



INSTITUTO DE FÍSICA DE SÃO CARLOS  
UNIVERSIDADE DE SÃO PAULO

[7600017] - INTRODUÇÃO À FÍSICA COMPUTACIONAL

---

## Projeto 4

---

**Docente:**

Francisco Castilho Alcaraz

**Aluno:**

Breno Henrique Pelegrin da Silva (13687303)

24 de novembro de 2023

## Sumário

<b>1</b>	<b>Contexto</b>	<b>3</b>
<b>2</b>	<b>Tarefa A</b>	<b>3</b>
2.1	Resultados e análise . . . . .	6
<b>3</b>	<b>Tarefa B</b>	<b>9</b>
3.1	Parte B1 e B2 . . . . .	10
3.1.1	Resultados e análise . . . . .	13
3.2	Parte B3 . . . . .	16
3.2.1	Resultados e análise . . . . .	17
3.3	Parte B4 . . . . .	18
3.3.1	Resultados e análise . . . . .	20
<b>4</b>	<b>Tarefa C</b>	<b>23</b>
4.1	Resultados e análise . . . . .	25
<b>5</b>	<b>Tarefa D</b>	<b>27</b>
5.1	Resultados e análise . . . . .	29
<b>6</b>	<b>Tarefa E</b>	<b>31</b>
6.1	Resultados e análise . . . . .	34

## Lista de Figuras

1	$\theta(t)$ para o Método de Euler . . . . .	7
2	$E(t)$ para o Método de Euler . . . . .	7
3	$\theta(t)$ para o Método de Euler-Cromer . . . . .	8
4	$E(t)$ para o Método de Euler-Cromer . . . . .	8
5	$E(t)$ para o Método de Euler e Método de Euler-Cromer . . . . .	9
6	$T(\theta_0)$ para cada método calculado. . . . .	14
7	Diferença absoluta entre $T_{sim}$ e $T_{int}$ . . . . .	15
8	Diferença absoluta entre $T_{apx}$ e $T_{int}$ . . . . .	16
9	$\theta(t)$ para $\gamma = 0.5$ e $F_0 = 0$ . . . . .	18
10	$\theta(t)$ e $\omega(t)$ para $\gamma = 0.05$ e $F_0 = 0$ . . . . .	20
11	$\theta(t)$ para $\gamma = 0.5$ e $F_0 = 0$ . . . . .	21
12	$\theta(t)$ para $\gamma = 0.05$ e $F_0 = 0.5$ . . . . .	21
13	$\theta(t)$ para $\gamma = 0.5$ e $F_0 = 0.5$ . . . . .	22
14	$\theta(t)$ para $\gamma = 0.05$ e $F_0 = 1.2$ . . . . .	22
15	$\theta(t)$ para $\gamma = 0.5$ e $F_0 = 1.2$ . . . . .	23
16	$\Delta\theta$ para $F_0 = 0.5$ (y em escala $\ln(y)$ ) . . . . .	26
17	$\Delta\theta$ para $F_0 = 1.2$ (y em escala $\ln(y)$ ) . . . . .	27
18	$\theta(\omega)$ para $F_0 = 0.5$ . . . . .	30
19	$\theta(\omega)$ para $F_0 = 1.2$ . . . . .	31
20	$\theta(\omega)$ para $F_0 = 0.5$ (seção de Poincaré) . . . . .	35
21	$\theta(\omega)$ para $F_0 = 1.2$ (seção de Poincaré) . . . . .	35

## Códigos

1	Código-fonte da tarefa A . . . . .	5
2	Código-fonte da tarefa B, parte B1-B2 . . . . .	11
3	Código-fonte da tarefa B, parte B3 . . . . .	16
4	Código-fonte da tarefa B, parte B4 . . . . .	18
5	Código-fonte da tarefa C . . . . .	24
6	Código-fonte da tarefa D . . . . .	28
7	Código-fonte da tarefa E . . . . .	32

## 1 Contexto

Nesse projeto, foram elaborados diversos programas em *Fortran 77* para explorar o movimento oscilatório de um pêndulo simples. Inicialmente, considerou-se a aproximação de pequenos ângulos, porém, posteriormente, a equação diferencial do pêndulo foi analisada sem aproximações, tornando possível explorar a evolução temporal do fenômeno em sua plenitude, bem como entender a emergência do caos e a análise deste para diferentes configurações do sistema.

Os gráficos e análises gráficas dispostos neste relatório foram feitos utilizando as bibliotecas e recursos do *Python*, como *matplotlib*, *pandas* e *numpy*. Todos os dados gerados nos programas em *Fortran 77* foram escritos no formato CSV (valores separados por vírgula), para facilitar o processamento dos dados em outros softwares.

Além disso, vale notar que, como são necessários muitos pontos para construir alguns dos gráficos expostos no relatório, foram utilizados vetores com grande tamanho (na ordem de 10.000 valores) e, portanto, é conveniente mudar o tamanho máximo da stack do compilador de Fortran 77 para algo superior à 10.000.

## 2 Tarefa A

Nessa tarefa, estudou-se o movimento de um pêndulo simples, utilizando a aproximação de ângulos pequenos. Dado um pêndulo de comprimento  $l$  e massa  $m$  que oscila formando um ângulo  $\theta$  com o eixo vertical, sua equação de movimento  $\theta(t)$  é dada pela Equação Diferencial Ordinária (EDO) exposta na equação 1.

$$ma_\theta = ml \frac{d^2\theta}{dt^2} = -mg \sin(\theta)$$

$$\frac{d^2\theta}{dt^2} = \frac{-g}{l} \sin(\theta) \quad (1)$$

Supondo que o pêndulo exerça pequenas oscilações, ou seja, que  $\theta \approx 0$ , então pode-se considerar que  $\sin(\theta) \approx \theta$  e, portanto, é possível reescrever a EDO para o caso de ângulos pequenos, como é exposto na equação 2, onde  $\omega(t) = \frac{d\theta}{dt}$ .

$$\theta \rightarrow 0 \implies \frac{d^2\theta}{dt^2} = \frac{-g}{l} \theta$$

$$\frac{d\theta}{dt} = \omega(t)$$

$$\frac{d\omega}{dt} = \frac{-g}{l} \theta(t) \quad (2)$$

A fim de analisar o comportamento de  $\theta(t)$  numericamente, pode-se utilizar o método de Euler. No método de Euler, o eixo temporal é particionado em intervalos de

tamanho  $\Delta t$ , de forma que para cada  $i$ -ésimo intervalo, o tempo  $t_{i+1}$  é dado pela equação 5.

Considerando essa discretização, pode-se aproximar o comportamento da função  $f(t)$  no intervalo  $i + 1$  por  $f(t_{i+1}) = f(t_i) + \Delta t f'(t_i)$ , ou seja, somando uma pequena variação da função no intervalo (sua derivada multiplicada pelo tamanho  $\Delta t$ ) ao valor da função no intervalo anterior. Assim, aplicando esse método à EDO do pêndulo simples com aproximação de pequenos ângulos, obtém-se as equações numéricas 3 e 3, que podem ser computadas em um programa *Fortran*.

$$w_{i+1} = w_i - \frac{g}{l} \theta_i \Delta t \quad (3)$$

$$\theta_{i+1} = \theta_i + \omega_i \Delta t \quad (4)$$

$$t_{i+1} = t_i + \Delta t, t_0 = 0 \quad (5)$$

A partir dos resultados apresentados na seção 2.1, observa-se que a amplitude de oscilação e a energia mecânica sofrem um pequeno aumento em função do tempo no método de Euler, o que representa uma descrição física errônea do fenômeno, uma vez que a energia deveria se conservar.

Para contornar esse problema, é realizada uma modificação no método de Euler, substituindo o termo  $\omega_i$  na equação 4 por  $w_{i+1}$ , sendo essa alteração conhecida como método de Euler-Cromer. As equações numéricas para o método de Euler-Cromer são expostas nas equações 6 e 7.

$$w_{i+1} = w_i - \frac{g}{l} \theta_i \Delta t \quad (6)$$

$$\theta_{i+1} = \text{mod}(\theta_i + w_{i+1} \Delta t, 2\pi), \text{ com } t = i \Delta t \quad (7)$$

$$t_{i+1} = t_i + \Delta t, \text{ com } t_0 = 0$$

A operação de divisão modular é empregada em  $\theta_{i+1}$  para que  $\theta$  esteja sempre entre 0 e  $2\pi$ . A energia mecânica do sistema pode ser calculada utilizando os valores de  $\theta_i$  para cada passo, de acordo com a equação 8.

$$E_i = T_i + U_i = \frac{1}{2} m (\omega_i l)^2 + mgl[1 - \cos(\theta_i)] \quad (8)$$

Para os cálculos dessa tarefa, foi considerado um ângulo inicial  $\theta_0 = 0.14 \text{ rad} \approx 8^\circ$ ,  $\Delta t = 10^{-2}$ ,  $g = l = 9.8$ ,  $m = 1.0$  e o número de passos simulados foi 2000 passos. O código do programa desenvolvido para esse projeto é exposto no Código 1, e a

explicação para esse código é dada logo em seguida, no verbete 1. Os resultados obtidos e a análise deles estão expostos na seção 2.1.

Código 1: Código-fonte da tarefa A

```

1      program TarefaA
2          implicit real*8 (a-h,o-z)
3          parameter(dt = 1e-2)
4          parameter(ipassos = 2000)
5          parameter(theta0 = 0.14d0)
6          parameter(g = 9.8d0)
7          parameter(al = 9.8d0)
8          parameter(m = 1.0d0)
9          dimension t(ipassos), w(ipassos), e(ipassos), theta(ipassos)
10         dimension w_ec(ipassos), e_ec(ipassos), theta_ec(ipassos)
11         pi = 4.0d0*atan(1.0d0)
12         theta(1) = theta0
13         theta_ec(1) = theta0
14         w(1) = 0.0d0
15         w_ec(1) = 0.0d0
16         e(1) = m*g*al*(1-cos(theta0))
17         e_ec(1) = m*g*al*(1-cos(theta0))
18         t(1) = 0.0d0
19         open(50, file="saida-a-13687303-theta_t.csv")
20         open(51, file="saida-a-13687303-theta_ec_t.csv")
21         open(60, file="saida-a-13687303-e_t.csv")
22         open(61, file="saida-a-13687303-e_ec_t.csv")
23         write(50,100)
24     100    format("t,theta")
25         write(51, 101)
26     101    format("t,theta_ec")
27         write(60,102)
28     102    format("t,e")
29         write(61, 103)
30     103    format("t,e_ec")
31         write(50,*) t(1), ",", theta(1)
32         write(51,*) t(1), ",", theta_ec(1)
33         write(60,*) t(1), ",", e(1)
34         write(61,*) t(1), ",", e_ec(1)
35         do i=1, ipassos-1, 1
36             t(i+1) = i*dt
37             w(i+1) = w(i) - (g/al)*theta(i)*dt
38             w_ec(i+1) = w_ec(i) - (g/al)*theta_ec(i)*dt
39             theta(i+1) = mod(theta(i) + w(i)*dt, 2.0d0*pi)
40             theta_ec(i+1) = mod(theta_ec(i) + w_ec(i+1)*dt, 2.0d0*pi)
41         c    E = U + K = mgl * (1 - cos(theta)) + 0.5 * m * (w*1)2
42             e(i+1) = m*g*al*(1-cos(theta(i+1)))+ 0.5d0*m*(w(i+1)*al)**2
43             e_ec(i+1) = m*g*al*(1-cos(theta_ec(i+1)))+0.5d0*m*(w_ec(i+1
44             &)*al)**2
45             write(50, *) t(i+1), ",", theta(i+1)
46             write(51, *) t(i+1), ",", theta_ec(i+1)
47             write(60, *) t(i+1), ",", e(i+1)
48             write(61, *) t(i+1), ",", e_ec(i+1)
49         end do
50         close(50)

```

```
51         close(51)
52         close(60)
53         close(61)
54     end program
```

#### Verbete 1: Explicação do código da tarefa A

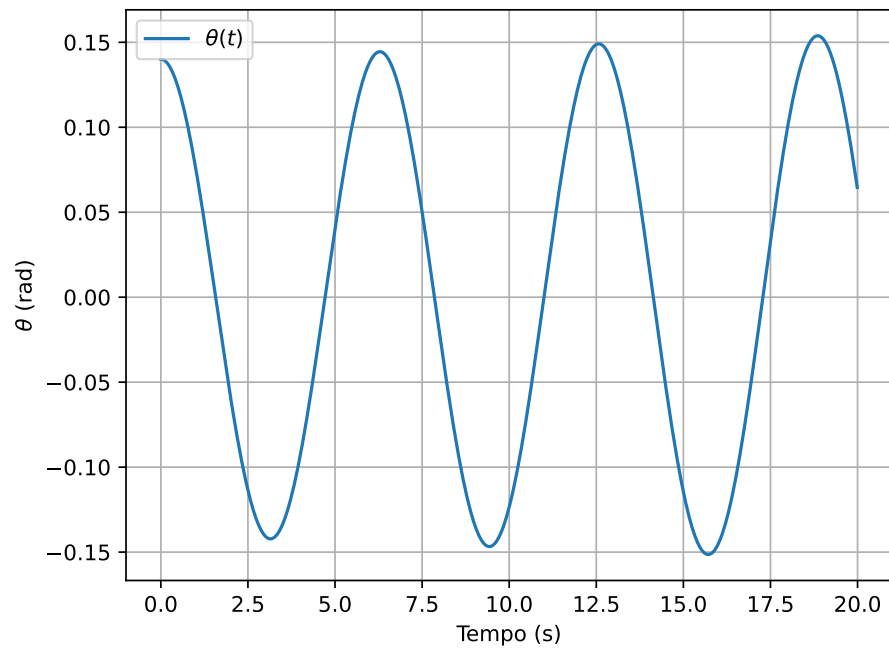
Inicialmente, define-se os parâmetros e os vetores a serem utilizados durante a simulação do pêndulo. Os vetores são inicializados com o tamanho de acordo com a quantidade  $N$  de passos. Como o pêndulo parte do repouso, tem-se que  $\omega_0 = 0$ .

Em seguida, após as definições e abertura do arquivo de saída, realiza-se um laço  $i$  de 1 a  $N - 1$ , de 1 em 1. A cada passo, são calculadas as equações numéricas para o método de Euler e o método de Euler-Cromer, assim como a energia mecânica total do sistema calculada utilizando os valores dos dois métodos.

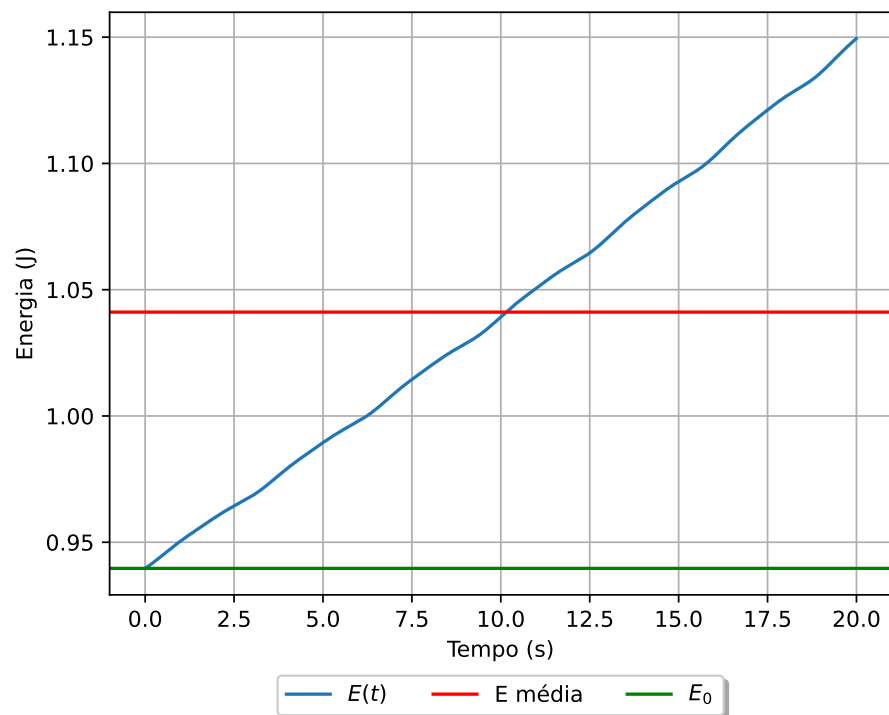
Os valores calculados para cada  $\theta_i$  e  $E_i$  em função de  $t_i$  são escritos em um arquivo para cada método numérico avaliado, para fins de comparação.

## 2.1 Resultados e análise

A partir dos resultados obtidos nessa tarefa, foram criados os gráficos de  $\theta(t)$  e  $E(t)$  para os dois métodos numéricos utilizados, de Euler e de Euler-Cromer. Os gráficos de  $\theta(t)$  e  $E(t)$  para o método de Euler são exibidos nas figuras 1 e 2, respectivamente, e os gráficos de  $\theta(t)$  e  $E(t)$  para o método de Euler-Cromer são exibidos nas figuras 3 e 4, respectivamente.

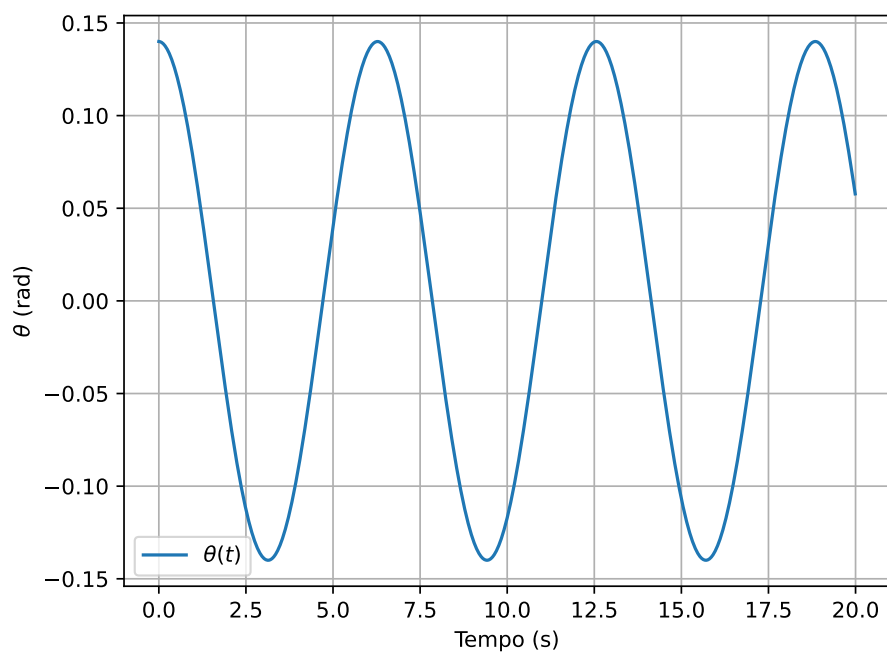
Figura 1:  $\theta(t)$  para o Método de Euler

Fonte: Autoria própria

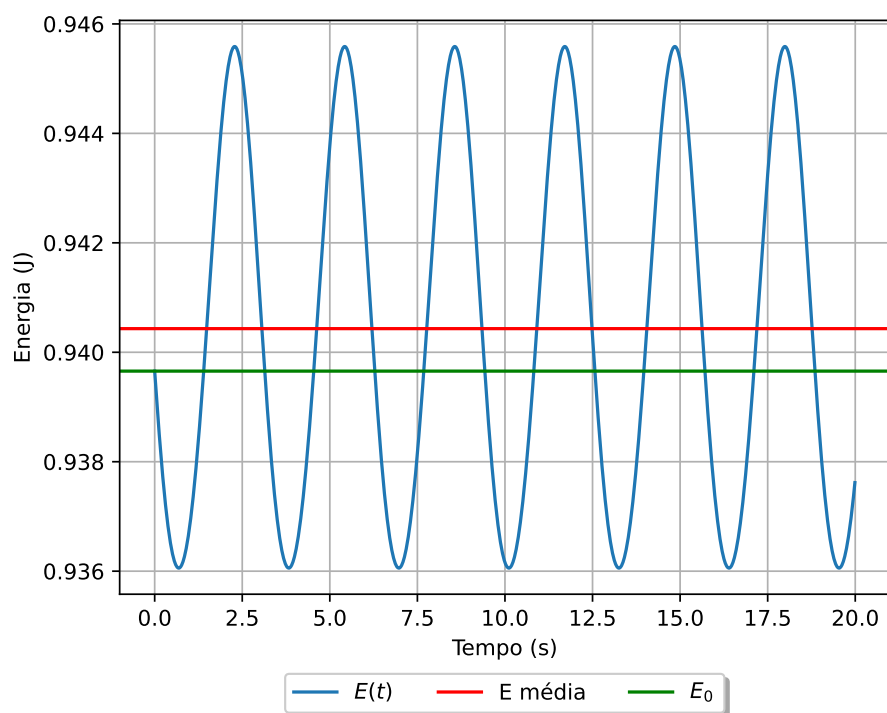
Figura 2:  $E(t)$  para o Método de Euler

Fonte: Autoria própria



Figura 3:  $\theta(t)$  para o Método de Euler-Cromer

Fonte: Autoria própria

Figura 4:  $E(t)$  para o Método de Euler-Cromer

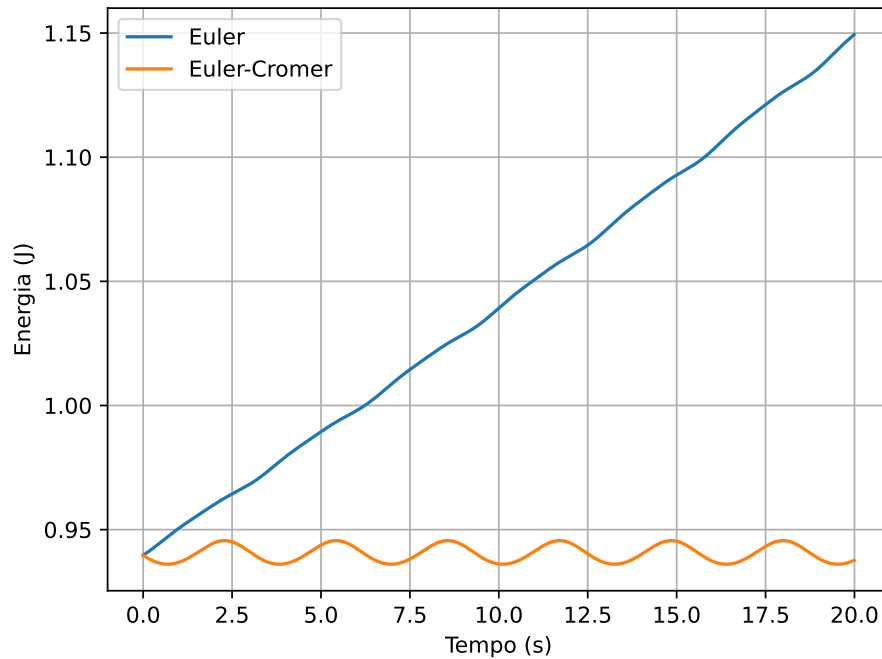
Fonte: Autoria própria

Ao observar os gráficos de  $\theta(t)$  e  $E(t)$  para os dois métodos, observa-se que, no método de Euler, a amplitude de  $\theta(t)$  e o valor de  $E(t)$  tendem a crescer com o tempo, o que representa uma inconsistência física, já que a amplitude de  $\theta(t)$  e o valor de  $E(t)$  deveriam ser aproximadamente constantes no tempo, exceto por variações de ponto flutuante.

Já nos gráficos para o método de Euler-Cromer, observa-se que a amplitude de  $\theta(t)$  permanece constante no tempo e a energia  $E(t)$  varia na escala de  $\Delta t$ , ou seja, é aproximadamente constante no tempo, tendo apenas variações de ponto flutuante. Assim, o método de Euler-Cromer descreve o fenômeno físico com maior precisão que o método de Euler.

A fim de comparar os dois métodos, foi criado um gráfico conjunto de  $E(t)$  para os dois métodos numéricos, exibido na figura 5. Ao analisar o gráfico, observa-se que, no método de Euler, a energia cresce linearmente com o tempo, enquanto no método de Euler-Cromer, a energia oscila regularmente em torno de um ponto, permanecendo aproximadamente constante no tempo.

Figura 5:  $E(t)$  para o Método de Euler e Método de Euler-Cromer



Fonte: Autoria própria

### 3 Tarefa B

Nessa tarefa, utilizando o Método de Euler-Cromer, explorou-se o fenômeno do pêndulo simples sem aproximação de ângulos pequenos, e acrescentando um termo resistivo e uma força externa na EDO, tornando a simulação mais geral e próxima da realidade. A EDO que descreve  $\theta(t)$  com os novos termos acrescentados é exibida

na equação 9, onde  $\gamma$  é o fator de resistência do meio,  $F_0$  é a amplitude da força externa e  $\Omega$  é a frequência da força externa.

$$\frac{d\omega}{dt} = -\frac{g}{l}\sin(\theta) - \gamma\frac{d\theta}{dt} + F_0\sin(\Omega t) \quad (9)$$

$$\frac{d\theta}{dt} = \omega$$

### 3.1 Parte B1 e B2

Nas partes B1 e B2, o período do movimento oscilatório foi calculado para diversos ângulos iniciais  $\theta_0$ , considerando  $F_0 = \gamma = 0$  e utilizando três métodos distintos:

1. O primeiro consiste em realizar a simulação numérica de vários pêndulos com ângulos iniciais  $\theta_0$  e determinar o período a partir da simulação.
2. O segundo consiste em resolver a EDO para encontrar o período, descrito pela integral da equação 12, que deve ser calculada numericamente considerando as singularidades da função.
3. O terceiro consiste em calcular o período através da equação aproximada 13, obtida a partir de uma expansão em série do período.

Ao discretizar a EDO da equação 9 utilizando o Método de Euler-Cromer, obtém-se as equações 10 e 11.

$$\omega_{i+1} = \omega_i + \Delta t \left[ -\frac{g}{l}\sin(\theta_i) - \gamma\omega_i + F_0\sin(\Omega t_i) \right] \quad (10)$$

$$\theta_{i+1} = \text{mod}(\theta_i + \omega_{i+1}\Delta t, 2\pi) \quad (11)$$

$$t_{i+1} = t_i + \Delta t, \text{ com } t_0 = 0$$

A integral da equação 12 pode ser obtida ao multiplicar a EDO 1 por  $\dot{\theta}$ , aplicar as condições iniciais  $\theta(t=0) = \theta_0$  e  $\dot{\theta}(t=0) = 0$ , e separar as variáveis  $\theta$  e  $t$ , como exposto abaixo.

$$\ddot{\theta}\dot{\theta} + \omega^2\sin(\theta)\dot{\theta} = 0 \implies \left(\frac{1}{2}\dot{\theta}^2\right)' - (\cos(\theta))'\omega^2 = 0$$

$$\int \left(\frac{1}{2}\dot{\theta}^2\right) - \cos(\theta)\omega^2 dt \implies \frac{1}{2}\dot{\theta}^2 - \omega^2\cos(\theta) = C$$

$$\theta(0) = \theta_0 \text{ e } \dot{\theta}(0) = 0 \implies C = -\omega^2\cos(\theta_0)$$

Aplicando as condições iniciais, chegamos na seguinte EDO separável:

$$\frac{d\theta}{dt} = \pm \omega \sqrt{2} \sqrt{\cos(\theta) - \cos(\theta_0)}$$

$$\int_0^{\frac{T}{2}} dt = \frac{T}{2} = \pm \frac{1}{\omega \sqrt{2}} \int_{-\theta_0}^{\theta_0} \frac{d\theta}{\sqrt{\cos(\theta) - \cos(\theta_0)}}$$

Escolhendo a solução positiva para o período, já que a negativa não faz sentido físico, temos:

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

Aproximando o período em série, obtém-se a equação 13.

$$T \approx 2\pi \sqrt{\frac{l}{g}} \left[ 1 + \frac{\theta_0^2}{16} \right] \quad (13)$$

O código elaborado para essa tarefa é exposto no Código 2, e sua explicação é dada logo em seguida, no verbete 2.

Código 2: Código-fonte da tarefa B, parte B1-B2

```

1      program TarefaB1eB2
2          implicit real*8 (a-h,o-z)
3          parameter (iangulos = 100)
4          parameter (idivisoes = 5000)
5          dimension theta(iangulos)
6
7          pi = 4.0d0*atan(1.0d0)
8          theta(1) = pi/2.0d0
9
10         open(50, file='saida-b1-b2-13687303-periodos.csv')
11         write(50,100)
12 100    format("theta,periodo_sim,periodo_int,periodo_apx")
13         do i=1, iangulos-1, 1
14             theta(i+1) = theta(i) - pi/(2*iangulos)
15             periodo = 0.0d0
16             eps = 1e-3
17             do j=1, 10, 1
18                 periodo = periodo + simulate(theta(i), 0d0, 0d0, 0d0)
19             end do
20             periodo = periodo/10.0d0
21             periodo_int = spint(-theta(i)+eps, theta(i)-eps, idivisoes,
22 &theta(i))*sqrt(2.0d0)+ 4*sqrt(2.0d0)*sqrt(eps/sin(theta(i)))
23             periodo_apx = 2.0d0*pi*(1 + (theta(i)**2)/16.0d0)
24             write(50,*) theta(i), ",", periodo, ",", periodo_int, ",",
25 &periodo_apx

```

```

26         end do
27         close(50)
28     end program
29
30     function simulate(theta0, gamma, f0, omega)
31         implicit real*8 (a-h,o-z)
32         parameter(ipassos = 3000)
33         parameter(dt = 1e-2)
34         parameter(g = 9.8d0)
35         parameter(al = 9.8d0)
36         parameter(m = 1.0d0)
37
38         dimension t(ipassos), w(ipassos), e(ipassos), theta(ipassos)
39
40         simulate = 0.0d0
41         pi = 4.0d0*atan(1.0d0)
42
43         theta(1) = theta0
44         w(1) = 0.0d0
45         e(1) = m*g*al*(1-cos(theta0))
46         t(1) = 0.0d0
47         iosc = 0
48
49         do i=1, ipassos-1, 1
50             t(i+1) = i*dt
51             w(i+1) = w(i) + (- (g/al)*sin(theta(i)) - gamma*w(i) + f0*s
52 &in(omega*t(i)))*dt
53             if(w(i+1) * w(i) .lt. 0) iosc = iosc + 1
54             if(iosc .eq. 2) then
55                 simulate = t(i)
56                 return
57             end if
58             theta(i+1) = mod(theta(i) + w(i+1)*dt, 2.0d0*pi)
59 c             E = U + K = mgl * (1 - cos(theta)) + 0.5 * m * (w*l)2
60             e(i+1) = m*g*al*(1-cos(theta(i+1)))+ 0.5d0*m*(w(i+1)*al)**2
61         end do
62     end function
63
64     function func(x, x0)
65         implicit real*8 (a-h,o-z)
66         func = 1.0d0/sqrt(cos(x) - cos(x0))
67         return
68     end function
69
70     function fn(x0, n, h, b0)
71         implicit real*8 (a-h,o-z)
72         fn = func(x0 + n*h, b0)
73         return
74     end function fn
75
76     function spint(a,b,n,b0)
77         implicit real*8 (a-h,o-z)
78         spint=0.0d0
79         h=(b-a)/real(n)
80         do i=1, n-1, 2

```

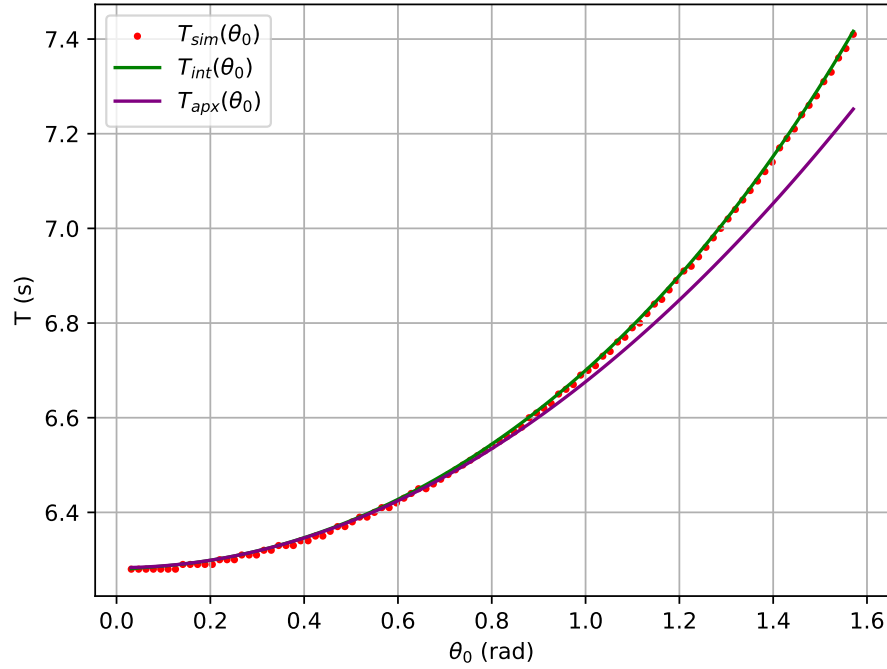
```
81         x0=a+i*h
82         spint=spint+(h/3.0d0)*(fn(x0, 1, h, b0) + 4.0d0*fn(x0, 0, h
83         &,b0) + fn(x0,-1, h, b0))
84     end do
85     return
86 end function
```

### Verbetes 2: Explicação do código da tarefa B

Inicialmente, define-se o número de passos e um vetor com vários valores de  $\theta_0$  para os quais o período será calculado, e depois é aberto o arquivo de saída de dados. A fim de tornar o código mais simples, a parte de simulação foi transformada em uma função *simulate*, que recebe os parâmetros da simulação e retorna o período do movimento. O período obtido a partir da simulação é calculado da seguinte maneira: conta-se quantas vezes  $\omega(t)$  trocou de sinal e, quando a contagem for 2, tem-se um período e a função retorna o valor do tempo  $t_i$  atual. Além disso, para cada  $\theta_0$ , a integral do período é calculada numericamente utilizando o método de Boole, e a expressão aproximada do período também é calculada. Após calcular cada período, os resultados de cada método em função do ângulo  $\theta_i$  são escritos no arquivo de saída.

#### 3.1.1 Resultados e análise

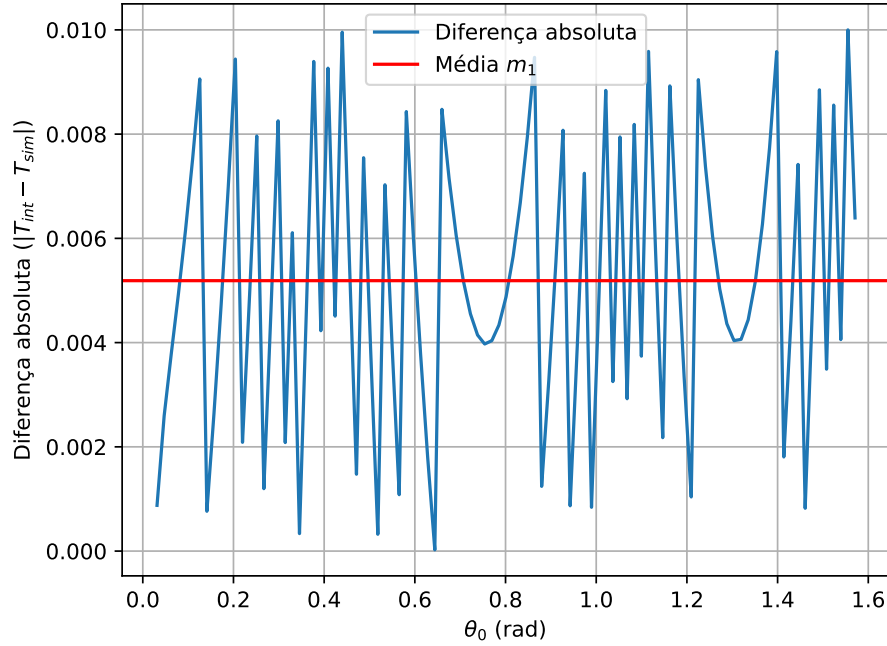
A partir dos períodos obtidos pela integral ( $T_{int}$ ), pela simulação ( $T_{sim}$ ) e pela equação aproximada ( $T_{apx}$ ), foi criado um gráfico conjunto de  $T(\theta_0)$  para os três métodos utilizados, a fim de compará-los, que é exposto na figura 6.

Figura 6:  $T(\theta_0)$  para cada método calculado.

Fonte: Autoria própria

Ao observar a figura 6, pode-se concluir que os métodos da integral e da simulação fornecem resultados aparentemente próximos, sendo difícil discernir graficamente o erro entre esses dois métodos.

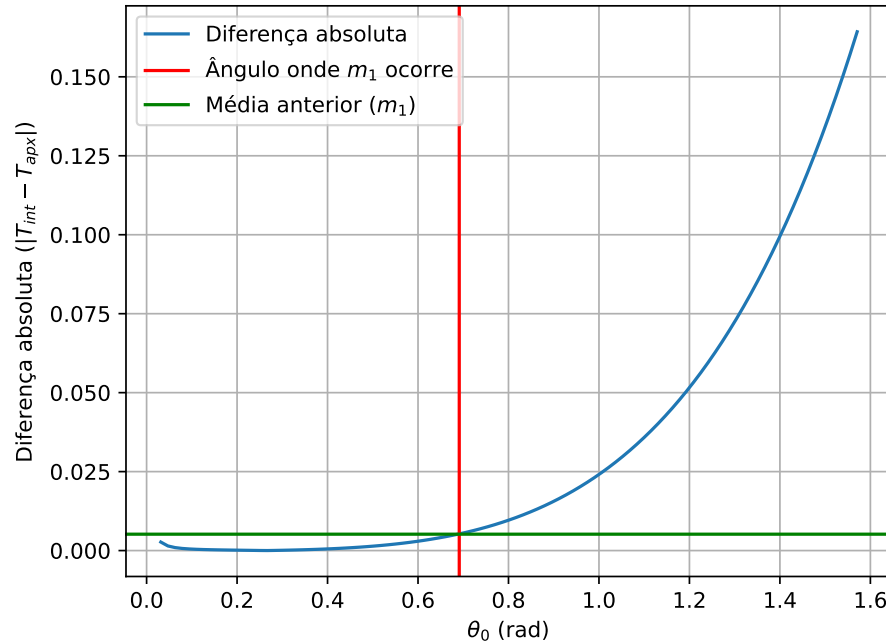
Dessa forma, para comparar o método da integral e da simulação de maneira mais precisa, foi criado um gráfico da diferença absoluta  $|T_{int} - T_{sim}|$  entre os métodos em função de  $\theta_0$ , exibido na figura 7. A partir da análise do gráfico, observa-se que os métodos fornecem resultados muito semelhantes, uma vez que a diferença absoluta oscila na ordem de  $\Delta t$  em torno da média, que está na escala de  $10^{-3}$ , ou seja, o erro entre os dois métodos é desprezível.

Figura 7: Diferença absoluta entre  $T_{sim}$  e  $T_{int}$ 

Fonte: Autoria própria

Analogamente, a fim de comparar os resultados da equação aproximada em relação aos métodos da integral e da simulação, foi criado o gráfico da diferença absoluta  $|T_{apx} - T_{int}|$  entre o método aproximado e o método da integral, exposto na figura 8. No gráfico, a média da diferença absoluta anterior, entre  $T_{int}$  e  $T_{sim}$ , está evidenciada pela linha verde, e o ângulo  $\theta_0$  onde ela ocorre está marcado pela linha vermelha. Assim, é possível encontrar a partir de qual ângulo o método aproximado fornece resultados com erro maior que o método da integral e da simulação.



Figura 8: Diferença absoluta entre  $T_{apx}$  e  $T_{int}$ 

Fonte: Autoria própria

Ao analisar o gráfico da figura 8, observa-se que o método aproximado funciona bem para ângulos menores que 0.7 rad, todavia, para ângulos maiores que 0.7 rad, o erro começa a crescer exponencialmente, fornecendo resultados distantes da realidade. Assim, podemos concluir que os métodos da integral e da simulação fornecem resultados aproximadamente iguais, e o método aproximado fornece bons resultados para ângulos de até 0.7 rad.

### 3.2 Parte B3

Na parte B3, considerou-se  $\gamma = 0.5$  e  $F_0 = 0$ , ou seja, movimento não forçado e com resistência, e os resultados de  $\theta(t)$  foram salvos em um arquivo. O código elaborado para essa tarefa é exposto no Código 3 e sua explicação é dada no verbete 3.

Código 3: Código-fonte da tarefa B, parte B3

```

1      program TarefaB3
2          implicit real*8 (a-h,o-z)
3          parameter(ipassos = 9000)
4          parameter(g = 9.8d0)
5          parameter(al = 9.8d0)
6          parameter(m = 1.0d0)
7
8          dimension t(ipassos), w(ipassos), e(ipassos), theta(ipassos)
9
10         pi = 4.0d0*atan(1.0d0)
11

```

```

12      theta0 = 0.14d0
13      omega = 0.0d0
14      gamma = 0.05d0
15      f0 = 0.0d0
16      dt=1e-2
17
18      open(50, file="saida-b3-13687303-theta_t.csv")
19      write(50,100)
20      100 format("t,theta")
21
22      theta(1) = theta0
23      w(1) = 0.0d0
24      e(1) = m*g*al*(1-cos(theta0))
25      t(1) = 0.0d0
26
27      do i=1, ipassos-1, 1
28          t(i+1) = i*dt
29          w(i+1) = w(i) + (- (g/al)*sin(theta(i)) - gamma*w(i) + f0*s
30      &in(omega*t(i)))*dt
31          theta(i+1) = mod(theta(i) + w(i+1)*dt, 2.0d0*pi)
32      c      E = U + K = mgl * (1 - cos(theta)) + 0.5 * m * (w(1))^2
33          e(i+1) = m*g*al*(1-cos(theta(i+1)))+ 0.5d0*m*(w(i+1)*al)**2
34          write(50,*) t(i), ",", theta(i)
35      end do
36      close(50)
37      end program

```

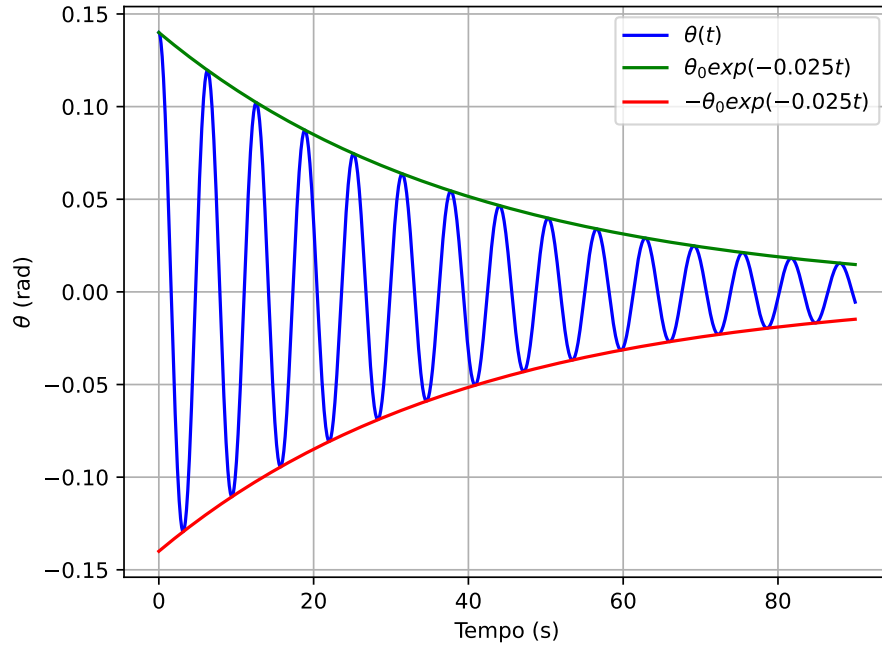
### Verbetes 3: Explicação do código da tarefa B parte B3

O código é muito semelhante ao código da parte B1 e B2, porém mais simplificado, uma vez que só precisamos simular os pêndulos e salvar os dados de  $\theta(t)$ . Inicialmente, são definidos os parâmetros da simulação, assim como o número  $N$  de passos, e abre-se um arquivo de saída.

Em seguida, é realizado um laço  $i$  de 1 a  $N - 1$ , de 1 a 1, onde são calculados os valores de  $t_{i+1}$ ,  $\theta_{i+1}$  e  $\omega_{i+1}$ . Após calculados os valores, salva-se  $\theta_{i+1}$  e  $t_{i+1}$  no arquivo para análise posterior.

#### 3.2.1 Resultados e análise

A partir dos valores de  $\theta(t)$ , foi criado um gráfico do comportamento do ângulo em função do tempo, exposto na figura 9. Ao analisar a figura, observa-se que o movimento descrito é uma oscilação subcrítica, pois a amplitude decresce de acordo com uma exponencial no tempo, da forma  $\theta_0 e^{-0.025t}$ .

Figura 9:  $\theta(t)$  para  $\gamma = 0.5$  e  $F_0 = 0$ 

Fonte: Autoria própria

### 3.3 Parte B4

Em seguida, para a parte B4, considerou-se  $\Omega = \frac{2}{3}$  e  $\Delta t = 0.04$  e criou-se vários gráficos conjuntos de  $\theta(t)$  e  $\omega(t)$  para  $\gamma = 0.05$ ,  $\gamma = 0.5$  e  $F_0 = 0$ ,  $F_0 = 0.5$  e  $F_0 = 1.2$ . Foram considerados dois valores de  $\gamma$  pois o valor  $\gamma = 0.05$  fornece resultados caóticos para todos os casos de  $F_0$  analisados, indicando um possível erro no projeto proposto. Assim, também foram realizados os cálculos para  $\gamma = 0.5$ , uma vez que, para esse  $\gamma$ , os resultados coincidem com o esperado e estão de acordo com os resultados apresentados no livro *Computational Physics* de *Nicholas Giordano*.

O código elaborado para essa tarefa é exposto no Código 4 e sua explicação é dada no verbete 4.

Código 4: Código-fonte da tarefa B, parte B4

```

1      program TarefaB4
2          implicit real*8 (a-h,o-z)
3          parameter(ipassos = 2500)
4          parameter(g = 9.8d0)
5          parameter(al = 9.8d0)
6          parameter(m = 1.0d0)
7
8          dimension t(ipassos), w(ipassos), e(ipassos), theta(ipassos),
9              &f0(3)
10         character fname*50
11         pi = 4.0d0*atan(1.0d0)
12

```

```

13      theta0 = 0.14d0
14      omega = (2.0d0/3.0d0)
15      gamma = 0.5d0
16      f0 = [0.0d0, 0.5d0, 1.2d0]
17      dt=0.04d0
18
19  c      Loop k_gamma para escolher qual gamma usar
20  c      k_gamma = 1 => gamma = 0.05 e k_gamma = 2 => gamma = 0.5
21      do k_gamma = 1, 2, 1
22      if(k_gamma .eq. 1) gamma = 0.05d0
23      if(k_gamma .eq. 2) gamma = 0.5d0
24
25  c      Loop j_f0 para escolher qual f0 usar
26  c      j_f0 = 1 => f0 = 0, j_f0 = 2 => f0 = 0.5 e j_f0 = 3 => f0=1.2
27      do j_f0=1, 3, 1
28      write(fname,100) j_f0, k_gamma
29  100      format('saida-b4-13687303-dados_f0_', i1, '_g_', i1, '.csv')
30      open(50, file=fname)
31      write(50,101)
32  101      format("t,theta,w")
33
34      theta(1) = theta0
35      w(1) = 0.0d0
36      e(1) = m*g*al*(1-cos(theta0))
37      t(1) = 0.0d0
38
39      do i=1, ipassos-1, 1
40      t(i+1) = i*dt
41      w(i+1) = w(i) + (- (g/al)*sin(theta(i)) - gamma*w(i) + f0(j
42  & f0)*sin(omega*t(i)))*dt
43      theta(i+1) = mod(theta(i) + w(i+1)*dt, 2.0d0*pi)
44  c      E = U + K = mgl * (1 - cos(theta)) + 0.5 * m * (w*l)2
45      e(i+1) = m*g*al*(1-cos(theta(i+1)))+ 0.5d0*m*(w(i+1)*al)**2
46      write(50,*) t(i), ",", theta(i), ",", w(i)
47      end do
48      close(50)
49
50      end do
51      end do
52  end program

```

#### Verbetes 4: Explicação do código da tarefa B parte B4

O código é muito semelhante ao código da parte B3, porém com a adição de um algoritmo para realizar a mesma simulação para valores de  $F_0$  e  $\gamma$  diferentes. Inicialmente, são definidos os parâmetros da simulação, assim como o número  $N$  de passos.

Para controlar qual o valor de  $\gamma$  e  $F_0$  usar, é realizado o seguinte:

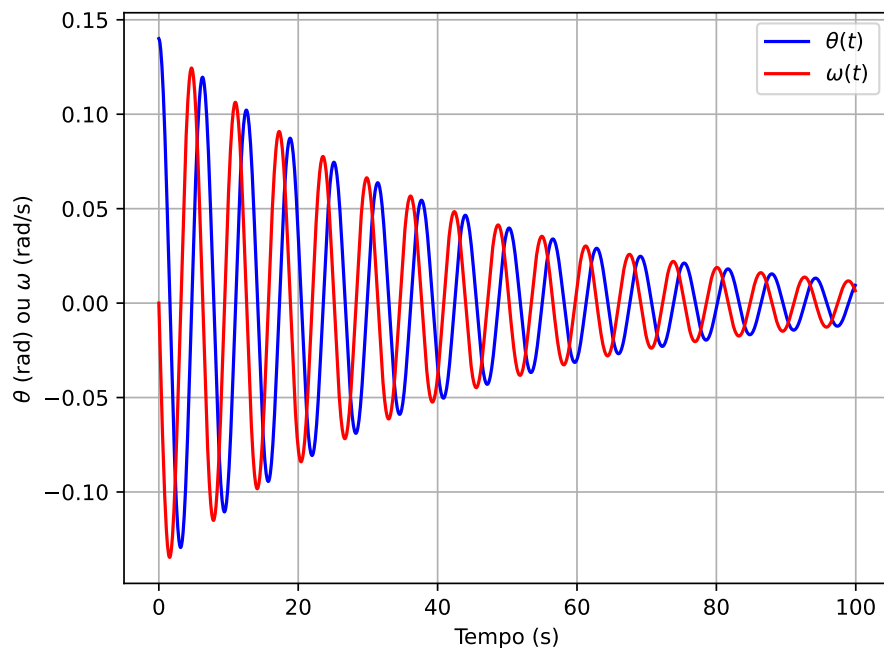
1. O programa inteiro é contido dentro de um laço  $k\_gamma$ , que vai escolher qual dos dois  $\gamma$  serão usados. Se  $k = 1$ , então  $\gamma = 0.05$  e se  $k = 2$ , então  $\gamma = 0.5$ .
2. Dentro do laço  $k\_gamma$ , é inserido o laço  $j\_f\theta$ , que vai escolher qual dos três  $F_0$  serão usados. Se  $j = 1$ , então  $F_0 = 0$ , se  $k = 2$ , então  $F_0 = 0.5$  e se  $j = 3$ , então  $F_0 = 1.2$ .
3. Os valores de  $j$  e  $k$  são escritos em uma string  $fname$ , que dará o nome ao arquivo de saída. Assim, é possível saber quais valores de  $F_0$  e  $\gamma$  foram utilizados para gerar aquele resultado.

Em seguida, é realizado um laço  $i$  de 1 a  $N - 1$ , de 1 a 1, onde são calculados os valores de  $t_{i+1}$ ,  $\theta_{i+1}$  e  $\omega_{i+1}$ . Após calculados os valores, salva-se  $\theta_{i+1}$ ,  $\omega_{i+1}$  e  $t_{i+1}$  no arquivo para análise posterior.

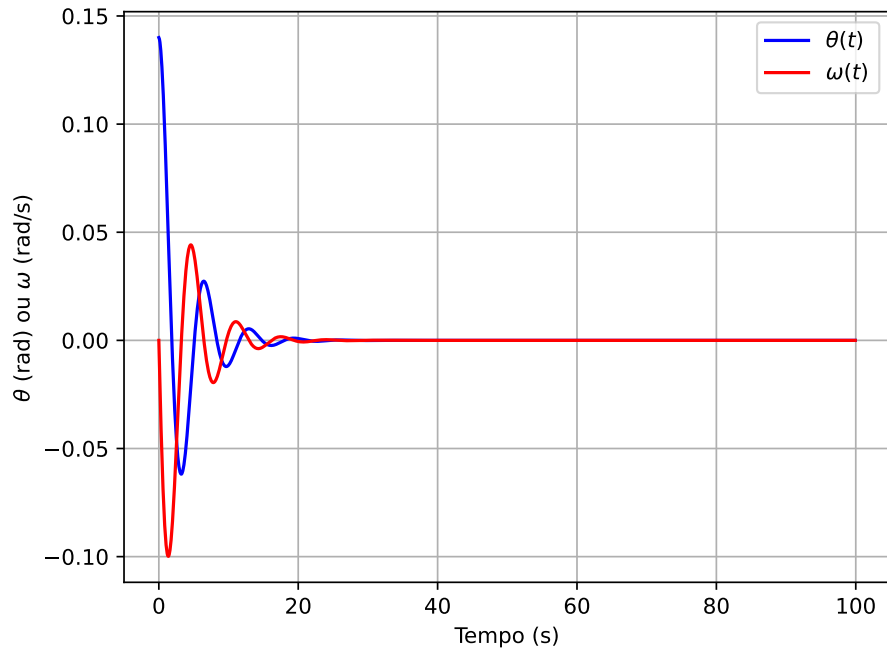
#### 3.3.1 Resultados e análise

Os gráficos elaborados são expostos abaixo, e as condições utilizadas na simulação estão expostas no título de cada figura.

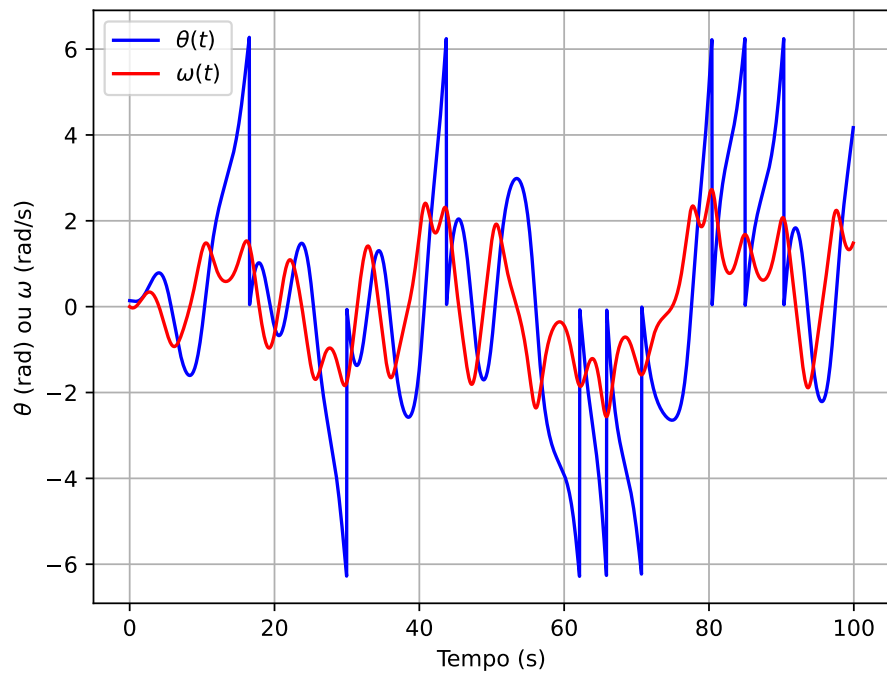
Figura 10:  $\theta(t)$  e  $\omega(t)$  para  $\gamma = 0.05$  e  $F_0 = 0$ .



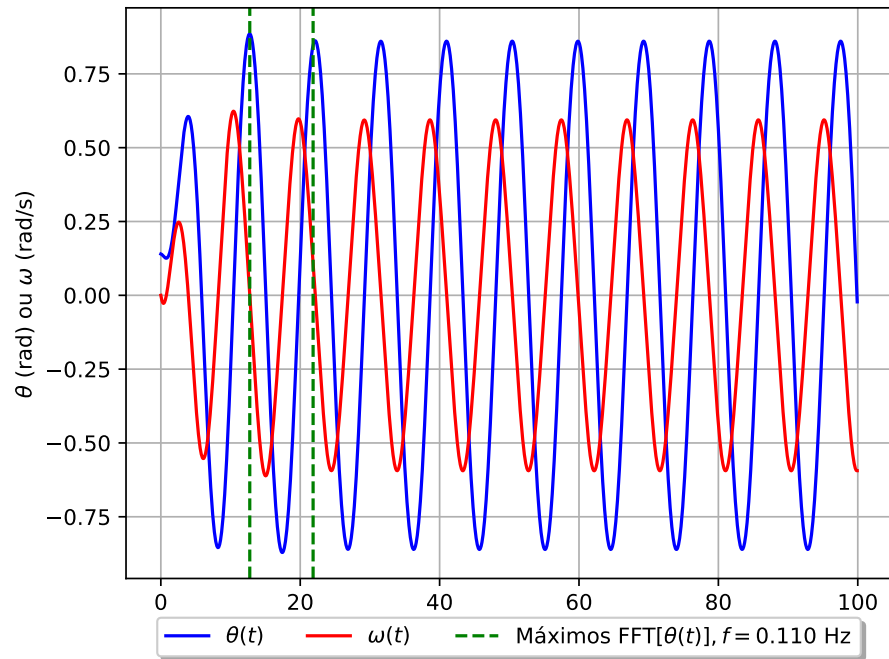
Fonte: Autoria própria

Figura 11:  $\theta(t)$  para  $\gamma = 0.5$  e  $F_0 = 0$ 

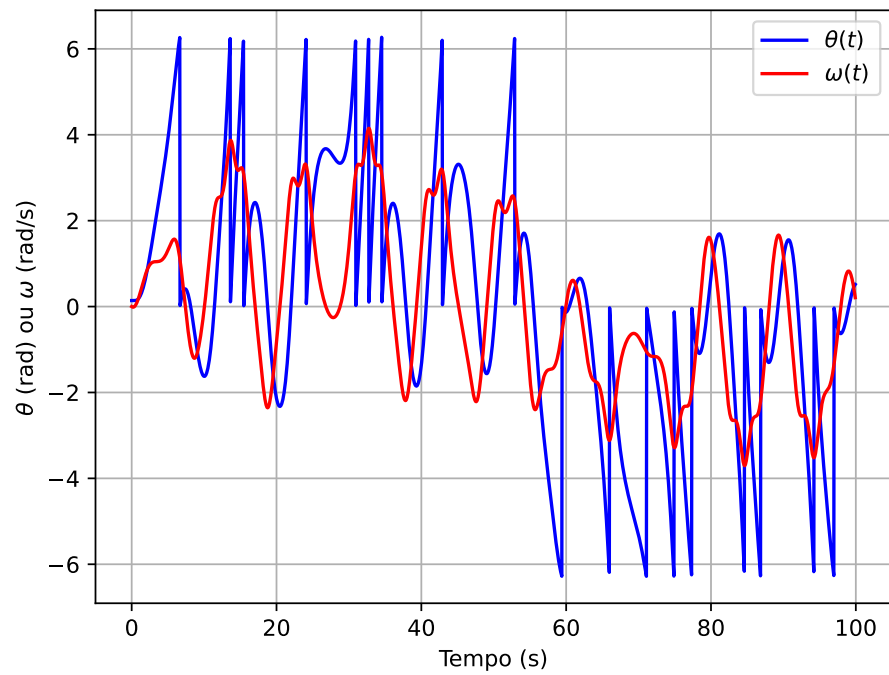
Fonte: Autoria própria

Figura 12:  $\theta(t)$  para  $\gamma = 0.05$  e  $F_0 = 0.5$ 

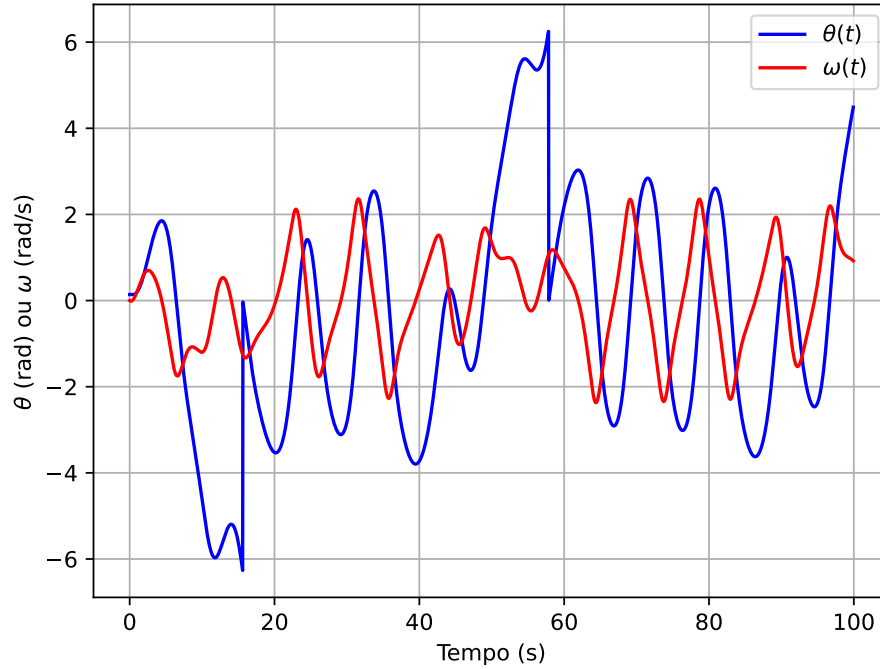
Fonte: Autoria própria

Figura 13:  $\theta(t)$  para  $\gamma = 0.5$  e  $F_0 = 0.5$ 

Fonte: Autoria própria

Figura 14:  $\theta(t)$  para  $\gamma = 0.05$  e  $F_0 = 1.2$ 

Fonte: Autoria própria

Figura 15:  $\theta(t)$  para  $\gamma = 0.5$  e  $F_0 = 1.2$ 

Fonte: Autoria própria

A partir da análise dos gráficos, observa-se que, para  $\gamma = 0.05$  e  $F_0 = 0$ , tem-se um amortecimento subcrítico e para  $\gamma = 0.5$  e  $F_0 = 0$  tem-se amortecimento crítico. Para  $\gamma = 0.05$  e  $F_0 = 0.5$ , tem-se movimento caótico e não periódico, já para  $F_0 = 0.5$  e  $\gamma = 0.5$ , tem-se um movimento forçado na fase estacionária, cuja frequência do movimento foi calculada utilizando Fast Fourier Transform (FFT) no pós-processamento dos dados e é exposta na equação 14.

$$f_{FFT} = 0.110 \text{ Hz} \implies \omega_{FFT} = 0.691 \text{ rad/s} \quad (14)$$

Finalmente, para os casos  $\gamma = 0.05$  e  $F_0 = 1.2$  e  $\gamma = 0.5$  e  $F_0 = 1.2$ , tem-se movimento caótico e não periódico novamente. Em todos os casos simulados, nota-se que  $\omega(t)$  está defasado de  $\frac{\pi}{2}$  em relação à  $\theta(t)$ , o que é esperado, já que  $\omega = \frac{d\theta}{dt}$  e  $\theta(t)$  depende de uma função cosseno ou seno, dependendo da fase inicial.

## 4 Tarefa C

Nessa tarefa, realizou-se a simulação de vários pêndulos para ângulos iniciais  $\theta_0$  ligeiramente diferentes e amplitudes de força  $F_0 = 0.5$  e  $F_0 = 1.2$ , variando 0.001 rad entre cada ângulo. A partir disso, calculou-se a diferença  $\Delta\theta$  em função do tempo, a fim de medir o grau de dispersão dos resultados em função do tempo, ou seja, avaliar se o comportamento do sistema é caótico ou não: se  $\Delta\theta$  tende a zero



com o tempo, então o sistema não é caótico, mas se não tender a zero, então o sistema é caótico e sensível às condições iniciais.

Em todos os cálculos, considerou-se  $\gamma = 0.5$ ,  $\Omega = \frac{2}{3}$  e  $\Delta t = 0.04$ . O valor de  $\gamma$  foi considerado como 0.5 pois, de acordo com os resultados da tarefa B, esse deveria ser o valor correto de  $\gamma$  para fornecer os resultados esperados. Inicialmente, o valor  $\gamma = 0.05$  foi testado, mas forneceu resultados que não contribuiriam com a discussão: ambos os casos de  $F_0$  resultaram em movimentos caóticos.

O código elaborado para essa tarefa é exposto no Código 5 e sua explicação é dada no verbete 5.

Código 5: Código-fonte da tarefa C

```

1      program TarefaC
2          implicit real*8 (a-h,o-z)
3          parameter(ipassos = 2000)
4          parameter(dt = 0.04d0)
5          parameter(g = 9.8d0)
6          parameter(al = 9.8d0)
7          parameter(m = 1.0d0)
8
9          dimension t1(ipassos), w1(ipassos), theta1(ipassos)
10         dimension t2(ipassos), w2(ipassos), theta2(ipassos)
11         dimension dtheta(ipassos)
12         dimension f0_arr(2), theta0_arr(2)
13         character fname*50
14
15         gamma = 0.5d0
16         omega = 2.0d0/3.0d0
17         f0_arr = [0.5d0, 1.2d0]
18         theta0_arr = [0.14d0, 0.14d0 - 0.001d0]
19
20         do i=1, 2, 1
21             f0 = f0_arr(i)
22
23             write(fname,100) i
24 100         format("saida-c-13687303-dados_f0_", I0, ".csv")
25             open(50, file=fname)
26             write(50,101)
27 101         format("t,dtheta")
28
29             theta0 = theta0_arr(1)
30             call simulate(ipassos,dt,g,al,m,theta0,gamma,f0,omega,t1,w1
31 &,theta1)
32             theta0 = theta0_arr(2)
33             call simulate(ipassos,dt,g,al,m,theta0,gamma,f0,omega,t2,w2
34 &,theta2)
35
36             do j=1, ipassos, 1
37                 dtheta(j) = theta2(j) - theta1(j)
38                 write(50,*) t1(j), ",", dtheta(j)
39             end do
40
41             close(50)

```

```

42         end do
43     end program
44
45     subroutine simulate(ipassos,dt,g,al,m,theta0,gamma,f0,omega,t,w,t
46 &heta)
47         implicit real*8 (a-h,o-z)
48         dimension t(*), w(*), theta(*)
49
50         pi = 4.0d0*atan(1.0d0)
51
52         theta(1) = theta0
53         w(1) = 0.0d0
54         t(1) = 0.0d0
55         iosc = 0
56
57         do i=1, ipassos-1, 1
58             t(i+1) = i*dt
59             w(i+1) = w(i) + (- (g/al)*sin(theta(i)) - gamma*w(i) + f0*s
60 &in(omega*t(i)))*dt
61             theta(i+1) = mod(theta(i) + w(i+1)*dt, 2.0d0*pi)
62         end do
63     end subroutine

```

#### Verbetes 5: Explicação do código da tarefa C

Inicialmente, são definidos os parâmetros a serem utilizados nas simulações, como o número de passos  $N$ , e os dois ângulos  $\theta_0$  a serem simulados para calcular a diferença  $\Delta\theta$ , bem como as duas amplitudes  $F_0$  a serem utilizadas. Em seguida, é feito um laço  $i$  de 1 a 2, para realizar as duas simulações para cada força. Assim, quando  $i = 1$ , utiliza-se  $F_0 = 0.5$  e quando  $i = 2$ , utiliza-se  $F_0 = 1.2$ . Dentro do laço, é aberto um arquivo de saída, onde serão salvos os dados, e no nome do arquivo é inserida a variável  $i$ , de forma que seja possível identificar qual  $F_0$  foi utilizado para gerar os dados.

Após abrir o arquivo, são realizadas duas simulações chamando a subrotina *simulate*, que possui a mesma lógica apresentada na tarefa B, porém com uma adição: a subrotina irá escrever os resultados de  $t$ ,  $\omega$  e  $\theta$  nos vetores passados como argumentos.

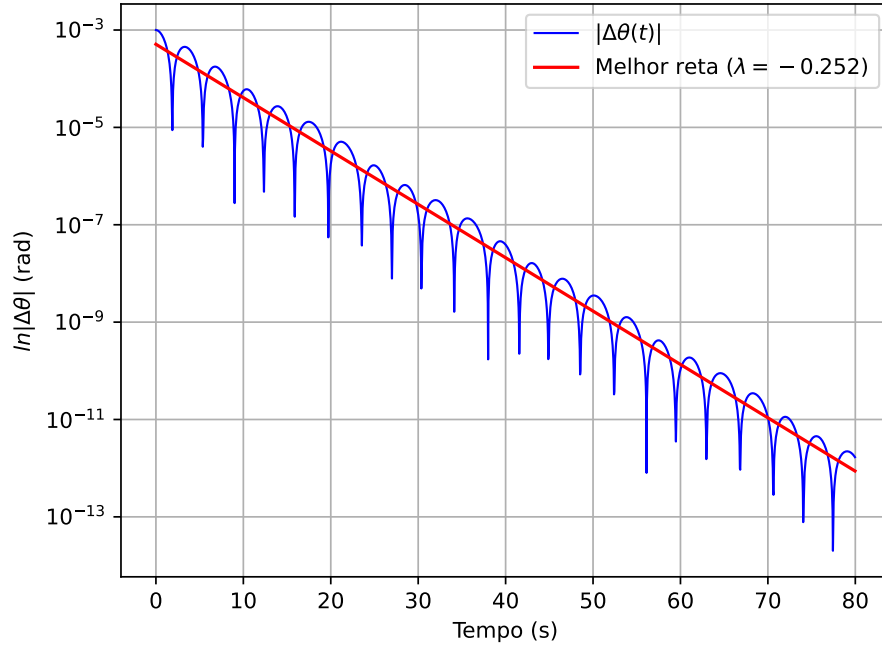
Após realizadas as duas simulações, é feito um laço de  $j = 1$  a  $j = N - 1$ , de 1 em 1, de modo que, para cada passo, seja calculado  $\Delta\theta_o = \theta_{2i} - \theta_{1i}$  e o resultado seja salvo no arquivo, junto com o tempo  $t_i$ .

## 4.1 Resultados e análise

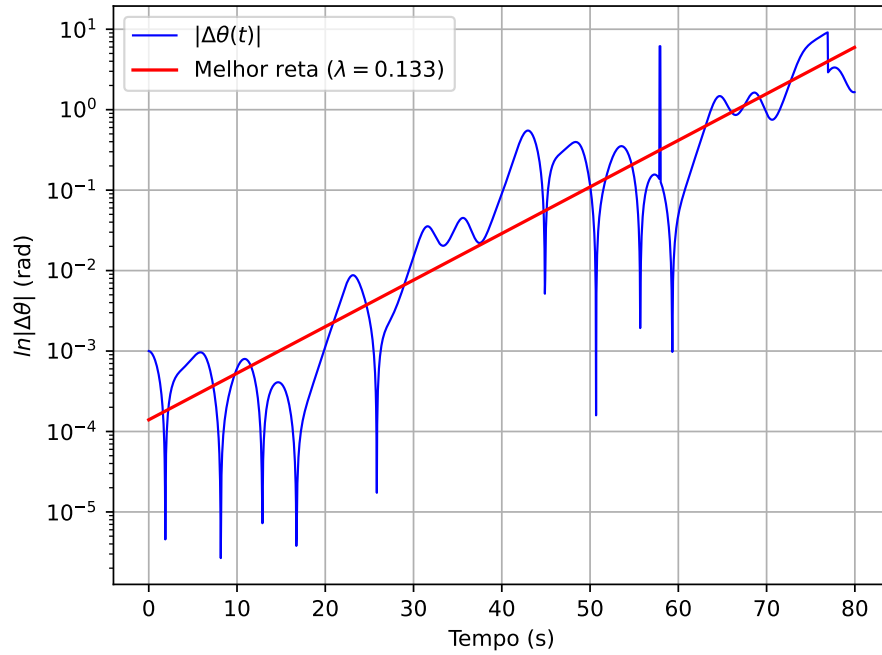
A partir das simulações realizadas, criou-se os gráficos de  $|\Delta\theta|$  em função do tempo e utilizou-se o Método dos Mínimos Quadrados (MMQ) para encontrar a melhor reta que descreve os dados e, a partir da reta, obteve-se o coeficiente angular  $\lambda$ , conhecido como coeficiente de Liapunov, que mede o grau do caos do sistema. O módulo de  $\Delta\theta$  foi utilizado para evitar problemas com o domínio da função logarítmica.

Na figura 16, é exibido o gráfico de  $|\Delta\theta|$  para  $F_0 = 0.5$ , e na figura 17, é exibido o gráfico de  $|\Delta\theta|$  para  $F_0 = 1.2$ , sendo os dois gráficos com escala logarítmica natural no eixo vertical e escala linear no eixo horizontal. Nas equações 15 e 16, são exibidos os coeficientes de Liapunov encontrados para os casos  $F_0 = 0.5$  e  $F_0 = 1.2$ , respectivamente.

Figura 16:  $\Delta\theta$  para  $F_0 = 0.5$  (y em escala  $\ln(y)$ )



Fonte: Autoria própria

Figura 17:  $\Delta\theta$  para  $F_0 = 1.2$  (y em escala  $\ln(y)$ )

Fonte: Autoria própria

Os coeficientes de Liapunov  $\lambda_1$  e  $\lambda_2$  encontrados a partir da melhor reta que descreve os dados foram:

$$\lambda_1 = -0.252 \quad (15)$$

$$\lambda_2 = 0.133 \quad (16)$$

A partir da análise dos gráficos e dos coeficientes obtidos, observa-se que, no caso  $F_0 = 0.5$ , temos movimento não caótico, uma vez que o coeficiente de Liapunov é negativo e, portanto,  $\Delta\theta$  tende a zero com o tempo. Já no caso  $F_0 = 1.2$ , observa-se movimento caótico, pois o coeficiente de Liapunov é positivo, ou seja,  $\Delta\theta$  tende a aumentar com o tempo, o que indica que o sistema é sensível às condições iniciais.

## 5 Tarefa D

Nessa tarefa, assim como na tarefa C, 4 pêndulos foram simulados com ângulos iniciais  $\theta_1$ ,  $\theta_2$ ,  $\theta_3$  e  $\theta_4$  ligeiramente diferentes, com uma diferença de 0.001 rad entre eles, para amplitudes de força  $F_0 = 0.5$  e  $F_0 = 1.2$ , considerando  $\gamma = 0.5$ . Após as simulações, os dados de  $\omega(t)$  e  $\theta(t)$  foram guardados, permitindo a criação de gráficos do espaço de fase  $\omega(\theta)$  para cada caso.

O espaço de fase  $\omega(\theta)$  indica o comportamento geral do sistema: se o traço  $\omega(\theta)$  é aproximadamente igual para dois ângulos iniciais ligeiramente diferentes, então o

sistema fornece resultados parecidos para as duas condições iniciais. Todavia, se o traço é diferente para dois ângulos iniciais ligeiramente diferentes, isso significa que o sistema é sensível à pequenas variações das condições iniciais, ou seja, é caótico.

O código elaborado para essa tarefa é exposto no Código 6 e sua explicação é dada no verbete 6.

Código 6: Código-fonte da tarefa D

```

1      program TarefaD
2          implicit real*8 (a-h,o-z)
3          parameter(ipassos = 3000)
4          parameter(dt = 0.04d0)
5          parameter(g = 9.8d0)
6          parameter(al = 9.8d0)
7          parameter(m = 1.0d0)
8          parameter(iangulos = 4)
9
10         dimension t(ipassos), w(ipassos), theta(ipassos)
11         dimension f0_arr(2), theta0_arr(iangulos)
12         character fname*50
13
14         gamma = 0.5d0
15         omega = 2.0d0/3.0d0
16         f0_arr = [0.5d0, 1.2d0]
17         do j=1, iangulos, 1
18             theta0_arr(j) = 0.14d0 - 1e-3*j
19         end do
20
21         do i=1, 2, 1
22             f0 = f0_arr(i)
23
24             do j=1, iangulos, 1
25
26                 write(fname,100) i, j
27 100             format("saida-d-13687303-dados_f0_", I0,"_theta0_", I0,
28 &".csv")
29                 open(50, file=fname)
30                 write(50,101)
31 101             format("t,theta,w")
32                 theta0 = theta0_arr(j)
33                 call simulate(ipassos,dt,g,al,m,theta0,gamma,f0,omega,t,w
34 &,theta)
35                 do k=1, ipassos, 1
36                     write(50,*) t(k), ",", theta(k), ",", w(k)
37                 end do
38                 close(50)
39
40             end do
41         end do
42     end program
43
44     subroutine simulate(ipassos,dt,g,al,m,theta0,gamma,f0,omega,t,w,t
45 &heta)
46         implicit real*8 (a-h,o-z)

```

```

47      dimension t(*), w(*), theta(*)
48
49      pi = 4.0d0*atan(1.0d0)
50
51      theta(1) = theta0
52      w(1) = 0.0d0
53      t(1) = 0.0d0
54      iosc = 0
55
56      do i=1, ipassos-1, 1
57          t(i+1) = i*dt
58          w(i+1) = w(i) + (- (g/al)*sin(theta(i)) - gamma*w(i) + f0*s
59      &in(omega*t(i)))*dt
60          theta(i+1) = mod(theta(i) + w(i+1)*dt, 2.0d0*pi)
61      end do
62  end subroutine

```

#### Verbetes 6: Explicação do código da tarefa D

Esse código é extremamente semelhante ao código da tarefa C, com algumas modificações:

1. É realizada apenas uma simulação para cada ângulo inicial  $\theta_0$ .
2. São definidos 4 ângulos  $\theta_0$ , de forma a realizar uma simulação para cada ângulo. Esses ângulos são definidos em um vetor, que é gerado automaticamente segundo a equação  $0.14 - 0.001j$ , onde  $j$  é o número do ângulo a ser gerado (1, 2, 3, 4).

Inicialmente, define-se os parâmetros das simulações, como o número  $N$  de passos e um vetor que contém as duas amplitudes  $F_0$  da força externa que serão consideradas. Em seguida, é feito um laço  $i = 1$  a  $i = 2$ , onde  $i$  define qual força será utilizada: se  $i = 1$ , então  $F_0 = 0.5$  e se  $i = 2$ , então  $F_0 = 1.2$ . Esse índice é escrito no nome de um arquivo de saída onde serão escritos os dados, de forma que seja possível reconhecer qual força  $F_0$  foi utilizada para gerar aqueles dados.

No interior do laço  $i$ , é feito um laço  $j$  onde  $j$  é o índice do vetor de ângulos iniciais, de forma que  $\theta_0 = \text{vetor}[j]$ . Esse índice também é escrito no nome do arquivo de saída, de forma a tornar possível identificar qual ângulo deu origem aos dados do arquivo.

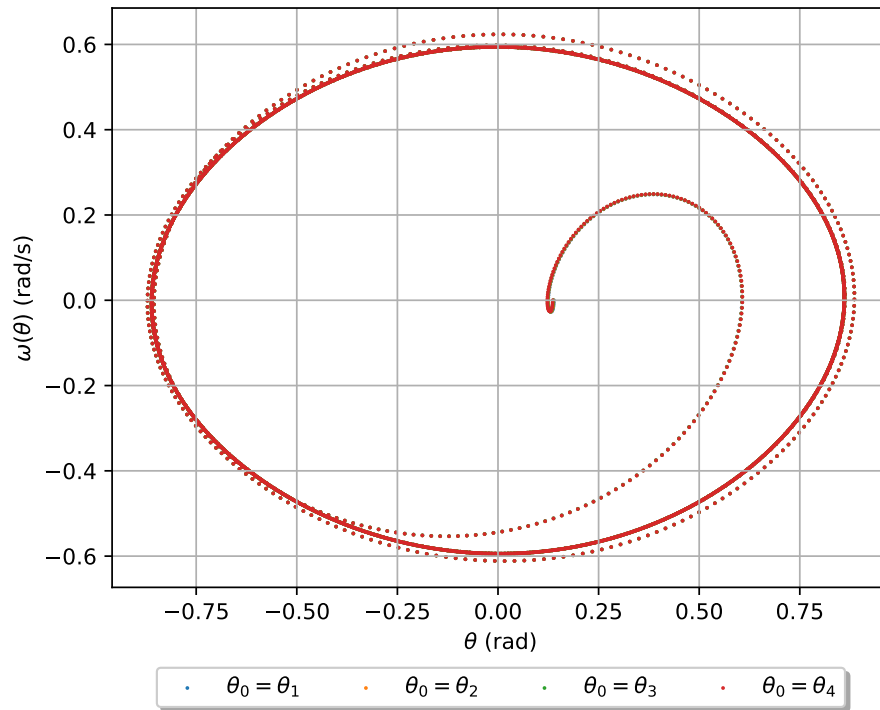
Por fim, no interior do laço  $i$ , é realizada a simulação chamando a subrotina *simulate*, e os valores de  $t_k$ ,  $\theta_k$  e  $\omega_k$  para  $k = 1$  a  $k = N$  são escritos no arquivo para processamento posterior.

## 5.1 Resultados e análise

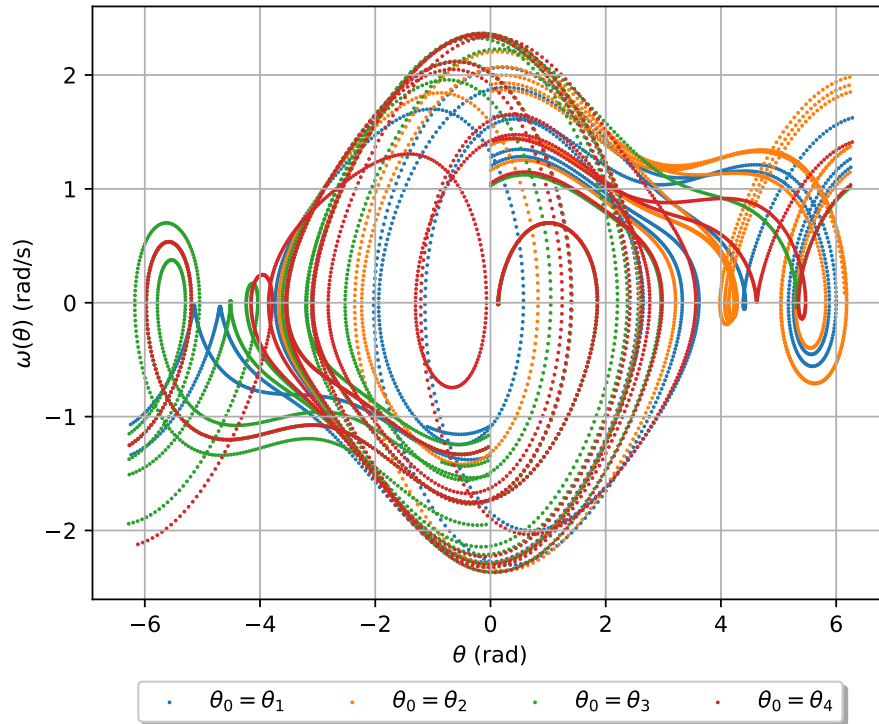
A partir dos dados gerados pelas simulações, foi possível criar o gráfico do espaço de fase para os casos  $F_0 = 0.5$  e  $F_0 = 1.2$ , expostos nas figuras 18 e 19, respectivamente. Vale notar que, de acordo com a análise realizada na tarefa C, o caso  $F_0 = 0.5$  deve ter um comportamento não caótico, enquanto o caso  $F_0 = 1.2$  deve ter um

comportamento caótico.

Figura 18:  $\theta(\omega)$  para  $F_0 = 0.5$



Fonte: Autoria própria

Figura 19:  $\theta(\omega)$  para  $F_0 = 1.2$ 

Fonte: Autoria própria

Analisando as duas figuras, pode-se perceber que o comportamento descrito pelo traço  $\omega(\theta)$  é exatamente o esperado. No caso  $F_0 = 0.5$ , onde era esperado que o sistema não fosse caótico, o traço é aproximadamente o mesmo para todos os ângulos iniciais simulados. Já no caso  $F_0 = 1.2$ , onde era esperado que o sistema fosse caótico, o traço é extremamente diferente para cada ângulo inicial.

Além de ser possível classificar o comportamento do sistema de acordo com o gráfico do espaço de fase, é possível notar que, mesmo no caso caótico, existem locais onde  $\omega(\theta)$  é mais concentrado e que existem locais onde há ausência de traços  $\omega(\theta)$ , o que indica um padrão no caos do sistema.

## 6 Tarefa E

Na tarefa E, a fim de observar o comportamento do espaço de fase do sistema quando a força externa é nula, avaliou-se  $\omega(\theta)$  apenas nos casos onde  $\Omega t = n\pi$ , ou seja, quando  $0 = F_{ext} = F_0 \sin(\Omega t)$ , já que  $\sin(x) = 0$  se  $x = n\pi$ . Essa condição especial, que configura um "corte" no espaço de fase, é chamada de seção de Poincaré. As seções de Poincaré foram geradas para dois casos, com  $F_0 = 0.5$  e com  $F_0 = 1.2$ , considerando  $\gamma = 0.5$  nos dois casos, permitindo assim estudar de maneira mais profunda os casos analisados anteriormente na tarefa D.

O processo utilizado para gerar os dados é o mesmo da tarefa D, ou seja, são si-



mulados 5 pêndulos com ângulos iniciais diferindo ligeiramente por 0.001 rad, e em seguida, são salvos os valores de  $\omega(t)$  e  $\theta(t)$  apenas quando  $\Omega t = n\pi$ . Para verificar se  $\Omega t = n\pi$ , basta verificar se  $\text{mod}(\Omega t, \pi) = 0$ , pois as duas afirmações são análogas matematicamente. Para levar em conta as variações de ponto flutuante ou erros do método numérico, a condição da inequação 17 foi utilizada para verificar se  $\Omega t = n\pi$ .

$$\text{mod}(\Omega t, \pi) < \frac{\Delta t}{2} \quad (17)$$

O código criado para essa tarefa é exposto no Código 7, e sua explicação é dada no verbete 7.

Código 7: Código-fonte da tarefa E

```

1      program TarefaE
2          implicit real*8 (a-h,o-z)
3          parameter(ipassos = 100000)
4          parameter(dt = 0.04d0)
5          parameter(g = 9.8d0)
6          parameter(al = 9.8d0)
7          parameter(m = 1.0d0)
8          parameter(iangulos = 5)
9
10         dimension t(ipassos), w(ipassos), theta(ipassos)
11         dimension f0_arr(2), theta0_arr(iangulos)
12         character fname*60
13
14         pi = 4.0d0*atan(1.0d0)
15         gamma = 0.5d0
16         omega = 2.0d0/3.0d0
17         f0_arr = [0.5d0, 1.2d0]
18         do j=1, iangulos, 1
19             theta0_arr(j) = pi/6.0d0 - 1e-3*j
20         end do
21
22         do i=1, 2, 1
23             f0 = f0_arr(i)
24
25             do j=1, iangulos, 1
26
27                 write(fname,100) i, j
28 100             format("saida-e-13687303-dados_f0_", I0,"_theta0_", I0,
29 &" ".csv")
30                 open(50, file=fname)
31                 write(50,101)
32 101             format("t,theta,w")
33                 theta0 = theta0_arr(j)
34                 call simulate(ipassos,dt,g,al,m,theta0,gamma,f0,omega,t,w
35 &,theta)
36                 do k=1, ipassos, 1
37                     if(mod(omega*t(k), pi) .lt. dt/2.0d0) then
38                         write(50,*) t(k), ",", theta(k), ",", w(k)
39                     end if
40                 end do

```

```
41         close(50)
42
43     end do
44 end do
45 end program
46
47 subroutine simulate(ipassos,dt,g,al,m,theta0,gamma,f0,omega,t,w,t
48 &heta)
49     implicit real*8 (a-h,o-z)
50     dimension t(*), w(*), theta(*)
51
52     pi = 4.0d0*atan(1.0d0)
53
54     theta(1) = theta0
55     w(1) = 0.0d0
56     t(1) = 0.0d0
57     iosc = 0
58
59     do i=1, ipassos-1, 1
60         t(i+1) = i*dt
61         w(i+1) = w(i) + (- (g/al)*sin(theta(i)) - gamma*w(i) + f0*s
62 &in(omega*t(i)))*dt
63         theta(i+1) = mod(theta(i) + w(i+1)*dt, 2.0d0*pi)
64     end do
65 end subroutine
```

### Verbetes 7: Explicação do código da tarefa E

O código da tarefa E é extremamente semelhante ao código da tarefa D, com apenas algumas modificações pontuais:

1. O número de ângulos simulados é 5
2. Antes de salvar os dados no arquivo, é verificado se  $\text{mod}(\Omega t, \pi) < \frac{\Delta t}{2}$ . Se a condição for verdadeira, os dados são salvos. Se a condição for falsa, eles não são salvos e o programa continua.

Toda a lógica restante é exatamente igual ao código da tarefa D.

Inicialmente, define-se os parâmetros das simulações, como o número  $N$  de passos e um vetor que contém as duas amplitudes  $F_0$  da força externa que serão consideradas.

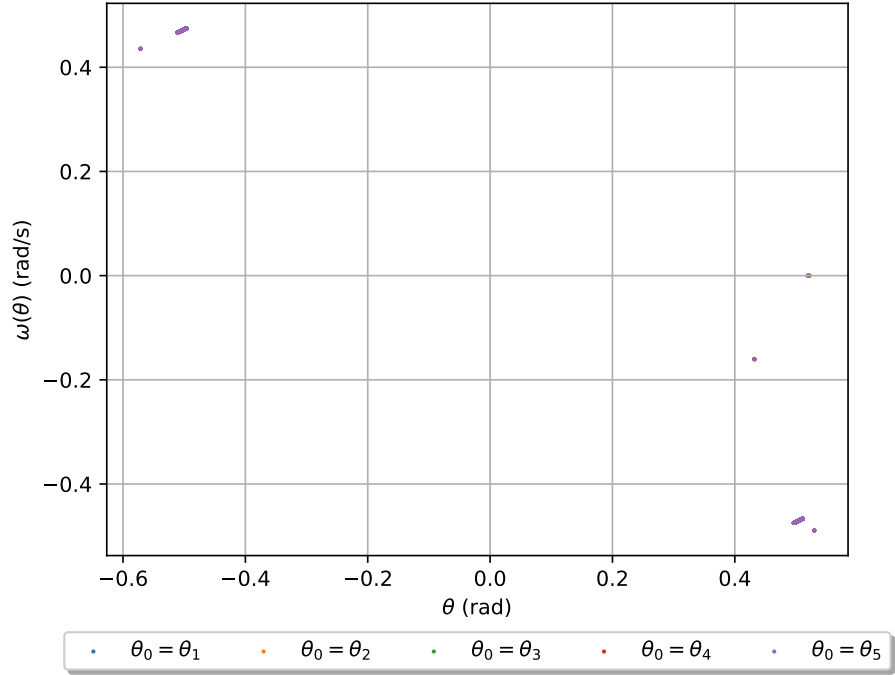
Em seguida, é feito um laço  $i = 1$  a  $i = 2$ , onde  $i$  define qual força será utilizada: se  $i = 1$ , então  $F_0 = 0.5$  e se  $i = 2$ , então  $F_0 = 1.2$ . Esse índice é escrito no nome de um arquivo de saída onde serão escritos os dados, de forma que seja possível reconhecer qual força  $F_0$  foi utilizada para gerar aqueles dados.

No interior do laço  $i$ , é feito um laço  $j$  onde  $j$  é o índice do vetor de ângulos iniciais, de forma que  $\theta_0 = \text{vetor}[j]$ . Esse índice também é escrito no nome do arquivo de saída, de forma a tornar possível identificar qual ângulo deu origem aos dados do arquivo.

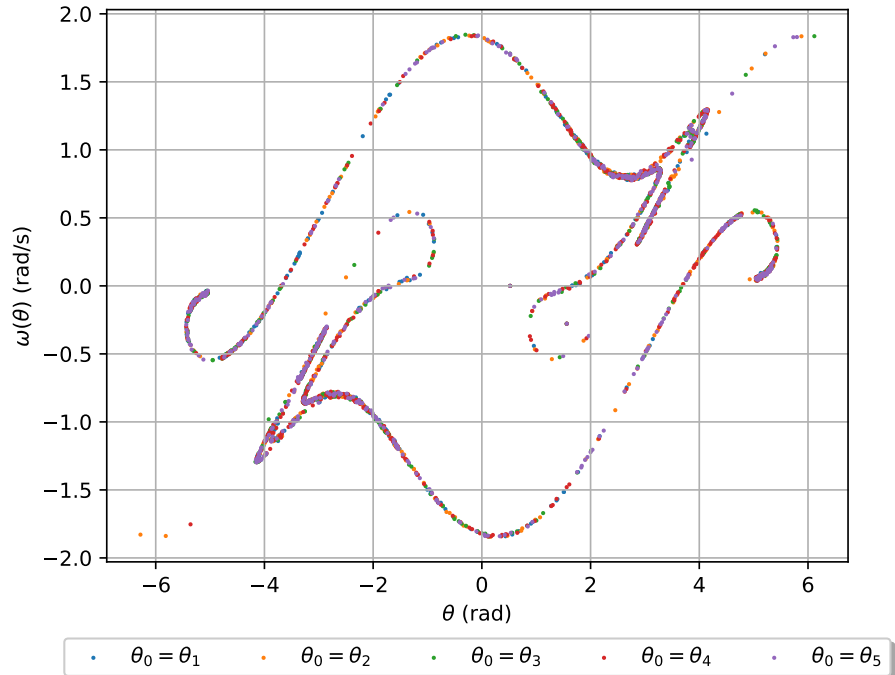
Por fim, no interior do laço  $i$ , é realizada a simulação chamando a subrotina *simulate*, e os valores de  $t_k$ ,  $\theta_k$  e  $\omega_k$  para  $k = 1$  a  $k = N$  são escritos no arquivo para processamento posterior.

## 6.1 Resultados e análise

A partir dos valores de  $\omega(t)$  e  $\theta(t)$  gerados com a condição da seção de Poincaré, foi possível criar gráficos de  $\omega(\theta)$  quando  $\Omega t = n\pi$ . Os gráficos criados para os casos  $F_0 = 0.5$  e  $F_0 = 1.2$  são expostos nas figuras 20 e 21, respectivamente.

Figura 20:  $\theta(\omega)$  para  $F_0 = 0.5$  (seção de Poincaré)

Fonte: Autoria própria

Figura 21:  $\theta(\omega)$  para  $F_0 = 1.2$  (seção de Poincaré)

Fonte: Autoria própria

Ao observar as figuras, nota-se que, no caso não caótico, com  $F_0 = 0.5$ , os pontos de  $\omega(\theta)$  estão isolados em pequenos aglomerados, de modo que a maior parte do espaço de fase se encontra vazio. Esses pequenos aglomerados são chamados de ressonâncias, regiões de recorrência de pontos da seção de Poincaré. Já no caso caótico, com  $F_0 = 1.2$ , o traço de  $\omega(\theta)$  forma um padrão fractal, evidenciando a existência de um padrão bem definido para o comportamento caótico do sistema.