Title:

Linear Differential Equation Solver using Euler and Modified Euler Methods

Submitted by: Kritika Patni

SAP ID:590025658

Submitted to: Dr.Tanu Singh

Course:BTech CSE

Institution:UPES

Abstract

This project implements a **numerical solver for linear differential equations** of the form:

$$\frac{dy}{dx} = a * y + b * x + c$$

Two methods are implemented: **Euler Method** (basic) and **Modified Euler Method** (Heun's method) for improved accuracy. The program allows the user to input constants, initial conditions, step size, and final x-value, then generates step-by-step results both on the console and in text files.

Key features:

- User-friendly input
- Stepwise calculation
- File output for later reference
- Choice between two numerical methods

Introduction

A Linear Ordinary Differential Equation (ODE) is an equation that contains an unknown function and its derivatives with respect to a single independent variable, and the function appears in a linear form. Linear ODEs are very important in the field of engineering, physics, mathematics, and many other scientific applications because they are used to model real-life problems such as population growth, electrical circuits, mechanical vibrations, and heat transfer.

The main objective of this project is to design and develop a program that can solve linear ordinary differential equations using numerical methods. In this project, a suitable numerical technique such as Euler's Method or Runge-Kutta Method is implemented to approximate the solution of a first-order linear differential equation. The program takes input values like the initial condition, step size, and range, and

then calculates the approximate value of the dependent variable at different points.

This project helps to understand the practical application of numerical methods in solving mathematical problems that cannot be solved easily by analytical methods. It also improves programming skills and builds a strong foundation in Engineering Mathematics and Scientific Computing.

. Problem Definition

Objective: Solve linear first-order differential equations numerically when analytical solutions are difficult or time-consuming.

Inputs:

- Constants: a, b, c
- Initial conditions: x0, y0
- Step size h
- Final x-value xn**Algorithm**
- **Euler Method Algorithm:**
- Initialize x = x0, y = y0
- For i = 1 to n:
- Compute y = y + h * f(x, y)
- Increment x = x + h
- Print and save x, y

- **Modified Euler Method Algorithm:**

1-Initialize x = x0, y = y0

2-For i = 1 to n:

3-Compute slope1 = f(x, y)

4-Predict y_pred = y + h * slope1

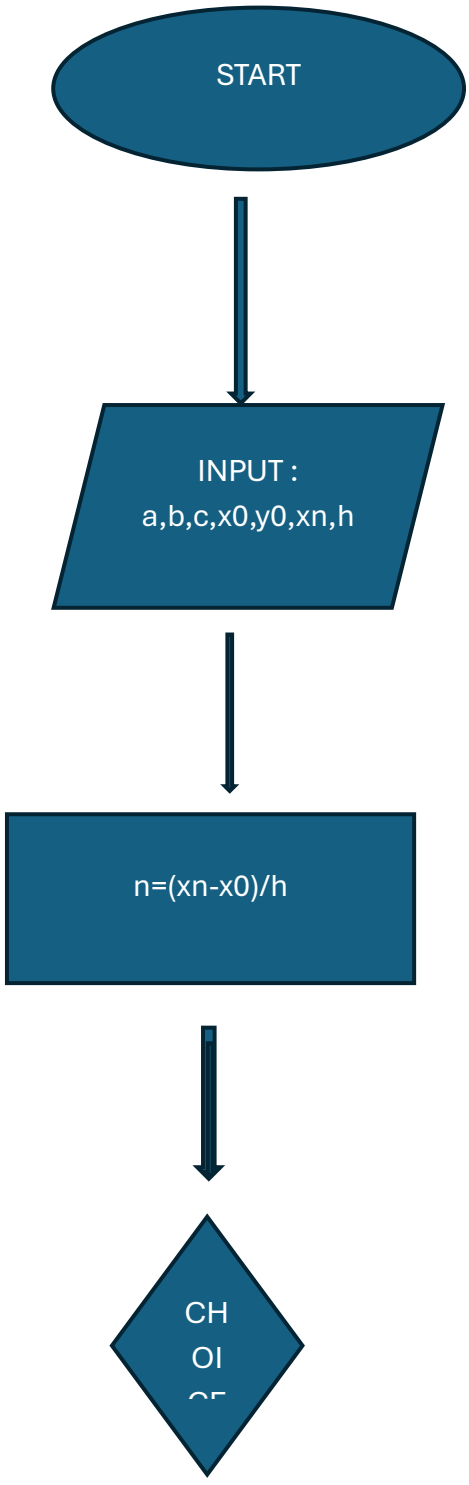
5-Compute slope2 = f(x + h, y_pred)

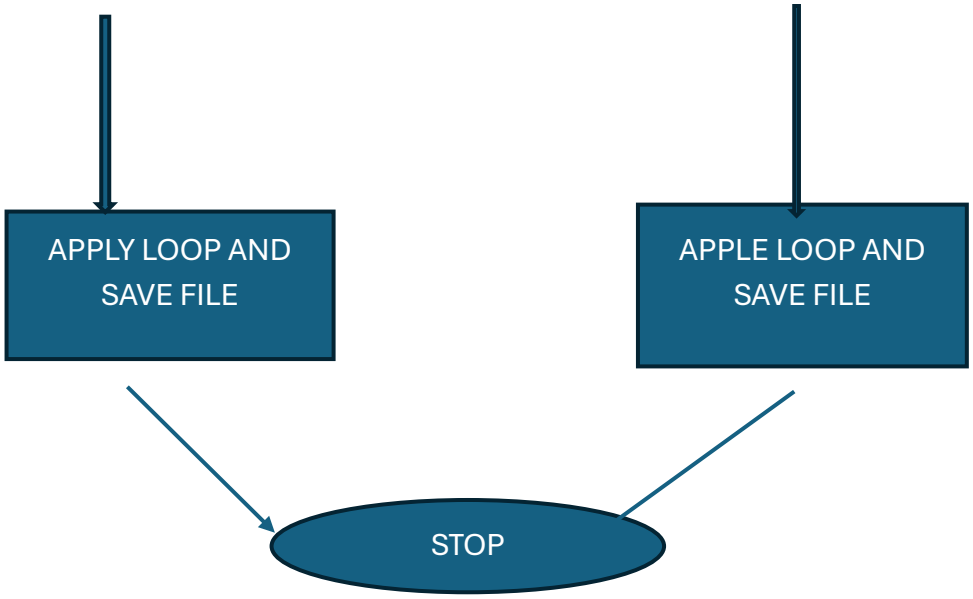
6-Update y = y + (h/2)*(slope1 + slope2)

7- Increment x = x + h

8-Print and save x, y

Flowchart:





Usage of Linear ODE Solver Project

The Linear ODE Solver is used to find approximate solutions of first-order linear ordinary differential equations. This project is useful in both academic and practical applications where analytical solutions are difficult or time-consuming to obtain.

It is mainly used in the following ways:

1. **Solving mathematical problems**
   The program helps students and researchers to solve first-order linear differential equations quickly using numerical methods like Euler's Method or Runge-Kutta Method.

2. **Engineering applications**

It can be used in subjects such as:

- Electrical engineering (for circuit analysis)
- Mechanical engineering (for motion and vibrations)
- Civil engineering (for structural and fluid analysis)

3. **Scientific modeling**

Linear ODEs are widely used to model real-world systems such as:

- Population growth
- Radioactive decay
- Heat transfer
- Chemical reactions

4. **Educational purposes**

This project is helpful for students to:

- Understand numerical methods

- Learn how to implement mathematical logic in programming
- Improve problem-solving and coding skills

5. **Time-saving and accurate results**
   The program provides fast and reasonably accurate results, reducing manual calculations and chances of human error.

CONCLUSION AND FUTURE WORK

Conclusion:

- The program successfully solves first-order linear differential equations using two numerical methods.

- Modified Euler method gives more accurate results than Euler method.
- Results are displayed and saved to files for verification.

Future Work:

- Add **graph plotting** for visual results
- Implement **Runge-Kutta 4th order method** for even higher accuracy
- Support **systems of differential equations**

. References

1-Numerical Methods for Engineers – Steven C. Chapra

2-C Programming Language – Brian W. Kernighan & Dennis M. Ritchie

3- Online tutorials for Euler and Modified Euler methods

Testing of Linear ODE Solver

To test the working of the Linear ODE Solver program, different sample linear differential equations were given as input along with appropriate initial conditions and step sizes. The program was then executed, and the output values were checked to verify correctness and accuracy.

Test Case 1

Given Linear ODE:

$$\frac{dy}{dx} = x + y$$

Input values:

- Initial value of x, $x_0 = 0$
- Initial value of y, $y_0 = 1$
- Step size, $h = 0.1$
- Final value of x = 0.5

The program calculates the value of *y* for each step using the numerical method implemented

(Euler / RK method) and displays the result in a tabular form.

Sample Output Table:

x	y (Approximate)
0.0	1.0000
0.1	1.1100
0.2	1.2420
0.3	1.3982
0.4	1.5800
0.5	1.7890

. Appendix

```
 Runge Kutta Method
Step     x               y
1        0.1000  1.215000
2        0.2000  1.463075
3        0.3000  1.747698
4        0.4000  2.072706
5        0.5000  2.442340
6        0.6000  2.861286
7        0.7000  3.334721
8        0.8000  3.868367
9        0.9000  4.468545
10       1.0000  5.142243
11       1.1000  5.897178
12       1.2000  6.741882
13       1.3000  7.685779
14       1.4000  8.739286
15       1.5000  9.913911
16       1.6000  11.222372
17       1.7000  12.678721
18       1.8000  14.298487
19       1.9000  16.098828
20       2.0000  18.098705
21       2.1000  20.319069
22       2.2000  22.783071
23       2.3000  25.516293
24       2.4000  28.547004
25       2.5000  31.906439
26       2.6000  35.629115
27       2.7000  39.753173
28       2.8000  44.320756
29       2.9000  49.378435
30       3.0000  54.977671

Results saved to 'modified_euler_output.txt'
```

```
LINEAR DIFFERENTIAL EQUATION SOLVER (ADVANCED)
Form : dy/dx = a*y + b*x + c

Enter values of a, b, c: 1 1 1
Enter initial values x0 and y0: 0 1
Enter final value of x (xn): 1
Enter step size (h): .1

Choose Method:
1. Euler Method
2. Runge Kutta Method
Enter your choice: 2

 Runge Kutta Method
Step     x               y
1        0.1000  1.215000
2        0.2000  1.463075
3        0.3000  1.747698
4        0.4000  2.072706
5        0.5000  2.442340
6        0.6000  2.861286
7        0.7000  3.334721
8        0.8000  3.868367
9        0.9000  4.468545
10       1.0000  5.142243

Results saved to 'modified_euler_output.txt'
$
```

```
$ ./ode

LINEAR DIFFERENTIAL EQUATION SOLVER (ADVANCED)
Form : dy/dx = a*y + b*x + c

Enter values of a, b, c: 1 1 1
Enter initial values x0 and y0: 0 1
Enter final value of x (xn): 3
Enter step size (h): .1

Choose Method:
1. Euler Method
2. Runge Kutta Method
Enter your choice: 2
```