DIFFERENTIAL EQUATION

COMPUTATIONAL PRACTICUM

Done by Bekzhan Talgat (BS18-05)

Innopolis University 2019

Problem statement:

$dy/dx = (2 - y^2) / (2yx^2)$ $y(1) = 1;$ $x \in [1, 6]$

Exact solution of IVP:

$dy/dx = (2 - y^2) / (2yx^2)$ $x \neq 0$ $y \neq 0$

1. $(- (y^2 - 2)\, dy) / (2y) = dx / x^2$
2. $- \ln |y^2 - 2| = c - 1/x$
3. $c + 1/x = \ln |y^2 - 2|$
4. $e^{c + 1/x} = y^2 - 2$
5. $y = \sqrt{(2 + e^{c + 1/x})}$

Here we can find that: $c = \ln (y^2 - 2) - 1/x$
To solve IVP and find c : $y^2 > 2$
IVP is not solvable for y=1

But it is possible to solve this problem with different methods such as Euler's method, Improved Euler's method, and Runge-Kutta method

Implementation

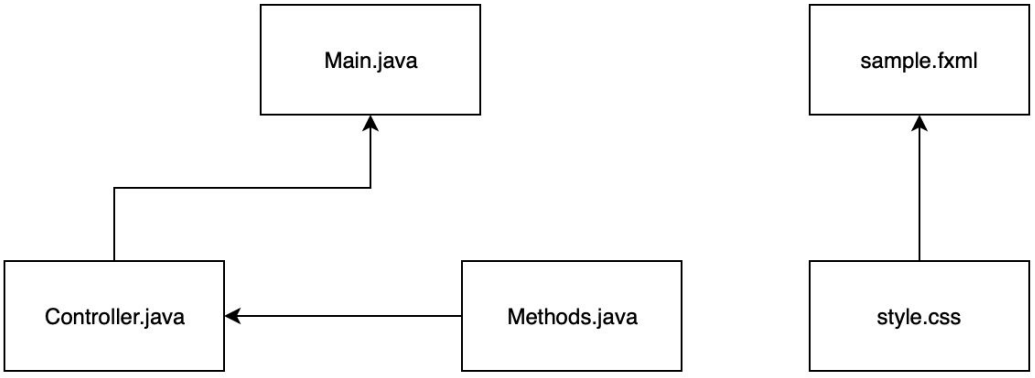
For implementation, I used JavaFX programming language with Scene Builder for making GUI elements

There are 5 code files: Main.java, Controller.java, Methods.java, sample.fxml, style.css
Main.java initializes application

Description of class and interesting moments of the code

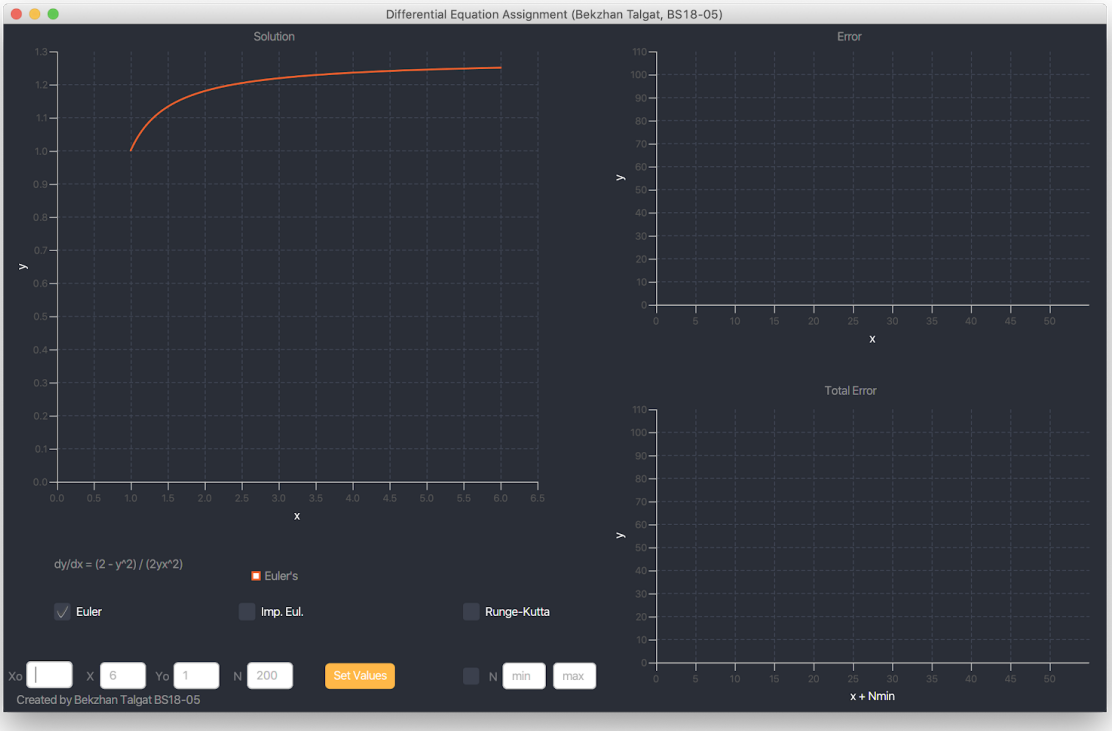
Controller.java has the main logic of the application, defines all functionalities of every entity in application
Methods.java contains computational methods for differential equation and has different methods: Exact solution, Euler's method, Improved Euler's method, Runge-Kutta method

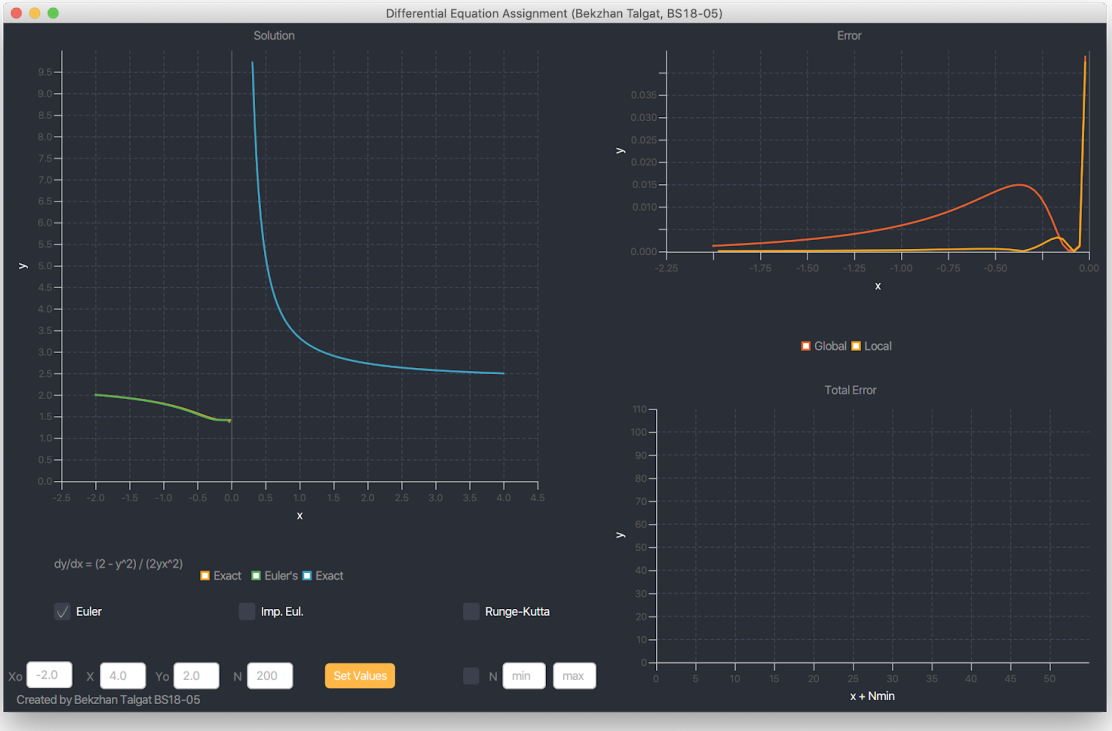
UML of code files:

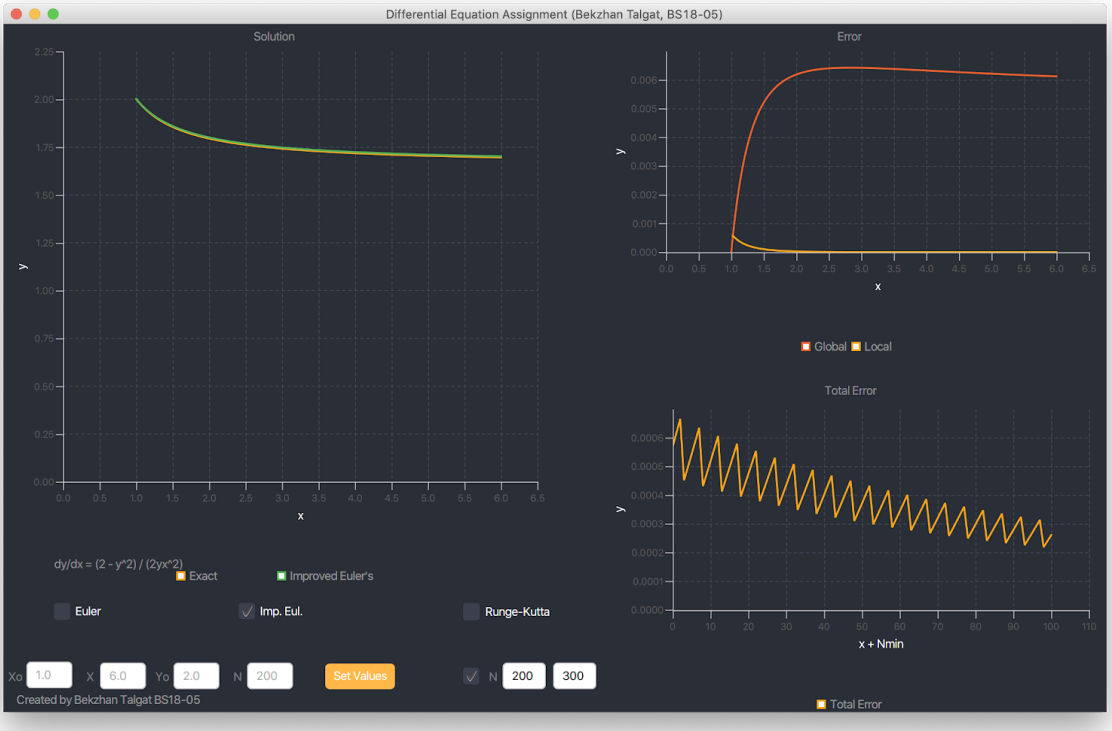


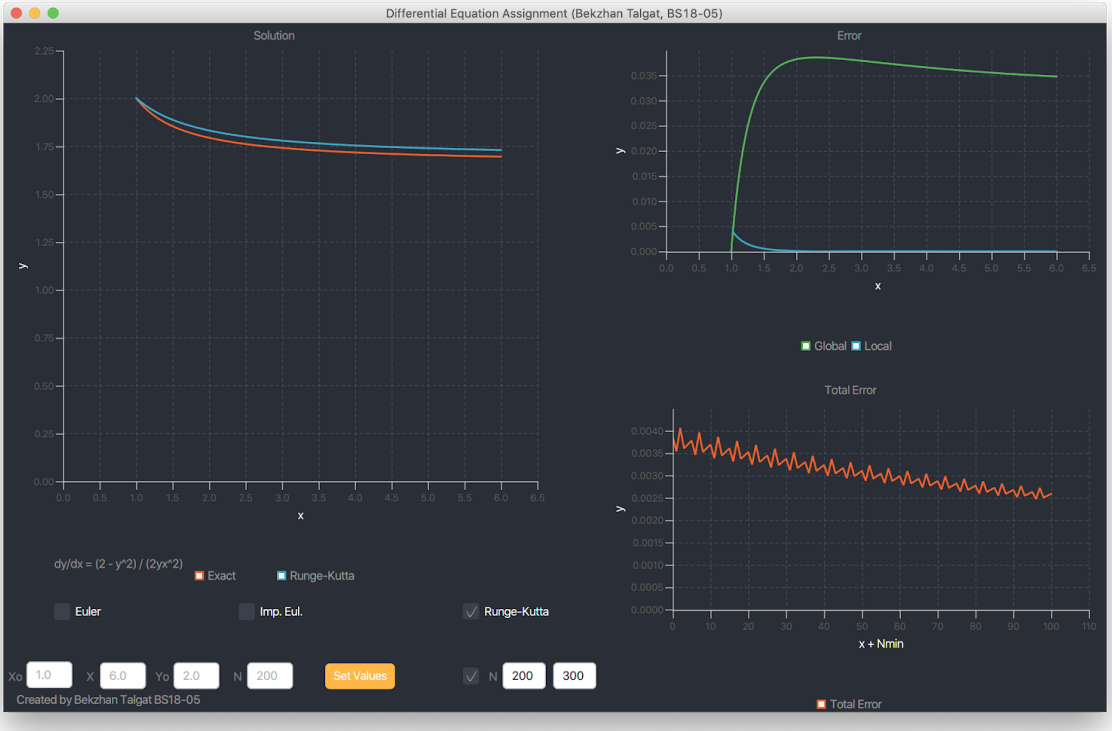
Link to the source code:
https://github.com/Beka-13/DifferentialEquation/tree/master/DifferentialEq

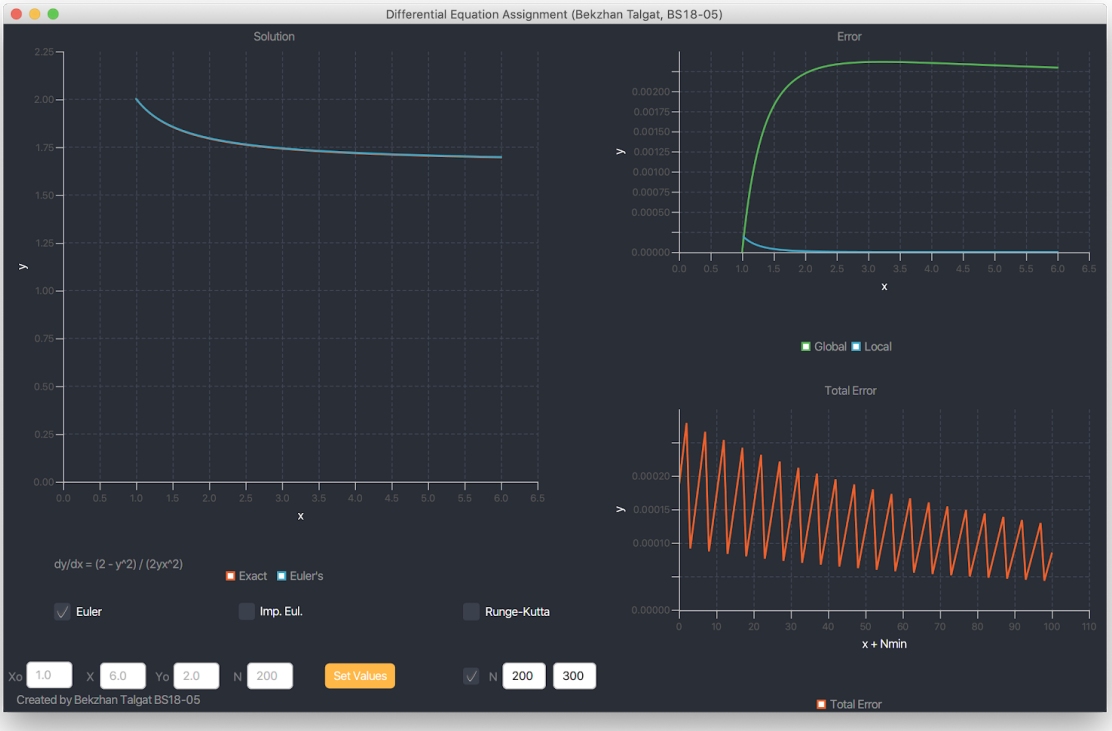
Here some screenshots of the application
On the left bottom part is control part, where you can choose which equation to illustrate and text fields to change values. Only one numerical method with exact solution is able to illustrate on the screen. Also checkbox for Nmin and Nmax text fields


Differential Equation Assignment (Bekzhan Talgat, BS18-05)
Solution
Error
Total Error
dy/dx = (2 - y^2) / (2yx^2)
Euler's
Euler
Imp. Eul.
Runge-Kutta
Xo
X
6
Yo
1
N
200
Set Values
N
min
max
Created by Bekzhan Talgat BS18-05
x + Nmin


Differential Equation Assignment (Bekzhan Talgat, BS18-05)
Solution
Error
Total Error
dy/dx = (2 - y^2) / (2yx^2)
Exact
Euler's
Exact
Global
Local
Euler
Imp. Eul.
Runge-Kutta
Xo
-2.0
X
4.0
Yo
2.0
N
200
Set Values
N
min
max
Created by Bekzhan Talgat BS18-05
x + Nmin


Differential Equation Assignment (Bekzhan Talgat, BS18-05)
Solution
Error
Total Error
dy/dx = (2 - y^2) / (2yx^2)
Exact
Improved Euler's
Euler
Imp. Eul.
Runge-Kutta
Global
Local
Total Error
Xo 1.0 X 6.0 Yo 2.0 N 200 Set Values N 200 300
Created by Bekzhan Talgat BS18-05
x
y
x + Nmin


Differential Equation Assignment (Bekzhan Talgat, BS18-05)
Solution
Error
Total Error
dy/dx = (2 - y^2) / (2yx^2)
Exact
Runge-Kutta
Euler
Imp. Eul.
Runge-Kutta
Global
Local
Total Error
Xo 1.0 X 6.0 Yo 2.0 N 200 Set Values N 200 300
Created by Bekzhan Talgat BS18-05
x
y
x + Nmin



Here are functions that compute differential equation with different methods such as: Euler's method, Improved Euler's method, Runge-Kutta method

```
double dydx(double x, double y) {
    if ( x == 0 || y == 0) { return CONST; }
    else { return ((2 - y*y) / (2*x*x*y)); }
}

double EulerEq(double x, double h, double y) {
    if (x== 0 || y == 0) { return CONST; }
    else { return (y + h*dydx(x,y)); }
}

double ImEulerEq(double x, double h, double y) {
    double k1 = dydx(x, y);
    double k2 = dydx((x+h), (y+(h*k1)));

    if (k1 == CONST || k2 == CONST){ return CONST; }
    else { return (y + (h/2)*(k1+k2)); }
}

double RungeKuttaEq(double x, double h, double y) {
    double k1 = dydx(x,y);
    double k2 = dydx((x + h/2),(y + k1/2));
    double k3 = dydx((x + h/2),(y + k2/2));
    double k4 = dydx((x + h),(y + k3));

    if (k1 == CONST || k2 == CONST || k3 == CONST || k4 == CONST) { return CONST; }
    else { return (y + (h/6) * (k1 + 2*k2 + 2*k3 + k4)); }
}
```