



**UNIVERSIDADE DE SÃO PAULO
FACULDADE DE FILOSOFIA, CIÊNCIAS E LETRAS DE RIBEIRÃO PRETO
DEPARTAMENTO DE FÍSICA
BACHARELADO EM FÍSICA MÉDICA
5910154 - FÍSICA COMPUTACIONAL**

Projeto para a disciplina de Física Computacional

SIMULAÇÃO DO MODELO NEURONAL DINÂMICO DE HINDMARSH-ROSE

ERICK ALMEIDA E PETRUS KIRSTEN

RIBEIRÃO PRETO, SP
JUNHO DE 2021

RESUMO

Os potenciais de ação das membranas das células nervosas podem ser modelados matematicamente. O modelo de Hindmarsh-Rose (HR) é um sistema dinâmico com a proposta de descrever essa atividade celular. Deste modo, o objetivo do presente projeto é estudar e simular o comportamento do sistema dinâmico de HR com base no trabalho de Lima et al.. O principal parâmetro a avaliar no modelo HR é a intensidade de estimulação externa I . Simulamos o modelo para três diferentes valores de I : 1,1, 1,2 e 3,0. Observamos que, para $I = 1,1$, o sistema aproxima de um ponto fixo; para $I = 1,2$ o sistema aproxima de um ciclo limite; e para $I = 3,0$ o sistema é um atrator caótico. Os resultados obtidos apenas se divergem com o resultado esperado para $I = 1,3$, pois, pelo artigo base, o sistema já deveria se mostrar caótico. Porém, o sistema se demonstrou caótico para valores de I maiores que 3,0.

Palavras-chave: sistema dinâmico; potencial de ação; modelo de HR.

SUMÁRIO

1	INTRODUÇÃO	3
1.1	CONCEITOS	3
1.1.1	Células nervosas	3
1.1.2	Potencial de ação	3
1.1.3	Modelo de Hindmarsh-Rose	5
1.2	MOTIVAÇÃO	5
1.3	OBJETIVOS	5
2	METODOLOGIA	6
3	RESULTADOS E DISCUSSÃO	7
3.1	ANÁLISE DA VARIAÇÃO DO PARÂMETRO DE PERTURBAÇÃO EX- TERNA I	7
3.2	ANÁLISE DE ESTABILIDADE	9
4	CONCLUSÕES	11
	REFERÊNCIAS	12
	APÊNDICE A – CÓDIGO FONTE DA SIMULAÇÃO DO SISTEMA DINÂMICO	13

1 INTRODUÇÃO

1.1 CONCEITOS

1.1.1 Células nervosas

O sistema nervoso é responsável em regular as funções e atividade no nosso organismo. O modo em que tal sistema faz esse controle é através das transmissões de pulsos elétricos e neurotransmissores pelo tecido nervoso. As células desse tecido responsável pela recepção e transmissão dos estímulos são os *neurônios*.

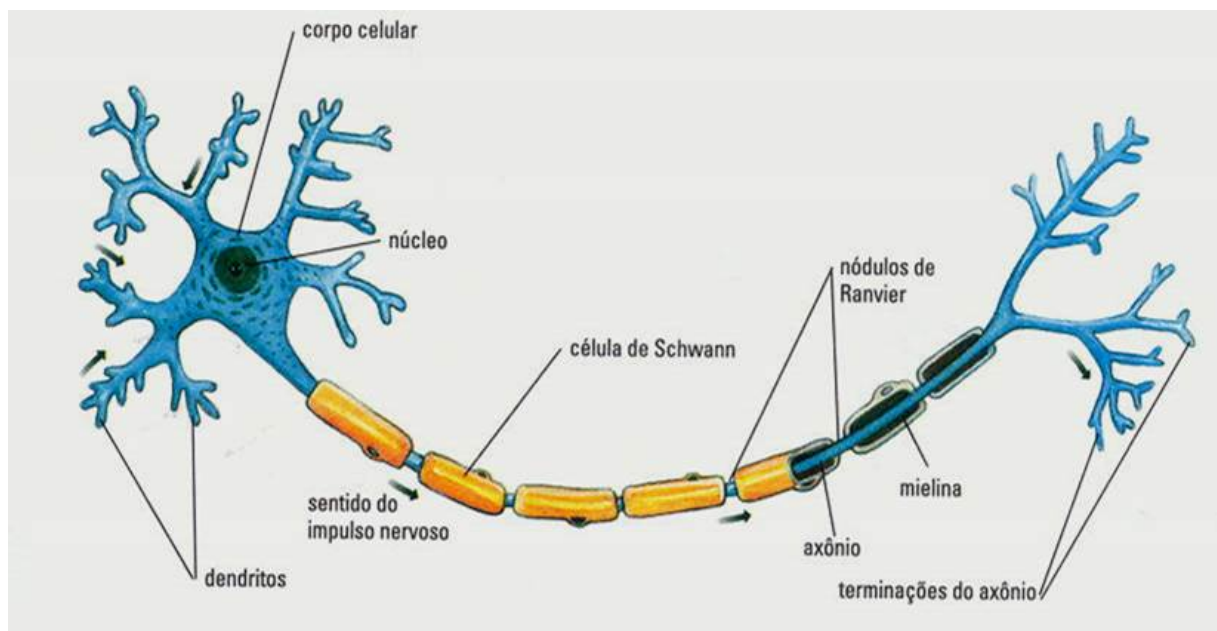


Figura 1 – Esquema de um neurônio. As setas simulam o sentido do impulso nervoso pela célula, que é um efeito da propriedade de condutibilidade.

Fonte: Retirado de: Histologia Interativa (UNIFAL,).

Podemos identificar duas propriedades fundamentais nos neurônios que possibilitam estudar suas funções: *excitabilidade* e *condutibilidade*. Quando falamos em excitabilidade, nos referimos à capacidade da célula em responder estímulos. Já a condutibilidade é a própria resposta dos neurônios. Essa resposta pode ser avaliada como sendo uma corrente elétrica em um fio condutor. Quando as células nervosas são excitadas, elas transmitem essa onda de excitação através de sua extensão, principalmente pelos axônios - ver figura (1). Esta onda de excitação chamamos de *impulso elétrico*.

1.1.2 Potencial de ação

De modo geral, as células possuem uma diferença de potencial entre o seu interior e o exterior, que chamamos de *potencial de membrana*. Numa célula animal,

essa tensão costuma valer -70 mV , ou seja, o interior da célula possui uma voltagem negativa em relação ao exterior. Normalmente, o potencial de membrana permanece constante. Entretanto, como os neurônios são células eletricamente ativas, podem haver rápidos picos e vales de tensão devido ao impulso elétrico. Esses pulsos elétricos são chamados de *potenciais de ação* (PA) que, nas células nervosas, podem durar milissegundos.

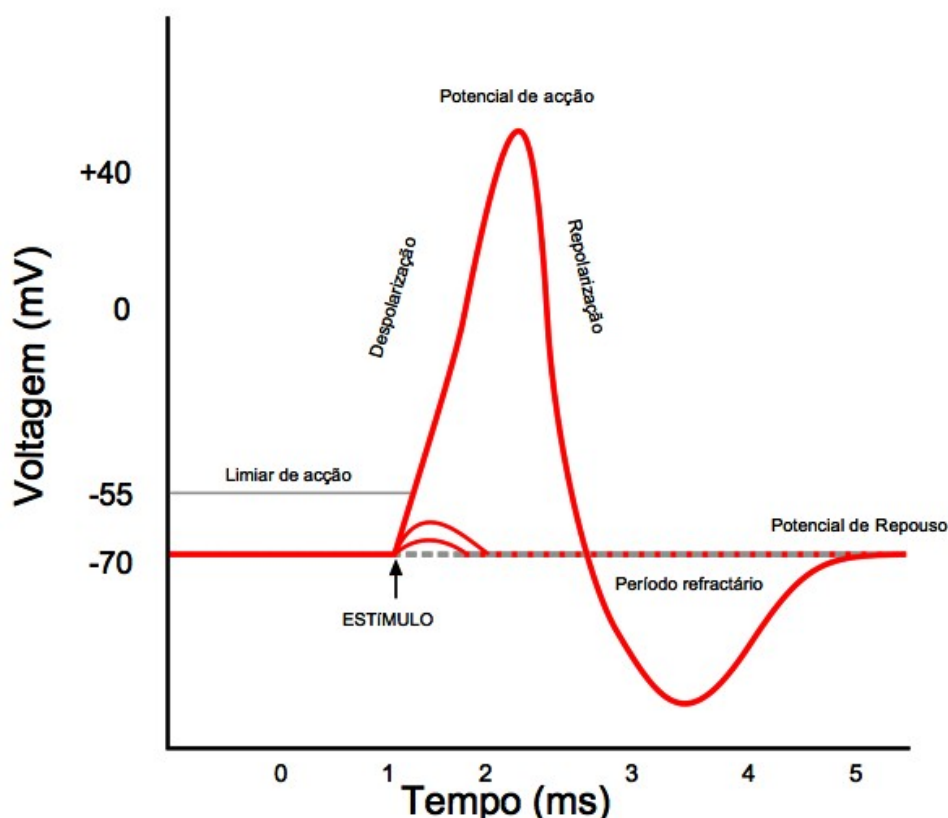


Figura 2 – Comportamento da diferença de potencial em função do tempo de uma célula quando há um potencial de ação. O potencial de repouso é a tensão padrão de uma célula. Quando há um estímulo, se for maior que o limiar de ação, pode haver um potencial de ação que dispara a despolarização da célula. Em seguida, ocorre a repolarização, imediatamente após, há um período refratário, em que a diferença de potencial fica rapidamente menor que o potencial de repouso padrão.

Fonte: Retirado de: (MOREIRA,).

É possível identificar comportamentos típicos da voltagem no decorrer ao tempo nos PA's. Antes da célula receber um estímulo, há o potencial de repouso, que é a tensão padrão da célula. Ao receber algum tipo de estímulo, se este for maior que o limiar de ação da célula, a mesma pode despolarizar. Este é o momento do pico da tensão, que faz com que inverta a polaridade da célula. Em seguida, ocorre a repolarização, em que a célula tende a voltar para o potencial de repouso. Porém, há um curto in-

intervalo de tempo em que a célula passa por um período refratário, que é quando sua voltagem fica rapidamente menor que seu potencial de repouso - ver figura (2)

1.1.3 Modelo de Hindmarsh-Rose

Há algumas décadas, o comportamento dos impulsos elétricos e dos potenciais vem sendo estudado matematicamente. Assim, diversos modelos tentam descrever e aproximar a atividade neuronal como sendo circuitos elétricos, onde podem-se relacionar íons como sendo baterias, os gradientes de concentração como sendo as forças eletromotrizes dessas baterias, a permeabilidade da membrana como sendo uma resistência, entre outras propriedades.

O primeiro modelo matemático que descreveu os potenciais de ação das membranas das células nervosas foi proposto por (HODGKIN; HUXLEY, 1952). Anos depois, (FITZHUGH, 1961) simplificaram e utilizaram o modelo de Hodgkin-Huxley a fim de criar um modelo com duas variáveis. Em cima desse modelo, (HINDMARSH; ROSE, 1984) propôs uma modificação onde introduziram um ponto de equilíbrio e um ponto de sela, que geram duas separatrizes que dividem o plano de fase em duas regiões. Assim, em uma certa região, as órbitas se aproximam de um ciclo limite e, em outra região, as órbitas se aproximam em um ponto de equilíbrio estável.

1.2 MOTIVAÇÃO

A motivação da nossa proposta de trabalho foi de estudar e avaliar um dos modelos neuronais mais fundamental: o modelo de Hodgkin-Huxley. Como nós dois gostamos e trabalhamos, de certa forma, com neurociência em nossas iniciações científicas, consideramos justo entender melhor um dos modelos matemáticos se envolve nesta área.

Entretanto, como este modelo é mais difícil de lidar, lidamos com uma simplificação, que foi proposto por FitzHugh-Nagumo. Para generalizar um pouco as equações de FitzHugh-Nagumo, utilizamos o modelo de Hindmarsh-Rose.

1.3 OBJETIVOS

O objetivo do presente projeto é desenvolver um algoritmo e simular o comportamento caótico do modelo de Hindmarsh-Rose pelo método numérico Runge Kutta com base no trabalho de (LIMA; CHAVARETTE; ROÉFERO, 2019) a fim de estudar conceitos relacionados aos sistemas dinâmicos e apresentar à disciplina de Física Computacional do curso de bacharelado em física médica pela Universidade de São Paulo.

2 METODOLOGIA

O modelo de Hindmarsh-Rose a ser explorado nesse projeto segue o sistema de equações não lineares (2.1), onde as variáveis de estado $x(t)$, $y(t)$ e $z(t)$ são adimensionais e representam o comportamento do potencial de membrana, fluxos iônicos rápidos de sódio e potássio e os fluxos de canais lentos, respectivamente (HINDMARSH; ROSE, 1984).

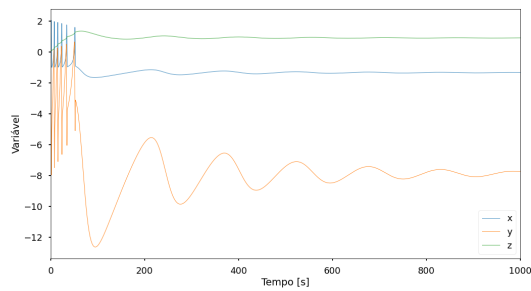
$$\begin{aligned}\dot{x} &= y - ax^3 + bx^2 - z + I \\ \dot{y} &= c - dx^2 - y \\ \dot{z} &= r[s(x - x_r) - z]\end{aligned}\tag{2.1}$$

Os parâmetros $a = 1$, $b = 3$, $c = 1$, $d = 5$, $r = 0,006$, $s = 4$ e $x_r = -1,56$ permitem simular o contexto experimental de forma muito precisa. O parâmetro I representa um valor de estimulação externa que funciona como parâmetro de controle; com a sua variação, podemos estudar o comportamento de um potencial de ação que tende ao caos quanto maior for seu valor.

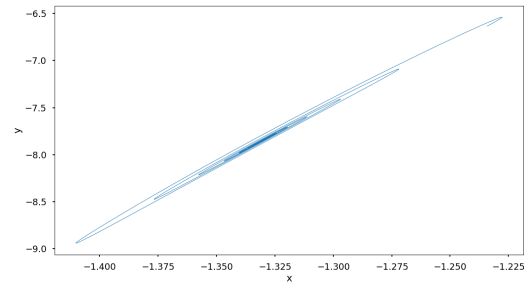
3 RESULTADOS E DISCUSSÃO

3.1 ANÁLISE DA VARIAÇÃO DO PARÂMETRO DE PERTURBAÇÃO EXTERNA I

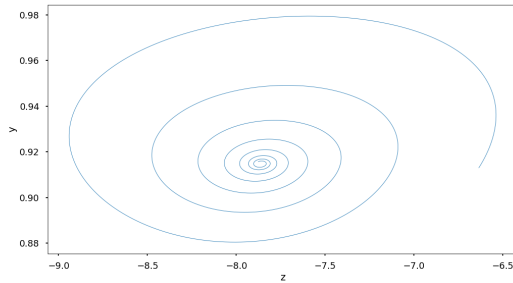
Para simular o sistema dinâmico, utilizamos três diferentes valores para a estimulação externa I do sistema (2.1). Primeiramente, usamos uma intensidade igual a 1,1 e os gráficos obtidos estão dispostos na figura (3). Para este valor de I , o sistema não se demonstra caótico. Podemos ver pelos planos de fase - figuras (3b), (3c) e (3d) - que o sistema tende para um ponto fixo. As partes iniciais das soluções (período transiente) foram eliminadas afim de não poluir a ilustração do comportamento caótico.



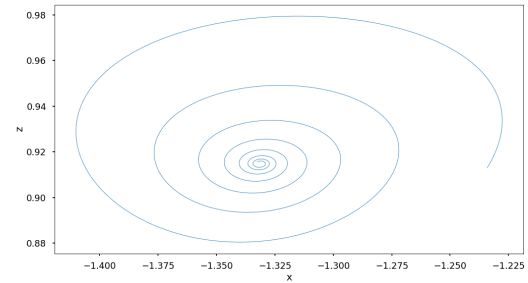
(a) Comportamento dinâmico das variáveis x , y e z para $I = 1.1$.



(b) Plano de fase entre y e x para $I = 1.1$.



(c) Plano de fase entre y e z para $I = 1.1$.



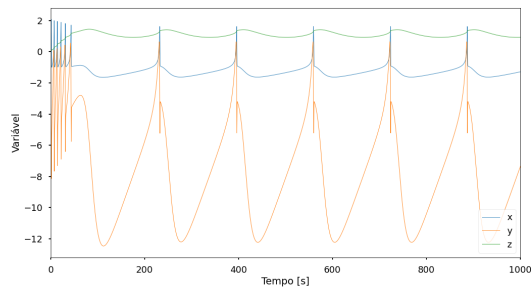
(d) Plano de fase entre z e x para $I = 1.1$.

Figura 3 – Soluções dinâmicas para $I = 1.1$.

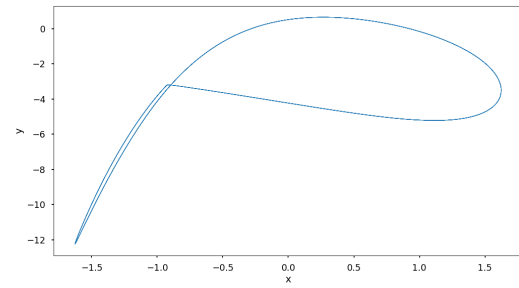
Fonte: Elaborado pelos autores, 2021

Quando aumentamos a estimulação para 1,2, o comportamento do sistema se altera bastante - conforme podemos observar na figura (4). Novamente, foram eliminados os períodos transientes. Com este I , o sistema não apresenta mais o ponto fixo observado anteriormente. Agora - como podemos ver nas figuras (4b), (4c) e (4d) - os planos de fase são ciclos limites.

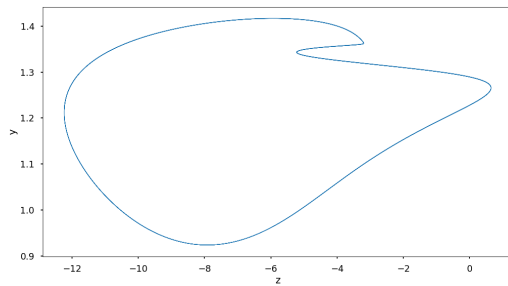
Para alcançar um comportamento caótico, precisa-se aumentar consideravelmente o valor de I ; assim utilizou-se $I = 3$. Plotando as soluções para x , y e z no tempo, pode-se observar a presença de pulsos aleatórios e instáveis, como mostrado na figura (5a). Os atratores caóticos estão ilustrados nas figuras (5b), (5c) e (5d). Assim como nos casos anteriores, os períodos transientes foram eliminados. Formalmente,



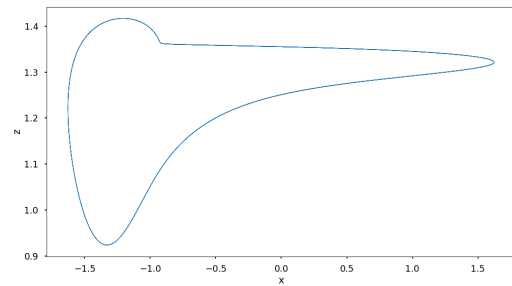
(a) Comportamento dinâmico das variáveis x , y e z para $I = 1.2$.



(b) Plano de fase entre y e x para $I = 1.2$.



(c) Plano de fase entre y e z para $I = 1.2$.

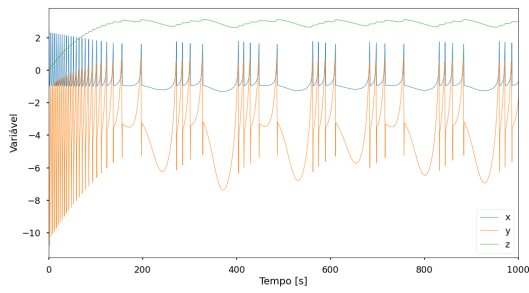


(d) Plano de fase entre z e x para $I = 1.2$.

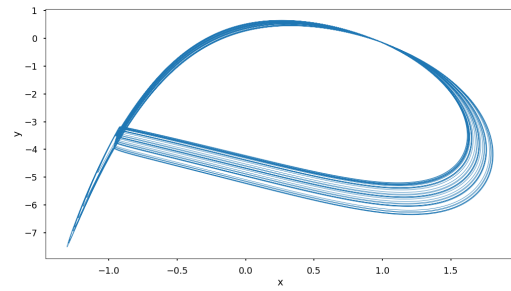
Figura 4 – Soluções dinâmicas para $I = 1.2$.

Fonte: Elaborado pelos autores, 2021

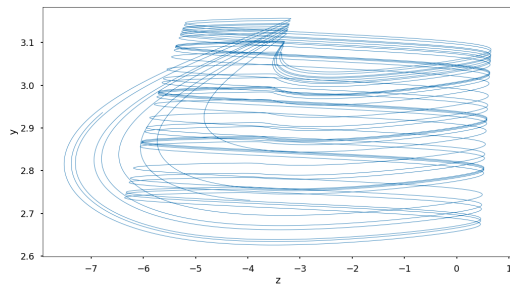
é necessário calcular os expoentes de Lyapunov para afirmar a existência de comportamento caótico, porém essa etapa não está dentro do escopo do projeto atual.



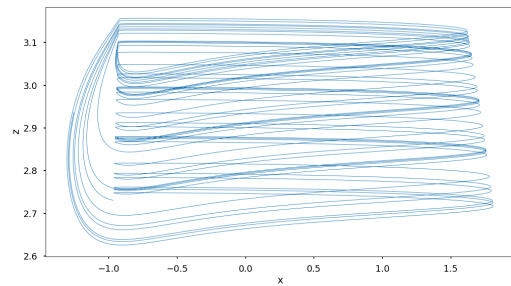
(a) Comportamento dinâmico das variáveis x , y e z para $I = 3.0$.



(b) Plano de fase entre y e x para $I = 3.0$.



(c) Plano de fase entre y e z para $I = 3.0$.



(d) Plano de fase entre z e x para $I = 3.0$.

Figura 5 – Soluções dinâmicas para $I = 3.0$.

Fonte: Elaborado pelos autores, 2021

3.2 ANÁLISE DE ESTABILIDADE

Encontrando os pontos fixos do sistema para cada valor de I , foi possível aplicá-los na matriz jacobiana do sistema e encontrar os respectivos autovalores como mostrado no código presente no apêndice A. Para cada caso, observou-se a existência de três pontos fixos. A Figura (6) resume a análise qualitativa de estabilidade feita sobre estes resultados.

$I = 1,1$	PF 1	<i>Foco estável</i>
	PF 2	<i>Foco instável</i>
	PF 3	<i>Foco instável</i>
$I = 1,2$	Ciclo Limite	
	PF 1	<i>Foco instável</i>
	PF 2	<i>Foco instável</i>
	PF 3	<i>Foco instável</i>
$I = 3$	Atratores Caóticos	
	PF 1	<i>Ponto de sela</i>
	PF 2	<i>Foco instável</i>
	PF 3	<i>Foco instável</i>

Figura 6 – Análise de estabilidade para cada ponto fixo calculado para cada valor de I .

Fonte: Elaborado pelos autores, 2021.

4 CONCLUSÕES

Neste projeto estudamos e simulamos o modelo neuronal dinâmico de Hindmarsh-Rose e pudemos concluir, através de nossas análises, que este sistema dinâmico é caótico de acordo com a excitação inicial I .

Cumprimos os objetivos de simular e observar o comportamento caótico do sistema utilizando diferentes valores de I , de acordo com o artigo base de (LIMA; CHAVARETTE; ROÉFERO, 2019). Assim, também pudemos avaliar a estabilidade dos pontos fixos encontrados computacionalmente para cada caso. Os padrões e gráficos se aproximaram significativamente do esperado. Entretanto, o sistema demonstrou ser caótico para valores de excitação acima de 3.0, diferentemente do que era esperado pelo artigo.

Este trabalho foi importante para nosso aprofundamento nos estudos em simulação de sistemas dinâmicos e física computacional. Nos permitiu conhecer melhor os modelos utilizados na neurociência computacional e seus métodos de estudo. Assim, foram aperfeiçoadas nossas competências para estudar sistemas dinâmicos computacionalmente.

REFERÊNCIAS

FITZHUGH, R. Impulses and physiological states in theoretical models of nerve membrane. *Biophysical journal*, Elsevier, v. 1, n. 6, p. 445–466, 1961. Citado na página 5.

HINDMARSH, J. L.; ROSE, R. M. A Model of Neuronal Bursting Using Three Coupled First Order Differential Equations. In: . [S.l.: s.n.], 1984. v. 221, n. 1222, p. 87–102. Citado 2 vezes nas páginas 5 e 6.

HODGKIN, A. L.; HUXLEY, A. F. The components of membrane conductance in the giant axon of loligo. *The Journal of physiology*, Wiley Online Library, v. 116, n. 4, p. 473–496, 1952. Citado na página 5.

LIMA, R. S. de; CHAVARETTE, F. R.; ROÉFERO, L. G. P. Estudo do comportamento dinâmico do modelo neuronal de hindmarsh-rose. In: *Colloquium Exactarum*. ISSN: 2178-8332. [S.l.: s.n.], 2019. v. 11, n. 4, p. 122–130. Citado 2 vezes nas páginas 5 e 11.

MOREIRA, C. *Potencial de Ação*. Disponível em: <<https://rce.casadasciencias.org/rceapp/art/2015/253/>>. Citado na página 4.

UNIFAL. *Histologia Interativa*. Disponível em: <<https://www.unifal-mg.edu.br/histologiainterativa/tecido-nervoso/>>. Citado na página 3.

APÊNDICE A – CÓDIGO FONTE DA SIMULAÇÃO DO SISTEMA DINÂMICO

```

1 import numpy as np
2
3
4 def lambdas(i):
5     import sympy as sm
6
7     # Parâmetros para o sistema
8     a = 1
9     b = 3
10    c = 1
11    d = 5
12    r = 0.006
13    s = 4
14    xr = - 1.56
15
16    x, y, z = sm.symbols('x, y, z')
17
18    # O sistema dinâmico a ser simulado
19    fx = y - a * x ** 3 + b * x ** 2 - z + i
20    fy = c - d * x ** 2 - y
21    fz = r * (s * (x - xr) - z)
22
23    # Igualar as equações do sistema a zero
24    eqx = sm.Eq(fx, 0)
25    eqy = sm.Eq(fy, 0)
26    eqz = sm.Eq(fz, 0)
27
28    # Resolver o sistema
29    pf1, pf2, pf3 = sm.solve([eqx, eqy, eqz])
30
31    sistema = sm.Matrix([fx, fy, fz]) # Sistema de matrizes
32    jacobiano_sistema = sistema.jacobian((x, y, z)) # Jacobiano das
matrizes    |A*I - lambda| = 0
33
34    # Substituir os pontos de equilibrio no sistema das matrizes jacobianas
35    jacobiano_pf1 = jacobiano_sistema.subs([(x, pf1.get(x)), (y, pf1.get(y)), (z, pf1.get(z))])
36    jacobiano_pf2 = jacobiano_sistema.subs([(x, pf2.get(x)), (y, pf2.get(y)), (z, pf2.get(z))])
37    jacobiano_pf3 = jacobiano_sistema.subs([(x, pf3.get(x)), (y, pf3.get(y)), (z, pf3.get(z))])
38
39    # Printar as soluções para os lambdas de cada função em cada
intensidade I
40    print(f'\n', f'*'*25, f'Lambdas para I = {i}', f'*'*25)

```

```
41     print(f'Lambdas para o ponto fixo 1:')
42     for x in jacobiano_pf1.eigenvals():
43         print(f'{x}')
44     print(f'\nLambdas para o ponto fixo 2:')
45     for y in jacobiano_pf2.eigenvals():
46         print(f'{y}')
47     print(f'\nLambdas para o ponto fixo 3:')
48     for z in jacobiano_pf3.eigenvals():
49         print(f'{z}')
50
51
52 def sist_EDO(edo, t, i):
53     # Função para nos retornar a matriz do sistema a ser resolvida pela
54     # função odeint
55
56     # Parâmetros para o sistema
57     a = 1
58     b = 3
59     c = 1
60     d = 5
61     r = 0.006
62     s = 4
63     xr = - 1.56
64
65     # Retorna a matriz que representa o sistema de equações
66     return np.array([edo[1] - a * edo[0] ** 3 + b * edo[0] ** 2 - edo[2] +
67                      i, # x' = y - a*x^3 + b*x^2 - z + I
68                      c - d * edo[0] ** 2 - edo[1],
69                      # y' = c - d*x^2 - y
70                      r * (s * (edo[0] - xr) - edo[2])],
71                      # z' = r*[s*(x - xr) - z]
72                      )
73
74 def sol_EDO(i):
75     # Função para resolver o sistema dinâmico que é retornado pela função
76     # sist_EDO
77
78     from scipy.integrate import odeint
79     import matplotlib.pyplot as plt
80     from matplotlib.pyplot import figure
81
82     plt.style.use('seaborn-poster') # Estilo do plot
83     tempo = np.arange(0, 1500, 0.01) # Matriz para a iteração de tempo
84     cond_inicial = np.array([0, 0, 0]) # Condições iniciais para o sistema
85
86     # Usamos a função odeint para resolver o sistema de equações retornado
87     # pela função
```

```
82     # sist_ED0 utilizando o tempo e as condições iniciais e diferentes
    intensidades I
83     solucao_edo = odeint(sist_ED0, cond_inicial, tempo, args=(i,))
84     x = solucao_edo[:, 0]
85     y = solucao_edo[:, 1]
86     z = solucao_edo[:, 2]
87
88     # Plot para as funções x, y e z em função do tempo
89     figure(figsize=(15, 8), dpi=80)
90     plt.xlabel('Tempo [s]')
91     plt.ylabel('Variável')
92     plt.xlim(0, 1000)
93     plt.plot(tempo, x, label='x', linewidth=0.75)
94     plt.plot(tempo, y, label='y', linewidth=0.75)
95     plt.plot(tempo, z, label='z', linewidth=0.75)
96     plt.legend(loc=4)
97     plt.show()
98
99     # Plot para o plano de fase de y vs. x
100    figure(figsize=(15, 8), dpi=80)
101    plt.xlabel('x')
102    plt.ylabel('y')
103    plt.plot(x[36000:], y[36000:], linewidth=0.75)
104    plt.show()
105
106    # Plot para o plano de fase de z vs. x
107    figure(figsize=(15, 8), dpi=80)
108    plt.xlabel('x')
109    plt.ylabel('z')
110    plt.plot(x[36000:], z[36000:], linewidth=0.75)
111    plt.show()
112
113    # Plot para o plano de fase de z vs. y
114    figure(figsize=(15, 8), dpi=80)
115    plt.xlabel('z')
116    plt.ylabel('y')
117    plt.plot(y[36000:], z[36000:], linewidth=0.75)
118    plt.show()
119
120    # Plot para o plano de fase 3D
121    figure(figsize=(15, 8), dpi=80)
122    fig = plt.figure()
123    ax = fig.gca(projection='3d')
124    ax.plot(x[36000:], y[36000:], z[36000:])
125    ax.set_xlabel("Eixo x")
126    ax.set_ylabel("Eixo y")
127    ax.set_zlabel("Eixo z")
```



```
128     ax.set_title("Atrator de Lorenz")
129     plt.show()
130
131
132 if __name__ == '__main__':
133     # Encontrar os autovalores lambdas para diferentes intensidades I
134     lambdas(1.1)
135     lambdas(1.2)
136     lambdas(3.0)
137
138     # Simular o sistema dinâmico para diferentes intensidades I
139     sol_ED0(1.1)
140     sol_ED0(1.2)
141     sol_ED0(3.0)
```