МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Кубанский государственный университет»

Факультет компьютерных технологий и прикладной математики

Кафедра математического моделирования

**Отчет о научно-исследовательской работе**

(практике по получению первичных навыков научно-исследовательской работы)

Выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ П.А. Зубков

Направление подготовки 02.03.02 Фундаментальная информатика и информационные технологии

Курс 2

Руководитель учебной практики

канд. физ.-мат. наук, доцент кафедры

математического моделирования \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.Е. Рубцов

Содержание

[1 Постановка задачи 3](#_Toc70614869)

[2 Описание метода Эйлера и расчетные формулы для вычисления  4](#_Toc70614870)

[3 Аналитическое решение задачи Коши 5](#_Toc70614871)

[4 Результаты вычислений 7](#_Toc70614872)

[Список использованных источников 9](#_Toc70614873)

[Приложение Текст программы 10](#_Toc70614874)

**1 Постановка задачи**

Дано дифференциальное уравнение

.

Методом ломанных Эйлера [2] получить приближенное решение задачи Коши для заданного дифференциального уравнения. Начальное условие , . Последовательность  строится следующим образом , .

Вычисления произвести при помощи программы, разработанной на языке высокого уровня для различных значений  (например, при ).

В программе предусмотреть ввод .

Получить аналитически точное решение задачи Коши.

В одной системе координат построить графики точного и приближенного решений. Вычислить максимальную невязку (наибольшую по абсолютной величине разность между точным и приближенным решениями для различных значений ).

Для построения графиков использовать графические возможности выбранного языка программирования.

**2 Описание метода Эйлера и расчетные формулы для вычисления **

Пусть дана задача Коши для уравнения первого порядка [1]:

|  |  |
| --- | --- |
| , |  |
| , |  |

где функция  определена на некоторой области . Решение ищется на интервале . На этом интервале введем узлы: . Приближенное решение в узлах , которое обозначим через , определяется по формуле:

|  |  |
| --- | --- |
| , .  Для задачи Коши  приближенное решение будет рассчитываться по итерационной формуле: |  |

**3 Аналитическое решение задачи Коши**

Изначальное уравнение относится к типу Бернулли, то есть первым шагом избавляемся от справа, умножив все уравнение на :

|  |  |
| --- | --- |
|  |  |

Получим:

|  |  |
| --- | --- |
|  | (1) |

В (1) сделаем замену переменных:

|  |  |
| --- | --- |
|  |  |

После приведенной замены имеем:

|  |  |
| --- | --- |
|  |  |
|  | (2) |
| Полученное уравнение является ЛНДУ 1-го порядка , где . Сначала решим соответствующее ЛОДУ: |  |
| . |  |
|  |  |
|  |  |
|  | (3) |
|  |  |
| Найдем частное решение:  ⇒ |  |
| Подставив в уравнение (2) и воспользовавшись правилами дифференцирования произведения и производной сложной функции, получим: |  |
|  |  |
| Подставляя в (3) и возвращаясь к замене , получим: |  |
|  |  |
| Найдем с помощью использования начальных значений : |  |
|  |  |
| Таким образом, решение задачи Коши представимо в виде: |  |
|  |  |

**4 Результаты вычислений**

Графики точного и приближенных решений, получены из программы, написанной на языке C#. Программа приведена в [приложении]. Для различных значений  можно увидеть соответствующие графики на рисунках 1–3. Для построения графиков использовались графические возможности платформы .Net Framework

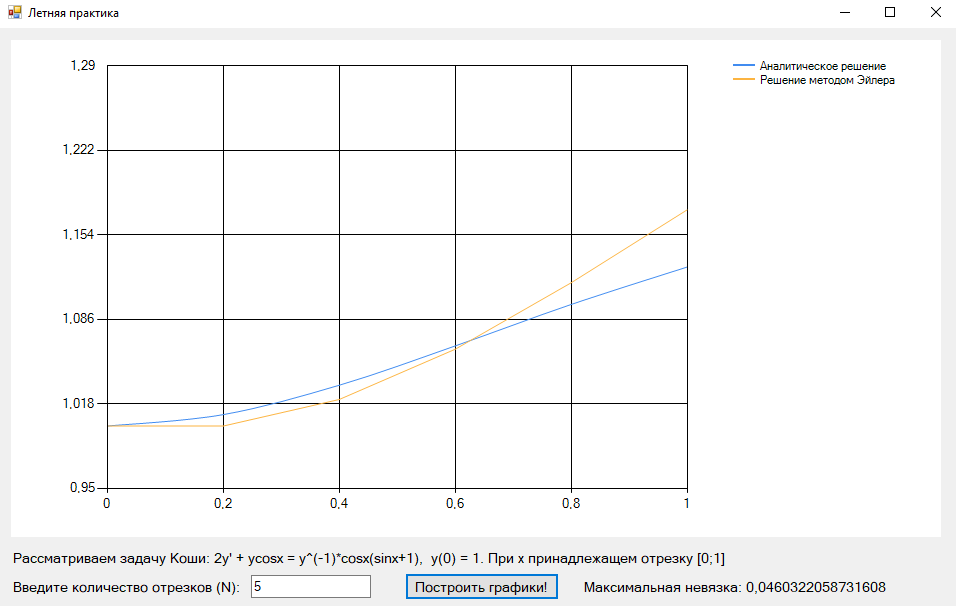


Рисунок 1 – Графики точного и приближенных решений, полученных для значения

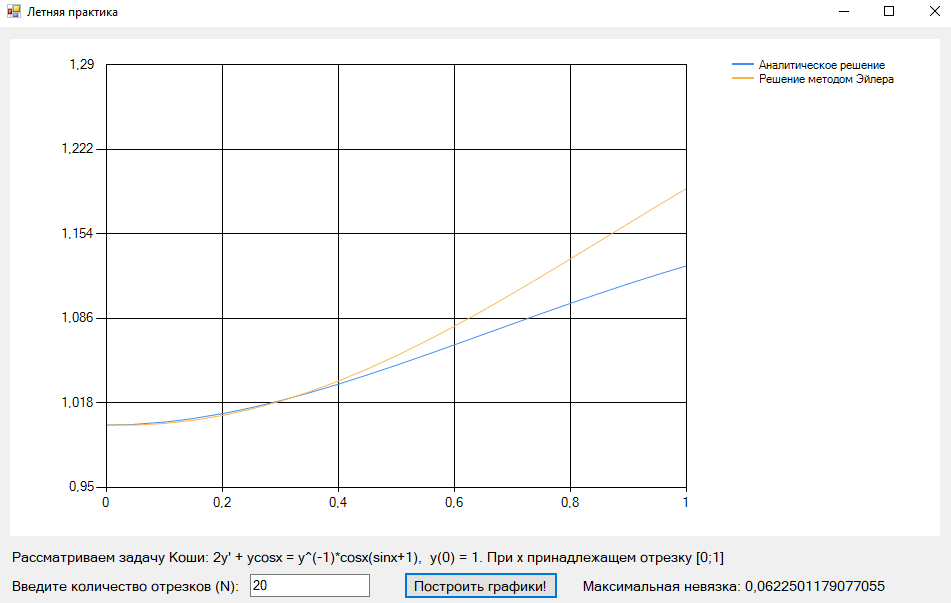


Рисунок 2 – Графики точного и приближенных решений, полученных для значения

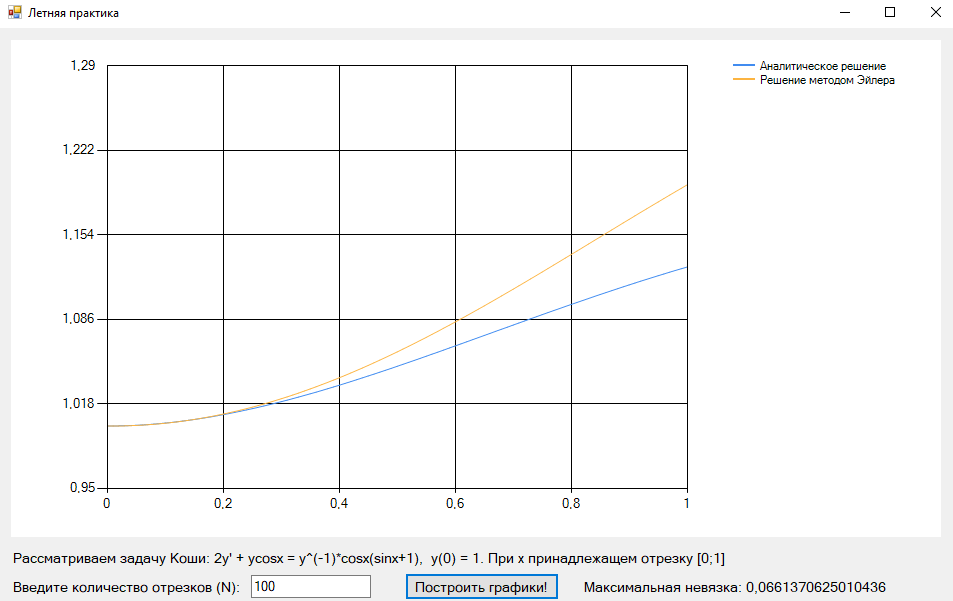


Рисунок 3 – Графики точного и приближенных решений, полученных для значения

Список использованных источников

1. Chart Класс (System.Windows.Forms.DataVisualization.Charting) | Microsoft Docs [Электронный ресурс]. – URL: https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.forms.datavisualization.charting.chart?view=netframework-4.8 (25.06.2021).
2. Дмитриев, В.И. Дифференциальные уравнения: учебник / В.И. Дмитриев – М.: Аргамак-Медиа, 2016. – 284 с. – ISBN 978-5-00024-049-6
3. Численное решение задачи Коши (дифференциального уравнения) методом Эйлера простым и модифицированным [Электронный ресурс]. – URL: http://kontromat.ru/?page\_id=5342 (25.06.2021).

Приложение

Текст программы

using System;

using System.Windows.Forms;

namespace SummerPractice2021Csharp

{

public partial class TaskCauchy : Form

{

public TaskCauchy()

{

InitializeComponent();

}

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

{

int N;

if (Int32.TryParse(textBoxN.Text, out N))

{

chart1.Series[0].Points.Clear();

chart1.Series[1].Points.Clear();

MaxNevyazka.Text = "";

//нач.условия

double x0 = 0, y0 = 1, xN = 1, step = (xN - x0) / N;

chart1.ChartAreas[0].AxisX.Minimum = x0;

chart1.ChartAreas[0].AxisX.Maximum = xN;

//Вычисляем и рисуем точки графика Эйлера

EulerMethod task = new EulerMethod(x0, y0, step);

task.Euler(N);

chart1.Series[1].Points.DataBindXY(task.X, task.Y);

//Построение графика аналит. функции

double[] xFunc = new double[N + 1];

double[] yFunc = new double[N + 1];

xFunc = task.X;

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

yFunc[i] = AnalyticsFunc(xFunc[i]);

chart1.Series[0].Points.DataBindXY(xFunc, yFunc);

MaxNev(yFunc, task.Y, N);

}

else

{

MessageBox.Show("error");

MaxNevyazka.Text = "";

chart1.Series[0].Points.Clear();

chart1.Series[1].Points.Clear();

}

}

//Функция, полученная при аналитическом решении

public double AnalyticsFunc (double pointX)

{

return Math.Sqrt(Math.Pow(Math.E, -Math.Sin(pointX)) + Math.Sin(pointX));

}

//Максимальная невязка

public void MaxNev (double[] AnalyticY, double[] EulerY, int N)

{

double maxNev = Math.Abs(AnalyticY[0] - EulerY[0]);

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

{

double t = Math.Abs(AnalyticY[i] - EulerY[i]);

if (t > maxNev)

maxNev = t;

}

MaxNevyazka.Text = " Максимальная невязка: " + Convert.ToString(maxNev);

}

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

{

}

}

public class EulerMethod

{

public double X0 { get; private set; }

public double Y0 { get; private set; }

public double Step { get; private set; } // шаг

public double[] X { get; private set; } // итоговый массив x графика

public double[] Y { get; private set; } // итоговый массив y графика

public void Euler(int N)

{

X = new double[N + 1];

Y = new double[N + 1];

X[0] = X0;

Y[0] = Y0;

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

{

X[i] = X[i - 1] + Step;

Y[i] = Y[i - 1] + Step \* DUFunction(X[i - 1], Y[i - 1]);

}

}

//Функция из исходного уравнения dy/dx = f(x,y)

public double DUFunction(double pointX, double pointY)

{

return ((1 / pointY) \* Math.Cos(pointX) \* (1 + Math.Sin(pointX) - pointY \* Math.Cos(pointX))) / 2;

}

public EulerMethod(double x0, double y0, double step)

{

this.X0 = x0;

this.Y0 = y0;

this.Step = step;

}

}

}