Universidade de São Paulo
Instituto de Física de São Carlos

Projeto 4

Pedro Calligaris Delbem 5255417

Professor: Francisco Alcaraz

Novembro de 2023

Sumário

1 Tarefa A

Tarefa: Escreva um codigo em FORTRAN77 que simule o movimento de um pêndulo simples utilizando o método de euler, além de calcular sua energia, e depois repita o processo para o método de euler-cromer e compare ambos com o resultado analítico.

Código Escrito (Euler):

```
      program euler

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

c     define o valor de pi
      pi = 4.0d0*datan(1.0d0)

c     define o valores da gravidade, comprimento e massa
c     referentes ao pendulo
      g = 9.8d0
      r = 9.8d0
      am = 1.0d0

c     inicia o valor de theta e omega
      theta = pi/6.0d0
      omega = 0.0d0

c     inicia o valor de theta analitico
      theta0 = pi/6.0d0

c     defini o "tempo" de analise, qual o espacamento de "tempo"
c     entre as incrementacoes em theta e omega e o tempo inical
      tempomax = 80.0d0
      deltat = 0.04d0
      tempo = 0.0d0

c     abre os arquivos onde serao salvas as informacoes
      open(unit=1,file="euler")
      open(unit=2,file="energia")
      open(unit=3,file="analitico")
      open(unit=4,file="energia-analitica")

c     inicia o loop de oscilacao
      do while(tempo.lt.tempomax)

c           define o tempo atual
            tempo = tempo + deltat

c           salva o theta antigo, antes de altera-lo
            thetaant = theta

c           incrementa theta e omega se acordo com o metodo de euler
            theta = theta + omega*deltat
            omega = omega - (g/r)*thetaant*deltat

c           calcula a energia
```

```
            energia = r*am*( ((omega**2)*r)/2.0d0 + g*(1-dcos(theta)) )

c            escreve o theta(tempo) atual no arquivo e se theta passar,
c            em modulo, de 2pi - faz a carrecao adequada
             if(abs(theta).ge.2.0d0*pi) then
                   write(1,*)tempo,mod(theta,2.0d0*pi)
             else
                   write(1,*)tempo,theta
             end if

c            escreve o energia(tempo) atual no arquivo
             write(2,*)tempo,energia

c            escreve o theta(tempo) analiticoatual, no arquivo
             thetaanalitico = theta0*dcos(dsqrt(g/r)*tempo)
             write(3,*)tempo,thetaanalitico

c            calcula a energiaanalitica
             omegaanalitico = -dsqrt(g/r)*theta0*dsin(dsqrt(g/r)*tempo)
             energiaanalitica = r*am*( ((omegaanalitico**2)*r)/2.0d0 + g*
     1(1-dcos(thetaanalitico)) )

c            escreve a energiaanalitica(tempo) atual no arquivo
             write(4,*)tempo,energiaanalitica

       end do

c      fecha os arquivos utilizados
       close(1)
       close(2)
       close(3)
       close(4)

       end program
```

Descrição:

Primeiramente defini-se todas as variáveis reais como dupla precisão.

A seguir, defini-se o valor inicial das variáveis utilizadas, como pi (=4arctan(1)), comprimento do pendulo (r=9.8), gravidade (g=9.8), massa (am=1), theta (=pi/6), omega (=0), theta0 (=pi/6) - que servirá para simular o pêndulo através da simulação do resultado analítico - tempomax (=80) - que será o tempo em que a simulação se encerrará - detat (=0.04) - que é o intervalo de tempo entre cada interação - e por fim o tempo (=0).

Abre-se três arquivos onde salver-se-a theta(tempo), pelo método de euler, energia(tempo), theta(tempo), analítico, e energia(tempo), analítica.

Após isso, inicia-se um loop até o tempo ser igual ao tempomax. Neste, incrementa-se o tempo por deltat a cada interação.

Durante uma interação, é salvo o theta, antes de alterálo, numa variável thetaant, e então - utilizando as equações abaixo

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

$$\theta_{i+1} = \theta_i + \omega_i \Delta t \tag{2}$$

Em seguida, calcula-se a energia, veja - abaixo - o precesso para chegar na fórmula utilizada: A energia, neste caso, é dada por

$$E = \frac{1}{2}mv^2 - mgh \tag{3}$$

sabendo que

$$v = \omega r \tag{4}$$

e também que tomando h = 0, em theta = 0,

$$h = r(1 - cos(\theta)) \tag{5}$$

obtemos, por fim, a fórmula utilizada é

$$rm(\frac{\omega^2 r}{2} + g(1 - cos(\theta))) \tag{6}$$

Assim, calcula-se a energia - para método de euler.

Em seguida, escreve-se tempo,theta (euler) - no arquivo correspondente (ajustando o valor para sempre fosse menor que o módulo de 2pi). Escreve-se tempo,energia (euler).

Após isso, utilizando a solução analítica do pêndulo

$$\theta = \theta_0 cos(\sqrt{\frac{g}{r}}t) \tag{7}$$

escreve-se tempo, thetaanalitico com sua derivada, sendo o omega analítico

$$\omega = -\sqrt{\frac{g}{r}}\theta_0 sin(\sqrt{\frac{g}{r}}t) \tag{8}$$

escreve-se, utilizando (6), a energia analítica.

Ao fim do loop, fecha-se todos os arquivos.

Código Escrito (Euler-Cromer):

```
      program eulercromer

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

c     define o valor de pi
```

```
      pi = 4.0d0*datan(1.0d0)

c     define o valores da gravidade, comprimento e massa
c     referentes ao pendulo
      g = 9.8d0
      r = 9.8d0
      am = 1.0d0

c     inicia o valor de theta e omega
      theta = pi/6.0d0
      omega = 0.0d0

c     defini o "tempo" de analise, qual o espacamento de "tempo"
c     entre as incrementacoes em theta e omega e o tempo inical
      tempomax = 80.0d0
      deltat = 0.04d0
      tempo = 0.0d0

c     abre os arquivos onde serao salvas as informacoes
      open(unit=1,file="euler-cromer")
      open(unit=2,file="energia-conservada")

c     inicia o loop de oscilacao
      do while(tempo.lt.tempomax)

c           define o tempo atual
            tempo = tempo + deltat

c           incrementa theta e omega se acordo com o metodo de euler
            omega = omega - (g/r)*theta*deltat
            theta = theta + omega*deltat

c           calcula a energia
            energia = r*am*( ((omega**2)*r)/2.0d0 + g*(1-dcos(theta)) )

c           escreve o theta(tempo) atual no arquivo e se theta passar,
c           em modulo, de 2pi - faz a carrecao adequada
            if(abs(theta).ge.2.0d0*pi) then
                  write(1,*)tempo,mod(theta,2.0d0*pi)
            else
                  write(1,*)tempo,theta
            end if

c           escreve o energia(tempo) atual no arquivo
            write(2,*)tempo,energia

      end do

c     fecha os arquivos utilizados
      close(1)
      close(2)

      end program
```

Descrição:

Faz o mesmo do códido anterior, mas utiliza o método de euler-cromer que resulta nas seguintes equações:

$$\omega_{i+1} = \omega_i - \frac{g}{r}\theta_i \Delta t \tag{9}$$

$$\theta_{i+1} = \theta_i + \omega_{i+1} \Delta t \tag{10}$$

Além disso, não calcula os resultados analíticos, uma vez que já os temos.

Resultados:

Plotando em um mesmo gráfico o resultado do método de euler e da solução analítica

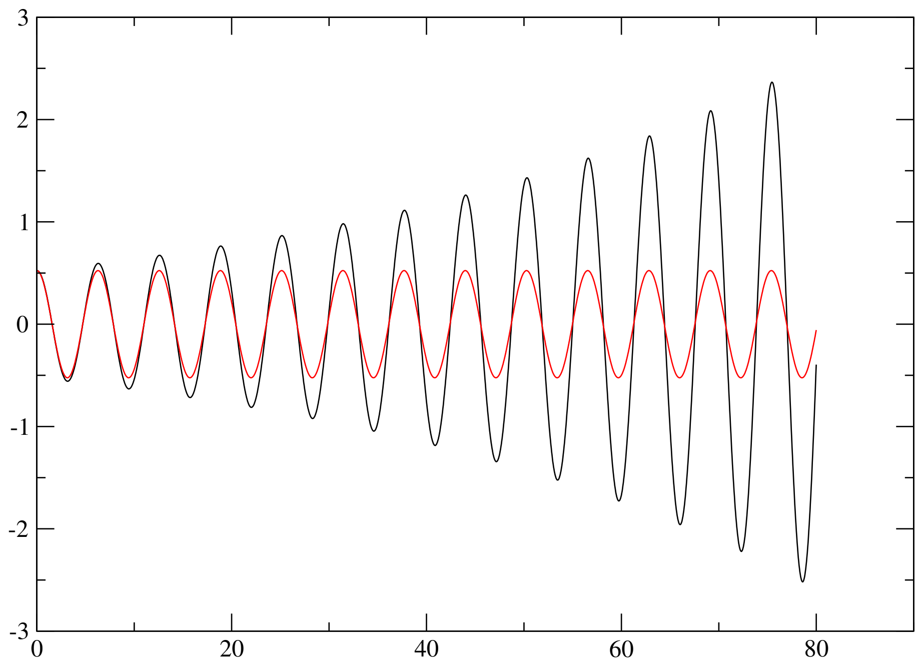

Gráfico: theta X tempo
em vermelho: solução analítica
em preto: método de euler

É notório que o método de euler cromer funciona apenas no começo da oscilação, após isso o mesmo diverge para ângulos maiores que o resultado real.

Analisando, agora, a energia

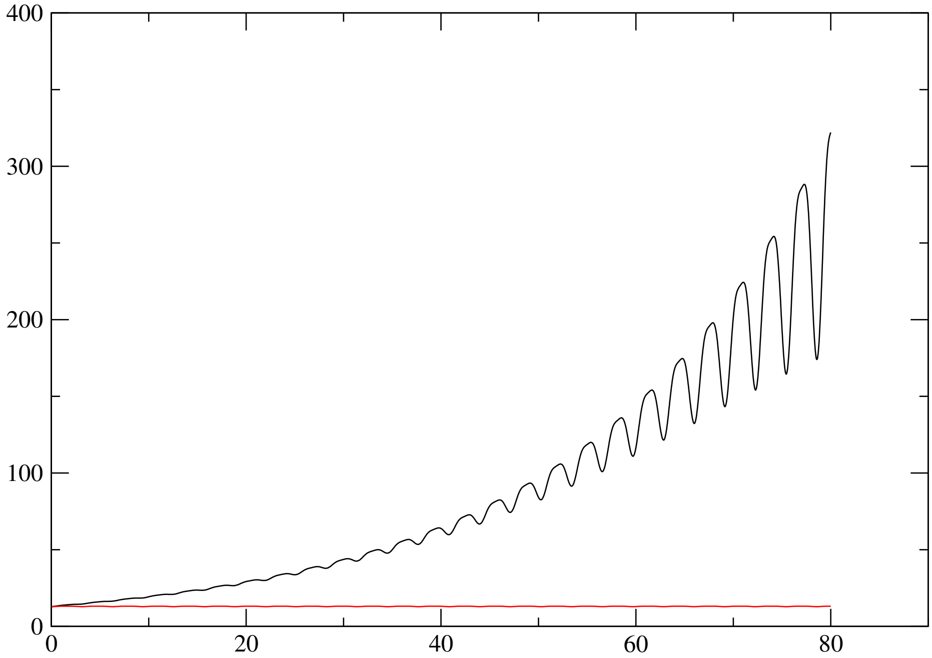

Gráfico: energia X tempo
em vermelho: solução analítica
em preto: método de euler

Analisando a energia, o problema fica ainda mais evidente. Enquanto a energia analítica é aproximadamente constante, a energia pelo método de euler cresce rapidamente.

Analisando agora theta, para o método de euler-cromer

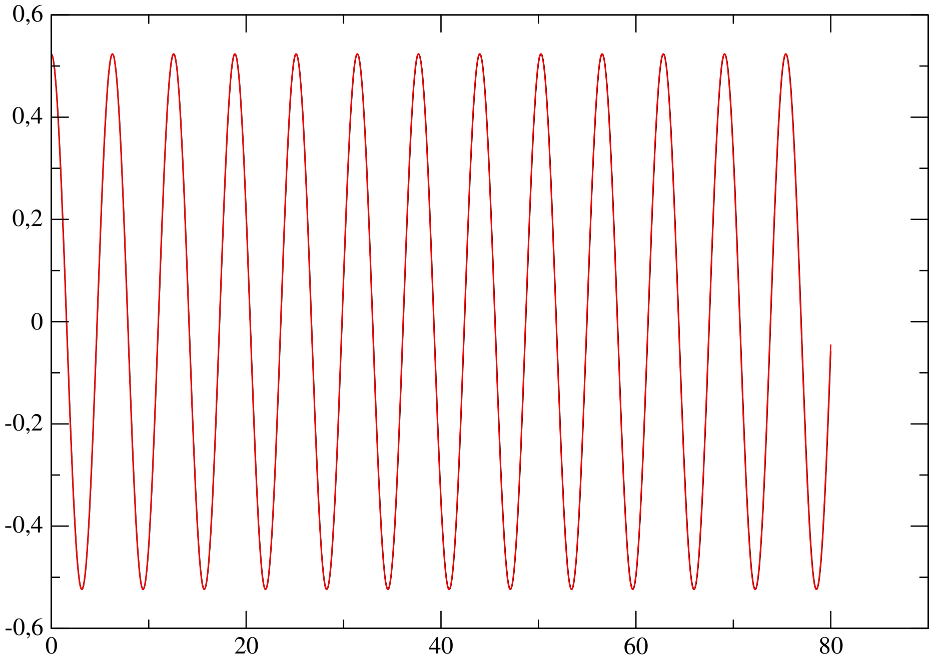

Gráfico: theta X tempo
em vermelho: solução analítica
em preto: método de euler

É notório que, apesar de ocorrerem pequenas divergências, o método de euler-cromer - praticamente - se iguala à solução analítica.

Analisando, então, a energia

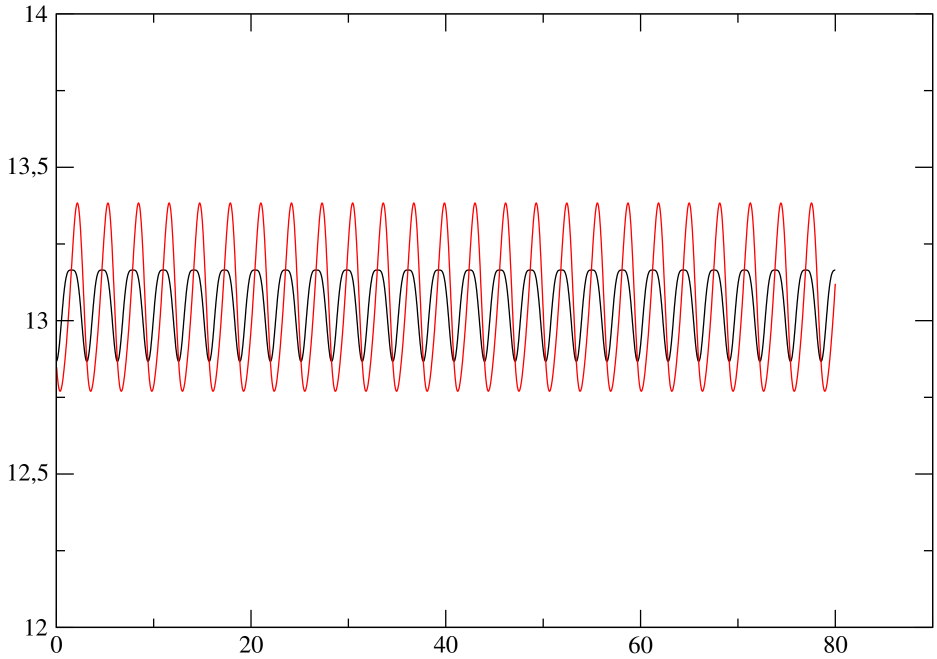

Gráfico: energia X tempo
em preto: solução analítica
em vermelho: método de euler

Por erro da solução analítica - que é calculada considerando ângulos pequenos -, a energia analítica oscila um pouso, mas é constante considerando um erro de $\pm 0,25$. Além disso, a energia para euler-cromer, também oscila devido a este erro e à própria aproximação do método, mas considerando um erro - um pouco maior - de $\pm 0,5$ temos uma energia constante, como esperado.

2 Tarefa B

2.1 B1

Tarefa: Escreva um codigo em FORTRAN77 que simule o movimento de um pêndulo simples utilizando o método de euler-cromer e, através da simulação, calcule o período do pêndulo, além calcular o período através da integral elíptica. Compare ambos os resultados.

Código escrito:

```
      program periodopendulo
```

```fortran

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

c     define o valor de pi
      pi = 4.0d0*datan(1.0d0)

c     define o valores da gravidade, comprimento e massa
c     referentes ao pendulo
      g = 9.8d0
      r = 9.8d0
      am = 1.0d0

c     defini qual o espacamento de "tempo" entre as
c     incrementacoes em theta
      deltat = 0.04d0

c     abre os arquivos onde serao salvas as informacoes
      open(unit=1,file="periodo")
      open(unit=2,file="periodo-analitico")

c     inicia o loop para thetas diferentes
      do i=1,30

c           inicia o valor de theta e omega
            theta = 0.1d0*i
            theta0 = theta
            omega = 0.0d0

c           (re)inicia o tempo e o pcontrolador
            tempo = 0.0d0
            pcontrolador = 0

c           inicia o loop de oscilacao ate que o pcontrolador
c           seja igual a 100
            do while(pcontrolador.lt.100)

c                 salva o valor de theta antes de altera-lo
                  omegaant = omega

c                 define o tempo atual
                  tempo = tempo + deltat

c                 incrementa theta e omega se acordo com o metodo de euler
                  omega = omega - (g/r)*dsin(theta)*deltat
                  theta = theta + omega*deltat

c                 incrementa um em pcontrolador se a velocidade mudar
                  if(omega*omegaant.lt.0.0d0)then
                        pcontrolador = pcontrolador + 1
                  end if

            end do

c           define o peiodo como tempo/50, pois ocorrerao 50 oscilacoes
            tempo = tempo/50.d0
```

```

c           escreve o theta(tempo) atual no arquivo e se theta passar,
c           em modulo, de 2pi - faz a carrecao adequada
            if(abs(theta).ge.2.0d0*pi) then
                  write(1,*)tempo,mod(theta0,2.0d0*pi)
            else
                  write(1,*)tempo,theta0
            end if

c           define o epson como 1% de theta0
            epson = theta0*0.01d0

c           define o valor inicial de h
            h = (theta0-epson)/1000000.0d0
            hi = h

c           (re)inicia o periodo
            periodo = 0.0d0

c           define o do pra somar os valores da integral
            do while(h.le.(theta0-epson))

                  valor = b(h-hi,theta0,hi)
                  periodo = periodo + valor

                  h = h + 4*hi

            end do

c           calcula o periodo
            periodo = 2.0d0*dsqrt(2.0d0*r/g)*(periodo+2.0d0*dsqrt(epson/
     1dsin(theta0)))

c           escreve o periodo(theta) no arquivo
            write(2,*)periodo,theta0

      end do

c     fecha os arquivos utilizados
      close(1)
      close(2)

      end program

c     define a integral que define o periodo
      real*8 function f(theta,theta0,h)
      implicit real*8 (a-h,o-z)

      f = 1.0d0/dsqrt(dcos(theta+h)-dcos(theta0))

      end function

c     define a regra de Boole
      real*8 function b(x,x0,h)
      implicit real*8 (a-h,o-z)

```

```
    b = 2.0d0*h/45.0d0*(7.0d0*f(x,x0,0.0d0)+32.0d0*f(x,x0,h*1.0d0
)+12.
  10d0*f(x,x0,h*2.0d0)+32.0d0*f(x,x0,3.0d0*h)+7.0d0*f(x,x0,4.0d0
*h))

    end function
```

Descrição:

Utilizando o a mesma simulação do problema anterior, mas modificando a equação (9) para (retirando a aproximação para pequenos ângulos)

$$\omega_{i+1} = \omega_i - \frac{g}{r} sin(\theta_i)\Delta t \tag{11}$$

(e abrindo dois arquivos para salvar o periodo e o período analítico), inicia-se um loop de i=1 até 30 onde defini-se theta como 0.1 vezes i, além de salvar tal valor em theta0, para utilizar no cálculo da integral. Também defini-se omega como 0.

Após isso, inicia-se o tempo em 0, defini-se deltat como 0.04 e um controlador (pcontrolador=0) que servirá para determinar os períodos. Inicia-se um subloop até que pcontrolador seja 100, onde é salvo o valor de omega, antes de alterá-lo, em omegaaant. Realiza-se, então, - a cada interação - o método de euler-cromer e ao fim da interação é feita a verificação se ocorreu mudança de sinal no omega e caso positivo incrementa-se pcontrolador.

Quando pcontrolador atinge o valor de 100, significa que ocorreram 50 oscilações, assim dividi-se o tempo por 50 para descobrir o período de um oscilação.

Salvando tempo,theta no arquivo "periodo"(fazendo a correção para o caso theta maior que 2pi).

Em seguida o programa calcula o perído através da seguinte integral

$$T = \sqrt{\frac{2r}{g}} \int_{-\theta_0}^{\theta_0} \frac{d\theta}{\sqrt{cos(\theta) - cos(\theta_0)}} \tag{12}$$

como esta integral tem problemas em $-\theta_0$ e em θ_0, tomamos um epson de modo que

$$T = \sqrt{\frac{2r}{g}} \int_{-\theta_0+\epsilon}^{\theta_0-\epsilon} \frac{d\theta}{\sqrt{cos(\theta) - cos(\theta_0)}} + 2A \tag{13}$$

onde A é o erro causado pela mudança em um extremo de integração (ambos os erros são iguais pois a integral é par). Como a integral é par, podemos fazer

$$N = \sqrt{\frac{2r}{g}} \int_{0}^{\theta_0\epsilon} \frac{d\theta}{\sqrt{cos(\theta) - cos(\theta_0)}} \tag{14}$$

de modo que

$$T = 2N + 2A \tag{15}$$

Para calcular o A, fazemos

$$\theta = -\theta_0 + \psi \tag{16}$$

deste modo, como ψ é pequeno, podemos fazer

$$cos(\theta) - cos(\theta_0) = sen(\theta_0)\psi \tag{17}$$

assim, calcula-se A por

$$A = \sqrt{\frac{2r}{g}} \int_0^{\epsilon} \frac{d\psi}{\sqrt{sin(\theta_0)\psi}} \tag{18}$$

obtendo, por fim

$$A = 2\sqrt{\frac{2r\epsilon}{gsin(\theta_0)}} \tag{19}$$

Definindo ϵ como 1% de theta0 e utilizando, então, o método de Boole (utilizado no projeto anterior) o programa calcula, numericamente, a integral de N salvando o resultado em periodo.

Em seguida mutiplicao resultado da integral por $\sqrt{\frac{2r}{g}}$ e soma o valor de 2A, salvando o resultado em periodo.

Por fim, escreve-se no arquivo "periodo-analitico"theta0,periodo.

Encerra-se o loop e fecha-se os arquivos. Ao fim do programa estão definidas as funções necessárias para o cálculo numérico da integral pelo método de Boole.

Resultados:

Plotando, em um mesmo gráfico, ambos os cálculos de período - em função do theta inicial -, obteve-se

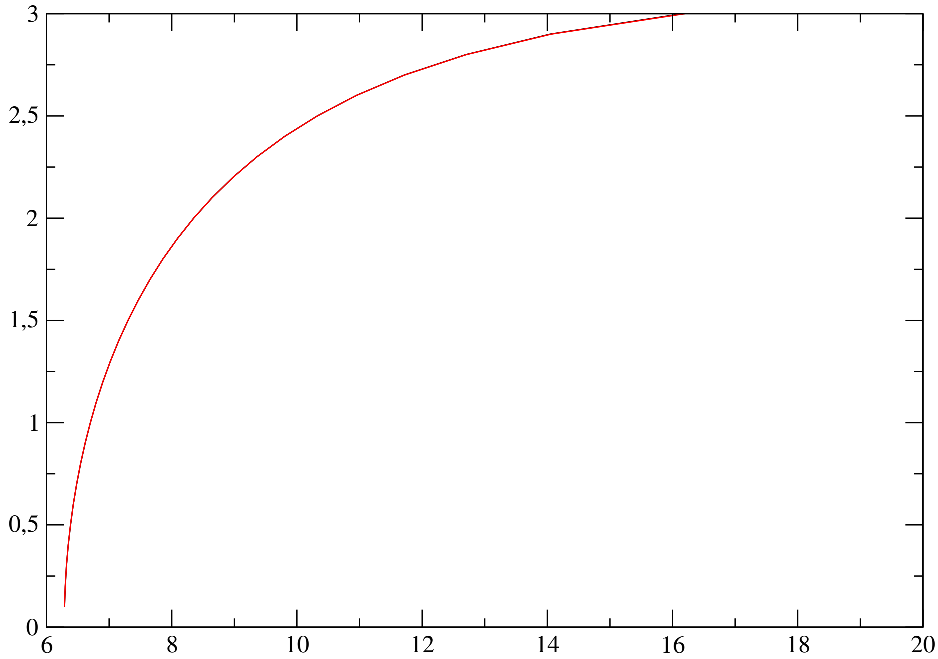

Gráfico: período X tempo
em preto: euler-cromer
em vermelho: integral elíptica

Nota-se que o gráfico preto é praticamente imperceptível. De modo que a integral elíptica nos dá o melhor valor possível dentro da dupla precisão.

2.2 B2

Tarefa: Escreva um codigo em FORTRAN77 que simule o movimento de um pêndulo simples utilizando o método de euler-cromer e, através da simulação, calcule o período do pêndulo, além calcular o período através da fórmula (19) válida para pequenos ângulos. Compare ambos os resultados.

Código escrito:

```
      program periodoangulopequeno

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

c     define o valor de pi
      pi = 4.0d0*datan(1.0d0)

c     define o valores da gravidade, comprimento e massa
c     referentes ao pendulo
```

```
      g = 9.8d0
      r = 9.8d0
      am = 1.0d0

c     defini qual o espacamento de "tempo" entre as
c     incrementacoes em theta
      deltat = 0.04d0

c     abre os arquivos onde serao salvas as informacoes
      open(unit=1,file="periodo")
      open(unit=2,file="periodo-analitico")

c     inicia o loop para thetas diferentes
      do i=1,20

c           inicia o valor de theta e omega
            theta = 0.1d0*i
            theta0 = theta
            omega = 0.0d0

c           (re)inicia o tempo e o pcontrolador
            tempo = 0.0d0
            pcontrolador = 0

c           inicia o loop de oscilacao ate que o pcontrolador
c           seja igual a 100
            do while(pcontrolador.lt.100)

c                 salva o valor de theta antes de altera-lo
                  omegaant = omega

c                 define o tempo atual
                  tempo = tempo + deltat

c                 incrementa theta e omega se acordo com o metodo de euler
                  omega = omega - (g/r)*dsin(theta)*deltat
                  theta = theta + omega*deltat

c                 incrementa um em pcontrolador se a velocidade mudar
                  if(omega*omegaant.lt.0.0d0)then
                        pcontrolador = pcontrolador + 1
                  end if

            end do

c           define o peiodo como tempo/50, pois ocorrerao 50 oscilacoes
            tempo = tempo/50.d0

c           escreve o theta(tempo) atual no arquivo e se theta passar,
c           em modulo, de 2pi - faz a carrecao adequada
            if(abs(theta).ge.2.0d0*pi) then
                  write(1,*)tempo,mod(theta0,2.0d0*pi)
            else
                  write(1,*)tempo,theta0
```

```
            end if

c           define o valor inicial de h
            h = (theta0-epson)/12.0d0
            hi = h

c           (re)calcula o periodo
            periodo = 2.0d0*pi*dsqrt(r/g)*(1+(theta0**2.0d0)/16.0d0
   )

            write(2,*)periodo,theta0

      end do

c     fecha os arquivos utilizados
      close(1)
      close(2)

      end program
```

Descrição:

Analogamente ao feito anteriormente, em um loop de 1 até 20, calcula-se o período para o euler-cromer do mesmo modo que o feito anteriormente e o período analítico por

$$T = 2\pi\sqrt{\frac{r}{g}}(1+\frac{\theta_0^2}{16}) \tag{20}$$

Resultados:

Plotando, em um mesmo gráfico, ambos os cálculos de período - em função do theta inicial -, obteve-se

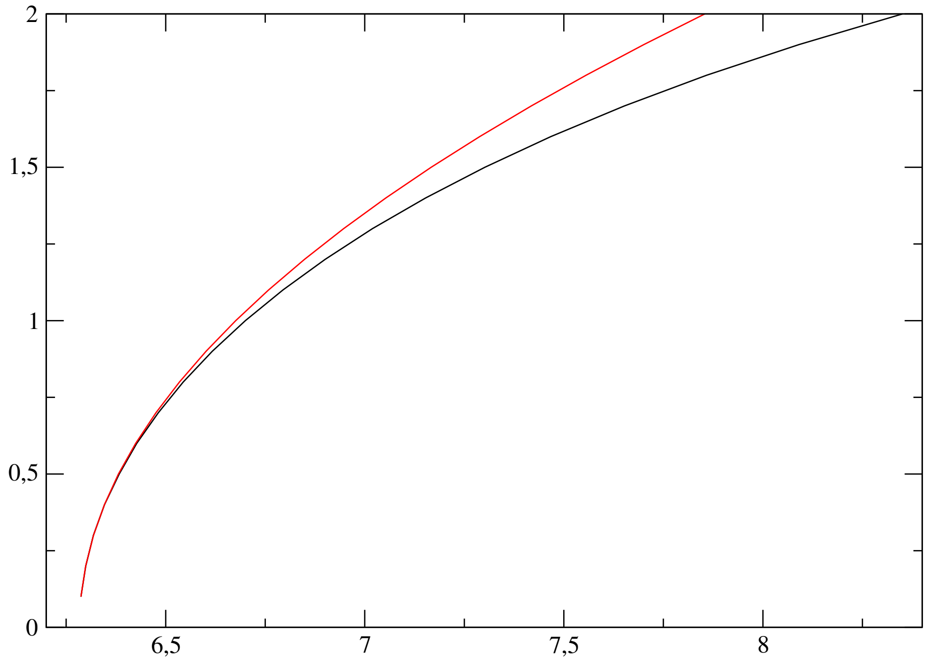

Gráfico: período X tempo
em preto: euler-cromer
em vermelho: equação(19)

Nota-se que, diferentemente do método anterior, este método só converge para o euler-cromer para pequenos ângulos.

2.3 B3

Tarefa: Escrever um programa em FORTRAN77 que simule o movimento de um pêndulo amortecido de coeficiente de amorteciment igual a 0,5.

Código escrito:

```
      program amortecido

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

c     define o valor de pi
      pi = 4.0d0*datan(1.0d0)

c     define o valores da gravidade, comprimento e massa
c     referentes ao pendulo
      g = 9.8d0
      r = 9.8d0
```

```
      am = 1.0d0

c     define a constante de amortecimento
      gamma = 0.5d0

c     inicia o valor de theta e omega
      theta = pi/6.0d0
      omega = 0.0d0

c     defini o "tempo" de analise, qual o espacamento de "tempo"
c     entre as incrementacoes em theta e omega e o tempo inical
      tempomax = 80.0d0
      deltat = 0.04d0
      tempo = 0.0d0

c     abre os arquivos onde serao salvas as informacoes
      open(unit=1,file="amortecido")

c     inicia o loop de oscilacao
      do while(tempo.lt.tempomax)

c           define o tempo atual
            tempo = tempo + deltat

c           incrementa theta e omega se acordo com o metodo
c           de euler amortecido
            omega = omega - (g/r)*theta*deltat - gamma*omega*deltat
            theta = theta + omega*deltat

c           escreve o theta(tempo) atual no arquivo e se theta passar,
c           em modulo, de 2pi - faz a carrecao adequada
            if(abs(theta).ge.2.0d0*pi) then
                  write(1,*)tempo,mod(theta,2.0d0*pi)
            else
                  write(1,*)tempo,theta
            end if

      end do

c     fecha os arquivos utilizados
      close(1)

      end program
```

Descrição:

Modificando a equação (9) para

$$\omega_{i+1} = \omega_i - \frac{g}{r}\theta_i \Delta t - \gamma \omega_i \Delta t \tag{21}$$

e realiza a mesma simulação da tarefa A, mas sem calcular a energia

Resultados:

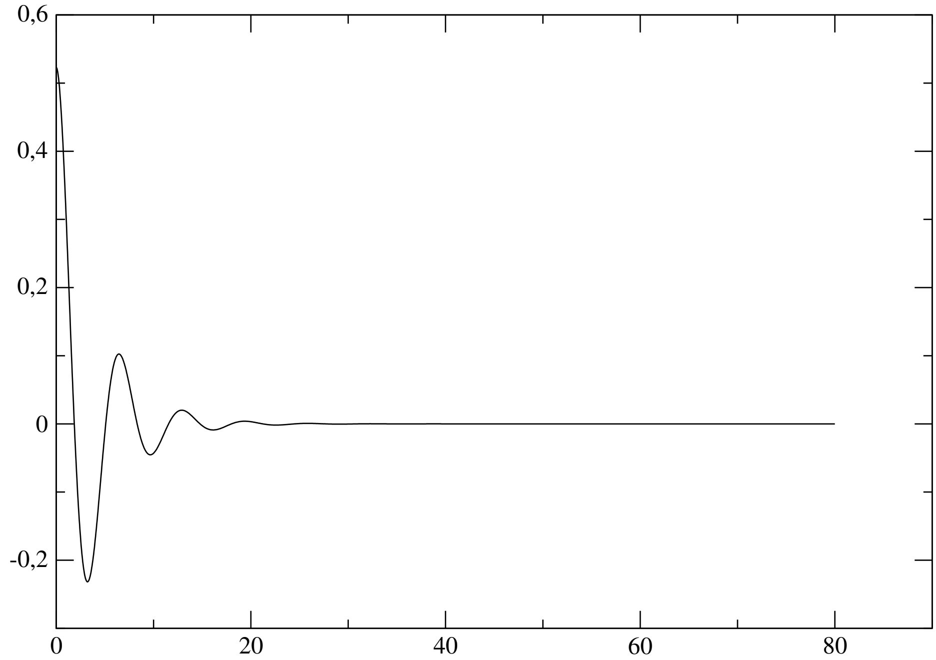

Gráfico: período X tempo
em preto: euler-cromer
em vermelho: equação(19)

Nota-se que este é o caso de amortecimento subcrítico, pois ocorrem oscilações antes do fim do movimento.

2.4 B4

Tarefa: Escrever um programa em FORTRAN77 que simule o movimento de um pêndulo amortecido, com coeficiente de amortecimento 0.05, de forçado com uma força F dada por:

$$F = F_0 sin(\Omega t) \tag{22}$$

para $\Omega = 2/3$ e F_0 igual a 0, 0.5 e 1.2

Código escrito:

```
      program amortecidoforcado

      implicit real*8(a-h,o-z)
      dimension amplitude(3)

c     define as amplitudes da forca
      amplitude(1) = 0
```

```
      amplitude(2) = 0.5d0
      amplitude(3) = 1.2d0

c     define o valor de pi
      pi = 4.0d0*datan(1.0d0)

c     define o valores da gravidade, comprimento e massa
c     referentes ao pendulo
      g = 9.8d0
      r = 9.8d0
      am = 1.0d0

c     define a constante de amortecimento e a frequencia da forca
      gamma = 0.05d0
      frequencia = 2.0d0/3.0d0

c     inicia o valor de theta e omega
      theta = pi/6.0d0
      omega = 0.0d0

c     defini o "tempo" de analise, qual o espacamento de "tempo"
c     entre as incrementacoes em theta e omega
      tempomax = 100.0d0
      deltat = 0.04d0

c     abre os arquivos onde serao salvas as informacoes
      open(unit=1,file="theta-livre")
      open(unit=4,file="omega-livre")
      open(unit=2,file="theta-forcado0.5")
      open(unit=5,file="omega-forcado0.5")
      open(unit=3,file="theta-forcado1.2")
      open(unit=6,file="omega-forcado1.2")

c     define o loop para cada amplitude
      do i=1,3

c           (re)define o tempo como 0
            tempo = 0.0d0

c           inicia o loop de oscilacao
            do while(tempo.lt.tempomax)

c                 define o tempo atual
                  tempo = tempo + deltat

c                 incrementa theta e omega se acordo com o metodo
c                 de euler amortecido
                  omega = omega - (g/r)*dsin(theta)*deltat - gamma*omega
     4*deltat + amplitude(i)*dsin(frequencia*tempo)*deltat
                  theta = theta + omega*deltat

c                 escreve o theta(tempo) atual no arquivo e se theta passar,
c                 em modulo, de 2pi - faz a carrecao adequada
                  if(abs(theta).ge.2.0d0*pi) then
                        write(i,*)tempo,mod(theta,2.0d0*pi)
                  else
```

```
                        write(i,*)tempo,theta
                  end if
c                 escreve o omega (theta) atual no arquivo
                        write(i+3,*)tempo,omega

            end do

      end do

c     fecha os arquivos utilizados
      close(1)
      close(2)
      close(3)
      close(4)
      close(5)
      close(6)

      end program
```

Descrição:

Primeiro defini-se uma lista de três valores contendo as aomplitudes de F, além de definir as variáveis também utilizadas nos códigos anteriores, defini-se a frequencia como 2/3.

Em seguida, abre-se os arquivos de theta e omega para cada amplitude.

Por fim, em um loop de i=1 até 3, realiza-se a mesma simulação do item B3, mas modificando (21) para

$$\omega_{i+1} = \omega_i - \frac{g}{r}\theta_i \Delta t - \gamma \omega_i \Delta t + F_0 sin(\Omega t)\Delta t \tag{23}$$

que no programa se traduz para a adição de um termo que consiste na multiplicação do seno (da frequencia pelo tempo) por deltat e pela lista amplitude indexada pelo i do loop.

Ao fim do loop escreve-se tempo,theta no arquivo de endereço 1 e tempo,omega no arquivo de endereço i+3.

Por fim, fecha-se todos os arquivos.

Resultados:

Gamma = 0.05:

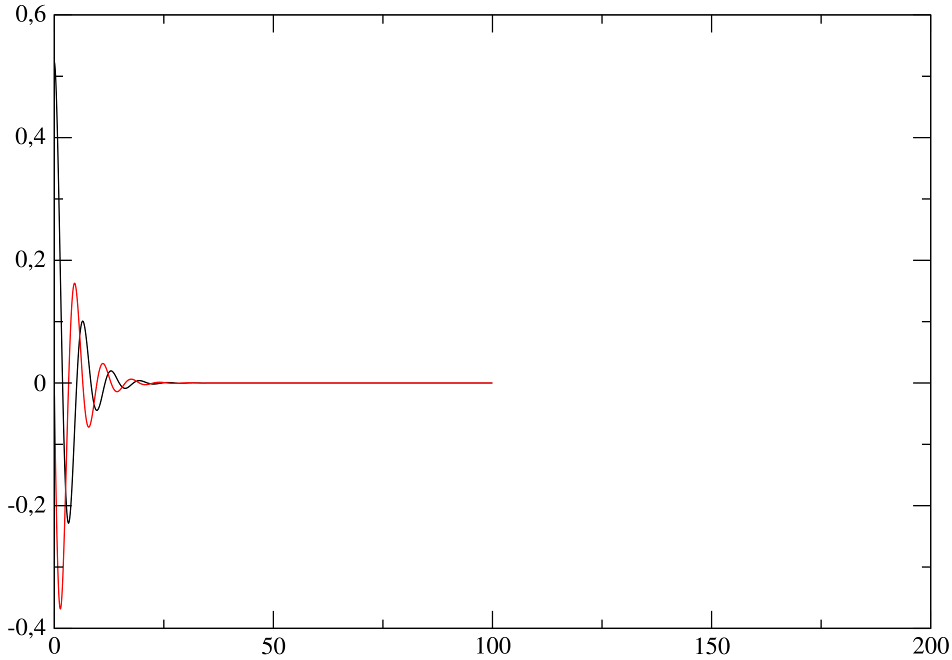

Gráfico: $F_0 = 0$
em preto: theta X tempo
em vermelho: omega X tempo

Tal qual o item B3 o gráfico mostra o esperado, o pêndulo oscila até ser completamente parado pela força de amortecimento.

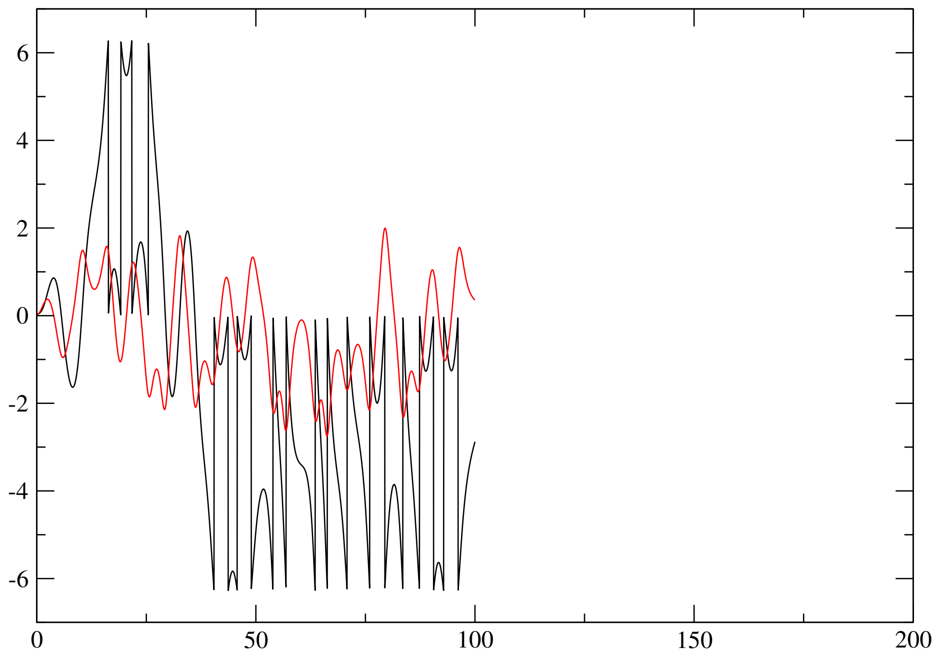

Gráfico: $F_0 = 0.5$
em preto: theta X tempo
em vermelho: omega X tempo

É notório que este movimento não é periódico, sendo entendido como caótico.

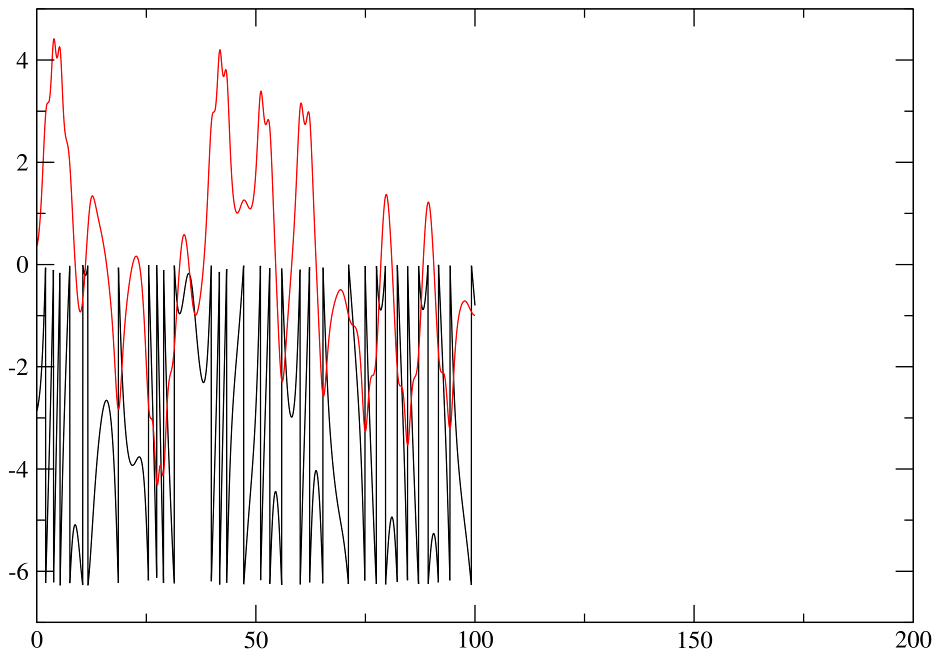

Gráfico: $F_0 = 1.2$
em preto: theta X tempo
em vermelho: omega X tempo

É notório que este movimento não é periódico, sendo entendido como caótico.

Gamma = 0.5:

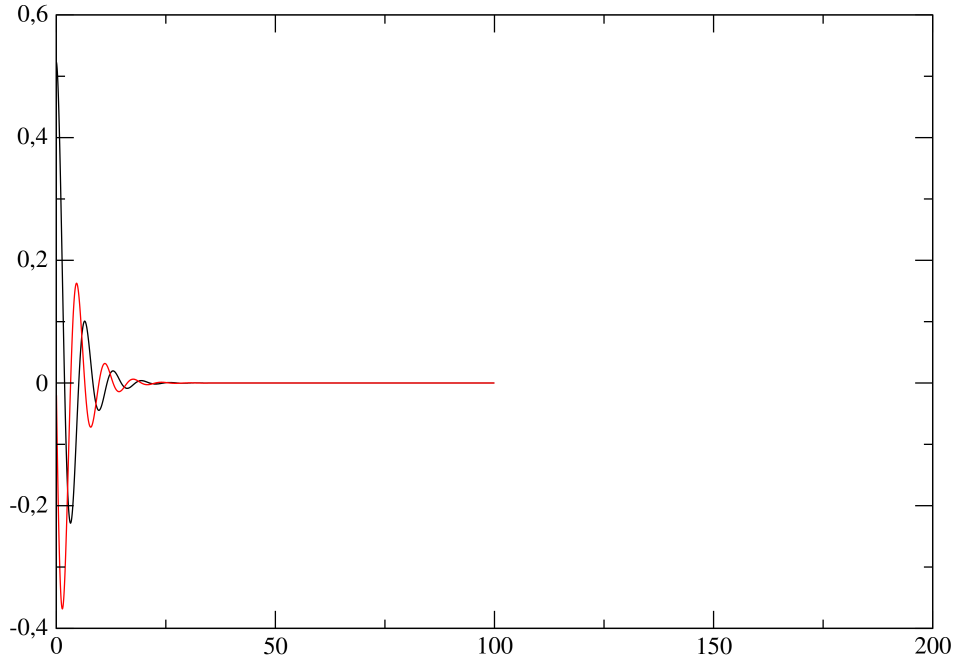

Gráfico: $F_0 = 0$
em preto: theta X tempo
em vermelho: omega X tempo

Tal qual o item B3 o gráfico mostra o esperado, o pêndulo oscila até ser completamente parado pela força de amortecimento.

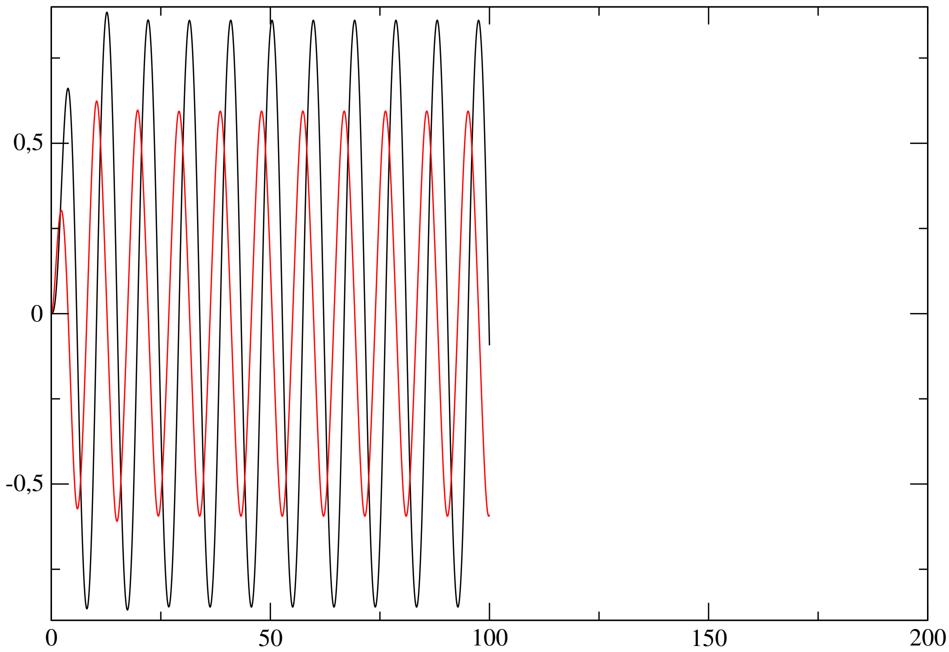

Gráfico: $F_0 = 0.5$
em preto: theta X tempo
em vermelho: omega X tempo

No início o gráfico apresenta uma frequência de oscilação diferente, mas devido ao termo de amortecimento, rapidamente passa a oscilar com uma frequência definida pela força F, ou seja, passa a oscilar com Ω.

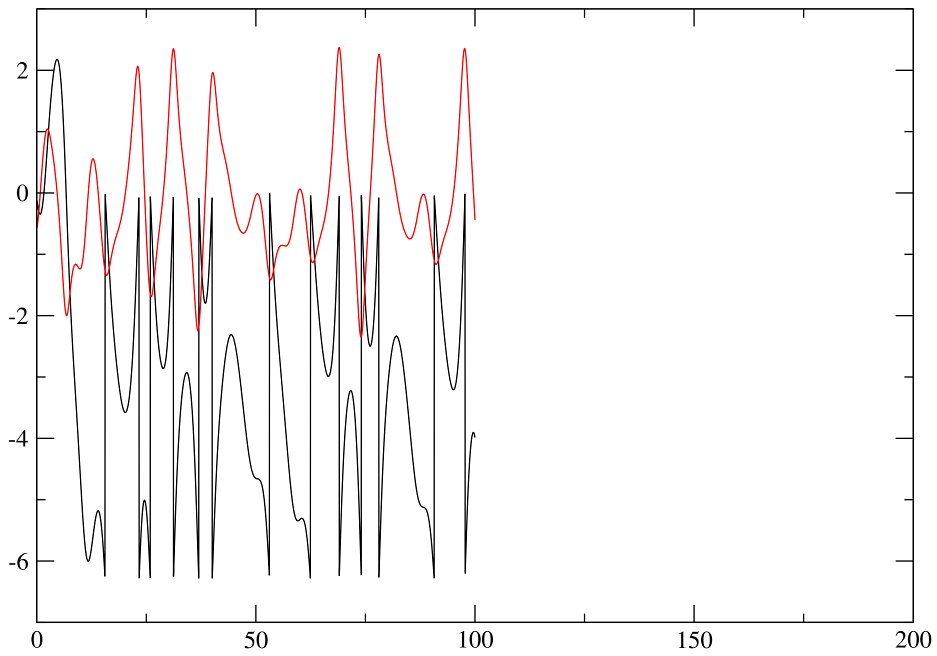

Gráfico: $F_0 = 1.2$
em preto: theta X tempo
em vermelho: omega X tempo

É notório que este movimento não é periódico, sendo entendido como caótico.

3 Tarefa C

Tarefa: Escrever um código em FORTRAN77 que através de dois valores de θ inicial que difiram por 0.001 radianos, plote um gráfico semi-logarítimo de $\Delta\theta$ X tempo - tanto para $F_0 = 0.5$ quanto para $F_0 = 1.2$ - afim de determinar qual movimento é caótico e qual não é.

Cógio escrito:

```
      program deltatheta

      implicit real*8(a-h,o-z)
      dimension amplitude(2)

c     define as amplitudes da forca
      amplitude(1) = 0.5d0
      amplitude(2) = 1.2d0

```

```
c     define o valor de pi
      pi = 4.0d0*datan(1.0d0)

c     define o valores da gravidade, comprimento e massa
c     referentes ao pendulo
      g = 9.8d0
      r = 9.8d0
      am = 1.0d0

c     define a constante de amortecimento e a frequencia da forca
      gamma = 0.05d0
      frequencia = 2.0d0/3.0d0

c     inicia o valor de theta e omega de acordo com a
c     solucao analitica
      theta1 = 1.0d0
      omega1 = 0.0d0
      theta2 = 1.001d0
      omega2 = 0.0d0

c     defini o "tempo" de analise, qual o espacamento de "tempo"
c     entre as incrementacoes em theta e omega
      tempomax = 40.0d0
      deltat = 0.04d0

c     abre os arquivos onde serao salvas as informacoes
      open(unit=1,file="amplitude0.5")
      open(unit=2,file="amplitude1.2")

c     define o loop para cada amplitude
      do i=1,2

c           (re)define o tempo como 0
            tempo = 0.0d0

c           inicia o loop de oscilacao
            do while(tempo.lt.tempomax)

c                 define o tempo atual
                  tempo = tempo + deltat

c                 incrementa theta1 e omega1 se acordo com o metodo
c                 de euler amortecido
                  omega1 = omega1 - (g/r)*dsin(theta1)*deltat - gamma*om
     1ega1*deltat + amplitude(i)*dsin(frequencia*tempo)*deltat
                  theta1 = theta1 + omega1*deltat

c                 incrementa theta2 e omega2 se acordo com o metodo
c                 de euler amortecido
                  omega2 = omega2 - (g/r)*dsin(theta2)*deltat - gamma*om
     2ega2*deltat + amplitude(i)*dsin(frequencia*tempo)*deltat
                  theta2 = theta2 + omega2*deltat

c                 escreve o theta(tempo), com escala semi-logaritmica,
c                 atual, no arquivo
```

```
                     write(i,*)tempo,dlog(abs(theta1-theta2))

               end do

        end do

c       fecha os arquivos utilizados
        close(1)
        close(2)

        end program
```

Descrição:

Utilizando uma lista com as duas amplitudes, tal qual no item B4, e iniciando duas variáveis para theta (theta1 (=1) e theta2 (=1.001)). Defni-se também o tempo máximo (tempomax = 40).

Analogamente ao feito no item B4, defini-se um loop de i=1 até 2 para realizar a simulação do pêndulo, utilizando (23), para ambas as amplitudes - realizando o mesmo processo para cada theta e omega.

Ao fim do loop, escreve-se no arquivo - correspondente à amplitude atual - tempo,ln($\Delta\theta$)

Ao fim, fecha-se os arquivos.

Resultados:

Gamma = 0.05:

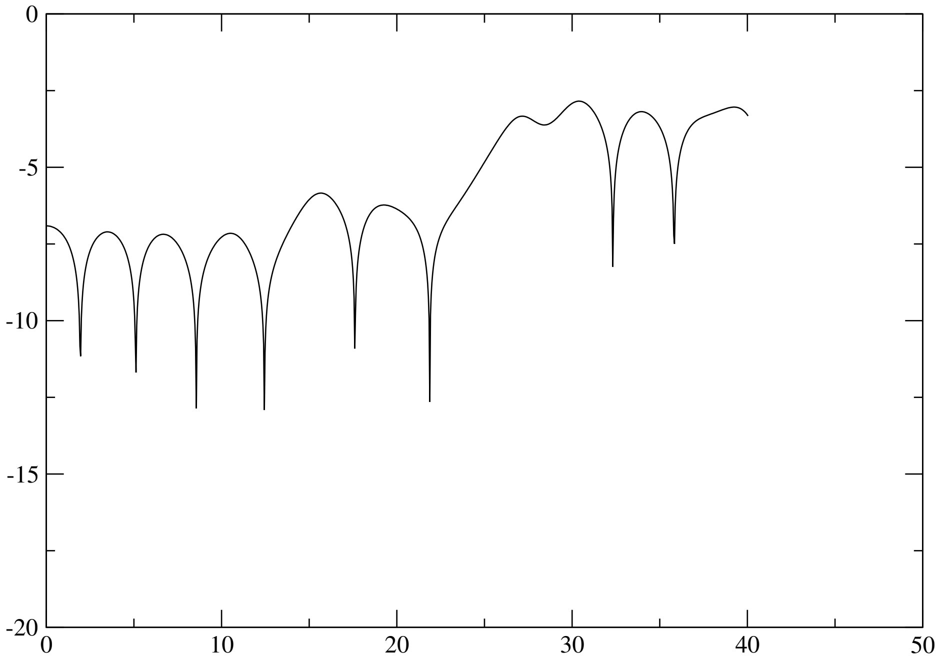

Gráfico: $F_0 = 0.5$
em preto: ln(Δtheta) X tempo

É notório que o gráfico é - em média - crescente, deste modo, é fácil implicar que o coeficiente de Liapunov maior que 0, tornando o movimento caótico. Tal qual visto, neste mesmo caso, no item B4.

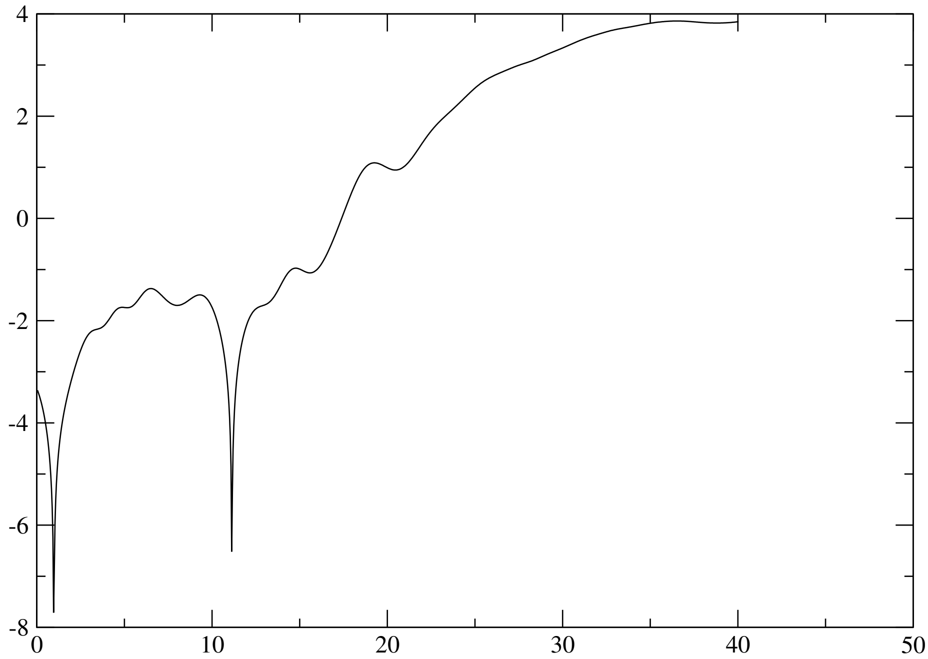

Gráfico: $F_0 = 1.2$
em preto: ln(Δtheta) X tempo

É notório que o gráfico é - em média - crescente, deste modo, é fácil implicar que o coeficiente de Liapunov maior que 0, tornando o movimento caótico. Tal qual visto, neste mesmo caso, no item B4.

Gamma = 0.5:

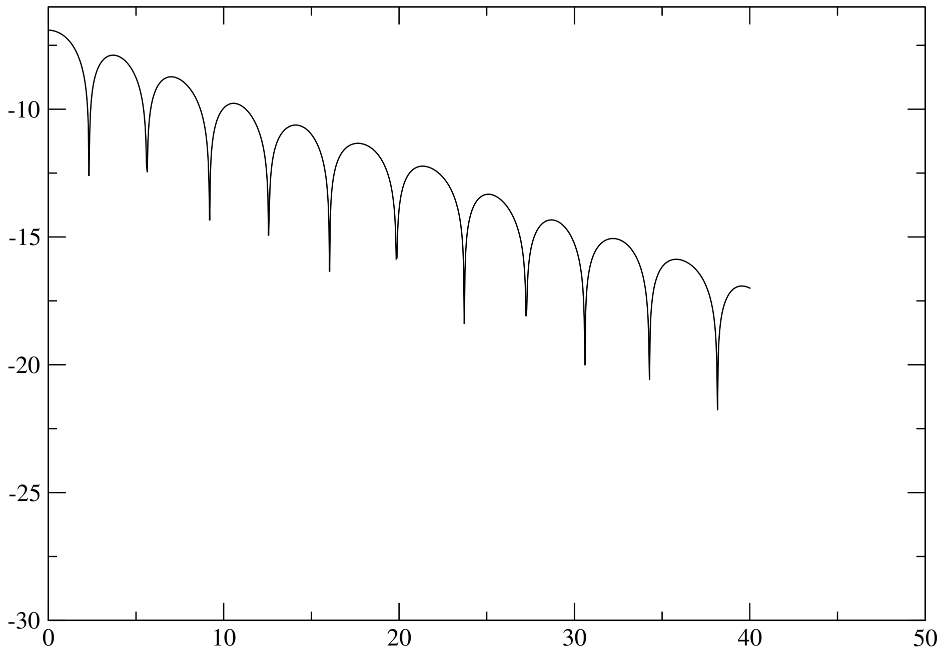

Gráfico: $F_0 = 0.5$
em preto: ln(Δtheta) X tempo

É notório que o gráfico é - em média - decrescente, deste modo, é fácil implicar que o coeficiente de Liapunov menor que 0, tornando o movimento periódico. Tal qual visto, neste mesmo caso, no item B4.

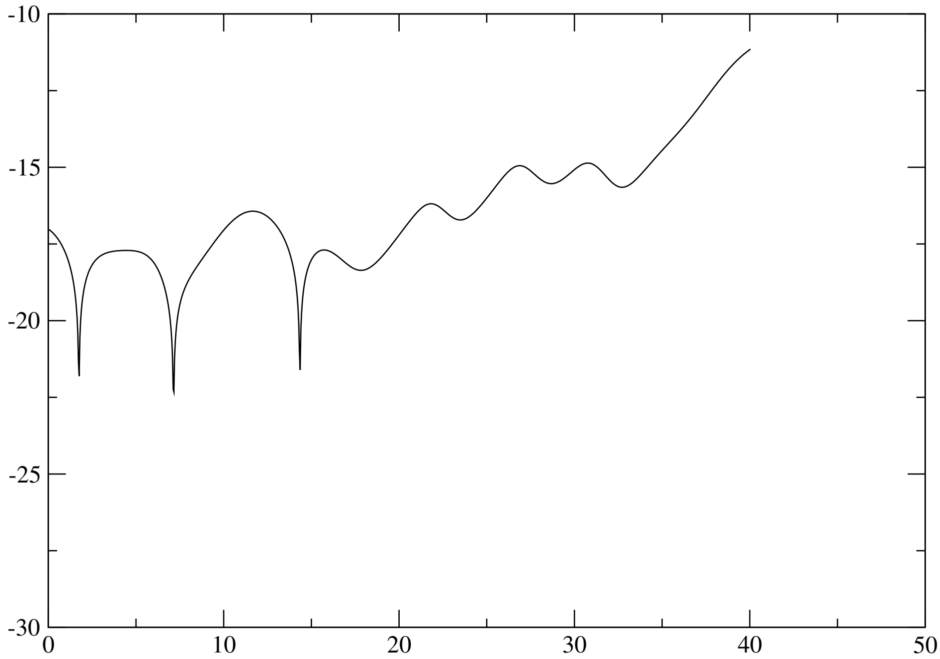

Gráfico: $F_0 = 1.2$
em preto: ln(Δtheta) X tempo

É notório que o gráfico é - em média - crescente, deste modo, é fácil implicar que o coeficiente de Liapunov maior que 0, tornando o movimento caótico. Tal qual visto, neste mesmo caso, no item B4.

4 Tarefa D

Tarefa: Escrever um código em FORTRAN77 que através de alguns valores de θ inicial que difiram por pouco, plote um gráfico de ω X θ, contendo cada θ inicial, - tanto para $F_0 = 0.5$ quanto para $F_0 = 1.2$.

Cógio escrito:

```
      program omegadetheta

      implicit real*8(a-h,o-z)
      dimension amplitude(2)

c     define as amplitudes da forca
      amplitude(1) = 0.5d0
      amplitude(2) = 1.2d0

```

```
c     define o valor de pi
      pi = 4.0d0*datan(1.0d0)

c     define o valores da gravidade, comprimento e massa
c     referentes ao pendulo
      g = 9.8d0
      r = 9.8d0
      am = 1.0d0

c     define a constante de amortecimento e a frequencia da forca
      gamma = 0.5d0
      frequencia = 2.0d0/3.0d0

c     inicia o valor de theta e omega de acordo com a
c     solucao analitica
      theta1 = 1.0d0
      omega1 = 0.0d0
      theta2 = 1.001d0
      omega2 = 0.0d0
      theta3 = 0.999d0
      omega3 = 0.0d0

c     defini o "tempo" de analise, qual o espacamento de "tempo"
c     entre as incrementacoes em theta e omega
      tempomax = 1000.0d0
      deltat = 0.04d0

c     abre os arquivos onde serao salvas as informacoes
      open(unit=1,file="amplitude0.5-1")
      open(unit=3,file="amplitude0.5-2")
      open(unit=5,file="amplitude0.5-3")
      open(unit=2,file="amplitude1.2-1")
      open(unit=4,file="amplitude1.2-2")
      open(unit=6,file="amplitude1.2-3")

c     define o loop para cada amplitude
      do i=1,2

c           (re)define o tempo como 0
            tempo = 0.0d0

c           inicia o loop de oscilacao
            do while(tempo.lt.tempomax)

c                 define o tempo atual
                  tempo = tempo + deltat

c                 incrementa theta1 e omega1 se acordo com o metodo
c                 de euler amortecido
                  omega1 = omega1 - (g/r)*dsin(theta1)*deltat - gamma*om
     1ega1*deltat + amplitude(i)*dsin(frequencia*tempo)*deltat
                  theta1 = theta1 + omega1*deltat


c                 incrementa theta2 e omega2 se acordo com o metodo
c                 de euler amortecido
                  omega2 = omega2 - (g/r)*dsin(theta2)*deltat -
```

```
    gamma*om
      2ega2*deltat + amplitude(i)*dsin(frequencia*tempo)*deltat
                    theta2 = theta2 + omega2*deltat

c                   incrementa theta3 e omega3 se acordo com o metodo
c                   de euler amortecido
                    omega3 = omega3 - (g/r)*dsin(theta3)*deltat -
    gamma*om
      2ega3*deltat + amplitude(i)*dsin(frequencia*tempo)*deltat
                    theta3 = theta3 + omega2*deltat

c                   escreve o omega(theta) atual, no arquivo - para
    cada
c                   theta inicial
                    write(i,*)theta1,omega1
                    write(i+2,*)theta2,omega2
                    write(i+4,*)theta3,omega3

              end do

        end do

c       fecha os arquivos utilizados
        close(1)
        close(2)
        close(3)
        close(4)
        close(5)
        close(6)

        end program
```

Descrição:

Utilizando gamma = 0.5, são abertos 4 arquivos para salvar gráfico $\omega(\theta)$ para cada θ inicial e para cada amplitude F_0 de F.

Analogamente ao feito no item anterior, faz-se as mesmas simulações - mas ao fim de cada interação escreve theta,omega para cada theta inicial no arquivo correspondente.

Ao fim, fecha-se todos os arquvios.

Resultados:

$F_0 = 0.5$

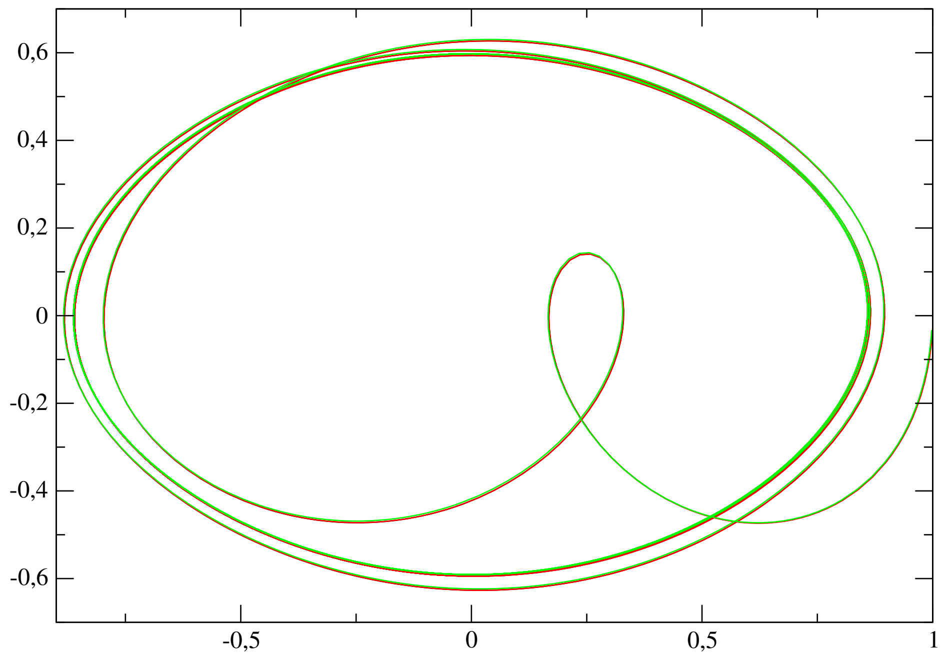

Gráfico: ω X θ
em preto: $\theta = 1$
em vermelho: $\theta = 1.001$
em verde: $\theta = 0.999$

É notório que os gráficos - para cada theta - se sobrepoem, o que já era esperado para um movimento periódico.

$F_0 = 1.2$

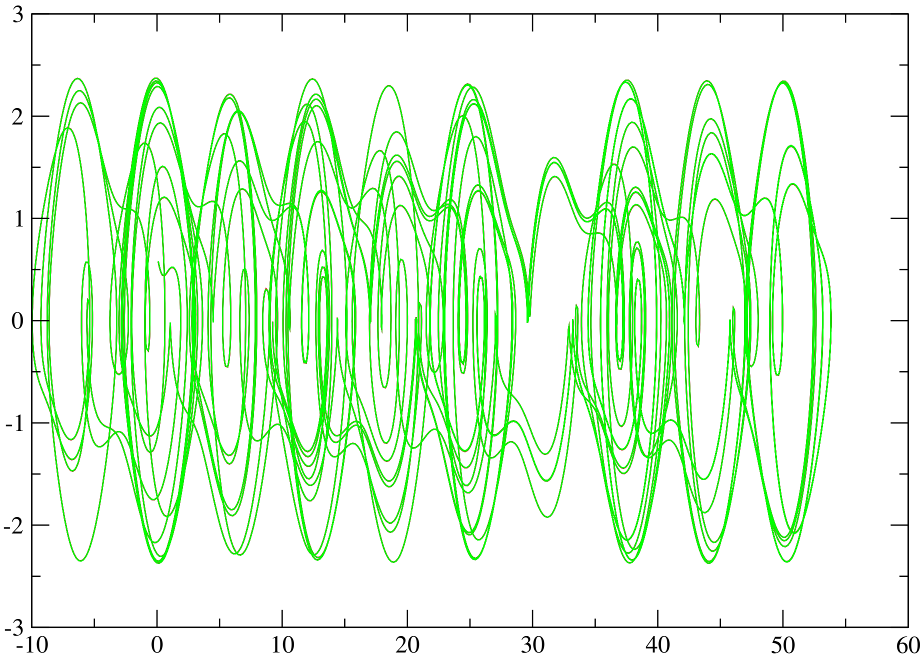

Gráfico: ω X θ
em preto: $\theta = 1$
em vermelho: $\theta = 1.001$
em verde: $\theta = 0.999$

Apesar do movimento para esta amplitude ser caótico, percebe-se - que mesmo variando as condições iniciais - o caos apresenta um padrão notável, uma vez, que existem regiões do gráfico que sequer foram visitadas.

5 Tarefa E

Tarefa: Escrever um código em FORTRAN77 que através de alguns valores de θ inicial que difiram por pouco, plote um gráfico ω X θ, contendo cada θ incial, (tanto para $F_0 = 0.5$ quanto para $F_0 = 1.2$) plotando apenas os valores que satisfazem Ωt = nπ, ou seja, que estão na seção de Poincaré.

Código escrito:

```
      program secaodepoincare

      implicit real*8(a-h,o-z)
      dimension amplitude(2)

c     define as amplitudes da forca
```

```
      amplitude(1) = 0.5d0
      amplitude(2) = 1.5d0

c     define o valor de pi
      pi = 4.0d0*datan(1.0d0)

c     define o valores da gravidade, comprimento e massa
c     referentes ao pendulo
      g = 9.8d0
      r = 9.8d0
      am = 1.0d0

c     define a constante de amortecimento e a frequencia da forca
      gamma = 0.5d0
      frequencia = 2.0d0/3.0d0

c     inicia o valor de theta e omega de acordo com a
c     solucao analitica
      theta1 = 1.0d0*pi/6.0d0
      omega1 = 0.0d0
      theta2 = 1.001d0*pi/6.0d0
      omega2 = 0.0d0
      theta3 = 0.999d0*pi/6.0d0
      omega3 = 0.0d0

c     defini o "tempo" de analise, qual o espacamento de "tempo"
c     entre as incrementacoes em theta e omega
      tempomax = 8000.0d0
      deltat = 0.04d0

c     abre os arquivos onde serao salvas as informacoes
      open(unit=1,file="amplitude0.5-1")
      open(unit=3,file="amplitude0.5-2")
      open(unit=5,file="amplitude0.5-3")
      open(unit=2,file="amplitude1.2-1")
      open(unit=4,file="amplitude1.2-2")
      open(unit=6,file="amplitude1.2-3")

c     define o loop para cada amplitude
      do i=1,2

c           (re)define o tempo como 0
            tempo = 0.0d0

c           inicia o loop de oscilacao
            do while(tempo.lt.tempomax)

c                 define o tempo atual
                  tempo = tempo + deltat

c                 incrementa theta1 e omega1 se acordo com o metodo
c                 de euler amortecido
                  omega1 = omega1 - (g/r)*dsin(theta1)*deltat -
   gamma*om
     1ega1*deltat + amplitude(i)*dsin(frequencia*tempo)*deltat
                  theta1 = theta1 + omega1*deltat

c                 incrementa theta2 e omega2 se acordo com o metodo
```

```
c                     de euler amortecido
                      omega2 = omega2 - (g/r)*dsin(theta2)*deltat - gamma*om
     2ega2*deltat + amplitude(i)*dsin(frequencia*tempo)*deltat
                      theta2 = theta2 + omega2*deltat

c                     incrementa theta3 e omega3 se acordo com o metodo
c                     de euler amortecido
                      omega3 = omega3 - (g/r)*dsin(theta3)*deltat - gamma*om
     2ega3*deltat + amplitude(i)*dsin(frequencia*tempo)*deltat
                      theta3 = theta3 + omega3*deltat

c                     se a frequencia vezes o tempo for um multipli inteiro de pi:
c                     escreve o omega(theta) atual, no arquivo - para cada
c                     theta inicial
                      n = frequencia*tempo/pi
                      if(abs(tempo-(n*pi/frequencia)).lt.deltat/2.0d0) then

                            write(i,*)theta1,omega1
                            write(i+2,*)theta2,omega2
                            write(i+4,*)theta3,omega3

                      end if

                end do

          end do

c         fecha os arquivos utilizados
          close(1)
          close(2)
          close(3)
          close(4)
          close(5)
          close(6)

          end program
```

Descrição:

Fazendo a mesma simulação do item D, defini-se n como a parte inteira da divisão de Ωtempo por π. Escrevendo os valores no arquivo, somente de a condição, módulo de (tempo - $n\pi/\Omega$) menor que deltat/2, for satisfeita.

Resultados:

$F_0 = 0.5$

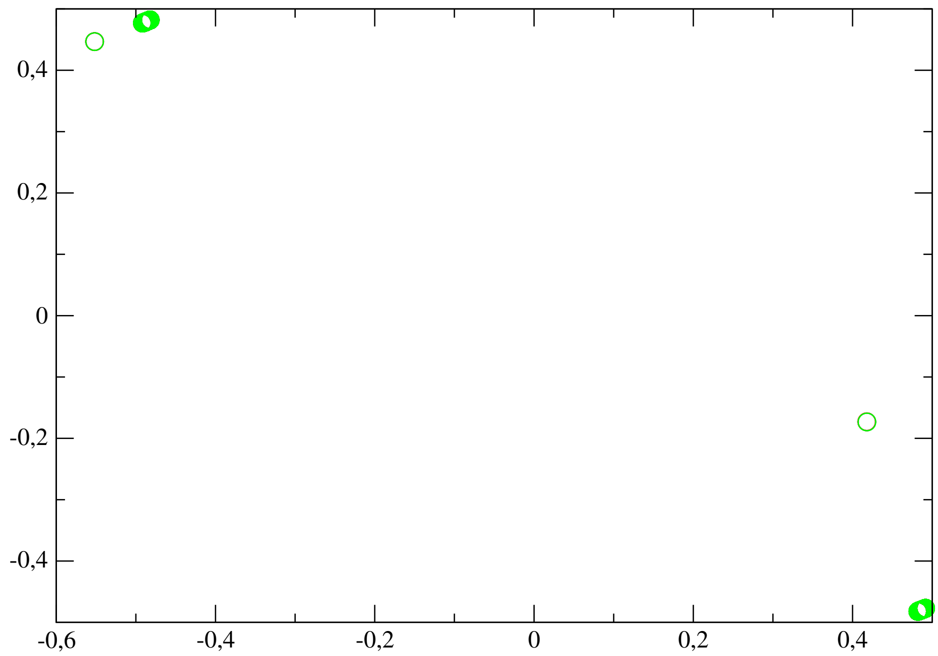

Gráfico: ω X θ
em preto: $\theta = 1$
em vermelho: $\theta = 1.001$
em verde: $\theta = 0.999$

É notório que os gráficos - para cada theta - se sobrepoem. Além disso, vê-se apenas aparições pontuais, concordando com a análise anterior de que o movimento é periódico

$F_0 = 1.2$

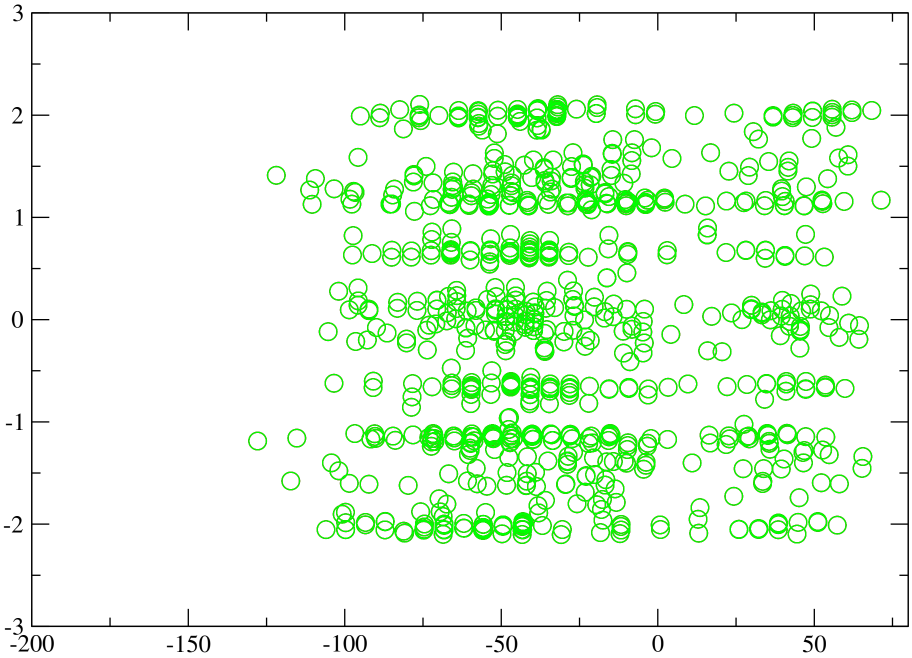

Gráfico: ω X θ
em preto: $\theta = 1$
em vermelho: $\theta = 1.001$
em verde: $\theta = 0.999$

É notório, que - apesar do movimento ser caótico - há um claro padrão - que se repete para cada theta - conhecido como o "R.G."do caos, ou então, o seu atrator estranho.