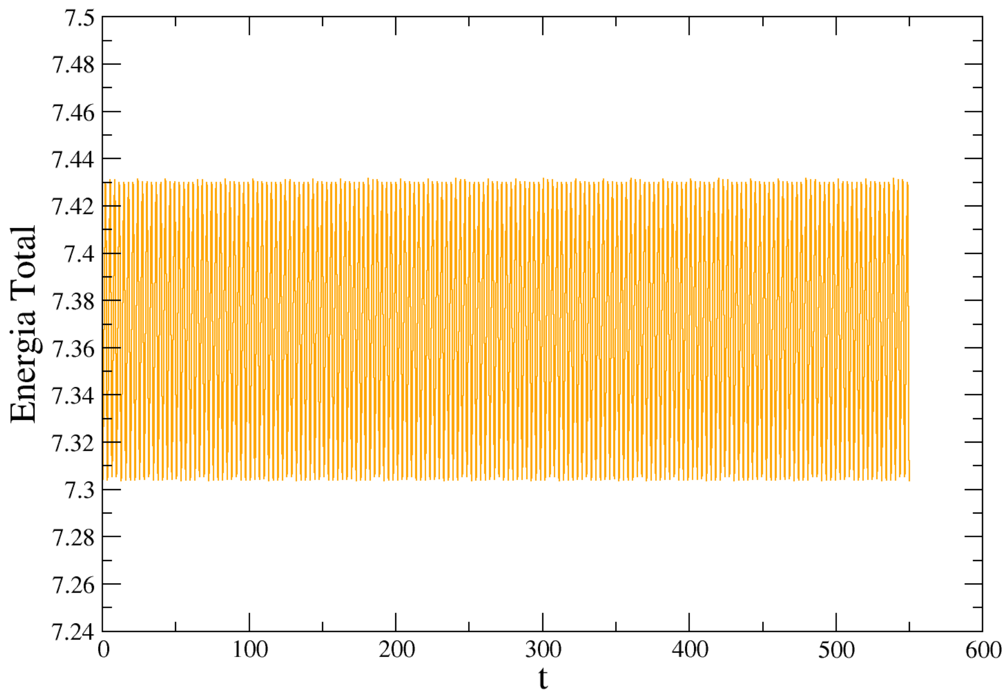
```
joao@joao-Inspiron-15-7000-Gaming: ~/Área de Trabalho/Intro a Fiscomp/Quarto P...
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
o Projeto$ cd tarefaA
(base) joao@joao-Inspiron-15-7000-Gaming:~/Área de Trabalho/Intro a Fiscomp/Quar
o Projeto/tarefaA$ f77 A.f -o A
(base) joao@joao-Inspiron-15-7000-Gaming:~/Área de Trabalho/Intro a Fiscomp/Quar
o Projeto/tarefaA$ ./A
(base) joao@joao-Inspiron-15-7000-Gaming:~/Área de Trabalho/Intro a Fiscomp/Quar
o Projeto/tarefaA$ 
```

Rende os arquivos de saída, que graficados:

Evolução de θ de um pendulo simples - método de euler



Evolução da energia de um pendulo simples - método de Euler



Mostram o resultado esperado, e correto, para um pêndulo simples, demonstrando que o método de Euler-Cromer é adequado a esta simulação.

2. Tarefa B

Para as próximas subtarefas, consideremos um pêndulo sob a ação de uma força dissipativa, e uma força externa oscilatória. A equação que representa esse sistema pode ser escrita como:

$$\frac{d\omega}{dt} = \frac{-g}{l}sin(\theta) - \gamma\frac{d\theta}{dt} + F_0 sin(\Omega t) \tag{5}$$

Que, traduzindo para o método Euler-Cromer, fica:

$$\omega_{i+1} = w_i + (\frac{-g}{l}sin(\theta) \ - \gamma w_i + F_0 sin(\Omega t))\Delta t \tag{6}$$

$$\theta_{i+1} = \theta_i + w_{i+1}\Delta t \tag{7}$$

A seguir, são testadas algumas condições iniciais para essa implementação.

2.1 B1 Pêndulo Simples

O primeiro caso a ser analisado é o de $\gamma = F_0 = 0$, ou seja, o de um pêndulo simples, para verificar se o comportamento se mantém, para uma série de θs iniciais. Desta vez, não analisamos a energia, mas sim seu período, calculado como função de θ_0, verificando toda vez que o período é completado e armazenando a diferença de tempo entre o atual e o último ciclo. Feito o cálculo explícito, é feita uma comparação com o seu cálculo feito por meio de uma integral elíptica:

$$T = \sqrt{\frac{2l}{g}} \int_{-\theta_0}^{+\theta_0} \frac{d\theta}{\sqrt{cos\theta - cos\theta_0}} \tag{9}$$

Para o cálculo da integral acima, usaremos o método de Simpson para cálculo numérico de integrais, com um cuidado especial. Note que o cálculo numérico da função da integral acima possui duas singularidades nesse intervalo: $-\ \theta_0$ e $+\ \theta_0$. Para evitar esses pontos problemáticos, calculamos a integral em um intervalo diferente, ao custo de um erro no cálculo final:

$$T = \sqrt{\frac{2l}{g}} \int_{-\theta_0+\epsilon}^{+\theta_0-\epsilon} \frac{d\theta}{\sqrt{cos\theta - cos\theta_0}} \tag{10}$$

Sendo ϵ um valor pequeno, apenas para escapar dos extremos problemáticos

O programa que implementa esse algoritmo, para $l = 9,8\,m$, $g = 9,8\,m/s^2$ e uma distribuição de θs iniciais, é:

```
double precision function f(theta, theta0, rl)
      implicit double precision (a-h, o-z)

      g = 980665d-5

      f = dsqrt(2d0*rl/g)*(1d0/dsqrt(dcos(theta) - dcos(theta0)))

      return

  end function f



  double precision function cf(theta0, rl)
      implicit double precision (a-h, o-z)

      !regra de simpson
      iseg = 1536
      rint = 0d0
      E = 5d-6
      a = -t*heta0 + E !adaptação para evitar os extremos
      b = +theta0 - E
      dh = (b - a)/(2d0*iseg)

      do i = 1, iseg

             rint = rint + (dh/3d0)*(f(a, theta0, rl)
      & + 4d0*f(a + dh, theta0, rl) + f(a + 2d0*dh, theta0, rl))
             a = a + 2d0*dh

      end do

      cf = rint

      return

  end function cf




  program main
      implicit double precision (a-h, o-z)
```

```
      double precision thetas(4)



      g = 98d-1
      pi = 4*datan(1d0)


      !B1
      rl = 98d-1
      m = 1d0
      gamma = 0d0
      omega = 0d0
      f0 = 0d0

      thetas(1) = pi/12d0
      thetas(2) = pi/16d0
      thetas(3) = pi/20d0
      thetas(4) = pi/24d0



      do j = 1, 4

             theta0 = thetas(j)
             theta  = theta0
             w = 0d0
             e = 0d0

             dt = 5d-2


             T = 0d0
             Taux = 0d0
             rcounter = 0d0


             do i = 1, 1000

                    w = w + (-(g/rl)*dsin(theta)-gamma*w
      &+f0*dsin(i*dt*omega))*dt
                    theta = theta + w*dt


                    !Calculando o período com base no theta0
                    if (dabs(theta - theta0) .LT. 1d-4 .AND.
      & i*dt - Taux .GT. 2*dt) then
                    !Se theta for quase theta0, completou-se um ciclo

                    T = T + (i*dt - Taux)
```

```fortran
                    Taux = i*dt
                    rcounter = rcounter + 1d0

                    end if


             end do

             T = T/rcounter

             Tc = cf(theta0, rl)
             write(*,*) "THETA =", theta0

             write(*,*) "Período explícito = ", T
             write(*,*) "Período integral  = ", Tc
             write(*,*)"Diferença entre períodos = ", dabs(T - Tc)
             write(*,*) " "
             write(*,*) " "
             write(*,*) " "
      end do

end program main
```

Que ao ser executado, rende:

```
joao@joao-Inspiron-15-7000-Gaming: ~/Área de Trabalho/Intro a Fiscomp/Quarto Projeto/tarefaB
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
(base) joao@joao-Inspiron-15-7000-Gaming:~/Área de Trabalho/Intro a Fiscomp/Quarto Projeto/tare
aB$ f77 B.f -o B
(base) joao@joao-Inspiron-15-7000-Gaming:~/Área de Trabalho/Intro a Fiscomp/Quarto Projeto/tare
aB$ ./B
 THETA =  0.26179938779914941
 Período explícito =    6.3071428571428578
 Período integral  =    6.3549308586150008
 Diferença entre períodos =    4.7788001472143016E-002



 THETA =  0.19634954084936207
 Período explícito =    6.2928571428571436
 Período integral  =    6.3231118013226535
 Diferença entre períodos =    3.0254658465509898E-002



 THETA =  0.15707963267948966
 Período explícito =    6.2857142857142856
 Período integral  =    6.3037715027389414
 Diferença entre períodos =    1.8057217024655792E-002



 THETA =  0.13089969389957470
 Período explícito =    6.2857142857142856
 Período integral  =    6.2903379581795500
 Diferença entre períodos =    4.6236724652644057E-003
```

2.2 B2

Nesta segunda parte, realizaremos o mesmo cálculo do período para um θ pequeno, porém ao invés compararmos com o cálculo por meio da integral elíptica, compararemos com a seguinte fórmula para θ pequenos, realizada por Bernoulli em 1749[1]:

$$T = 2\pi\sqrt{\frac{l}{g}(1+\frac{\theta_0^2}{16})} \tag{11}$$

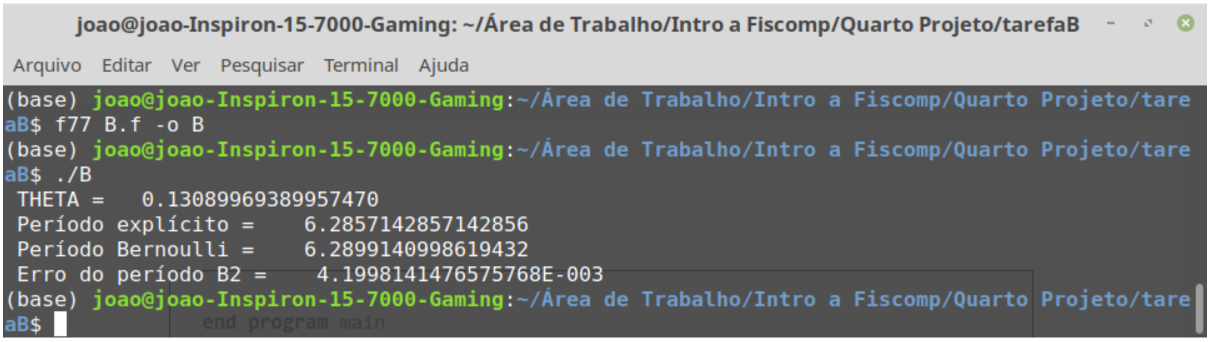
Para $\theta = \pi/24$ e os mesmos l e g usados anteriormente, o programa que implementa este algoritmo é:

```
program main
       implicit double precision (a-h, o-z)
       !B2
       rl = 98d-1
       m = 1d0
       gamma = 0d0
       omega = 0d0
       f0 = 0d0

       theta0 = pi/24d0
       theta  = theta0
       w = 0d0

       dt = 5d-2

       T = 0d0
       Taux = 0d0
       rcounter = 0d0


       do i = 1, 1000

              w = w + (-(g/rl)*dsin(theta)-gamma*w
       & + f0*dsin(i*dt*omega))*dt
              theta = theta + w*dt

       !Calculando o período com base no theta0
              if (dabs(theta - theta0) .LT. 1d-4 .AND.
       & i*dt - Taux .GT. 2*dt) then
                     !Se theta for quase theta0, completou-se um ciclo
```

1 https://pt.wikipedia.org/wiki/Pêndulo

```
                    T = T + (i*dt - Taux)

                    Taux = i*dt
                    rcounter = rcounter + 1d0

              end if
        end do

        T = T/rcounter

        Tc = 2d0*pi*dsqrt(rl/g)*(1d0 + (theta0**2)/16d0)


        write(*,*) "THETA = ", theta0
        write(*,*) "Período explícito = ", T
        write(*,*) "Período Bernoulli = ", Tc
        write(*,*) "Erro do período B2 = ", dabs(T - Tc)
       stop
end program main
```

Que rodando, rende:



Mostrando que, de fato, o período está sendo dado pela relação (11), caracterizando assim um estado de oscilações harmônicas, independentes de θ_0.

2.3 B3

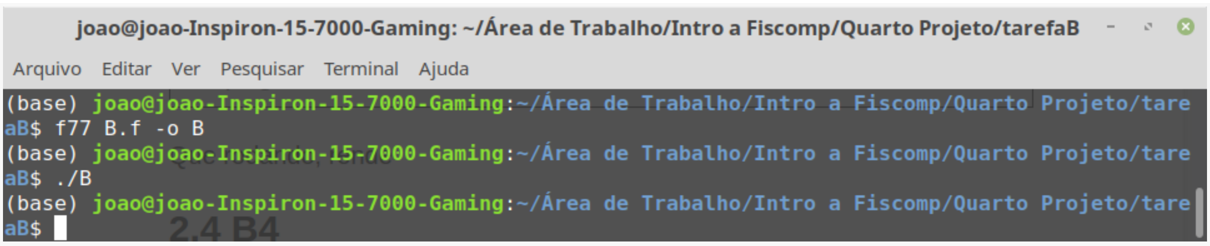
Nesta terceira parte, agora consideremos o regime de amortecimento, ou seja, $\gamma = 0.5$ na equação (5). Aqui analisaremos $\theta \times t$ a partir de um gráfico.

O programa que implementa esse algorítmo é:

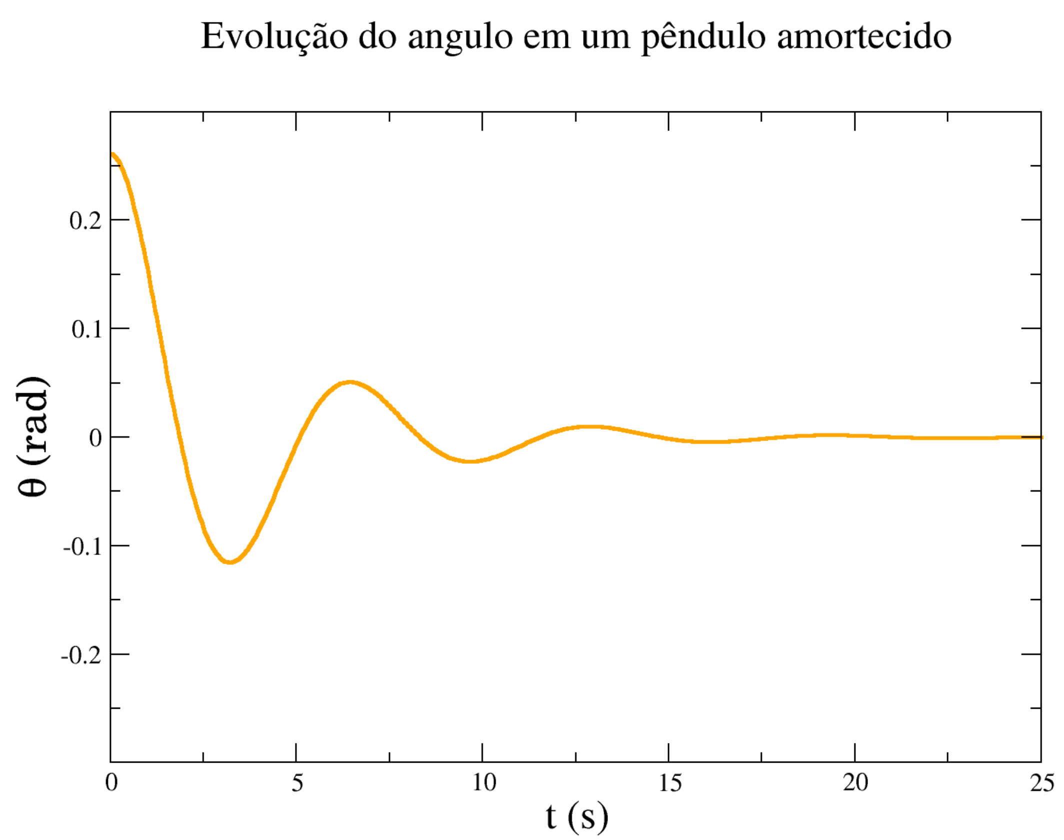
```
program main
```

```fortran
        implicit double precision (a-h, o-z)
        !B3
        rl = 98d-1
        m = 1d0
        gamma = 5d-1 !coeficiente de amortecimento
        omega = 0d0
        f0 = 0d0

        theta0 = pi/12d0
        theta  = theta0
        w = 0d0

        dt = 5d-2


        open(25, file = "saida-B-1-10799783.dat")


        do i = 1, 500

               w = w + (-(g/rl)*dsin(theta)-gamma*w
        & +f0*dsin(i*dt*omega))*dt
               theta = theta + w*dt


               write(25,*) i*dt, theta


        end do



        close(25)
         stop
end program main
```

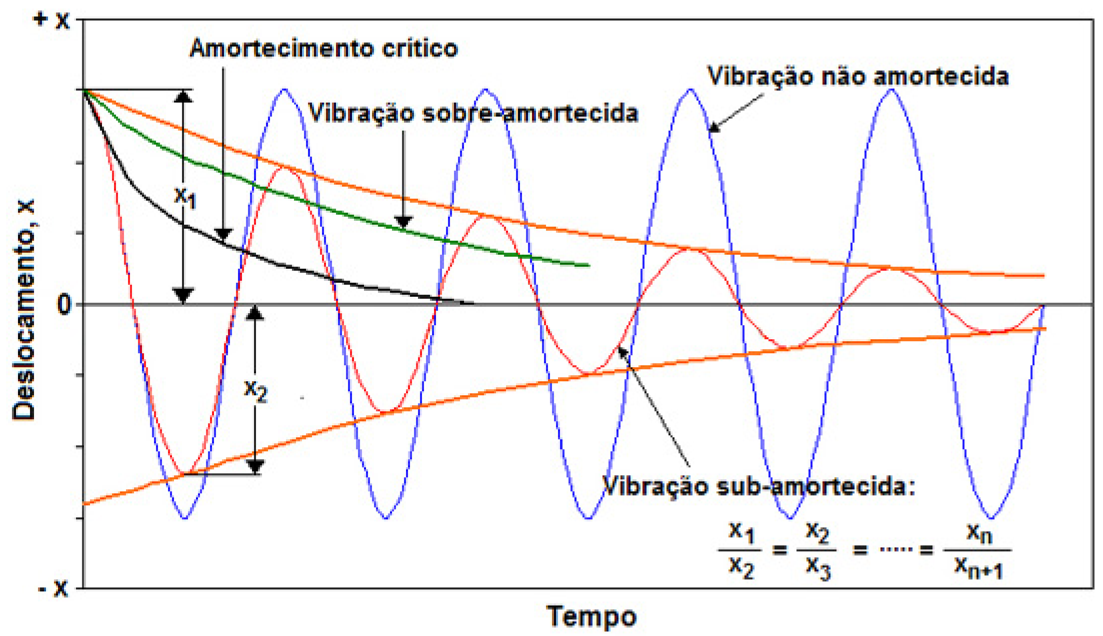
Que rodando, rende



E um grande arquivo de saída, que graficado, rende:



Que, ao ser analisado, é visualmente possível inferir que, para $\gamma = 0.5$, o sistema entra em regime de amortecimento sub-crítico, como descrito pela imagem abaixo:



Fonte: https://www.ctborracha.com/artefactos/a-borracha-no-amortecimento-de-vibracoes/

2.4 B4

Para esta quarta parte do item B, consideremos agora o caso com $\gamma = 0.5$, $\Omega = 2/3$, e $\Delta t = 0.03$, para três forças de valores $F_0 = 0.0,\ 0.5,\ 1.2$, os quais nos permitirão análise dos gráficos de $\theta(t)$ e $\omega(t)$. O programa que implementa o algoritmo é:

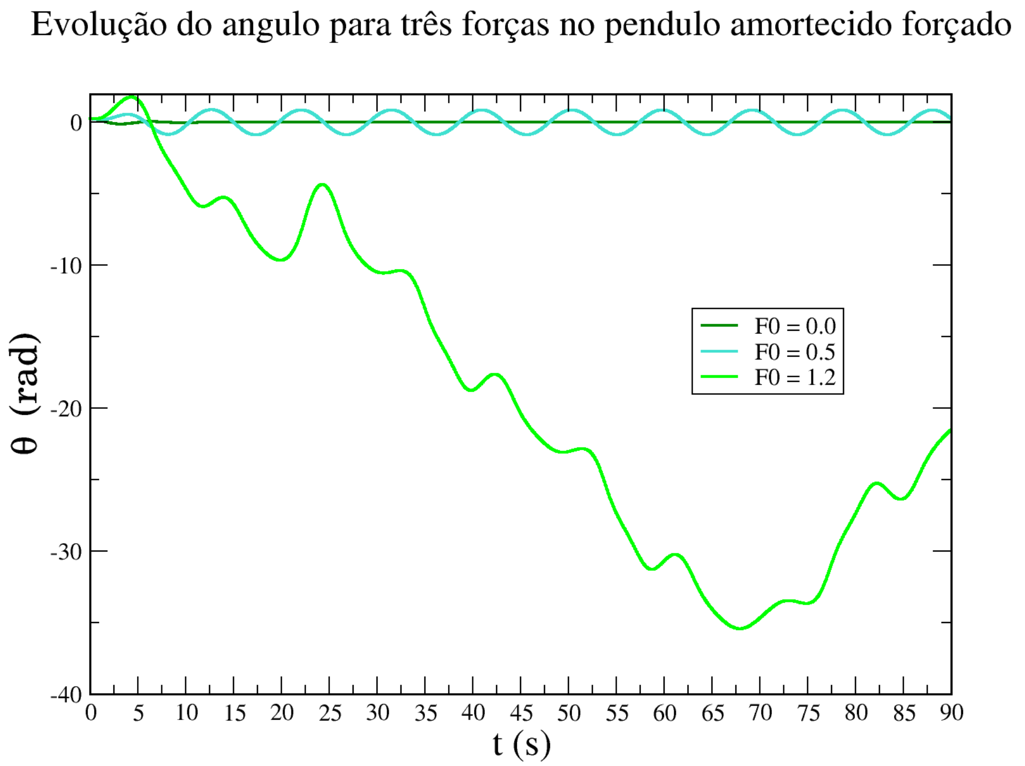
```
program main
implicit double precision (a-h, o-z)
!B4
rl = 1d0
m = 1d0
gamma = 5d-1
omega = 2d0/3d0
f01 = 0d0
f02 = 5d-1
f03 = 12d-1

theta0 = pi/12d0
theta1 = theta0
theta2 = theta0
theta3 = theta0
w1 = 0d0
w2 = 0d0
w3 = 0d0


dt = 3d-2

open(26, file = "saida-B-2-10799783.dat")
open(27, file = "saida-B-3-10799783.dat")



do i = 1, 500

       !Pendulo para f01 = 0d0
       w1 = w1 + (-(g/rl)*dsin(theta1)-gamma*w1
&+f01*dsin(i*dt*omega))*dt
       theta1 = theta1 + w1*dt

       !Pendulo para f02 = 5d-1
       w2 = w2 + (-(g/rl)*dsin(theta2)-gamma*w2
&+f02*dsin(i*dt*omega))*dt
```

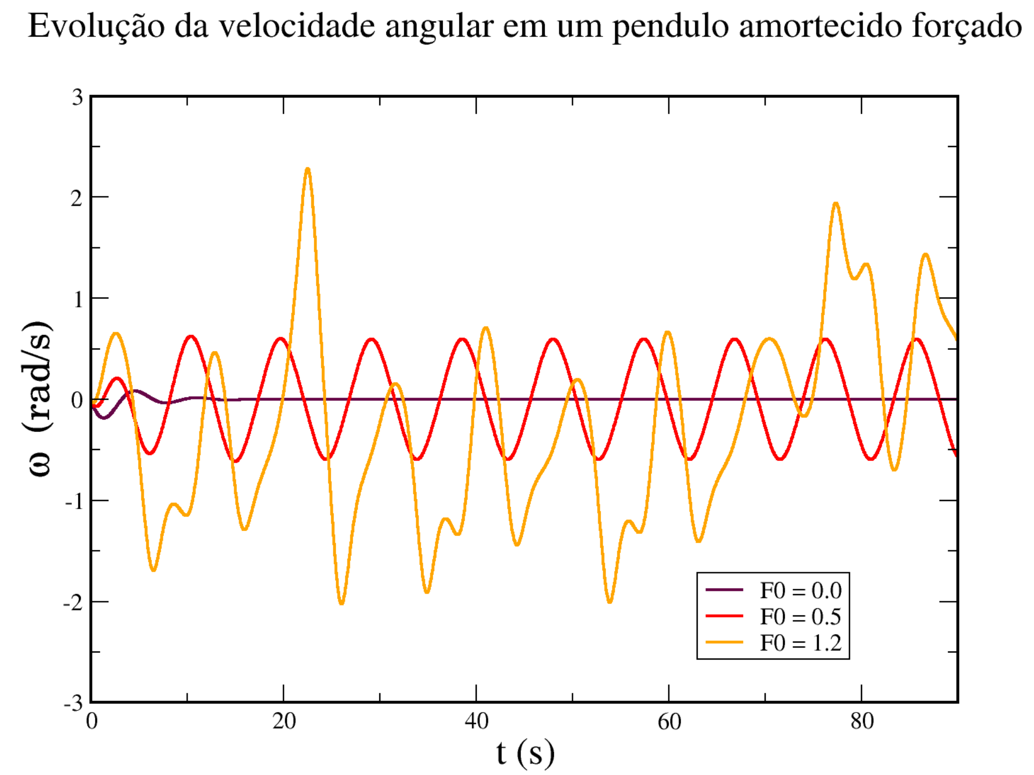
```
            theta2 = theta2 + w2*dt

            !Pendulo para f03 = 12d-1
            w3 = w3 + (-(g/rl)*dsin(theta3)-gamma*w3
      &+f03*dsin(i*dt*omega))*dt
            theta3 = theta3 + w3*dt


            write(26,*) i*dt, w1, w2, w3
            write(27,*) i*dt, theta1, theta2, theta3


      end do



      close(26)
      close(27)



      stop
      end program main
```

Que rodando, rende:

```
joao@joao-Inspiron-15-7000-Gaming: ~/Área de Trabalho/Intro a Fiscomp/Quarto Projeto/tarefaB
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
(base) joao@joao-Inspiron-15-7000-Gaming:~/Área de Trabalho/Intro a Fiscomp/Quarto Projeto/tarefaB$ f77 B.f -o B
(base) joao@joao-Inspiron-15-7000-Gaming:~/Área de Trabalho/Intro a Fiscomp/Quarto Projeto/tarefaB$ ./B
(base) joao@joao-Inspiron-15-7000-Gaming:~/Área de Trabalho/Intro a Fiscomp/Quarto Projeto/tarefaB$
```

E dois grandes arquivos de saída, que graficados, ficam:





Os dois gráficos revelam um comportamento interessante, principalmente o primeiro. É possível perceber que, desde o começo, enquanto o pêndulo sem forças externas segue seu caminho previsto de amortecimento, o movimento do pêndulo sob força 1.2 desanda de seu comportamento previsto, adquirindo um de certa forma imprevisível, perdendo totalmente sua periodicidade na janela de tempo simulada. O pêndulo sob força 0.5, diferente da força 1.2, não perde sua periodicidade, sendo responsável apenas por aumentar a amplitude do movimento natural do pêndulo. Pelo gráfico, é possível estimar sua frequência, o inverso do período, como algo em torno de 1/10.

3. Tarefa C

Nesta tarefa, adotamos os mesmos valores usados em B4 para as demais variáveis, porém agora consideramos dois pêndulos de mesmas configurações, porém com diferenças de ângulos iniciais $\Delta\theta_0 = 0.001\, rad$. A partir disso, pretende-se analisar a evolução dessa diferença com o tempo $\Delta\theta(t)$, na procura pela suposta caoticidade dos regimes sob as duas forças diferentes. Essa diferença deve ter o seguinte formato:

$$\Delta\theta(t) \approx e^{\lambda t} \tag{12}$$

Sendo λ o que chama-se de expoente de Liapunov. Espera-se, que para casos não-caóticos, como o da força = 0.5, a curva $\Delta\theta$ tenha formato modelado por uma exponencial de expoente negativo, ou seja, com $\lambda < 0$. O contrário é então esperando para casos caóticos como o da força = 1.2, tendo a curva do $\Delta\theta$ modelada a partir de uma exponencial de expoente positivo, ou seja, com $\lambda > 0$.

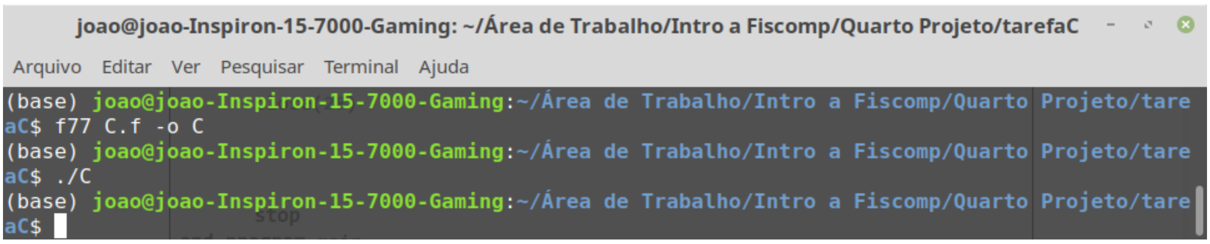
O programa que implementa este algoritmo é:

```
program main
       implicit double precision (a-h, o-z)



       g = 98d-1
       pi = 4*datan(1d0)
```

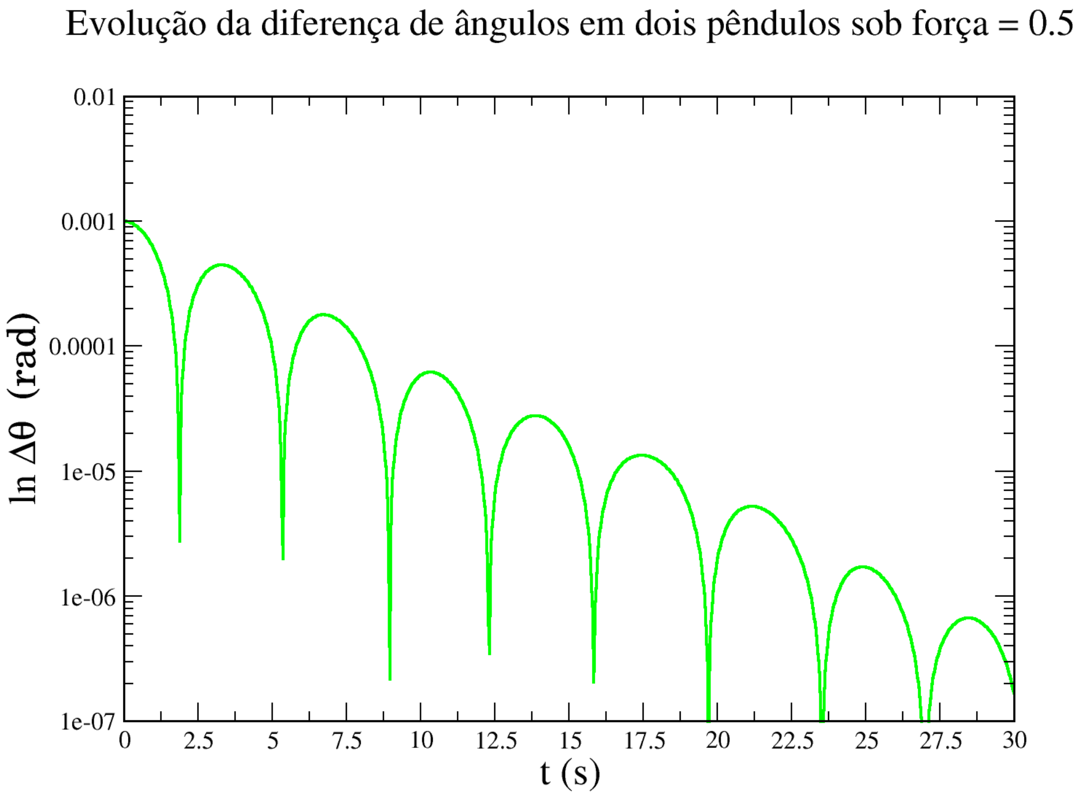
```fortran
rl = 98d-1
m = 1d0
gamma = 5d-1
omega = 2d0/3d0
f01 = 5d-1
f02 = 12d-1


theta01 = pi/12d0
theta02 = theta01 + 1d-3

!f01 = 5d-1
theta11 = theta01
theta21 = theta02

w11 = 0d0
w21 = 0d0

!f02 = 12d-1
theta12 = theta01
theta22 = theta02

w12 = 0d0
w22 = 0d0


dt = 3d-2

open(27, file = "saida-C-1-10799783.dat")
open(28, file = "saida-C-2-10799783.dat")


do i = 1, 3000

!f01 = 5d-1

      !pendulo 1
      w11 = w11 + (-(g/rl)*dsin(theta11)-gamma*w11
&+f01*dsin(i*dt*omega))*dt
      theta11 = theta11 + w11*dt

      !pendulo 2
      w21 = w21 + (-(g/rl)*dsin(theta21)-gamma*w21
&+f01*dsin(i*dt*omega))*dt
      theta21 = theta21 + w21*dt



!f02 = 12d-1
```

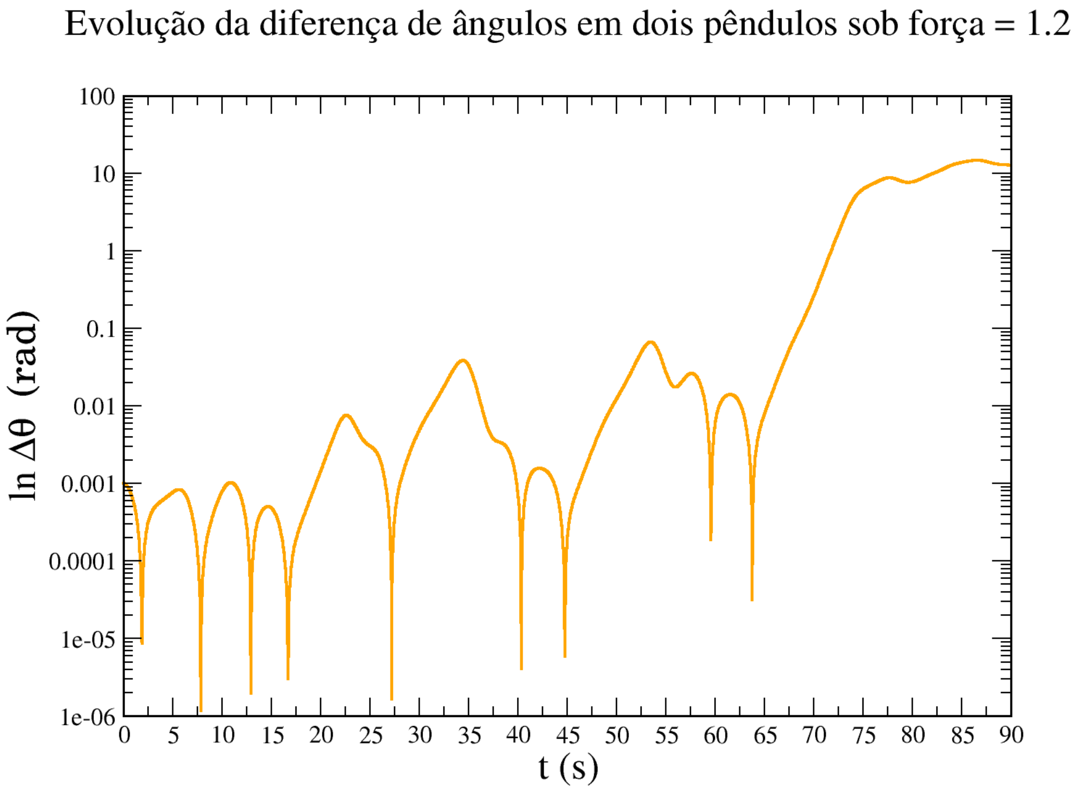
```
          !pendulo 1
          w12 = w12 + (-(g/rl)*dsin(theta12)-gamma*w12
     &+f02*dsin(i*dt*omega))*dt
          theta12 = theta12 + w12*dt

          !pendulo 2
          w22 = w22 + (-(g/rl)*dsin(theta22)-gamma*w22
     &+f02*dsin(i*dt*omega))*dt
          theta22 = theta22 + w22*dt



          deltat1 = dabs(theta21 - theta11)
          deltat2 = dabs(theta22 - theta12)


          write(27,*) i*dt, deltat1
          write(28,*) i*dt, deltat2


      end do



      close(27)
      close(28)



      stop
end program main
```

Que rodando, rende:



Tendo como arquivos de saída os dados para os gráficos dos casos das duas forças. Graficando-os em escala semi-log, temos:





Pelos gráficos, é possível verificar com sucesso o comportamento descrito anteriormente, com o caso da força = 0.5 obedecendo o regime não caótico, e a força = 1.2 ocorrendo sob um regime caótico. É possível também estimar os respectivos expoentes contabilizando os picos, afinal o seguinte gráfico em semi-log deve ter o formato:

$$ln(\Delta\theta) \approx \lambda t \tag{13}$$

Pelo gráfico do regime não-caótico, pegando o pico inicial em $(0,\ ln\, 10^{-3})$ e o pico final em $(28.75,\ ln\, 10^{-6})$, temos:

$$\lambda \approx \frac{ln10^{-6} - ln10^{-3}}{28.75 - 0.0} \approx -0.24 \tag{14}$$

É possível também, apesar de uma aproximação grosseira, estimar o expoente no caso caótico. Pegando o pico inicial também em $(0,\ ln\, 10^{-3})$ e o pico final em $(85,\ ln\, 10)$, temos:

$$\lambda \approx \frac{ln10 - ln10^{-3}}{85 - 0.0} \approx 0.108 \tag{15}$$

4. Tarefa D

Nesta tarefa, o objetivo foi de, a partir dos mesmos coeficientes anteriores, fazer gráficos de $\omega(\theta)$, o espaço de fase, para algumas condições iniciais próximas, sob o regime das forças 0.5 e 1.2.

O programa que implementa o algoritmo é:

```
program main
       implicit double precision (a-h, o-z)
       double precision thetas(4)




       g = 98d-1
       pi = 4*datan(1d0)


       rl = 98d-1
       m = 1d0
       gamma = 5d-1
```

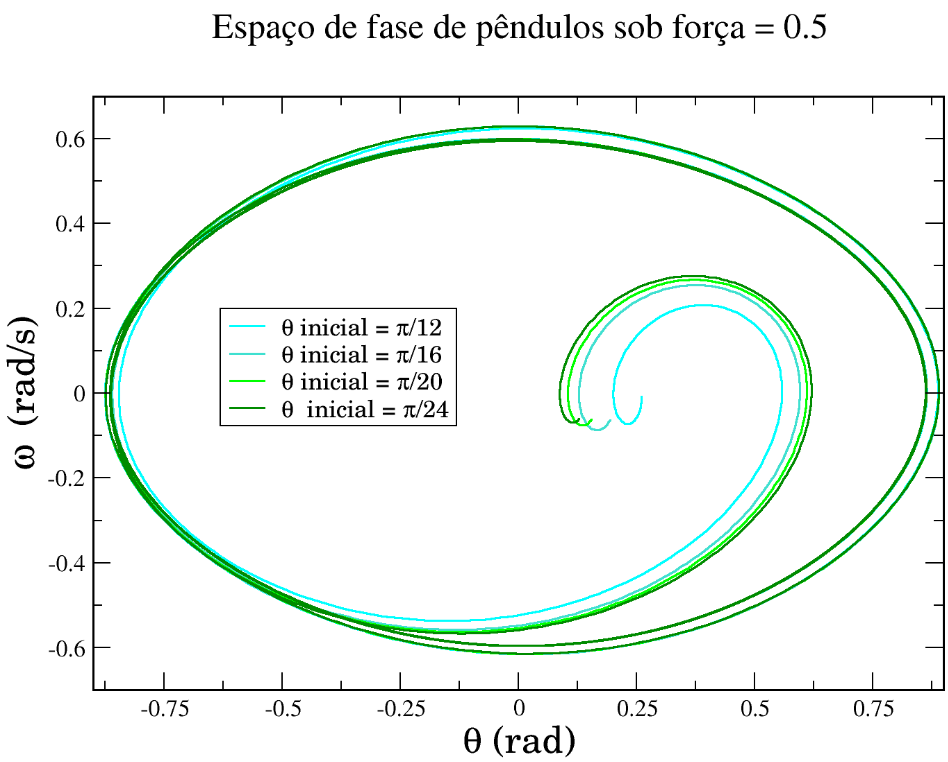
```fortran
    omega = 2d0/3d0
    f01 = 5d-1
    f02 = 12d-1


    thetas(1) = pi/12d0
    thetas(2) = pi/16d0
    thetas(3) = pi/20d0
    thetas(4) = pi/24d0

    !pendulo 1


    w11 = 0d0
    w21 = 0d0

    !pendulo 2


    w12 = 0d0
    w22 = 0d0



    dt = 3d-2


    open(28, file = "saida-D-1-10799783.dat")
    open(29, file = "saida-D-2-10799783.dat")
    open(30, file = "saida-D-3-10799783.dat")
    open(31, file = "saida-D-4-10799783.dat")

    open(32, file = "saida-D-5-10799783.dat")
    open(33, file = "saida-D-6-10799783.dat")
    open(34, file = "saida-D-7-10799783.dat")
    open(35, file = "saida-D-8-10799783.dat")





    do j = 0, 3


        !determinando thetas iniciais próximos
        theta11 = thetas(j + 1)
        theta12 = thetas(j + 1)
```

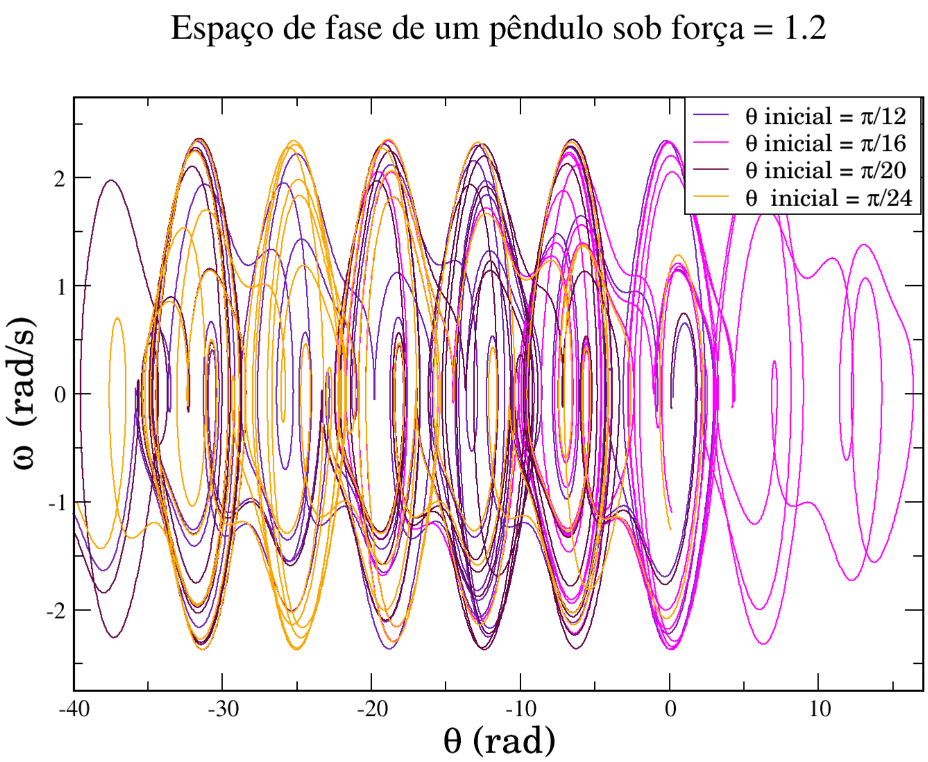
```
             do i = 1, 1000

                   w11 = w11 + (-(g/rl)*dsin(theta11)-gamma*w11
       &+f01*dsin(i*dt*omega))*dt
                   theta11 = theta11 + w11*dt



                   w12 = w12 + (-(g/rl)*dsin(theta12)-gamma*w12
       &+f02*dsin(i*dt*omega))*dt
                   theta12 = theta12 + w12*dt


                   n = 28 + j
                   m = 32 + j

                   write(n,*) theta11, w11
                   write(m,*) theta12, w12


             end do

       end do


       close(28)
       close(29)
       close(30)
       close(31)
       close(32)
       close(33)
       close(34)
       close(35)



       stop
end program main
```

Rodando, temos:

```
joao@joao-Inspiron-15-7000-Gaming: ~/Área de Trabalho/Intro a Fiscomp/Quarto Projeto/tarefaD
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
(base) joao@joao-Inspiron-15-7000-Gaming:~/Área de Trabalho/Intro a Fiscomp/Quarto Projeto/tare
aD$ f77 D.f -o D
(base) joao@joao-Inspiron-15-7000-Gaming:~/Área de Trabalho/Intro a Fiscomp/Quarto Projeto/tare
aD$ ./D
(base) joao@joao-Inspiron-15-7000-Gaming:~/Área de Trabalho/Intro a Fiscomp/Quarto Projeto/tare
aD$
```

Que rende uma série de arquivos que nos possibilitam graficar os espaços de fase para as condições impostas. Graficando-os, temos:





Percebe-se, comparando as duas situações de forças diferentes, como o movimento caótico se comporta em relação ao não-caótico. Enquanto no movimento não-caótico obtivemos uma região elipsoidal uniforme, no movimento caótico obtivemos uma figura mais complexa, mas que, por incrível que pareça, possui também uma região bem definida, porém mais extensa. É possível concluir, por exemplo, que o pêndulo caótico nunca assumirá o par posição-velocidade angular = $(10, \ -2.5)$, o que parece ir contra a ideia de "caos" anteriormente assumida para este caso.

5. Tarefa E

Nesta tarefa, o objetivo foi de, a partir dos conceitos apresentados no item anterior, visualizar o espaço de fase do pêndulo sob as duas forças, porém sob condições diferentes. Aqui, analisamos o espaço de fase em locais específicos, no caso, quando $\Omega t = n\pi$, locais conhecidos como Secções de Poincaré. Para fazer essa análise, alterou-se o código anterior para apenas imprimir a coordenada no arquivo de saída quando a seguinte condição fosse atingida:

$$|t - \frac{n\pi}{\Omega}| < \frac{\Delta t}{2} \tag{16}$$

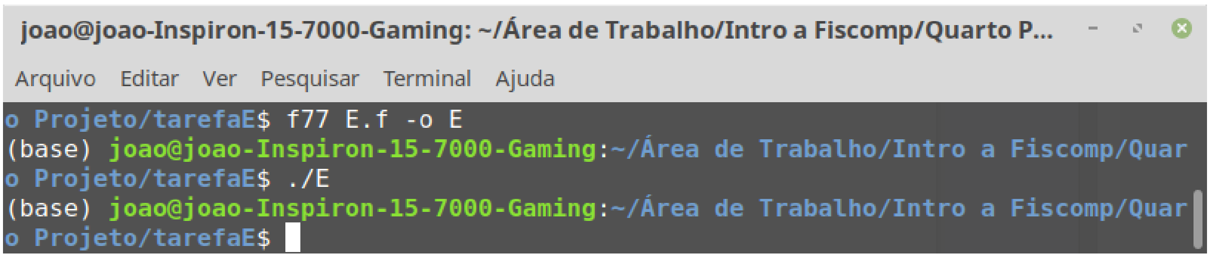
O código que implementa esse algoritmo é:

```
program main
      implicit double precision (a-h, o-z)
      double precision thetas(4)




      g = 98d-1
      pi = 4*datan(1d0)


      rl = 98d-1
      m = 1d0
      gamma = 5d-1
      omega = 2d0/3d0
      f01 = 5d-1
      f02 = 12d-1



      thetas(1) = pi/60d-1
      thetas(2) = pi/61d-1
      thetas(3) = pi/62d-1
```

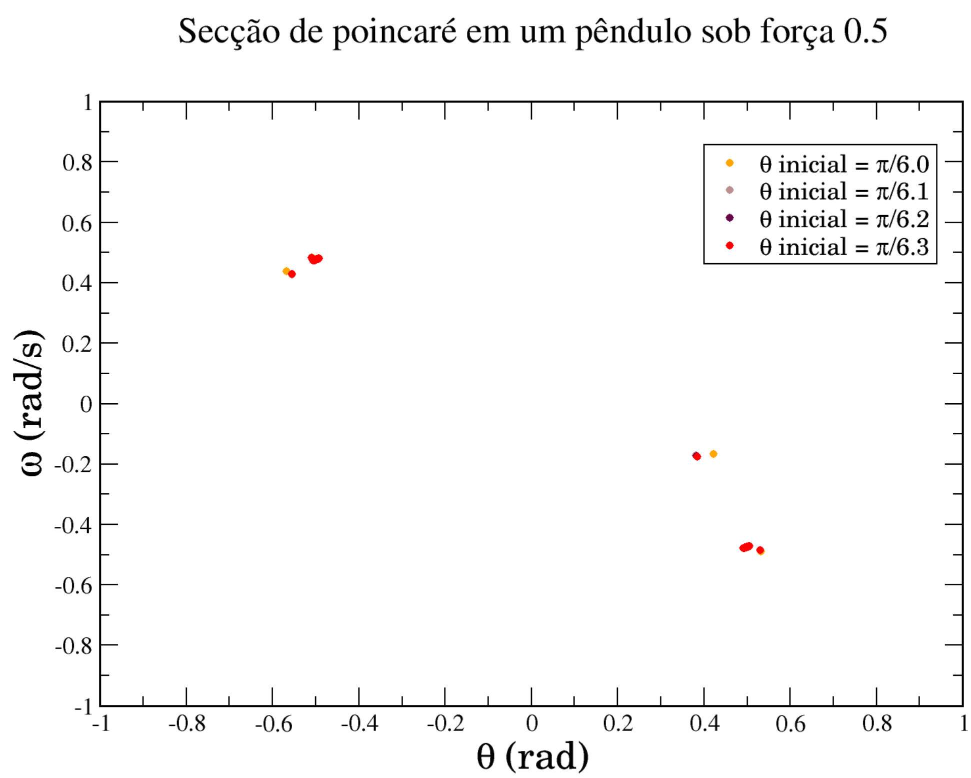
```
thetas(4) = pi/63d-1



!pendulo 1

w11 = 0d0
w21 = 0d0

!pendulo 2



w12 = 0d0
w22 = 0d0



dt = (3d-2)


open(28, file = "saida-E-1-10799783.dat")
open(29, file = "saida-E-2-10799783.dat")
open(30, file = "saida-E-3-10799783.dat")
open(31, file = "saida-E-4-10799783.dat")
open(32, file = "saida-E-5-10799783.dat")
open(33, file = "saida-E-6-10799783.dat")
open(34, file = "saida-E-7-10799783.dat")
open(35, file = "saida-E-8-10799783.dat")




do j = 0, 3

       n = 1

       theta11 = thetas(j + 1)
       theta12 = thetas(j + 1)


       do i = 1, 100000


              w11 = w11 + (-(g/rl)*dsin(theta11)-gamma*w11
&+f01*dsin(i*dt*omega))*dt
              theta11 = theta11 + w11*dt
```

```
                  w12 = w12 + (-(g/rl)*dsin(theta12)-gamma*w12
      &+f02*dsin(i*dt*omega))*dt
                  theta12 = theta12 + w12*dt




                  if ( dabs(i*dt- n*pi/omega) .LT. dt/2d0) then
                  write(28 + j,*) theta11, w11
                  write(32 + j,*) theta12, w12
                  n = n + 1
                  end if

            end do



      end do



      close(28)
      close(29)
      close(30)
      close(31)
      close(32)
      close(33)
      close(34)
      close(35)



      stop
end program main
```

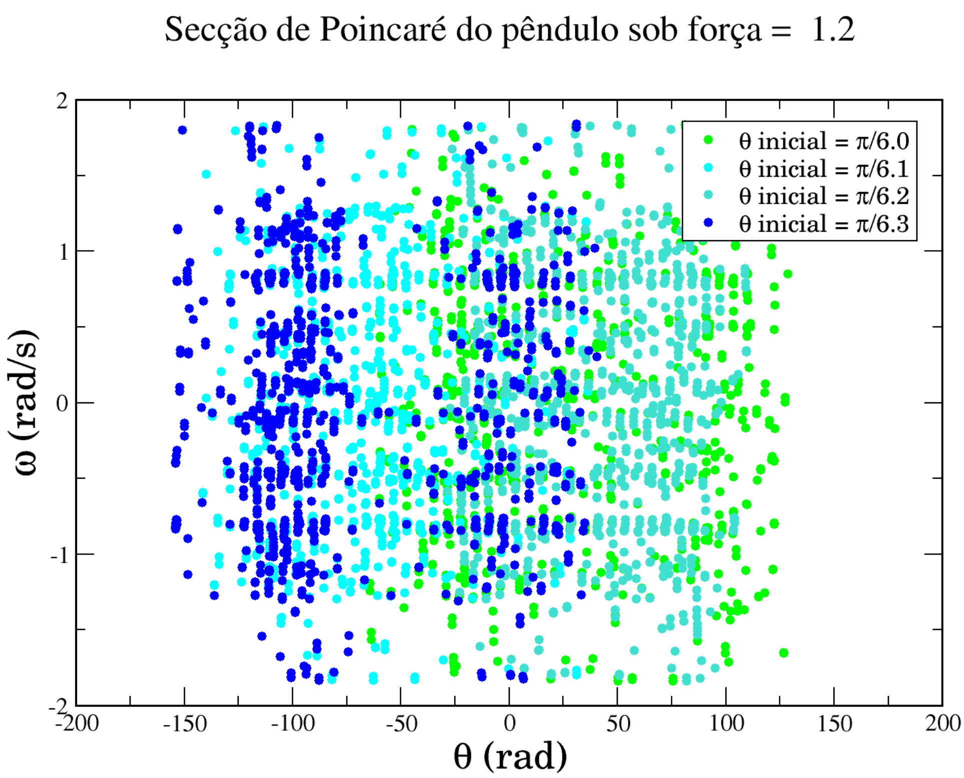
Que rodando, rende:



E também oito arquivos de saída, sendo os quatro primeiros relativos ao caso da força 0.5 para condições iniciais próximas, e os quatro últimos o caso da força 1.2. Graficando-os, temos:

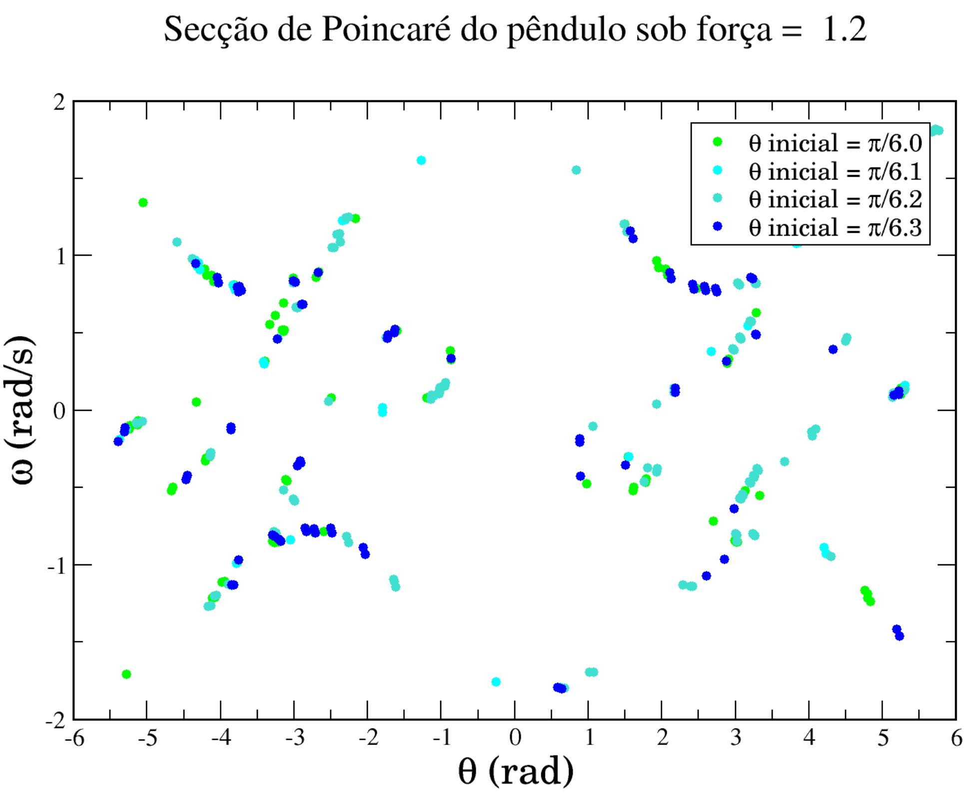


Secção de poincaré em um pêndulo sob força 0.5
θ inicial = π/6.0
θ inicial = π/6.1
θ inicial = π/6.2
θ inicial = π/6.3
ω (rad/s)
θ (rad)



Secção de Poincaré do pêndulo sob força = 1.2
θ inicial = π/6.0
θ inicial = π/6.1
θ inicial = π/6.2
θ inicial = π/6.3
ω (rad/s)
θ (rad)

Para melhor análise, aproximemos o gráfico 2 para um segmento menor:



Que são, por si só, figuras interessantes, e mostram que mesmo para condições iniciais diferentes, o comportamento geral do caos é, por mais paradoxal que possa soar, caoticamente previsível.