МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Тульский государственный университет»

Институт *прикладной математики и компьютерных наук*

Кафедра *вычислительной техники*

Курсовая работа по дисциплине

Численные методы

на тему: Численные методы решения дифференциальных уравнений

Студент группы 220681 Илюхин Д.М. \_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О.) (Подпись, дата)

Руководитель работы Волошко А. Г. \_\_ \_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О., должность) (Подпись, дата)

Комиссия: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Тула 2020

МИНОБРНАУКИ РОССИИ

ФГБОУ ВО «Тульский государственный университет»

Институт прикладной математики и компьютерных наук

Кафедра «Вычислительная техника»

**З А Д А Н И Е**

На курсовой проект (курсовую работу) по дисциплине (наименование дисциплины указать полностью) \_Численные методы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

студенту(ке) группы \_220681\_\_\_\_\_ Ф.И.О. студента \_Илюхин Дмитрий Михайлович\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Тема проекта (работы) \_Численное решение дифференциальных уравнений\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Входные данные \_Методы Эйлера и Рунге-Кутта. Уравнение . \_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Задание получил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ «\_\_15\_\_» \_\_февраля\_\_\_\_\_\_\_\_\_20\_20\_г.

(подпись студента)

Срок предоставления задания «\_\_23\_\_» \_\_\_\_\_\_\_\_июня\_\_\_\_\_\_\_\_\_\_\_\_ 20\_20\_г.

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (расшифровка подписи)

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_г.

К защите. Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (расшифровка подписи)

Замечания руководителя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_г.

*При защите курсового проекта (работы) наличие рецензии обязательно.*

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc42083141)

[1. ЗАДАНИЕ 5](#_Toc42083142)

[2. МАТЕМАТИЧЕСКОЕ ОПИСАНИЕ МЕТОДОВ 5](#_Toc42083143)

[3. ОПИСАНИЕ ВХОДНЫХ И ВЫХОДНЫХ ДАННЫХ 9](#_Toc42083144)

[4. АЛГОРИТМЫ РЕШЕНИЯ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ 9](#_Toc42083145)

[5. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ 11](#_Toc42083146)

[6. РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ 14](#_Toc42083147)

[7. ПРОВЕРКА КОРРЕКТНОСТИ РАБОТЫ ПРОГРАММЫ 16](#_Toc42083148)

[8. СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ РЕШЕНИЯ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ ПО КРИТЕРИЯМ ТОЧНОСТИ, ВЫЧИСЛИТЕЛЬНОЙ СЛОЖНОСТИ 18](#_Toc42083149)

[ЗАКЛЮЧЕНИЕ 20](#_Toc42083150)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 21](#_Toc42083151)

[ПРИЛОЖЕНИЕ 22](#_Toc42083152)

# ВВЕДЕНИЕ

Целями и задачами работы является получения навыков:

* анализа различных численных методов решения дифференциальных уравнений;
* разработки программных средств для решения численными методами дифференциальных уравнений.

В ходе выполнения курсовой работы необходимо разработать программу для вычисления значения дифференциального уравнения и погрешности методами Эйлера и Рунге-Кутта.

# 1. ЗАДАНИЕ

Разработать на языке программирования C# программу, реализующую решение уравнения на отрезке [a, b] ([a, b] = [x0, xn]) с шагом h методом Рунге-Кутта и методом Эйлера. Начальные значения x и y, конечное значение xn, задаются пользователем.

# 2. МАТЕМАТИЧЕСКОЕ ОПИСАНИЕ МЕТОДОВ

МЕТОД ЭЙЛЕРА

Решить дифференциальное уравнение y' = f (x, y) численным методом - это значит для заданной последовательности аргументов 𝑥0, 𝑥1,..., 𝑥𝑛 и числа 𝑦0 , не определяя функцию у = F(x), найти такие значения 𝑦1, 𝑦2 ,..., 𝑦𝑛 что 𝑦𝑖 = F(𝑥𝑖) (i = 1,2,...,n) и F(𝑥0) = 𝑦0.

Таким образом, численные методы позволяют вместо нахождения функции у = F(x) получить таблицу значений этой функции для заданной последовательности аргументов. Величина h = 𝑥𝑘 - 𝑥𝑘−1 называется шагом интегрирования. Рассмотрим некоторые из численных методов.

Метод Эйлера является сравнительно грубым и применяется в основном для ориентировочных расчётов.

Пусть дано дифференциальное уравнение первого порядка

𝑦′=𝑓(𝑥,𝑦) (1)

с начальным условием

y(𝑥0) = 𝑦0 (2)

Требуется найти решение уравнения (1) на отрезке [а, b].

Разобьём отрезок [а, b] на n равных частей и получим последовательность 𝑥0 , 𝑥1 , 𝑥2 ,..., 𝑥𝑛 , где 𝑥𝑖 = 𝑥0 + ih (i = 1, 2,..., n), a h = (b - a) / n – шаг интегрирования.

Выберем k-й участок [𝑥𝑘 , 𝑥𝑘+1] и проинтегрируем уравнение (1):

= 𝑓(𝑥𝑘,𝑦𝑘)(𝑥𝑘+1−𝑥𝑘)=𝑦′*k*ℎ (3)

Тогда формула (3) примет вид

𝑦𝑘+1 = 𝑦𝑘+𝑦′ℎ (4)

Обозначив, 𝑦𝑘+1−𝑦𝑘 =∆𝑦𝑘 т.е. 𝑦′ℎ = ∆𝑦𝑘, получим

𝑦𝑘+1=𝑦𝑘+∆𝑦𝑘 (5)

Продолжая этот процесс и каждый раз принимая подынтегральную функцию на соответствующем участке постоянной и равной её значению в начале участка, получим таблицу решений дифференциального уравнения на заданном отрезке [а, b].

Если функция f(x, y) в некотором прямоугольнике

R{|𝑥−𝑥0|≤𝑎,|𝑦−𝑦0|≤𝑏}

удовлетворяет условию

|𝑓(𝑥1,𝑦1)−𝑓(𝑥2,𝑦2)|≤𝑁|𝑦1−𝑦2| (N=const) (6)

и, кроме того

|𝑑𝑓/𝑑𝑥|=|𝜕𝑓/𝜕𝑥+𝑓 \* 𝜕𝑓/𝜕𝑦|≤𝑀 (7)

то имеет место следующая оценка погрешности:

|𝑦(𝑥𝑛)−𝑦𝑛|≤ℎ𝑀/2𝑁[(1+ℎ𝑁)𝑛−1] (8)

где y(𝑥𝑛) - значение точного решения уравнения (1) при x = 𝑥𝑛 , а 𝑦𝑛 - приближенное значение, полученное на n-м шаге.

Формула (8) имеет в основном теоретическое применение. На практике, как правило, применяют "двойной просчёт". Сначала расчёт ведётся с шагом h, затем шаг дробят и повторный расчёт ведётся с шагом h/2. Погрешность более точного значения 𝑦∗𝑛 оценивается формулой

|𝑦∗𝑛−𝑦(𝑥𝑛)|≈|𝑦∗𝑛−𝑦𝑛| (9)

МЕТОД РУНГЕ-КУТТА

Метод Рунге-Кутта является одним из методов повышенной точности. Он имеет много общего с методом Эйлера.

Пусть на отрезке [а, b] требуется найти численное решение уравнения

y' = f (x, y) (10)

с начальным условием

y(x0) = y0 (11)

Разобьём отрезок [а, b] на n равных частей точками xi = x0 + ih (i = 1,2,..., n, a h = (b - a) / n - шаг интегрирования). В методе Рунге-Кутта, так же,

как и в методе Эйлера, последовательные значения у, искомой функции у определяются по формуле:

yi+1 = yi + ∆yi (12)

Если разложить функцию у в ряд Тейлора и ограничиться членами до h4 включительно, то приращение функции ∆y можно представить в виде

∆𝑦=𝑦(𝑥+ℎ)−𝑦(𝑥)=ℎ𝑦′(𝑥)+ℎ2/2\*𝑦′′(𝑥)+ℎ3/6\*𝑦′′′(𝑥)+ℎ4/24\*𝑦′′′′(𝑥) (13)

где производные y''(x), y'''(x), y''''(x) находят последовательным дифференцированием из уравнения (12).

Вместо непосредственных вычислений по формуле (12) методом Рунге-Кутта определяют четыре числа:

𝑘1=ℎf (𝑥,𝑦); 𝑘2=ℎf(𝑥+ ℎ/2,𝑦+𝑘1/2); 𝑘3=ℎf (𝑥+ ℎ/2,𝑦+𝑘2/2); 𝑘4=ℎf (𝑥+ℎ,𝑦+𝑘3) (14)

Можно доказать, что если числам k1, k2, k3, k4 придать соответственно веса 1/6; 1/3; 1/3; 1/6, то средневзвешенное этих чисел, т. е.

(1/6)\*𝑘1+(1/3)\*𝑘2+(1/3)\*𝑘3+(1/6)\*𝑘4 (15)

с точностью до четвёртых степеней равно значению ∆y , вычисленному по формуле (13):

∆y=(1/6)\*(𝑘1+2𝑘2+2𝑘3+𝑘4) (16)

Таким образом, для каждой пары текущих значений xi и yi определяют значения

𝑘1(𝑖)=ℎf (𝑥i,𝑦i); 𝑘2(𝑖)=ℎf (𝑥i+ ℎ/2,𝑦i+𝑘1(𝑖)/2); 𝑘3(𝑖)=ℎf (𝑥i+ ℎ/2,𝑦i+𝑘2(𝑖)/2); 𝑘4(𝑖)=ℎf (𝑥i+ℎ,𝑦i+𝑘3(𝑖)) (17)

Метод Рунге-Кутта имеет порядок точности h4 на всем отрезке [а, b]. Оценка точности этого метода очень затруднительна. Грубую оценку погрешности можно получить с помощью "двойного просчёта" по формуле

|𝑦∗𝑖−𝑦(𝑥𝑖)|≈(𝑦∗𝑖−𝑦𝑖)/15 (18)

Где 𝑦(𝑥𝑖) - значение точного решения уравнения (10) в точке 𝑥𝑖 а 𝑦∗𝑖 и 𝑦𝑖 - приближенные значения, полученные с шагом h/2 и h.

Если e - заданная точность решения, то число n (число делений) для (b-a)/n определения шага интегрирования h = выбирается таким образом, чтобы

ℎ4 < e (19)

Однако шаг расчёта можно менять при переходе от одной точки к другой.

Для оценки правильности выбора шага h используют равенство:

𝑞=|(𝑘2(𝑖)−𝑘3(𝑖))/(𝑘1(𝑖)−𝑘2(𝑖))| (20)

где q должно быть равно нескольким сотым, в противном случае шаг h уменьшают.

Метод Рунге-Кутта может быть применён и к решению систем дифференциальных уравнений.

Пусть задана система дифференциальных уравнений первого порядка

{𝑦′=𝑓(𝑥,𝑦,𝑧);𝑧′=𝑔(𝑥,𝑦,𝑧) (21)

с начальными условиями

x =𝑥0 , y(𝑥0) = 𝑦0 , z(𝑥0) = 𝑧0 (22)

В этом случае параллельно определяются числа ∆yi и ∆zi:

∆𝑦𝑖=(1/6)\*(𝑘1(𝑖)+2𝑘2(𝑖)+2𝑘3(𝑖)+𝑘4(𝑖)); ∆𝑧𝑖=(1/6)\*(𝑙1(𝑖)+2𝑙2(𝑖)+2𝑙3(𝑖)+𝑙4(𝑖))} (23)

где 𝑘1(𝑖)= h𝑓(𝑥𝑖 , 𝑦𝑖 , 𝑧𝑖);

𝑙1(𝑖)= hg(𝑥𝑖 ,𝑦𝑖 ,𝑧𝑖);

𝑘2(𝑖)= hf(𝑥𝑖+ℎ/2,𝑦𝑖+𝑘1(𝑖)/2,𝑧𝑖+𝑙1(𝑖)/2);

𝑙2(𝑖)= hg(𝑥𝑖+ℎ/2,𝑦𝑖+𝑘1(𝑖)/2,𝑧𝑖+𝑙1(𝑖)/2);

𝑘3(𝑖)= hf(𝑥𝑖+ℎ/2,𝑦𝑖+𝑘2(𝑖)/2,𝑧𝑖+𝑙2(𝑖)/2);

𝑙3(𝑖)= hg(𝑥𝑖+ℎ/2,𝑦𝑖+𝑘2(𝑖)/2,𝑧𝑖+𝑙2(𝑖)/2);

𝑘4(𝑖)= hf(𝑥𝑖+ℎ,𝑦𝑖+𝑘3(𝑖),𝑧𝑖+𝑙3(𝑖));

𝑙4(𝑖)= hg(𝑥𝑖+ℎ,𝑦𝑖+𝑘3(𝑖),𝑧𝑖+𝑙3(𝑖)).

Тогда получим решение системы

𝑦𝑖+1=𝑦𝑖+∆𝑦𝑖, 𝑧𝑖+1=𝑧𝑖+∆𝑧𝑖 (24)

# 3. ОПИСАНИЕ ВХОДНЫХ И ВЫХОДНЫХ ДАННЫХ

Входные данные:

1. Начальное значение для x — ассоциированная переменная x0. Тип данных — double.
2. Начальное значение для y — ассоциированная переменная y0. Тип данных — double.
3. Последнее значение для x — ассоциированная переменная xn. Тип данных - double.

Условия для входных данных:

1. Начальное значение (переменная x0) не может быть больше Последнего (переменная xn).

# 4. АЛГОРИТМЫ РЕШЕНИЯ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

4.1.После получения входных данных вычисляется значение шага (переменная h). После чего в цикле, который выполняется 2n раз, производятся вычисления x, y(при шаге = h) и \_у(при шаге = h/2). Каждое значение y при нечётном i увеличивается на h \* f(xi-h/2, yi), каждое значение x увеличивается на h/2, каждое значение \_у увеличивается на (h/2) \* f(xi, \_yi). В ходе цикла выводятся x, y и погрешности = |yi - \_yi| (Рисунок 1).

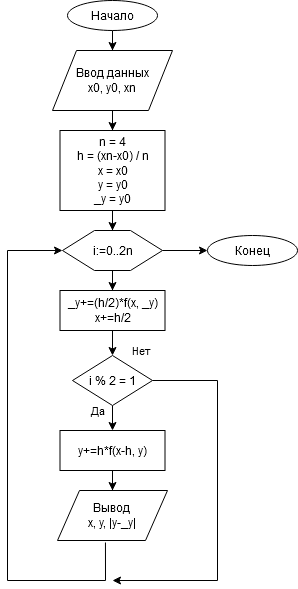


Рисунок 1. Блок-схема алгоритма Эйлера

4.2. После получения входных данных вычисляется значение шага (переменная h). Затем в цикле, который выполняется 2n раз, производятся вычисления \_y с половиной шага \_h = h/2. Для этого считаются k1, k2, k3 и k4 как \_h \* f(dxi, d\_yi). При подсчёте k1 значение dxi = xi, d\_yi = yi. При подсчёте k2 значение dxi = xi+\_h/2, d\_yi = yi+k1/2. При подсчёте k3 значение dxi = xi+\_h/2, d\_yi = yi+k2/2. При подсчёте k4 значение dxi = xi+\_h, d\_yi = yi+k3. Значение \_yi считается как сумма k1, 2\*k2, 2\*k3 и k4, делённая на 6.

Затем в цикле, который выполняется n раз, производятся вычисления y и х. Для этого считаются k1, k2, k3 и k4 как h \* f(dxi, dyi). При подсчёте k1 значение dxi = xi, dyi = yi. При подсчёте k2 значение dxi = xi+h/2, dyi = yi+k1/2. При подсчёте k3 значение dxi = xi+h/2, dyi = yi+k2/2. При подсчёте k4 значение dxi = xi+h, dyi = yi+k3. Значение yi считается как сумма k1, 2\*k2, 2\*k3 и k4, делённая на 6. Далее производится вывод полученных x, y и погрешности = |yi - \_yi|/15.

# C:\Users\VEV\Downloads\Рунге-Кутт.png

Рисунок 2 - Блок-схема алгоритма Рунге-Кутта

# 5. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

Основной класс содержит следующие глобальные переменные:

* int n = 4 – количество точек, для которых необходимо рассчитать значения;
* double x0, y0, xn – значения для трёх точек, вводимые пользователем.

Метод button1\_Click вызывается по нажатию кнопки “Подсчитать” и он берёт значения для точек их объектов numericUpDown, проверяет их корректность, заносит их в таблицу и вызывает один из методов для расчётов в зависимости от выбранного в объекте checkedListBox1:

private void button1\_Click(object sender, EventArgs e)

{

x0 = (double)numericUpDown1.Value;

xn = (double)numericUpDown3.Value;

if(x0>=xn)

{

MessageBox.Show("Первое значение х должно быть меньше последнего","Ошибка");

return;

}

y0 = (double)numericUpDown2.Value;

dataGridView1[0, 0].Value = x0;

dataGridView1[1, 0].Value = y0;

dataGridView1[2, 0].Value = 0;

if (checkedListBox1.SelectedIndex == 0)

Euler();

else

Runge\_Kutta();

}

Метод Form1\_Load срабатывает при загрузке формы и устанавливает количество строк в таблице и выбранный изначально метод:

private void Form1\_Load(object sender, EventArgs e)

{

dataGridView1.RowCount = n + 1;

checkedListBox1.SetItemChecked(0, true);

checkedListBox1.SelectedIndex = 0;

}

Метод checkedListBox1\_SelectedIndexChanged срабатывает при выборе метода. Он снимает флажок с другого метода и очищает таблицу:

private void checkedListBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

if(checkedListBox1.CheckedItems.Count > 1)

checkedListBox1.SetItemChecked(1-checkedListBox1.SelectedIndex, false);

for (int i = 0; i < 3; i++)

for (int j = 0; j < n + 1; j++)

dataGridView1[i, j].Value = null;

textBox1.Text = "";

}

Метод F задаёт функцию:

private double F(double x, double y) => 2\*x\*y+x\*Math.Pow(Math.E, -2\*x);

Метод Euler вычисляет значения для x, y, dy и заносит их в таблицу. Также метод вычисляет затраченное на вычисление время и заносит его в объект textBox1:

private void Euler()

{

Stopwatch t = new Stopwatch();

t.Start();

double h = (xn - x0) / n;

double x = x0;

double y = y0;

double \_y = y0;

for (int i = 0; i < 2\*n; i++)

{

\_y += (h/2) \* F(x, \_y);

x += h / 2;

if (i % 2 == 1)

{

y += h \* F(x-h, y);

dataGridView1[0, (i+1)/2].Value = Math.Round(x, 5);

dataGridView1[1, (i+1)/2].Value = Math.Round(y, 5);

dataGridView1[2, (i + 1) / 2].Value = Math.Round(Math.Abs(y - \_y), 5);

}

}

t.Stop();

textBox1.Text += t.Elapsed.TotalMilliseconds + " мс\r\n";

}

Метод Runge\_Kutta вычисляет значения для x, y, dy и заносит их в таблицу. Также метод вычисляет затраченное на вычисление время и заносит его в объект textBox1:

private void Runge\_Kutta()

{

Stopwatch t = new Stopwatch();

t.Start();

double h = (xn - x0) / n;

double \_h = h / 2;

double[] \_y = new double[4];

double x = x0;

double y = y0;

for (int i = 0; i < 2 \* n; i++)

{

double k1 = \_h \* F(x, y);

double k2 = \_h \* F(x+\_h/2, y+k1/2);

double k3 = \_h \* F(x+\_h/2, y+k2/2);

double k4 = \_h \* F(x+\_h, y+k3);

y += (k1 +2\*k2 +2\*k3 + k4) / 6;

x += \_h;

if (i % 2 == 1)

\_y[(i - 1) / 2] = y;

}

x = x0;

y = y0;

for (int i = 0; i < n; i++)

{

double k1 = h \* F(x, y);

double k2 = h \* F(x + h / 2, y + k1 / 2);

double k3 = h \* F(x + h / 2, y + k2 / 2);

double k4 = h \* F(x + h, y + k3);

y += (k1 + 2 \* k2 + 2 \* k3 + k4) / 6;

x += h;

dataGridView1[0, i+1].Value = Math.Round(x, 5);

dataGridView1[1, i+1].Value = Math.Round(y, 5);

dataGridView1[2, i+1].Value = Math.Round(Math.Abs(y - \_y[i])/15, 5);

}

t.Stop();

textBox1.Text += t.Elapsed.TotalMilliseconds + " мс\r\n";

}

# 6. РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ

При запуске программы откроется окно, в котором необходимо выбрать нужный метод, ввести начальные значения и нажать кнопку “Подсчитать”. При подсчёте методом Эйлера программа продемонстрирует следующие результаты (Рисунок 3).

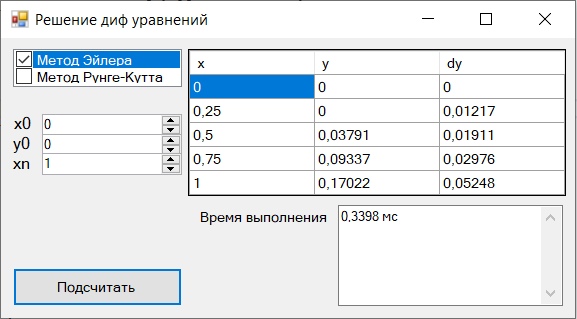


Рисунок 3. Результаты при подсчёте методом Эйлера

При подсчёте методом Эйлера программа продемонстрирует следующие результаты (Рисунок 4).

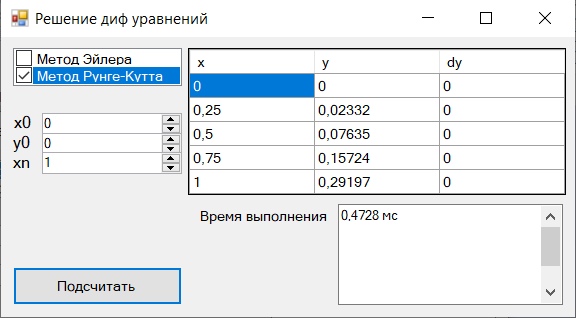


Рисунок 4. Результаты при подсчёте методом Рунге-Кутта

# 7. ПРОВЕРКА КОРРЕКТНОСТИ РАБОТЫ ПРОГРАММЫ

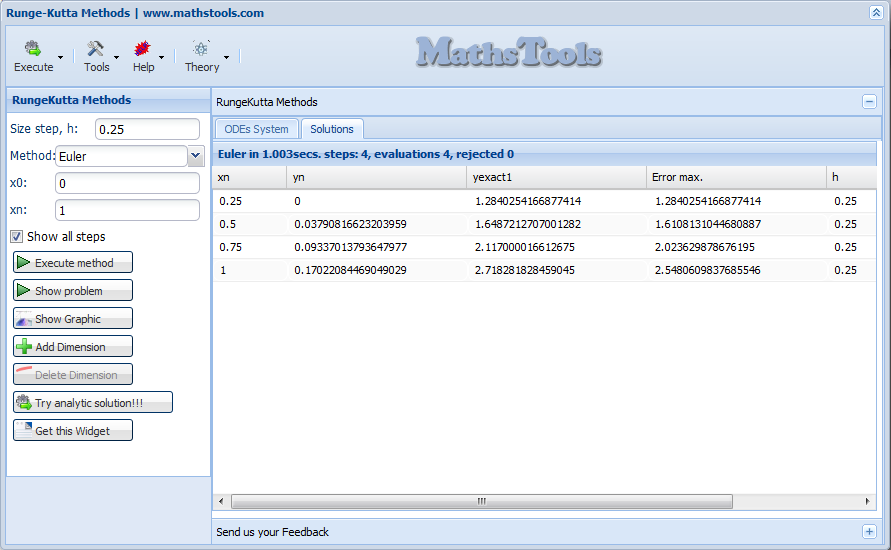
7.1 Проверка корректности работы программы для метода Эйлера с помощью онлайн калькулятора MathsTools с сайта [www.mathstools.com](http://www.mathstools.com/). Программа при таких данных дает следующий результат:

x0 = 0, x1 = 0.25, x2 = 0.5, x3 = 0.75, x4 = 1;

y0 = 0, y1 = 0, y2 = 0.03791, y3 = 0.09337, y4 = 0.17022;

dy0 = 0, dy1 = 0.01217, dy2 = 0.01911, dy3 = 0.02976, dy4 = 0.05248.

Результат проверки приведен на рисунке 5.

Рисунок 5. Проверка корректности работы программы для метода Эйлера

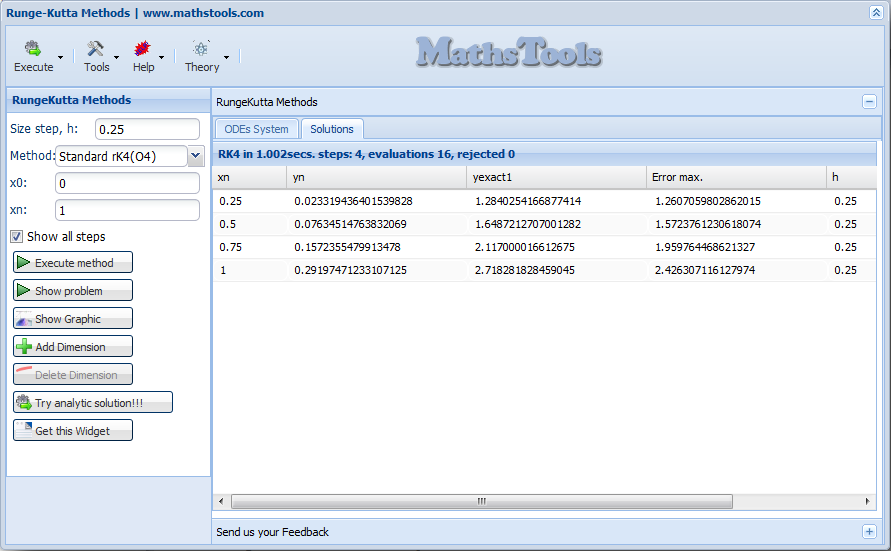
7.2 Проверка корректности работы программы для метода Рунге-Кутта с помощью того же онлайн калькулятора MathsTools. Программа при таких данных дает следующий результат:

x0 = 0, x1 = 0.25, x2 = 0.5, x3 = 0.75, x4 = 1;

y0 = 0, y1 = 0.02332, y2 = 0.07635, y3 = 0.15724, y4 = 0.29197;

dy0 = 0, dy1 = 0, dy2 = 0, dy3 = 0, dy4 = 0.

Результат проверки приведен на рисунке (Рисунок 9).

Рисунок 6. Проверка корректности работы программы для метода Рунге-Кутта

# 8. СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ РЕШЕНИЯ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ ПО КРИТЕРИЯМ ТОЧНОСТИ, ВЫЧИСЛИТЕЛЬНОЙ СЛОЖНОСТИ

Тестирование методов производилось на компьютере со следующими характеристиками:

* Операционная система Windows 10;
* Процессор Intel(R) Core(TM) i5-8250U с частотой 1.60 GHz;
* Оперативная память 6 ГБ;
* Видеокарта NVIDIA GeForce GTX 1050.

Метод Рунге-Кутта оказывается точнее как по фактической погрешности времени жизни решения, так и по количеству итераций. Но при этом он работает дольше, так как в цикле для вычисления нового значения методом Эйлера выполняется две операции, а методом Рунге-Кутты - 11 операций.

Для сравнения скорости вычислений в программу был добавлен таймер и вывод его результата для каждого из методов. Таким образом, при y0 = 1, x0 = 0, xn = 1 методом Эйлера решение было найдено в среднем за 0.0119 миллисекунд по результатам 5 экспериментов (Рисунок 7). Для этих же значений, решение методом Рунге-Кутта было найдено в среднем за 0.0163 миллисекунд (Рисунок 8), т. е. больше чем для метода Эйлера на 0,0044 миллисекунд. При решении уравнения методом Эйлера, средняя погрешность равна 0.2838. В то время как для метода Рунге-Кутта средняя погрешность равна 0, т.е. меньше средней погрешность для метода Эйлера на 0.2838. Соответственно можно сделать вывод, что метод Рунге-Кутта имеет большую точность чем метод Эйлера.

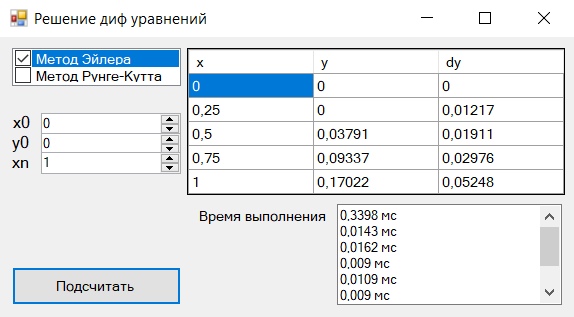


Рисунок 7. Время вычисления уравнения методом Эйлера

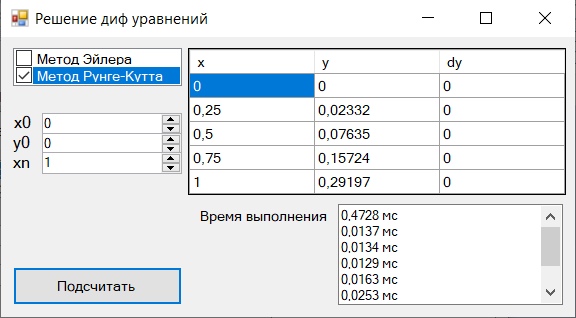


Рисунок 8. Время вычисления уравнения методом Рунге-Кутта

# ЗАКЛЮЧЕНИЕ

В ходе выполнения данной работы были анализированы 2 численных метода решения дифференциальных уравнений: метод Эйлера и метод Рунге-Кутта. Было разработано программное средство для решения численными методами Эйлера и Рунге-Кутта дифференциальных уравнений.

Также были выполнены цели и задачи работы: получены навыки анализа различных численных методов решения дифференциальных уравнений; получены навыки разработки программных средств для решения численными методами дифференциальных уравнений.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Википедия. Метод Эйлера [Электронный ресурс]. Режим доступа: https://ru.wikipedia.org/wiki/Метод\_Эйлера.

2. Википедия. Метод Рунге-Кутты [Электронный ресурс]. Режим доступа: https://ru.wikipedia.org/wiki/Метод\_Рунге\_—\_Кутты.

3. MathsTools [Электронный ресурс]. Режим доступа: www.mathstools.com.

4. Студопедия. Численное решение ОДУ методом Рунге-Кутты 2 порядка [Электронный ресурс]. Режим доступа: https://studopedia.ru/21\_47207\_chislennoe-reshenie-odu-metodom-runge-kutti—poryadka.html.

5. Хабр. Численное решение математических моделей объектов заданных системами дифференциальных уравнений [Электронный ресурс]. Режим доступа : https://habr.com/ru/post/418139/.

# ПРИЛОЖЕНИЕ

**Листинг файла Form1.cs**

using System;

using System.Windows.Forms;

using System.Diagnostics;

namespace difur

{

public partial class Form1 : Form

{

int n = 4;

double x0, y0, xn;

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

x0 = (double)numericUpDown1.Value;

xn = (double)numericUpDown3.Value;

if(x0>=xn)

{

MessageBox.Show("Первое значение х должно быть меньше последнего","Ошибка");

return;

}

y0 = (double)numericUpDown2.Value;

dataGridView1[0, 0].Value = x0;

dataGridView1[1, 0].Value = y0;

dataGridView1[2, 0].Value = 0;

if (checkedListBox1.SelectedIndex == 0)

Euler();

else

Runge\_Kutta();

}

private void Form1\_Load(object sender, EventArgs e)

{

dataGridView1.RowCount = n + 1;

checkedListBox1.SetItemChecked(0, true);

checkedListBox1.SelectedIndex = 0;

}

private void checkedListBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

if(checkedListBox1.CheckedItems.Count > 1)

checkedListBox1.SetItemChecked(1-checkedListBox1.SelectedIndex, false);

for (int i = 0; i < 3; i++)

for (int j = 0; j < n + 1; j++)

dataGridView1[i, j].Value = null;

textBox1.Text = "";

}

private double F(double x, double y) => 2\*x\*y+x\*Math.Pow(Math.E, -2\*x);

private void Euler()

{

Stopwatch t = new Stopwatch();

t.Start();

double h = (xn - x0) / n;

double x = x0;

double y = y0;

double \_y = y0;

for (int i = 0; i < 2\*n; i++)

{

\_y += (h/2) \* F(x, \_y);

x += h / 2;

if (i % 2 == 1)

{

y += h \* F(x-h, y);

dataGridView1[0, (i+1)/2].Value = Math.Round(x, 5);

dataGridView1[1, (i+1)/2].Value = Math.Round(y, 5);

dataGridView1[2, (i + 1) / 2].Value = Math.Round(Math.Abs(y - \_y), 5);

}

}

t.Stop();

textBox1.Text += t.Elapsed.TotalMilliseconds + " мс\r\n";

}

private void Runge\_Kutta()

{

Stopwatch t = new Stopwatch();

t.Start();

double h = (xn - x0) / n;

double \_h = h / 2;

double[] \_y = new double[4];

double x = x0;

double y = y0;

for (int i = 0; i < 2 \* n; i++)

{

double k1 = \_h \* F(x, y);

double k2 = \_h \* F(x+\_h/2, y+k1/2);

double k3 = \_h \* F(x+\_h/2, y+k2/2);

double k4 = \_h \* F(x+\_h, y+k3);

y += (k1 +2\*k2 +2\*k3 + k4) / 6;

x += \_h;

if (i % 2 == 1)

\_y[(i - 1) / 2] = y;

}

x = x0;

y = y0;

for (int i = 0; i < n; i++)

{

double k1 = h \* F(x, y);

double k2 = h \* F(x + h / 2, y + k1 / 2);

double k3 = h \* F(x + h / 2, y + k2 / 2);

double k4 = h \* F(x + h, y + k3);

y += (k1 + 2 \* k2 + 2 \* k3 + k4) / 6;

x += h;

dataGridView1[0, i+1].Value = Math.Round(x, 5);

dataGridView1[1, i+1].Value = Math.Round(y, 5);

dataGridView1[2, i+1].Value = Math.Round(Math.Abs(y - \_y[i])/15, 5);

}

t.Stop();

textBox1.Text += t.Elapsed.TotalMilliseconds + " мс\r\n";

}

}

}

**Листинг файла Form1.Designer.cs**

namespace difur

{

partial class Form1

{

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

private System.ComponentModel.IContainer components = null;

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

/// <param name="disposing">истинно, если управляемый ресурс должен быть удален; иначе ложно.</param>

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Код, автоматически созданный конструктором форм Windows

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

private void InitializeComponent()

{

this.dataGridView1 = new System.Windows.Forms.DataGridView();

this.Column2 = new System.Windows.Forms.DataGridViewTextBoxColumn();

this.Column3 = new System.Windows.Forms.DataGridViewTextBoxColumn();

this.Column1 = new System.Windows.Forms.DataGridViewTextBoxColumn();

this.checkedListBox1 = new System.Windows.Forms.CheckedListBox();

this.label1 = new System.Windows.Forms.Label();

this.label2 = new System.Windows.Forms.Label();

this.label3 = new System.Windows.Forms.Label();

this.numericUpDown1 = new System.Windows.Forms.NumericUpDown();

this.numericUpDown2 = new System.Windows.Forms.NumericUpDown();

this.numericUpDown3 = new System.Windows.Forms.NumericUpDown();

this.button1 = new System.Windows.Forms.Button();

this.label4 = new System.Windows.Forms.Label();

this.textBox1 = new System.Windows.Forms.TextBox();

((System.ComponentModel.ISupportInitialize)(this.dataGridView1)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.numericUpDown1)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.numericUpDown2)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.numericUpDown3)).BeginInit();

this.SuspendLayout();

//

// dataGridView1

//

this.dataGridView1.AllowUserToAddRows = false;

this.dataGridView1.AllowUserToDeleteRows = false;

this.dataGridView1.ColumnHeadersHeightSizeMode = System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;

this.dataGridView1.Columns.AddRange(new System.Windows.Forms.DataGridViewColumn[] {

this.Column2,

this.Column3,

this.Column1});

this.dataGridView1.Location = new System.Drawing.Point(187, 11);

this.dataGridView1.Margin = new System.Windows.Forms.Padding(3, 2, 3, 2);

this.dataGridView1.Name = "dataGridView1";

this.dataGridView1.ReadOnly = true;

this.dataGridView1.RowHeadersVisible = false;

this.dataGridView1.RowTemplate.Height = 24;

this.dataGridView1.Size = new System.Drawing.Size(378, 147);

this.dataGridView1.TabIndex = 0;

//

// Column2

//

this.Column2.HeaderText = "x";

this.Column2.Name = "Column2";

this.Column2.ReadOnly = true;

//

// Column3

//

this.Column3.HeaderText = "y";

this.Column3.Name = "Column3";

this.Column3.ReadOnly = true;

//

// Column1

//

this.Column1.HeaderText = "dy";

this.Column1.Name = "Column1";

this.Column1.ReadOnly = true;

//

// checkedListBox1

//

this.checkedListBox1.CheckOnClick = true;

this.checkedListBox1.FormattingEnabled = true;

this.checkedListBox1.Items.AddRange(new object[] {

"Метод Эйлера",

"Метод Рунге-Кутта"});

this.checkedListBox1.Location = new System.Drawing.Point(12, 11);

this.checkedListBox1.Margin = new System.Windows.Forms.Padding(3, 2, 3, 2);

this.checkedListBox1.Name = "checkedListBox1";

this.checkedListBox1.Size = new System.Drawing.Size(169, 38);

this.checkedListBox1.TabIndex = 1;

this.checkedListBox1.SelectedIndexChanged += new System.EventHandler(this.checkedListBox1\_SelectedIndexChanged);

//

// label1

//

this.label1.AutoSize = true;

this.label1.Font = new System.Drawing.Font("Microsoft Sans Serif", 10F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.label1.Location = new System.Drawing.Point(9, 75);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(26, 20);

this.label1.TabIndex = 2;

this.label1.Text = "х0";

//

// label2

//

this.label2.AutoSize = true;

this.label2.Font = new System.Drawing.Font("Microsoft Sans Serif", 10F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.label2.Location = new System.Drawing.Point(8, 95);

this.label2.Name = "label2";

this.label2.Size = new System.Drawing.Size(26, 20);

this.label2.TabIndex = 3;

this.label2.Text = "у0";

//

// label3

//

this.label3.AutoSize = true;

this.label3.Font = new System.Drawing.Font("Microsoft Sans Serif", 10F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.label3.Location = new System.Drawing.Point(8, 115);

this.label3.Name = "label3";

this.label3.Size = new System.Drawing.Size(26, 20);

this.label3.TabIndex = 4;

this.label3.Text = "xn";

//

// numericUpDown1

//

this.numericUpDown1.Location = new System.Drawing.Point(41, 76);

this.numericUpDown1.Name = "numericUpDown1";

this.numericUpDown1.Size = new System.Drawing.Size(140, 22);

this.numericUpDown1.TabIndex = 6;

//

// numericUpDown2

//

this.numericUpDown2.Location = new System.Drawing.Point(41, 96);

this.numericUpDown2.Name = "numericUpDown2";

this.numericUpDown2.Size = new System.Drawing.Size(140, 22);

this.numericUpDown2.TabIndex = 7;

//

// numericUpDown3

//

this.numericUpDown3.Location = new System.Drawing.Point(41, 115);

this.numericUpDown3.Name = "numericUpDown3";

this.numericUpDown3.Size = new System.Drawing.Size(140, 22);

this.numericUpDown3.TabIndex = 8;

this.numericUpDown3.Value = new decimal(new int[] {

1,

0,

0,

0});

//

// button1

//

this.button1.Location = new System.Drawing.Point(12, 230);

this.button1.Name = "button1";

this.button1.Size = new System.Drawing.Size(169, 38);

this.button1.TabIndex = 10;

this.button1.Text = "Подсчитать";

this.button1.UseVisualStyleBackColor = true;

this.button1.Click += new System.EventHandler(this.button1\_Click);

//

// label4

//

this.label4.AutoSize = true;

this.label4.Location = new System.Drawing.Point(196, 170);

this.label4.Name = "label4";

this.label4.Size = new System.Drawing.Size(135, 17);

this.label4.TabIndex = 11;

this.label4.Text = "Время выполнения\r\n";

//

// textBox1

//

this.textBox1.Location = new System.Drawing.Point(337, 167);

this.textBox1.Multiline = true;

this.textBox1.Name = "textBox1";

this.textBox1.ScrollBars = System.Windows.Forms.ScrollBars.Vertical;

this.textBox1.Size = new System.Drawing.Size(225, 101);

this.textBox1.TabIndex = 12;

//

// Form1

//

this.AutoScaleDimensions = new System.Drawing.SizeF(8F, 16F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.ClientSize = new System.Drawing.Size(574, 280);

this.Controls.Add(this.textBox1);

this.Controls.Add(this.label4);

this.Controls.Add(this.button1);

this.Controls.Add(this.numericUpDown3);

this.Controls.Add(this.numericUpDown2);

this.Controls.Add(this.numericUpDown1);

this.Controls.Add(this.label3);

this.Controls.Add(this.label2);

this.Controls.Add(this.label1);

this.Controls.Add(this.checkedListBox1);

this.Controls.Add(this.dataGridView1);

this.Margin = new System.Windows.Forms.Padding(3, 2, 3, 2);

this.Name = "Form1";

this.Text = "Решение диф уравнений";

this.Load += new System.EventHandler(this.Form1\_Load);

((System.ComponentModel.ISupportInitialize)(this.dataGridView1)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.numericUpDown1)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.numericUpDown2)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.numericUpDown3)).EndInit();

this.ResumeLayout(false);

this.PerformLayout();

}

#endregion

private System.Windows.Forms.DataGridView dataGridView1;

private System.Windows.Forms.DataGridViewTextBoxColumn Column2;

private System.Windows.Forms.DataGridViewTextBoxColumn Column3;

private System.Windows.Forms.DataGridViewTextBoxColumn Column1;

private System.Windows.Forms.CheckedListBox checkedListBox1;

private System.Windows.Forms.Label label1;

private System.Windows.Forms.Label label2;

private System.Windows.Forms.Label label3;

private System.Windows.Forms.NumericUpDown numericUpDown1;

private System.Windows.Forms.NumericUpDown numericUpDown2;

private System.Windows.Forms.NumericUpDown numericUpDown3;

private System.Windows.Forms.Button button1;

private System.Windows.Forms.Label label4;

private System.Windows.Forms.TextBox textBox1;

}

}