Universidade Federal Rural de Pernambuco

Departamento de Física

Computação Aplicada ao Ensino da Física

2 V.A. Parte – B

Bruno da Silva Ramos

Respostas:

1. **A)**

#Universidade Federal Rural de Pernambuco

#Bruno da Silva Ramos

#Questão 1 - Parte B (2. VA - Computação Aplicada ao Ensino da Física)

#####################################################################

#Importando as Bibliotecas!

import matplotlib.pyplot as plt

#Definindo Condições Iniciais

l = 0.01

h = 0.01

c = 0.001

t = 0

y = 1

tempo, eixoy = [], []

#Definindo o Loop

while t <= 10:

tempo.append(t)

eixoy.append(y)

#Incrementando as variaveis

y = y + h\*(-y\*l - 2\*c\*t)

t = t + h

#Criando o Gráfico

plt.plot(tempo, eixoy)

plt.xlabel("Tempo")

plt.ylabel("N(T)/N0")

plt.show()

#Letra B.

for n in range(len(eixoy)):

f = eixoy[n]

f = int(f\*(100))

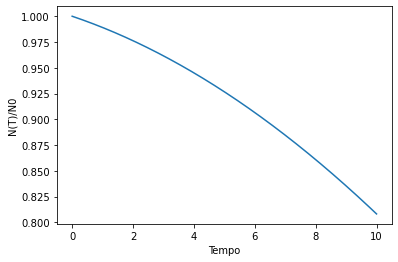
if(f==85):

index = n

break

#Achando o Tempo Correspondente

print("O tempo t' é de: {:.2f} s".format(tempo[index]))



**B)** Como já está no próprio programa, o tempo para ser atingido 85% é de aproximadamente 8,04s. (Se considerar a ultima vez que temos o elemento 0.85 esse tempo é 8,42s. Pode verificar isto tirando o “break” dentro do for no código).

1. **A) B)**

#Universidade Federal Rural de Pernambuco

#Bruno da Silva Ramos

#Questão 2 - Parte B (2. VA - Computação Aplicada ao Ensino da Física)

#####################################################################

#Importando as Bibliotecas!

import matplotlib.pyplot as plt

import numpy as np

#Definindo as Condições Iniciais

h = 0.1

y = -1

t = 0

tempo, eixoy = [], []

#Loop

while t<=10:

tempo.append(t)

eixoy.append(y)

k1 = -3\*(y + 1) + 3\*(2\*np.cos(3\*t) - 3\*t)

k2 = -3\*((y+h\*k1)+1) + 3\*(2\*np.cos(3\*(t+h)) - 3\*(t+h))

y = y + (h/2)\*(k1 + k2)

t = t + h

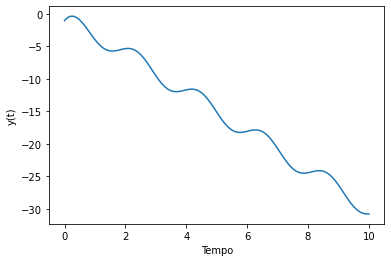
#Plotando o Gráfico

plt.plot(tempo, eixoy)

plt.xlabel("Tempo")

plt.ylabel("y(t)")

plt.show()



**c)** Como o erro global para o método de Euler modificado pode ser obtido a partir do passo utilizado na forma . Nós temos que .

1. **A) B)**

#Universidade Federal Rural de Pernambuco

#Bruno da Silva Ramos

#Questão 3 - Parte B (2. VA - Computação Aplicada ao Ensino da Física)

#####################################################################

#Importando as Bibliotecas!

import matplotlib.pyplot as plt

#Difinindo Condições Iniciais

a = 0.3

g = 9.8

v = 0

t = 0

h = 0.01

tempo, eixoy = [], []

#loop

while t<=30:

tempo.append(t)

eixoy.append(v)

k1 = g - a\*v

k2 = g - a\*(v+h\*k1)

v = v + (h/2)\*(k1+k2)

t = t + h

#Plotando o Gráfico

plt.plot(tempo, eixoy)

plt.xlabel('Tempo (t)')

plt.ylabel('V(t)')

plt.show()

#Procurando a velocidade terminal

vt = eixoy[len(eixoy)-1]

print("A velocidade terminal é de: {:.2f} m/s".format(vt))

vt = int(vt\*100) #Só para deixar mais fácil de procurar o tempo!

#Procurando em qual instante de tempo foi alcançado essa velocidade

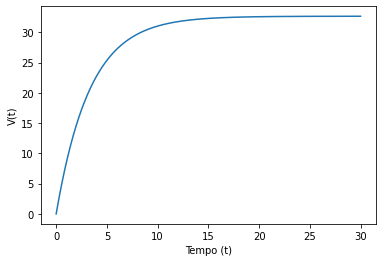
for n in range(len(eixoy)):

if (int(eixoy[n]\*100)==vt):

tempofinal = tempo[n]

break

print("O tempo estimado para se atingir a velocidade terminal é de: {:.2f}s".format(tempofinal))



**C)**

A velocidade terminal é de: 32.66 m/s

O tempo estimado para se atingir a velocidade terminal é de: 28.33s

#Universidade Federal Rural de Pernambuco

#Bruno da Silva Ramos

#Questão 4 - Parte B (2. VA - Computação Aplicada ao Ensino da Física)

#####################################################################

#Importando as Bibliotecas!

import matplotlib.pyplot as plt

#Definindo Condições Iniciais

r = 0.05

c = 0.1

b = 0.1

m = 0.01

h = 0.1

t = 0

n = 0.3

p = 0.7

tempo, ny, py = [], [], []

#Loop

while t<=2000:

tempo.append(t)

ny.append(n)

py.append(p)

#Para o N(t)

kn1 = r\*n - c\*n\*p

kn2 = r\*(n+(h\*kn1)/2) - c\*(n+(h\*kn1)/2)\*p

kn3 = r\*(n+(h\*kn2)/2) - c\*(n+(h\*kn2)/2)\*p

kn4 = r\*(n+h\*kn3) - c\*(n+h\*kn3)\*p

#Para o P(t)

kp1 = b\*n\*p - m\*p

kp2 = b\*n\*(p+(h+kp1)/2) - m\*(p+(h\*kp1)/2)

kp3 = b\*n\*(p+(h+kp2)/2) - m\*(p+(h\*kp2)/2)

kp4 = b\*n\*(p+h\*kp3) - m\*(p+h\*kp3)

#Incrementando as Variaveis

n = n + (h/6)\*(kn1+2\*kn2+2\*kn3+kn4)

p = p + (h/6)\*(kp1+2\*kp2+2\*kp3+kp4)

t = t + h

#Plotando os Gráficos

plt.plot(tempo, ny, "g--", tempo, py, "b--")

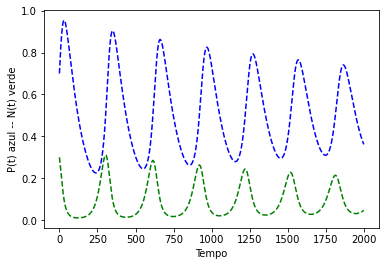
plt.show()

print('São possiveis ver 7 gerações de presas e predatores!')

plt.plot (ny, py)

plt.show()

B) São possíveis de ver 7 gerações de presa e predador.



C) O diagrama nos diz que essas 2 espécies podem viver em harmonia.

