МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА

(САМАРСКИЙ УНИВЕРСИТЕТ)»

ФАКУЛЬТЕТ ИНФОРМАТИКИ

кафедра программных систем

Отчет к курсовой работе по дисциплине

«Вычислительные методы»

Вариант 13

Выполнил:

Поборознюк П. Д.

6313-020302D

Проверил:

Лобанков А.А.

Самара 2019

СОДЕРЖАНИЕ

Оглавление

[ВВЕДЕНИЕ 3](#_Toc28168825)

[В данной работе производится численное дифференцирование заданного уравнения. 3](#_Toc28168826)

[1 Описание и анализ вычислительных методов, постановка задачи. 4](#_Toc28168827)

[1.1 Описание и анализ вычислительных методов, близких к заданному. 4](#_Toc28168828)

[1.1.1 Метод Рунге-Кутты 4](#_Toc28168829)

[1.2 Описание метода, реализуемого в проекте. 4](#_Toc28168830)

[1.3 Постановка задачи. 6](#_Toc28168831)

[2 Разработка программы. 7](#_Toc28168832)

[2.1 Описание алгоритма вычислений. 7](#_Toc28168833)

[2.2 Перечень исключительных ситуаций. 8](#_Toc28168834)

[2.3 Выбор и обоснование комплекса программных средств. 8](#_Toc28168835)

[2.3.1 Выбор языка программирования и среды разработки. 8](#_Toc28168836)

[2.3.2 Выбор операционной системы. 9](#_Toc28168837)

[3 Реализация программы. 9](#_Toc28168838)

[3.1 Разработка и описание интерфейса пользователя. 9](#_Toc28168839)

[3.2 Описание тестовых примеров. 11](#_Toc28168840)

[ЗАКЛЮЧЕНИЕ 13](#_Toc28168841)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 14](#_Toc28168842)

[ПРИЛОЖЕНИЕ 15](#_Toc28168843)

[Приложение А. Руководство пользователя. 15](#_Toc28168844)

[Приложение Б. Листинг программы. 15](#_Toc28168845)

# ВВЕДЕНИЕ

В данной работе производится численное дифференцирование заданного уравнения.

Задача численного интегрирования состоит в нахождении приближенного значения определенного интеграла с помощью некоторой приближенной формулы через известные значения подынтегральной функции f(x) (иногда через значения ее производных) в заданных точках.

Численное интегрирование применяется, когда:

* сама подынтегральная функция не задана аналитически, а, например, представлена в виде таблицы значений;
* аналитическое представление подынтегральной функции известно, но её первообразная не выражается через аналитические функции (или это делать крайне сложно).

# 1 Описание и анализ вычислительных методов, постановка задачи.

## 1.1 Описание и анализ вычислительных методов, близких к заданному.

### 1.1.1 Метод Рунге-Кутты

Идея метода Рунге-Кутты была предложена К.Рунге, а затем была развита В.Куттой. Окончательно методы Рунге-Кутты сформировались в современном виде в 60-х годах прошлого века благодаря работам Батчера. Если метод Эйлера использует одно вычисление производной на каждом шаге численного интегрирования, то в методах Рунге- Кутты таких вычислений несколько, что позволяет повысить точность интегрирования системы. Общая формула методов Рунге-Кутты имеет вид

где *S* - количество этапов метода. Формулы методов Рунге-Кутты являются одношаговыми, поэтому удобны при программировании и позволяют легко менять шаг интегрирования в процессе интегрирования. Возможно, наиболее известной является в настоящее время формула 4-х этапного классического метода Рунге-Кутты [1].

## 1.2 Описание метода, реализуемого в проекте.

В данном проекте реализован метод Хемминга коррекции-прогноза.

Методами прогноза и коррекции (которые также называют предсказывающе-исправляющими методами или схемами "предиктор-корректор") в вычислительной математике называют группу методов численного решения различных задач, у которых каждый шаг состоит из двух основных действий. Первое (предиктор) заключается в вычислении грубого начального приближения искомой величины, выполняется однократно. Второе действие (корректор) уточняет его. Данное действие может выполняться как однократно, так и итерационно неограниченное число раз - для достижения требуемой точности. В зависимости от сочетания предиктора (способа расчета начального приближения) и корректора (способа его уточнения) могут быть получены различные варианты численных методов решения задач рассматриваемого вида.

В моём задании используется метод Хемминга со следующими формулами:

Формула прогноза:

Формула коррекции:

## 1.3 Постановка задачи.

Решение начальной задачи Коши. Задано дифференциальное уравнение , начальная точка . Найти решение на интервале . Реализовать метод Хемминга прогноза-коррекции, с оценкой глобальной методом Рунге. Вывести графики функции.

# 2 Разработка программы.

## 2.1 Описание алгоритма вычислений.

В упрощенном виде предложенные методы можно представить в виде блок- схемы (рисунок 1).

Рисунок 1 – Упрощенная блок-схема метода вычисления интеграла

Вычисление значений x(1), x(2) и x(3) методом Рунге-Кутты

Вычисление прогноза методом Хемминга

Вычисление корректированного значения методом Хемминга

Вычисление значения погрешности методом Рунге

Вычисление x(4)..x(N) на [0,T]

Так как метод прогноза-коррекции Хемминга имеет 4 порядок точности, то и вычисление значений x(1), x(2) и x(3), соответственно, производится методом Рунге-Кутты 4-го порядка точности.

Пользователь вводит значение правой границы интегрирования Т и начальную точку x(0), а также частоту дискретизации посредством задания количества точек. При загрузке приложения все значения уже заданы по умолчанию, поэтому их ввод, вообще говоря, необязателен. Затем вычисляются все значения функции f(x) и готовятся к выводу на график. После этого, вычисляются значения для «разгона» метода прогноза-коррекции. Вычисление производится с помощью метода Рунге-Кутты. Далее программа численно интегрирует все значения производной методом прогноза-коррекции: вначале для каждой точки считаются вначале прогноз, а затем, на его основе, откорректированное значение точки.

Затем это повторяется второй раз для количества точек 2N и, соответственно, с шагом h/2. По завершении и этих вычислений, считается глобальная погрешность методом Рунге для 4-го порядка точности.

## 2.2 Перечень исключительных ситуаций.

Программа обрабатывает единственную возможную исключительную ситуацию: Неправильный ввод исходных данных.

## 2.3 Выбор и обоснование комплекса программных средств.

### 2.3.1 Выбор языка программирования и среды разработки.

Я выбрал интегрированную среду разработки Visual Studio 2017 и язык C# в связи с их надёжностью и известностью, а также возможностью простого создания приятного и качественного пользовательского интерфейса с помощью мощного встроенного конструктора приложений Windows-форм.

В программе используется пакет NuGet “ZedGraph”, предоставляющий инструмент для точного построения графиков, возможность их детального рассмотрения, а также их сохранения и печати.

### 2.3.2 Выбор операционной системы.

В связи с повсеместной распространённостью операционных систем Windows, реализованное программное обеспечение поддерживается большинством персональных компьютеров.

# 3 Реализация программы.

## 3.1 Разработка и описание интерфейса пользователя.

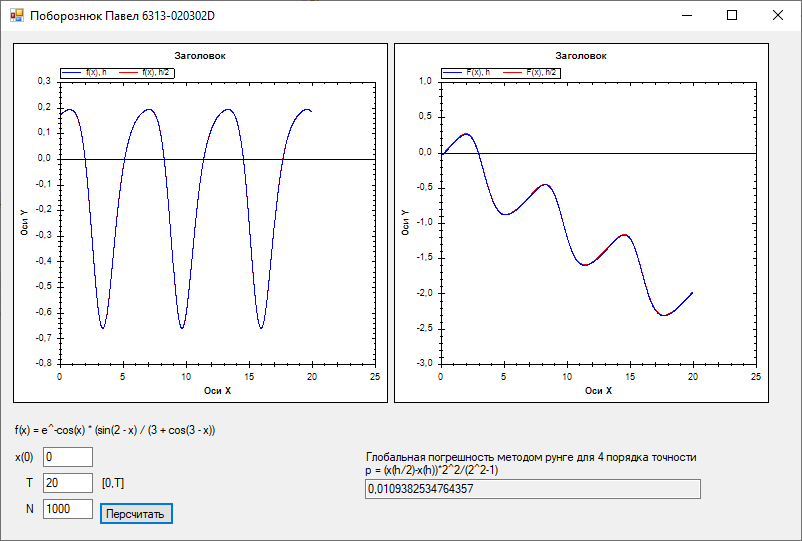
Пользовательский интерфейс интуитивно понятен. На Windows-форме присутствуют 2 графика: слева - график функции f(x) на отрезке [0,T] и справа - график численно проинтегрированной функции на том же отрезке.

Для задания интервала интегрирования и начального значения нужно воспользоваться соответствующими textbox элементами. Для обработки результатов необходимо нажать кнопку «Пересчитать».

Графики функции и её численного интегрирования интерактивны, принцип работы с ними описан в приложении А(Руководстве пользователя).

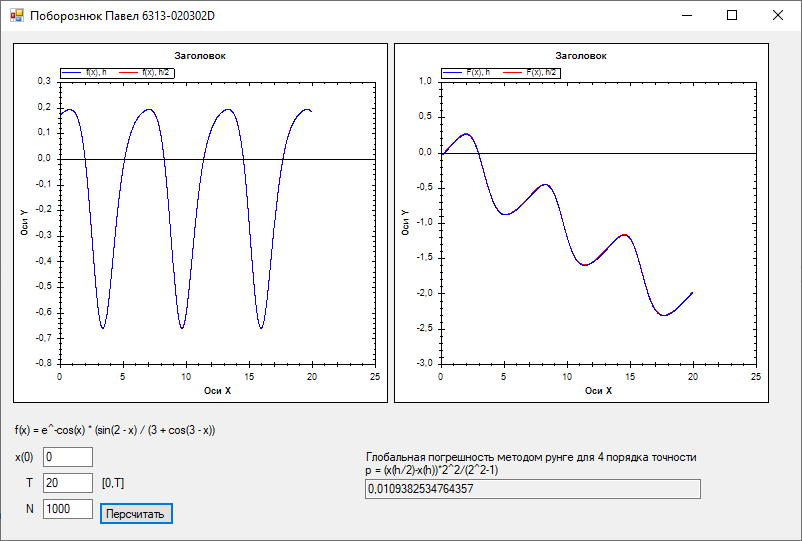
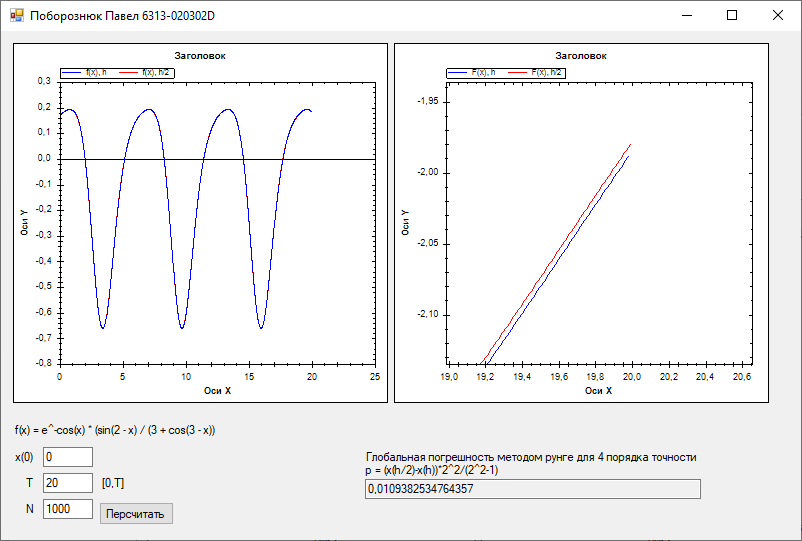
Значение глобальной погрешности выводится в соответствующем textbox в правой-нижней части формы. Также описывается сама формула для вычисления погрешности, где x(h/2) – это последнее значение численной интеграции с шагом h/2, а x(h), соответственно, последнее значение численной интеграции с шагом h.

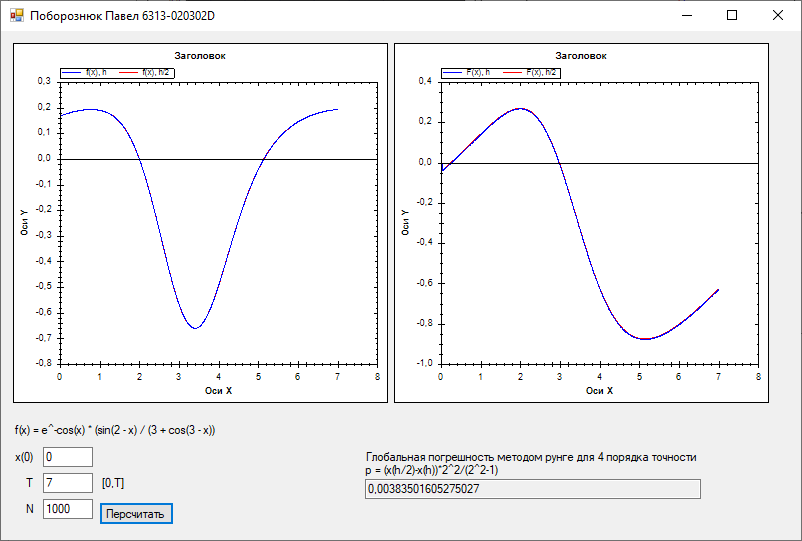
Пользовательский интерфейс приведен на рисунке 3.

  
Рисунок 3 – Пользовательский интерфейс

## 3.2 Описание тестовых примеров.

Пример со вводом интервала интегрирования с явно выраженной колебательной структурой графика проиллюстрирован на рисунке 4. На рисунке 5 выведены в более крупном масштабе последние значения численно-проинтегрированного графика. На рисунке 6 выведен график, находящийся в пределах одного периода колебаний.

Рисунок 4 – График с выраженной периодичностью. Рисунок 5 – Приближение последнего значения графика из рисунка 4.

  
Рисунок 6 – График в пределах одного периода колебаний.

# ЗАКЛЮЧЕНИЕ

В этой работе был рассмотрен метод прогноза-коррекции Хемминга четвёртого и порядка точности с оценкой глобальной погрешности методом Рунге и было реализовано desktop-приложение для расчётов и визуализации интегрирования заданной функции.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Методические указания к лабораторным работам по вычислительным методам / Ю.М. Заболотнов. Кафедра программных систем. Самарский университет. 2018.
2. Теория автоматического управления: Учеб. для вузов по спец. «Автоматика и телемеханика». В 2-х ч. Ч. I. Теория линейных систем автоматического управления / Н. А. Бабаков, А. А. Воронов, А. А. Воронова и др.; Под ред. А. А. Воронова.—2-е изд., перераб. и доп. — М.: Высш. шк., 1986. — 367 с.

# ПРИЛОЖЕНИЕ

# Приложение А. Руководство пользователя.

При загрузке приложения, пользователь должен задать необходимые данные в соответствии с потребностями в вычислениях (правую границу интегрирования, начальную точку и частоту дискретизации), после чего приложение обработает значения и выведет на экран графики, а в соответствующее текстовое поле глобальную погрешность, рассчитанные по полученным параметрам.

Более подробно рассмотреть графики можно, используя мышь:

* Выделение области графика левой кнопкой мыши отобразит выбранную область в большем масштабе;
* При нажатии средней кнопки мыши(колёсика) можно передвинуть область отображения графика;
* Для увеличения и уменьшения масштаба используется колёсико мыши (от себя и на себя соответственно);
* При нажатии правой кнопкой мыши появляется контекстное меню с возможностью сброса и установки по умолчанию масштабирования графика, отмены последнего масштабирования области, а также сохранения, печати либо копирования графика в виде картинки.

# Приложение Б. Листинг программы.

**Form1.Designer.cs:**  
namespace VM\_Kurs

{

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.components = new System.ComponentModel.Container();

this.zedGraphControl1 = new ZedGraph.ZedGraphControl();

this.textBoxx0 = new System.Windows.Forms.TextBox();

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

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

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

this.zedGraphControl2 = new ZedGraph.ZedGraphControl();

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

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

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

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

this.label5 = new System.Windows.Forms.Label();

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

this.label6 = new System.Windows.Forms.Label();

this.SuspendLayout();

//

// zedGraphControl1

//

this.zedGraphControl1.Location = new System.Drawing.Point(12, 12);

this.zedGraphControl1.Name = "zedGraphControl1";

this.zedGraphControl1.ScrollGrace = 0D;

this.zedGraphControl1.ScrollMaxX = 0D;

this.zedGraphControl1.ScrollMaxY = 0D;

this.zedGraphControl1.ScrollMaxY2 = 0D;

this.zedGraphControl1.ScrollMinX = 0D;

this.zedGraphControl1.ScrollMinY = 0D;

this.zedGraphControl1.ScrollMinY2 = 0D;

this.zedGraphControl1.Size = new System.Drawing.Size(375, 360);

this.zedGraphControl1.TabIndex = 0;

this.zedGraphControl1.UseExtendedPrintDialog = true;

//

// textBoxx0

//

this.textBoxx0.Location = new System.Drawing.Point(42, 416);

this.textBoxx0.Name = "textBoxx0";

this.textBoxx0.Size = new System.Drawing.Size(50, 20);

this.textBoxx0.TabIndex = 1;

this.textBoxx0.Text = "0";

//

// label1

//

this.label1.AutoSize = true;

this.label1.Location = new System.Drawing.Point(12, 419);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(24, 13);

this.label1.TabIndex = 2;

this.label1.Text = "x(0)";

//

// label3

//

this.label3.AutoSize = true;

this.label3.Location = new System.Drawing.Point(22, 445);

this.label3.Name = "label3";

this.label3.Size = new System.Drawing.Size(14, 13);

this.label3.TabIndex = 6;

this.label3.Text = "T";

//

// textBoxT

//

this.textBoxT.Location = new System.Drawing.Point(42, 442);

this.textBoxT.Name = "textBoxT";

this.textBoxT.Size = new System.Drawing.Size(50, 20);

this.textBoxT.TabIndex = 5;

this.textBoxT.Text = "10";

//

// zedGraphControl2

//

this.zedGraphControl2.Location = new System.Drawing.Point(393, 12);

this.zedGraphControl2.Name = "zedGraphControl2";

this.zedGraphControl2.ScrollGrace = 0D;

this.zedGraphControl2.ScrollMaxX = 0D;

this.zedGraphControl2.ScrollMaxY = 0D;

this.zedGraphControl2.ScrollMaxY2 = 0D;

this.zedGraphControl2.ScrollMinX = 0D;

this.zedGraphControl2.ScrollMinY = 0D;

this.zedGraphControl2.ScrollMinY2 = 0D;

this.zedGraphControl2.Size = new System.Drawing.Size(375, 360);

this.zedGraphControl2.TabIndex = 7;

this.zedGraphControl2.UseExtendedPrintDialog = true;

//

// label2

//

this.label2.AutoSize = true;

this.label2.Location = new System.Drawing.Point(12, 392);

this.label2.Name = "label2";

this.label2.Size = new System.Drawing.Size(206, 13);

this.label2.TabIndex = 8;

this.label2.Text = "f(x) = e^-cos(x) \* (sin(2 - x) / (3 + cos(3 - x))";

//

// label4

//

this.label4.AutoSize = true;

this.label4.Location = new System.Drawing.Point(22, 471);

this.label4.Name = "label4";

this.label4.Size = new System.Drawing.Size(15, 13);

this.label4.TabIndex = 10;

this.label4.Text = "N";

//

// textBoxN

//

this.textBoxN.Location = new System.Drawing.Point(42, 468);

this.textBoxN.Name = "textBoxN";

this.textBoxN.Size = new System.Drawing.Size(50, 20);

this.textBoxN.TabIndex = 9;

this.textBoxN.Text = "100";

//

// button1

//

this.button1.Location = new System.Drawing.Point(98, 471);

this.button1.Name = "button1";

this.button1.Size = new System.Drawing.Size(75, 23);

this.button1.TabIndex = 11;

this.button1.Text = "Персчитать";

this.button1.UseVisualStyleBackColor = true;

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

//

// label5

//

this.label5.AutoSize = true;

this.label5.Location = new System.Drawing.Point(98, 445);

this.label5.Name = "label5";

this.label5.Size = new System.Drawing.Size(29, 13);

this.label5.TabIndex = 12;

this.label5.Text = "[0,T]";

//

// textBoxErr

//

this.textBoxErr.Location = new System.Drawing.Point(364, 448);

this.textBoxErr.Name = "textBoxErr";

this.textBoxErr.ReadOnly = true;

this.textBoxErr.Size = new System.Drawing.Size(336, 20);

this.textBoxErr.TabIndex = 14;

//

// label6

//

this.label6.AutoSize = true;

this.label6.Location = new System.Drawing.Point(362, 419);

this.label6.Name = "label6";

this.label6.Size = new System.Drawing.Size(338, 26);

this.label6.TabIndex = 15;

this.label6.Text = "Глобальная погрешность методом рунге для 4 порядка точности\r\np = (x(h/2)-x(h))\*2^" +

"2/(2^2-1) ";

//

// Form1

//

this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

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

this.ClientSize = new System.Drawing.Size(800, 509);

this.Controls.Add(this.label6);

this.Controls.Add(this.textBoxErr);

this.Controls.Add(this.label5);

this.Controls.Add(this.button1);

this.Controls.Add(this.label4);

this.Controls.Add(this.textBoxN);

this.Controls.Add(this.label2);

this.Controls.Add(this.zedGraphControl2);

this.Controls.Add(this.label3);

this.Controls.Add(this.textBoxT);

this.Controls.Add(this.label1);

this.Controls.Add(this.textBoxx0);

this.Controls.Add(this.zedGraphControl1);

this.Name = "Form1";

this.Text = "Поборознюк Павел 6313-020302D";

this.ResumeLayout(false);

this.PerformLayout();

}

#endregion

private ZedGraph.ZedGraphControl zedGraphControl1;

private System.Windows.Forms.TextBox textBoxx0;

private System.Windows.Forms.Label label1;

private System.Windows.Forms.Label label3;

private System.Windows.Forms.TextBox textBoxT;

private ZedGraph.ZedGraphControl zedGraphControl2;

private System.Windows.Forms.Label label2;

private System.Windows.Forms.Label label4;

private System.Windows.Forms.TextBox textBoxN;

private System.Windows.Forms.Button button1;

private System.Windows.Forms.Label label5;

private System.Windows.Forms.TextBox textBoxErr;

private System.Windows.Forms.Label label6;

}

}

**Form1.cs:**  
using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using ZedGraph;

namespace VM\_Kurs

{

public partial class Form1 : Form

{

double x0;

double T = 0; //докуда

int N; //кол-во точек

double h; //шаг = T/N

double[] t;

double[] x1;

double[] x;

double lasth;

double lasth2;

public Form1()

{

InitializeComponent();

Calculate();

}

public double f(double x)

{

double fx = Math.Pow(Math.E, -Math.Cos(x)) \* Math.Sin(2.0 - x) / (3.0 + Math.Cos(3.0 - x));

return fx;

}

public double f(double x, double t)

{

double fx = Math.Pow(Math.E, -Math.Cos(x)) \* Math.Sin(2.0 - x) / (3.0 + Math.Cos(3.0 - x));

return fx;

}

public static void FillGraph(ZedGraphControl zg, double[] x, double[] y, string name)

{

GraphPane pane = zg.GraphPane;

pane.CurveList.Clear();

PointPairList list = new PointPairList();

for (int i = 0; i < x.Length; i++)

{

list.Add(x[i], y[i]);

}

LineItem curve = pane.AddCurve(name, list, Color.Blue, SymbolType.None);

zg.AxisChange();

zg.Invalidate();

}

public static void AddGraph(ZedGraphControl zg, double[] x, double[] y, string name)

{

GraphPane pane = zg.GraphPane;

PointPairList list = new PointPairList();

for (int i = 0; i < x.Length; i++)

{

list.Add(x[i], y[i]);

}

LineItem curve = pane.AddCurve(name, list, Color.Red, SymbolType.None);

zg.AxisChange();

zg.Invalidate();

}

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

{

Calculate();

}

private void Calculate()

{

try

{

T = Convert.ToDouble(textBoxT.Text);

N = Convert.ToInt32(textBoxN.Text);

x0 = Convert.ToDouble(textBoxx0.Text);

h = T / N;

t = new double[N];

x1 = new double[N];

x = new double[N];

t[0] = x0;

for (int i = 0; i < N - 1; i++)

t[i + 1] = t[i] + h;

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

x1[i] = f(t[i]);

FillGraph(zedGraphControl1, t, x1, "f(x), h");

Hemming();

FillGraph(zedGraphControl2, t, x, "F(x), h");

lasth = x[N - 1];

N \*= 2;

h = T / N;

t = new double[N];

x1 = new double[N];

x = new double[N];

t[0] = x0;

for (int i = 0; i < N - 1; i++)

t[i + 1] = t[i] + h;

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

x1[i] = f(t[i]);

AddGraph(zedGraphControl1, t, x1, "f(x), h/2");

Hemming();

AddGraph(zedGraphControl2, t, x, "F(x), h/2");

lasth2 = x[N - 1];

textBoxErr.Text = (Math.Abs((lasth2 - lasth) \* 4.0 / 3.0)).ToString();

}

catch (System.Exception s)

{

textBoxT.Text = "10,0";

textBoxN.Text = "100";

textBoxx0.Text = "0,0";

textBoxErr.Text = "Ошибка: в полях должны быть записаны числа соотв. им типа";

}

}

public double GetK1(double x, double t){

return h \* f(x, t);

}

public double GetK2(double x, double t){

return h \* f(x + GetK1(x, t) / 2.0, t + h / 2.0);

}

public double GetK3(double x, double t){

return f(x + GetK2(x, t) / 2.0, t + h / 2.0);

}

public double GetK4(double x, double t){

return f(x + GetK3(x, t), t + h);

}

public void Hemming()

{

double[] pred = new double[N];

double[] d = new double[N];

double[] corr = new double[N];

x[0] = x0;

x[1] = x[0] + 1.0 / 6.0 \* (GetK1(x[0], t[0]) + 2.0 \* GetK2(x[0], t[0]) + 2.0 \* GetK3(x[0], t[0]) + GetK4(x[0], t[0]));

x[2] = x[1] + 1.0 / 6.0 \* (GetK1(x[1], t[1]) + 2.0 \* GetK2(x[1], t[1]) + 2.0 \* GetK3(x[1], t[1]) + GetK4(x[1], t[1]));

//прогноз

for (int i = 3; i < N - 1; i++)

{

pred[i + 1] = ( x[i - 3] + 4.0 / 3.0 \* h \* (2.0 \* x1[i] - x1[i - 1] + 2.0 \* x1[i - 2]));

// pred[i + 1] = (2.0 \* x[i] - 1.0 + x[i - 2]) / 3.0 + h \* (191.0\*x1[i] - 107.0\*x1[i - 1] + 109.0\*x1[i - 2] - 25.0\*x1[i - 3])/ 72.0;

/\*//поправка

d[i + 1] = (2.0 \* x[i] - 1.0 + x[i - 2]) / 3.0 + h \* (191.0 \* x1[i] - 107.0 \* x1[i - 1] + 109.0 \* x1[i - 2] - 25.0 \* x1[i - 3]) / 72.0;

//коррекция

corr[i+1] = (2.0\*x1[i-1] + x1[i-2])/3.0 + h\*(25.0\*f(x[i+1],d[i+1]) + 91.0\*x1[i] + 43.0\*x1[i-1] +9.0\*x1[i-2])/72.0;\*/

x[i + 1] = 1.0 / 8.0 \* (9.0 \* x[i] - x[i - 2] + 3.0 \* h \* (x1[i + 1] + 2.0 \* x1[i] - x1[i - 1]));

//вычисление

/\* x[i + 1] = corr[i + 1] + (43.0 / 750.0) \* (d[i + 1] - corr[i + 1]);}\*/

}

}

}

}