1 Постановка задачі

Необхідно розв'язати диференційне рівняння $y' = 1/(2x - y^2)$ за допомогою метода Рунге-Кутти третього порядку та модифікованого метода Ейлера (варіант 2).

2 Аналітичний розв'язок задачі

Розв'яжемо диференційне рівняння:

$$\frac{dy}{dx} = \frac{1}{(2x - y^2)}$$
$$\frac{dx}{dy} = 2x - y^2$$
$$x' - 2x = -y^2$$
$$x(y) = ce^{2y} + \frac{1}{4}(2y^2 + 2y + 1)$$

В якості початкової точки візьмемо точку $(x_0, y_0) = (0, 2)$, тоді знаходимо константу c наступним чином:

$$c = -\frac{1}{4}(2 * 2^2 + 2 * 2 + 1) * e^{-4} \approx -0.0595$$

3 Алгоритм методу чисельного диференціювання

Задаємо крок інтегрування $h = 0.01; h = 0.001$ і покроково інегруємо згідно формул Рунге-Кутти 3-го порядку та модифікованого Ейлера. У

методі Рунге-Кутти 3-го порядку це наступні формули:

$$k_1 = f(x_i, y_i)$$
$$k_2 = f(x_i + h/2, y_i + h * k_1/2)$$
$$k_3 = f(x_i + h/2, y_i + h * (2 * k_2 - k_1))$$
$$y_{i+1} = y_i + h/6 * (k1 + 4 * k_2 + k_3)$$
$$x_{i+1} = x_i + h$$

Та в методі Ейлера це наступні ітераційні формули:

$$k = f(x_i, y_i)$$
$$y_{i+1} = y_i + h/2 * (k + f(x_i + h, y_i + h * k))$$
$$x_{i+1} = x_i + h$$

4 Результати експерименту

Вихідний файл містить результати покрокового обчислення $x_i, y_i, f(x_i)$, де $f(y)$ - аналітичний розв'язок рівняння, а також різниці між реальним значенням yта отриманим за допомогою вище наведених методів. Наведемо фрагмент файла:

```
Euler 0.01:
x[i]    y[i]    Ans(y[i])       Difference
0       2       0       0
0.005   1.9987476476738364      0.00500000749264728 5    -7.492647284730591E-09
0.01    1.9974905674629544      0.010000015030023235    -1.503002323467728E-08
0.015   1.996228718498231       0.015000022612736696    -2.2612736697014135E-08
0.02    1.9949620593451478      0.02000003024140984     -3.024140983812207E-08
0.025   1.9936905479929499      0.025000037916674156    -3.791667415459221E-08
0.030000000000000002    1.9924141418435382      0.03000004563917713     -4.563917712849252 5E-08
0.035   1.9911327977000879      0.03500005340957646     -5.34095764559428E-08
0.04    1.9898464717553837      0.04000006122854227     -6.122854227103014E-08
0.045   1.9885551195798616      0.045000069096759354    -6.909675935584669E-08
0.049999999999999996    1.9872586961093521      0.050000077014924926    -7.701492493045192E-08
0.05499999999999999     1.985957155632512       0.05500008498375175     -8.498375175802764E-08
0.05999999999999999     1.984650451777937       0.06000009300396325     -9.300396325989668E-08
0.06499999999999999     1.9833385375009454      0.06500010107630105     -1.0107630106503951E-07
0.06999999999999999     1.98202136507002        0.07000010920152144     -1.092015214504416E-07
0.075   1.9806988860529031      0.07500011738039358     -1.173803935855533E-07
```

А також обчислюється середнє значення похибки. Були отримані наступні результати:

м. Ейлера $h = 0.01$	м. Ейлера $h = 0.001$	м. Рунге-Кутти $h = 0.01$	м. Рунге-Кутти $h = 0.001$
0.00171	0.00147	0.01067	0.00954

5 Аналіз роботи програми та висновки

Бачимо, що модифікований метод Ейлера дав кращі результати, ніж метод Рунге-Кутти третього порядку, причому покращення при переході до більш дрібного кроку є не дуже суттєвим.

6 Лістинг програми

```
using System;

namespace DifferentialEquation
{
    public class Function
    {
        public static double x0 = 0;
        public static double y0 = 2;
        public static double F(double x, double y)
        {
            return 1/(2 * x - y * y);
        }
        public static double AnsF(double y)
        {
            var c = (x0 - (2 * y0 * y0 + 2 * y0 + 1) / 4) /
   Math.Pow(Math.E, 2 * y0);
            return c * Math.Pow(Math.E, 2 * y) + (2 * y * y + 2
    * y + 1) / 4;
        }
    }
}
```

```
 using   System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;

namespace DifferentialEquation
{
    public class DifferentialEquation
    {
        public delegate double Func(double x, double y);
        public delegate double FuncAns(double x);
        double h; double[] x, y;
        Func F;
        public DifferentialEquation(double x0, double y0, int n
   , Func f)
```

```
        {
            x = new double[n + 1];
            y = new double[n + 1];
            x[0] = x0; y[0] = y0; h = 1.0 / (double)n;
            F = f;
        }

        public void EvaluateEuler()
        {
            for (int i = 0; i < x.Length - 1; i++)
            {
                var k = F(x[i], y[i]);
                y[i + 1] = y[i] + h / 2 * (k + F(x[i]+h, y[i] + h * k));
                x[i + 1] = x[i] + h;
            }
        }

        public void EvaluateRungeThirdOrder()
        {
            for (int i = 0; i < x.Length - 1; i++)
            {
                double k1, k2, k3;
                k1 = F(x[i], y[i]);
                k2 = F(x[i] + h / 2, y[i] + h * k1 / 2);
                k3 = F(x[i] + h / 2, y[i] + h * (2* k2 -k1));
                y[i + 1] = y[i] + h / 6 * (k1 + 4 * k2 + k3);
                x[i + 1] = x[i] + h;
            }
        }
        public void Print(StreamWriter outp, FuncAns Ans)
        {
            outp.WriteLine("x[i]\ty[i]\tAns(y[i])\tDifference");
            double avr_error = 0;
            for (int i = 0; i < x.Length; i++)
            {
                var x_real = Ans(y[i]);
                avr_error += x[i]-x_real;
                outp.WriteLine("{0}\t{1}\t{2}\t{3}", x[i], y[i], x_real, x[i]-x_real);
            }
            outp.WriteLine();
            outp.WriteLine("Average error: {0}", avr_error / x.Length);
        }
    }
}
```

```
 using   System;
using System.IO;

namespace DifferentialEquation
{
    class Program
    {
        static void Main(string[] args)
        {
            StreamWriter outp = new StreamWriter("output.txt");
            {
                DifferentialEquation de = new
   DifferentialEquation(Function.x0, Function.y0, 100,
   Function.F);
                de.EvaluateEuler(); outp.WriteLine("Euler
   0.01:"); de.Print(outp, Function.AnsF);
            }
            {
                DifferentialEquation de = new
   DifferentialEquation(Function.x0, Function.y0, 1000,
   Function.F);
                de.EvaluateEuler(); outp.WriteLine("Euler
   0.001:"); de.Print(outp, Function.AnsF);
            }
            {
                DifferentialEquation de = new
   DifferentialEquation(Function.x0, Function.y0, 100,
   Function.F);
                de.EvaluateRungeThirdOrder(); outp.WriteLine("
   RungeThirdOrder 0.01:"); de.Print(outp, Function.AnsF);
            }
            {
                DifferentialEquation de = new
   DifferentialEquation(Function.x0, Function.y0, 1000,
   Function.F);
                de.EvaluateRungeThirdOrder(); outp.WriteLine("
   RungeThirdOrder 0.001:"); de.Print(outp, Function.AnsF);
            }

            outp.Close();
        }
    }
}
```