Innopolis University

Differential Equations
Computational practicum

by Maxim Latypov

2021

Introduction

The task of the assignment was to create the solution to first-order differential equation using numerical methods such as Euler's method, Improved Euler's method and Runge-Kutta method and to present visual graphs of these methods, local and total approximation errors.

Exact solution

In order to compare the numerical solutions and find local and total approximation errors, there is a need in a proper exact solution that can help us see the graph of the equation.

The equation is:

$$y' = xy - xy^3$$

Initial conditions are:

$$y_0 = \sqrt{\frac{1}{2}},\ x_0 = 0,\ \ x_{max} = 3$$

This equation is an example of Bernoulli equation which could be solved by changing variables:

First subtract xy from both sides and divide by -½ y^3:

$$y' - xy = -xy^3$$

$$-\frac{2y'}{y^3} + \frac{2x}{y^2} = 2x$$

Then change variables:

$$v = \frac{1}{y^2} \qquad \frac{dv}{dx} = -\frac{2y'}{y^3}$$

It gives the following equation:

$$\frac{dv}{dx} + 2vx = 2x$$

Next step is to find integrating factor:

$$\mu = e^{\int 2xdx} = e^{x^2}$$

After that substitute integrating factor in equation:

$$e^{x^2}\frac{dv}{dx} + v\left(2xe^{x^2}\right) = 2xe^{x^2}$$

Then express v :

$$\int (e^{x^2}v)\,dx = \int 2e^{x^2}xdx$$

$$e^{x^2}v = e^{x^2} + c$$

$$v(x) = ce^{-x^2} + 1$$

Solve for $y(x)$:

$$y(x) = \frac{e^{x^2/2}}{\sqrt{e^{x^2} + c}}$$

Exact solution is :

$$c = \frac{e^{x_0^2/2}}{y_0^2} - e^{x^2}$$

Implementation

Java language has been chosen using JavaFX library. The code is built using OOP and follows SOLID principles to be readable and understandable.
As far as the application is object oriented, it can be divided into classes:

·Main ·Solution ·NumericalMethod ·ExactSolution ·EulerMethod ·RungeKuttaMethod ·ImprovedEulerMethod

UML diagram attached in png file “UML-diagram.png”

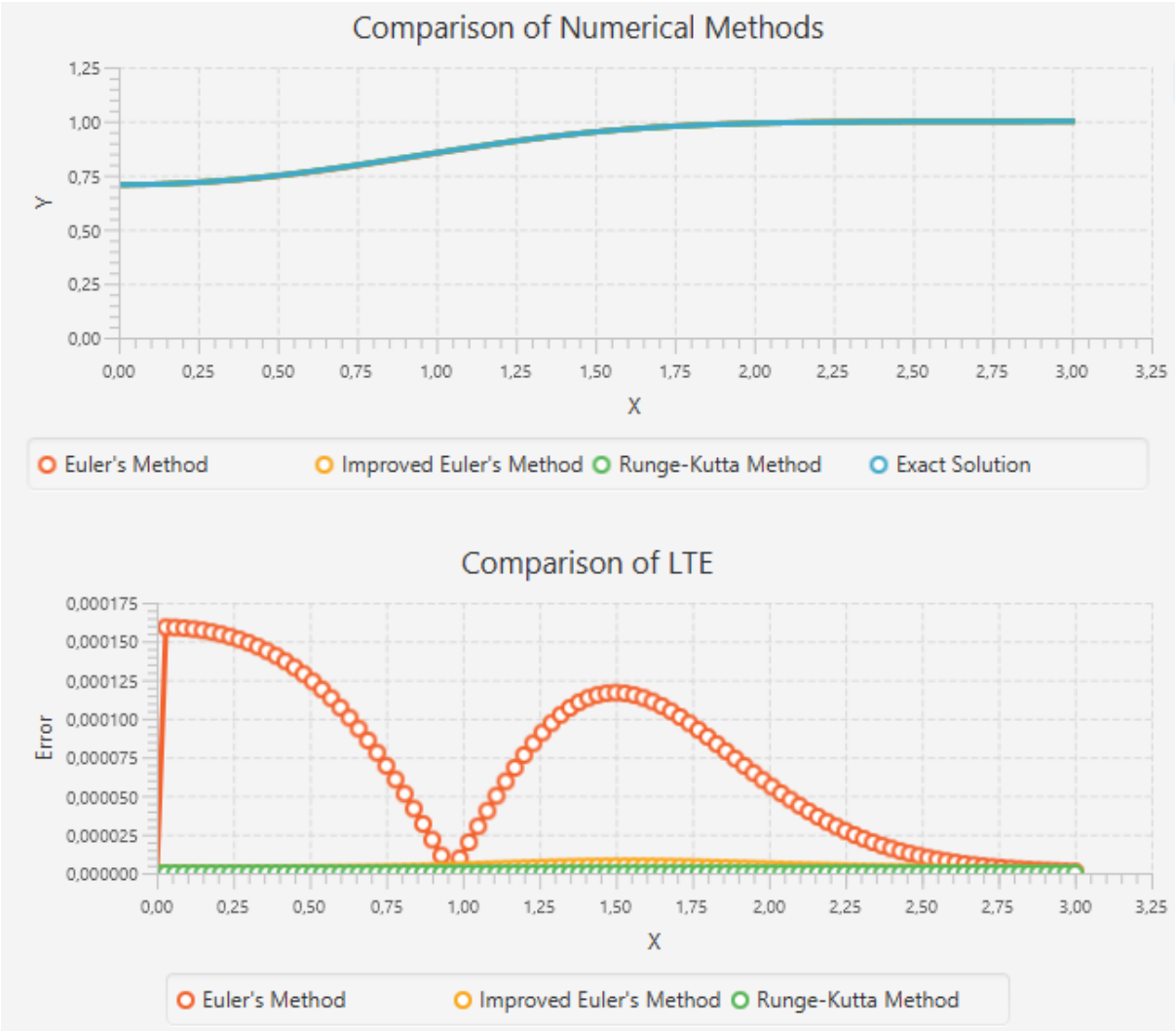
11 Variant

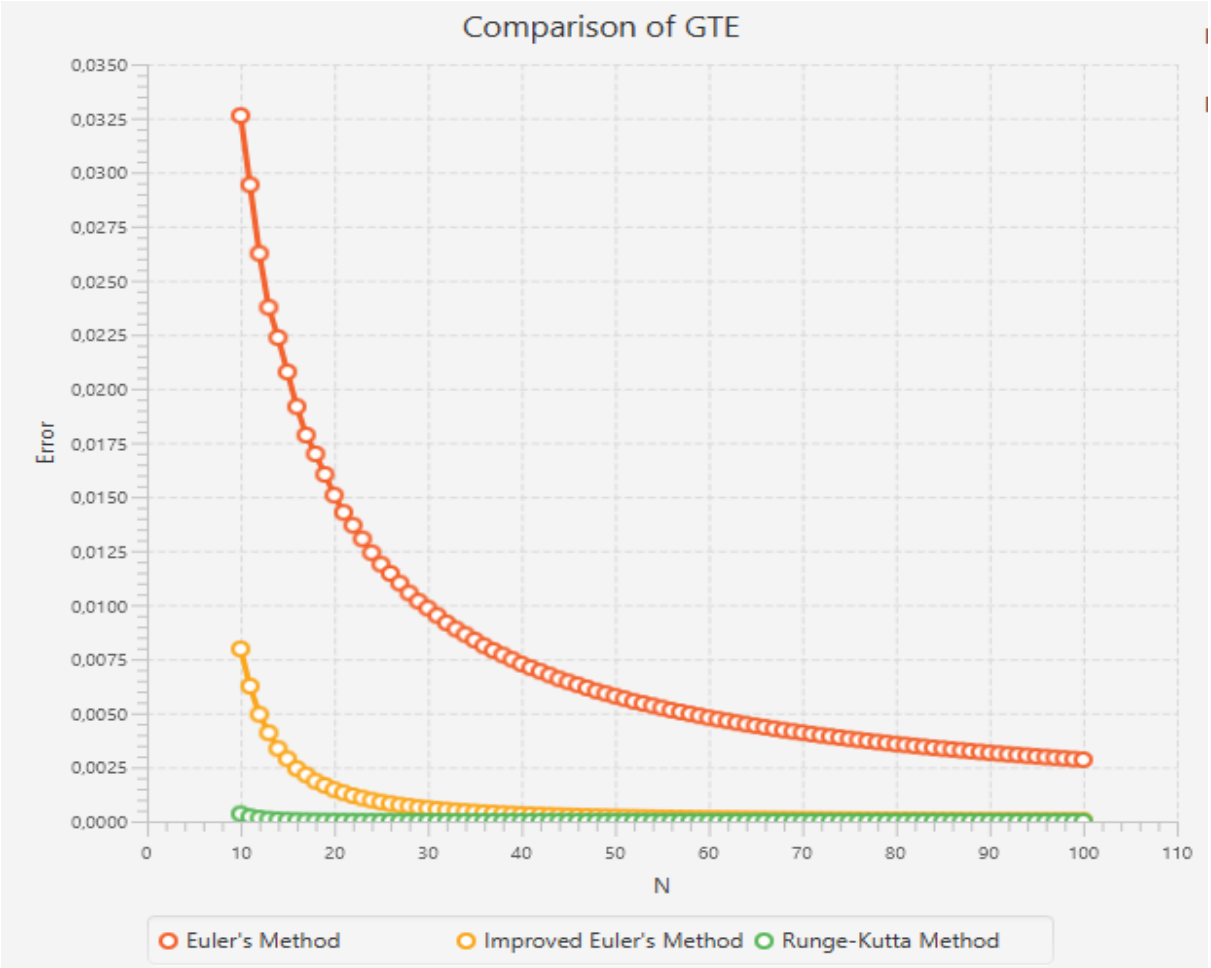
```
private double findSolution(double x) {
    return (Math.exp(Math.pow(x, 2) / 2)) / Math.sqrt(this.C + Math.exp(Math.pow(x, 2)));
}
```

```
public ExactSolution(double x0, double y0, double X, int N) {
    super(x0, y0, X, N);
    this.C = (Math.exp(Math.pow(x0, 2)) / Math.pow(y0, 2)) - Math.exp(Math.pow(x0, 2));
    generate();
}
```

Results

Here are obtained graphs:





Conclusion

As it was shown, numerical solutions are easy to understand and very helpful in case you need numbers to operate with your task. Runge-Kutta method gives the best accuracy compared with 2 other methods, so it is the most preferable method in our case.