METHODS FOR SOLVING ORDINARY DIFFERENTIAL EQUATIONS

Wissam Razouki

CSE 3802

BACKGROUND

Introduction

The ODE to be solved is given as:

$$\frac{dy}{dx} = f(y,x) = xy^2e^{-x^4}$$

With initial value $y(0) = 2$ and an interval of $x \in [0, 2.5]$.

The goal of this project is to approximate the solution using various methods, and then compare the root mean square (RMS) error and the percent error using the exact solutions provided to determine the best method for solving the problem.

Methods Used

Reference method:

- The exact solution

Experimental methods:

- Euler’s method
- Runge-Kutta 4th order method (RK4)
- Modified Euler method

METHODS AND VERIFICATION

Experimental Methods

- **Euler's method**
  - *First-order method*
  - *Considered the most basic method for predicting solutions to ODEs*
  - *Approximations are generated at various values within the interval*
- **Runge-Kutta 4^{th} order method (RK4)**
  - *A 4^{th} order method that has the high-order local truncation error of the Taylor methods, but without the need to compute derivatives*
- **Modified Euler method**
  - *A predictor-corrector method that modifies the Euler method by adding a corrector term after predicting the solution*

Step Size

The exact y solutions to the ODE are given in a “Project 4 data.dat” file.

The x values have a step size of 0.05, so I decided to use this as the step size for all experimental methods, calculated from:

$$h = \frac{b - a}{N}$$

Where $a = 0$ (lower limit), $b = 2.5$ (upper limit), and $N = 50$ (number of iterations)

Method Verification

The following simple ODE was used to verify my implementation of the methods:

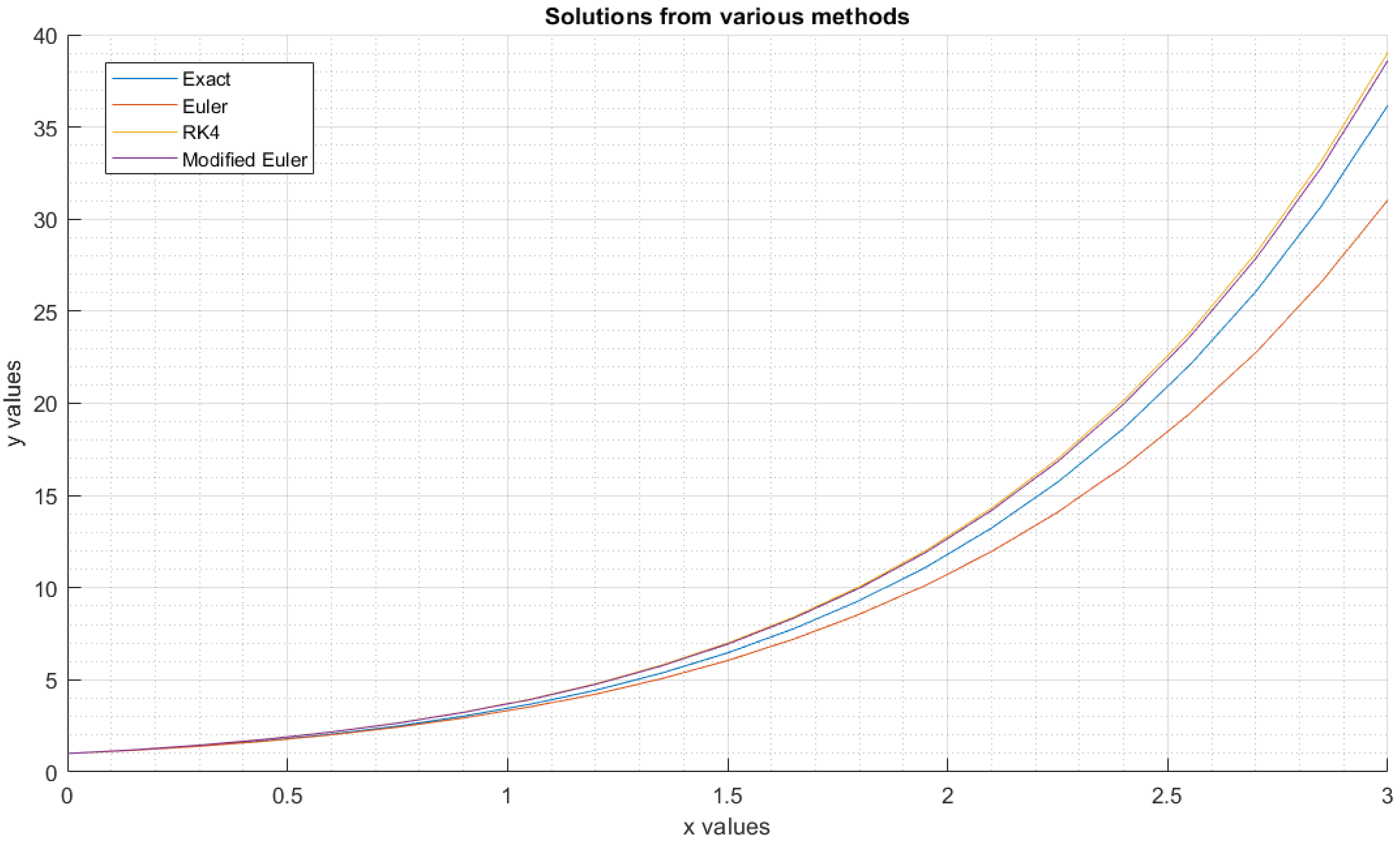
$$\frac{dy}{dx} = f(y, x) = x + y$$

With an initial value of $y(0) = 1$ and an interval of $x \in [0, 3]$.

The **exact** solution was computed using MATLAB's `dsolve(…)` function.

The resulting MATLAB plots for the solutions are given below.

(Note: RK4 and Modified Euler lie very close to each other)


Solutions from various methods
Exact
Euler
RK4
Modified Euler
y values
x values
0
5
10
15
20
25
30
35
40
0
0.5
1
1.5
2
2.5
3

RESULTS AND ANALYSIS

Solutions and Percent Error

The **exact solutions** (as provided from the “.dat” file) and **experimental solutions** are plotted below.

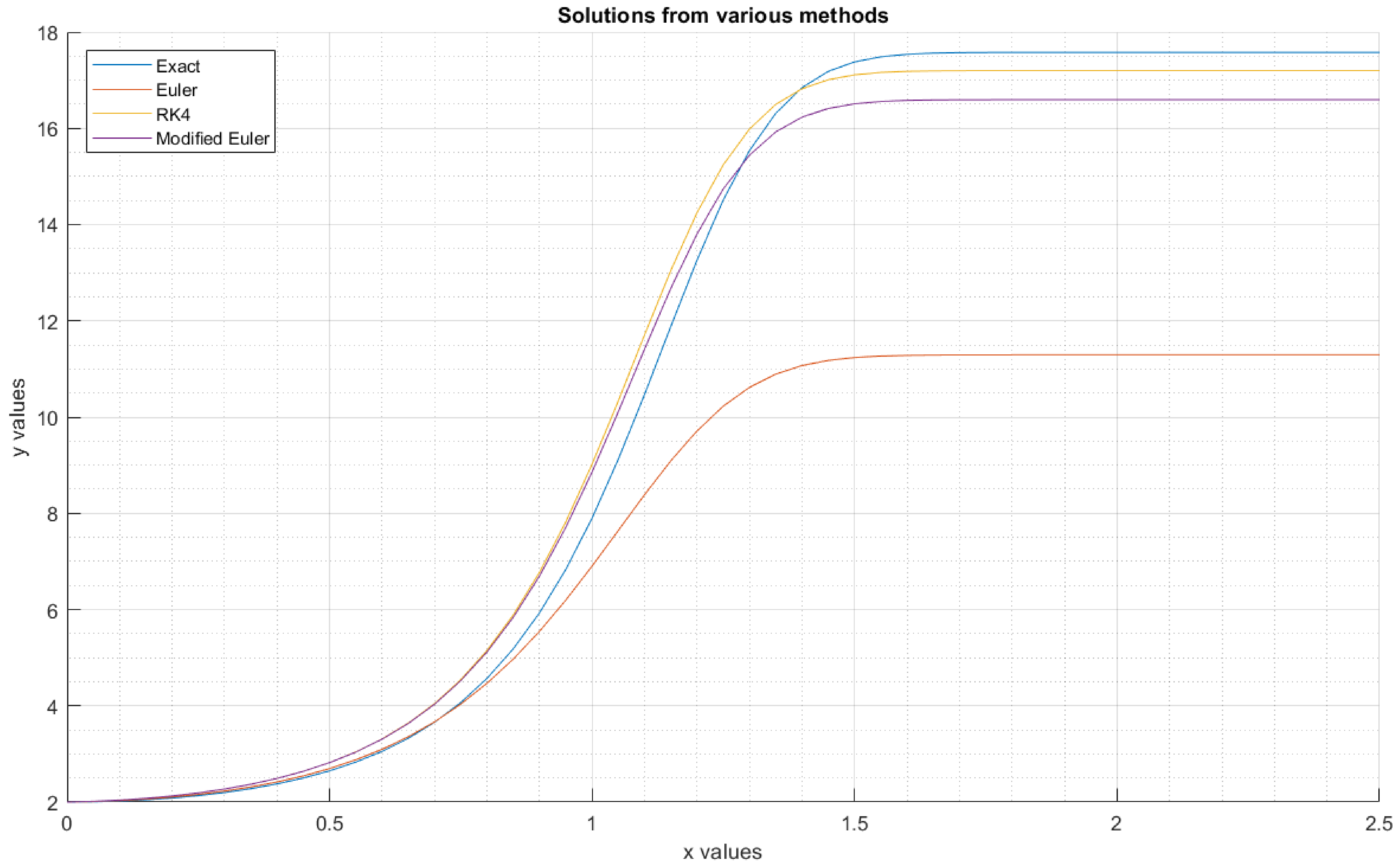
- Numerical values are provided in tabular format in a separate Excel file: **“solutions.xlsx”**

Percent error can be calculated at each x value using the formula:

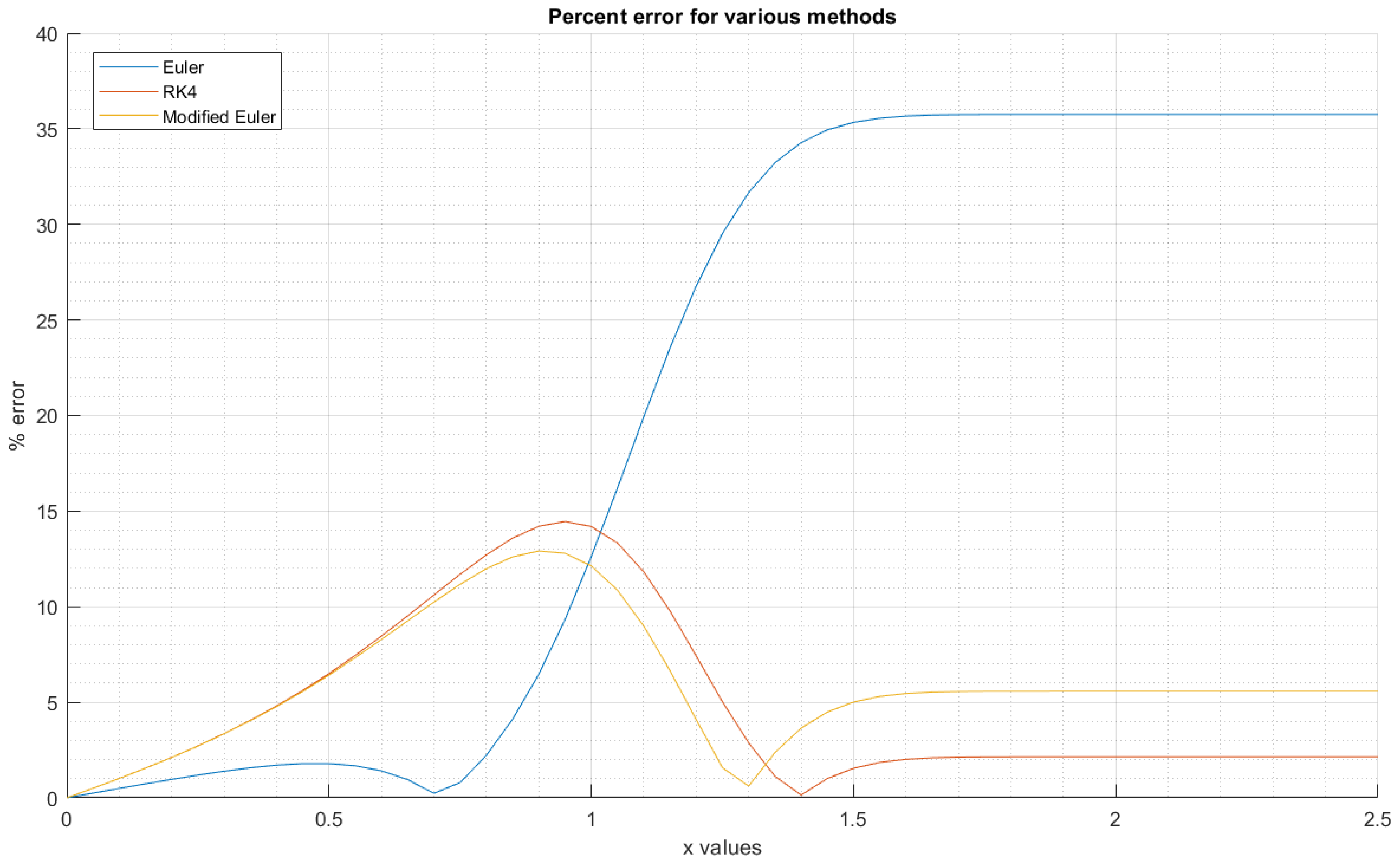
$$\frac{|E - R|}{R} \times 100\%$$

Where E is the **experimental solution**, and R is the **reference/exact solution.**

This is also plotted below.



Solutions from various methods
Exact
Euler
RK4
Modified Euler
18
16
14
12
10
8
6
4
2
0
0.5
1
1.5
2
2.5
y values
x values


Percent error for various methods
Euler
RK4
Modified Euler
% error
x values
0
5
10
15
20
25
30
35
40
0
0.5
1
1.5
2
2.5

RMS Error

RMS error can be calculated using the formula:

$$\sqrt{\frac{1}{N+1}\sum_{i=1}^{N+1}(E_i - R_i)^2}$$

Where $N = 50$ (number of iterations), E is the experimental solution, and R is the reference/exact solution.

Method	RMS error
Euler	4.4197
RK4	0.5137
Modified Euler	0.7343

Analysis and Conclusions

- From these results, we can conclude that the **Runge-Kutta 4th order method** is best suited for this particular ODE
  - *This is because it has the lowest RMS error and percent error (on average), and gives the closest approximations to the exact solutions (on average)*
  - *RK4 is a 4th order method, meaning it has lower error*
- However, we can see that the **modified Euler method** provides better results at earlier values of x
- **Euler's method** performs the worst, with a maximum error of ~35%
  - *This is a simple first-order method, thus we expect error to be higher*

None of the methods encountered fatal issues, and they all provided consistent solutions.