

Projeto 5

Juliana Naomi Yamauti Costa, nºUSP 10260434

10 Maio 2020

Fenômenos oscilatórios estão constantemente presentes em nossas vidas. Tudo aquilo que se move em dois sentidos de forma alternada em torno de uma posição de equilíbrio é dado como um fenômeno oscilatório. Neste projeto, iremos nos concentrar no objeto que nos proporciona um dos mais simples exemplos desse movimento: **o pêndulo**, o qual além de oscilatório, apresenta periodicidade.

Quando o ângulo de lançamento do pêndulo é pequeno o suficiente, sua equação do movimento se reduz a do Movimento Harmônico Simples (MHS), cuja solução é conhecida e também pode ser obtida numericamente. Esses tipos de osciladores harmônicos são normalmente observados operando próximos ao equilíbrio, governados por forças restauradoras lineares ou forças elásticas (Lei de Hooke). Além disso, para pequenas oscilações em torno do equilíbrio, forças restauradoras não-lineares também podem gerar MHS - como ocorre com o pêndulo.

No entanto, quando fora dessas exceções, a não-linearidade pode se tornar relevante e os padrões constantes de oscilações, antes predizíveis, podem se tornar **caóticos** - um movimento não repetitivo caracterizado por um tipo particular de imprevisibilidade.

1 Tarefa A

Nesta tarefa iremos resolver numericamente o pêndulo simples dentro da aproximação harmônica utilizando o método de Euler. As linhas destacadas apresentam as condições iniciais utilizadas. Foram utilizadas duas discretizações de tempo para comparação: $\Delta t = 0.1$ e 0.01 s.

1.1 Subtarefa A1

tarefaA1-10260434.f90

```
Program Tarefa_A1
2
3  IMPLICIT REAL*8 (a-h,o-z) !dupla precisão
4  parameter (nmax = 500) !número de iterações
5  dimension omega(nmax), theta(nmax), tempo(nmax), energia(nmax)
6
7  pi = 4.d0*datan(1.d0)
8  dt = 0.1 !s
9  l = 9.8 !m
10 g = 9.8 !m/s
11 m = 1.d0 !kg
12 tempo(0) = 0.d0 !s
13 omega(0) = 0.d0 !rad/s
14 theta(0) = (10.d0/180.d0)*pi !10 graus
```

```

16  open(10, file = 'saida-tarefaA1-10260434.dat')
do i=0, nmax      !Método de Euler
18      tempo(i+1) = tempo(i) + dt
      omega(i+1) = omega(i) - (g/l)*theta(i)*dt
20      theta(i+1) = theta(i) + omega(i)*dt
      energia(i) = (1.d0/2.d0)*m*omega(i)**2 +
        ↪ ((m*g)/(2.d0*l))*theta(i)**2
22      write(10,*) tempo(i), omega(i), theta(i), energia(i)
end do
24  close(10)
End Program Tarefa_A1

```

1.1.1 Resultados

Temos como primeiro problema neste método numérico o crescimento da amplitude com o tempo, mesmo para pequenas discretizações.

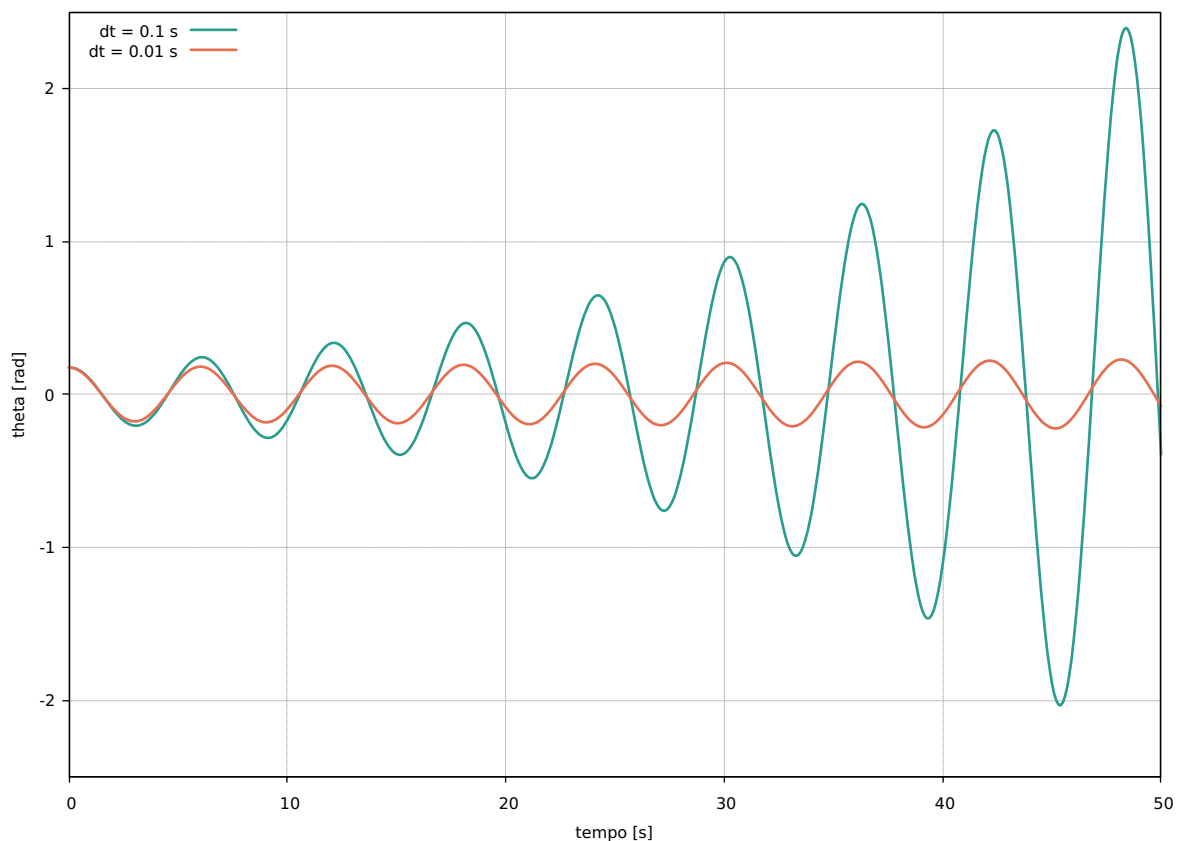


Figura 1: Comparação das simulações para as mesmas condições iniciais, com diferentes discretizações de tempo.

Como visto abaixo, basta aumentar o tempo de execução e o inconveniente aparece novamente. O aumento da amplitude implica que a energia também cresce, quando a mesma deveria ser constante (como consequência física) - não importando a escolha do intervalo de tempo.

Como observamos abaixo, o comportamento da curva indica o crescimento da energia.

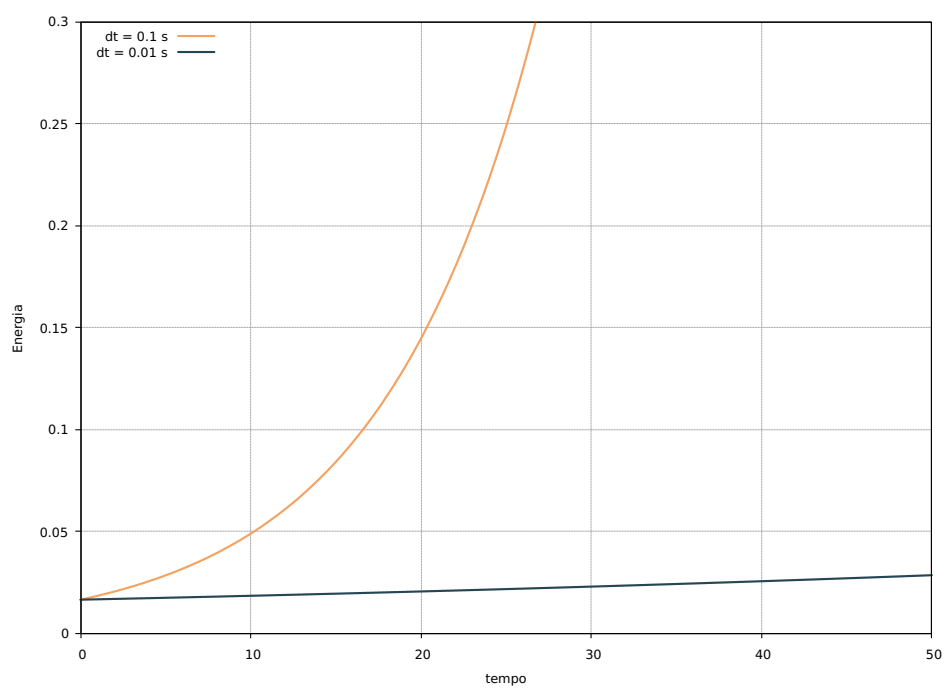


Figura 2: Energia total do sistema.

Para discretizações menores, basta aumentar o tempo de execução que observamos o mesmo comportamento.

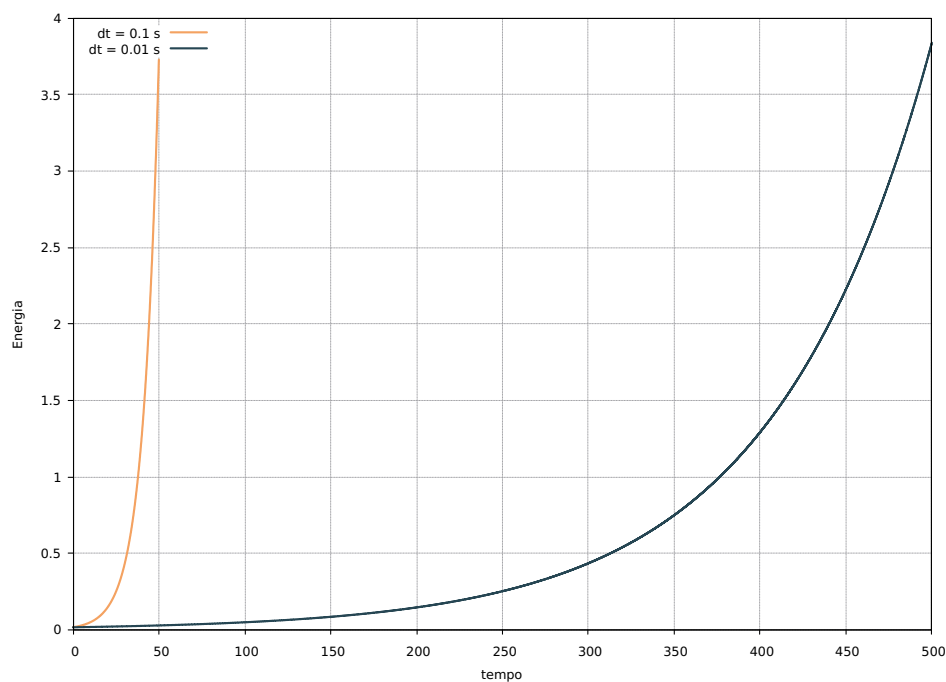


Figura 3: Energia total do sistema.

Isso ocorre pois há um erro sistemático a cada passo, o qual é proporcional a $\frac{1}{N}$, sendo $N = \frac{1}{\Delta t}$. O erro ocasionalmente se acumula e pode se tornar grande o suficiente a ponto de afetar a solução drasticamente para tempos muito longos de execução.

Este método pode ser inadequado para o caso particular de movimentos periódicos, onde a verdadeira natureza do movimento se torna clara apenas após um tempo decorrido, assim, com os erros acumulados é possível que surjam resultados divergentes.

1.2 Subtarefa A2

A fim de corrigir este erro, o algoritmo anterior foi adaptado para o método de Euler-Cromer. Como forma de comparação, foram utilizadas as mesmas condições iniciais anteriores.

tarefaA2-10260434.f90

```

2  Program Tarefa_A2
3
4  IMPLICIT REAL*8 (a-h,o-z) !dupla precisão
5  parameter (nmax = 5000)
6  dimension omega(nmax), theta(nmax), tempo(nmax), energia(nmax)
7
8  pi = 4.d0*datan(1.d0)
9  dt = 0.01           !s
10 a_l = 9.8           !m
11 g = 9.8             !m/s
12 m = 1.d0            !kg
13 tempo(0) = 0.d0      !s
14 omega(0) = 0.d0      !rad/s
15 theta(0) = (10.d0/180.d0)*pi !10 graus
16
17 open(10, file = 'saida-tarefaA2.2-10260434.dat')
18 do i=0, nmax          !Método de Euler-Cromer
19     tempo(i+1) = tempo(i) + dt
20     omega(i+1) = omega(i) - (g/a_l)*theta(i)*dt
21     theta(i+1) = theta(i) + omega(i+1)*dt
22     energia(i) = (1.d0/2.d0)*m*omega(i)**2 +
23         ↪ ((m*g)/(2.d0*a_l))*theta(i)**2
24     write(10,*) tempo(i), omega(i), theta(i), energia(i)
25 end do
26 close(10)
27 End Program Tarefa_A2

```

1.2.1 Resultados

A nova modificação estabelece a conservação da energia em problemas oscilatórios. Embora o erro se mantenha na mesma proporção, o algoritmo se encaixa melhor a movimentos periódicos. Como podemos observar no gráfico a seguir, mesmo alterando a discretização do tempo as trajetórias se mantêm aproximadamente as mesmas.

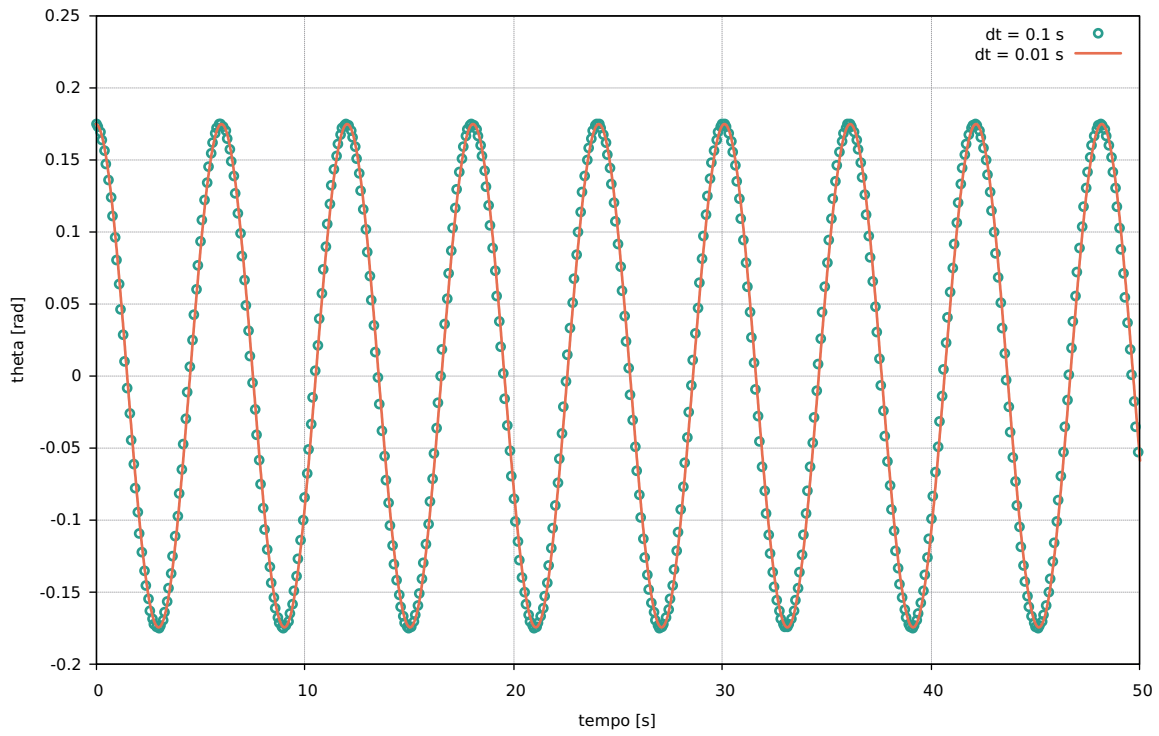


Figura 4: θ vs tempo utilizando o método de Euler-Cromer.

Ao examinar a energia, podemos verificar que ocorre conservação, apenas apresentando um comportamento oscilatório devido a aproximação. As diferenças de amplitude são dadas pelo aumento da precisão conforme se aumenta a discretização.

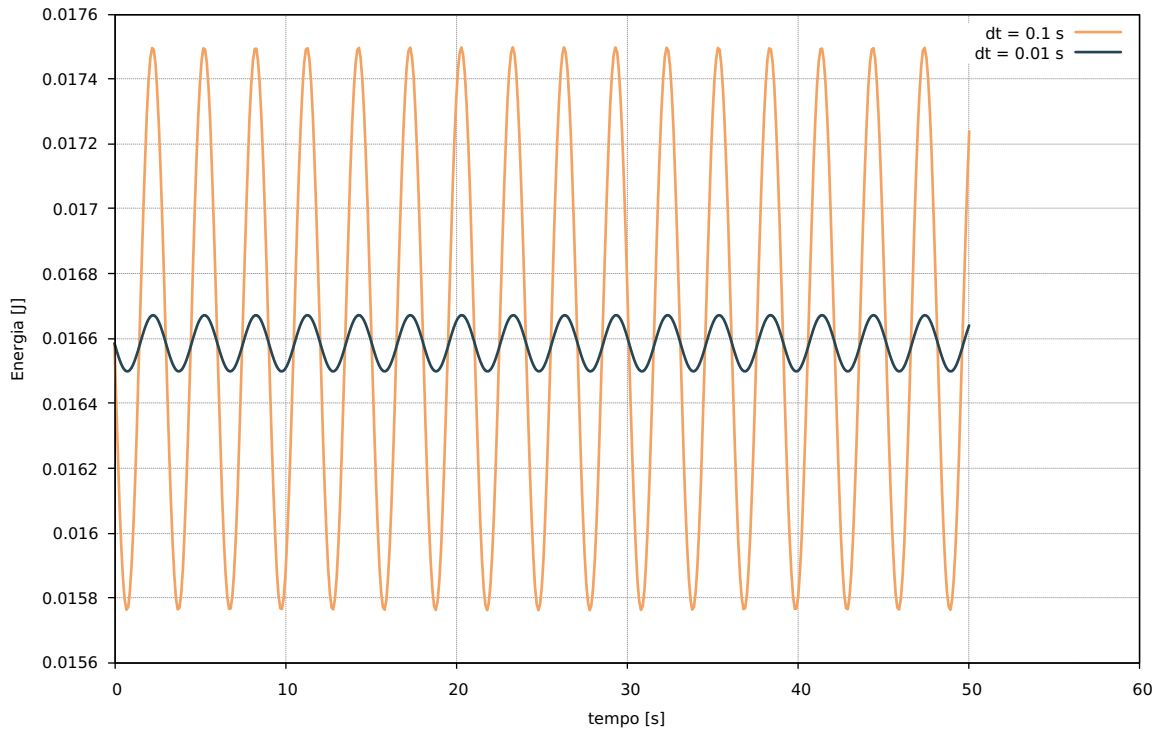


Figura 5: Energia total do sistema.

A conservação de energia a cada passo é lida como:

$$E_{i+1} = E_i + \frac{g}{2l}(\omega_i^2 - \frac{g}{l}\theta_i^2)\Delta t^2 \quad (1)$$

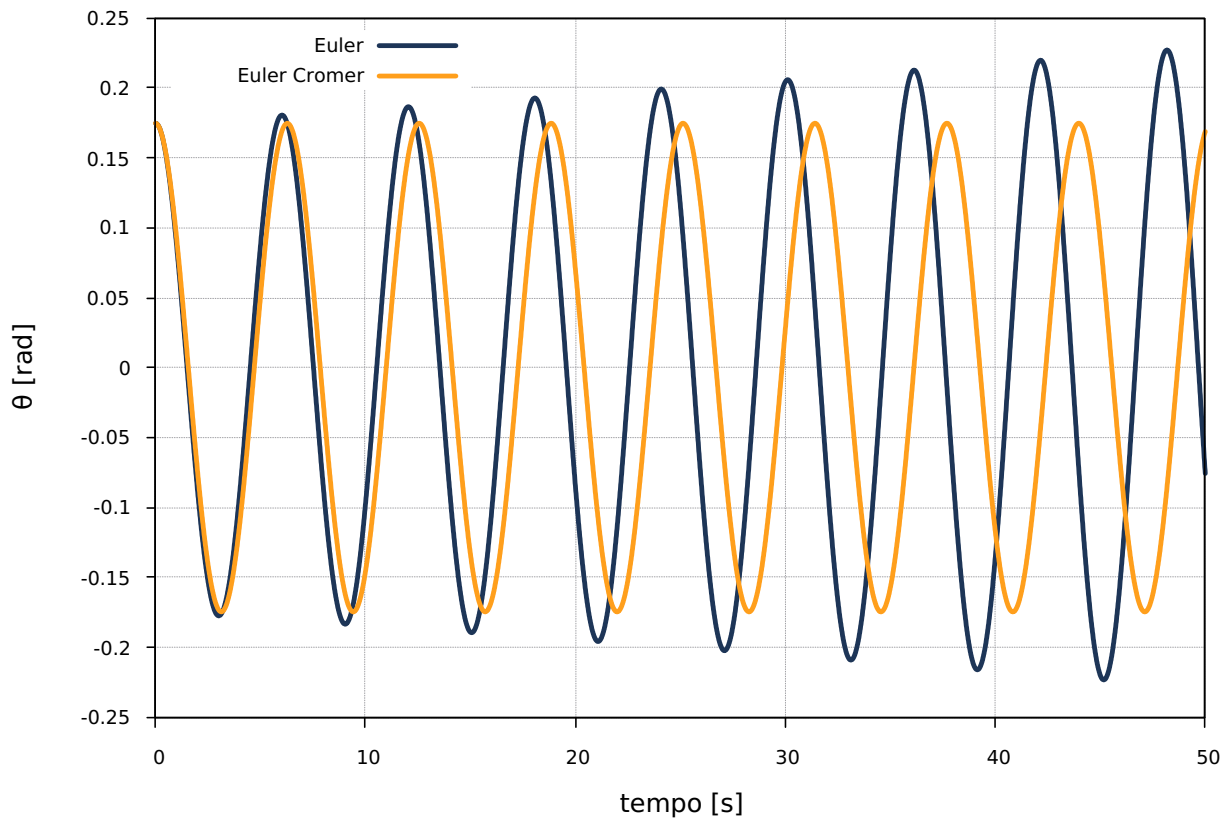
E a energia do pêndulo simples é:

$$E_i = \frac{1}{2}\omega_i^2 + \frac{g}{2l}\theta_i^2 \quad (2)$$

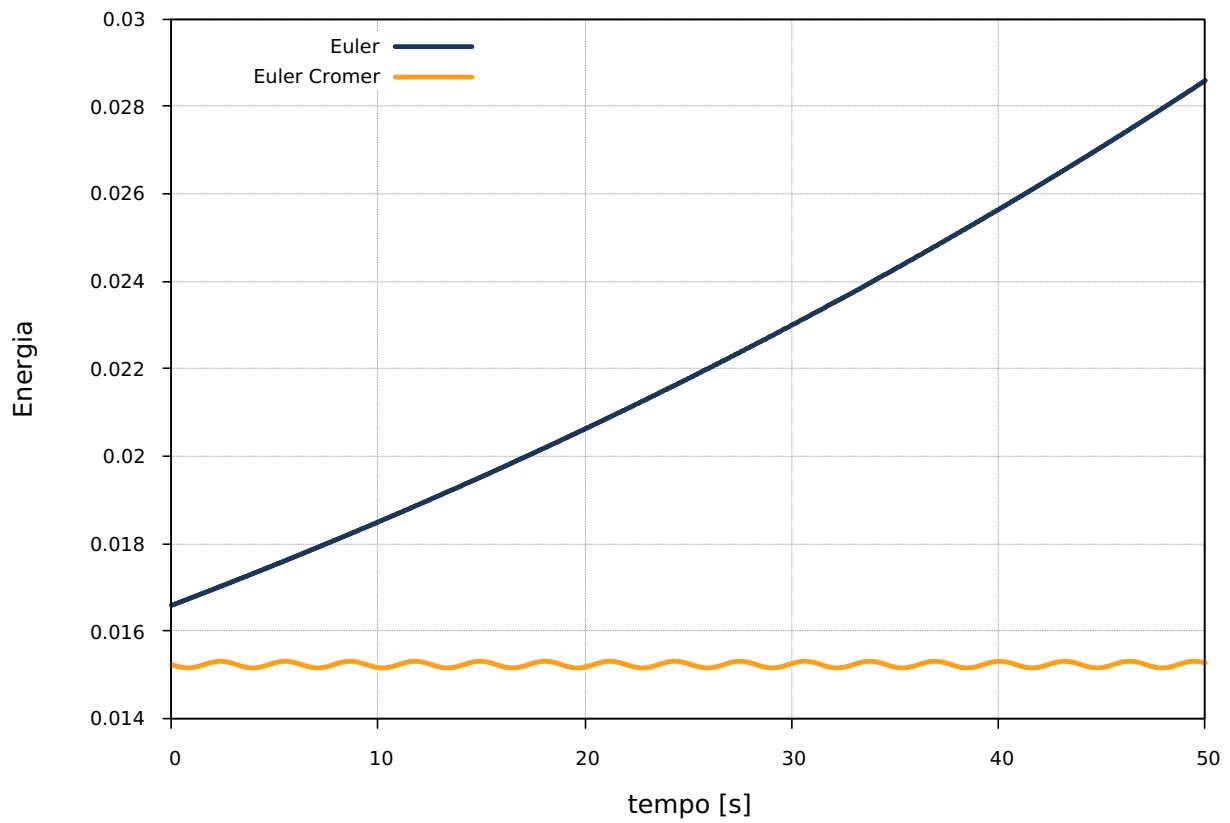
Podemos ver que o erro se mantém o mesmo do método de Euler, no entanto, seu coeficiente se dá precisamente igual a diferença entre as energia potencial e cinética a cada passo i . Neste caso, o erro acumulado a cada passo se suprime pois, em média, as duas energia são iguais, enquanto no algoritmo de Euler esse coeficiente é a soma das energia, não gerando cancelamento algum.

1.3 Comparação entre os métodos

Para demonstrar claramente as diferenças entre os resultados dos dois métodos, podemos comparar graficamente cada solução numérica.



(a) Comparação de θ entre os métodos de Euler e Euler-Cromer.



(b) Comparação da energia entre os métodos de Euler e Euler-Cromer.

Figura 6: Comparação entre os métodos de Euler e Euler-Cromer.

Assim, fica notável a diferença entre os dois métodos, nos indicando que o método de Euler-Cromer é o mais adequado para o problemas oscilatórios.

2 Tarefa B

O seguinte código foi utilizado para análise do período e a presença de uma força externa oscilatória em um pêndulo não-linear (ou seja, não fazemos $\sin(\theta) \approx \theta$). Cada parâmetro foi ajustado a respectiva tarefa sendo executada. O período do pêndulo foi calculado por três diferentes métodos:

- Integral elíptica através da regra de Bode, atentando-se aos pontos de singularidade e adicionando o resultado analítico ao numérico posteriormente. 'Analítico'
- Encontrando o período através dos próprios dados, encontrando o tempo entre dois nós consecutivos e multiplicando por 2 (valor equivalente a um período). 'Numérico'
- Aproximação da equação do período para ângulos pequenos. 'Aproximação'

____ tarefaB-10260434.f90 _____

```
Program Tarefa_B
2
3  IMPLICIT REAL*8 (a-h,o-z) !dupla precisão
4  parameter (nmax = 3000)
5  dimension omega(nmax), theta(nmax), tempo(nmax), time(5)
6
7  f(x) = 1.d0/sqrt( dcos(x)-dcos(theta(0) ))           !integral
8
9  pi = 4.d0*datan(1.d0)
10 dt = 0.03           !s
11 a_l = 9.8           !m
12 g = 9.8             !m/s
13 m = 1.d0            !kg
14 F_0 = 0.0
15 res = 0.5
16 freq = 2.0/3.0
17 tempo(0) = 0.d0     !s
18 omega(0) = 0.d0     !m/s
19 theta(0) = (30.d0/180.d0)*pi      !60 graus
20
21 open(10, file = 'saida-tarefaB4-10260434.dat')
22 do i=0, nmax          !Inclusão das forças
23
24     if (theta(i) .ge. 2.d0*pi) then
25         theta(i) = theta(i) - 2.d0*pi
26     else if (theta(i) .le. -2.d0*pi) then
27         theta(i) = theta(i) + 2.d0*pi
28     end if            !limitação do ângulo 0 < theta < 2*pi
29
30     tempo(i+1) = tempo(i) + dt
31     omega(i+1) = omega(i) - (g/a_l)*sin(theta(i))*dt -
32         ↪ res*omega(i)*dt + F_0*sin(freq*tempo(i))*dt
33     theta(i+1) = theta(i) + omega(i+1)*dt
34     write(10,*) tempo(i), omega(i), theta(i)
```



```

34  end do

36  !regra de Bode p/ o cálculo da integral
N = 3000
38  a = -abs(theta(0)) + 0.001
b = abs(theta(0)) - 0.001
40  h = (b-a)/N
soma = 0.d0

42
do i=0, N-4, 4
44     soma = soma + (2.d0/45.d0)*h*(    7*f(a + i*h) + 32*f(a + (i+1)*h) +
        ↪ 12*f(a + (i+2)*h) &
                                & + 32*f(a + (i+3)*h) +7*f(a + (i+4)*h) )
46 end do
soma = sqrt((2.d0*a_l/g)*(soma + 4.d0*sqrt(0.001/dsin(theta(0) ) ) ) )
48 !inclusão do termo analítico

50 j=0
i=0
52 do while (j .lt. 5)                                !cálculo do período numericamente
    if (theta(i)*theta(i+1) .lt. 0.0) then
54         time(j) = tempo(i)                            !encontra duas raízes
        ↪ consecutivas
            j = j +1
56     end if
    i = i + 1
58 end do

60 ang_peq =
    ↪ 2.d0*pi*sqrt(a_l/g)*(1.d0+theta(0)**2/16.d0)          !aproximação
perodo = (time(4)-time(3))*2.d0                            !período

62
print*, 'Numérico = ',perodo
64 print*, 'Analítico = ',soma
print*, 'Aproximação = ', ang_peq
66
close(10)
68 End Program Tarefa_B

```

2.1 Resultados B1

Os valores abaixo foram obtidos após 1000 iterações, com $dt = 0.01$ e $\theta_0 = 70^\circ$. Podemos observar que os resultados numérico e analítico são próximos, com uma casa decimal de precisão. No entanto, o método que utiliza a Aproximação já se mostra não muito preciso, uma vez que ele é válido apenas para ângulos pequenos, tornando não adequado seu uso as condições de teste propostas inicialmente.

Método	T [s]
Numérico	6.9199998453259468
Integral elíptica	6.9287472083301678
Aproximação	6.8693379141759943

2.2 Resultado B2

Aplicando o mesmo teste anterior, porém alterando o ângulo inicial para 10° podemos ver que todos os resultados se mostraram suficientes, com 2 casas decimais de precisão. Desta vez, o uso da aproximação foi adequado ao teste, uma vez que sua utilização está dentro de sua proposta, se mostrando eficiente para ângulos iniciais menores.

Método	T [s]
Numérico	6.2999998591840267
Integral elíptica	6.2949734529422878
Aproximação	6.2951476052815529

2.3 Resultados B3

Aqui estudaremos o efeito de forças dissipativas, incluindo a resistência do ar a simulação. Foram realizadas 2000 iterações, com $\theta_0 = 30^\circ$. Com isso, ocorre o **amortecimento**, onde a energia armazenada na oscilação é dissipada, gerando a redução e conseqüente impedimento do movimento.

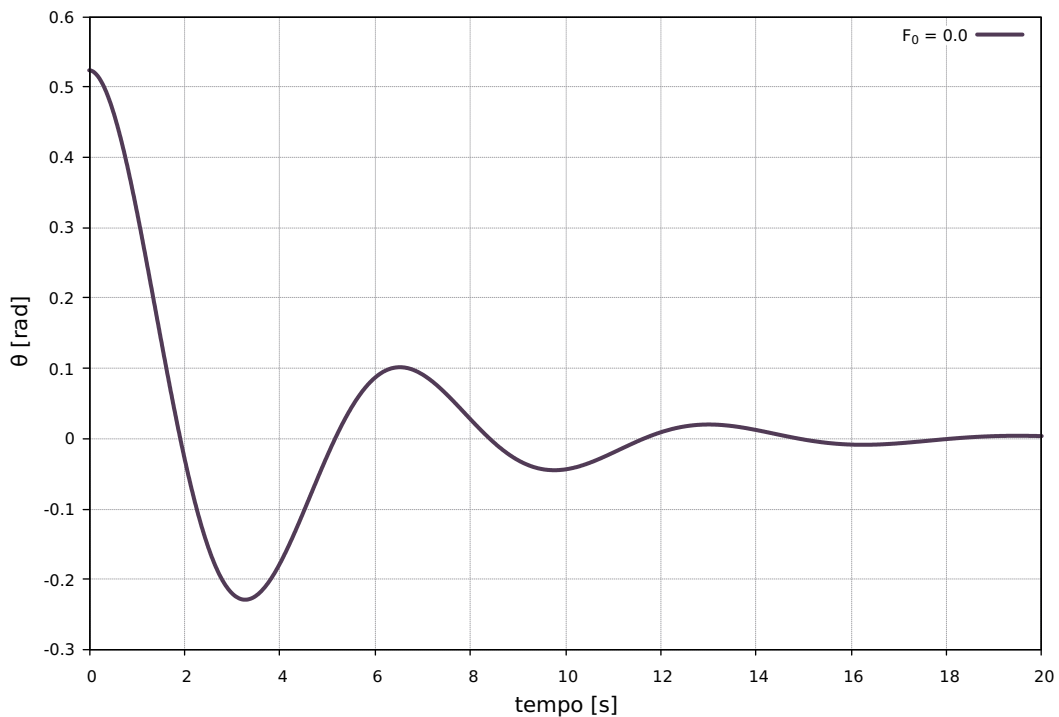


Figura 7: Amortecimento subcrítico.

Podemos observar que o gráfico possui comportamento de um **amortecimento subcrítico**.

2.4 Resultados B4

Neste caso, além da inclusão de uma força dissipativa, foi incluída uma força externa oscilatória atuando no pêndulo.

- $F_0 = 0.0 \text{ rad/s}^2$

Este caso apresenta o comportamento do pêndulo quando há forças dissipativas, como a resistência do ar. Portanto, a amplitude diminui com o tempo e algum tipo de amortecimento se manifesta. Aqui temos o **amortecimento subcrítico**, como na subarefa B3.

- $F_0 = 0.5 \text{ rad/s}^2$

A oscilação de mantém constante pois a força oscilatória mantém a energia do sistema constante, mesmo com a presença da resistência do ar. Assim, temos uma **oscilação forçada**.

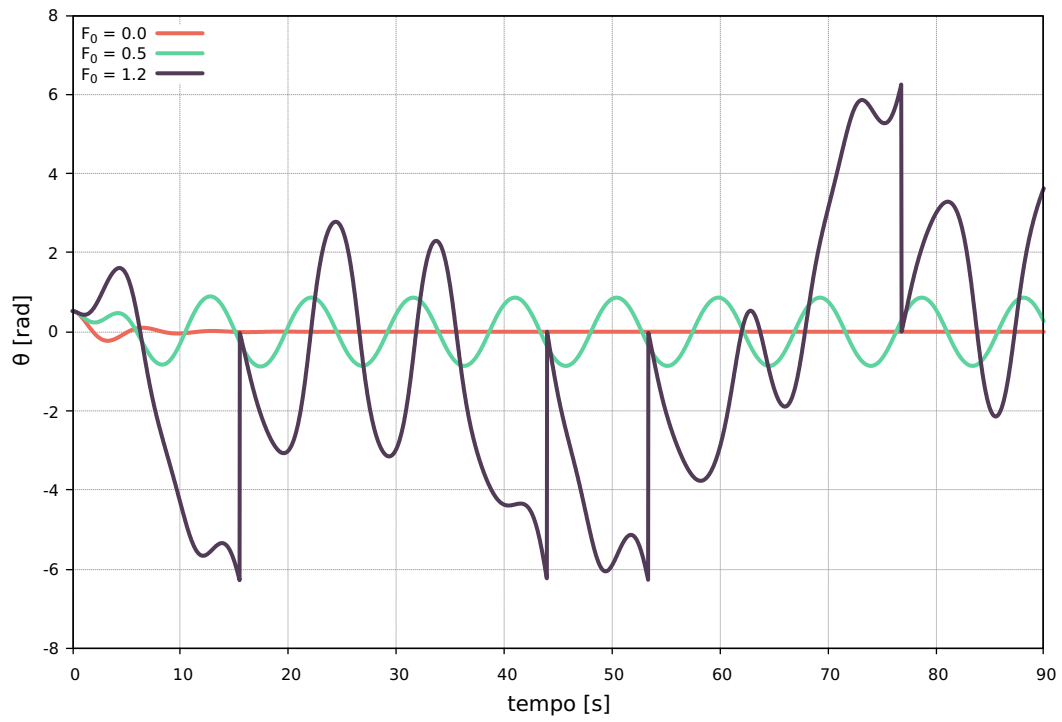
Quanto a frequência, devemos notar que ela é igual ao da força, e não a frequência natural do pêndulo.

$$T = \frac{1}{f} 2\pi \quad (3)$$

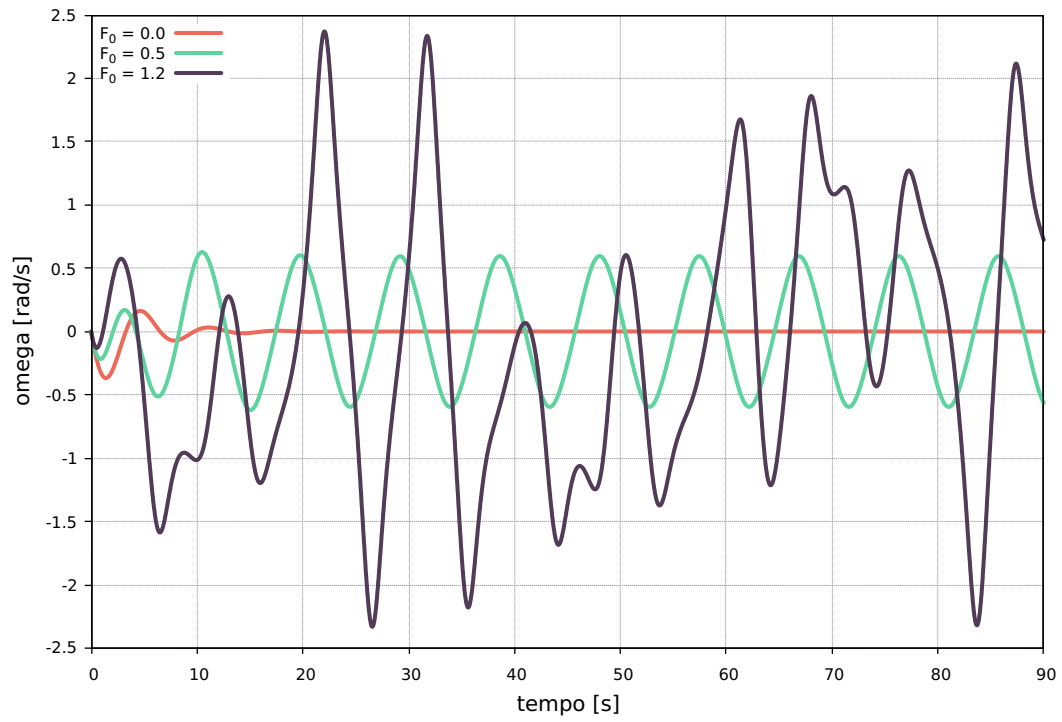
Como a $freq = 2/3 \text{ Hz}$, o período será $T_{analitico} = 9.42477796076938 \text{ s}$. O resultado obtido numericamente foi $T_{numerico} = 9.4799997881054878 \text{ s}$, portanto, concluímos que obtivemos um resultado satisfatório e correspondente ao fenômeno físico.

- $F_0 = 1.2 \text{ rad/s}^2$

Aparecimento do movimento caótico, não sendo possível determinar seu período. Foi colocado dessa forma pois nenhum tipo de padrão parece se manifestar e θ não aparenta ser previsível.



(a) Comportamento de θ no tempo.



(b) Comportamento da velocidade angular no tempo.

Figura 8: Comportamento do pêndulo com atuação de uma força oscilatória.

3 Tarefa C

O algoritmo anterior foi adaptado para incluir o segundo pêndulo, mantendo a mesma evolução temporal com exceção do ângulo inicial, como destacado no código abaixo.

Os dados da diferença entre os ângulos θ (dtheta(i)) já foram calculados em logaritmo natural para a obtenção do **expoente de Lyapunov**.

tarefaC-10260434.f90

```
Program Tarefa_C
```

```
IMPLICIT REAL*8 (a-h,o-z) !dupla precisão
```

```
parameter (nmax = 2000)
```

```
dimension omega(nmax), theta(nmax), tempo(nmax), omega1(nmax),  
↳ theta1(nmax), dtheta(nmax)
```

```
!Inclusão do segundo pêndulo, diferentes condições iniciais
```

```
!porém mesmo contador de tempo
```

```
pi = 4.d0*datan(1.d0)
```

```
dt = 0.03 !s
```

```
a_l = 9.8 !m
```

```
g = 9.8 !m/s
```

```
m = 1.d0 !kg
```

```
F_0 = 0.5
```

```
res = 0.5
```

```
freq = 2.0/3.0
```

```
tempo(0) = 0.d0 !s
```

```
omega(0) = 0.d0 !m/s
```

```
theta(0) = (30.d0/180.d0)*pi !30 graus
```

```
omega1(0) = 0.d0 !m/s
```

```
theta1(0) = theta(0) - 0.001
```

```
open(10, file = 'saida-tarefaC-10260434.dat')
```

```
do i=0, nmax
```

```
tempo(i+1) = tempo(i) + dt
```

```
omega(i+1) = omega(i) - (g/a_l)*dsin(theta(i))*dt -
```

```
↳ res*omega(i)*dt + F_0*dsin(freq*tempo(i))*dt
```

```
theta(i+1) = theta(i) + omega(i+1)*dt
```

```
omega1(i+1) = omega1(i) - (g/a_l)*dsin(theta1(i))*dt -
```

```
↳ res*omega1(i)*dt + F_0*dsin(freq*tempo(i))*dt
```

```
theta1(i+1) = theta1(i) + omega1(i+1)*dt
```

```
dtheta(i) = abs( theta1(i)-theta(i) )
```

```
write(10,*) tempo(i), dlog(dtheta(i)) !dados já em
```

```
↳ logaritmo natural
```

```
end do
```

```
close(10)
```

```
End Program Tarefa_C
```

3.1 Resultados

O fitting dos dados foi feito através do Gnuplot, utilizando como base que

$$\Delta\theta(t) \approx e^{\lambda t} \quad (4)$$

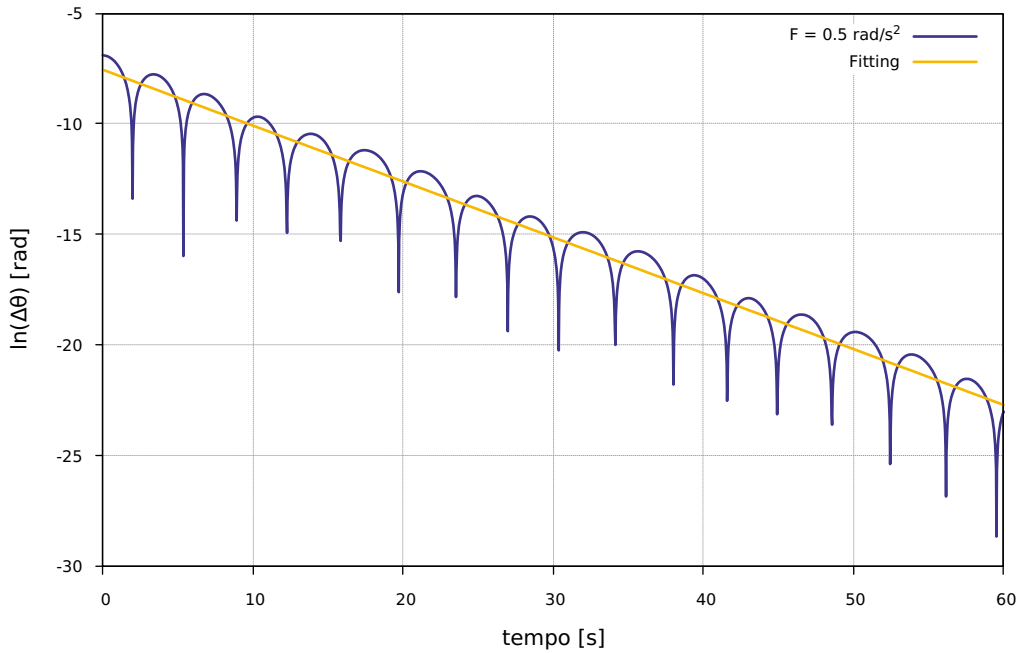
sendo λ o expoente de Lyapunov. Ao aplicar \ln em ambos os lados, teremos

$$\ln(\Delta\theta) = \lambda t \quad (5)$$

Portanto, para um gráfico semi-logarítimo de $\ln(\Delta\theta)$ vs tempo, temos como função base para o fitting dos dados uma reta, dada pela equação

$$f(x) = ax + b \quad (6)$$

Para o caso em que $F_0 = 0.5 \text{ rad/s}^2$, o movimento não é caótico. Portanto, esperávamos uma tendência de decrescimento na reta (pequenas diferenças nas condições iniciais se mantêm pequenas), com coeficiente 'a' negativo, como foi obtido.



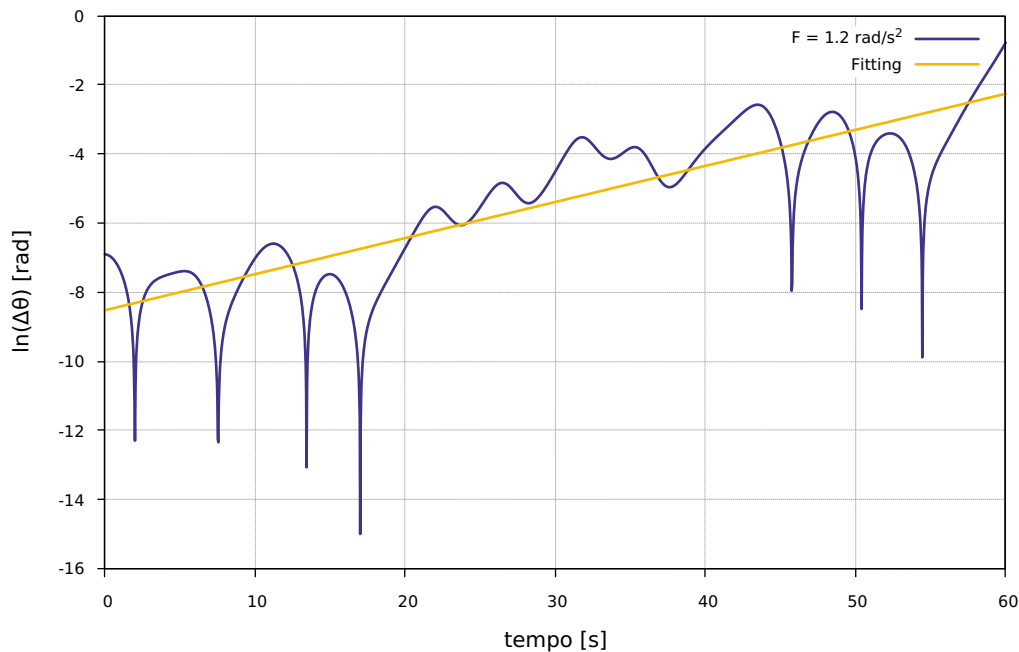
(a) Fitting da curva por uma reta em amarelo.

Final set of parameters		Asymptotic Standard Error	
=====		=====	
a	= -0.252585	+/- 0.001191	(0.4714%)
b	= -7.55802	+/- 0.04125	(0.5458%)

(b) Coeficientes obtidos, sendo 'a' corresponde ao expoente de Lyapunov.

Figura 9: Estimativa do expoente de Lyapunov.

Já para $F_0 = 1.2 \text{ rad/s}^2$, esperávamos uma tendência de crescimento da reta (pequenas diferenças se amplificam rapidamente), com coeficiente 'a' positivo devido ao movimento caótico. Também se obteve o resultado esperado.



(a) Fitting da curva por uma reta em amarelo..

Final set of parameters		Asymptotic Standard Error	
=====		=====	
a	= 0.104417	+/- 0.001444	(1.383%)
b	= -8.5259	+/- 0.05002	(0.5867%)

(b) Coeficientes obtidos, sendo 'a' corresponde ao expoente de Lyapunov.

Figura 10: Estimativa do expoente de Lyapunov.

Portanto, concluímos que é possível verificar qual será a evolução do sistema (caótico ou não) através do expoente de Lyapunov.

Podemos observar que pequenas diferenças nas condições iniciais podem se amplificar exponencialmente gerando soluções divergentes, manifestando o **caos determinístico** - mesmo que a princípio o sistema possa ser perfeitamente determinístico, seu comportamento é completamente imprevisível na prática.

Essas conclusões implicam que não podemos obter nenhum tipo de resposta numérica, uma vez que erros mínimos são amplicados catastróficamente. Por mais preciso que o método seja, ainda existirão erros que se acumularão durante a simulação, tornando qualquer solução numérica insuficiente para a descrição do sistema.

3.2 Tarefa D

Utilizando a mesma estrutura dos algoritmos anteriores foram colocados no arquivo de saída os valores de `theta(i)` e `omega(i)` para elaboração dos diagramas de espaço de fase - uma importante ferramenta para a caracterização de sistemas dinâmicos.

tarefaD-10260434.f90

```
Program Tarefa_D
2
3  IMPLICIT REAL*8 (a-h,o-z) !dupla precisão
4  parameter (nmax = 8000)
5  dimension omega(nmax), theta(nmax), tempo(nmax)
6
7  !Confeção do gráfico omega vs theta
8  pi = 4.d0*datan(1.d0)
9  dt = 0.03          !s
10 a_l = 9.8          !m
11 g = 9.8            !m/s
12 m = 1.d0          !kg
13 F_0 = 0.5
14 res = 0.5
15 freq = 2.0/3.0
16 tempo(0) = 0.d0    !s
17 omega(0) = 0.d0     !m/s
18 theta(0) = (30.d0/180.d0)*pi    !30 graus
19
20 open(10, file = 'saida-tarefaD1.2-10260434.dat')
21 do i=0, nmax
22
23     if (theta(i) .ge. 2.d0*pi) then
24         theta(i) = theta(i) - 2.d0*pi
25     else if (theta(i) .le. -2.d0*pi) then
26         theta(i) = theta(i) + 2.d0*pi
27     end if          !limitação do ângulo 0 < theta < 2*pi
28
29     tempo(i+1) = tempo(i) + dt
30     omega(i+1) = omega(i) - (g/a_l)*dsin(theta(i))*dt -
31         ↪ res*omega(i)*dt + F_0*dsin(freq*tempo(i))*dt
32     theta(i+1) = theta(i) + omega(i+1)*dt
33     write(10,*) omega(i) , theta(i)
34 end do
35 close(10)
36 End Program Tarefa_D
```

3.3 Resultados

Podemos observar que a curva corresponde a trajetória do pêndulo não-linear e que pequenas mudanças nas condições iniciais não alteram o comportamento do elipsóide, chamado de atrator periódico.

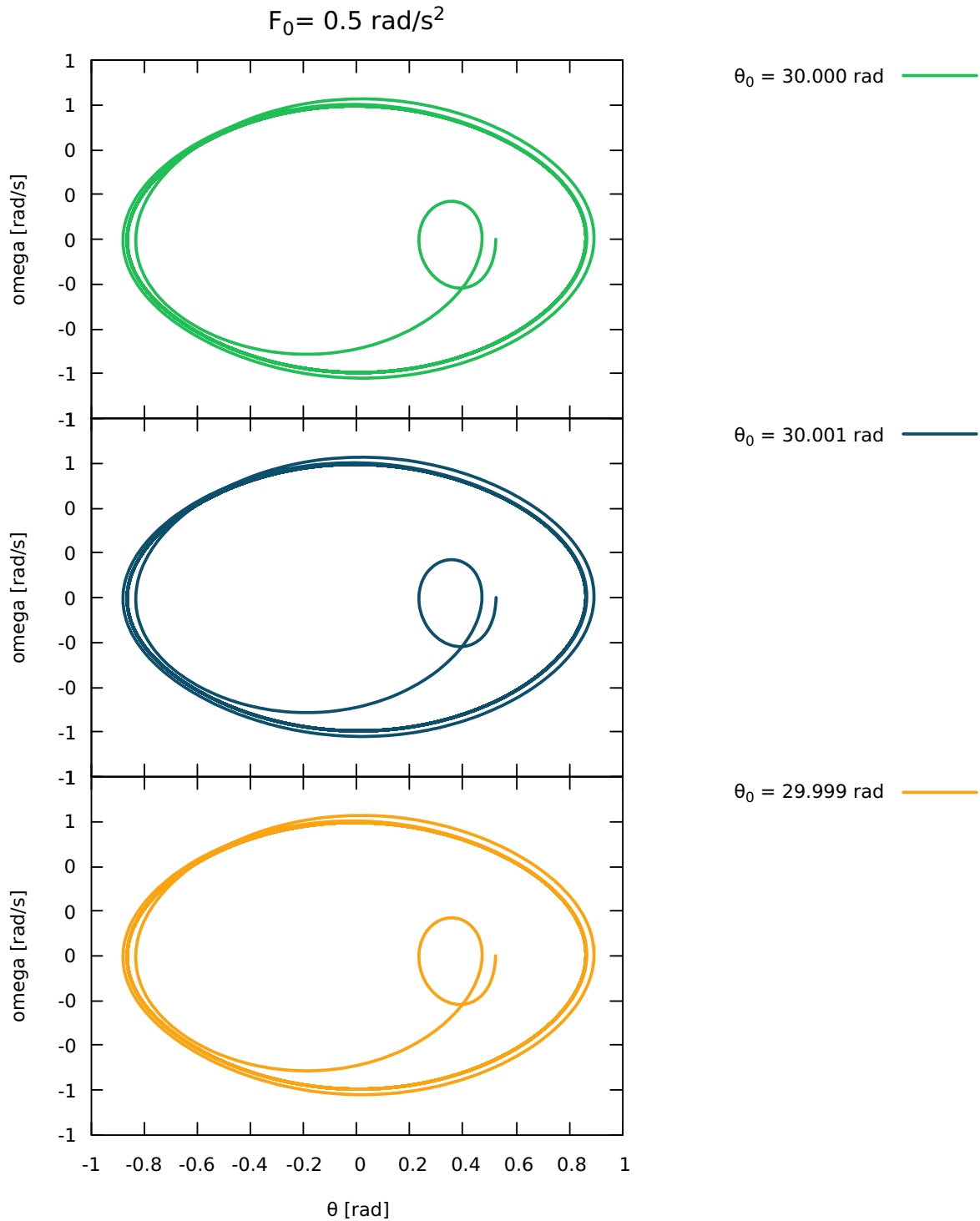


Figura 11: Diagrama de fase para diferentes angulos iniciais.

Quanto ao movimento não caótico (Figura 12), apesar das expectativas serem de não reconhecermos algum tipo de padrão, podemos notar que há certa estrutura se manifestando. Deste modo, temos como resultado um **atrator estranho** proveniente do movimento caótico.

Primeiramente, percebe-se que há uma mudança no formato da elipse, um deslocamento para a esquerda - quebrando a simetria com o eixo vertical -, assim como o surgimento de mais atratores.

O que presenciamos na figura abaixo corresponde a várias oscilações em torno de pontos de equilíbrio estável desse pêndulo. O pêndulo oscila algumas vezes em torno dessa posição -

com diferentes amplitudes - e, em algumas delas a amplitude aumenta tanto ao ponto em que o objeto descreve uma volta completa. Após quantas voltas, ou em qual sentido o pêndulo seguirá não parece existir nenhum padrão identificável.

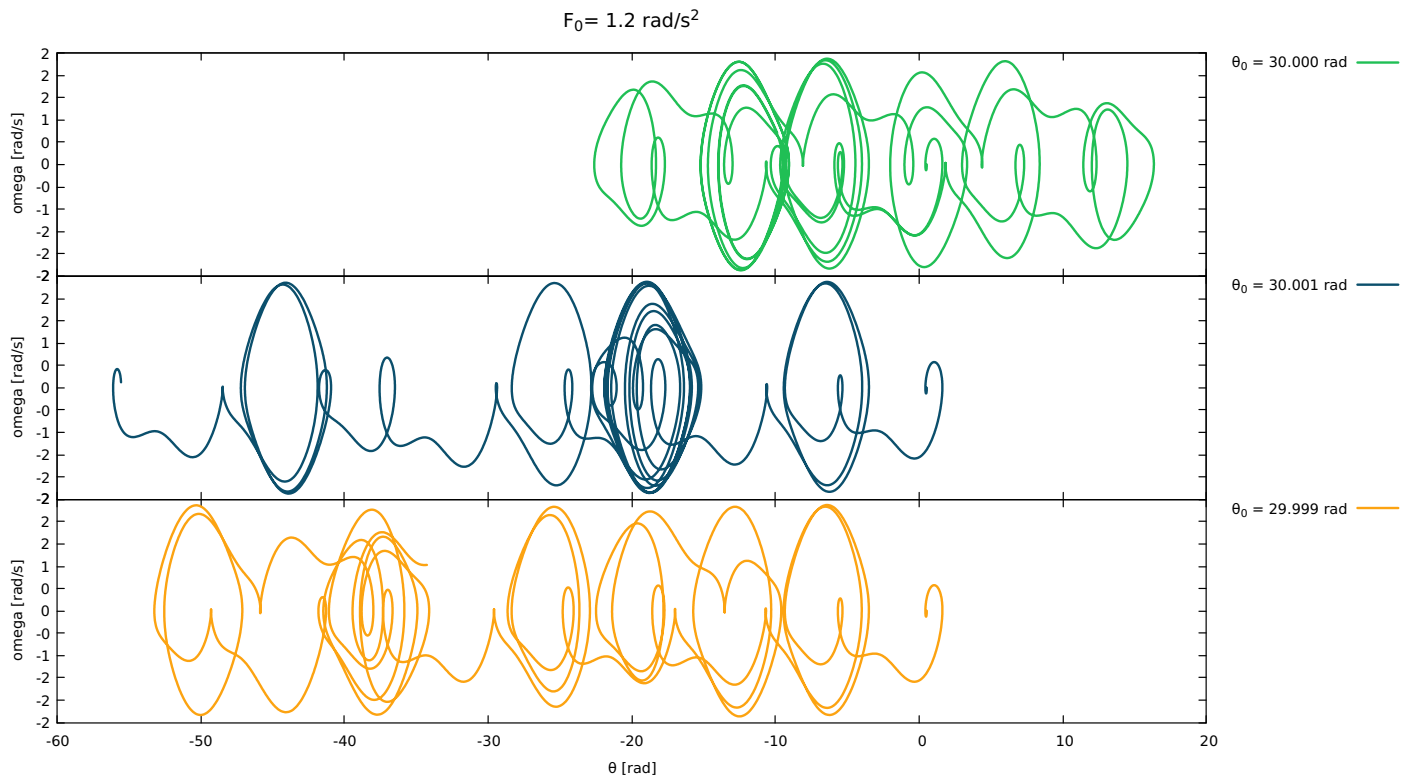


Figura 12: Diagrama de fase para diferentes angulos iniciais.

O que podemos pontuar é que o movimento não é aleatório, sendo determinado pelas equações de evolução. Porém, dada a mínima diferença nas condições iniciais, trajetórias totalmente distintas são evidenciadas.

4 Tarefa E

O código anterior foi adaptado para evidenciar as seções de Poincaré, apenas incluindo sua condição, fazendo com que apenas os pontos dentro dela sejam guardados no arquivo.

Essa condição apenas determina um plano o qual irá conter todos os pontos em que a trajetória do espaço de fase o cruza indo num sentido específico.

tarefaE-10260434.f90

```
Program Tarefa_E
2
3  IMPLICIT REAL*8 (a-h,o-z) !dupla precisão
4  parameter (nmax = 100000)
5  dimension omega(nmax), theta(nmax), tempo(nmax)
6
7  pi = 4.d0*datan(1.d0)
8  dt = 0.03 !s
9  a_l = 9.8 !m
10 g = 9.8 !m/s
11 m = 1.d0 !kg
12 F_0 = 1.2
13 res = 0.5
14 tempo(0) = 0.d0 !s
15 freq = 2.0/3.0
16 omega(0) = 0.d0 !m/s
17 theta(0) = (30.d0/180.d0)*pi - 0.001 !30 graus
18
19 open(10, file = 'saida-tarefaE3.2-10260434.dat')
20
21 i_n = 1
22 do i = 0, nmax
23
24     if (theta(i) .ge. 2.d0*pi) then
25         theta(i) = theta(i) - 2.d0*pi
26     else if (theta(i) .le. -2.d0*pi) then
27         theta(i) = theta(i) + 2.d0*pi
28     end if !limitação do ângulo 0 < theta < 2*pi
29
30     tempo(i+1) = tempo(i) + dt
31     omega(i+1) = omega(i) - (g/a_l)*dsin(theta(i))*dt -
32         ↪ res*omega(i)*dt + F_0*dsin(freq*tempo(i))*dt
33     theta(i+1) = theta(i) + omega(i+1)*dt
34     !seções de Poincaré - condição
35     if (abs((tempo(i+1)-(pi*i_n)/freq)) .lt. dt/2.0) then
36         write(10,*) omega(i) , theta(i)
37         i_n = i_n + 1
38     end if
39 end do
40 close(10)
```

4.1 Resultados

Podemos verificar que para o caso determinístico apenas alguns pontos são evidenciados, interseccionando o plano repetidamente sempre nos mesmos locais, indicando o movimento periódico.

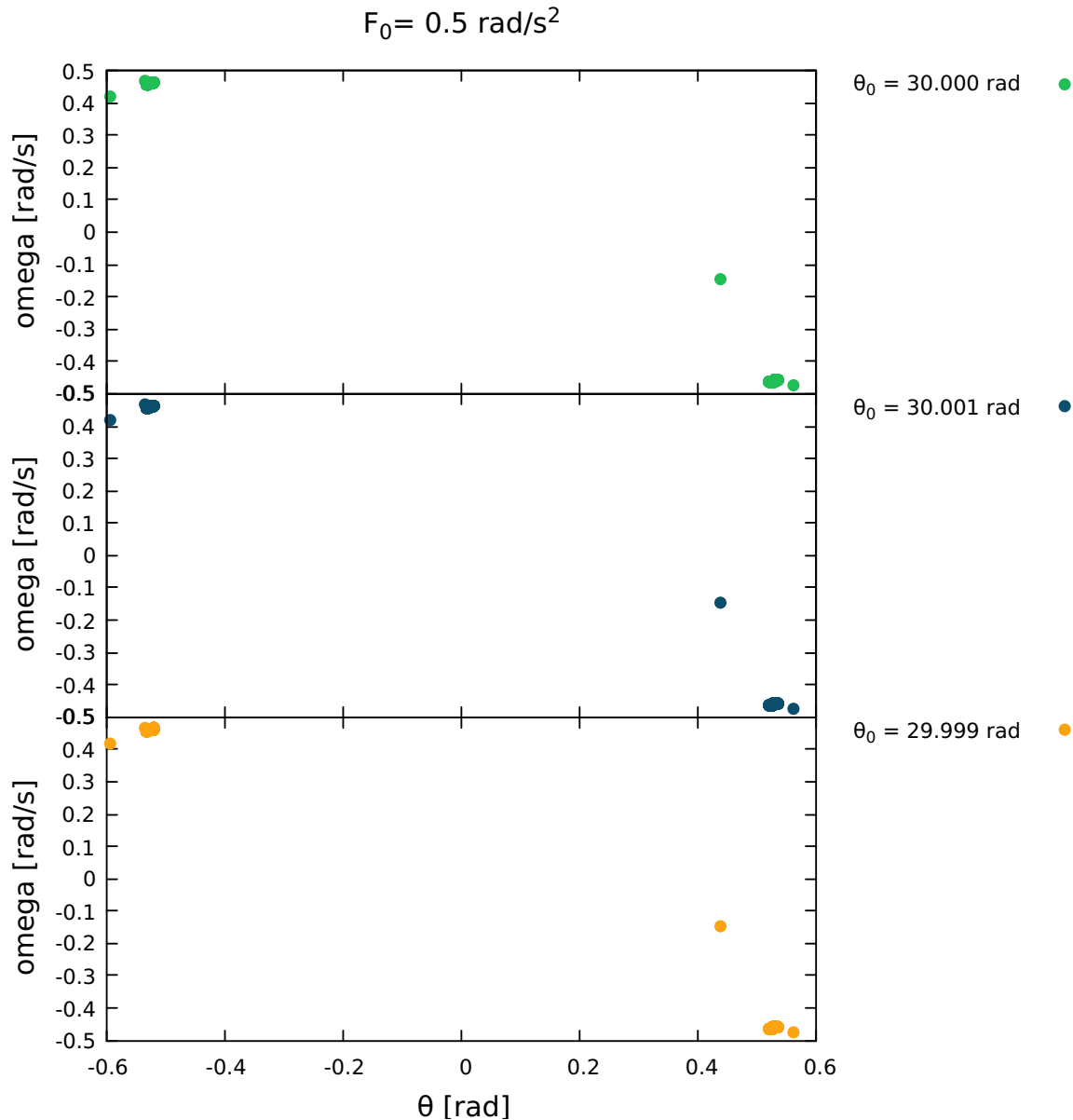


Figura 13: Seção de Poincaré para o movimento não caótico.

Para o segundo caso temos que há um padrão no diagrama de fases, indicando que apesar do regime ser caótico, ele não é randômico - manifestando certa 'universalidade'. Além disso, as curvas apontam que sua construção é independente das condições iniciais e podem conter a estrutura de um fractal. Esta curva é chamada de atrator estranho.

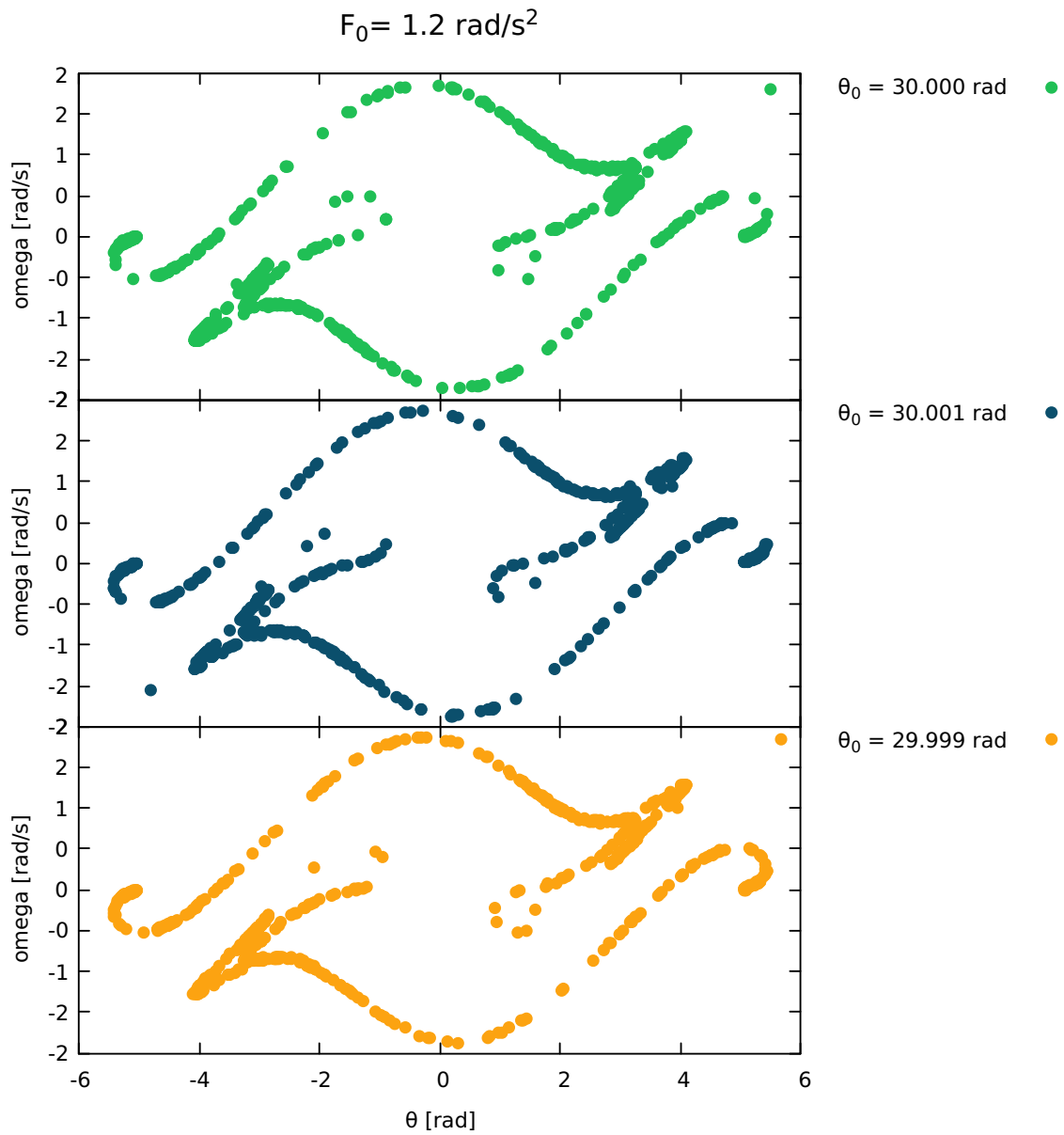


Figura 14: Seção de Poincaré para o movimento caótico.

O segundo ponto a se notar é a necessidade do aumento do número de iterações para a obtenção da Figura 14. Isso porque um atrator estranho é uma curva de evolução infinita, sem princípio nem fim, a qual ocupa uma região do espaço de fase - consequência do movimento caótico, o qual nunca se repete.

Referências

RUSSELL, Daniel A. Acoustics and vibration animations. **Graduate Program in Acoustics, The Pennsylvania State University**, 2014.

YDRI, Badis. Computational physics: an introduction to Monte Carlo simulations of matrix field theory. **arXiv preprint arXiv:1506.02567**, 2015.

YDRI, Badis; BOUCHAREB, Adel; CHEMAM, Rafik. Lectures on Computational Physics. **Badji Mokhtar University, Annaba, Algeria**, 2013.