NCERT 9.4.3

EE24BTECH11032 - JOHN BOBBY

Question

Find the solution of the differential equation $\frac{dy}{dx} = 1 - y$, using the trapezoidal method.Assume $y\left(0\right) = 0$

Trapezoidal Rule

$$\int_a^b f(x)\,dx \approx (b-a)\left(\frac{f(a)+f(b)}{2}\right) \tag{1}$$

$$f(y) = 1 - y = \frac{dy}{dx} \tag{2}$$

Integrating the equation (2) from n to $n+1$

$$A = A_{n+1} - A_n = (x_{n+1} - x_n)\left(\frac{f(y_{n+1}) + f(y_n)}{2}\right) \tag{3}$$

$$A_{n+1} - A_n = (x_{n+1} - x_n)(1 - y_{n+1} + 1 - y_n) \tag{4}$$

$$x_{n+1} - x_n = h \tag{5}$$

$$A_{n+1} - A_n = h(2 - y_{n+1} - y_n) \tag{6}$$

On rearranging we get the difference equation,

$$y_{n+1} = y_n + \frac{2-h}{2+h}y_n + \frac{2h}{2+h} \tag{7}$$

Laplace Transform

$$\mathcal{L}\left(\frac{dy}{dx}\right) = \mathcal{L}\left(1 - y\right) \tag{8}$$

$$\mathcal{L}\left(\frac{dy}{dx}\right) = sY(s) - y(0), \quad \mathcal{L}\left(1\right) = \frac{1}{s}, \quad \mathcal{L}\left(y(t)\right) = Y(s)$$

$$sY(s) - y(0) = \frac{1}{s} - Y(s) \tag{9}$$

$$sY(s) + Y(s) = \frac{1}{s} + y(0) \tag{10}$$

$$Y(s)(s+1) = \frac{1}{s} + y(0) \tag{11}$$

$$Y(s) = \frac{1}{s(s+1)} \tag{12}$$

Laplace Transform

Taking inverse Laplace Transform,

$$\mathcal{L}^{-1}\left(Y\left(s\right)\right) = y\left(x\right) \tag{13}$$

$$\mathcal{L}^{-1}\left(\frac{1}{s\left(s+1\right)}\right) = 1 - e^{-x} \tag{14}$$

$$y\left(x\right) = 1 - e^{-x} \tag{15}$$

Bilinear Transform

Let the laplace transform of $f\left(y\right) = 1 - y$ be $X\left(s\right)$

$$X\left(s\right) = \mathcal{L}\left(f\left(x,y\right)\right) \tag{16}$$

Applying laplace transform on both sides of equation

$$sY\left(s\right) = X\left(s\right) \tag{17}$$

Let $H\left(s\right)$ be defined such that

$$H\left(s\right) = \frac{Y\left(s\right)}{X\left(s\right)} \tag{18}$$

$$H\left(s\right) = 1/s \tag{19}$$

Bilinear Transform

Applying bilinear transform which converts s-domain to z-domain

$$s = \frac{2}{h}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right) \tag{20}$$

$$H(z) = \frac{h}{2}\left(\frac{1 + z^{-1}}{1 - z^{-1}}\right) \tag{21}$$

$$Y(z) = \frac{h}{2}\left(\frac{1 + z^{-1}}{1 - z^{-1}}\right) X(z) \tag{22}$$

On rearranging,

$$zY(z) - Y(z) = \frac{h}{2}\left(zX(z) + X(z)\right) \tag{23}$$

Bilinear Transform

Applying Z inverse transform,

$$y_{n+1} - y_n = \frac{h}{2}\left(f\left(x_{n+1}, y_{n+1}\right) + f\left(x_n, y_n\right)\right) \tag{24}$$

$$y_{n+1} - y_n = \frac{h}{2}\left(1 - y_{n+1} + 1 - y_n\right) \tag{25}$$

$$y_{n+1} - y_n = \frac{h}{2}\left(2 - y_{n+1} - y_n\right) \tag{26}$$

$$y_{n+1} = y_n + \frac{h}{2}\left(2 - y_{n+1} - y_n\right) \tag{27}$$

Equation (34) is the same difference equation obtained in equation (6)

C-Code

```c
#include <math.h>
void trapezoidal(double *x, double *y,double *y_trapezoid, int n, double h) {
    y_trapezoid[0] = 0;
    for (int i = 0; i < n - 1; i++) {
        y_trapezoid[i+1]=y_trapezoid[i]+(h/2)*(2-y[i]-y[i+1]);
    }
}
void function(double *x,double *y,int n){
        y[0]=0;
        for(int i=0;i<n;i++){
                y[i]=1-exp(-1*x[i]);
        }

}
```

Python-Code

```
import numpy as np
import matplotlib.pyplot as plt
import ctypes
# Load the shared library
trapezoidal = ctypes.CDLL('./trapezoidal.so')
# Set argument and return types for the C functions
trapezoidal.trapezoidal.argtypes = [ctypes.POINTER(ctypes.c_double),
↪  ctypes.POINTER(ctypes.c_double),ctypes.POINTER(ctypes.c_double),
↪  ctypes.c_int, ctypes.c_double]
trapezoidal.function.argtypes = [ctypes.POINTER(ctypes.c_double),
↪  ctypes.POINTER(ctypes.c_double), ctypes.c_int]
# Parameters
x_start = 0
x_end = 5
h = 0.1
n_steps = 51
#Intialising the x and y arrays
x = np.linspace(x_start, x_end, n_steps)
y = np.zeros(n_steps)
y_trapezoidal=np.zeros(n_steps)
```

Python-Code

```
#Converting array to ctypes
x_ctypes = x.ctypes.data_as(ctypes.POINTER(ctypes.c_double))
y_ctypes = y.ctypes.data_as(ctypes.POINTER(ctypes.c_double))
y_trapezoidal_ctypes =
↪  y_trapezoidal.ctypes.data_as(ctypes.POINTER(ctypes.c_double))
# Call the C functions
trapezoidal.function(x_ctypes, y_ctypes, n_steps)
trapezoidal.trapezoidal(x_ctypes, y_ctypes,y_trapezoidal_ctypes, n_steps,h)
# Plotting
plt.figure(figsize=(10, 6))
plt.plot(x, y, label="Theory", linestyle='-', color='b',linewidth=10)
plt.plot(x, y_trapezoidal, label="Trapezoidal", linestyle='--',
↪  color='r',linewidth=7)
plt.xlabel("x")
plt.ylabel("y")
#plt.legend()
plt.legend(['Theory','Trapezoidal'])
plt.grid()
#plt.show()
plt.savefig('plot.png')
```

Plot