

PATUAKHALI SCIENCE AND TECHNOLOGY UNIVERSITY

COURSE CODE CCE 312
Numerical Methods Sessional

SUBMITTED TO:

Prof. Dr. Md Samsuzzaman

Department of Computer and Communication Engineering
Faculty of Computer Science and Engineering

SUBMITTED BY:

Md. Sharafat Karim

ID: 2102024,

Registration No: 10151

Faculty of Computer Science and Engineering

Assignment 12

Assignment title: Euler's Method

Date of submission: 20 Sat, Sep 2025



Differential Equations

Sharafat Karim

A differential equation is a mathematical equation that relates a function with its derivatives. In simpler terms, it describes how a quantity changes in relation to another quantity. Differential equations are fundamental in various fields such as physics, engineering, and economics, as they model dynamic systems and processes.

Euler's Method

Euler's method is a simple and widely used numerical technique for solving ordinary differential equations (ODEs) with a given initial value. It is particularly useful for approximating solutions to first-order ODEs of the form:

$$\frac{dy}{dx} = f(x, y) \quad (1)$$

with an initial condition $y(x_0) = y_0$.

Let's consider an example,

$$\frac{dy}{dx} = x + y \quad (2)$$

$$y(0) = 1 \quad (3)$$

Let's import libraries first,

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

Our differential equation,

$$\frac{dy}{dx} = x + y \quad (4)$$

```
def f(x, y):
    return x + y
```

And our euler function,

```
def euler_method(f, x0, y0, h=0.1, n=100):
    x_values = [x0]
    y_values = [y0]

    for i in range(n):
        y0 = y0 + h * f(x0, y0)
        x0 = x0 + h
        x_values.append(x0)
        y_values.append(y0)

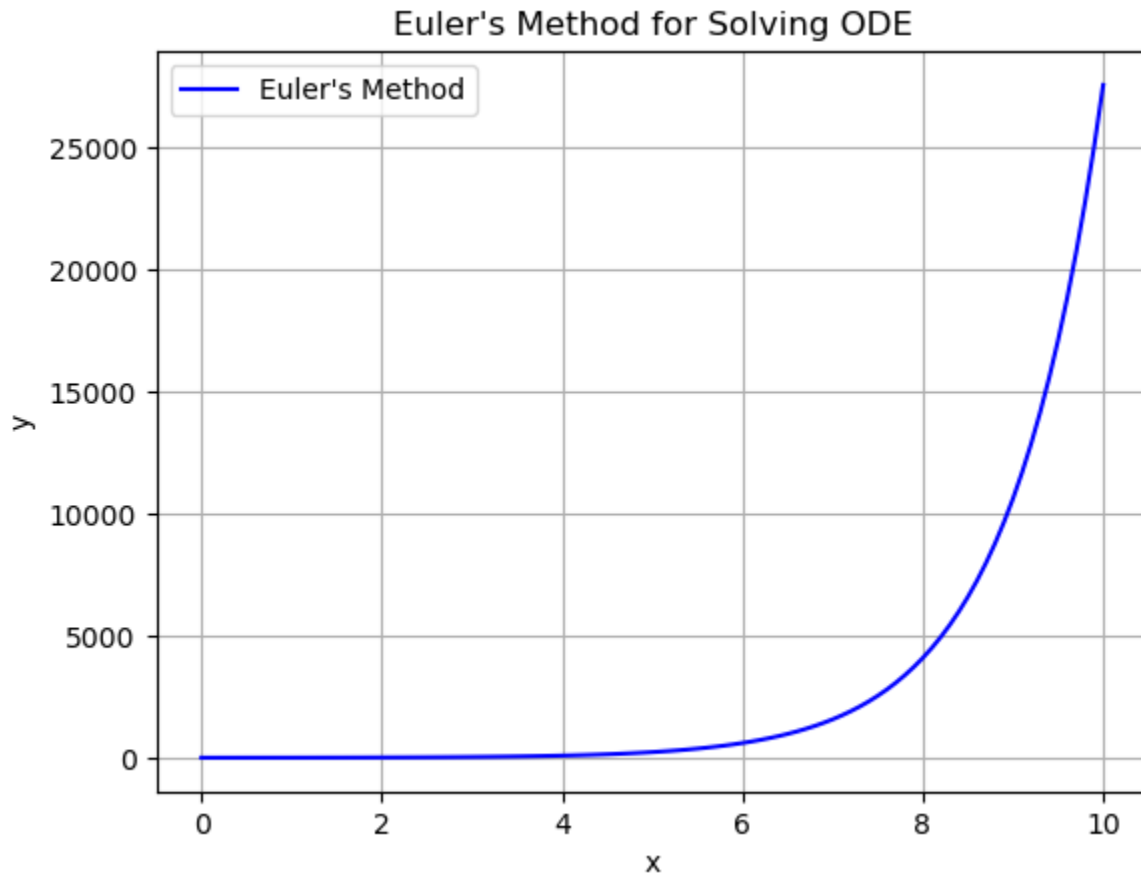
    return np.array(x_values), np.array(y_values)
```

Let's plot and visualize the results,

```
# Initial conditions and parameters
x0 = 0
y0 = 1
h = 0.1 # Step size
n = 100 # Number of steps
x_values, y_values = euler_method(f, x0, y0, h, n)

# Plotting the results
plt.plot(x_values, y_values, label="Euler's Method", color='blue')
plt.title("Euler's Method for Solving ODE")
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
```

```
plt.grid()
```



Numerical Methods
Polynomial Regression

Numerical Methods
Integrals

