Universidade de São Paulo

Instituto de Física de São Carlos

Projeto 4

Pedro Calligaris Delbem 5255417

Professor: Francisco Alcaraz

Sumário

1	Tarefa A	2
2	Tarefa B	9
	2.1 B1	9
	2.2 B2	14
	2.3 B3	17
	2.4 B4	19
3	Tarefa C	27
4	Tarefa D	33
5	Tarefa E	37

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
2
        implicit real*8(a-h,o-z)
3
5
        define o valor de pi
        pi = 4.0d0*datan(1.0d0)
6
        define o valores da gravidade, comprimento e massa
9 C
        referentes ao pendulo
        g = 9.8d0
        r = 9.8d0
11
        am = 1.0d0
12
13
        inicia o valor de theta e omega
14 C
        theta = pi/6.0d0
        omega = 0.0d0
16
17
        inicia o valor de theta analitico
18 C
        theta0 = pi/6.0d0
20
        defini o "tempo" de analise, qual o espacamento de "tempo"
21 C
        entre as incrementacoes em theta e omega e o tempo inical
22 C
        tempomax = 80.0d0
        deltat = 0.04d0
24
        tempo = 0.0d0
25
26
        abre os arquivos onde serao salvas as informacoes
        open(unit=1,file="euler")
2.8
        open(unit=2,file="energia")
2.9
        open(unit=3,file="analitico")
        open(unit=4,file="energia-analitica")
32
        inicia o loop de oscilacao
33 C
        do while(tempo.lt.tempomax)
35
36 C
               define o tempo atual
               tempo = tempo + deltat
37
               salva o theta antigo, antes de altera-lo
39 C
               thetaant = theta
40
41
               incrementa theta e omega se acordo com o metodo de
     euler
               theta = theta + omega*deltat
43
               omega = omega - (g/r)*thetaant*deltat
44
45
46 C
               calcula a energia
```

```
energia = r*am*((omega**2)*r)/2.0d0 + g*(1-dcos(theta)
47
     )))
48
               escreve o theta(tempo) atual no arquivo e se theta
49 C
     passar,
               em modulo, de 2pi - faz a carrecao adequada
50
               if (abs(theta).ge.2.0d0*pi) then
                      write(1,*)tempo, mod(theta,2.0d0*pi)
52
               else
                      write(1,*)tempo,theta
54
               end if
55
56
               escreve o energia(tempo) atual no arquivo
               write(2,*)tempo,energia
58
59
               escreve o theta(tempo) analiticoatual, no arquivo
60
               thetaanalitico = theta0*dcos(dsqrt(g/r)*tempo)
61
               write (3,*) tempo, thetaanalitico
62
63
               calcula a energiaanalitica
64 C
               omegaanalitico = -dsqrt(g/r)*theta0*dsin(dsqrt(g/r)*
65
     tempo)
               energiaanalitica = r*am*((omegaanalitico**2)*r)/2.0d0
66
       1(1-dcos(thetaanalitico)) )
67
68
               escreve a energiaanalitica(tempo) atual no arquivo
69
               write(4,*)tempo,energiaanalitica
71
        end do
72
73
        fecha os arquivos utilizados
74
75
        close(1)
        close(2)
76
        close(3)
        close(4)
79
        end program
80
```

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)

define o valor de pi
```

```
pi = 4.0d0*datan(1.0d0)
 6
                      define o valores da gravidade, comprimento e massa
 8 C
 9 C
                      referentes ao pendulo
                      g = 9.8d0
                      r = 9.8d0
11
                      am = 1.0d0
12
13
                      inicia o valor de theta e omega
14 C
                      theta = pi/6.0d0
15
                      omega = 0.0d0
16
17
                      defini o "tempo" de analise, qual o espacamento de "tempo"
18 C
19
                      entre as incrementacoes em theta e omega e o tempo inical
                      tempomax = 80.0d0
20
                      deltat = 0.04d0
21
                      tempo = 0.0d0
22
23
24 C
                      abre os arquivos onde serao salvas as informacoes
                      open(unit=1,file="euler-cromer")
25
                      open(unit=2,file="energia-conservada")
27
                      inicia o loop de oscilacao
28 C
29
                      do while(tempo.lt.tempomax)
                                       define o tempo atual
31 C
                                       tempo = tempo + deltat
32
33
34 C
                                       incrementa theta e omega se acordo com o metodo de
               euler
                                       omega = omega - (g/r)*theta*deltat
35
                                       theta = theta + omega*deltat
36
37
                                       calcula a energia
38 C
                                       energia = r*am*( ((omega**2)*r)/2.0d0 + g*(1-dcos(theta)) + g*(1
39
               ))))
40
                                       escreve o theta(tempo) atual no arquivo e se theta
41 C
               passar,
                                       em modulo, de 2pi - faz a carrecao adequada
42 C
                                       if (abs(theta).ge.2.0d0*pi) then
43
                                                        write(1,*)tempo,mod(theta,2.0d0*pi)
44
45
                                       else
                                                        write(1,*)tempo,theta
                                       end if
47
48
                                       escreve o energia(tempo) atual no arquivo
49 C
                                       write(2,*)tempo,energia
50
51
                      end do
52
                      fecha os arquivos utilizados
                       close(1)
55
                      close(2)
56
57
                   end program
```

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

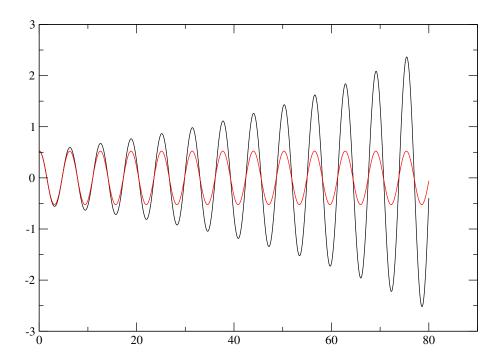


Gráfico: theta X tempo em vermelho: solução analítica em preto: método de euler

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

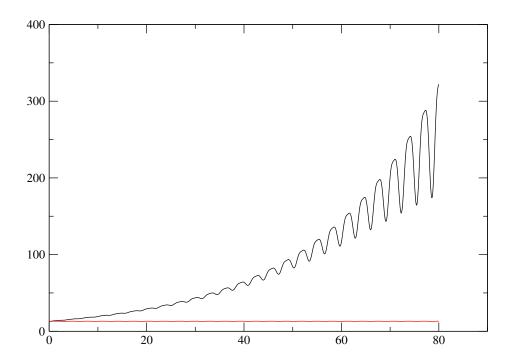


Gráfico: energia X tempo em vermelho: solução analítica em preto: método de euler

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

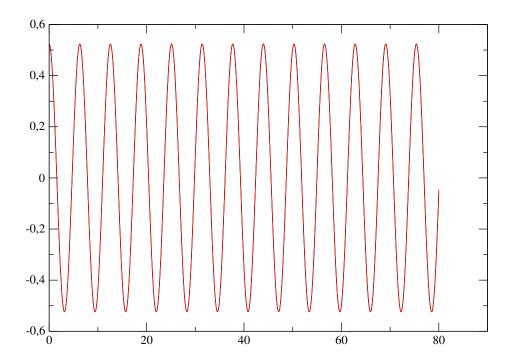


Gráfico: theta X tempo em vermelho: solução analítica em preto: método de euler

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

Analisando, então, a energia

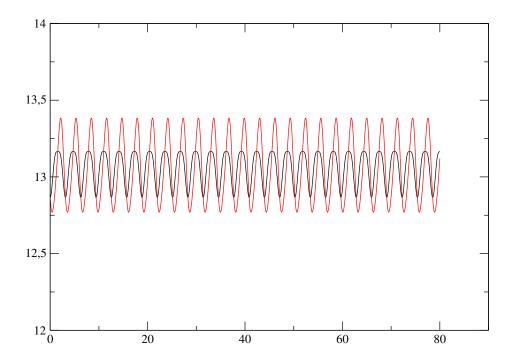


Gráfico: energia X tempo em preto: solução analítica em vermelho: método de euler

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

```
2
        implicit real*8(a-h,o-z)
4
        define o valor de pi
5 C
        pi = 4.0d0*datan(1.0d0)
6
        define o valores da gravidade, comprimento e massa
8 C
        referentes ao pendulo
        g = 9.8d0
        r = 9.8d0
11
        am = 1.0d0
12
13
        defini qual o espacamento de "tempo" entre as
14 C
15 C
        incrementacoes em theta
        deltat = 0.04d0
16
17
18 C
        abre os arquivos onde serao salvas as informacoes
        open(unit=1,file="periodo")
19
        open(unit=2,file="periodo-analitico")
20
21
22 C
        inicia o loop para thetas diferentes
        do i=1,30
23
24
25 C
               inicia o valor de theta e omega
               theta = 0.1d0*i
               theta0 = theta
27
               omega = 0.0d0
28
29
               (re)inicia o tempo e o pcontrolador
               tempo = 0.0d0
31
               pcontrolador = 0
32
33
               inicia o loop de oscilacao ate que o pcontrolador
35 C
               seja igual a 100
               do while (pcontrolador.lt.100)
36
                      salva o valor de theta antes de altera-lo
38 C
                     omegaant = omega
39
40
41 C
                     define o tempo atual
                     tempo = tempo + deltat
42
43
44 C
                     incrementa theta e omega se acordo com o metodo
     de euler
                      omega = omega - (g/r)*dsin(theta)*deltat
45
                     theta = theta + omega*deltat
46
47
                     incrementa um em pcontrolador se a velocidade
     mudar
                      if (omega*omegaant.lt.0.0d0) then
49
                            pcontrolador = pcontrolador + 1
50
                      end if
               end do
53
54
55 C
               define o peiodo como tempo/50, pois ocorrerao 50
     oscilacoes
               tempo = tempo/50.d0
```

```
57
                escreve o theta(tempo) atual no arquivo e se theta
      passar,
                em modulo, de 2pi - faz a carrecao adequada
59 C
                if(abs(theta).ge.2.0d0*pi) then
60
                      write(1,*)tempo,mod(theta0,2.0d0*pi)
61
                else
62
                      write(1,*)tempo,theta0
63
                end if
                define o epson como 1% de theta0
66 C
                epson = theta0*0.01d0
67
69
                define o valor inicial de h
                h = (theta0-epson)/1000000.0d0
70
               hi = h
71
                (re)inicia o periodo
73 C
                periodo = 0.0d0
74
75
                define o do pra somar os valores da integral
               do while(h.le.(theta0-epson))
77
78
79
                      valor = b(h-hi,theta0,hi)
                      periodo = periodo + valor
81
                      h = h + 4*hi
82
                end do
85
                calcula o periodo
86 C
                periodo = 2.0d0*dsqrt(2.0d0*r/g)*(periodo+2.0d0*dsqrt(
87
      epson/
        1dsin(theta0)))
88
89
                escreve o periodo(theta) no arquivo
                write (2,*) periodo, theta0
91
92
         end do
03
         fecha os arquivos utilizados
95
         close(1)
96
         close(2)
97
         end program
99
100
         define a integral que define o periodo
101
         real *8 function f(theta, theta0, h)
         implicit real*8 (a-h,o-z)
         f = 1.0d0/dsqrt(dcos(theta+h)-dcos(theta0))
105
106
         end function
107
108
         define a regra de Boole
110
         real *8 function b(x,x0,h)
         implicit real*8 (a-h,o-z)
111
```

```
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))

115

end function
```

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

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

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

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

$$N = \sqrt{\frac{2r}{g}} \int_0^{\theta_0 \epsilon} \frac{d\theta}{\sqrt{\cos(\theta) - \cos(\theta_0)}}$$
 (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

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

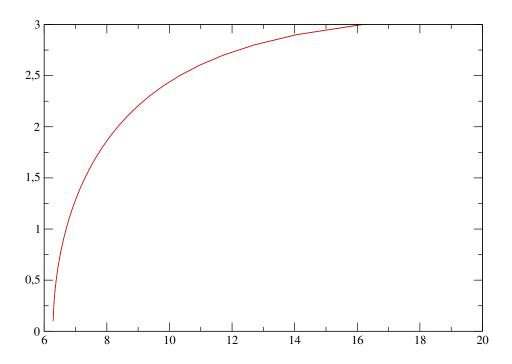


Gráfico: período X tempo em preto: euler-cromer em vermelho: integral elíptica

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)

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

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

```
g = 9.8d0
10
        r = 9.8d0
11
        am = 1.0d0
13
        defini qual o espacamento de "tempo" entre as
14 C
        incrementacoes em theta
15
        deltat = 0.04d0
16
17
        abre os arquivos onde serao salvas as informacoes
        open(unit=1,file="periodo")
        open(unit=2,file="periodo-analitico")
20
21
        inicia o loop para thetas diferentes
22 C
23
        do i=1,20
24
               inicia o valor de theta e omega
25 C
               theta = 0.1d0*i
               theta0 = theta
27
               omega = 0.0d0
2.8
29
               (re)inicia o tempo e o pcontrolador
               tempo = 0.0d0
31
               pcontrolador = 0
32
33
               inicia o loop de oscilacao ate que o pcontrolador
34 C
35 C
               seja igual a 100
               do while(pcontrolador.lt.100)
36
37
                      salva o valor de theta antes de altera-lo
                     omegaant = omega
39
40
                     define o tempo atual
41 C
42
                     tempo = tempo + deltat
43
                     incrementa theta e omega se acordo com o metodo
44 C
     de euler
                     omega = omega - (g/r)*dsin(theta)*deltat
45
                     theta = theta + omega*deltat
46
47
                     incrementa um em pcontrolador se a velocidade
     mudar
                      if (omega*omegaant.lt.0.0d0) then
49
                            pcontrolador = pcontrolador + 1
50
                      end if
               end do
53
54
               define o peiodo como tempo/50, pois ocorrerao 50
     oscilacoes
               tempo = tempo/50.d0
56
57
               escreve o theta(tempo) atual no arquivo e se theta
58
     passar,
               em modulo, de 2pi - faz a carrecao adequada
59 C
               if(abs(theta).ge.2.0d0*pi) then
60
61
                     write(1,*)tempo,mod(theta0,2.0d0*pi)
               else
62
                     write(1,*)tempo,theta0
63
```

```
end if
64
65
                define o valor inicial de h
66 C
               h = (theta0-epson)/12.0d0
67
69
                (re)calcula o periodo
70 C
                periodo = 2.0d0*pi*dsqrt(r/g)*(1+(theta0**2.0d0)/16.0d0
71
      )
72
                write(2,*)periodo,theta0
73
         end do
76
         fecha os arquivos utilizados
77
         close(1)
78
         close(2)
79
80
         end program
```

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

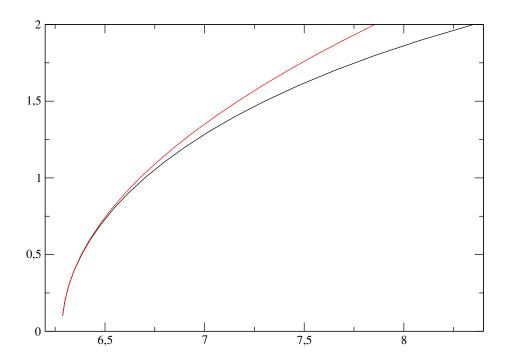


Gráfico: período X tempo em preto: euler-cromer em vermelho: equação(19)

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)

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
12
13
        define a constante de amortecimento
14 C
        gamma = 0.5d0
15
        inicia o valor de theta e omega
17 C
        theta = pi/6.0d0
18
        omega = 0.0d0
19
        defini o "tempo" de analise, qual o espacamento de "tempo"
        entre as incrementacoes em theta e omega e o tempo inical
22 C
        tempomax = 80.0d0
23
        deltat = 0.04d0
        tempo = 0.0d0
26
        abre os arquivos onde serao salvas as informacoes
27 C
        open(unit=1,file="amortecido")
29
30 C
        inicia o loop de oscilacao
        do while(tempo.lt.tempomax)
31
32
33 C
               define o tempo atual
               tempo = tempo + deltat
34
35
               incrementa theta e omega se acordo com o metodo
               de euler amortecido
37 C
               omega = omega - (g/r)*theta*deltat - gamma*omega*deltat
38
               theta = theta + omega*deltat
               escreve o theta(tempo) atual no arquivo e se theta
41 C
     passar,
               em modulo, de 2pi - faz a carrecao adequada
42 C
43
               if (abs (theta).ge.2.0d0*pi) then
                     write(1,*)tempo,mod(theta,2.0d0*pi)
44
               else
                     write (1,*) tempo, theta
               end if
47
48
        end do
49
        fecha os arquivos utilizados
51 C
        close(1)
53
        end program
```

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:

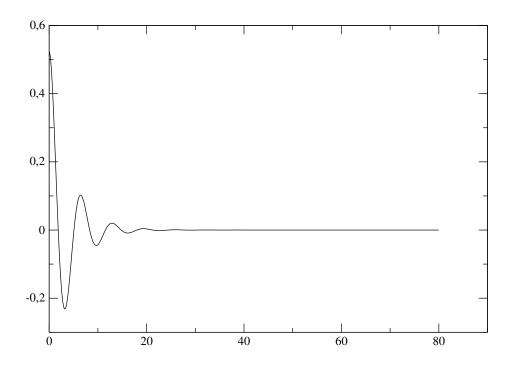


Gráfico: período X tempo em preto: euler-cromer em vermelho: equação(19)

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
8
9
        amplitude(3) = 1.2d0
        define o valor de pi
11 C
        pi = 4.0d0*datan(1.0d0)
12
13
        define o valores da gravidade, comprimento e massa
14 C
        referentes ao pendulo
15 C
        g = 9.8d0
        r = 9.8d0
17
        am = 1.0d0
18
19
        define a constante de amortecimento e a frequencia da forca
20 C
21
        gamma = 0.05d0
        frequencia = 2.0d0/3.0d0
22
23
        inicia o valor de theta e omega
24 C
        theta = pi/6.0d0
25
        omega = 0.0d0
26
27
        defini o "tempo" de analise, qual o espacamento de "tempo"
        entre as incrementacoes em theta e omega
29
        tempomax = 100.0d0
30
        deltat = 0.04d0
31
32
33 C
        abre os arquivos onde serao salvas as informacoes
        open(unit=1,file="theta-livre")
34
        open(unit=4,file="omega-livre")
        open(unit=2,file="theta-forcado0.5")
        open(unit=5,file="omega-forcado0.5")
37
        open(unit=3,file="theta-forcado1.2")
38
        open(unit=6,file="omega-forcado1.2")
39
40
41 C
        define o loop para cada amplitude
        doi=1,3
42
               (re)define o tempo como 0
44
               tempo = 0.0d0
45
46
               inicia o loop de oscilacao
47 C
               do while(tempo.lt.tempomax)
48
49
50 C
                      define o tempo atual
                      tempo = tempo + deltat
52
                      incrementa theta e omega se acordo com o metodo
53 C
                     de euler amortecido
54 C
                      omega = omega - (g/r)*dsin(theta)*deltat - gamma*
     omega
       4*deltat + amplitude(i)*dsin(frequencia*tempo)*deltat
56
                     theta = theta + omega*deltat
57
58
                      escreve o theta(tempo) atual no arquivo e se
59 C
     theta passar,
                     em modulo, de 2pi - faz a carrecao adequada
60 C
61
                      if(abs(theta).ge.2.0d0*pi) then
                            write(i,*)tempo,mod(theta,2.0d0*pi)
62
                      else
63
```

```
write(i,*)tempo,theta
64
                       end if
65
                       escreve o omega (theta) atual no arquivo
66 C
                              write(i+3,*)tempo,omega
67
                end do
69
70
         end do
71
         fecha os arquivos utilizados
73
         close(1)
74
         close(2)
         close(3)
         close(4)
77
         close(5)
         close(6)
         end program
```

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:

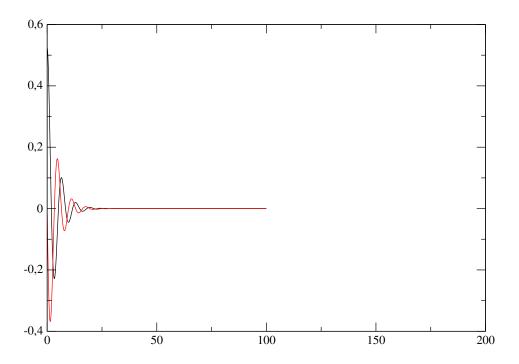


Gráfico: $F_0 = 0$ em preto: theta X tempo em vermelho: omega X tempo

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

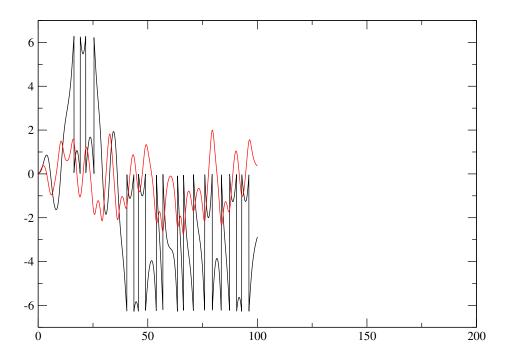


Gráfico: $F_0 = 0.5$ em preto: theta X tempo em vermelho: omega X tempo

 $\acute{\rm E}$ notório que este movimento não é periódico, sendo entendido como caótico.

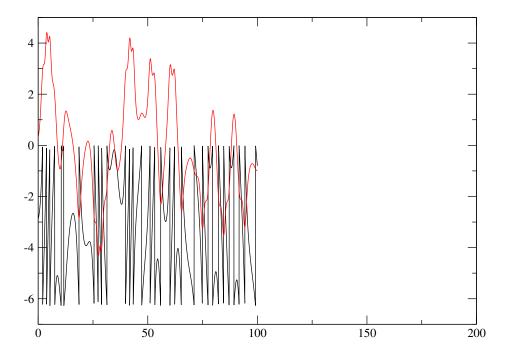


Gráfico: $F_0 = 1.2$ em preto: theta X tempo em vermelho: omega X tempo

 $\acute{\rm E}$ notório que este movimento não é periódico, sendo entendido como caótico.

Gamma = 0.5:

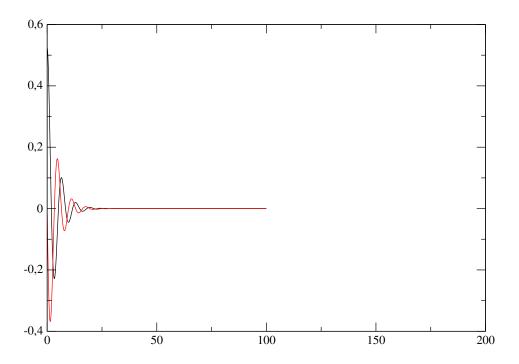


Gráfico: $F_0 = 0$ em preto: theta X tempo em vermelho: omega X tempo

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

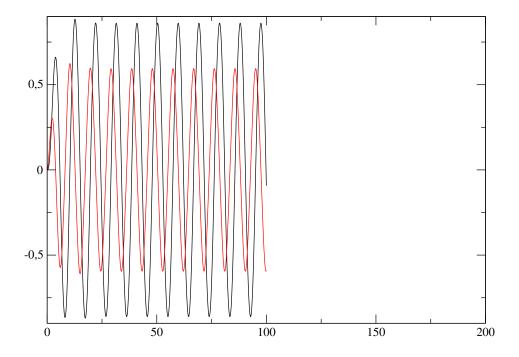


Gráfico: $F_0 = 0.5$ em preto: theta X tempo em vermelho: omega X tempo

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 .

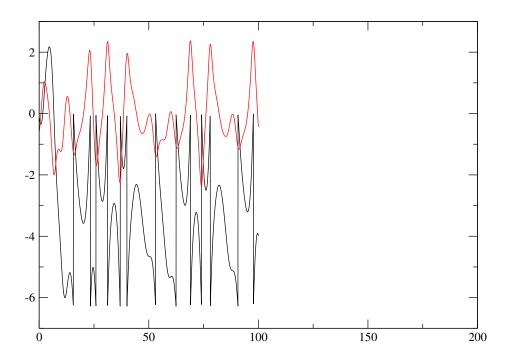


Gráfico: $F_0 = 1.2$ em preto: theta X tempo em vermelho: omega X tempo

 $\acute{\rm E}$ 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
```

```
10 C
        define o valor de pi
        pi = 4.0d0*datan(1.0d0)
11
12
        define o valores da gravidade, comprimento e massa
13 C
14 C
        referentes ao pendulo
        g = 9.8d0
        r = 9.8d0
16
        am = 1.0d0
17
        define a constante de amortecimento e a frequencia da forca
19 C
        gamma = 0.05d0
20
        frequencia = 2.0d0/3.0d0
21
22
23 C
        inicia o valor de theta e omega de acordo com a
        solucao analitica
24 C
        theta1 = 1.0d0
25
        omega1 = 0.0d0
        theta2 = 1.001d0
27
        omega2 = 0.0d0
2.8
29
        defini o "tempo" de analise, qual o espacamento de "tempo"
30 C
        entre as incrementacoes em theta e omega
31 C
        tempomax = 40.0d0
32
33
        deltat = 0.04d0
        abre os arquivos onde serao salvas as informacoes
35 C
        open(unit=1,file="amplitude0.5")
36
        open(unit=2,file="amplitude1.2")
        define o loop para cada amplitude
39 C
        doi=1,2
40
41
42 C
               (re)define o tempo como 0
               tempo = 0.0d0
43
44
               inicia o loop de oscilacao
45 C
               do while(tempo.lt.tempomax)
46
47
48 C
                     define o tempo atual
                     tempo = tempo + deltat
50
                     incrementa theta1 e omega1 se acordo com o metodo
51 C
52 C
                     de euler amortecido
                     omega1 = omega1 - (g/r)*dsin(theta1)*deltat -
53
     gamma*om
       lega1*deltat + amplitude(i)*dsin(frequencia*tempo)*deltat
55
                     theta1 = theta1 + omega1*deltat
                     incrementa theta2 e omega2 se acordo com o metodo
57 C
                     de euler amortecido
58 C
                     omega2 = omega2 - (g/r)*dsin(theta2)*deltat -
59
       2ega2*deltat + amplitude(i)*dsin(frequencia*tempo)*deltat
60
                     theta2 = theta2 + omega2*deltat
61
62
63 C
                     escreve o theta(tempo), com escala semi-
     logaritmica,
                     atual, no arquivo
64 C
```

```
write(i,*)tempo,dlog(abs(theta1-theta2))
65
66
                end do
67
68
         end do
70
         fecha os arquivos utilizados
71 C
         close(1)
72
         close(2)
73
74
         end program
```

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:

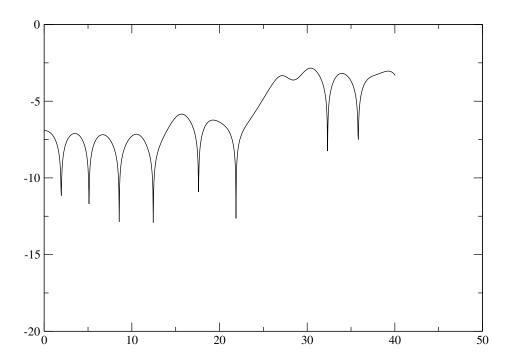


Gráfico: $F_0 = 0.5$ em preto: $\ln(\Delta theta)$ X tempo

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.

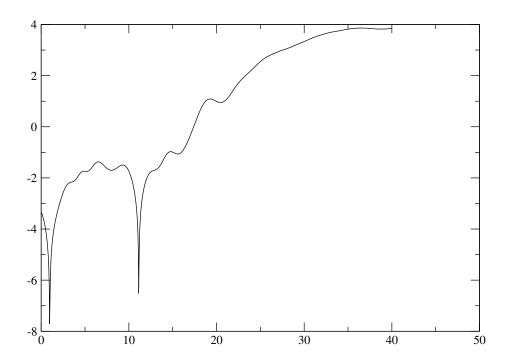


Gráfico: $F_0 = 1.2$ em preto: $\ln(\Delta theta)$ X tempo

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:

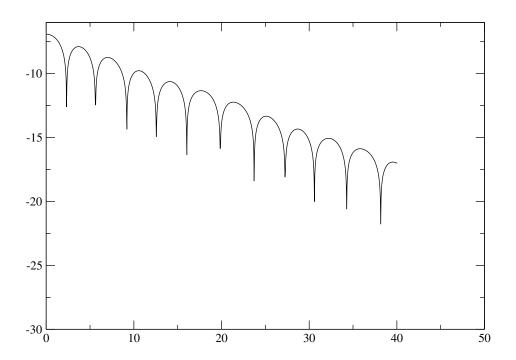


Gráfico: $F_0 = 0.5$ em preto: $\ln(\Delta theta)$ X tempo

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.

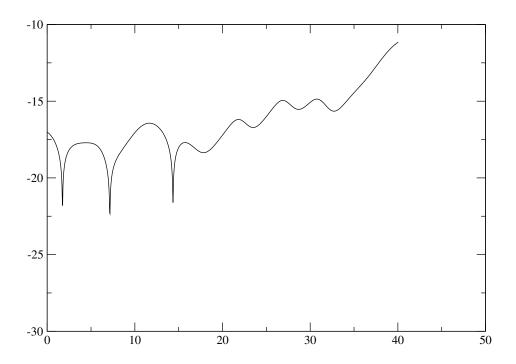


Gráfico: $F_0 = 1.2$ em preto: $\ln(\Delta theta) X$ tempo

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
```

```
10 C
        define o valor de pi
        pi = 4.0d0*datan(1.0d0)
11
12
        define o valores da gravidade, comprimento e massa
13 C
14 C
        referentes ao pendulo
        g = 9.8d0
        r = 9.8d0
16
        am = 1.0d0
17
        define a constante de amortecimento e a frequencia da forca
19 C
        gamma = 0.5d0
20
        frequencia = 2.0d0/3.0d0
21
22
23 C
        inicia o valor de theta e omega de acordo com a
        solucao analitica
24 C
        theta1 = 1.0d0
25
        omega1 = 0.0d0
        theta2 = 1.001d0
27
        omega2 = 0.0d0
29
        theta3 = 0.999d0
        omega3 = 0.0d0
31
        defini o "tempo" de analise, qual o espacamento de "tempo"
32 C
33 C
        entre as incrementacoes em theta e omega
        tempomax = 1000.0d0
        deltat = 0.04d0
35
36
        abre os arquivos onde serao salvas as informacoes
37
        open(unit=1,file="amplitude0.5-1")
        open(unit=3,file="amplitude0.5-2")
39
        open(unit=5,file="amplitude0.5-3")
40
        open(unit=2,file="amplitude1.2-1")
41
42
        open(unit=4,file="amplitude1.2-2")
        open(unit=6,file="amplitude1.2-3")
43
44
45 C
        define o loop para cada amplitude
        doi=1,2
46
47
48 C
               (re)define o tempo como 0
               tempo = 0.0d0
50
               inicia o loop de oscilacao
51 C
               do while(tempo.lt.tempomax)
54 C
                     define o tempo atual
                     tempo = tempo + deltat
55
56
                     incrementa theta1 e omega1 se acordo com o metodo
57 C
                     de euler amortecido
58 C
                     omega1 = omega1 - (g/r)*dsin(theta1)*deltat -
59
     gamma*om
       1ega1*deltat + amplitude(i)*dsin(frequencia*tempo)*deltat
                     theta1 = theta1 + omega1*deltat
61
62
63
                     incrementa theta2 e omega2 se acordo com o metodo
65 C
                     de euler amortecido
                     omega2 = omega2 - (g/r)*dsin(theta2)*deltat -
66
```

```
gamma*om
        2ega2*deltat + amplitude(i)*dsin(frequencia*tempo)*deltat
67
                      theta2 = theta2 + omega2*deltat
68
69
                      incrementa theta3 e omega3 se acordo com o metodo
70 C
                      de euler amortecido
71
                      omega3 = omega3 - (g/r)*dsin(theta3)*deltat -
72
     gamma*om
       2ega3*deltat + amplitude(i)*dsin(frequencia*tempo)*deltat
73
                      theta3 = theta3 + omega2*deltat
74
75
                      escreve o omega(theta) atual, no arquivo - para
76 C
     cada
77
                      theta inicial
                      write(i,*)theta1,omega1
78
                      write(i+2,*)theta2,omega2
79
                      write(i+4,*)theta3,omega3
81
               end do
82
83
        end do
85
        fecha os arquivos utilizados
86 C
87
        close(1)
        close(2)
88
        close(3)
89
        close(4)
90
         close(5)
         close(6)
93
        end program
94
```

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:

 $F_0 = 0.5$

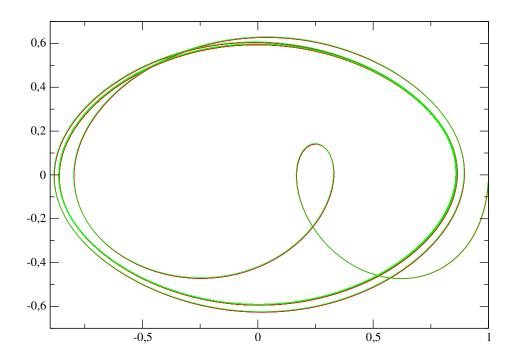


Gráfico: ω X θ em preto: $\theta = 1$ em vermelho: $\theta = 1.001$ em verde: $\theta = 0.999$

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

 $F_0 = 1.2$

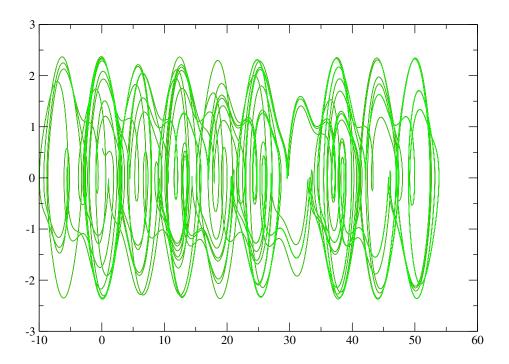


Gráfico: $\omega \times \theta$ em preto: $\theta = 1$ em vermelho: $\theta = 1.001$ em verde: $\theta = 0.999$

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 Ω t = $n\pi$, ou seja, que estão na seção de Poincaré.

Código escrito:

```
program secaodepoincare

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

define as amplitudes da forca
```

```
amplitude(1) = 0.5d0
        amplitude(2) = 1.5d0
9
        define o valor de pi
10 C
        pi = 4.0d0*datan(1.0d0)
11
12
        define o valores da gravidade, comprimento e massa
13 C
        referentes ao pendulo
14 C
        g = 9.8d0
        r = 9.8d0
        am = 1.0d0
17
18
        define a constante de amortecimento e a frequencia da forca
19 C
20
        gamma = 0.5d0
        frequencia = 2.0d0/3.0d0
21
22
        inicia o valor de theta e omega de acordo com a
23 C
24 C
        solucao analitica
        theta1 = 1.0d0*pi/6.0d0
25
        omega1 = 0.0d0
        theta2 = 1.001d0*pi/6.0d0
        omega2 = 0.0d0
28
        theta3 = 0.999d0*pi/6.0d0
29
        omega3 = 0.0d0
30
31
        defini o "tempo" de analise, qual o espacamento de "tempo"
32 C
        entre as incrementacoes em theta e omega
33 C
        tempomax = 8000.0d0
        deltat = 0.04d0
36
        abre os arquivos onde serao salvas as informacoes
37 C
        open(unit=1,file="amplitude0.5-1")
38
39
        open(unit=3,file="amplitude0.5-2")
        open(unit=5,file="amplitude0.5-3")
40
        open(unit=2,file="amplitude1.2-1")
41
        open(unit=4,file="amplitude1.2-2")
        open(unit=6,file="amplitude1.2-3")
43
44
45 C
        define o loop para cada amplitude
        do i=1,2
47
               (re)define o tempo como 0
48 C
49
               tempo = 0.0d0
50
51 C
               inicia o loop de oscilacao
               do while(tempo.lt.tempomax)
52
53
                     define o tempo atual
54 C
55
                     tempo = tempo + deltat
56
                     incrementa theta1 e omega1 se acordo com o metodo
57 C
                     de euler amortecido
58
                     omega1 = omega1 - (g/r)*dsin(theta1)*deltat -
59
     gamma*om
       lega1*deltat + amplitude(i)*dsin(frequencia*tempo)*deltat
61
                     theta1 = theta1 + omega1*deltat
62
                     incrementa theta2 e omega2 se acordo com o metodo
63 C
```

```
de euler amortecido
64 C
                      omega2 = omega2 - (g/r)*dsin(theta2)*deltat -
65
     gamma*om
       2ega2*deltat + amplitude(i)*dsin(frequencia*tempo)*deltat
66
                      theta2 = theta2 + omega2*deltat
68
                      incrementa theta3 e omega3 se acordo com o metodo
69
                      de euler amortecido
70
                      omega3 = omega3 - (g/r)*dsin(theta3)*deltat -
71
     gamma*om
       2ega3*deltat + amplitude(i)*dsin(frequencia*tempo)*deltat
72
                      theta3 = theta3 + omega3*deltat
73
75
                      se a frequencia vezes o tempo for um multipli
     inteiro de pi:
                      escreve o omega(theta) atual, no arquivo - para
76 C
     cada
                      theta inicial
77 C
                      n = frequencia*tempo/pi
78
                      if(abs(tempo-(n*pi/frequencia)).lt.deltat/2.0d0)
      then
80
                            write(i,*)theta1,omega1
81
82
                            write(i+2,*)theta2,omega2
                            write(i+4,*)theta3,omega3
83
84
                      end if
85
               end do
88
        end do
89
90
91 C
        fecha os arquivos utilizados
        close(1)
92
        close(2)
93
         close(3)
         close(4)
95
         close(5)
96
         close(6)
97
        end program
99
```

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

Resultados:

 $F_0 = 0.5$

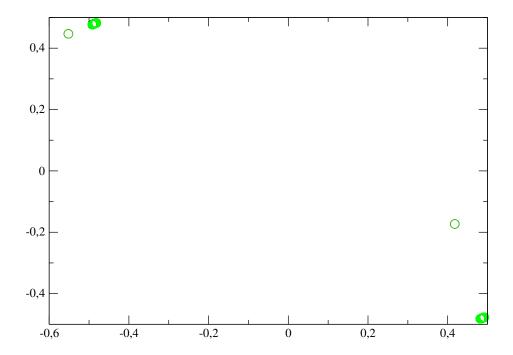


Gráfico: ω X θ em preto: $\theta = 1$ em vermelho: $\theta = 1.001$ em verde: $\theta = 0.999$

 $\acute{\rm E}$ 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

 $F_0 = 1.2$

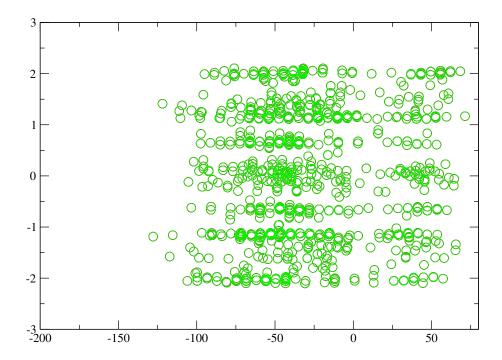


Gráfico: ω X θ em preto: $\theta = 1$ em vermelho: $\theta = 1.001$ em verde: $\theta = 0.999$

 \acute{E} 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.