2023

ASSIGNMENT 9

NUMERICAL METHODS (CS-406)

RUNGE KUTTA 4^{TH} ORDER IMPLEMENTATION IN PYTHON

SUBMITTED BY: UMANG KANCHAN

SUBMITTED TO: DR. AYESHA CHOUDHARY

```python
def runge_kutta(f, x0, y0, xf, h):
    # Initialize x and y lists with initial values
    x_list = [x0]
    y_list = [y0]

    # Loop until final value of x is reached
    while x0 < xf:
        # Calculate k1, k2, k3, k4 using the 4th order Runge-Kutta method
        k1 = h * f(x0, y0)
        k2 = h * f(x0 + h/2, y0 + k1/2)
        k3 = h * f(x0 + h/2, y0 + k2/2)
        k4 = h * f(x0 + h, y0 + k3)

        # Calculate the next value of y using the weighted sum of k1, k2, k3, k4
        y_next = y0 + (k1 + 2*k2 + 2*k3 + k4) / 6

        # Update x, y and append them to the respective lists
        x0 += h
        y0 = y_next
        x_list.append(x0)
        y_list.append(y_next)

    # Return the x and y lists
    return x_list, y_list


# Get input from the user
ode_eqn = input("Enter the differential equation in terms of x and y: ")
y0 = float(input("Enter the initial value of y: "))
x0 = float(input("Enter the initial value of x: "))
xf = float(input("Enter the final value of x: "))
h = float(input("Enter the step size: "))

# Define the function f(x, y) from the input equation
f = lambda x, y: eval(ode_eqn)

# Compute the numerical solution using the 4th order Runge-Kutta method
x_vals, y_vals = runge_kutta(f, x0, y0, xf, h)

# Print the results
print("Numerical solution:")
for x, y in zip(x_vals, y_vals):
    print(f"x = {x:.2f}, y = {y:.4f}")
```

SAMPLE OUTPUT

Enter the differential equation in terms of x and y: x+y

Enter the initial value of y: 0

Enter the initial value of x: 0

Enter the final value of x: 1

Enter the step size: 0.1

Numerical solution:

x = 0.00, y = 0.0000

x = 0.10, y = 0.0052

x = 0.20, y = 0.0214

x = 0.30, y = 0.0499

x = 0.40, y = 0.0918

x = 0.50, y = 0.1487

x = 0.60, y = 0.2221

x = 0.70, y = 0.3138

x = 0.80, y = 0.4255

x = 0.90, y = 0.5596

x = 1.00, y = 0.7183

x = 1.10, y = 0.9042