Differential Equations
Utkarsh Kalra
BS18-03

COMPUTATIONAL PRACTICUM REPORT

LINK TO GITHUB:https://github.com/kalraUtkarsh/Differential_equations/tree/master/computational%20practicum

Analytical Solution

$$\frac{d\,y(x)}{dx} = \frac{4}{x^2} - y(x)^2 - \frac{y(x)}{x}$$

$$y(x) = -v(x)$$

$$-\frac{d\,v(x)}{dx} = -v(x)^2 + \frac{v(x)}{x} + \frac{4}{x^2} \qquad v(x) = -\frac{\frac{du(x)}{dx}}{u(x)}$$

$$-\frac{\frac{d^2u(x)}{dx^2}}{u(x)} + \frac{\left(\frac{du(x)}{dx}\right)^2}{u(x)^2} = \frac{\left(\frac{du(x)}{dx}\right)^2}{u(x)^2} + \frac{\frac{du(x)}{dx}}{x\,u(x)} - \frac{4}{x^2}$$

$$x^2\,\frac{d^2u(x)}{dx^2} + x\,\frac{du(x)}{dx} - 4\,u(x) = 0$$

$$\lambda^2 - 4 = 0$$

$$u(x) = u_1(x) + u_2(x) = \frac{c_1}{x^2} + c_2\,x^2 \qquad v(x) = \frac{2\,c_1 - 2\,c_2\,x^4}{c_1\,x + c_2\,x^5}$$

$$\frac{c_2\left(-2\,x^4 + \frac{2\,c_1}{c_2}\right)}{c_2\left(x^5 + \frac{c_1\,x}{c_2}\right)} \qquad v(x) = \frac{2}{x} - \frac{4\,x^3}{x^4 + c_1} \qquad y(x) = -\frac{2}{x} + \frac{4\,x^3}{x^4 + c_1}$$

IVP

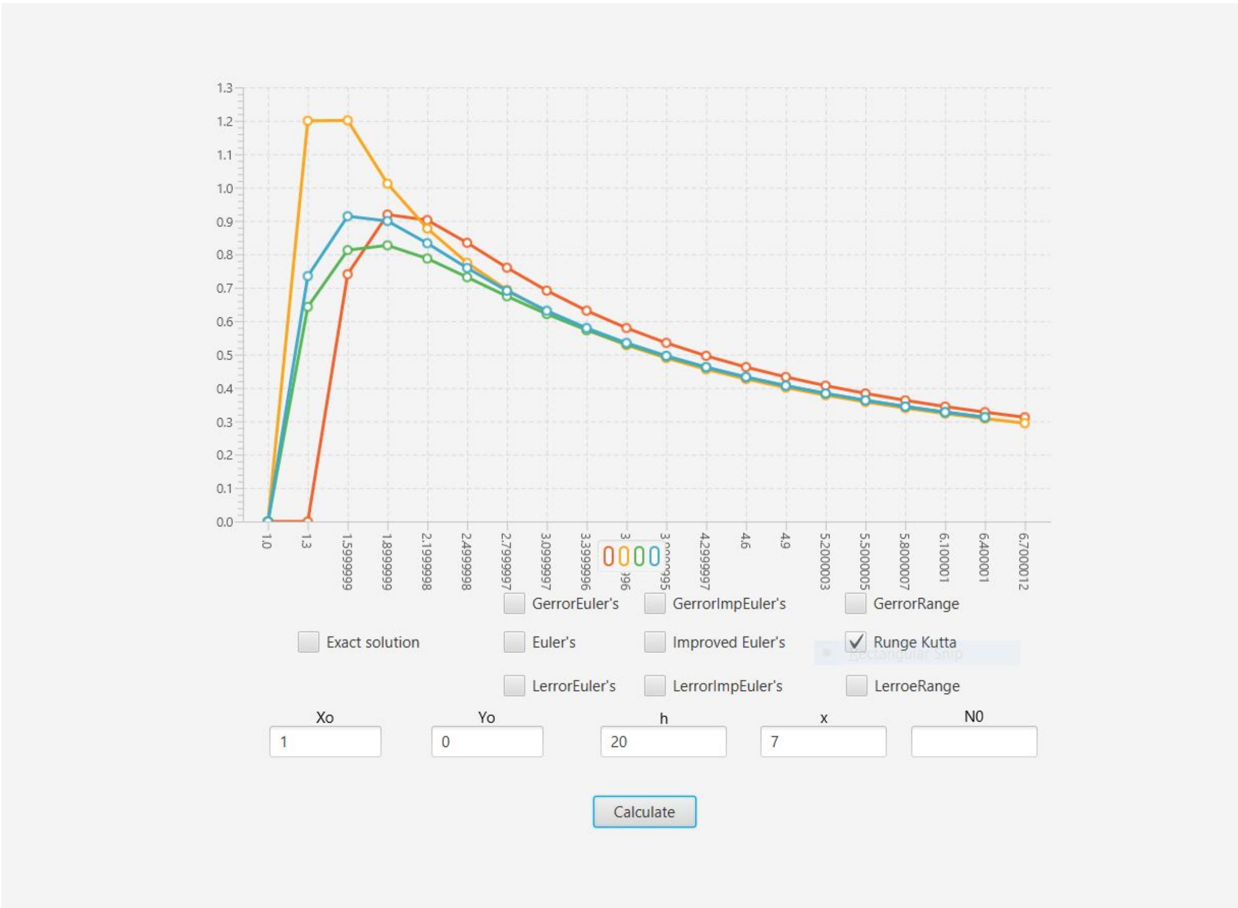
c = (2*x0^ 4 - y0 *x0^ 5) / (2 + y0*x0);

c = 1.

screenshots

Exact solution- red
Euler’s method- orange
Improved Euler’s method - green

Runge kutta method- blue

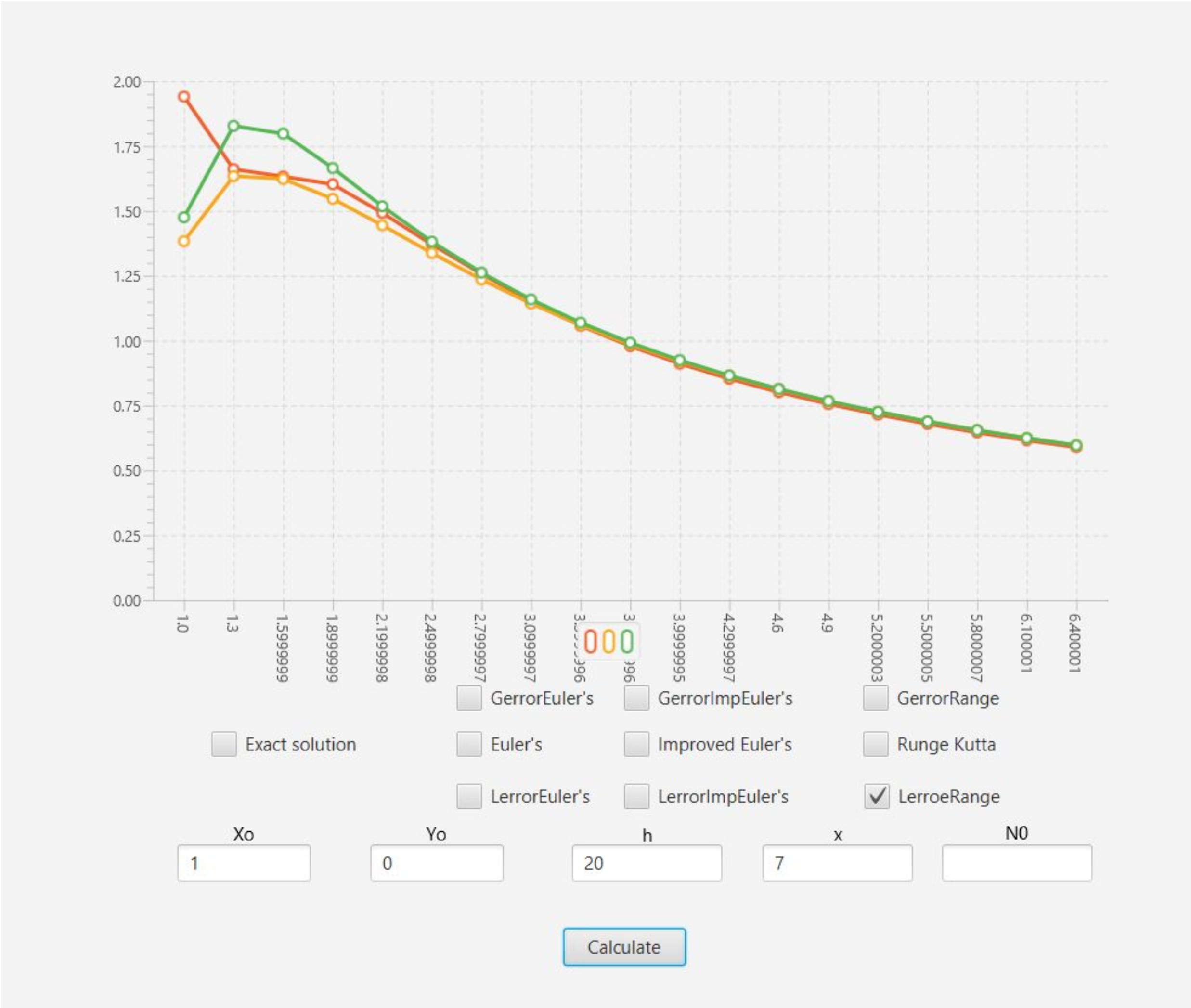


Local errors

Euler- red

Impeuler euler- orange

Runge kutta - green



Global errors
Euler- red
Improved Euler’s - orange
Runge-kutta-green

RESULTS

The first image shows 4 graphs solving the differential equations using four different methods. The second image shows local errors, Local errors is an error which calculated on each step separately, regardless of the previous step's error, For finding local errors first I find two global errors at the point at which I want to find a local error and at the points in front of it, then I subtract from first point second one. From the first and second screenshot, we see that the

Euler method is the worst one, Runge-Kutta is the best and Improved Euler is in the middle. In the last chart, we can see the dependency on the quality of the approximations on the number of points. We see that the more points the more accurately the methods work. To build a graph, I solved equations with a different number of points (from N0 to N) and calculated the global error at the last point.

ANALYSIS OF THE CODE.

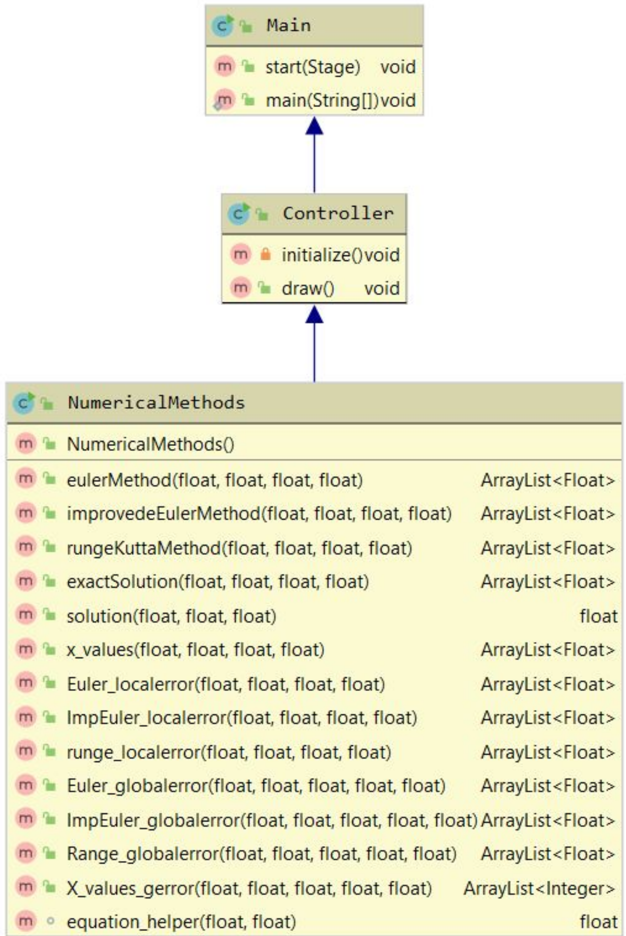
The whole project was made with JAVAFX, it has 3 classes, Tha Main class, The Controller, The numerical methods class and there is a FXML file which contains the layout of the application. The principal of OOP are used

OOP

The Numerical method class has most of the methods being used, Eulermethod, Improved Euler method, Runge-Kutta method, Localerror methods for all three methods, Global error Methods for all three methods, a solution method which has the exact solution, an Exact method, x_values method, and x_values for global error methods.

An object of the Numerical methods class is created in the controller class and the methods are called using this object.

UML


Main
start(Stage) void
main(String[]) void
Controller
initialize() void
draw() void
NumericalMethods
NumericalMethods()
eulerMethod(float, float, float, float) ArrayList<Float>
improvedeEulerMethod(float, float, float, float) ArrayList<Float>
rungeKuttaMethod(float, float, float, float) ArrayList<Float>
exactSolution(float, float, float, float) ArrayList<Float>
solution(float, float, float) float
x_values(float, float, float, float) ArrayList<Float>
Euler_localerror(float, float, float, float) ArrayList<Float>
ImpEuler_localerror(float, float, float, float) ArrayList<Float>
runge_localerror(float, float, float, float) ArrayList<Float>
Euler_globalerror(float, float, float, float, float) ArrayList<Float>
ImpEuler_globalerror(float, float, float, float, float) ArrayList<Float>
Range_globalerror(float, float, float, float, float) ArrayList<Float>
X_values_gerror(float, float, float, float, float) ArrayList<Integer>
equation_helper(float, float) float