МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ 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 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.А. Курченко

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  3](#_Toc70614870)

[3 Аналитическое решение задачи Коши 4](#_Toc70614871)

[4 Результаты вычислений 5](#_Toc70614872)

[Список использованных источников 7](#_Toc70614873)

[Приложение А Текст программы 8](#_Toc70614874)

**1 Постановка задачи**

Дано дифференциальное уравнение , 

1. Методом ломанных Эйлера [1] 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 .

1. Получить аналитически точное решение задачи Коши.
2. 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 [2]:

|  |  |
| --- | --- |
| , | (1) |
| , | (2) |

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:

|  |  |
| --- | --- |
| , . | (3) |

**3 Аналитическое решение задачи Коши**

Проверив, что  является решением, делим обе части уравнения

|  |  |
| --- | --- |
|  | (4) |

на :

|  |  |
| --- | --- |
| . | (5) |

Замена переменных:

|  |  |
| --- | --- |
|  | (6) |

дает:

|  |  |
| --- | --- |
| , | (7) |
| . | (8) |
| . | (9) |

Делим на ,

|  |  |
| --- | --- |
| , | (10) |
|  | (11) |
|  | (12) |
| . | (13) |

Результат:

|  |  |
| --- | --- |
| , | (14) |

Найдем решение задачи Коши: 

, таким образом, решение задачи Коши 

**4 Результаты вычислений**

Графики точного и приближенных решений, полученных для различных значений  можно увидеть на рисунке 1. Для построения графиков использовались графические возможности библиотеки ZedGraph [3].

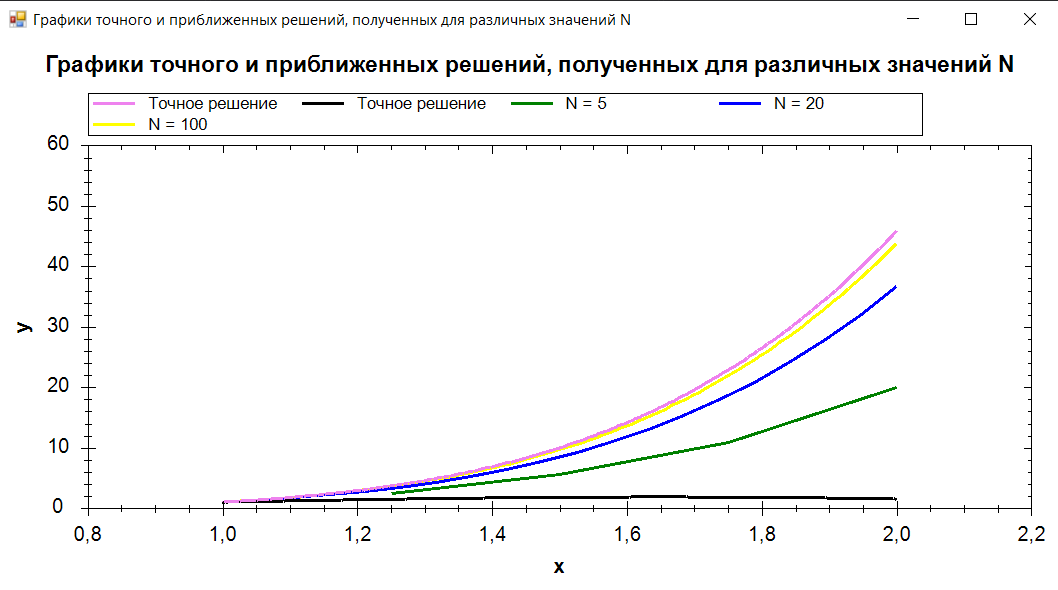


Рисунок 1 – Графики точного и приближенных решений, полученных для различных значений 

По данным графикам можно сделать вывод, что чем больше значение ‍‍, тем точнее расчет, но больше объем вычислений. Также метод Эйлера находит лишь одно из возможных решений.

На рисунке 2 представлены вычисленные максимальные невязки:

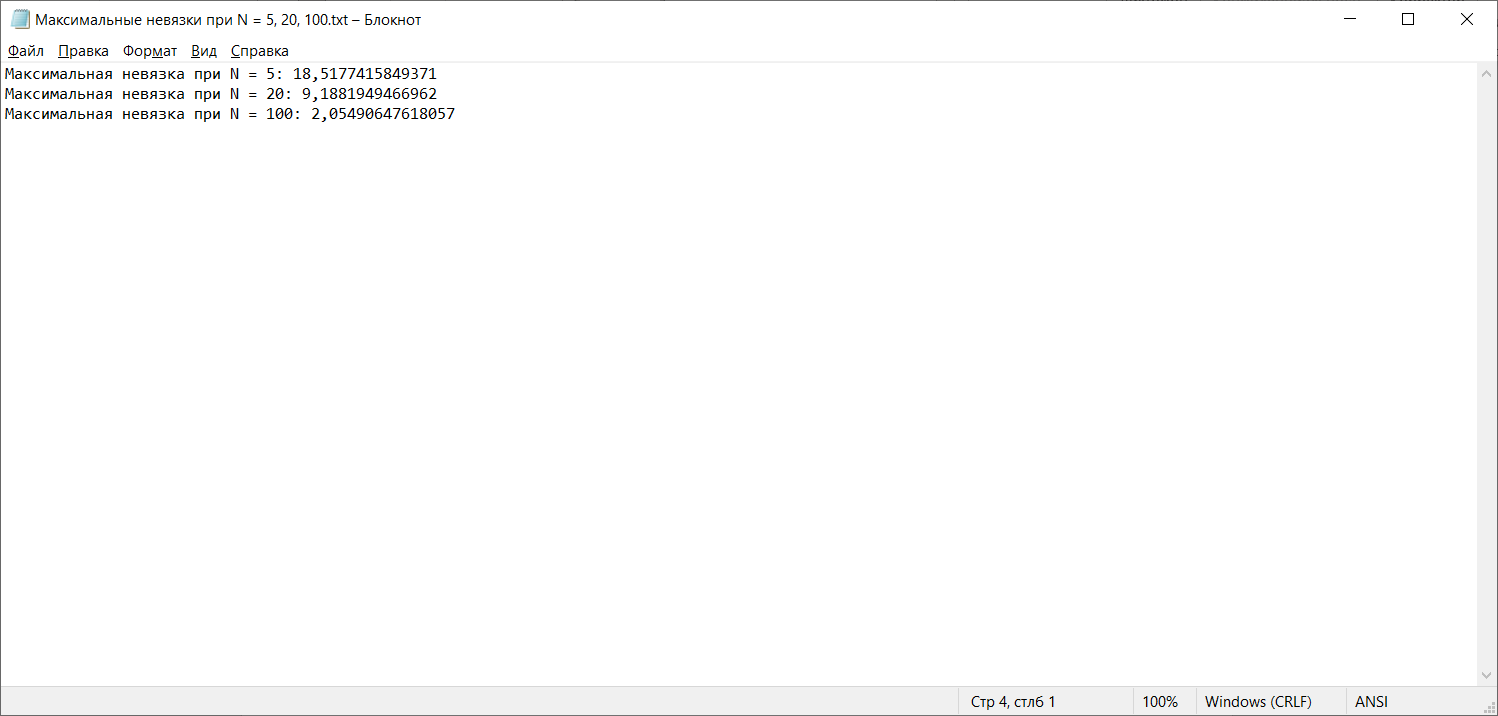


Рисунок 2 – Вычисленные максимальные невязки

По данному рисунку можно сделать вывод, что чем больше значение ‍‍, тем максимальная невязка меньше.

**Список использованных источников**

1. Петровский И. Г. Лекции по теории обыкновенных дифференциальных уравнений / Под ред. А. Д. Мышкнса, О. А. Олейник. – М.: Изд-во МГУ, 1984. – 296 с.
2. Метод Эйлера — Википедия [Электронный ресурс].-URL: <https://ru.wikipedia.org/wiki/Метод_Эйлера> (дата обращения 30 апреля 2021).
3. Шпаргалка по ZedGraph | jenyay.net [Электронный ресурс].-URL: <https://jenyay.net/Programming/ZedGraph> (дата обращения 30 апреля 2021).

Приложение А

**Текст программы**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using ZedGraph;

namespace SummerPractise2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

DrawGraph();

}

double f1(double x)

{

return Math.Pow(x, 4) \* Math.Pow(Math.Log(Math.Abs(x)) + 1, 2);

}

double f2(double x)

{

return Math.Pow(x, 4) \* Math.Pow(Math.Log(Math.Abs(x)) - 1, 2);

}

double g(double x, double y)

{

return 4 \* y / x + 2 \* x \* Math.Sqrt(y);

}

public void DrawGraph()

{

// Получим панель для рисования

GraphPane pane = zedGraph.GraphPane;

// Изменим тест надписи по оси X

pane.XAxis.Title.Text = "x";

// Изменим текст по оси Y

pane.YAxis.Title.Text = "y";

// Изменим текст заголовка графика

pane.Title.Text = "Графики точного и приближенных решений, полученных для различных значений N";

// Очистим список кривых на тот случай, если до этого сигналы уже были нарисованы

pane.CurveList.Clear();

// Создадим списки точек

PointPairList list1 = new PointPairList();

PointPairList list2 = new PointPairList();

PointPairList list3 = new PointPairList();

PointPairList list4 = new PointPairList();

PointPairList list5 = new PointPairList();

int n = 200;

double a = 1, b = 2, h = (b - a) / (n - 1), x, y;

// Заполняем список точек

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

{

x = a + i \* h;

double y1 = f1(x), y2 = f2(x);

// добавим в списки точки

list1.Add(x, y1);

list2.Add(x, y2);

}

string writePath = "Максимальные невязки.txt";

n = 5;

h = (b - a) / (n - 1);

x = a;

y = 1;

double mx1 = Math.Abs(y - f1(x)), mx2 = Math.Abs(y - f2(x));

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

{

y = y + h \* g(x, y);

x = a + i \* h;

mx1 = Math.Max(mx1, Math.Abs(y - f1(x)));

mx2 = Math.Max(mx2, Math.Abs(y - f2(x)));

// добавим в список точку

list3.Add(x, y);

}

using (StreamWriter sw = new StreamWriter(writePath, false, Encoding.Default))

{

sw.WriteLine($"Максимальная невязка при N = 5: {Math.Min(mx1, mx2)}");

}

n = 20;

h = (b - a) / (n - 1);

x = a;

y = 1;

mx1 = Math.Abs(y - f1(x));

mx2 = Math.Abs(y - f2(x));

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

{

y = y + h \* g(x, y);

x = a + i \* h;

mx1 = Math.Max(mx1, Math.Abs(y - f1(x)));

mx2 = Math.Max(mx2, Math.Abs(y - f2(x)));

// добавим в список точку

list4.Add(x, y);

}

using (StreamWriter sw = new StreamWriter(writePath, false, Encoding.Default))

{

sw.WriteLine($"Максимальная невязка при N = 20: {Math.Min(mx1, mx2)}");

}

n = 100;

h = (b - a) / (n - 1);

x = a;

y = 1;

mx1 = Math.Abs(y - f1(x));

mx2 = Math.Abs(y - f2(x));

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

{

y = y + h \* g(x, y);

x = a + i \* h;

mx1 = Math.Max(mx1, Math.Abs(y - f1(x)));

mx2 = Math.Max(mx2, Math.Abs(y - f2(x)));

// добавим в список точку

list5.Add(x, y);

}

using (StreamWriter sw = new StreamWriter(writePath, false, Encoding.Default))

{

sw.WriteLine($"Максимальная невязка при N = 100: {Math.Min(mx1, mx2)}");

}

// Создадим кривую с названием "Точное решение",

// которая будет рисоваться фиолетовым цветом (Color.Violet),

// Опорные точки выделяться не будут (SymbolType.None)

LineItem f1\_curve = pane.AddCurve("Точное решение", list1, Color.Violet, SymbolType.None);

// Создадим кривую с названием "Точное решение",

// которая будет рисоваться черным цветом (Color.Black),

// Опорные точки выделяться не будут (SymbolType.None)

LineItem f2\_curve = pane.AddCurve("Точное решение", list2, Color.Black, SymbolType.None);

// Создадим кривую с названием "N = 5",

// которая будет рисоваться зеленым цветом (Color.Green),

// Опорные точки выделяться не будут (SymbolType.None)

LineItem f3\_curve = pane.AddCurve("N = 5", list3, Color.Green, SymbolType.None);

// Создадим кривую с названием "N = 20",

// которая будет рисоваться синим цветом (Color.Blue),

// Опорные точки выделяться не будут (SymbolType.None)

LineItem f4\_curve = pane.AddCurve("N = 20", list4, Color.Blue, SymbolType.None);

// Создадим кривую с названием "N = 100",

// которая будет рисоваться желтым цветом (Color.Yellow),

// Опорные точки выделяться не будут (SymbolType.None)

LineItem f5\_curve = pane.AddCurve("N = 100", list5, Color.Yellow, SymbolType.None);

// Вызываем метод AxisChange (), чтобы обновить данные об осях.

// В противном случае на рисунке будет показана только часть графика,

// которая умещается в интервалы по осям, установленные по умолчанию

zedGraph.AxisChange();

// Обновляем график

zedGraph.Invalidate();

}

}

}