Differential Equation Computational Practicum Report

Farkhod Khakimiyon, Group 3

Fall 2019, Innopolis University

Github link

1 Variant: 11th problem

Problem 1. $y' = xy - xy^3$, $x_0 = 0$, $y_0 = \sqrt{1/2}$, $X = 3$

Solution 1. *The equation is* 1^{st} *ODE in the form of Bernoulli equation: Given function*

$$f(x, y) = y' = xy - xy^3$$

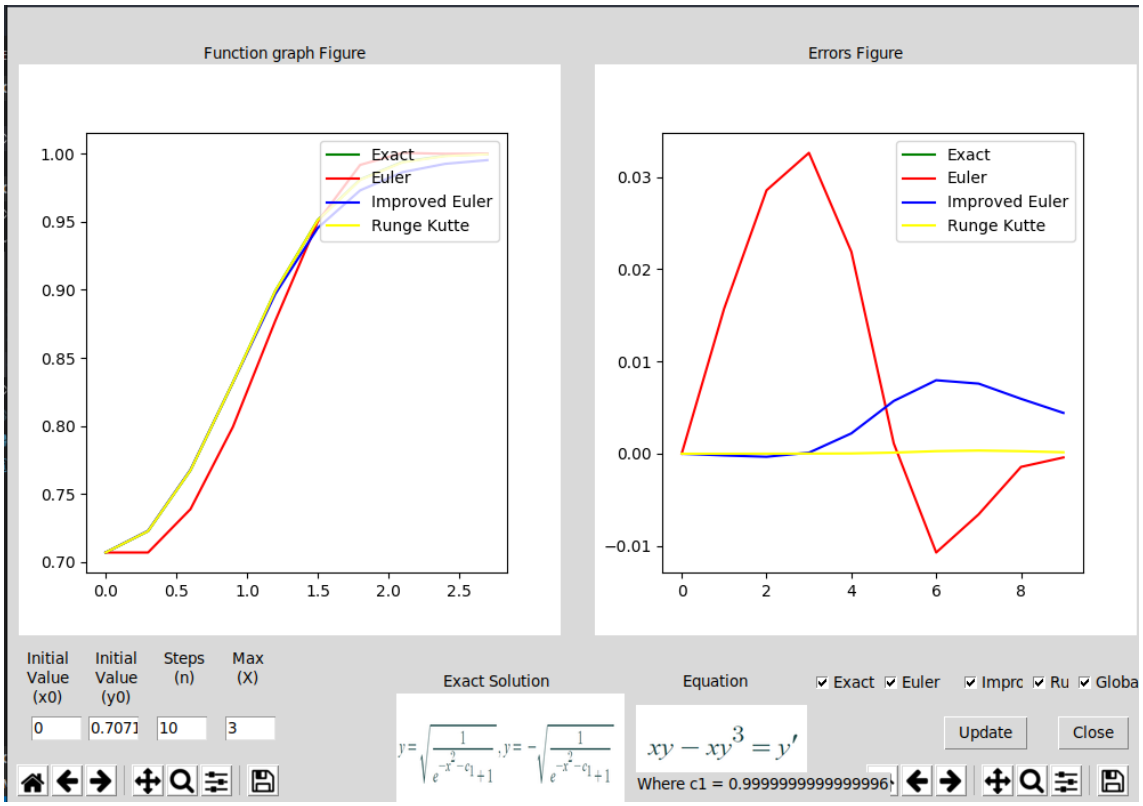
with IVP: $y(x_0) = y_0$
Let's transform equation to the first order Bernoulli ODE: $y' - xy = -xy^3$ *and we have* $p(x) = -x$, $q(x) = -x$, $n = 3$. *Let's substitute* $v = y^{1-n} = y^{-2}$ *and obtain* $-\frac{v'}{2} - xv = -x$. *Solving complementary we obtain* $v(x) = k * e^{-x^2}$ *and by method of variation of parameter we get the final solution for* $v(x) = c_1 e^{-x^2} + 1$
Than, substituting back $v = y^{-2}$ *we get:* $y^{-2} = c_1 \cdot e^{-x^2} + 1$, *which is*

$$y = \sqrt{\frac{1}{c_1 \cdot e^{-x^2} + 1}}, y = -\sqrt{\frac{1}{c_1 \cdot e^{-x^2} + 1}}$$

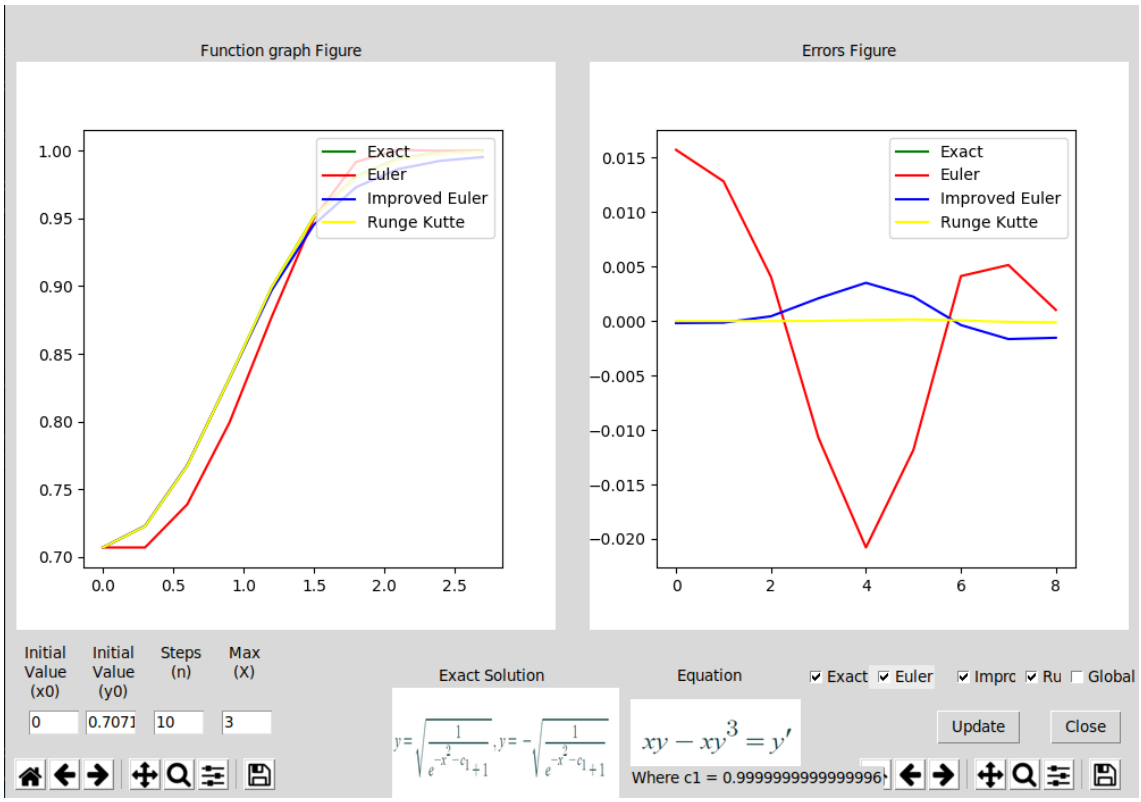
With given IVP: $y(0) = \sqrt{\frac{1}{2}}$, *we can derive formula for* $c_1 = \frac{(1-y_0^2) \cdot e^{x_0^2}}{y_0^2}$ *and compute* c_1 *for given particular case* $c_1 = 1$

2 Graphs from the program

Graph 1. *The following graph show the full functionality GUI of the program with global error.*



Graph 2. *The following graph show the full functionality GUI of the program with local error.*



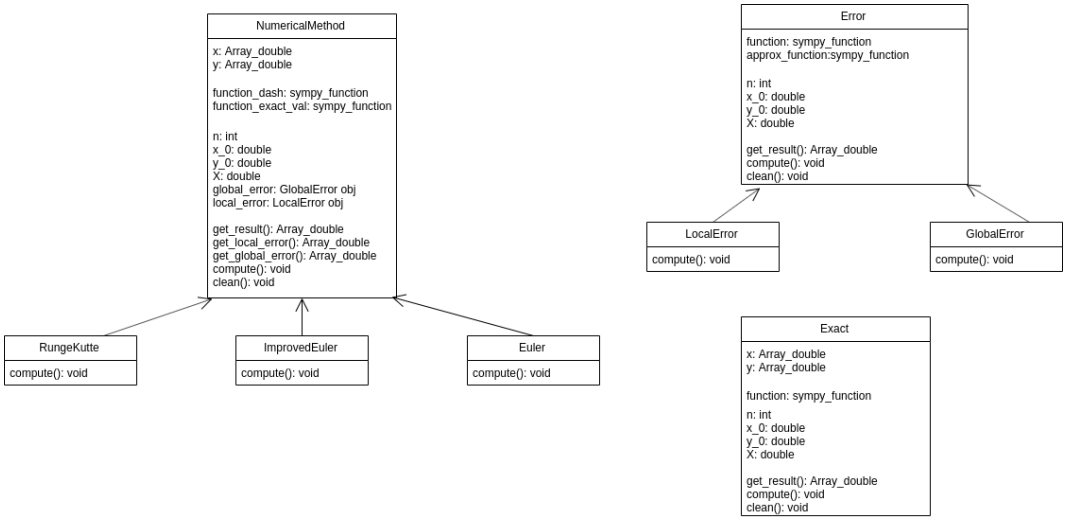
3 Details about the program

MVC design pattern has been used in order to have Model.py (the model) which will interact with the app.py the user and combine the View and the Controller. While the GUI has been fully implemented in View.py (the view) and the controller and the real implementations of the methods exist in Controller.py (the controller).

So, there exist 4 files: Model.py, Controller.py, View.py and app.py.

4 UML diagram for the software

Graph 3. *The following graph shows the UML of the Controller.*


NumericalMethod
x: Array_double
y: Array_double
function_dash: sympy_function
function_exact_val: sympy_function
n: int
x_0: double
y_0: double
X: double
global_error: GlobalError obj
local_error: LocalError obj
get_result(): Array_double
get_local_error(): Array_double
get_global_error(): Array_double
compute(): void
clean(): void
RungeKutte
compute(): void
ImprovedEuler
compute(): void
Euler
compute(): void
Error
function: sympy_function
approx_function:sympy_function
n: int
x_0: double
y_0: double
X: double
get_result(): Array_double
compute(): void
clean(): void
LocalError
compute(): void
GlobalError
compute(): void
Exact
x: Array_double
y: Array_double
function: sympy_function
n: int
x_0: double
y_0: double
X: double
get_result(): Array_double
compute(): void
clean(): void