NCERT 9.4.3

EE24BTECH11032 - JOHN BOBBY

Question

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

Theoretical Approach

$$\frac{dy}{dx} = 1 - y$$

On rearranging the terms,

$$\begin{aligned}\frac{dy}{1-y} &= dx \\ \int \frac{dy}{1-y} &= \int dx \\ -\log|y-1| + c_1 &= x + c_2\end{aligned}$$

On simplification

$$\begin{aligned}-\log|y-1| &= x + c \\ |y-1| &= \pm e^{-x} \\ y &= 1 \pm e^{-x}\end{aligned}$$

$$y = 1 - e^{-x} \quad (y(0) = 0)$$

Laplace Transform

$$\mathcal{L}\left(f\left(x\right)\right) = F\left(s\right) = \int_0^\infty e^{-sx} f\left(x\right) dx$$

On applying laplace transform on both sides,

$$\begin{aligned}
\mathcal{L}\left(\frac{dy}{dx}\right) &= \mathcal{L}\left(1\right) - \mathcal{L}\left(y\right) \\
\left\{s.Y\left(s\right) - y\left(0\right)\right\} &= \left\{\frac{1}{s}\right\} - \left\{Y\left(s\right)\right\} \\
s.Y\left(s\right) &= \frac{1}{s} - Y\left(s\right) \quad \left(y\left(0\right) = 0\right) \\
Y\left(s\right) &= \frac{1}{s\left(s+1\right)}
\end{aligned}$$

Laplace Transform

Using partial fractions,

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

Applying inverse laplace transform,

$$y(x) = 1 - e^{-x} \quad \left(\mathcal{L}(1) = \frac{1}{s}\right) \quad \left(\mathcal{L}(e^{ax}) = \frac{1}{s-a}\right)$$

Z Transform

$$Y(z) = \mathcal{Z}(y_n) = \sum_{n=0}^{\infty} y_n.z^{-n}$$
$$\frac{dy}{dx} = \frac{y_{n+1} - y_n}{h}$$
$$y_{n+1} = y_n + \frac{dy}{dx}h$$
$$y_{n+1} = y_n + (1 - y_n)h$$

Taking Z transform on both sides of the difference equation,

$$\mathcal{Z}(y_{n+1}) = \mathcal{Z}(y_n) + \mathcal{Z}(h(1 - y_n))$$
$$\{z.Y(z) - z.y(0)\} = \{Y(z)\} + \left\{h\left(\frac{1}{1 - z^{-1}} - Y(z)\right)\right\}$$

Z Transform

$$z.Y(z) = Y(z) + h\left(\frac{1}{1-z^{-1}} - Y(z)\right) \quad (y(0)=0)$$

On rearranging,

$$Y(z) = \frac{hz}{(z-1)(z+h-1)} = \frac{A}{z-1} + \frac{B}{z+h-1}$$

Using partial fractions, $A = 1, B = h-1$

$$Y(z) = \frac{1}{z-1} + \frac{h-1}{z+h-1}$$

Z Transform

On applying inverse z transform,

$$y(x) = 1 - (1-h)^{x+1} \quad \mathcal{Z}(a^x) = \frac{1}{z-a}$$

As $h = 0.1$

$$y(x) = 1 - (0.9)^{x+1}$$

Finite Differences

$$\frac{dy}{dx} = \lim_{h \to 0} \frac{y(x+h) - y(x)}{h} = \lim_{h \to 0} \frac{y_{n+1} - y_n}{h}$$

If h is sufficiently small,

$$\frac{dy}{dx} = \frac{y_{n+1} - y_n}{h}$$

$$y_{n+1} = y_n + \frac{dy}{dx} h$$

$$y_{n+1} = y_n + (1 - y_n) h$$

1. The interval $[0, 5]$ is divided into 51 equal parts each of width 0.1units
2. On starting from a known point (x, y) the value for $y(x + 0.1)$ is calculated. This procedure is repeated until the value of x reaches 5.
3. Knowing all the y values for the equally spaced x values in interval, the solution plot can be plotted.

C-Code

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

void analytical_solution(double *x, double *y, int n) {
    for (int i = 0; i < n; i++) {
        y[i] = 1 - exp(-x[i]);
    }
}
```

Python-Code

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

# Loading the shared library
solver = ctypes.CDLL('./solver.so')

# Setting argument type and return type for functions in the c file
solver.finite_differences.argtypes = [ctypes.POINTER(ctypes.c_double),
↪  ctypes.POINTER(ctypes.c_double), ctypes.c_int, ctypes.c_double]
solver.analytical_solution.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
```

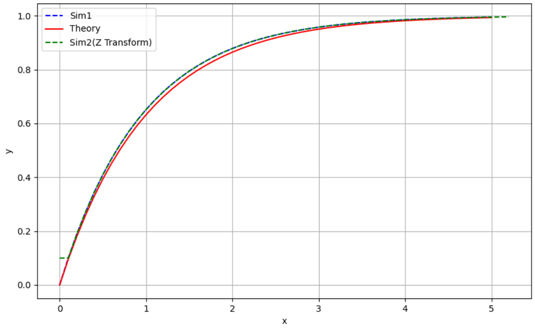
Python-Code

```
# Initialize x and y arrays
x = np.linspace(x_start, x_end, n_steps)
y_fd = np.zeros(n_steps)
y_exact = np.zeros(n_steps)
x_z=[0.0]
y_z=[0.1]
for i in range(52):
    x_z.append(x_z[i]+h)
    y_z.append(1-(0.9)**(i+1))

# Convert arrays to ctypes
x_ctypes = x.ctypes.data_as(ctypes.POINTER(ctypes.c_double))
y_fd_ctypes = y_fd.ctypes.data_as(ctypes.POINTER(ctypes.c_double))
y_exact_ctypes =
↪   y_exact.ctypes.data_as(ctypes.POINTER(ctypes.c_double))

# Calling the functions
solver.finite_differences(x_ctypes, y_fd_ctypes, n_steps, h)
solver.analytical_solution(x_ctypes, y_exact_ctypes, n_steps)
```

Python-Code

```python
# Plotting
plt.figure(figsize=(10, 6))
plt.plot(x, y_fd, label="Simulation", linestyle='--', color='b')
plt.plot(x, y_exact, label="Theory", linestyle='-', color='r')
plt.plot(x_z, y_z, label="Simulation 2", linestyle='--', color='g')

plt.xlabel("x")
plt.ylabel("y")
#plt.legend()
plt.legend(['Sim1', 'Theory','Sim2(Z Transform)'])
plt.grid()
#plt.show()
plt.savefig('plot.png')
```

Plot


Sim1
Theory
Sim2(Z Transform)
y
x