Министерство науки И ВЫСШЕГО ОБРАЗОВАНИЯ Российской Федерации

Федеральное государственное бюджетное образовательное учреждение   
высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**  
**Факультет компьютерных технологий и прикладной математики  
Кафедра математического моделирования**

**ОТЧЁТ О ПРОХОЖДЕНИИ УЧЕБНОЙ ПРАКТИКИ**(практика по получению первичных профессиональных умений и навыков)

Работу выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Д.И.Потапенко

Направление подготовки 01.03.02 Прикладная математика и информатика

Курс 2

Руководитель учебной практики

канд. физ.-мат. наук, доцент кафедры

математического моделирования \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_С.Е.уебцов

Краснодар

2023 г.

**СОДЕРЖАНИЕ**

[1 Постановка задачи 3](#_Toc131892989)

[2 Описание метода Эйлера 4](#_Toc131892990)

[3 Аналитическое решение задачи Коши 5](#_Toc131892991)

[4 Описание программы 7](#_Toc131892992)

[5 Результаты вычислений 8](#_Toc131892993)

[Список использованных источников 11](#_Toc131892994)

[Приложение А Текст программы 12](#_Toc131892995)

**1 Постановка задачи**

Решить задачу Коши аналитическим методом и методом Эйлера и построить графики решений в одной системе координат вычислить максимальную невязку между решениями:

Для решения задачи напишем программу на языке C++ (3.10) с использованием программы Excel для построения графика.

**2 Описание метода Эйлера**

Метод Эйлера – простейший численный метод решения систем обыкновенных дифференциальных уравнений. Впервые описан Леонардом Эйлером в 1768 году в работе «Интегральное исчисление». Метод Эйлера является явным, одношаговым методом первого порядка точности. Он основан на аппроксимации интегральной кривой кусочно-линейной функцией – так называемой ломаной Эйлера.

Пусть дана задача Коши:

где функция – определена на некоторой области *.* Решение ищется на полуинтервале . На этом же промежутке введем узлы . Приближенное решение в узлах , которое обозначим , определяется по формуле:

В программе X будет представлять массив чисел от X0 до X1 (), разбитый на N элементов.

В нашем конкретном случае программа будет считать решение по методу Эйлера для уравнения:

**3 Аналитическое решение задачи Коши**

Решение задачи Коши:

Поделим все на :

Вводим замену и применяем:

Рассмотрим линейное уравнение:

Получаем:

Теперь рассмотрим первоначальное уравнение от при условии :

Решим полученное уравнение:

Обратная замена:

Применим начальное условие:

Получаем решение ДУ:

**4 Описание программы**

Программа содержит следующие переменные:

double x0 = 1.0, y0 = 1.0; //начальное условие y(x0)=y0 и начальное значение x0

double xn = 2.0; //конечное значение xn

double disc = 0;//максимальная невязка

double k1 = dx \* df(x0, y0); //значение функции при вычислении методом Эйлера

double k2 = f(x0) \* f(x0); //аналитическое значение функции

int n;//разбиение

double dx = (xe - x0) / n; //определение кол-ва шагов метода

Основные параметры программы указаны в константах, где n – число элементов разбиения, x0 и xn берутся из области определения x, y0 – условие Коши.

//исходная функция

double df(double x, double y) {

return (4 \* y + 2 \* x \* x \* sqrt(y)) / x;

}

//аналитическая функция

double f(double x) {

return (x \* x \* log(x) + x \* x);

}

Также необходимо заполнить функцию – решение, которую мы получаем аналитически и исходную функцию равную , согласно методу Эйлера.

Программа возвращает максимальную невязку и график, содержащий оба решения. Также в ходе выполнения программы заполняются файлы resY.txt, resX.txt, в которых хранятся соответственно значения для решения методом Эйлера и координаты, для которых считались значения. Далее выполним загрузку полученных данных в Excel: на вкладке “Данные” выберем “Получить данные”, далее “Из файла”, “Из текстового/CSV-файла”, выбираем файл “resX.txt” и нажимаем “Импорт”, “Загрузить”. Проделываем аналогичные действия для “resY.txt”.

Важно заметить, что более старые версии Microsoft Excel (нами используется версия 2016 года, для которой данное замечание нельзя проигнорировать) используют по умолчанию запятую в качестве разделителя между целой и дробной частью десятичной дроби, из-за чего десятичные дроби с разделителем в виде точки будут интерпретированы как дата. Чтобы избежать эту ошибку, необходимо перейти во вкладку “Файл”, далее “Параметры”. В появившемся окне нужно перейти в меню “Дополнительно” и снять галочку “Использовать системные разделители”, ниже в поле “Разделитель целой и дробной части” указать точку вместо запятой, затем нажать “Ок”.

Воспользуемся средствами Microsoft Excel для визуализации данных: будем использовать точечную диаграмму с прямыми отрезками. Сделаем 3 графика для значений n = 10, n = 50, n = 100, нанеся на каждый по 2 кривые для наборов точек x и y, полученных в результате численного и аналитического решений задачи Коши. Приведем их ниже.

**5 Результаты вычислений**

На рисунках 1, 2 представлены решения при n = 10:

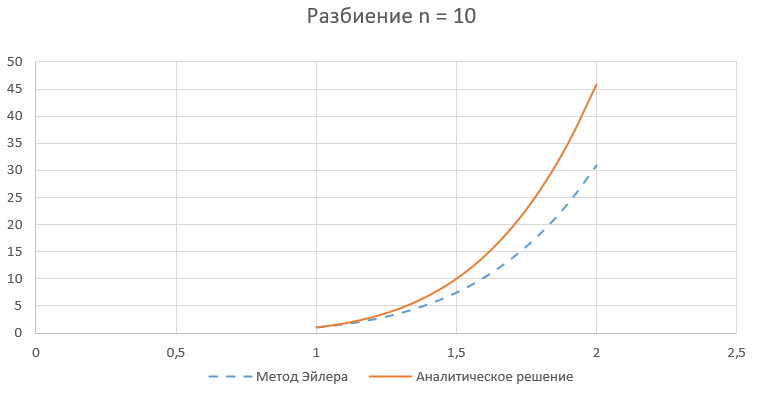


Рисунок 1 – Графики при n = 10

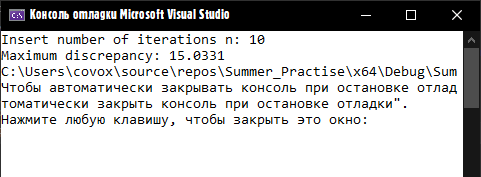


Рисунок 2 – максимальная невязка при n = 10

На рисунках 3, 4 представлены решения при n = 50:

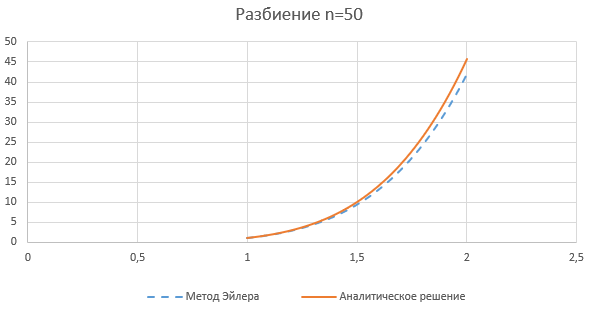
****

Рисунок 3 – Графики при n = 50

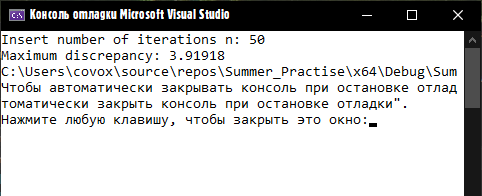


Рисунок 4 – максимальная невязка при n = 50

На рисунках 5, 6 представлены решения при n = 100:

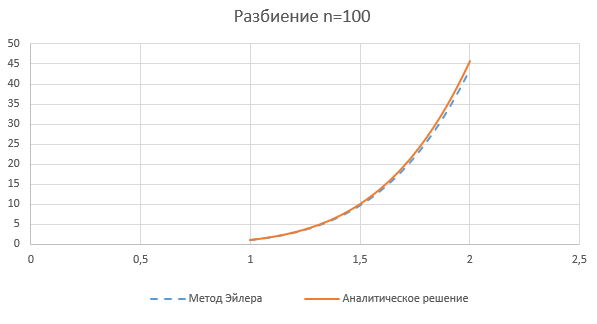
****

Рисунок 5 – Графики при n = 100

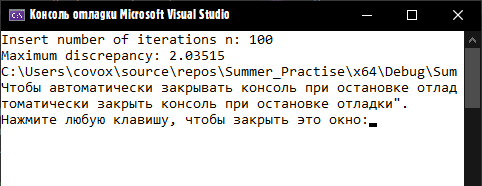


Рисунок 6 – максимальная невязка при n = 100

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Филиппов, А.Ф. Сборник задач по дифференциальным уравнениям / А. Ф. Филиппов – Ижевск: НИЦ «Регулярная и хаотичная динамика», 2000. – 176 стр. – ISBN 5-93972-008-0
2. Документация С++: [Электронный ресурс] – URL: https://learn.microsoft.com/ru-ru/cpp/standard-library/fstream?view=msvc-170 (19.05.2023)
3. Александер Майкл, Куслейка Ричард. Excel 2019. Библия пользователя / Майкл Александер, Ричард Куслейка; [Пер. с англ.] – СПб.: ООО "Диалектика",2019. — 1136 с. — ISBN 978-5-907144-44-6.

# **ПРИЛОЖЕНИЕ А Текст программы**

#include <iostream>

#include <cmath>

#include <fstream>

using namespace std;

//исходная функция

double df(double x, double y) {

return (4 \* y + 2 \* x \* x \* sqrt(y)) / x;

}

//аналитическая функция

double f(double x) {

return (x \* x \* log(x) + x \* x);

}

int main() {

ofstream fout1("resX.txt");

ofstream fout2("resY.txt");

double x0 = 1.0, y0 = 1.0; //начальное условие y(x0)=y0

double xn = 2.0; //до какого x считаем по методу

double disc = 0;//максимальная невязка

int n;//разбиение

cout << "Insert number of iterations n: ";

cin >> n;

double dx = (xn - x0) / n; //определение кол-ва шагов метода

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

{

double k1 = dx \* df(x0, y0); //значение функции при вычислении методом Эйлера

double k2 = f(x0) \* f(x0); //аналитическое значение функции

y0 += k1; //y+dy

x0 += dx; //x+dx

fout1 << x0 << "\n";

fout2 << y0 << "\n";

disc = abs(k2 - y0) > disc ? abs(k2 - y0) : disc;

}

fout1.close();

fout2.close();

cout << "Maximum discrepancy: " << disc;

return 0;

}