

Variant 17

Task

$$y' = f(x, y) = (1 - 2y)e^x + y^2 + e^{2x}$$

$$y(-5) = 2$$

$$x \in [-5; 0]$$

Exact solution

$$f(x, y) = y' = (1 - 2y)e^x + y^2 + e^{2x}$$

$$y' + (2y - 1)e^x - y^2 - e^{2x} = 0$$

$$y' + 2e^x y - y^2 = e^x(1 + e^x) \quad - \text{DE Reccati}$$

$$\text{For substitution } y = y_1 + u$$

$$\text{Let's } y_1 = ce^x, \quad y_1' = ce^x$$

$$ce^x + 2e^x ce^x - c^2 e^{2x} - e^x - e^{2x} = 0$$

$$e^x(c - 1) + e^{2x}(2c - c^2 - 1) = 0$$

$$\text{We can see that } c = 1 \text{ is a solution}$$

$$\text{So } y_1 = e^x$$

$$\text{Then } y = e^x + u, \quad y' = e^x + u'$$

$$e^x + u' + 2e^x(e^x + u) - (e^x + u)^2 = e^x + e^{2x}$$

$$u' = u^2 \quad - \text{Usual separable differential equation}$$

$$\frac{du}{dx} = u$$

$$\int \frac{du}{u^2} = \int dx$$

$$-\frac{1}{u} = x + c$$

$$u = \frac{-1}{x+c}$$

$$y = e^x - \frac{1}{x+c} \quad - \text{general solution}$$

Let's check solution

$$y' = \frac{1}{(x+c)^2}$$

$$\frac{1}{(x+c)^2} = e^x - 2e^x(e^x - \frac{1}{x+c}) + (e^x - \frac{1}{x+c})^2 + e^{2x}$$

$$\frac{1}{(x+c)^2} = -2e^{2x} + \frac{2e^x}{x+c} + e^{2x} - \frac{2e^x}{x+c} + \frac{1}{(x+c)^2} + e^{2x}$$

$$\frac{1}{(x+c)^2} = \frac{1}{(x+c)^2} \Rightarrow \text{Solution is correct}$$

Let's find partial solution for initial conditions

$$2 = e^{-5} - \frac{1}{c-5}$$

$$e^{-5} - 2 = \frac{1}{c-5}$$

$$c = \frac{1}{e^{-5}-2} + 5$$

So partial solution: $y = e^x - \frac{1}{x + \frac{1}{e^{-5}-2} + 5}$

Chart of solution

For good displaying I take this initial conditions:

$$y(0) = 0$$

$$x \in [0; 5]$$

Let's find partial solution for initial conditions

$$0 = e^0 - \frac{1}{0+c}$$

$$1 = \frac{1}{c}$$

$$c = 1$$

So this partial solution: $y = e^x - \frac{1}{x+1}$

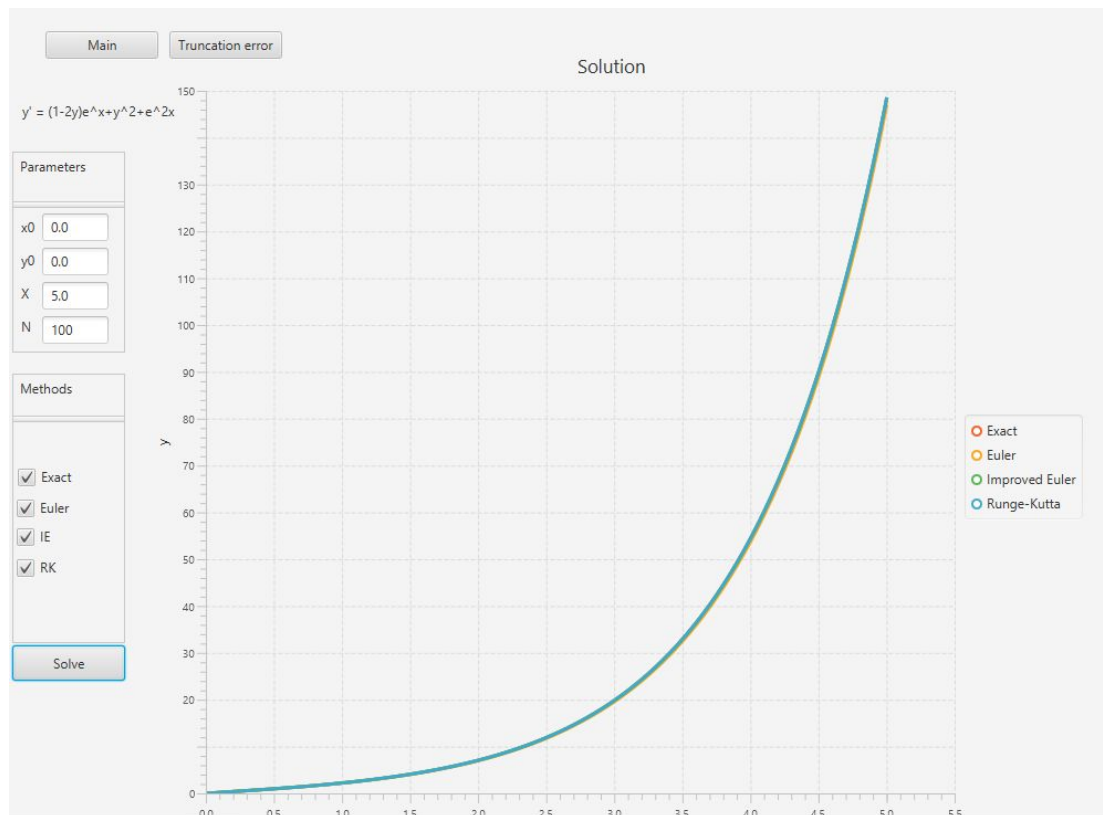


Chart of truncation error



Code

Main class

This class loads view ("Chart.fxml" file) and sets up scene.

```
public class Main extends Application {

    @Override
    public void start(Stage stage) throws Exception {
        Parent root = FXMLLoader.load(getClass().getResource("Chart.fxml"));
        Scene scene = new Scene(root);
        stage.setScene(scene);
        stage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```

Controller class

It initializes views with default values. This class contains methods which will be triggered when some button is pressed. This methods can call addition private methods for relevant cases.

```
// onClick for buttonMainView, it hide elements from Truncation error scene and make visible main scene
@FXML
private void mainView() throws IOException {
    paneTrunc.setVisible(false);
    lineCharTrunc.setVisible(false);
    vboxTrunc.setVisible(false);

    paneMain.setVisible(true);
    buttonMainSolve.setVisible(true);
    buttonTruncSolve.setVisible(false);
}

// OnClick for buttonMainSolve, this function update main chart with values from different fields
@FXML
private void updateMainChartFromFields() {
    double x0 = Double.parseDouble(textFieldMainX0.getText());
    double y0 = Double.parseDouble(textFieldMainY0.getText());
    double X = Double.parseDouble(textFieldMainX.getText());
    int N = Integer.parseInt(textFieldMainN.getText());
    deModel = new DEModel(x0, y0, X, N);
    List<XYChart.Series> plots = deModel.getSolutions(checkBoxMainExact.isSelected(),
        checkBoxMainEuler.isSelected(),
        checkBoxMainIe.isSelected(),
        checkBoxMainRk.isSelected());
    updateMainChart(plots);
}

private void updateMainChart(List<XYChart.Series> plots) {
    lineChartMain.getData().clear();
    for (int i = 0; i < 4; i++) {
        if (plots.get(i) != null)
            lineChartMain.getData().add(plots.get(i));
    }
}
```

```

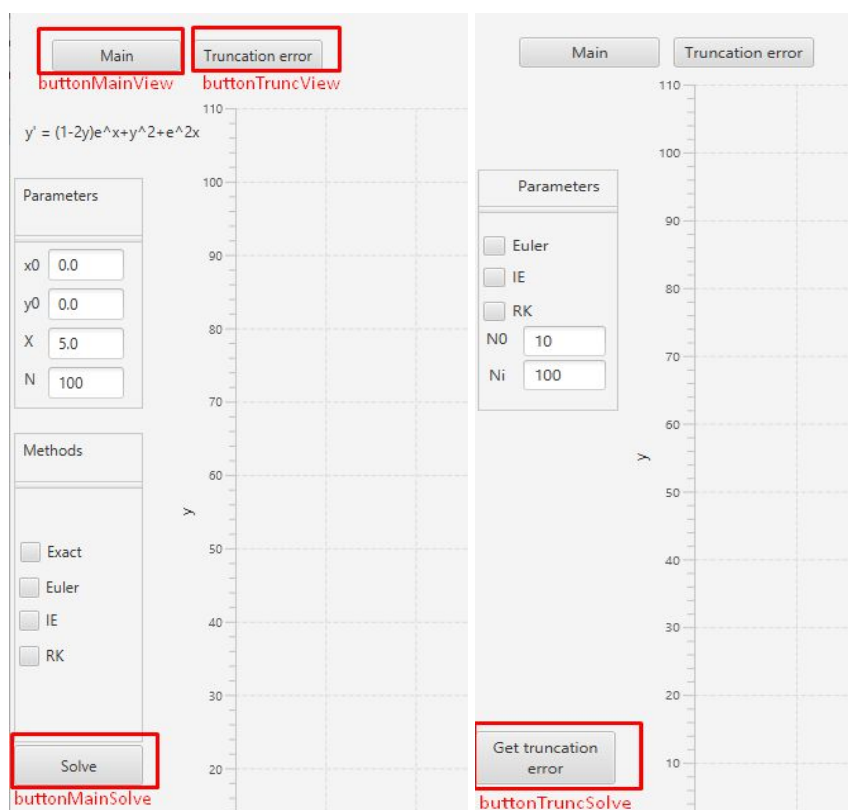
// onClick for buttonTruncView, it hide elements from paneMain (Main scene) and make visible elements of Truncation error scene
@FXML
private void truncView() throws IOException {
    paneMain.setVisible(false);
    buttonMainSolve.setVisible(false);

    paneTrunc.setVisible(true);
    lineCharTrunc.setVisible(true);
    vBoxTrunc.setVisible(true);
    buttonTruncSolve.setVisible(true);
}

// OnClick for buttonTruncSolve, this function update truncation error chart with values from different fields
@FXML
private void updateTruncChartFromFields() {
    double x0 = Double.parseDouble(textFieldMainX0.getText());
    double y0 = Double.parseDouble(textFieldMainY0.getText());
    double X = Double.parseDouble(textFieldMainX.getText());
    int N = Integer.parseInt(textFieldMainN.getText());
    deModel = new DEModel(x0, y0, X, N);
    int n0 = Integer.parseInt(textFieldTruncN0.getText());
    int ni = Integer.parseInt(textFieldTruncNi.getText());
    List<XYChart.Series> plots = deModel.getTrunc(checkBoxTruncEuler.isSelected()
        , checkBoxTruncIe.isSelected()
        , checkBoxTruncRk.isSelected()
        , n0
        , ni);
    updateTruncChart(plots);
}

```

Where buttonMainView, buttonTruncView, buttonMainSolve, buttonTruncSolve



DEModel class

This class is model for differential equation. It contains functions for solving DE with different methods and for finding truncation errors for relevant methods.

```
9
10 public class DEModel {
11     // Initial values
12     private double x0;
13     private double y0;
14     private double X; // Final value of x
15     private int N; // Number of steps
16
17     public DEModel(double x0, double y0, double X, int N) {
18         this.x0 = x0;
19         this.y0 = y0;
20         this.X = X;
21         this.N = N;
22     }
23
24     // Return the list of all required solution Series
25     public List<XYChart.Series> getSolutions(boolean exact, boolean euler, boolean ie, boolean rk) {...}
26
27     private XYChart.Series<Number, Number> exact(Double x[], Double y[], int N) {...}
28     private Double findC() { return 1 / (exp(x0) - y0) - x0; }
29
30     private XYChart.Series<Number, Number> euler(Double x[], Double y[], double h, int N) {...}
31
32     private XYChart.Series<Number, Number> ie(Double x[], Double y[], double h, int N) {...}
33
34     private XYChart.Series rk(Double x[], Double y[], double h, int N) {...}
35
36     private Double f(Double x, Double y) { return ((1 - (2 * y)) * (exp(x))) + (y * y) + ((exp(x)) * (exp(x))); }
37
38     // Create XYChart.Series from 2 arrays of double and set name to plot
39     private XYChart.Series<Number, Number> doublesToSeries(Double x[], Double y[], String name) {...}
40
41     // Return the list of all required truncation error of solution Series
42     public List<XYChart.Series> getTrunc(boolean euler, boolean ie, boolean rk, int n0, int ni) {...}
```

Success
JavaFX-