UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE FÍSICA DE SÃO CARLOS

Introdução à Física Computacional - Projeto 4

Silvio Lacerda de Almeida

Novembro, 2023
São Carlos, SP

Sumário

Lista de Figuras

1 Introdução

O segundo projeto de Introdução à Física Computacional contempla 5 tarefas, e delas explicarei detalhadamente o pensamento que originou o código, assim como discutirei o resultado dos mesmos.

Todos os códigos foram escritos em Fortran 77, aderindo à sintaxe implícita, ou seja, quaisquer variáveis declaradas tendo inicialmente letras no intervalo [i-n] são inteiras, e o resto são reais de precisão dupla. A declaração de variáveis entretanto se fez obrigatória quando foi necessário criar funções e instanciar as suas variáveis em seus respectivos escopos.

2 Tarefas

2.1 Tarefa 1

Figura 1: Tarefa 1

```
      program EDOAngulo
      implicit real*8 (a-h,o-z)

      parameter(ite=2000)
      parameter(dt=1e-2)
      parameter(g=9.8d0)
      parameter(al=9.8d0)
      parameter(m=1.0d0)

      dimension t(ite), w(ite), e(ite), theta(ite)
      dimension w_ec(ite), e_ec(ite), theta_ec(ite)
      theta0 = 0.14

      open(1, file="saida-theta-t.dat")
      open(2, file="saida-theta-t_ec.dat")
      open(3, file="saida-e-t.dat")
      open(4, file="saida-e-t_ec.dat")

      PI = 4.0d0*atan(1.0d0)
      theta(1) = theta0
      theta_ec(1) = theta0
      w(1) = 0.0d0
      w_ec(1) = 0.0d0
      e(1) = m*g*al*(1-cos(theta0))
      e_ec(1) = m*g*al*(1-cos(theta0))
      do i=1,ite-1,1
         t(i+1) = i*dt
         w(i+1) = w(i) - (g/al)*theta(i)*dt
         w_ec(i+1) = w_ec(i) - (g/al)*theta_ec(i)*dt

         theta(i+1) = mod(theta(i) + w(i)*dt, 2.0d0*pi)
         theta_ec(i+1) = mod(theta_ec(i) + w_ec(i+1)*dt, 2.0d0*pi)

         e(i+1) = m*g*al*(1-cos(theta(i+1))) + (m/2)*(w(i+1)*al)**2
         e_ec(i+1) = m*g*al*(1-cos(theta_ec(i+1))) +
     & (m/2)*(w_ec(i+1)*al)**2
            write(1,*) t(i+1), theta(i+1)
            write(2,*) t(i+1), theta_ec(i+1)
            write(3,*) t(i+1), e(i+1)
            write(4,*) t(i+1), e_ec(i+1)
        end do
        close(1)
        close(2)
        close(3)
        close(4)
        end program
```

2.1.1 Explicando a tarefa 1

No código eu começo definindo as variáveis que vou usar, assim como os parâmetros. Também são criados os vetores para armazenar tempo, ω da oscilação, energia e θ. Como também irei calcular estes valores para o método de Euler, então crio simetricamente estes vetores.

Logo em seguida defino os valores iniciais para θ e energia, para posteriormente iniciar o laço. No laço eu aplico diretamente as equações dadas no trabalho.

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

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

E com as correções de Euler temos:

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

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

2.1.2 Resultados

Dadas as condições iniciais sendo $\theta_0 = 0.14$ radianos, $\omega_0 = 0$ rad/s, é possível construir os seguintes gráficos.

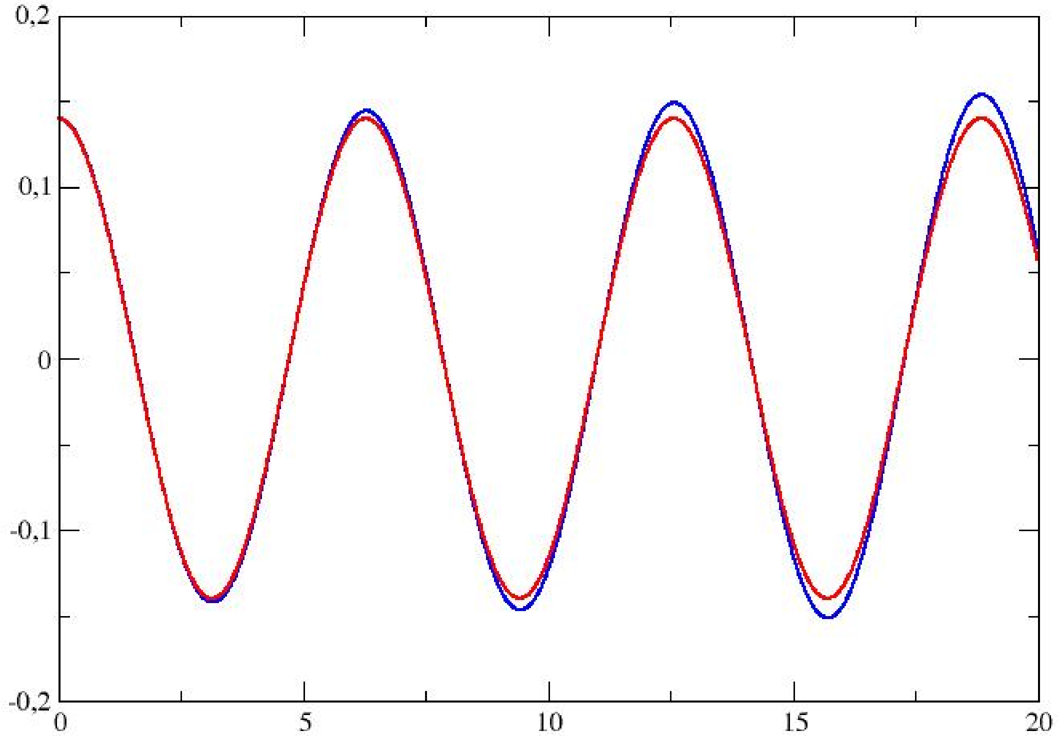

Figura 2: Gráfico mostrando $\theta(t)$. Em azul está o calculado sem as correções, e em vermelho com as correções de Euler.

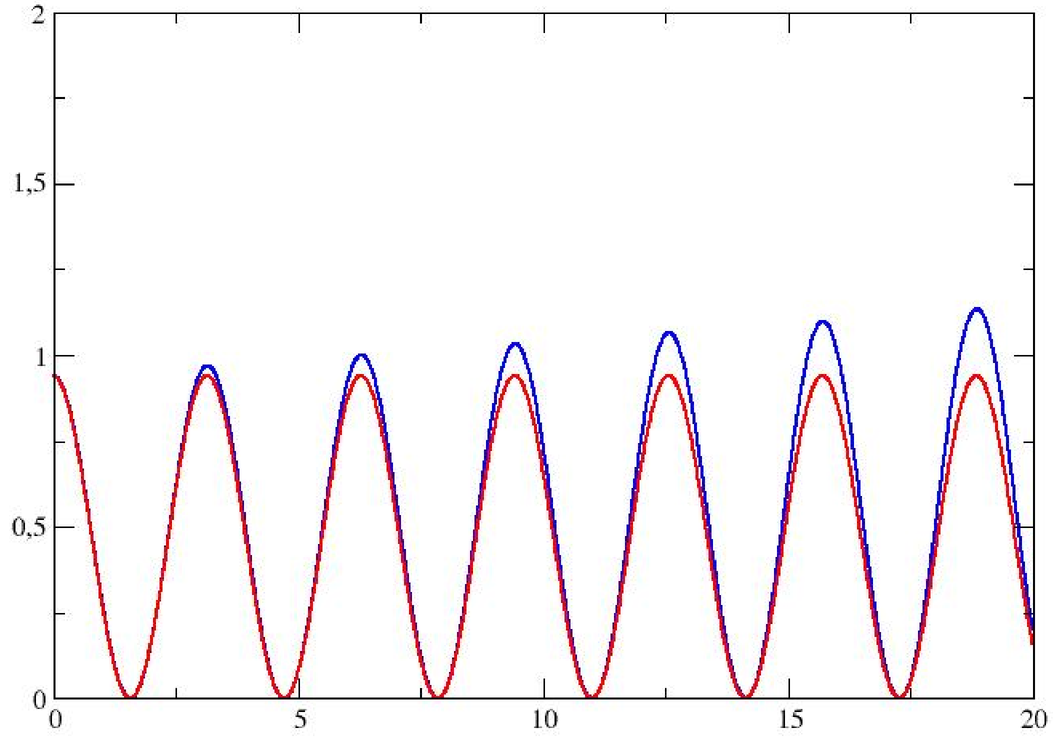

Figura 3: Gráfico mostrando $E(t)$. Em azul está o calculado sem as correções, e em vermelho com as correções de Euler.

2.2 Tarefa 2

Para fazer esta tarefa, dividi ela em duas partes. A primeira parte é responsável por responder as perguntas **B1** e **B2** e a segunda parte é responsável por responder as perguntas **B3** e **B4**.

Figura 4: Tarefa 2 - Arquivo 1

```
      program PeriodoPenduloAmortecido
      implicit real*8 (a-h,o-z)
      parameter(dt=1e-2)
      parameter(g=9.8d0)
      parameter(al=9.8d0)
      parameter(n=5000)
      parameter(gama=0.05d0)
      parameter(omega=2d0/3d0)
      character*50 fnametheta
      character*50 fnamew
      dimension vec_f(3)

      vec_f = [0d0,0.5d0,1.2d0]

      do j=1,3,1
      pi = 4.0d0*atan(1.0d0)

      wi = 0.0d0
      w = 0.0d0
      thetai = pi/3d0
      theta = 0.0d0
      t = 0
      f0 = vec_f(j)

      write(fnametheta, 100) j
100  format('saida-2-13783203-theta-',I0,'.dat')

      write(fnamew, 101) j
101   format('saida-2-13783203-w-',I0,'.dat')

      open(j, file=fnametheta)
      open(j+10, file=fnamew)

      do i=1,n,1
          t=i*dt
          w=wi-(g/al)*sin(theta)*dt-gama*w*dt+f0*sin(omega*t)*dt
          theta=mod(thetai+w*dt, 2.0d0*pi)
          write(j,*) t, theta
          write(j+10,*) t, w

          wi=w
          thetai=theta
      end do

      close(j)
      close(j+10)
      end do
      end program
```

2.2.1 Explicando a tarefa 2 - Arquivo 1

Inicialmente defino os parâmetros iniciais que usarei durante todo o código, e logo depois vou colocar no terminal o resultado dos períodos calculados.

A função *f(x)* é justamente a integral que para resolver é necessário utilizar um ϵ que permita que as contas não terminem em uma singularidade (integral elíptica). A função *periodo-eliptica* serve justamente para fazer este cálculo, onde o método de integração escolhido foi o método de Simpson. A função período aproximado utiliza a seguinte aproximação:

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

2.2.2 Explicando a tarefa 2 - Arquivo 2

Nesta segunda parte, reutilizamos a lógica para calcular θ e ω, e assim podemos escrever ambos em função de t, com o adicional de que agora os termos γ, F_0 e Ω foram adicionados na equação, demonstrando amortecimento e forças externas.

2.2.3 Resultados

Para responder as perguntas **B1** e **B2** utilizaremos o arquivo 1, e testando valores diferentes de θ_0 veremos que quanto menor for o valor, mais próximos os cálculos estão. Isso se deve, claro, pois todas as premissas para calcular que utilizamos partia justamente de que θ oscila entre valores pequenos.

Figura 5: $\theta_0 = \frac{\pi}{3}$ rad

```
Periodo integral:    6.7427075722809224
Periodo aproximado:    6.7138280388504166
Periodo analitico:    6.2831853071795862
```

Figura 6: $\theta_0 = \frac{\pi}{10}$ rad

```
Periodo integral:    6.3195338648152930
Periodo aproximado:    6.3219431530299612
Periodo analitico:    6.2831853071795862
```

Figura 7: $\theta_0 = \frac{\pi}{50}$ rad

```
Periodo integral:    6.2537074985718366
Periodo aproximado:    6.2847356210136009
Periodo analitico:    6.2831853071795862
```

Para responder as tarefas **B3** e **B4** podemos analisar as respostas apenas de **B4**, uma vez que o gráfico de **B3** está contido nestas respostas.

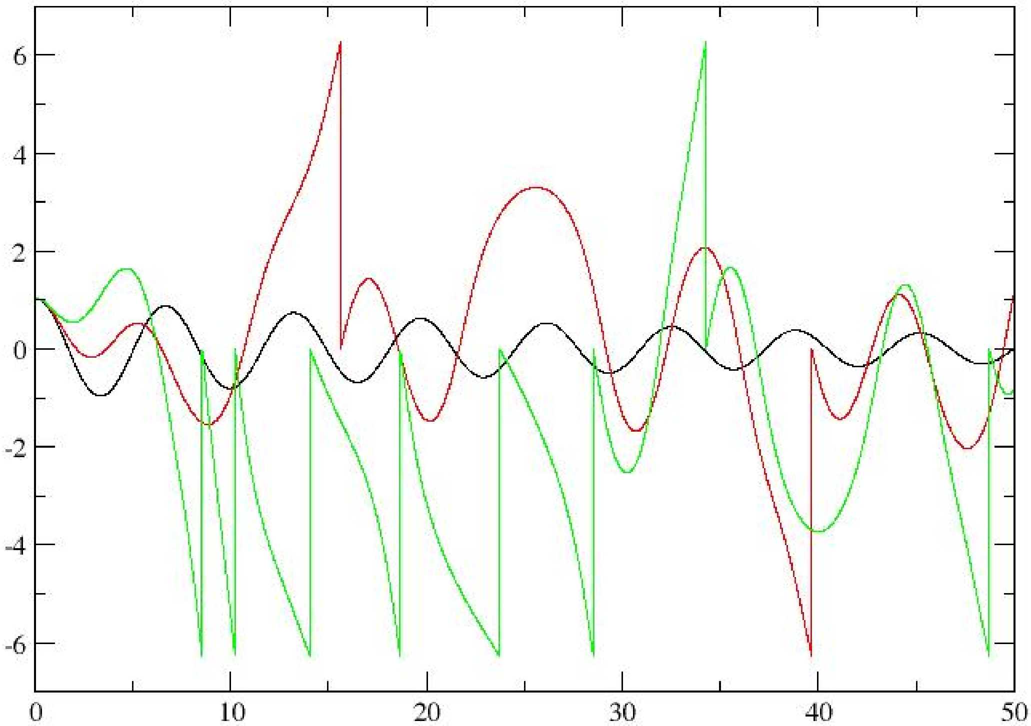

Figura 8: Em azul $F_0 = 0$, em vermelho $F_0 = 0.5$, em verde $F_0 = 1.2$.

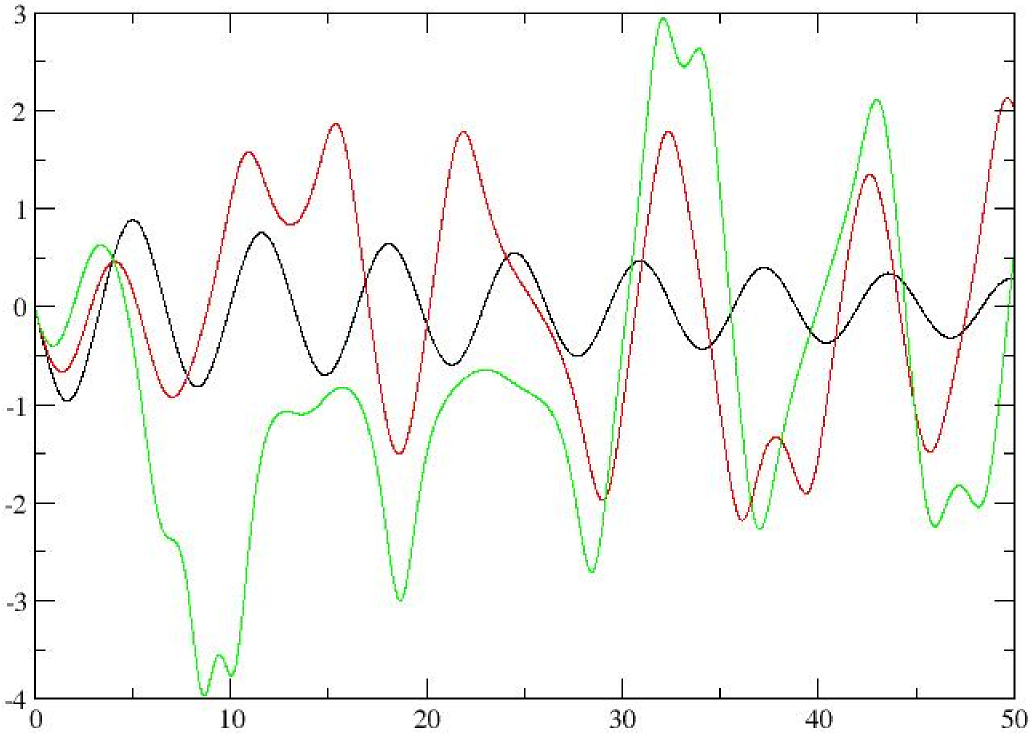

Figura 9: Em azul $F_0 = 0$, em vermelho $F_0 = 0.5$, em verde $F_0 = 1.2$.

É notório que quanto maior a força, mais a frequência se altera ao longo do tempo, assim como a amplitude também sofre mudanças, como podemos ver no gráfico 8. Vemos também que quando $F_0 = 0$ temos um amortecimento sub-amortecido, aonde a amplitude diminui lentamente mas a frequência permanece a mesma. Dado o gráfico 9 é possível ver a frequência aumentando bruscamente perto do final do gráfico, isso se deve ao ganho de energia dada pela força F_0 que ultrapassa o amortecimento.

2.3 Tarefa 3

Figura 10: Tarefa 3

```
      program PenduloCaotico
      implicit real*8 (a-h,o-z)
      parameter(dt=0.04d0)
      parameter(g=9.8d0)
      parameter(al=9.8d0)
      parameter(n=5000)
      parameter(gama=0.05d0)
      parameter(omega=2d0/3d0)
      parameter(f0=0.3d0) !1.2

      pi = 4.0d0*atan(1.0d0)

      wi = 0.0d0
      w = 0.0d0
      wi2 = 0.0d0
      w2 = 0.0d0
      thetai = 1d0
      thetai2 = thetai + 0.001d0
      theta = 0.0d0
      theta2 = 0.0d0
      t = 0
      open(1, file='saida-3-13783203.dat')

      do i=1,n,1
          t=i*dt
          w=wi-(g/al)*sin(theta)*dt-gama*w*dt+f0*sin(omega*t)*dt
          theta=mod(thetai+w*dt, 2.0d0*pi)


          t=i*dt
          w2=wi2-(g/al)*sin(theta2)*dt-gama*w2*dt+f0*sin(omega*t)*dt
          theta2=mod(thetai2+w2*dt, 2.0d0*pi)

          write(1,*) t, dlog(abs(theta-theta2))

          wi=w
          thetai=theta

          wi2=w2
          thetai2=theta2
      end do

      close(1)
      end program
```

2.3.1 Explicando a tarefa 3

Nesta tarefa, o objetivo é conseguir um $\Delta\theta$, e construímos θ_1 e θ_2 de forma que $\theta_2 = \theta_1 + 0.001$. Depois repito o mesmo código que usei na tarefa passada, apenas criando variáveis a mais para me referir ao segundo θ. Como queremos encontrar λ que pertence à $\Delta\theta(t) \approx \exp \lambda t$, então já calculo o log da diferença.

2.3.2 Resultados

Analisando o gráfico podemos perceber uma clara diferença, aonde o caso não-caótico demonstra um aparente padrão. Dado $\theta_0 = 1$ rad, temos que:

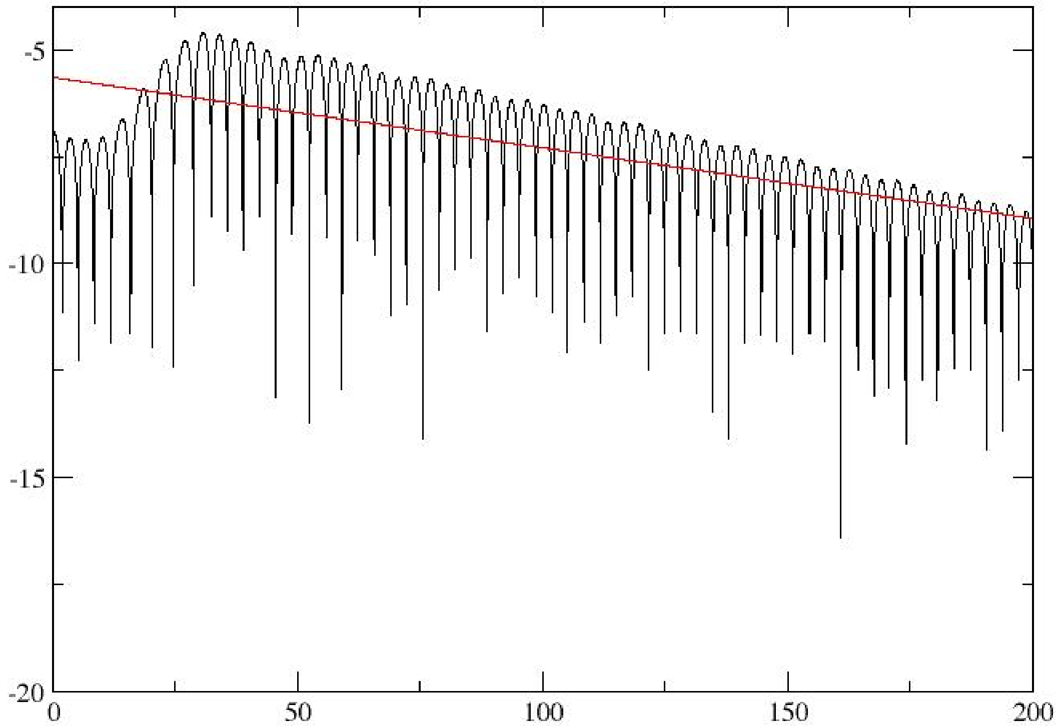

Figura 11: $F_0 = 0.3$. Temos $\lambda = -0.016475$, e é possível ver que é o caso não caótico.

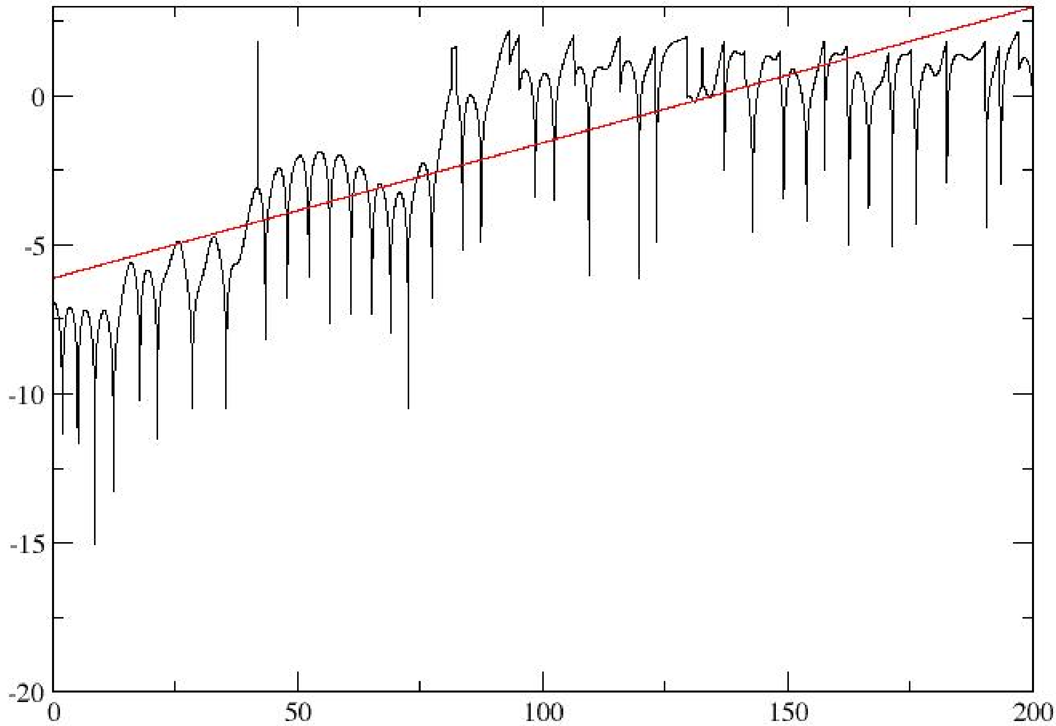

Figura 12: $F_0 = 0.5$. Temos $\lambda = +0.04546$, e é possível ver que é o caso caótico.

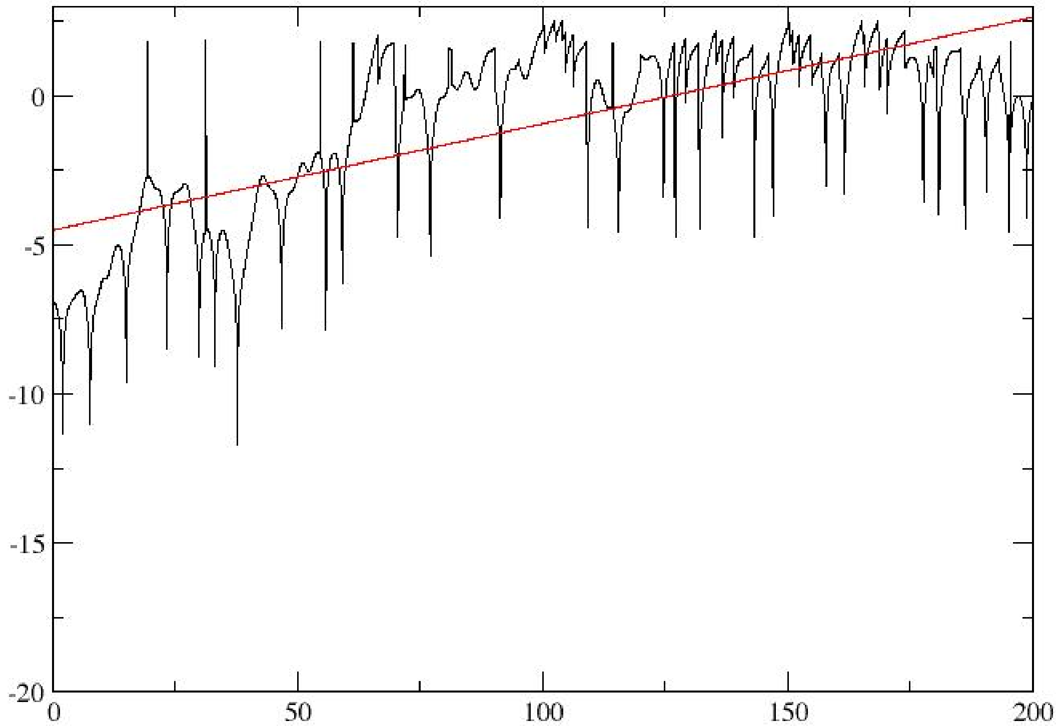

Figura 13: $F_0 = 1.2$. Temos $\lambda = +0.035706$, e é possível ver que é o caso caótico.

2.4 Tarefa 4

Figura 14: Tarefa 4

```
      program OmegaCaotico
      implicit real*8 (a-h,o-z)
      dimension f0(2)
      dimension thetas(3)
      character*26 fn

      f0 = [0.5d0,1.2d0]
      thetas = [1d0, 1.0001d0, 1.05d0]

      do jj=1,3
      do ii=1,2
          write(fn, 102) jj,ii
102       format('saida-4-13783203-',I0,'-3-',I0,'.dat')

          call run(thetas(jj),f0(ii),fn, ii)
      end do
      end do
      end program

      subroutine run(theta0,f0,name,indice)
      implicit real*8 (a-h,o-z)
      parameter(dt=0.04d0)
      parameter(g=9.8d0)
      parameter(al=9.8d0)
      parameter(n=5000)
      parameter(gama=0.5d0)
      parameter(omega=2d0/3d0)

      character*26 name

      pi = 4.0d0*atan(1.0d0)

      wi = 0.0d0
      w = 0.0d0
      thetai = theta0
      theta = 0.0d0
      t = 0

      open(indice,file=name)
      do i=1,n,1
          t=i*dt
          w=wi-(g/al)*sin(theta)*dt-gama*w*dt+f0*sin(omega*t)*dt
          theta=thetai+w*dt

          write(indice,*) theta, w

          wi=w
          thetai=theta
      end do
      close(indice)
      return
      end
```

2.4.1 Explicando a tarefa 4

Novamente, repeti o bruto do código da tarefa passada de forma que alterasse apenas para facilitar a obtenção de resultados deste caso. Sendo assim, fiz a sub-rotina *run* que me permite refazer toda lógica da tarefa passada, mas agora alterando F_0 e θ_0. Os valores escolhidos para θ_0 são $\pi/3$, $\pi/10$ e $\pi/30$.

2.4.2 Resultados

Sendo assim, os resultados foram:

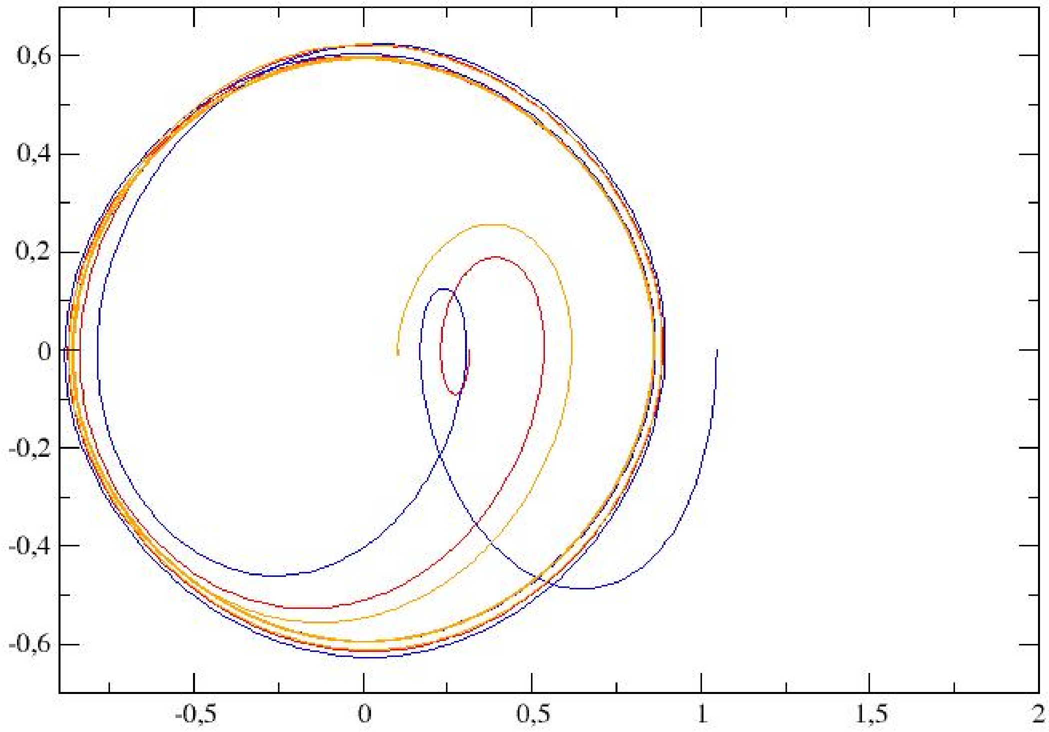

Figura 15: $F_0 = 0.5$. Em azul $\theta_0 = \pi/3$, em vermelho $\theta_0 = \pi/10$, em laranja$\theta_0 = \pi/30$

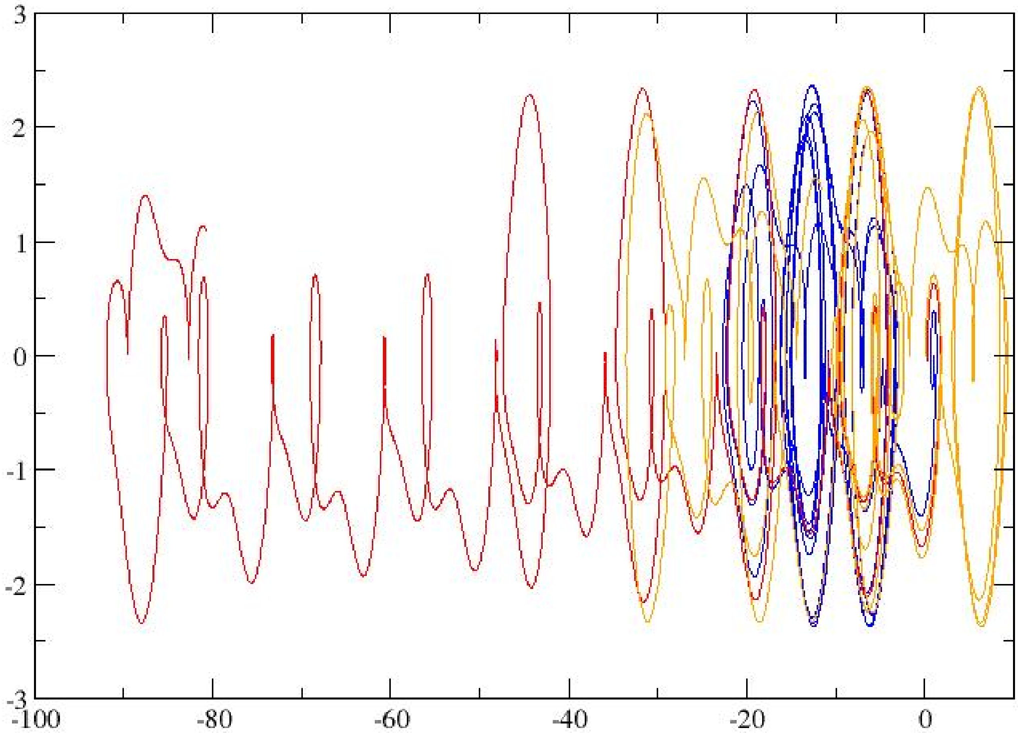

Figura 16: $F_0 = 1.2$. Em azul $\theta_0 = \pi/3$, em vermelho $\theta_0 = \pi/10$, em laranja$\theta_0 = \pi/30$

2.5 Tarefa 5

Figura 17: Tarefa 5

```
        program Poincare
        implicit real*8 (a-h,o-z)
        parameter(dt=0.04d0)
        parameter(g=9.8d0)
        parameter(al=9.8d0)
        parameter(n=1000000)
        parameter(gama=0.05d0)
        parameter(omega=2d0/3d0)
        parameter(f0=1.2d0)

        pi = 4.0d0*atan(1.0d0)

        wi = 0.0d0
        w = 0.0d0
        thetai = pi/3d0
        theta = 0.0d0
        t = 0
        open(1, file='saida-5-13783203.dat')

        do i=1,n,1
            t=i*dt
            w=w-(g/al)*dsin(theta)*dt-gama*w*dt+f0*dsin(omega*t)*dt
            theta=mod(theta+w*dt, 2.0d0*pi)

            if (mod(omega*t,2d0*pi) .lt. dt/2.0d0) then
                write(1,*) theta, w
            end if
        end do

        close(1)
        end program
```

2.5.1 Explicando a tarefa 5

Por último, mais uma vez utilizei a mesma lógica da tarefa passada, mas agora eu adicionei uma condição, para filtrar quais pontos poderão ser colocados no arquivo.

2.5.2 Resultados

Os resultados desta tarefa mostram exatamente o ”RG”que é dito na formulação do enunciado da tarefa, é possível notar que quanto mais caótico, menos denso é a secção de Poincaré.

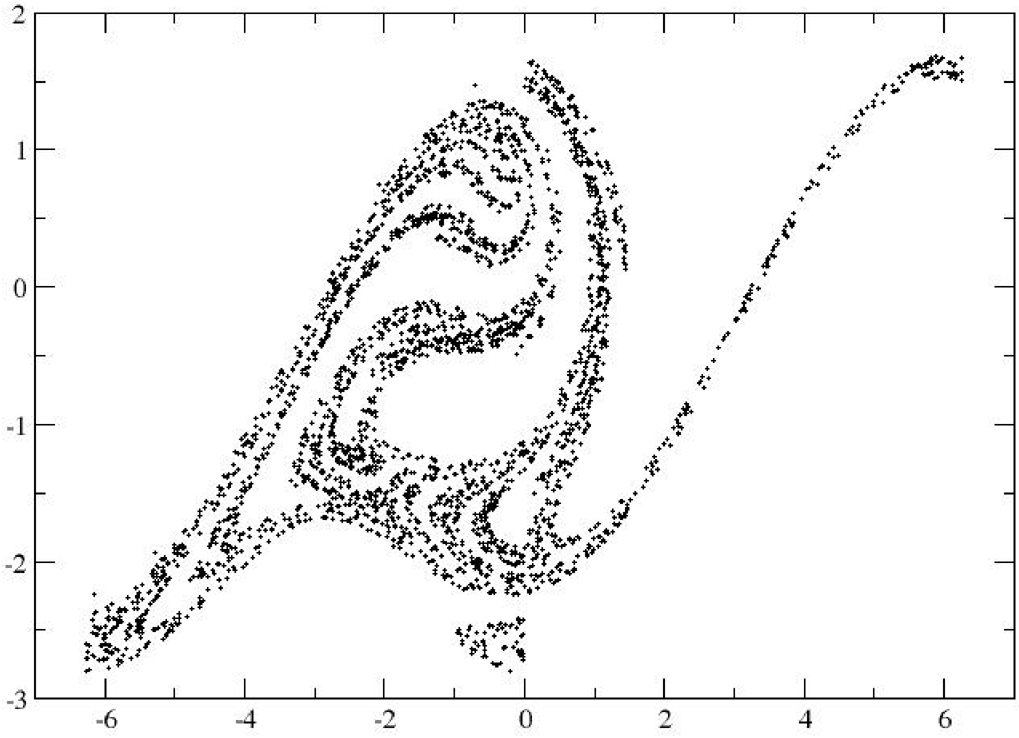

Figura 18: $F_0 = 0.5$.

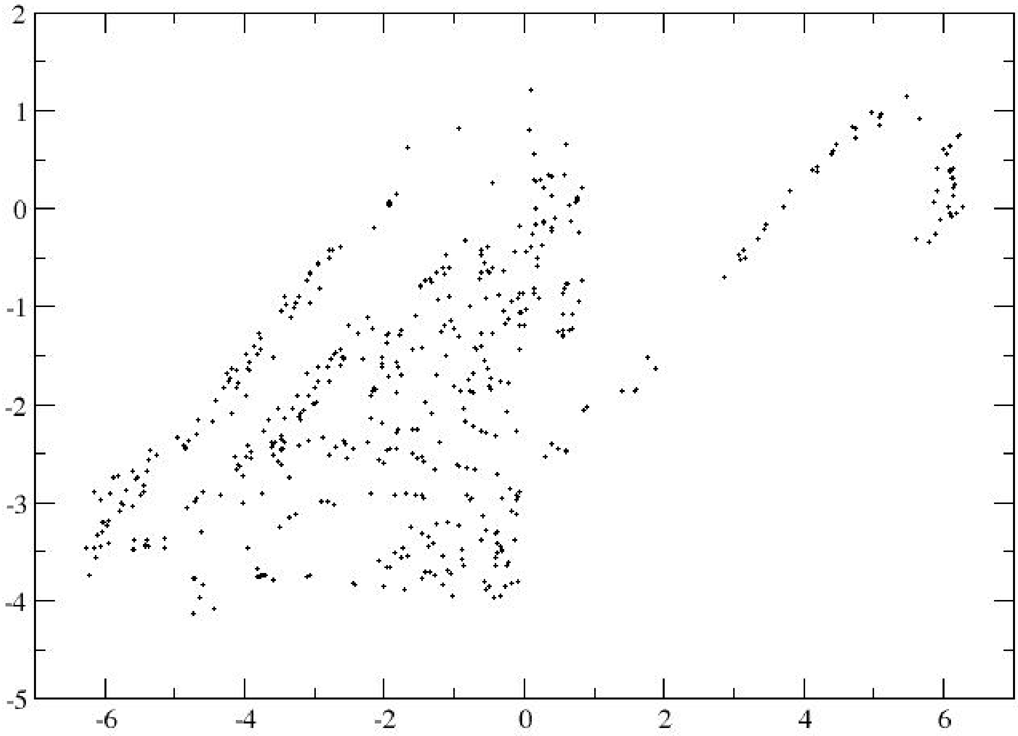

Figura 19: $F_0 = 1.2$.