МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(МОСКОВСКИЙ ПОЛИТЕХ)**

09.03.01 Информатика и вычислительная техника

Образовательная программа (профиль)

«Интеграция и программирование в САПР»

Кафедра «СМАРТ технологии»

ОТЧЕТ

по дисциплине:

**Проектная деятельность**

на тему:

ПО инженерного анализа

Преподаватель: / Толстиков А.В., к.т.н. /

*подпись ФИО, уч. звание и степень*

Студент: / Сергичева В.Н., 191-324 /

*подпись ФИО, группа*

Москва, 2021 г.

Оглавление

[ВВЕДЕНИЕ 3](#_Toc93783060)

[1 ЗАДАЧИ 4](#_Toc93783061)

[2 ИНДИВИДУАЛЬНЫЙ ПЛАН 4](#_Toc93783062)

[3 СОДЕРЖАТЕЛЬНЫЕ ГЛАВЫ 4](#_Toc93783063)

[3.1 Теоретическая часть 4](#_Toc93783064)

[3.2 Практическая часть 6](#_Toc93783065)

[4 Результаты 11](#_Toc93783066)

[ПРИЛОЖЕНИЕ 1 12](#_Toc93783067)

# **ВВЕДЕНИЕ**

Целью работы является изучение книги «Математика и САПР» и перевод кода с языка Fortran на C#. В работе представлены теоретические сведения и подробный разбор кода.

# **ЗАДАЧИ**

1. Изучение книги «Математика и САПР»
2. Поиск литературы
3. Перевод кода с Fortran на C#
4. Написание методических указаний;
5. Создание отчетности по проекту.

# **ИНДИВИДУАЛЬНЫЙ ПЛАН**

|  |  |
| --- | --- |
| **Задача** | **Время выполнения, ак. часов** |
| Встреча с командой | 5 |
| Постановка задач | 2 |
| Поиск литературы | 2 |
| Изучение книги «Математика и САПР» | 20 |
| Изучение основ языка Фортран по книге «Современный Фортран» | 20 |
| Написание кода на C# | 9 |
| Работа с отчетностью | 7 |
| Подготовка видео, итогового отчета, лендинг | 8 |
|  | **73** |

# **СОДЕРЖАТЕЛЬНЫЕ ГЛАВЫ**

## **Теоретическая часть**

Методы Ру́нге — Ку́тты — большой класс численных методов решения задачи Коши для обыкновенных дифференциальных уравнений и их систем. Первые методы данного класса были предложены около 1900 года немецкими математиками К. Рунге и М. В. Куттой.

Метод Рунге — Кутты четвёртого порядка при вычислениях с постоянным шагом интегрирования столь широко распространён, что его часто называют просто методом Рунге — Кутты.

Рассмотрим задачу Коши для системы обыкновенных дифференциальных уравнений первого порядка (рисунок 1).



Рисунок 1 – Задача Коши для системы обыкновенных дифференциальных уравнений первого порядка

Тогда приближенное значение в последующих точках вычисляется по итерационной формуле (рисунок 2):



Рисунок 2 – Итерационная формула для расчета приближенного значения в последующих точках

Вычисление нового значения проходит в четыре стадии (рисунок 3):

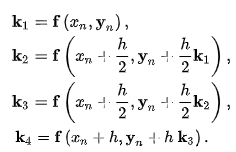


Рисунок 3 – Вычисление нового значения

где h — величина шага сетки по x.

Для решения должна использоваться А-устойчивая формула. Метод называется А-устойчивым, если, применяя его к уравнению y`= -y, получаем yn→0 при n→∞ независимо от величины h.

Для метода Рунге-Кутта четвертого порядка формула А-устойчива, если h<2.75.

## **Практическая часть**

Для примера дан код на языке Fortran (рисунок 4).

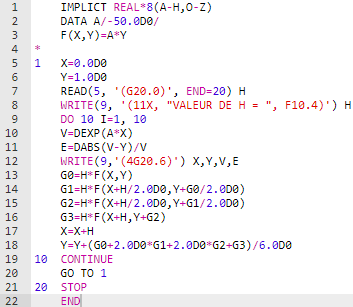


Рисунок 4 – Код программы на языке Fortran

Данная программа позволяет решить уравнение Y=A\*Y при A=-50 методом Ругне-Кутта четвертого порядка.

Будем переводить код на C# с помощью Windows Forms (.NET Framework). В строке 1 программы на языке Fortran написана инициализация возможных переменных. Код программы на C# представлен в приложении 1. В строке 3 проинициализирована константа. Инициализация переменных представлена на рисунке 5 в строках 16-20.

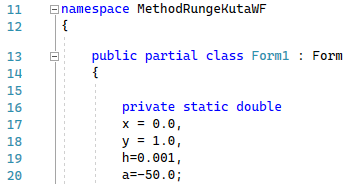


Рисунок 5 – Инициализация переменных

На форме создаем структуру окна: добавляем Lable, TextBox, Botton. В окне свойств добавляем необходимые события для элементов TextBox и Botton.

Создадим расчет максимального шага сетки, для корректных вычислений (этого нет в программе на Fortran). Событие будет обрабатываться, при нажатии на кнопку «Рассчитать максимальный шаг сетки h». Обработка события нажатия кнопки представлена на рисунке 6.

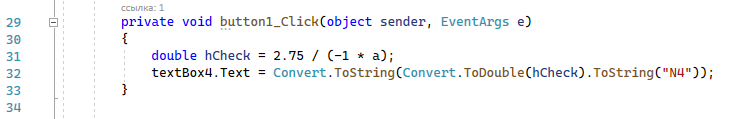


Рисунок 6 – Обработка события нажатия кнопки «Рассчитать максимальный шаг сетки h»

В строках 5 и 6 программы на языке Fortran производится присваивание значений переменным x и y. В программе на C# переменные вводятся в элементы TextBox. Также к этим элементам применена проверка для корректности ввода данных. Код реализации проверки ввода переменных x и y представлен на рисунке 7.

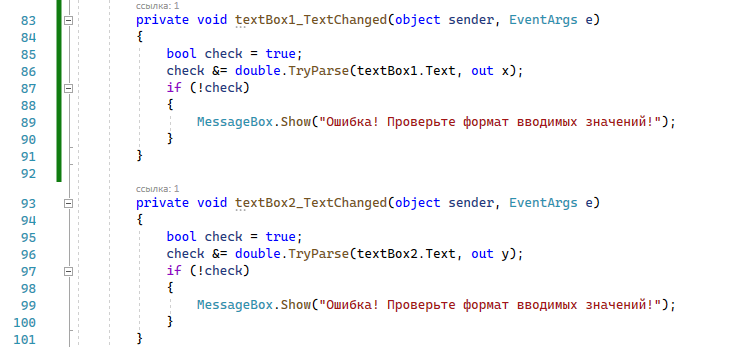


Рисунок 7 – Код реализации проверки ввода переменных x и y

В строке 7 программы на языке Fortran написано считывание переменной H. В программе на C# переменные вводятся в элементы TextBox. Также к этим элементам применена проверка для корректности ввода данных. Код реализации проверки представлен на рисунке 8.

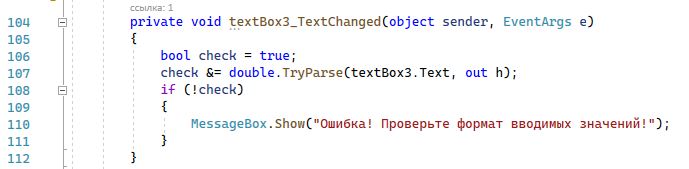


Рисунок 8 – Код реализации проверки

В строке 8 начало цикла DO, который продолжается до метки 10 (расположенной на 19 строке). Реализация данного цикла на C# расположена в событии Click кнопки «Решить уравнение» (рисунок 9).

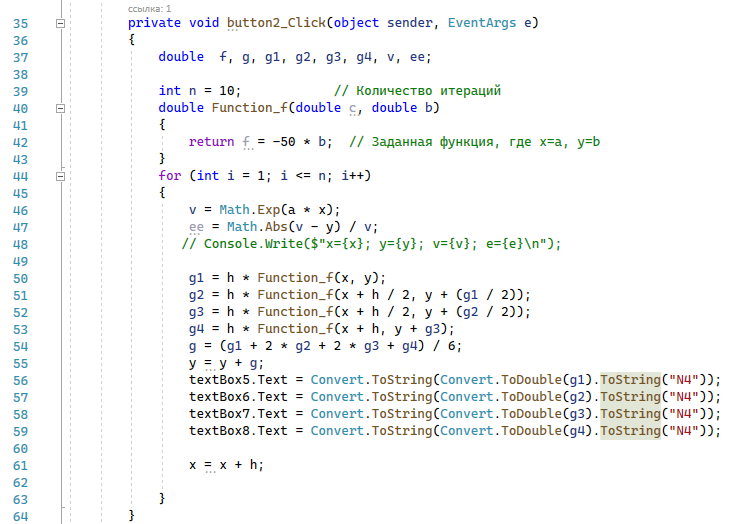


Рисунок 9 – Реализация данного цикла DO

В строке 3 программы на языке Fortran прописана функция F. Также она прописана в строках 40-43 на языке C#/

Расчет переменных v и e представлен в строках 10, 11 для Fortran и 46, 47 для C#.

В строках 13-18 программы на языке Fortran производится расчет коэффициентов. Также производится расчет коэффициентов и на C# на строках 50-55 и 61 (рисунок 9). На строках 56-59 выполняется вывод переменных в TextBox.

В строке 20 программы на языке Fortran пополняется переход на метку 1 (т.е. на строку 5. В нашем же случае для этого создана кнопка «Очистить», которая возвращает переменным изначальные параметры. Создано это для того, чтобы производился корректный расчет при вводе новых данных. Реализация события на кнопке «Очистить» представлена на рисунке 10.

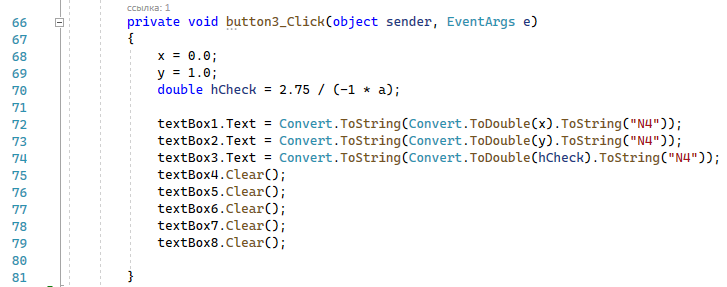


Рисунок 10 – Реализация события на кнопке «Очистить»

Пример работы программы представлен на рисунке 11.

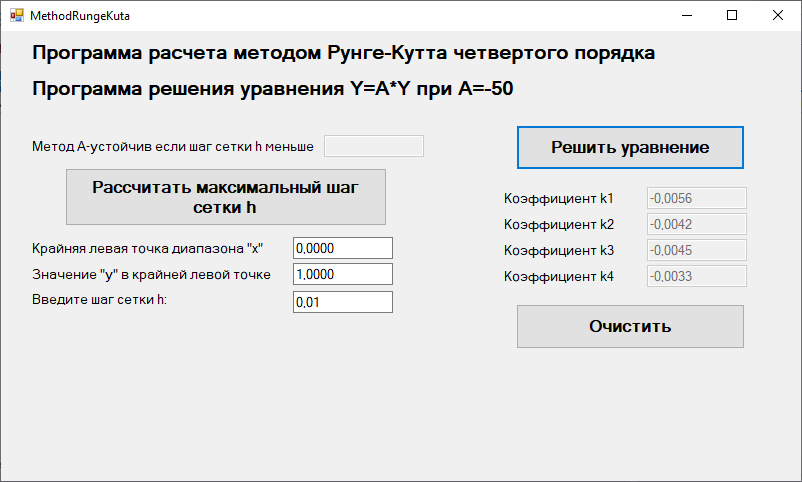


Рисунок 11 – Пример работы программы

# **Результаты**

В результате работы сделана программа по расчету уравнения методом Рунге-Кутта четвертого порядка. На экран выводятся результаты расчета, а также по кнопке считается максимальный шаг метки.

Доступ к результатам – ссылка на git-репозиторий: <https://github.com/AlekseyPanaskin/Soft-for-Engineers-2021>

# **ПРИЛОЖЕНИЕ 1**

Листинг1 – Решение уравнения методом Рунге-Кутта четвертого порядка

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;

namespace MethodRungeKutaWF

{

public partial class Form1 : Form

{

private static double

x = 0.0,

y = 1.0,

h=0.001,

a=-50.0;

public Form1()

{

InitializeComponent();

}

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

{

double hCheck = 2.75 / (-1 \* a);

textBox4.Text = Convert.ToString(Convert.ToDouble(hCheck).ToString("N4"));

}

private void button2\_Click(object sender, EventArgs e)

{

double f, g, g1, g2, g3, g4, v, ee;

int n = 10; // Количество итераций

double Function\_f(double c, double b)

{

return f = -50 \* b; // Заданная функция, где х=а, у=b

}

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

{

v = Math.Exp(a \* x);

ee = Math.Abs(v - y) / v;

// Console.Write($"x={x}; y={y}; v={v}; e={e}\n");

g1 = h \* Function\_f(x, y);

g2 = h \* Function\_f(x + h / 2, y + (g1 / 2));

g3 = h \* Function\_f(x + h / 2, y + (g2 / 2));

g4 = h \* Function\_f(x + h, y + g3);

g = (g1 + 2 \* g2 + 2 \* g3 + g4) / 6;

y = y + g;

textBox5.Text = Convert.ToString(Convert.ToDouble(g1).ToString("N4"));

textBox6.Text = Convert.ToString(Convert.ToDouble(g2).ToString("N4"));

textBox7.Text = Convert.ToString(Convert.ToDouble(g3).ToString("N4"));

textBox8.Text = Convert.ToString(Convert.ToDouble(g4).ToString("N4"));

x = x + h;

}

}

private void button3\_Click(object sender, EventArgs e)

{

x = 0.0;

y = 1.0;

double hCheck = 2.75 / (-1 \* a);

textBox1.Text = Convert.ToString(Convert.ToDouble(x).ToString("N4"));

textBox2.Text = Convert.ToString(Convert.ToDouble(y).ToString("N4"));

textBox3.Text = Convert.ToString(Convert.ToDouble(hCheck).ToString("N4"));

textBox4.Clear();

textBox5.Clear();

textBox6.Clear();

textBox7.Clear();

textBox8.Clear();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

bool check = true;

check &= double.TryParse(textBox1.Text, out x);

if (!check)

{

MessageBox.Show("Ошибка! Проверьте формат вводимых значений!");

}

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

bool check = true;

check &= double.TryParse(textBox2.Text, out y);

if (!check)

{

MessageBox.Show("Ошибка! Проверьте формат вводимых значений!");

}

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

bool check = true;

check &= double.TryParse(textBox3.Text, out h);

if (!check)

{

MessageBox.Show("Ошибка! Проверьте формат вводимых значений!");

}

}

}

}