

Computational Practicum Report

Roman Voronov r.voronov@innopolis.university

October 27, 2021

1 Exact Solution of Differential Equation

Given Initial Value Problem (IVP):

$$\begin{cases} y' = f(x, y) \\ y(x_0) = y_0 \\ x \in (x_0; X) \end{cases}$$

In my case I obtain the following IVP:

$$\begin{cases} y' = \frac{y^2 - y}{x} \\ y(1) = \frac{1}{2} \\ x \in (1; 9) \end{cases}$$

1.1 Find General Solution

1.1.1 $y' = \frac{y^2 - y}{x}$ - non-linear first order differential equation.

This equation is equivalent to differential equation in separable form.

Since there is a division by $x \Rightarrow x \neq 0$

1.1.2 Let's consider border cases: 1) $y = 0$, 2) $y = 1$ on $\mathbb{R}/\{0\}$

1) $y = 0 \Rightarrow y' = 0$

Check: $0 = \frac{0}{x} \Rightarrow 0 = 0$; Therefore $y = 0$ is a trivial solution on $\mathbb{R}/\{0\}$

2) $y = 1 \Rightarrow y' = 0$

Check: $0 = \frac{1-1}{x} \Rightarrow 0 = 0$; Therefore $y = 1$ is a trivial solution on $\mathbb{R}/\{0\}$

1.1.3 Assume $y \neq 0$ and $y \neq 1$:

$$\frac{y'}{y^2 - y} = \frac{1}{x}$$

Let's bring differential equation to differential form:

$$\frac{dy}{y^2 - y} = \frac{dx}{x}$$

Let's integrate both parts:

$$\int \frac{dy}{y^2 - y} = \int \frac{dx}{x}$$

$$\int \left(\frac{1}{y-1} - \frac{1}{y} \right) dy = \ln|x| + \ln|C|, \quad C - \text{const} \in R/\{0\}$$

$$\ln|y-1| + \ln|y^{-1}| = \ln|xC|$$

$$(y-1)y^{-1} = xC$$

Let's represent equation in explicit form:

$$y = \frac{1}{1-Cx}, \quad x \neq \frac{1}{C}$$

1.2 Let's move on to the solution of initial value problem:

$$\begin{cases} y = \frac{1}{1-Cx} \\ y(1) = \frac{1}{2} \end{cases} \Rightarrow \frac{1}{2} = \frac{1}{1-C} \Rightarrow C = -1, \quad x \neq -1$$

1.3 Answer: $y = \frac{1}{x+1}$ on $R/\{-1, 0\}$

2 Analysis of points of discontinuity

The equation contains two points of discontinuity:

1) $x = -1$:

$$L^+ = \lim_{x \rightarrow -1^+} \frac{1}{x+1} = \infty$$

$$L^- = \lim_{x \rightarrow -1^-} \frac{1}{x+1} = -\infty$$

Both L^- and L^+ don't exist so $x = -1$ is point of infinite discontinuities

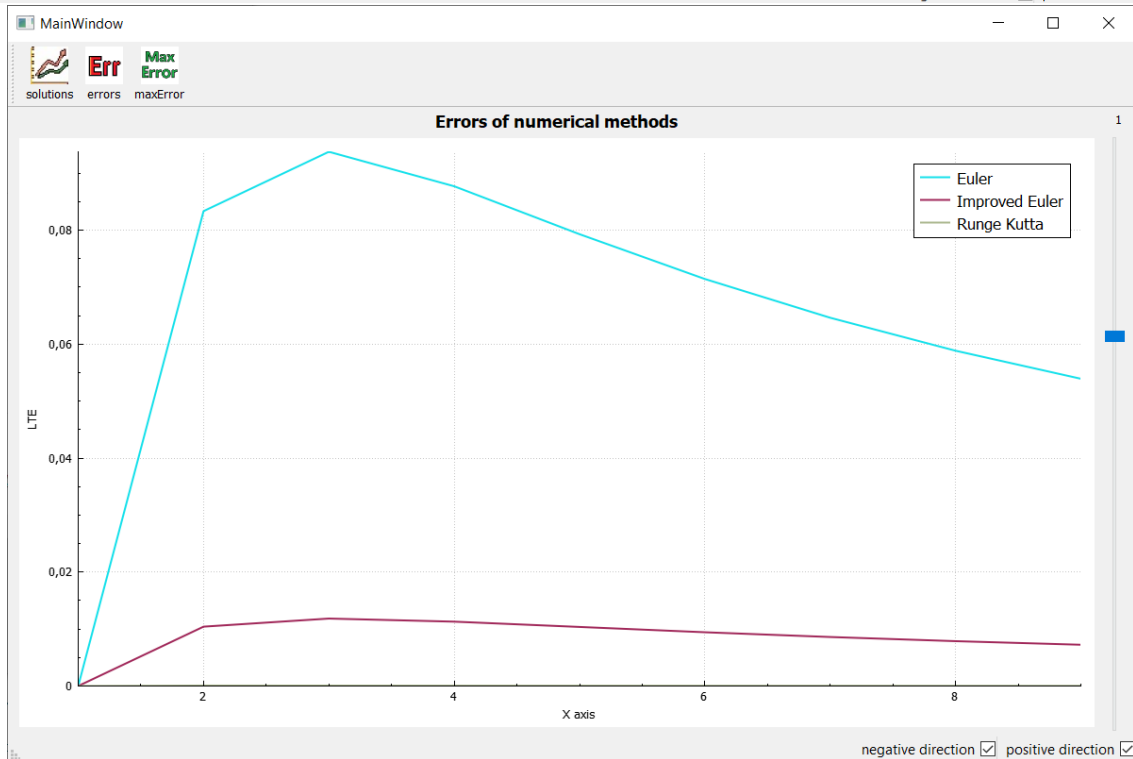
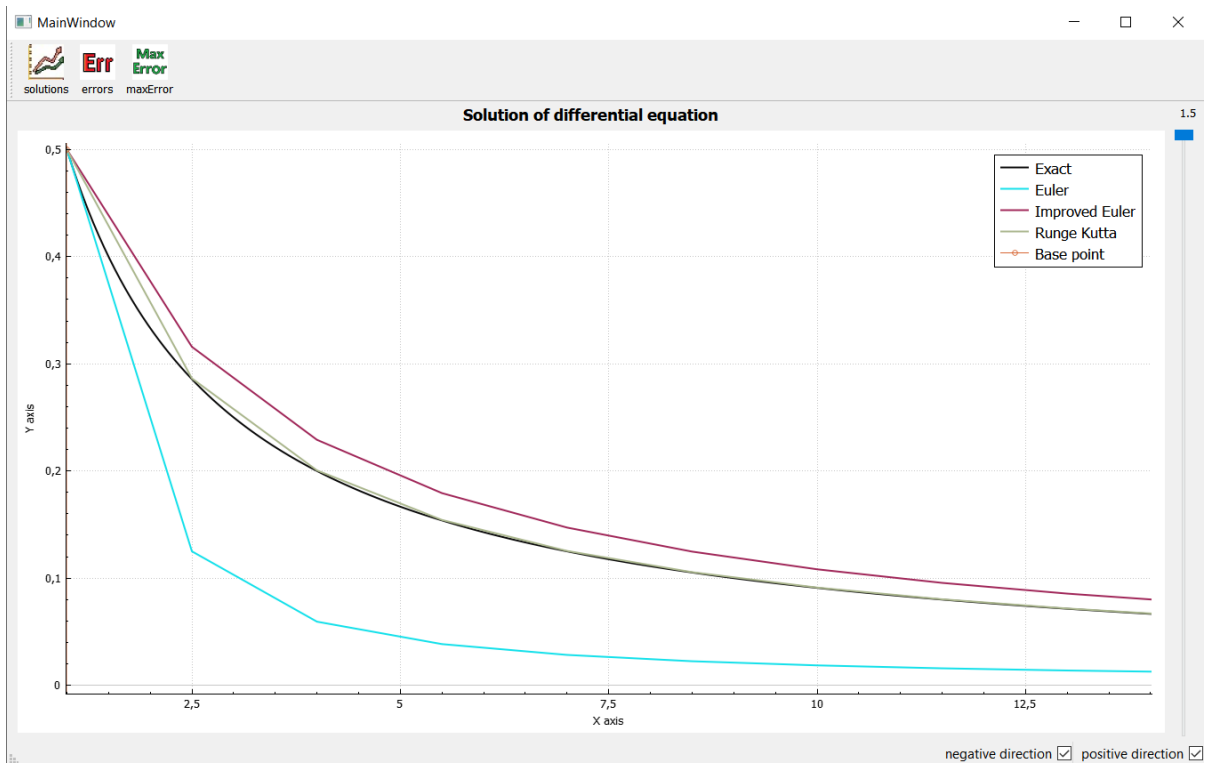
2) $x = 0$:

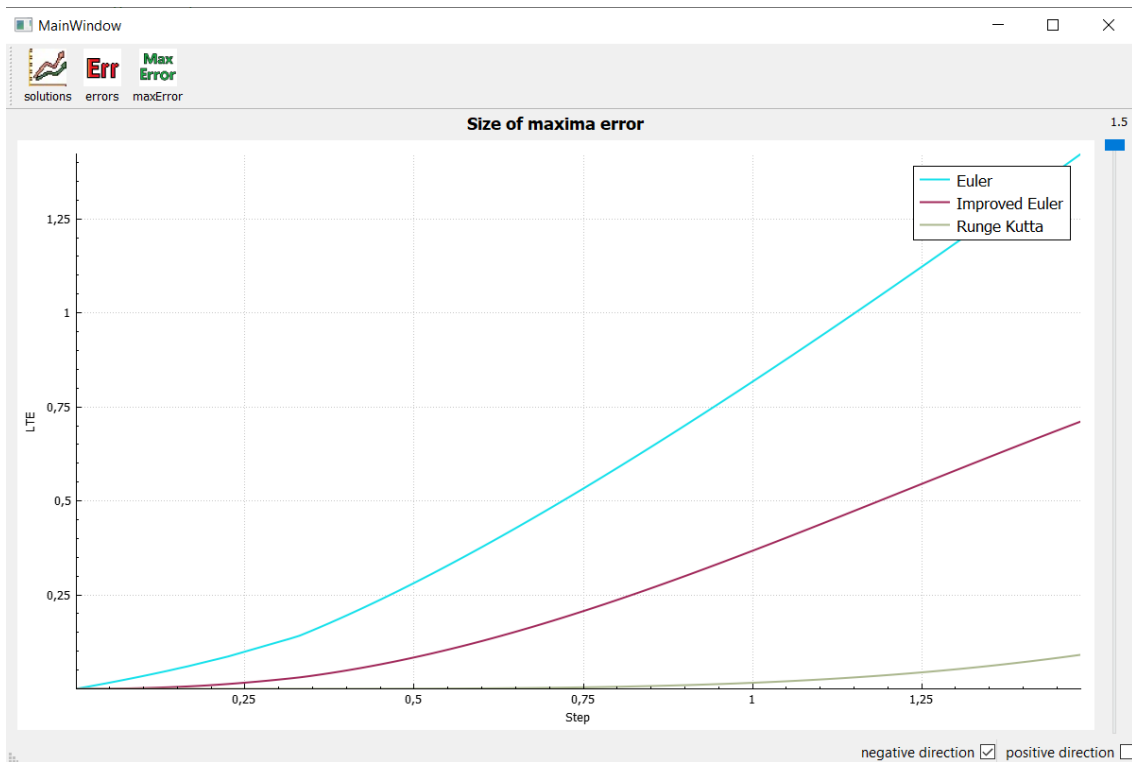
$$L^+ = \lim_{x \rightarrow 0^+} \frac{1}{x+1} = 1$$

$$L^- = \lim_{x \rightarrow -1^-} \frac{1}{x+1} = 1$$

Both L^- and L^+ exist and equal so $x = 0$ is point of removable discontinuities

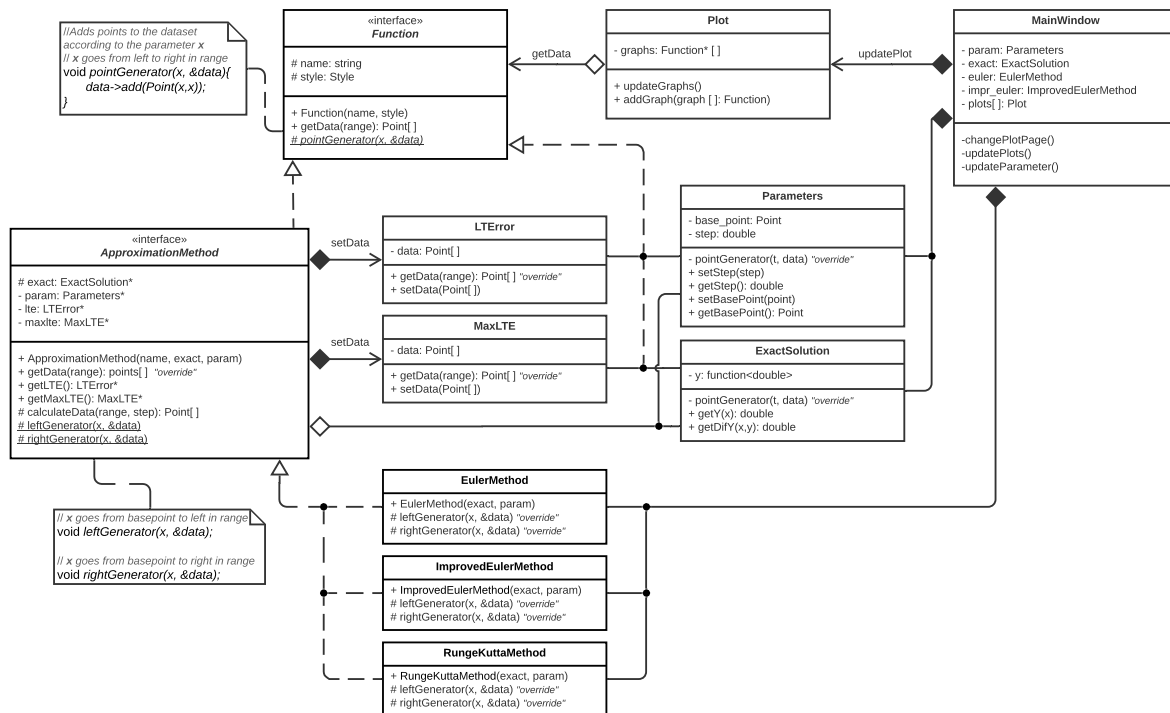
3 Computed Solution of Initial Value Problem (Graphs)



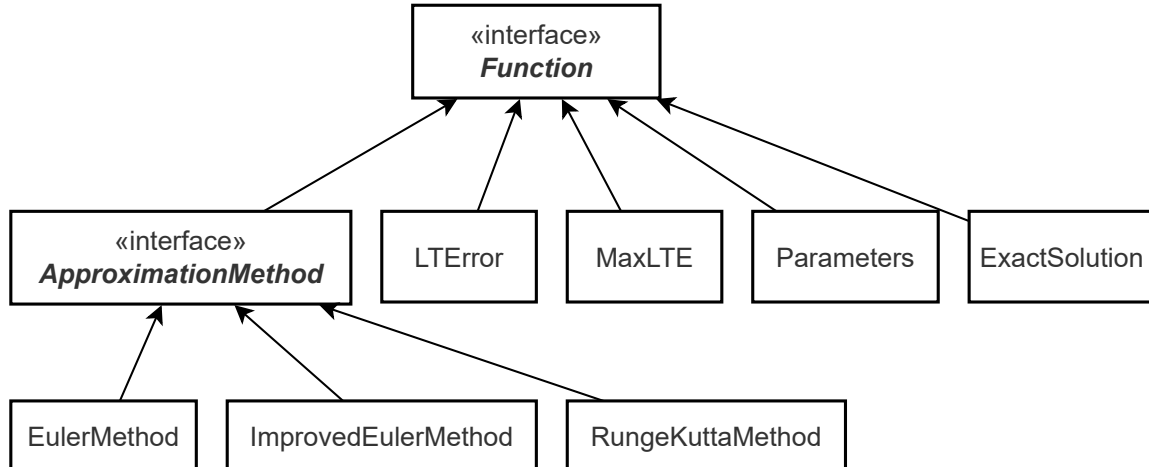


4 Code Comments

4.1 UML class diagram



4.2 Class Hierarchy



4.3 Used libraries

Qt libraries were used to display the window. In addition, the QCustomPlot library was used to build the graphs.

4.4 Class explanation

Interface layer:

- **MainWindow**: main class, it configures the relationships of objects and processes user actions.

Drawer layer:

- **Plot**: responsible for displaying functions.
- **Function**: provides an interface for creating graphs. It is enough to define the `pointGenerator` function.
- **ApproximationMethod**: provides an interface for creating numerical approximation methods. It is enough to define the functions `leftGenerator` and `rightGenerator`.

Graphs layer:

- **ExactSolution**: stores information about the function y .
- **Parameters**: stores information for numerical methods.
- **LError**, **MaxLTE**: auxiliary classes to display numerical method errors.
- **Methods(3)**: define the way a differential equation is approximated.

4.5 Expanding and changing functionality

4.5.1 Replacing a estimated function

To replace the function to be evaluated, it is enough to change the formulas for the derivative and its solution, and to specify the points of discontinuity.

```
ExactSolution::ExactSolution()
    :Function("Exact", QColor(10,10,10))
{
    y    = [](double x)      { return 1./(x+1); };
    difY = [](double x, double y){ return y*(y-1)/x;};
    infDisc = {-1}; // Infinite discontinuities points
    otherDisc = {0}; // all others discontinuities points
}
```

4.5.2 Adding a new numerical method

To add a new numerical method, you need to inherit ApproximationMethod and define two methods: leftGenerator and rightGenerator

```
void EulerMethod::leftGenerator(double t, QCPGraphData& prev, double step,
                                QSharedPointer<QCPGraphDataContainer> &data) const
{
    double k1 = eq->getDifY(prev);
    QCPGraphData point{t, prev.value - step*k1};
    data->add(point);
    prev = point;
}

void EulerMethod::rightGenerator(double t, QCPGraphData& prev, double step,
                                 QSharedPointer<QCPGraphDataContainer> &data) const
{
    double k1 = eq->getDifY(prev);
    QCPGraphData point{t, prev.value + step*k1};
    data->add(point);
    prev = point;
}
```
