ODE VISUALIZATION USING MATLAB

Afif hossain Irfan
ID – C211005

INTRODUCTION

Ordinary Differential Equations (ODEs) are pivotal in modeling various engineering systems and processes, encompassing fields such as mechanics, electrical circuits, and chemical reactions. This project, titled "ODE Visualization Using MATLAB" aims to develop a MATLAB based tool for numerically solving and visualizing the behavior of solutions to ODEs. By implementing multiple numerical methods and providing graphical representations of the results, this tool will assist engineers and researchers in understanding and analyzing ODEs more effectively.

Features

1. **Functionality:** Implement core functionalities to solve and visualize ODEs using MATLAB.
2. **Method Support:** Support multiple numerical methods for solving ODEs.
3. **Visualization:** Provide clear and informative plots of the solutions.
4. **User Control:** Allow users to adjust parameters such as initial values, step sizes, and the point of evaluation.
5. **Comparison and Analysis:** Offer features to compare the performance and accuracy of different methods.

Flowchart of the Process

Step 1: Initialize the MATLAB Environment

Step 2: Handle User Input

Step 3: Calculate the Number of Steps

Step 4: Implement, Execute and Plotting Euler, Heun’s, Midpoint and Range-Kutta Method

Step 5: Compare All Methods

Function Descriptions

1. **Euler Method:** A basic iterative approach that uses the slope at the current point to estimate the next value.

$$yi+1 = yi + h\ f(xi\ ,\ yi\ )$$

2. **Heun's Method:** An improved version of Euler's method that uses the average of slopes at the current point and the predicted next point for better accuracy.

$$k1 = hf(xi\ ,\ yi\ )$$
$$k2 = hf(xi + h,\ yi + k1\ )$$

$$yi+1 = yi + 1/2\ (k1 + k2\ )$$

Function Descriptions

3 **Midpoint Method:** A method that uses the slope at the midpoint of the interval to estimate the next value, offering improved accuracy over Euler's method.

$$k1 = hf(xi\ , yi\ )$$

$$k2 = hf\ (xi + h\ 2\ , yi + k1\ 2\ )$$

$$yi{+}1 = yi + k2$$

Function Descriptions

4. **Runge-Kutta Method:** A higher-order method that computes multiple intermediate slopes and combines them to provide a highly accurate solution.

$$k1 = hf(xi\ , yi\ )$$
$$k2 = hf\ (xi + h\ 2\ , yi + k1\ 2\ )$$
$$k3 = hf\ (xi + h\ 2\ , yi + k2\ 2\ )$$
$$k4 = hf(xi + h\ , yi + k3\ )$$

$$yi{+}1 = yi + 1/6\ (k1 + 2k2 + 2k3 + k4\ )$$

DATASET

Chemical Reaction Kinetics :

The Ordinary Differential Equation (ODE) [dy/dx + 20y = 7e ^(−0.5x)] , the initial condition y(0) = 1

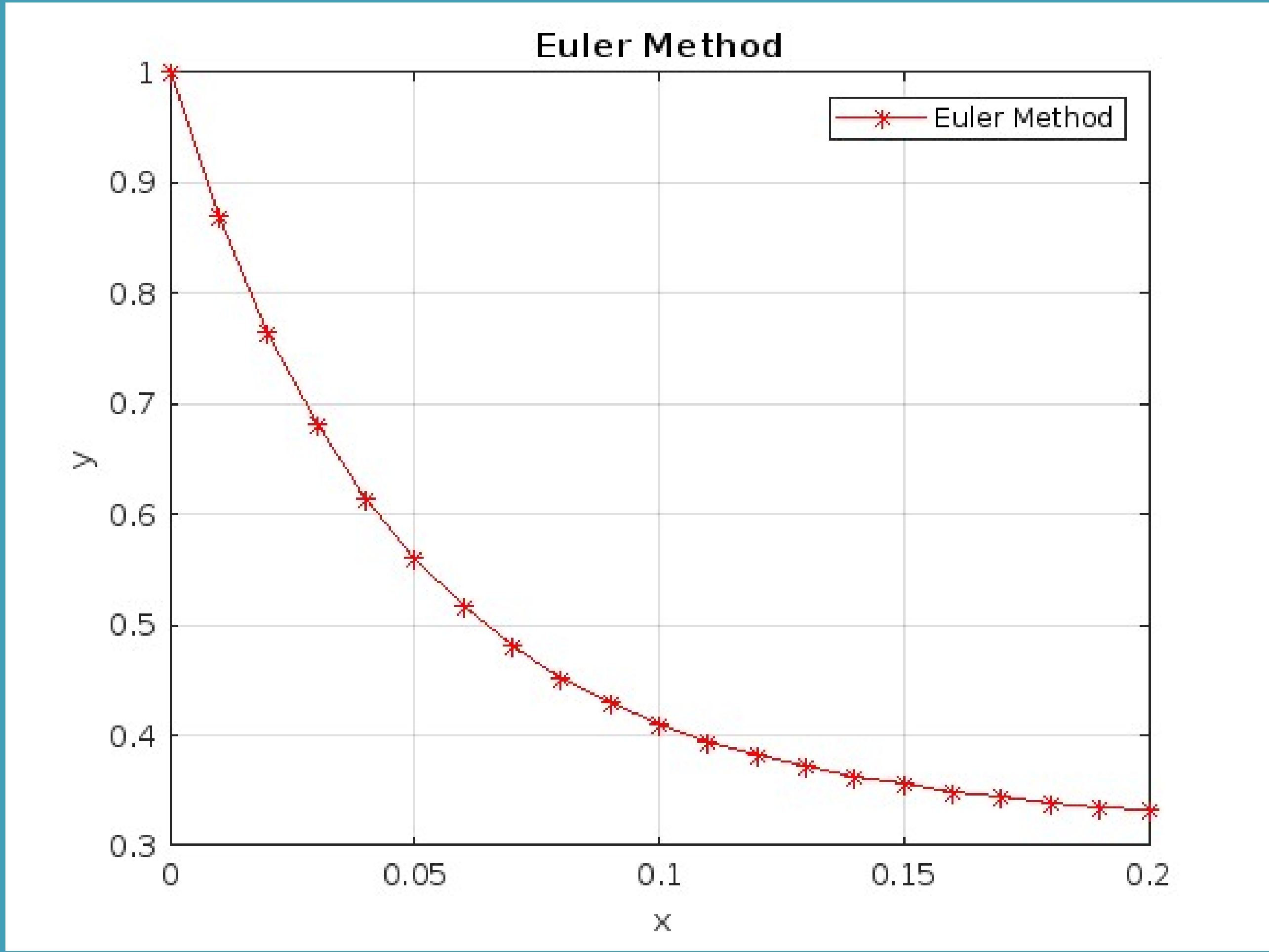
In the context of chemical reaction kinetics, this ODE can represent a first-order reaction where the concentration of a substance y changes over time x.

1. dx/dy: This term represents the rate of change of the concentration y with respect to time x.
2. 20y: This term can be seen as a linear decay term where 20 is a constant rate coefficient. It indicates that the substance is decaying or reacting at a rate proportional to its current concentration.
3. 7e^{-0.5x} : This term represents an external input or source term that adds to the concentration y. The exponential function e^{-0.5x} suggests that this input decreases over time.

RESULT

Euler Method:

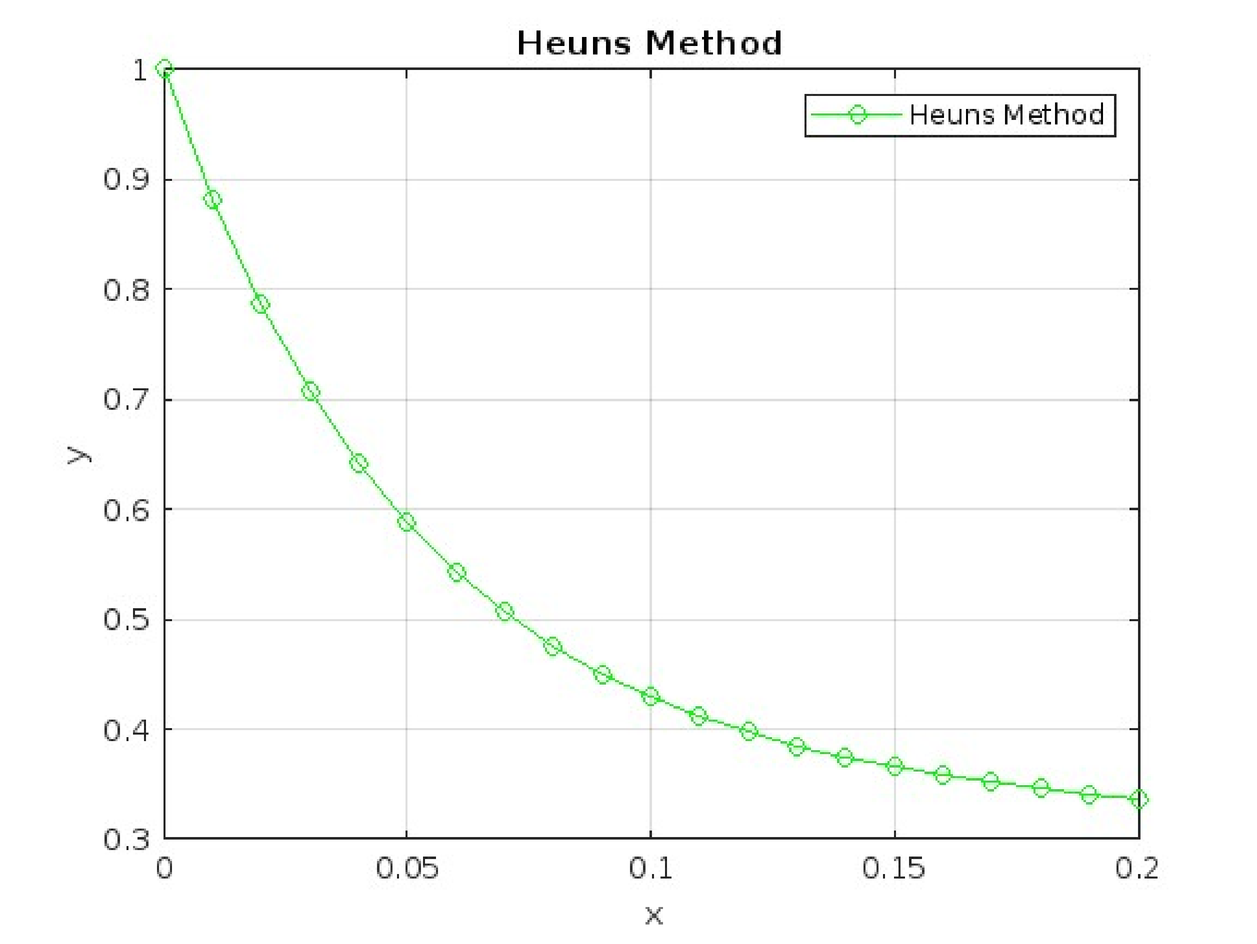
```
y(   0.0100) =    0.870000
y(   0.0200) =    0.765651
y(   0.0300) =    0.681824
y(   0.0400) =    0.614417
y(   0.0500) =    0.560148
y(   0.0600) =    0.516390
y(   0.0700) =    0.481043
y(   0.0800) =    0.452427
y(   0.0900) =    0.429197
y(   0.1000) =    0.410277
y(   0.1100) =    0.394808
y(   0.1200) =    0.382100
y(   0.1300) =    0.371604
y(   0.1400) =    0.362878
y(   0.1500) =    0.355570
y(   0.1600) =    0.349398
y(   0.1700) =    0.344136
y(   0.1800) =    0.339605
y(   0.1900) =    0.335659
y(   0.2000) =    0.332183
```



RESULT

Heun's Method:

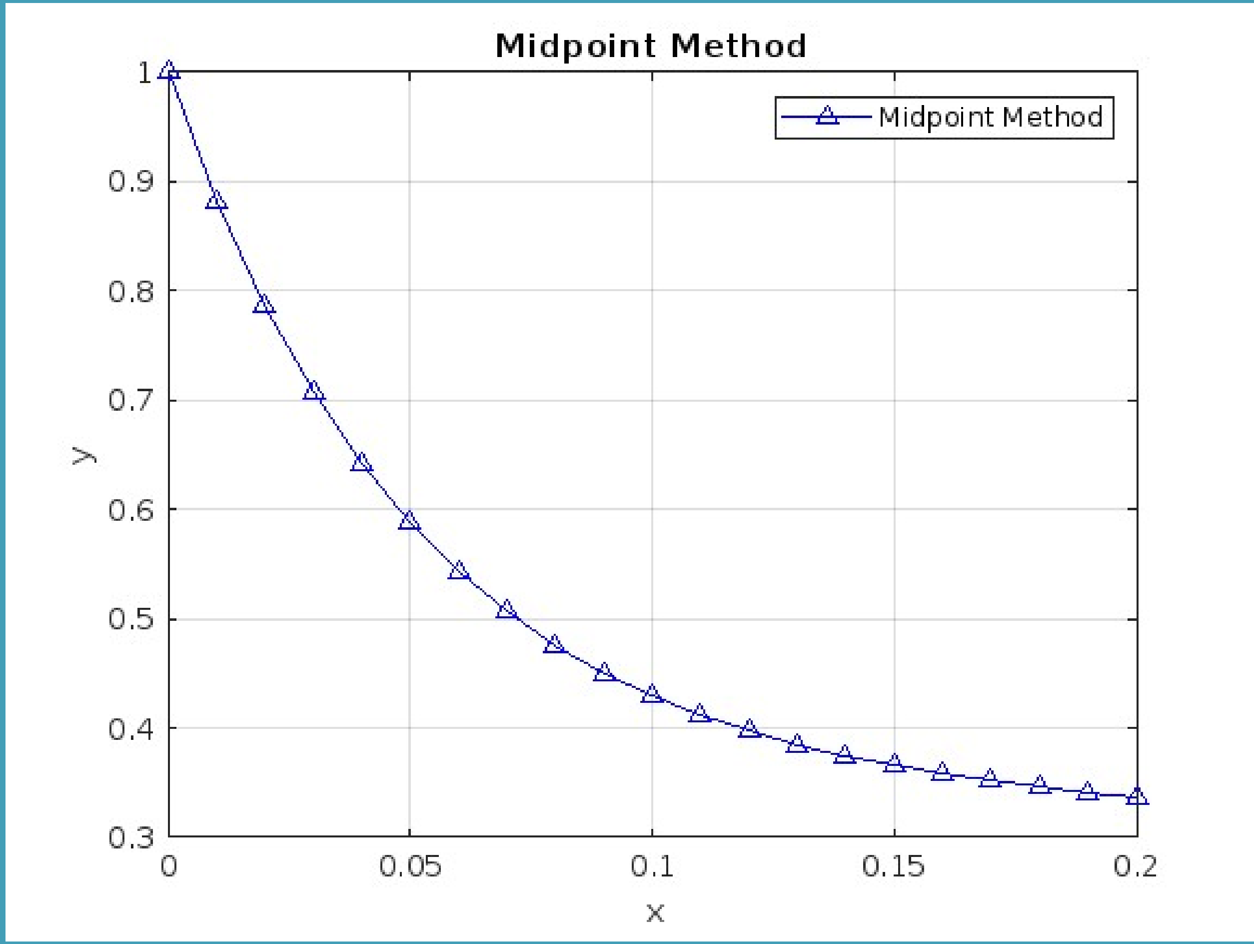
```
y(  0.0100) =   0.882825
y(  0.0200) =   0.786429
y(  0.0300) =   0.707072
y(  0.0400) =   0.641689
y(  0.0500) =   0.587767
y(  0.0600) =   0.543243
y(  0.0700) =   0.506428
y(  0.0800) =   0.475935
y(  0.0900) =   0.450629
y(  0.1000) =   0.429577
y(  0.1100) =   0.412014
y(  0.1200) =   0.397315
y(  0.1300) =   0.384965
y(  0.1400) =   0.374543
y(  0.1500) =   0.365703
y(  0.1600) =   0.358163
y(  0.1700) =   0.351689
y(  0.1800) =   0.346091
y(  0.1900) =   0.341212
y(  0.2000) =   0.336926
```



RESULT

Midpoint Method:

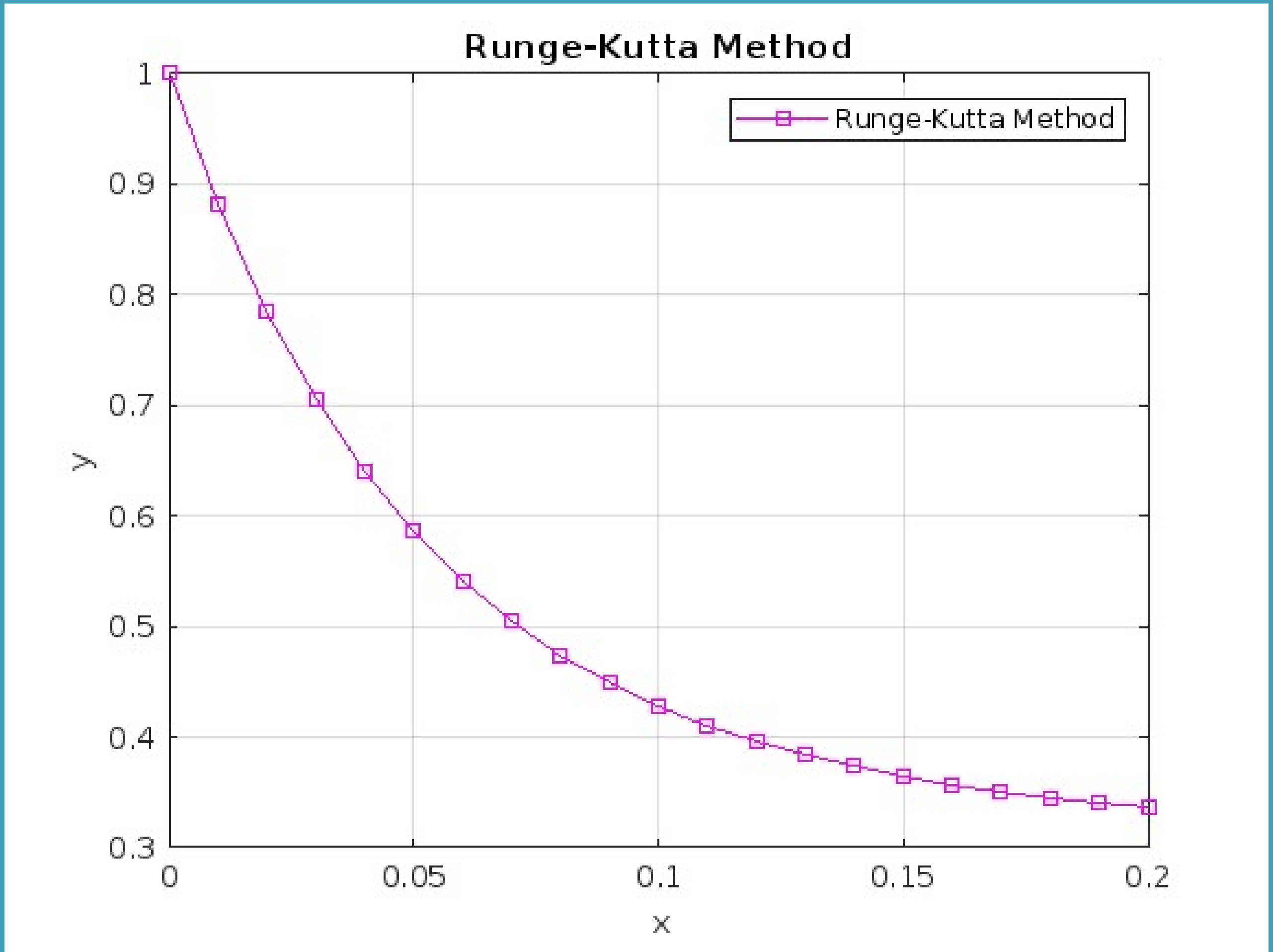
```
y(  0.0100) =   0.882825
y(  0.0200) =   0.786429
y(  0.0300) =   0.707072
y(  0.0400) =   0.641689
y(  0.0500) =   0.587766
y(  0.0600) =   0.543242
y(  0.0700) =   0.506427
y(  0.0800) =   0.475934
y(  0.0900) =   0.450628
y(  0.1000) =   0.429576
y(  0.1100) =   0.412013
y(  0.1200) =   0.397314
y(  0.1300) =   0.384964
y(  0.1400) =   0.374542
y(  0.1500) =   0.365702
y(  0.1600) =   0.358162
y(  0.1700) =   0.351687
y(  0.1800) =   0.346089
y(  0.1900) =   0.341211
y(  0.2000) =   0.336925
```



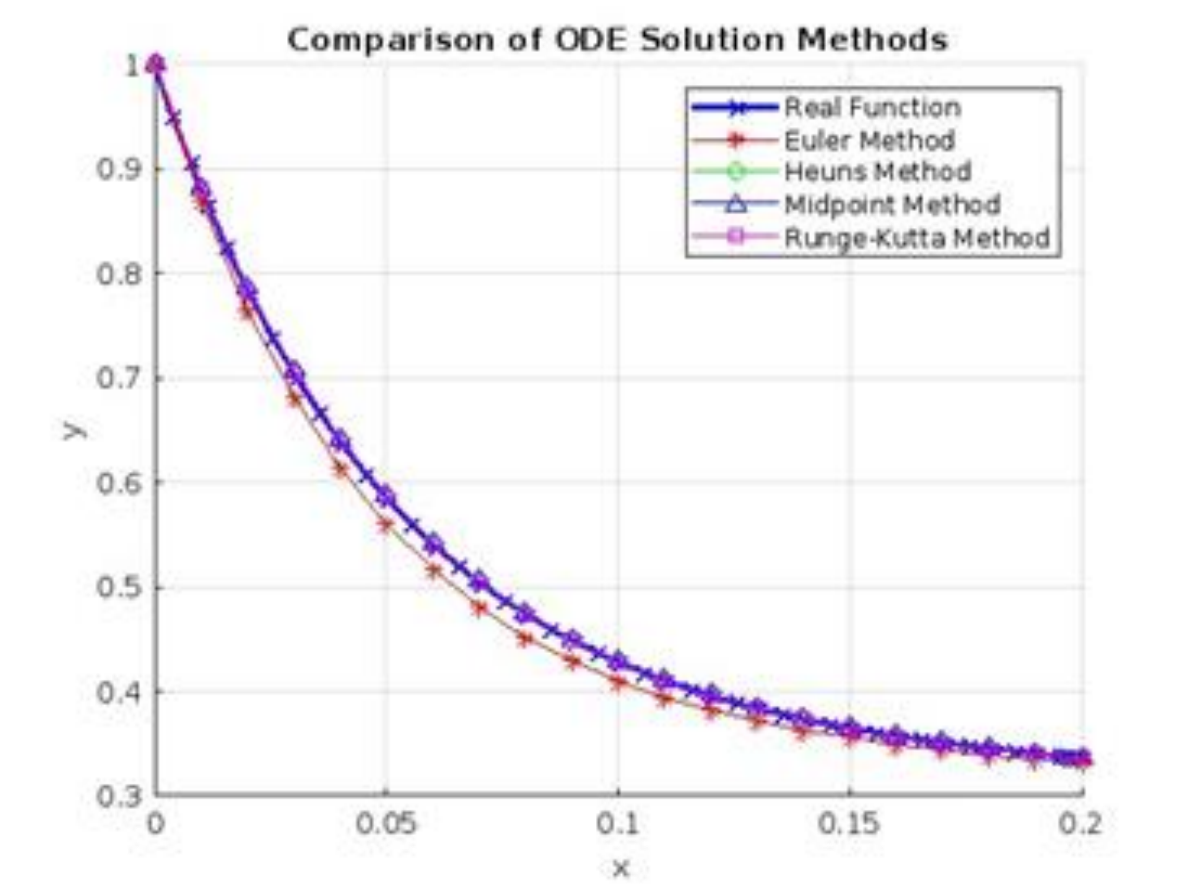
RESULT

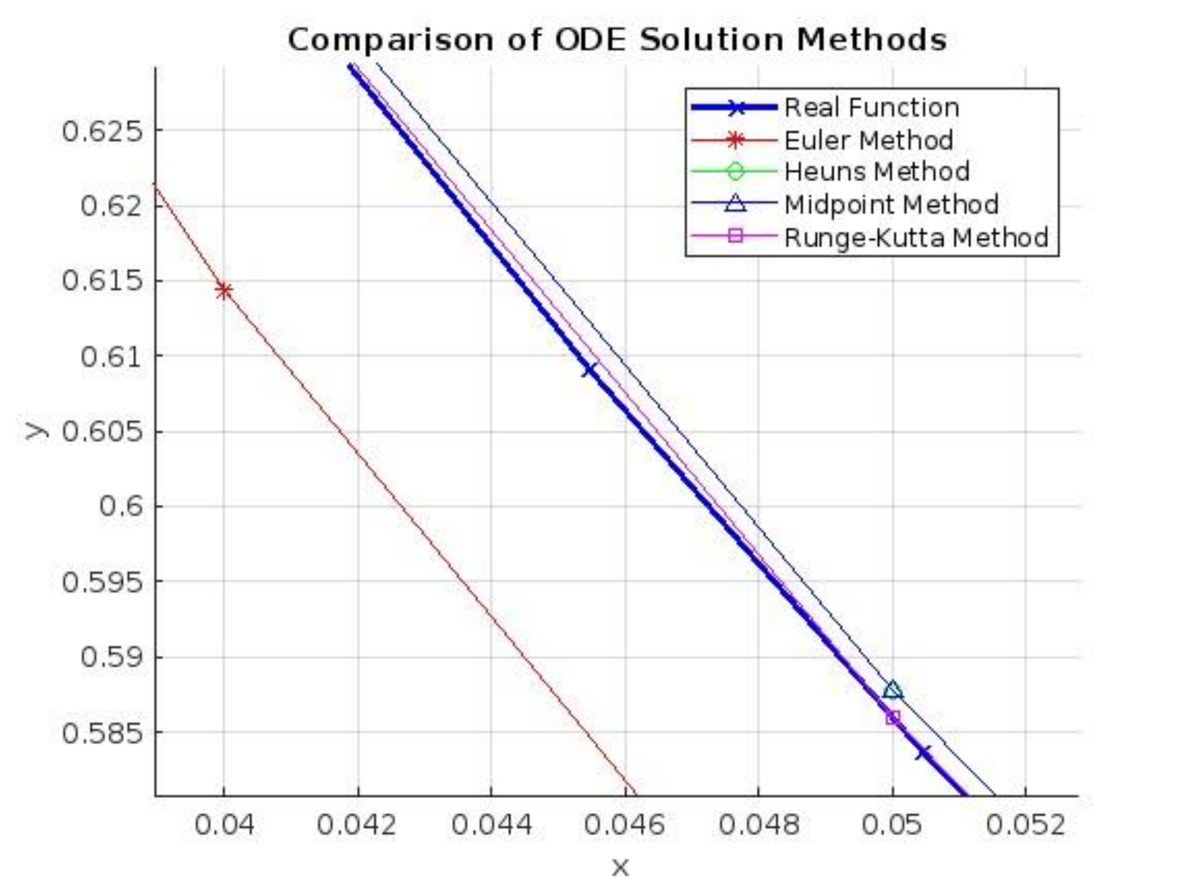
Range-Kutta Method:

```
y(  0.0100) =   0.882013
y(  0.0200) =   0.785098
y(  0.0300) =   0.705436
y(  0.0400) =   0.639901
y(  0.0500) =   0.585935
y(  0.0600) =   0.541442
y(  0.0700) =   0.504706
y(  0.0800) =   0.474323
y(  0.0900) =   0.449142
y(  0.1000) =   0.428223
y(  0.1100) =   0.410794
y(  0.1200) =   0.396224
y(  0.1300) =   0.383996
y(  0.1400) =   0.373688
y(  0.1500) =   0.364952
y(  0.1600) =   0.357506
y(  0.1700) =   0.351117
y(  0.1800) =   0.345594
y(  0.1900) =   0.340783
y(  0.2000) =   0.336555
```



Comparison


Comparison of ODE Solution Methods
Real Function
Euler Method
Heuns Method
Midpoint Method
Runge-Kutta Method
y
x
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
0.05
0.1
0.15
0.2


Comparison of ODE Solution Methods
Real Function
Euler Method
Heuns Method
Midpoint Method
Runge-Kutta Method
y
x
0.625
0.62
0.615
0.61
0.605
0.6
0.595
0.59
0.585
0.04
0.042
0.044
0.046
0.048
0.05
0.052

Accuracy

- Accuracy of Euler Method: 96.99%
- Accuracy of Heun's Method: 99.78%
- Accuracy of Midpoint Method: 99.78%
- Accuracy of Runge-Kutta Method: 99.99%

Thank YOU