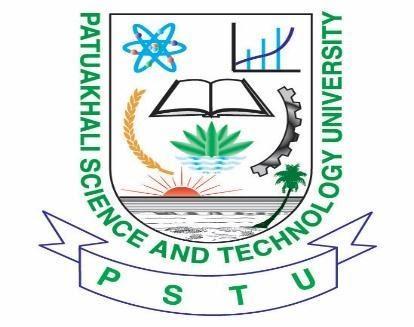
**Project Report**



|  |
| --- |
| Remarks : |

**Course Title : Numerical Methods**

**Lab Report on Simulation of Physical Systems using Euler's Method in Python**

**Course Code: CCE 312**

**Submitted To**

**Dr. Md Samsuzzaman Sobuz**

**Professor,**

**Department of Computer and Communication Engineering,**

**Faculty of Computer Science and Engineering.**

**Submitted By**

**Md.Abdur Rakib**

**ID No:1902066**

**Reg No:08772**

**Level- 3, Semester- 1**

**Session: 2019-2020**

**Patuakhali Science and Technology University**

**Introduction:**

The project aims to utilize Euler's Method in Python for simulating the behavior of physical systems over time. It encompasses the implementation of Euler's Method algorithm, mathematical model formulation, system simulation, visualization, and user-friendly program development.

**Euler's Method Overview:**

Euler's Method is a numerical technique for solving ordinary differential equations by approximating the solution step by step. It's particularly suitable for dynamic scenarios like particle motion or fluid dynamics.

**Project Objectives:**

Implement Euler's Method for numerical simulation.

Formulate mathematical models representing physical systems' behavior.

Apply Euler's Method to simulate system evolution.

Visualize and analyze simulation results.

Develop a user-friendly Python program for interaction and analysis.

**Project Components:**

**Euler's Method Implementation:** Algorithm designed to handle diverse physical systems.

Mathematical Model Formulation: Equations representing system behavior with defined parameters.

**System Simulation:** Application of Euler's Method for numerical simulation.

Visualization and Analysis: Graphical representation and data analysis of simulation results.

**User-friendly Python Program**: Interactive interface for input, visualization, and analysis.

**Documentation and User Guide:** Comprehensive documentation and user guide for understanding and utilizing the simulation tool.

**Analysis and Implementation:**

The project successfully implemented Euler's Method for numerical simulation, formulated mathematical models, and simulated system behavior. Graphical visualization and user-friendly interface enhance user interaction and understanding. Additionally, the incorporation of fluid dynamics simulation demonstrates real-life applications of Euler's Method.

**Python Sample Code For Implementation:**

**For Value Calculation :**

def f(x, y, equation):  
 return eval(equation)  
  
  
*# Euler method*def euler(x0, y0, xn, n, equation):  
 *# Calculating step size* h = (xn - x0) / n  
  
 results = []  
  
 for i in range(n):  
 slope = f(x0, y0, equation)  
 yn = y0 + h \* slope  
 results.append([x0, y0, slope, yn])  
 y0 = yn  
 x0 = x0 + h  
  
 slope = f(xn, yn, equation)   
 results.append([xn, yn, slope, yn])   
  
 return results  
  
  
def calculate():try:  
 x0 = float(entry\_x0.get())  
 y0 = float(entry\_y0.get())  
 xn = float(entry\_xn.get())  
 step = float(entry\_step.get())   
 except ValueError:  
 result\_text.config(state=tk.NORMAL)  
 result\_text.delete(1.0, tk.END). . .

**For Fluid dynamics :**

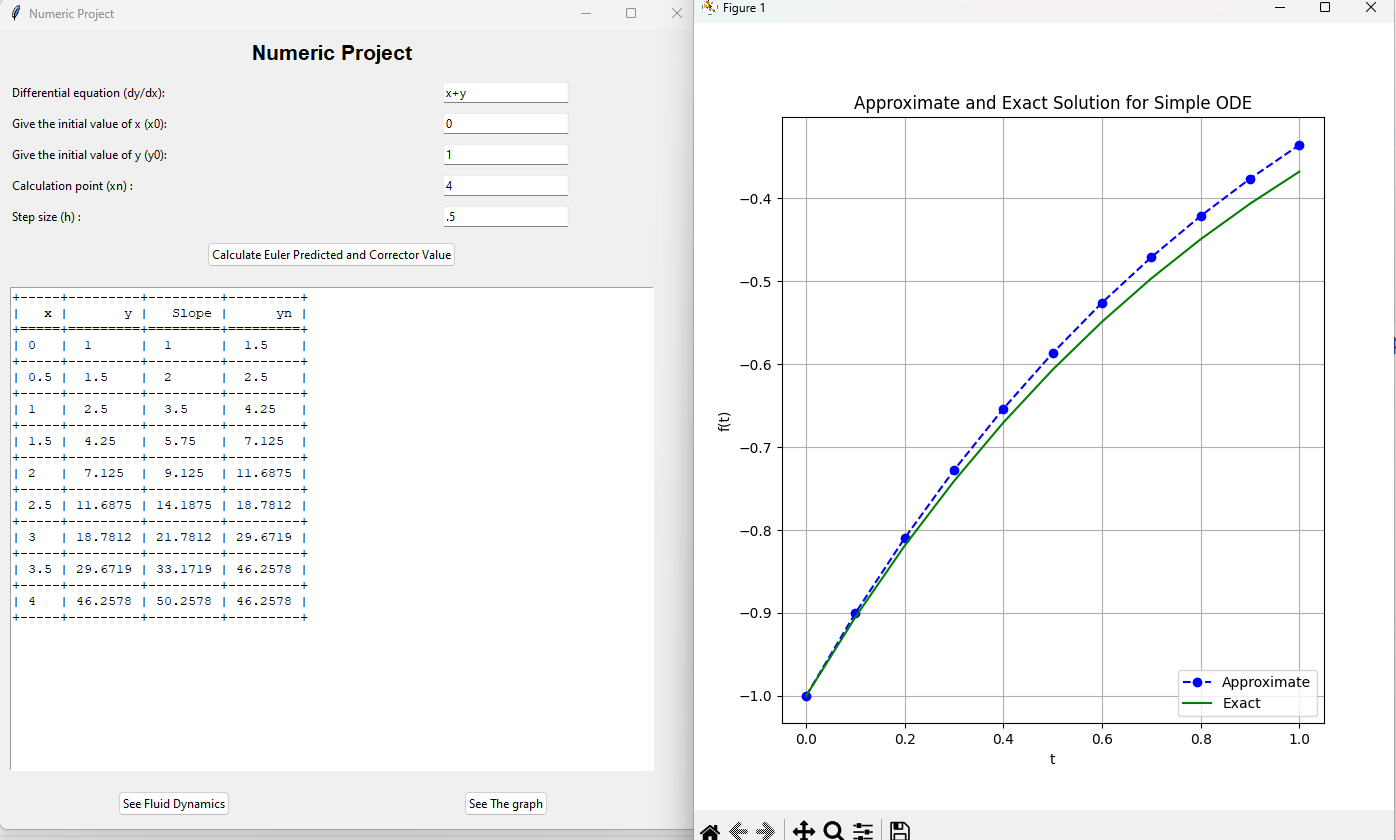
import tkinter as tk  
from tkinter import ttk  
import matplotlib.pyplot as plt  
from matplotlib.animation import FuncAnimation  
from matplotlib.backends.backend\_tkagg import FigureCanvasTkAgg  
import numpy as np  
  
*# Parameters*length = 20.0   
width = 10.0   
num\_points\_x = 100   
num\_points\_y = 50   
dx = length / num\_points\_x   
dy = width / num\_points\_y   
velocity = 1.0   
diffusion\_coefficient = 0.1   
time\_step = 0.01   
num\_frames = 2000   
  
  
fluid\_density = np.zeros((num\_points\_y, num\_points\_x))  
fluid\_density[num\_points\_y // 4:num\_points\_y // 2,  
 num\_points\_x // 4:num\_points\_x // 2] = 1.0

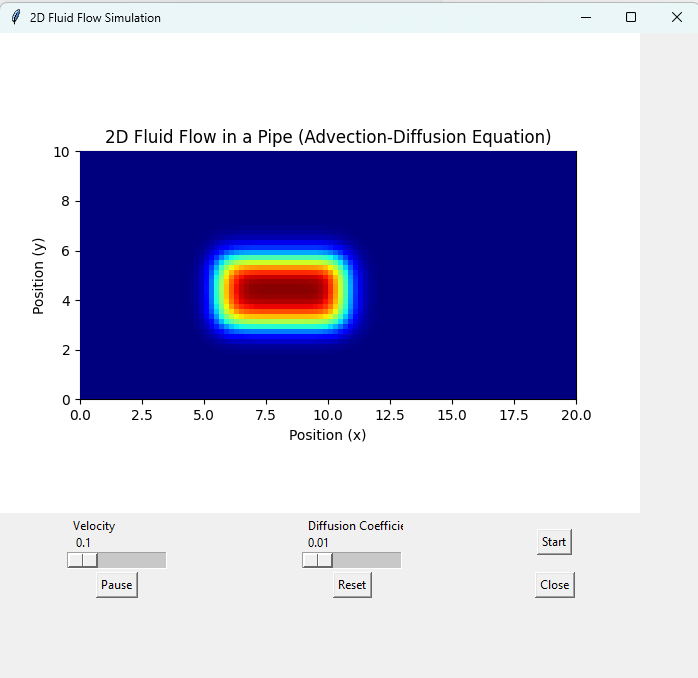
def euler\_method\_advection\_diffusion(u\_initial, velocity, diffusion\_coefficient, dt, dx, dy):

**For Graph:**

import matplotlib.pyplot as plt  
import numpy as np  
  
*# Define parameters*f = lambda t, s: np.exp(-t) *# ODE*h = 0.1 *# Step size*t = np.arange(0, 1 + h, h) *# Numerical grid*s0 = -1 *# Initial Condition  
  
  
# Explicit Euler Method*s = np.zeros(len(t))  
s[0] = s0  
  
for i in range(0, len(t) - 1):  
 s[i + 1] = s[i] + h \* f(t[i], s[i])

**Output Window :**

****

****

**Conclusion:**

The project demonstrates the efficacy of Euler's Method in simulating physical systems, providing insights into dynamic behavior. It serves as a valuable tool for computational physics, offering a practical approach to studying and analyzing various scenarios.

**Potential Extensions:**

Investigate the impact of varying parameters on system behavior.

Extend the application to simulate complex interconnected systems.

Overall, the project contributes to advancing computational physics and provides a versatile tool for studying real-world phenomena.