**Atividade 10 – Integração Numérica e Equações Diferenciais**

1 – Calcular sendo que o intervalo entre 2 e 3 deve conter 4 subintervalos. Calcular pela regra do trapézio(trapz) e pela fórmula de Simpson. Calcular o valor exato da integral por meio da função si.quad.

Comandos:

import numpy as np

import matplotlib.pyplot as plt

def trapezio(f,a,b,N):

  x = np.linspace(a,b,N+1) # N+1 pontos

  y = f(x)

  y\_inter = y[1:N] # todos valores menos o primeiro e o ultimo

  dx = (b - a)/N

  T = (dx)\*np.sum(y\_inter)+(dx/2)\*(y[0]+y[N])

  return T

I=trapezio(lambda x: x\*2.71\*\*(x/2),2,3,4)

print(I)

import scipy.integrate as si

f= lambda x:x\*2.71\*\*(x/2)

i = si.quad(f, 2, 3)

print (i)

import numpy as np

def f(x):

  return x\*2.71\*\*(x/2)

def simpson(a, b,n):

  h= (b-a)/n

  inter = f(a)+f(b)

  for i in range(1,n):

    k= a+i\*h

    if i%2==0:

      inter =inter +2\*f(k)

    else:

      inter =inter+4\*f(k)

  inter = inter\*h/3

  return inter

print("Resultado por Simpson: %0.3f" % simpson(2,3,4))

Resultados:

8.958033487919646

(8.928249521755072, 9.912348188651023e-14)

Resultado por Simpson: 8.928

1. A determinação da área da seção reta de rios e lagos é importante em projetos de prevenção de enchentes (para o cálculo de vazão da água) e nos projetos de reservatórios (para o cálculo do volume total de água). A menos que dispositivos tipo sonar sejam usados na obtenção do perfil do fundo de rios/lagos, o engenheiro civil deve trabalhar com valores de profundidade, obtidos em pontos discretos da superfície. Um exemplo típico de seção reta de um rio está mostrado, aproximadamente, na figura a seguir. Calcule a área da seção reta dada na figura, usando a regra do trapézio. Utilizar x como a distância da margem esquerda e y com a profundidade. Obter os valores diretos do gráfico. Obs: x e y deverão tem a mesma quantidade de pontos.

Diagrama

Descrição gerada automaticamente

Comandos:

x = [0,2,4,6,8,10,12,14,16,18,20]

y = [0,1.8,2,4,4,6,4,3.6,3.4,2.8,0]

I = np.trapz(y,x)

print(I)

Resultados:

63.2

1. Aplique a formulação de Euler e estime o valor de y(2,2) da EDO y’ = 3 – y/x  
   com condição inicial y(2) = 2 e passo h = 0,1. Comparar o resultado usando odeint.

Comandos:

import numpy as np

import matplotlib.pyplot as plt

def odeEuler(f,y0,t):

    y = np.zeros(len(t))

    y[0] = y0

    for n in range(0,len(t)-1):

        y[n+1] = y[n] + f(y[n],t[n])\*(t[n+1] - t[n])

    return y

t = np.linspace(2,2.2,3)

y0 = 2

f = lambda y,t: 3-(y/t)

y = odeEuler(f,y0,t)

print(y)

plt.plot(t,y,'b.-')

plt.legend(['Euler'])

plt.axis([2,2.2,2,2.4])

plt.grid(True)

plt.show()

Resultados:

[2. 2.2 2.3952381]

Gráficos:

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

**Método Odeint**

**Comandos:**

**[**import numpy as np

from scipy.integrate import odeint

import matplotlib.pyplot as plt

# function that returns dy/dt

def model(y,t):

    dydt = 3 - (y/t)

    return dydt

# initial condition

y0 = 2

# time points

t = np.linspace(2,2.2,3)

# solve ODE

y = odeint(model,y0,t)

print(y)

# plot results

plt.plot(t,y)

plt.xlabel('x')

plt.ylabel('y(x)')

plt.grid(True)

plt.show()

**]**

**Resultados:**

**[Gráfico, Gráfico de linhas

Descrição gerada automaticamente]**