Nguyễn Minh Đức - ITITIU21045 - Lab 10

Ngày .... Tháng ..... Năm........

Q1: a)

t	2	2.1	2.2	2.7	3	3.4
z	6	7.752	10.256	36.576	66	145.168

$y_5(x) = f(x_0) + (x-x_0)f[x_1, x_0] + (x-x_0)(x-x_1)f[x_2,x_1,x_0] + \dots + (x-x_0)(x-x_1)(x-x_2)(x-x_3)(x-x_4)f[x_5,x_4,x_3,x_2,x_1,x_0]$

	t	z	1st	2nd	3rd	4rd	5th
t_0	2	6	17.52	37.6	12	0	32.7
t_1	2.1	7.752	25.04	46	12	45.8	
t_2	2.2	10.256	52.64	56.8	71.5		
t_3	2.7	36.576	98.08	142.6			
t_4	3	66	197.92				
t_5	3.4	145.168					

$\Rightarrow y = 6 + 17.52(x-2) + 37.6(x-2)(x-2.1) + 12(x-2)(x-2.1)(x-2.2) + 32.7(x-2)(x-2.1)(x-2.2)(x-2.7)(x-3)$

$\Rightarrow y(2.5) = 23.1962$

c) $y = y_2 + \dfrac{y_3 - y_2}{x_3 - x_2}(x - x_2)$

$\Rightarrow y = 10.256 + \dfrac{36.576 - 10.256}{2.7 - 2.2}(x - 2.2)$

$\Rightarrow y(2.5) = 10.256 + \dfrac{36.576 - 10.256}{2.7 - 2.2}(2.5 - 2.2) = 26.048$

Q2:

t	0	1	2.5	3	4.5	5	6
z	26	15.5	5.375	3.5	2.375	3.5	5

a) $y_6(x) = f[x_0] + (x-x_0)f[x_1,x_0] + (x-x_0)(x-x_1)f[x_2,x_1,x_0] + \dots + (x-x_0)(x-x_1)(x-x_2)(x-x_3)(x-x_4)(x-x_5)f[x_6,x_5,x_4,x_3,x_2,x_1,x_0]$

	t	z	1st	2nd	3rd	4th	5th	6th
t_0	0	26	-10.5	1.5	0	0	0	-0.011
t_1	1	15.5	-6.75	1.5	0	0	-0.064	
t_2	2.5	5.375	-3.75	1.5	0	-0.223		
t_3	3	3.5	-0.75	1.5	-0.67			
t_4	4.5	2.375	2.25	-0.5				
t_5	5	3.5	1.5					
t_6	6	5						

$y_6(x) = 26 \cancel{-10.5(x} - 10.5x + 1.5x(x-1) - 0.011(x-1)(x-2.5)(x-3)(x-4.5)(x-5)(x-6)x$

$\Rightarrow y_6(3.5) = 2.56$

b) Polynomials to find

$P_1(x) = a_1x^2 + b_1x + c_1$

$P_2(x) = a_2x^2 + b_2x + c_2$

$P_3(x) = a_3x^2 + b_3x + c_3$

We choose four points between range of 3.5 and subsitute:

- $P_1(x_2) = y_2 = 5.375 = a_1(2.5)^2 + 2.5b_1 + c_1$

  $\Rightarrow 5.375 = 6.25a_1 + 2.5b_1 + c_1$

- $P_1(x_3) = y_3 = 3.5 = a_1(3)^2 + 3b_1 + c_1$

  $\Rightarrow 3.5 = 9a_1 + 3b_1 + c_1$

- $P_2(x_3) = y_3 = 3.5 = a_2(3)^2 + 3b_2 + c_2$

  $\Rightarrow 3.5 = 9a_2 + 3b_2 + c_2$

- $P_2(x_4) = y_4 = 2.375 = a_2(4.5)^2 + 4.5b_2 + c_2$

  $\Rightarrow 2.375 = 20.25a_2 + 4.5b_2 + c_2$

- $P_3(x_4) = y_4 = 2.375 = a_3(4.5)^2 + 4.5b_3 + c_3$

  $\Rightarrow 2.375 = 20.25a_3 + 4.5b_3 + c_3$

• $P_3(x_5) = y_5 = 3.5 = a_3(5)^2 + 5b_3 + c_3$

$\Rightarrow 3.5 = 25a_3 + 5b_3 + c_3$

• $\left.\dfrac{dP_1}{dx}\right|_{x=x_3} = \left.\dfrac{dP_2}{dx}\right|_{x=x_3} \Rightarrow 2a_1x + b_1\big|_{x=3} = 2a_2x + b_2\big|_{x=3}$

$\Rightarrow 6a_1 + b_1 = 6a_2 + b_2$

• $\left.\dfrac{dP_2}{dx}\right|_{x=x_4} = \left.\dfrac{dP_3}{dx}\right|_{x=x_4} \Rightarrow 2a_2x + b_2\big|_{x=4.5} = 2a_3x + b_3\big|_{x=4.5}$

$\Rightarrow 9a_2 + b_2 = 9a_3 + b_3$

• Assume that $\left.\dfrac{d^2P_1}{dx^2}\right|_{x=x_1} = 0 \Rightarrow 2a_1 = 0 \Rightarrow \cancel{a_1 = 0}$

$$
\begin{array}{c}
\begin{array}{ccccccccc} a_1 & b_1 & c_1 & a_2 & b_2 & c_2 & a_3 & b_3 & c_3 \end{array} \\
\begin{bmatrix}
6.25 & 2.5 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
9 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 9 & 3 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 20.25 & 4.5 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 20.25 & 4.5 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 25 & 5 & 1 \\
6 & 1 & 0 & -6 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 9 & 1 & 0 & -9 & -1 & 0 \\
2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\end{array}
\begin{bmatrix} a_1 \\ b_1 \\ c_1 \\ a_2 \\ b_2 \\ c_2 \\ a_3 \\ b_3 \\ c_3 \end{bmatrix}
=
\begin{bmatrix} 5.375 \\ 3.5 \\ 3.5 \\ 2.375 \\ 2.375 \\ 3.5 \\ 0 \\ 0 \\ 0 \end{bmatrix}
$$

Therefore:

$$
\left.\begin{aligned}
a_1 &= 0 \\
b_1 &= -15/4 \\
c_1 &= 59/4 \\
a_2 &= 2 \\
b_2 &= -63/4 \\
c_2 &= 131/4 \\
a_3 &= 0 \\
b_3 &= 9/4 \\
c_3 &= -31/4
\end{aligned}\right\}
\Rightarrow
\begin{aligned}
P_1(x) &= \frac{-15}{4}x + \frac{59}{4} \\
P_2(x) &= 2x^2 - \frac{63}{4}x + \frac{131}{4} \\
P_3(x) &= \frac{9}{4}x - \frac{31}{4}
\end{aligned}
$$

$\Rightarrow P_2(3.5) = 2(3.5)^2 - \frac{63}{4}(3.5) + \frac{131}{4} = 2.125$

Q3: Fourth order RK

$\frac{d^2y}{dx^2} + 0.5\frac{dy}{dx} + 7y = 0$; $y(0) = 4$, $y'(0) = 0$

Solve $x = 0$ to 5 ; $h = 0.5$

Let $z = \frac{dy}{dx} \Rightarrow \frac{dz}{dx} = \frac{d^2y}{dx^2}$

Therefore, $\frac{dz}{dx} + 0.5z + 7y = 0 \Rightarrow \frac{dz}{dx} = -0.5z - 7y$

We have: $\frac{dy}{dx} = z$ and $\frac{dz}{dx} = -0.5z - 7y$

$f(x, y, z) = z$ and $g(x, y, z) = -0.5z - 7y$

Hence, $y(0) = 4 \Rightarrow x_0 = 0$, $y_0 = 4$ and $y'(0) = 0 \Rightarrow z_0 = x_0 = 0$

~~We have~~ Fourth-order RK equation:

$y(x) = y_0 + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4)$

We have: $\cdot K_1 = hf(x_0, y_0, z_0) = 0.5f(0, 4, 0) = 0.5 \times 0 = 0$

$\cdot l_1 = hg(x_0, y_0, z_0) = 0.5g(0, 4, 0) = -14$

$\cdot$ ~~$K_2 = h$~~

$\cdot K_2 = hf\left[\left(x_0 + \frac{h}{2}\right), \left(y_0 + \frac{k_1}{2}\right), \left(z_0 + \frac{l_1}{2}\right)\right]$

$= 0.5f\left[\left(0 + \frac{0.5}{2}\right), \left(4 + \frac{0}{2}\right), \left(0 + \frac{-14}{2}\right)\right] = 0.5 \times f(0.25, 4, -7)$

$= -3.5$

$\cdot l_2 = hg\left[\left(x_0 + \frac{h}{2}\right), \left(y_0 + \frac{k_1}{2}\right), \left(z_0 + \frac{l_1}{2}\right)\right] = 12.25$

$\cdot K_3 = hf\left[\left(x_0 + \frac{h}{2}\right), \left(y_0 + \frac{k_2}{2}\right), \left(z_0 + \frac{l_2}{2}\right)\right] = 3.0625$

Ngày .... Tháng ..... Năm........

- $l_3 = hg\left[\left(x_0 + \frac{h}{2}\right), \left(y_0 + \frac{k_2}{2}\right), \left(z_0 + \frac{l_2}{2}\right)\right] = 9.4$

- $k_4 = \cancel{hf\left[\left(x_0 + \frac{h}{2}\right)\right.} \; h f(x_0 + h, y_0 + k_3, z_0 + l_3) = 4.7$

- $l_4 = hg(x_0 + h, y_0 + k_3, z_0 + l_3)$

Therefore $y(x) = y_0 + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) = 4 + \frac{1}{6}(0 + 2(-3.5) + 2 * 3.0625 + 4.7)$

$= 4.638$

$x(0) = x_0 + \frac{1}{6}[k_1 + 2k_2 + 2k_3 + k_4]$

$x = 0$ to 5

$x(0) = 0 + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) = 0.638$; $x(0.5) = 1.138$

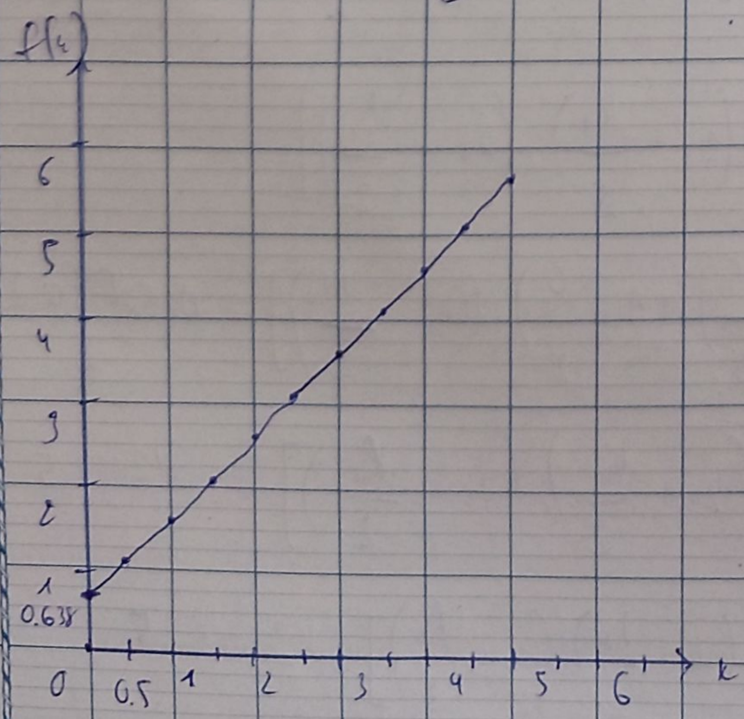
$x(1) = 1 + 0.638 = 1.638$; $x(1.5) = 2.138$

$x(2) = 2.638$; $x(2.5) = 3.138$

$x(3) = 3.638$; $x(3.5) = 4.138$

$x(4) = 4.638$; $x(4.5) = 5.138$

$x(5) = 5.638$;



```
# Code designed by Nguyen Minh Duc - ITITIU21045 - TMC Lab 10
# Question 1

# Given data
t_values = [2, 2.1, 2.2, 2.7, 3, 3.4]
z_values = [6, 7.752, 10.256, 36.576, 66, 145.168]

# Function to calculate divided differences
def divided_difference(x, y):
    n = len(x)
    if n == 1:
        return y[0]
    else:
        return (divided_difference(x[1:], y[1:]) - divided_difference(x[:-1], y[:-1])) / (x[-1] - x[0])

# Newton interpolating polynomial
def newton_interpolation(t, t_values, z_values):
    result = z_values[0]
    for i in range(1, len(t_values)):
        term = divided_difference(t_values[:i+1], z_values[:i+1])
        for j in range(i):
            term *= (t - t_values[j])
        result += term
    return result

# Calculate z at t = 2.5
t_target = 2.5
z_at_t_2_5 = newton_interpolation(t_target, t_values, z_values)
print(f"Newton Interpolation at t = 2.5: {z_at_t_2_5}")
```

```
Newton Interpolation at t = 2.5: 23.19623233908947
```

```
# Using numpy library for Newton interpolation
import numpy as np

# Create a Newton interpolating polynomial
poly_coefficients = np.polyfit(t_values, z_values, len(t_values) - 1)

# Evaluate the polynomial at t = 2.5
z_at_t_2_5_np = np.polyval(poly_coefficients, t_target)
print(f"Numpy Newton Interpolation at t = 2.5: {z_at_t_2_5_np}")
```

```
Numpy Newton Interpolation at t = 2.5: 23.1962323390826
```

```
# Using scipy library for linear spline interpolation
from scipy.interpolate import interp1d

# Create a linear spline interpolation function
linear_spline = interp1d(t_values, z_values, kind='linear', fill_value='extrapolate')

# Evaluate the linear spline at t = 2.5
z_at_t_2_5_spline = linear_spline(t_target)
print(f"Linear Spline Interpolation at t = 2.5: {z_at_t_2_5_spline}")
```

```
Linear Spline Interpolation at t = 2.5: 26.04799999999999
```

```python
# Code designed by Nguyen Minh Duc - ITITIU21045 - TMCLab
# Question 2

# (a)
# Given data
t_values = [0, 1, 2.5, 3, 4.5, 5, 6]
z_values = [26, 15.5, 5.375, 3.5, 2.375, 3.5, 5]

# Function to calculate divided differences
def divided_difference(x, y):
    n = len(x)
    if n == 1:
        return y[0]
    else:
        return (divided_difference(x[1:], y[1:]) - divided_difference(x[:-1], y[:-1])) / (x[-1] - x[0])

# Newton interpolating polynomial
def newton_interpolation(t, t_values, z_values):
    result = z_values[0]
    for i in range(1, len(t_values)):
        term = divided_difference(t_values[:i+1], z_values[:i+1])
        for j in range(i):
            term *= (t - t_values[j])
        result += term
    return result

# Calculate z at t = 3.5
t_target = 3.5
z_at_t_3_5 = newton_interpolation(t_target, t_values, z_values)
print(f"Newton Interpolation at t = 3.5: {z_at_t_3_5}")
```

```
Newton Interpolation at t = 3.5: 2.3333333333333335
```

```python
# (b)
# Using numpy library for Newton interpolation
import numpy as np

# Create a Newton interpolating polynomial
poly_coefficients = np.polyfit(t_values, z_values, len(t_values) - 1)

# Evaluate the polynomial at t = 3.5
z_at_t_3_5_np = np.polyval(poly_coefficients, t_target)
print(f"Numpy Newton Interpolation at t = 3.5: {z_at_t_3_5_np}")
```

```
Numpy Newton Interpolation at t = 3.5: 2.33333333333335
```

```python
# (c)
# Using scipy library for cubic spline interpolation
from scipy.interpolate import CubicSpline

# Create a cubic spline interpolation function
cubic_spline = CubicSpline(t_values, z_values)

# Evaluate the cubic spline at t = 3.5
z_at_t_3_5_spline = cubic_spline(t_target)
print(f"Cubic Spline Interpolation at t = 3.5: {z_at_t_3_5_spline}")
```

```
Cubic Spline Interpolation at t = 3.5: 2.350447082966872
```

```
# Code designed by Nguyen Minh Duc - ITITIU21045 - TMC Lab
# Question 3

import numpy as np
import matplotlib.pyplot as plt

def f(x, y):
    # System of ODEs
    dy1dx = y[1]
    dy2dx = -0.5 * y[1] - 7 * y[0]
    return np.array([dy1dx, dy2dx])

def rk4_step(f, x, y, h):
    # One step of the RK4 method
    k1 = h * f(x, y)
    k2 = h * f(x + 0.5 * h, y + 0.5 * k1)
    k3 = h * f(x + 0.5 * h, y + 0.5 * k2)
    k4 = h * f(x + h, y + k3)
    return y + (k1 + 2 * k2 + 2 * k3 + k4) / 6

def solve_ode_rk4(f, y0, dy0, x_range, h):
    # Initial conditions
    y = np.array([y0, dy0])

    # Lists to store the results
    x_values = [x_range[0]]
    y1_values = [y0]

    # RK4 integration
    for x in np.arange(x_range[0], x_range[1], h):
        y = rk4_step(f, x, y, h)
        x_values.append(x + h)
        y1_values.append(y[0])

    return np.array(x_values), np.array(y1_values)

# Initial conditions
y0 = 4
dy0 = 0

# Solve the ODE using RK4
x_range = (0, 5)
h = 0.5
x_values, y_values = solve_ode_rk4(f, y0, dy0, x_range, h)

# Plotting the results
plt.plot(x_values, y_values, label='RK4 Method')
plt.xlabel('x')
plt.ylabel('y')
plt.title('Solution of the ODE using RK4 Method')
plt.legend()
plt.grid(True)
plt.show()
```