9.1.5

EE24BTECH11004 - Ankit Jainar

Question

Solve the differential equation:

$$\frac{d^2y}{dx^2} = \cos(3x) + \sin(3x)$$

Solution: Theoretical Approach

Given:

$$\frac{d^2y}{dx^2} = \cos(3x) + \sin(3x) \tag{1}$$

Integrating both sides:

$$\frac{dy}{dx} = \int \cos(3x)dx + \int \sin(3x)dx + c_1 \tag{2}$$

$$= \frac{\sin(3x)}{3} - \frac{\cos(3x)}{3} + c_1 \tag{3}$$

Integrating again:

$$y = \int \left(\frac{\sin(3x)}{3} - \frac{\cos(3x)}{3} + c_1 \right) dx + c_2 \tag{4}$$

$$= -\frac{\cos(3x)}{9} - \frac{\sin(3x)}{9} + c_1 x + c_2 \tag{5}$$

Solution: Theoretical Approach (contd.)

Assuming the initial conditions:

- $y(0) = 0$
- $y'(0) = 0$

Substituting $y(0) = 0$:

$$-\frac{\cos(0)}{9} - \frac{\sin(0)}{9} + c_1 \cdot 0 + c_2 = 0 \tag{6}$$

$$c_2 = \frac{1}{9} \tag{7}$$

Substituting $y'(0) = 0$:

$$\frac{\sin(0)}{3} - \frac{\cos(0)}{3} + c_1 = 0 \tag{8}$$

$$c_1 = \frac{1}{3} \tag{9}$$

Thus, the solution is:

$$y(x) = -\frac{\cos(3x)}{9} - \frac{\sin(3x)}{9} + \frac{x}{3} + \frac{1}{9} \tag{10}$$

Solution: Computational Approach

The given differential equation is:

$$\frac{d^2y}{dx^2} = \cos(3x) + \sin(3x) \tag{11}$$

Using discretization, the second-order derivative can be approximated as:

$$y''(t+h) = y''(t) + h \cdot (\cos(3t) + \sin(3t)) \tag{12}$$

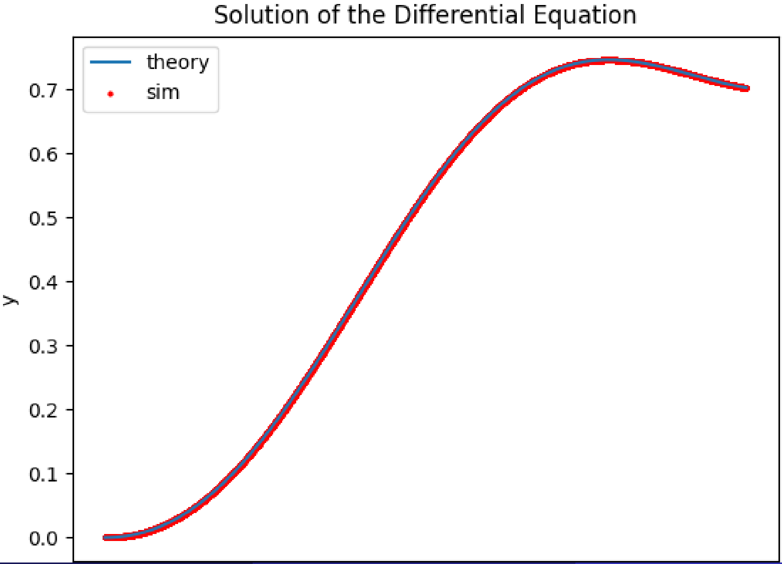
Solution: Computational Approach (contd.)

Expressing this as a system of equations:

$$\vec{y}_{k+1} = \begin{bmatrix} 1 & h & \frac{h^2}{2} \\ 0 & 1 & h \\ 0 & 0 & 1 \end{bmatrix} \cdot \vec{y}_k + \begin{bmatrix} 0 \\ 0 \\ h^2 \left(\cos(3x_k) + \sin(3x_k)\right) \end{bmatrix} \tag{13}$$

By iterating over time steps, the computational solution is computed. Comparison between the theoretical and computational solutions is shown below.

Comparison of Results



C-Code

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

double h = 0.1; //step size which I am considering

double f(double x, double y, double v) {
    return cos(3*x)+sin(3*x);          /* d2y/dx2 = f(x,y,dy/dx) */
}

void solve(double *x,double *y,double *v,int steps){
    for(int i=0;i<steps;i++){
        *y += h * (*v); //y value using dy/dx

        *v += h * f(*x,*y,*v); //value v updating using d2y/dx2

        *x += h; //x value
    }
}

```

Code for plot

```python
import ctypes
import numpy as np
import matplotlib.pyplot as plt

# Load the shared object file (assuming it's in the same directory)
code = ctypes.CDLL('./code.so')

# Define argument and return types for the solve function
code.solve.argtypes = [ctypes.POINTER(ctypes.c_double),
                       ctypes.POINTER(ctypes.c_double),
                       ctypes.POINTER(ctypes.c_double),
                       ctypes.c_int]
# Parameters
h = 0.1  # step size
steps = 10000  # number of steps

# Initial values
x = 0.0
y = 0.0
v = 0.0

# Create arrays to hold the results
x_vals = np.zeros(steps)
y_vals = np.zeros(steps)
v_vals = np.zeros(steps)

# Prepare ctypes variables for passing to the solve function
x_ctypes = ctypes.c_double(x)
y_ctypes = ctypes.c_double(y)
v_ctypes = ctypes.c_double(v)

# Collect initial values
x_vals[0] = x
y_vals[0] = y
v_vals[0] = v

# Call the solve function and collect values at each step
for i in range(1, steps):
    # Call the solve function for each step
    code.solve(ctypes.byref(x_ctypes), ctypes.byref(y_ctypes), ctypes.byref(v_ctypes), 1)

    # Store the results after the update
    x_vals[i] = x_ctypes.value
    y_vals[i] = y_ctypes.value
    v_vals[i] = v_ctypes.value
# Plot the results
plt.plot(x_vals, y_vals, label='theory')
plt.xlabel('x')
plt.ylabel('y')
plt.scatter(x_vals, y_vals, color='red', s=4, label='sim')
plt.legend()
plt.show()
```