**Gabarito Capítulo 9 – Utilizando-se os Programas em Python fornecidos**

**Exercício 1**

Euler Modificado

y0 = 0.5

y1 = 0.8260000000000001

y2 = 1.2069200000000002

**Opção Correta:**

**A)** 1,2069200

**Exercício 2**

Rk4

y0 = 0.5

y1 = 0.8292933333333334

**Opção Correta:**

**C)** 0,8292933

**Exercício 3**

1. O valor obtido para , em , pelo Método de Euler é 24; **(V)**
2. O erro cometido com o Método de Euler para a obtenção de , em , é de aproxima­damente 14%; **(F) – Erro nulo, pois a função é uma constante.**
3. O valor obtido para , em , pelo Método de Runge-Kutta de 4ª ordem é 21. **(F) – Em x = 7 o método Rk4 obtém o valor 24 como resultado.**

Euler

y0 = 6

y1 = 12

y2 = 18

y3 = 24

Euler Modificado

y0 = 6

y1 = 12

y2 = 18

y3 = 24

Rk4

y0 = 6

y1 = 12

y2 = 18

y3 = 24

**Opção Correta:**

**C)** apenas (I) é ***V***

**Exercício 4**

**Aplicando-se as condições iniciais para a obtenção da constante c:**

**Solução Analítica:**

Substituindo :

(I) VERDADEIRA

Com o Método de Euler:

(II) VERDADEIRA

🡺 erro percent. = |0,3459 – 0,2718|/0,3459 = **21,4%**

(III) VERDADEIRA

**Solução correta:**

**A)** todas são V

**Exercício 5**

1. Adotando o Método de Euler, em obtém-se 1,120422; **(F)**
2. Com o Método de Euler Modificado, em encontra-se 0,560211; **(F)**
3. Pelo Método de Runge-Kutta de 4ª ordem, em obtém-se 0,296997. **(F)**

**Saída do Programa:**

**Euler**

**y0 = 0**

**y1 = 0.0**

**y2 = 0.41218031767503205**

**Euler Modificado**

**y0 = 0**

**y1 = 0.20609015883751602**

**y2 = 0.7826155365335193**

**Rk4**

**y0 = 0**

**y1 = 0.1355730433983252**

**y2 = 0.6253188094781492**

**Conclusão:**

**A**) todas são ***F***

**Exercício 6**

**Euler Explícito**

**y0 = 1**

**y1 = 1.0**

**y2 = 1.2304**

**Conclusão:**

**A**) 1,2304

**Exercício 7**

**Euler Modificado**

**y0 = 1**

**y1 = 1.1152**

**y2 = 1.23208192**

**Conclusão:**

**C**) 1,1152

**Exercício 8**

**Euler Modificado**

**y0 = 1**

**y1 = 0.75**

**y2 = 0.78125**

**Conclusão:**

**B**) 0,7812

**Exercício 9**

**Rk4**

**y0 = 1**

**y1 = 0.7135416666666667**

**Conclusão:**

**C**)0,7135

**Exercício 10**

**Euler**

**y0 = -1**

**y1 = -1.4375**

**y2 = -1.65625**

**y3 = -0.796875**

**y4 = 2.8046875**

1. Em , pelo Método de Euler é - 0,7969; **(V)**

**Euler Modificado**

**y0 = -1**

**y1 = -1.328125**

**y2 = -0.939453125**

**y3 = 1.739013671875**

**y4 = 9.732147216796875**

1. Com o Método de Euler Modificado em é - 1,3984; **(F)**

**Rk4**

**y0 = -1**

**y1 = -1.36767578125**

**y2 = -1.0011100769042969**

**y3 = 1.7711193263530731**

**y4 = 10.190575608285144**

1. Em , pelo Método de Runge-Kutta de 4ª ordem é - 1,3677. **(V)**

**Conclusão:**

**A)** apenas (II) é *F*

**Exercícios Computacionais**

**Exercício 11**

**Executando-se adequadamente o programa fornecido conclui-se que a resposta é:**

**E)** NRA

**Exercício 12**

**Executando-se, adequadamente, os programas em Python fornecidos obtém-se as seguintes respostas:**

**Euler**

**y0 = 10**

**y1 = 27.82608695652174**

**y2 = 66.64584531930632**

**y3 = 103.37952077694278**

**Euler Modificado**

**y0 = 10**

**y1 = 38.32292265965316**

**y2 = 68.76832479706805**

**y3 = 73.1975374935126**

**Runge Kutta**

**y0 = 10**

**y1 = 42.79891104868474**

**y2 = 79.0927950105316**

**y3 = 88.11659739027225**

**Conclusão:**

**C)** adotando o método RK4, chega-se aproximadamente a 88 alunos, decorridas 3 horas.

**Exercício 13**

**Utilizando-se o algoritmo do exercício anterior para o método de RK4, com passo de 1/6 h, obtemos o seguinte resultado:**

**y0 = 10| 0 horas**

**y1 = 13.380571808904449 | 0.16666666666666666horas**

**y2 = 17.65791245472849 | 0.3333333333333333horas**

**y3 = 22.904284551820922 | 0.5horas**

**y4 = 29.099330288880726 | 0.6666666666666666horas**

**y5 = 36.09471517682727 | 0.8333333333333333horas**

**y6 = 43.60598327718183 | 0.9999999999999999horas**

**y7 = 51.247493444555076 | 1.1666666666666665horas**

**y8 = 58.60640459102762 | 1.3333333333333333horas**

**y9 = 65.3280347977829 | 1.5horas**

**y10 = 71.17734669346834 | 1.6666666666666667horas**

**y11 = 76.05687026777397 | 1.8333333333333335horas**

**y12 = 79.9858899571449 | 2.0horas**

**y13 = 83.0604097250448 | 2.1666666666666665horas**

**y14 = 85.41289847080175 | 2.333333333333333horas**

**y15 = 87.18221005683455 | 2.4999999999999996horas**

**y16 = 88.4957668968011 | 2.666666666666666horas**

**y17 = 89.46160733171614 | 2.8333333333333326horas**

**y18 = 90.16675289320473 | 2.999999999999999horas**

**Conclui-se então que a resposta correta é:**

B) 80 alunos, após 2h

**O programa, utilizando-se a função ode segue abaixo:**



*//Programa Ex. 13 Cap. 9*

clc *// limpa o console*

clear all *// limpa a memória*

funcprot(0);

*//definição da função externa (lado direito da EDO)*

function **ydot**=f(**t**, **y**)

**ydot**= (2\***y**)\*(92-**y**)/92

endfunction

t0=0; *//tempo inicial*

y0=10; *//valor inicial da função y(t)*

t=0 *//variável independente no intervalo [0;30]*

h=1/6 *//passo*

while t<=30

y=ode(y0,t0,t,f);

disp(t)

disp(y)

disp("----------")

t=t+h

end

printf("FIM ======\n")

**Exercício 14**



*//Programa Ex. 14 Cap. 9*

clc *// limpa o console*

clear all *// limpa a memória*

funcprot(0);

*//definição da função externa (lado direito da EDO)*

function **ydot**=f(**t**, **y**)

**ydot**= ((2e-6)\*(100000-**y**))\***y**

endfunction

t0=0; *//tempo inicial*

y0=1000; *//valor inicial de infectados*

t=0 *//variável independente no intervalo [0;30]*

while t<=30 *//numero de dias*

y=ode(y0,t0,t,f);

disp(t)

disp(y)

disp("----------")

t=t+1

end

printf("FIM ======\n")

**Executando-se o programa acima obtemos:**

**Dia 4**

**Aprox. 2199 indivíduos infectados**

**Dia 30.**

**Aprox. 80296 indivíduos infectados**

**Opção correta:**

**C)** no 4º dia, supera o dobro do número inicial de contaminados

**Exercício 15**

**O algoritmo em Python, abaixo, aplica o método RK4 para o problema em questão. Para tal utiliza-se um passo de 2 segundos.**

import numpy as np

import matplotlib.pyplot as plt

**def** func(t,q,R,C):

    dq\_dt = -(1/R\*C)\*q

    return dq\_dt

**def** rk4(y0,x0,h,R,C):

    x= []

    y= []

    corrente = (y0/C)/R

    x.append(x0)

    y.append(corrente)

    print(**f**"I = {corrente} | t = {x0}")

    while corrente >= 10:

        k1= func(x0,y0,R,C)

        k2= func(x0 + (1/2)\*h, y0 + (1/2)\*k1\*h,R,C)

        k3= func(x0 + (1/2)\*h, y0 + (1/2)\*k2\*h,R,C)

        k4= func(x0 + h, y0 + k3\*h,R,C)

        kmedia = (k1 + 2\*k2 + 2\*k3 + k4)/6

        yk = y0 + kmedia\*h

        x0 = x0 + h

        y0 = yk

        print(**f**"I = {(y0/C)/R} | t = {x0}")

        x.append(x0)

        y.append((y0/C)/R)

        corrente = (y0/C)/R

    \_,ax = plt.subplots()

    ax.set(xlabel ='tempo(s)', ylabel ='Corrente(A)')

    plt.plot(x,y)

    plt.plot(x,[10]\*len(x),'--')

    plt.legend()

    plt.show()

    return

if \_\_name\_\_ == '\_\_main\_\_':

*# Dados do circuitos*

    R= 8

    C= 0.01

    q0= 1

*#Pvi*

    y0= q0 *#q0*

    x0= 0 *#t0*

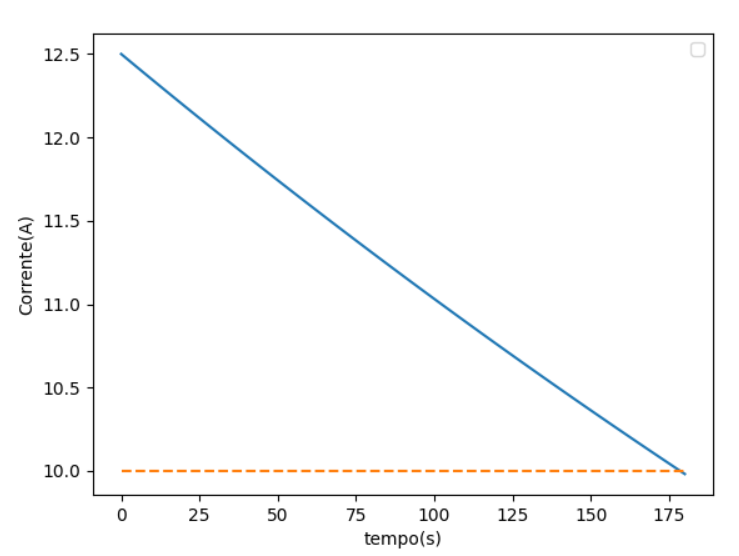
*# Passo*

    h = 2 *#segundos*

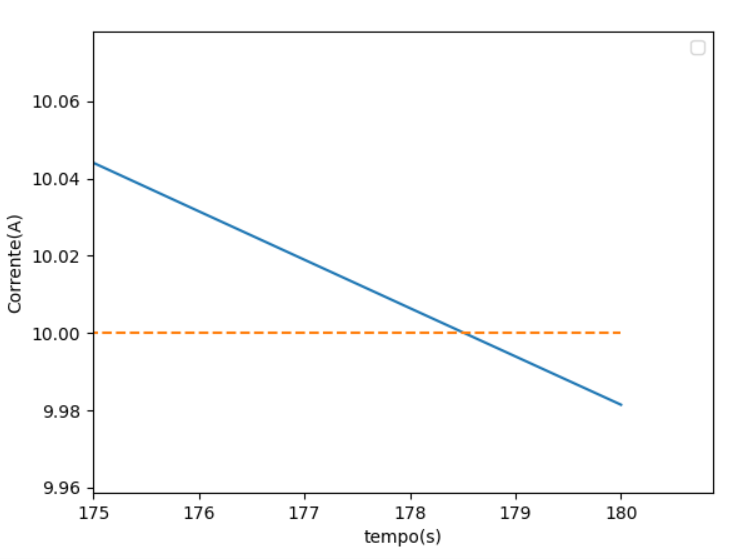
    rk4(y0,x0,h,R,C)



**Os gráficos obtidos com o programa acima facilitam a obtenção da solução:**



Aplicando um “zoom” podemos concluir que a resposta é aproximadamente 178.5 segundos.



**Resposta:**

**O desligamento da lâmpada ocorre após ela permanecer aproximadamente 178.5 segundos ligada.**