

Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

Кафедра программных систем

**Задание №8**

**Решения задачи для дифференциального уравнения второго порядка**

Выполнил: Мороз

Илья Олегович

Группа № K3220

Преподаватель: Иванов Сергей Евгеньевич

Санкт-Петербург

2020

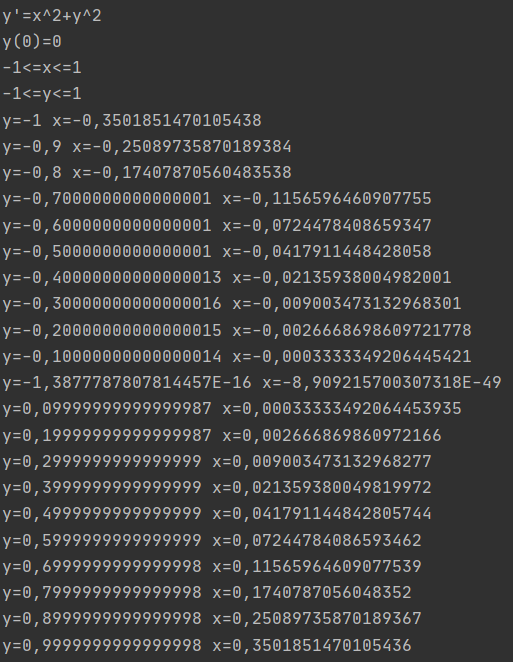
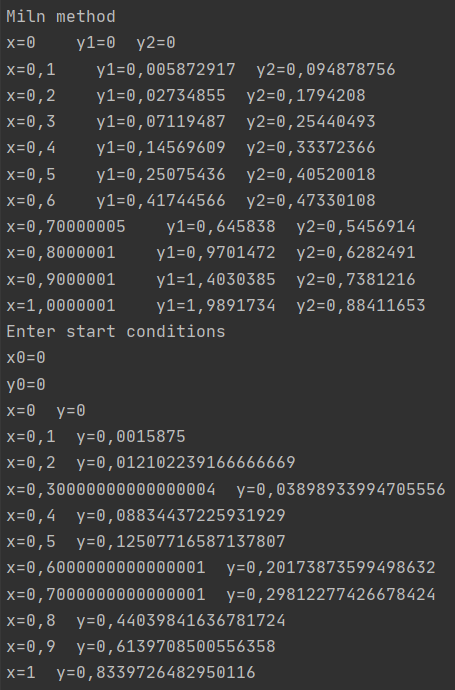
**Задание:** реализовать решение второго порядка.

**Исходный код:**

**Program.cs**

using System;  
  
namespace task8  
{  
 class Program  
 {  
 */\*  
 \* Example for Successive approximation method  
 \*/* private static double Function1(double x)  
 {  
 return (x \* x \* x) / 3;  
 }  
  
 private static double Function2(double x)  
 {  
 return Function1(x) + ((x \* x \* x \* x \* x \* x \* x) / 63);  
 }  
  
 private static double Function3(double x)  
 {  
 return Function2(x) + (2 \* Math.Pow(x, 11) / 2079) + (Math.Pow(x, 15) / 59535);  
 }  
  
 */\*  
 \* System of equations  
 \*/* private static float F1(float xa, float ya, float yb)  
 {  
 return 2 \* xa \* xa + 2 \* ya + yb;  
 }  
  
 private static float F2(float xa, float ya, float yb)  
 {  
 return 1 - 2 \* xa \* xa + 2 \* ya - yb;  
 }  
  
 */\*  
 \* Adam's method  
 \*/* private static void AdamsMethod()  
 {  
 const float **a** = 0;  
 const float **b** = 1;  
 const double **h** = 0.1;  
  
 Func<double, double, double> function = (x, y) => 5 \* x \* x - 2 \* y;  
  
 */\*  
 \* Enter initial state  
 \*/* Console.WriteLine("Enter start conditions");  
 double x0, y0;  
 Console.Write("x0=");  
 String temp = Console.ReadLine();  
 x0 = float.Parse(temp);  
 Console.Write("y0=");  
 temp = Console.ReadLine();  
 y0 = float.Parse(temp);  
  
 int numberOfSteps = (int) ((**b** - **a**) / **h**);  
 double[] values = new double[numberOfSteps];  
  
 */\*  
 \* 4th order Runge-Kutta algorithm  
 \*/* for (int i = 0; i < 4; i++)  
 {  
 double prevX = (i \* **h**) + x0;  
 double prevY;  
 if (i == 0)  
 prevY = y0;  
 else  
 prevY = values[i - 1];  
  
 double k1 = function(prevX, prevY);  
 double k2 = function(prevX + **h** / 2, prevY + **h** \* k1 / 2);  
 double k3 = function(prevX + **h** / 2, prevY + **h** \* k2 / 2);  
 double k4 = function(prevX + **h**, prevY + **h** \* k3);  
  
 values[i] = prevY + (**h** / 6 \* (k1 + 2 \* k2 + 2 \* k3 + k4));  
 }  
  
 for (int i = 4; i < numberOfSteps; i++)  
 {  
 values[i] = values[i - 1] + (**h** / 24.0) \*  
 (55 \* function(((i - 1) \* **h**) + x0, values[i - 1]) -  
 59 \* function(((i - 2) \* **h**) + x0, values[i - 2]) +  
 37 \* function(((i - 3) \* **h**) + x0, values[i - 3])  
 - 9 \* function(((i - 4) \* **h**) + x0, values[i - 4]));  
 }  
  
 */\*  
 \* Write solution  
 \*/* Console.WriteLine($"x={x0} y={y0}");  
 for (int i = 0; i < numberOfSteps; i++)  
 {  
 Console.WriteLine($"x={(i + 1) \* **h** + x0} y={values[i]}");  
 }  
 }  
  
 static void Main()  
 {  
 */\*  
 \* Successive approximation method  
 \*/* Console.WriteLine("y'=x^2+y^2");  
 Console.WriteLine("y(0)=0");  
 Console.WriteLine("-1<=x<=1");  
 Console.WriteLine("-1<=y<=1");  
 for (double j = -1; j <= 1; j += 0.1)  
 {  
 Console.WriteLine($"y={j} x={Function3(j)}");  
 }  
  
 Console.WriteLine();  
  
 #region MilnMethod  
  
 */\*  
 \* Initial state  
 \*/* Console.WriteLine("Miln method");  
 float h; *// Шаг* float a, b, k1, k2, k3, k4;  
 float r1, r2, r3, r4;  
 float eps, abs\_pogr;  
 float[] zkor = new float[12];  
 float[] zpr = new float[12];  
 float[] ypr = new float[12];  
 float[] ykor = new float[12];  
 float[] x = new float[12];  
 float[] y1 = new float[12];  
 float[] y2 = new float[12];  
  
 a = 0;  
 b = 1;  
 x[0] = a;  
 y1[0] = 0;  
 y2[0] = 0;  
 h = 0.1f;  
 eps = 0.0001f;  
  
 */\*  
 \* Solution of the system of equations by the Runge-Kutta method  
 \*/* int i = 0;  
 while (i <= 3)  
 {  
 k1 = h \* F1(x[i], y1[i], y2[i]);  
 r1 = h \* F2(x[i], y1[i], y2[i]);  
 k2 = h \* F1(x[i] + h / 2, y1[i] + k1 / 2, y2[i] + r1 / 2);  
 r2 = h \* F2(x[i] + h / 2, y1[i] + k1 / 2, y2[i] + r1 / 2);  
 k3 = h \* F1(x[i] + h / 2, y1[i] + k2 / 2, y2[i] + r2 / 2);  
 r3 = h \* F2(x[i] + h / 2, y1[i] + k2 / 2, y2[i] + r2 / 2);  
 k4 = h \* F1(x[i] + h, y1[i] + k3, y2[i] + r3);  
 r4 = h \* F2(x[i] + h, y1[i] + k3, y2[i] + r3);  
  
 y1[i + 1] = y1[i] + (k1 + 2 \* k2 + 2 \* k3 + k4) / 6;  
 y2[i + 1] = y2[i] + (r1 + 2 \* r2 + 2 \* r3 + r4) / 6;  
  
 x[i + 1] = x[i] + h;  
 i = i + 1;  
 }  
  
 i = 4;  
  
 */\*  
 \* Solution of the system of equations by the Milne method  
 \*/* while (x[i] <= b + h)  
 {  
 ypr[i] = y1[i - 4] + (4 \* h) / 3 \* (2 \* F1(x[i - 3], y1[i - 3], y2[i - 3]) -  
 F1(x[i - 2], y1[i - 2], y2[i - 2]) +  
 2 \* F1(x[i - 1], y1[i - 1], y2[i - 1]));  
 ykor[i] = y1[i - 2] + (h / 3) \* (F1(x[i - 2], y1[i - 2], y2[i - 2]) +  
 4 \* F1(x[i - 1], y1[i - 1], y2[i - 1]) + F1(x[i], ypr[i], y2[i]));  
 zpr[i] = y2[i - 4] + (4 \* h) / 3 \* (2 \* F2(x[i - 3], y1[i - 3], y2[i - 3]) -  
 F2(x[i - 2], y1[i - 2], y2[i - 2]) +  
 2 \* F2(x[i - 1], y1[i - 1], y2[i - 1]));  
 zkor[i] = y2[i - 2] + (h / 3) \* (F2(x[i - 2], y1[i - 2], y2[i - 2]) +  
 4 \* F2(x[i - 1], y1[i - 1], y2[i - 1]) + F2(x[i], zpr[i], y2[i]));  
  
 abs\_pogr = Math.Abs(ykor[i] - ypr[i]) / 29;  
 if (abs\_pogr > eps) y1[i] = ykor[i];  
 else y1[i] = ypr[i];  
 abs\_pogr = Math.Abs(zkor[i] - zpr[i]) / 29;  
 if (abs\_pogr > eps) y2[i] = zkor[i];  
 else y2[i] = zpr[i];  
  
 x[i + 1] = x[i] + h;  
 i = i + 1;  
 }  
  
 for (i = 0; i < 11; i++)  
 {  
 Console.WriteLine($"x={x[i]} y1={y1[i]} y2={y2[i]}");  
 }  
  
 #endregion  
  
 */\*  
 \* Adam's method  
 \*/* AdamsMethod();  
  
 Console.ReadLine();  
 }  
 }  
}

**Результат работы:**

****

**Вывод:** был реализовано нахождение решений системы дифференциальных уравнений второго порядка различными способами.

Исходный код также доступен на GitHub: [ilyamore88/applied-informatics/task8](https://github.com/ilyamore88/applied-informatics/tree/master/task8).