Danis Alukaev - Variant 9.

Objectives.
Create an application that allow user to investigate impact of Numerical methods applied to ordinary differential equation provided by Variant 9.

Tasks.
1.Analytically solve given by Variant 9 differential equation.
2.Implement Euler's method, Improved Euler's method and Runge-Kutta method.
3.Implement code units to find Local Truncation Errors and Global Truncation Errors for each method.
4.Implement GUI that allow user to change settings x_0, y_0, X, N and plot the graphs of exact solution and its approximations.
5.Implement GUI that plot the graphs of Local Truncation Errors and Global Truncation Errors for a particular configuration of x_0, y_0, X, N.
6.Implement GUI that provide ability to analyze the maximal Global Truncation Errors depending on the number of grid cells.
7.Draw conclusions.

Analytical approach.
Initial equation: $y' = \frac{4}{x^2} - \frac{y}{x} - y^2$

The given function has a point of discontinuity at $x = 0$. Assume that $x \neq 0$.

Indeed, the initial equation is the Riccati equation of the form $y' + a(x)y + b(x)y^n = c(x)$ with $a(x) = \frac{1}{x}$, $b(x) = 1$, $c(x) = \frac{4}{x^2}$ and $n = 2$.

Primarily, let's find a particular solution $y_1 = y_1(x)$. One suitable solution is $y_1 = \frac{2}{x}$. Check it by substitution in initial equation:

$$-\frac{2}{x^2} = \frac{4}{x^2} - \frac{2}{x^2} - \frac{4}{x^2};\ 0 = 0 \text{ that is true.} \quad (*)$$

Let $y = y_1 + z = \frac{2}{x} + z$, where $z = z(x)$.
Make a substitution $y = \frac{2}{x} + z$ in initial equation: $-\frac{2}{x^2} + z' = \frac{4}{x^2} - \frac{1}{x} \cdot (\frac{2}{x} + z) + (\frac{2}{x} + z)^2$.
After simplification, the equation has the following form:

$$z' = -\frac{5z}{x} - z^2 \quad (1)$$

The obtained equation is the Bernoulli equation of form $z' + a(x)z = b(x)z^n$ with $a(x) = \frac{5}{x}$, $b(x) = -1$ and $n = 2$.

Consider complementary equation: $z_c' = -\frac{5z_c}{x}$, where z_c is a non-trivial solution of the complementary equation.
Divide both sides by $z_c \neq 0$: $\frac{z_c'}{z_c} = -\frac{5}{x}$. This is the separable differential equation.
Convert this equation to differential form: $\frac{dz_c}{z_c} = -\frac{5dx}{x}$.
In order to solve it integrate both sides: $\int \frac{dz_c}{z_c} = -5\int \frac{dx}{x}$.
Intermediate result: $\ln|z_c| = -5\ln|x|$.
Exponentiate both sides: $e^{\ln z_c} = e^{-5\ln x}$. After simplification, the equation has the following form: $z_c = x^{-5}$.

Let $z = uz_c = \frac{u}{x^5}$, where $u = u(x)$.
Find the derivative of z: $z' = (\frac{u}{x^5})' = \frac{u'x^5 - 5ux^4}{x^{10}} = \frac{u'}{x^5} - \frac{5u}{x^6}$.
Furthermore, make the substitution $z = \frac{u}{x^5}$ in the equation (1):
$\frac{u'}{x^5} - \frac{5u}{x^6} = -\frac{5u}{x^6} - \frac{u^2}{x^{10}}$.
The second term on left hand side and the first term on right hand side are cancelled out: $\frac{u'}{x^5} = -\frac{u^2}{x^{10}}$.
Divide both sides of obtained equation by $u^n z_c = u^2 x^{-5} \neq 0$.
We get $\frac{u'}{u^2} = -\frac{1}{x^5}$.
Convert this equation to differential form: $\frac{du}{u^2} = -\frac{dx}{x^5}$.
Integrate both sides of equation: $\int \frac{du}{u^2} = -\int \frac{dx}{x^5}$.
Intermediate result: $\frac{1}{u} = \frac{1}{4x^4} + C_1$, where constant $C_1 \in \mathbb{R}$.
After simplification, the equation has the following form: $u = -\frac{4x^4}{1+C_2x^4}$, where $C_2 = 4C_1$.
Make back substitution of u in z: $z = \frac{-4x^4}{x^5(1+C_2x^4)} = \frac{-4}{x(1+C_2x^4)}$. And, finally, substitute z in y: $y = \frac{2}{x} + z = \frac{2}{x} + \frac{-4}{x(1+C_2x^4)} = \frac{2+2C_2x^4-4}{x(1+C_2x^4)} = \frac{2(C_2x^4-1)}{x(C_2x^4+1)}$.
Indeed, we got:

$$y = \frac{2(C_2x^4-1)}{x(C_2x^4+1)} \quad (2)$$

Let's express C_2 in terms of x_0 and y_0. Substitute x_0 and y_0 in (2): $y_0 = \frac{2(C_2x_0^4-1)}{x_0(C_2x_0^4+1)}$.
Multiply this equation by $x_0(C_2x_0^4+1)$: $y_0x_0(C_2x_0^4+1) = 2(C_2x_0^4-1)$. Open the brackets: $y_0C_2x_0^5 + y_0x_0 = 2C_2x_0^4 - 2$. Combine terms with C_2: $C_2(y_0x_0^5 - 2x_0^4) = -2 - y_0x_0$.
And derive the C_2: $C_2 = \frac{-2-y_0x_0}{y_0x_0^5-2x_0^4}$.
After substitution of C_2 in (2) we get the exact solution in terms of x_0, y_0:

$$y = \frac{2((\frac{-2-y_0x_0}{y_0x_0^5-2x_0^4})x^4 - 1)}{x((\frac{-2-y_0x_0}{y_0x_0^5-2x_0^4})x^4 + 1)}$$

Let's determine the constraints on x (points of discontinuity). The numerator should not be equal to zero: $x((\frac{-2-y_0x_0}{y_0x_0^5-2x_0^4})x^4+1) \neq 0$. Therefore, $x \neq 0$ and $x \neq \pm\sqrt[4]{-\frac{y_0x_0^5-2x_0^4}{-2-y_0x_0}}$.

The critical points of the solution could be found by taking the derivative of the general solution and equalizing it to zero: $y' = \frac{-2C_2^2x^8+16C_2x^4+2}{(C_2x^5+x)^2} = 0$. It yields us: $x = \pm\sqrt[4]{\frac{4\pm\sqrt{17}}{C_2}}$, where $C_2 = \frac{-2-y_0x_0}{y_0x_0^5-2x_0^4} \neq 0$.

Consider our assumptions $z_c \neq 0$ and $u \neq 0$.
If $z_c = 0$, then $z = 0 \cdot u = 0$. Substitute $z = 0$ in y: $y = \frac{2}{x} + z = \frac{2}{x}$. It is a particular solution as was shown in (*). If $u = 0$, then $z = \frac{0}{x^5} = 0$. Substitute $z = 0$ in y: $y = \frac{2}{x} + z = \frac{2}{x}$. The same as for the previous case.

Solve initial value problem:
Given: $y(1) = 0$, i.e., $y_0 = 0$ $x_0 = 0$. Substitute in equation for C_2: $C_2 = \frac{-2-0\cdot1}{0\cdot1^5-2\cdot1^4} = 1$. Therefore, the solution for an IVP is $y = \frac{2(x^4-1)}{x(x^4+1)}$.

Answer: given equation does not have a trivial solution and has the most general non-trivial solution for the initial equation: $y = \frac{2((\frac{-2-y_0x_0}{y_0x_0^5-2x_0^4})x^4-1)}{x((\frac{-2-y_0x_0}{y_0x_0^5-2x_0^4})x^4+1)}$ on $\{x|\ x \in \mathbb{R} \ \wedge\ x_0 \in \mathbb{R} \ \wedge\ y_0 \in \mathbb{R} \ \wedge\ x \neq 0 \ \wedge\ x \neq \pm\sqrt[4]{-\frac{y_0x_0^5-2x_0^4}{-2-y_0x_0}}\}$.

Programming part.
For this computational practicum was used the C# programming language and open-source graphical (GUI) class library Windows Forms. The application has a form of Windows user interface with one resizeable window. This window contains two tabs (see Figure 1): *Approximation* and *GTE Analysis*.

The first tab displays plots of approximations obtained by Numerical methods (upper chart) and Local Truncation errors (LTE) at each grid step (bottom chart). This tab allows user to set coordinates of the first point x_0 and y_0, x coordinate of a point which approximation is required X, and number of grid steps N.
The second tab allows users to investigate convergence of Global Truncation Errors. The user is able to set the interval of interest by changing the number of grid steps N_0 and the maximum number of grid steps N. This tab incorporates plots of Global Truncation errors for N_0 (upper chart) and Global Truncation errors on the interval $[N_0, N]$ (bottom chart).

In order to increase the size of a certain chart, containers with GUI components could be scaled accordingly within the window.

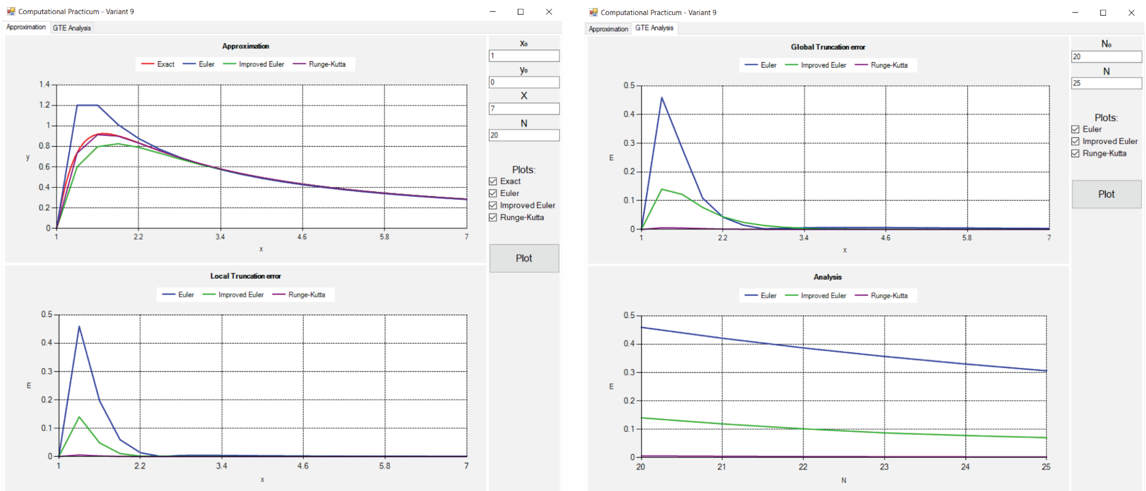

Figure 1: Interface.

When the user clicks the *Plot* button, either on the first or second tab, the charts on both tabs are updated according to the data entered in Text boxes. Moreover, the values of the number of grid steps on both tabs are synchronized for further convenience.
Both tabs contain Check boxes that allow to show/hide plots describing a correspondent solution. When Check box changes its state, all four charts are updated accordingly. It allows to show some plots with enhanced scaling.

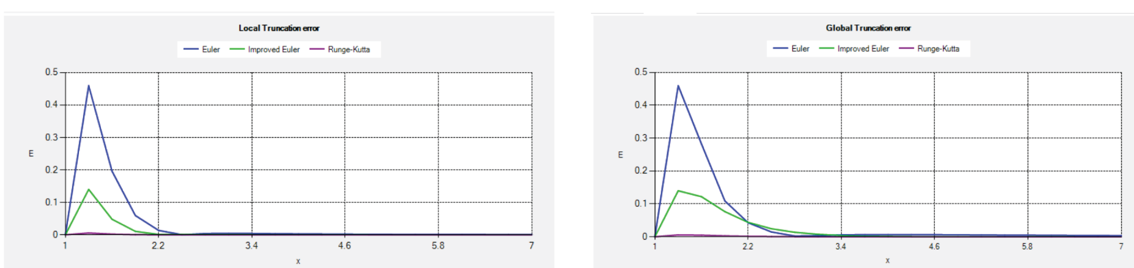

Figure 2: LTE and GTE plots.

The application provides charts with plots of LTEs (Figure 2, left) and GTEs (Figure 2, right). These values are calculated by their definitions. Specifically, the LTE is the error that the Numerical method causes during a single iteration, assuming perfect knowledge of the exact solution at the previous iteration, i.e., the absolute value of the difference between the value of the exact solution at point i and approximation based on the value of the exact solution at point $i-1$. Likewise, the GTE is the accumulation of the LTE over all of the previous iterations, i.e., the absolute value of the difference between the value of the exact solution at point i and value approximated by the Numerical method at point i).

The application is tolerant of user errors. In case there is an error in the data provided by the user, the application shows a message about the issues in the input. In particular, it might show a message of "N0 should be greater than N.", "X should be greater than x0.". Besides, the application checks whether the interval specified by the user contains points of discontinuity (See Figure 3). In case it has, the program outputs "Solution is not defined in { particular points, where it is not defined }.". Finally, due to limitations on the values for *System.Windows.Forms.DataVisualization.Charting*, WinForms fails to plot values, which coordinates exceed 10^{20}. Consequently, the application notifies the user with the comment "Due to an inaccuracy of numerical methods overflow occurred.".

```csharp
/* Checks whether solution of First order differential equation  y' = f(x, y) = 4 / x^2 - y / x - y^2
 * is continuous on specified interval. There are 3 possible points x = 0 and for positive constant ± ( -1 / C )^(1/4).
 * See report in order to find more details.
 * Returns empty array if it is continuous, otherwise - array of points that belong to interval.
 */
public override double[] IsContinuousOnInterval()
{
    // Get boundaries of the interval.
    double x0 = GetX0();
    double X = GetXMax();
    // Create a list to store points that are in interval.
    List<double> Failed = new List<double>();
    // As shown in report there are 3 possible points of discontinuity.
    double Point1, Point2, Point3;
    // First is at x = 0.
    // Check it.
    Point1 = 0;
    if (x0 <= Point1 && Point1 <= X)
        Failed.Add(Point1);
    // Get the constant.
    double c = Constant;
    if (c < 0)
    {
        // For constants that are greater than 0, there exist points ± ( -1 / C )^(1/4).
        // Check them.
        Point2 = Math.Pow(-1 / c, 1 / 4);
        if (x0 <= Point2 && Point2 <= X)
            Failed.Add(Point2);
        Point3 = -Math.Pow(-1 / c, 1 / 4);
        if (x0 <= Point3 && Point3 <= X)
            Failed.Add(Point3);
    }
    return Failed.ToArray();
}
```

Figure 3: Variant9.cs : points of discontinuity.

The program obeys OOP-design standards, in particular, the SOLID principles. The UML diagram of created classes is shown on the Figure 4.

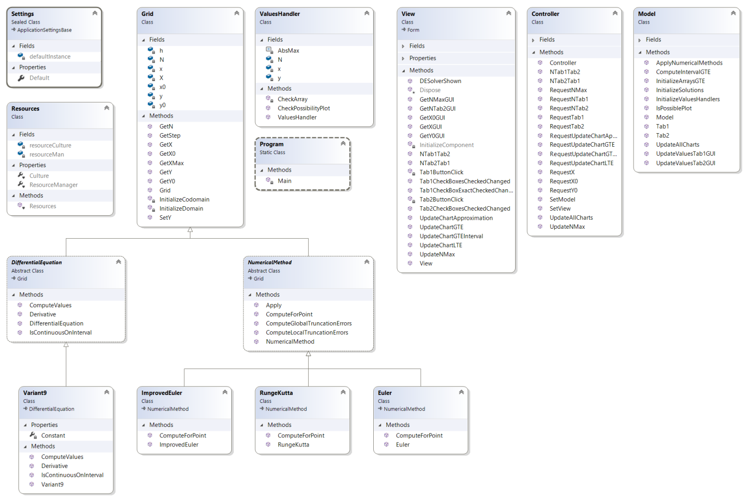

Figure 4: UML diagram.

There was developed class *Grid* used to represent set of points on coordinate plane. From this class were inherited abstract class *DifferentialEquation* used to represent first order differential equation $y' = f(x,y)$ and abstract class *NumericalMethod* used to represent Numerical Methods for solving ordinary differential equations with a given initial value. As a subtype, the abstract class *DifferentialEquation* has class *Variant9* used to represent ordinary differential equation $y' = f(x,y) = \frac{4}{x^2} - \frac{y}{x} - y^2$ given by Variant 9. As a subtypes of an abstract class *NumericalMethod* were introduced classes *Euler*, *ImprovedEuler* and *RungeKutta* used to approximate solution of an ordinary differential equation using Euler, Improved Euler, Runge-Kutta methods respectively.
Since *System.Windows.Forms.DataVisualization.Charting* has restrictions on possible values of coordinates, there introduced auxiliary class *ValuesHandler* used to check whether the sets of x and y coordinates contain values that module is not able to work with.
The program is based on Model-View-Controller schema. Class View contains methods to plot charts and monitor state of GUI components. Once state of the button or Check box changed, class View invoke Controller class to request changing of the Model. Controller acts as a pipeline between View and Model. The class Model manages the data, logic and rules of the application.

Application allows the user to analyze the Global Truncation Errors depending on the number of grid cells. Figure 5 shows the plot of GTEs for each method in a given range $[20, 40]$ with initial conditions for Variant 9 ($x_0 = 1$, $y_0 = 0$, $X = 7$).

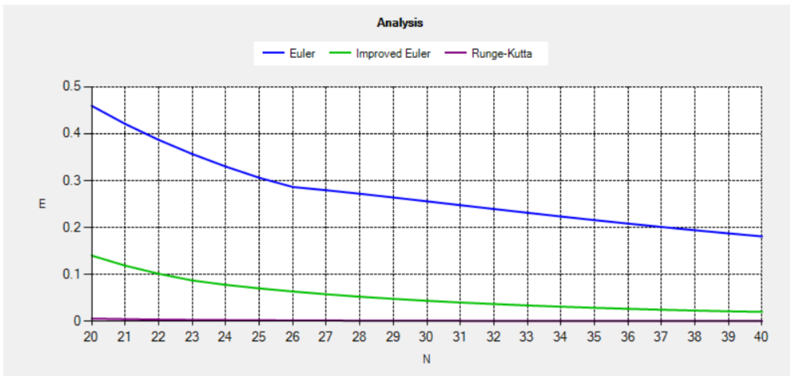

Figure 5: GTE Convergence.

As it might be seen, on the given interval the GTE functions converge, i.e., all the methods converge to the exact solution. Indeed, they require different numbers of steps to reach the necessary precision. Also, one can notice that GTEs of the Euler method are the greatest on the specified interval, while GTEs of the Runge-Kutta method are smallest.

Code is properly documented. Each class contains comprehensive description of what it does and which methods has.
[Online] Available:
`https://github.com/DanisAlukaev/F20-DE-Computational-Practicum/tree/master/Source%20Code`

Conclusions.
As a result of the practicum, an application was written to demonstrate the performance of different Numerical methods for approximation of ordinary differential equation solution. The application allows the user to change initial conditions, investigate plots of approximations, their LTEs, GTEs, and analyze GTEs convergence on a specified interval. The program is based on Model-View-Controller schema and it obeys OOP-design standards. As a result, the program will be acknowledged in a live-grading session.