Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии

НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ РАБОТА

НА ТЕМУ:

«Алгоритм и программная реализация одномерной линейной интерполяции сеточной функции кубическими сплайнами»

Студентка ИУ7-53Б(В) ________ **О.О.Ишкова-Запольская**
(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель НИР ________ **В.М.Градов**
(Подпись, дата) (И.О.Фамилия)

2021 г.

Оглавление

Введение

Целью научно-исследовательской работы является решение задачи одномерной линейной интерполяции сеточной функции кубическими сплайнами.

Линейная интерполяция – интерполяция алгебраическим двучленом $P_1(x) = ax+b$ функции $f(x)$, заданной в двух точках x_0 и x_1 отрезка $[a, b]$.

Кубический сплайном $P_{3,\Delta}(x)$ на сетке Δ называется функция, непрерывная на отрезке $[a, b]$ вместе со своими первой и второй производными, совпадающая с кубическим полиномом на каждом отрезке $[x_{j-1}, x_j]$, $j=1,\dots,N$ и при этом удовлетворяющая условиям $P_{3,\Delta}(x_i) = y_i$, $i=0,\dots,N$.

Для реализации программы был выбран язык программирования высокого уровня C++/CLI и среда разработки MVS 2019 с использованием фреймворка .NET. Программа предоставляет пользователю возможность выбора входных данных, выполняет расчет с ними и отображает на экране результат в графическом виде.

Цель программы – знакомство с алгоритмом интерполяции данных кубическими сплайнами, а также его программная реализация.

Описание метода

Поставленная задача представляет собой вычисление кубических сплайнов на каждом отрезке сеточной функции. Кубический сплайн – функция, область определения которой разбита на конечное число отрезков, на каждом из которых она совпадает с некоторым кубическим многочленом.

На каждом отрезке $[x_{(i-1)}, x_i]$ можно построить полином третьей степени $\psi_i(x)$ коэффициенты которого надо определить. Такой интерполяционный полином на участке между каждой парой соседних точек можно записать в виде:

$$\psi_i(x) = a_i + b_i (x - x_i) + c_i (x - x_i)^2 + d_i (x - x_i)^3, \text{ где } x_{i-1} \le x \le x_i, 0 \le i \le N$$

В узлах значения многочлена и интерполируемой функции совпадают:

$$\psi_i(x_{i-1}) = y_{i-1} = a_i,$$
$$\psi_i(x_i) = y_i = a_i + b_i h_i + c_i h_i^2 + d_i h_i^3,$$
$$h_i = x_i - x_{i-1}, \text{ где } 1 \le i \le N$$

Число таких уравнений меньше числа неизвестных в два раза. Недостающие уравнения получают, приравнивая во внутренних узлах первые и вторые производные, вычисляемые по коэффициентам на соседних участках. Положив, что вторая производная равна нулю на концах участка интерполирования, можно получить недостающие условия. Полученную систему уравнений для нахождения неизвестных коэффициентов a_i, b_i, c_i, d_i можно упростить, приведя к специальному виду. В итоге, коэффициенты будут определятся по следующим формулам:

$$a_i = y_{i-1}, \text{ где } 1 \le i \le N$$

$$b_i = \frac{y_i - y_{i-1}}{h_i} - \frac{h_i}{3}(c_{i+1} + 2c_i), \text{где } 1 \le i \le N-1$$

$$d_i = \frac{c_{i+1} - c_i}{3h_i}, \text{где } 1 \le i \le N-1$$

$$c_1 = 0,$$

$$c_n = -3d_n h_n,$$

$$h_i c_i + 2(h_i + h_{i+1})c_{i+1} + h_{i+1}c_{i+2} = 3\left(\frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i}\right), \qquad ★$$

$$\text{где } 2 \le i \le N-1$$

Полученное разностное уравнение ★ можно решить методом прогонки, который является частным случаем метода Гаусса и используется для решения систем линейных уравнений вида $Ax = B$, где A – трёхдиагональная матрица. Применение метода исключения Гаусса для решения системы уравнений с трёхдиагональной матрицей приводит к тому, что система уравнений преобразуется к виду, когда в каждом уравнении содержится только два неизвестных, и при обратном ходе одно из этих неизвестных выражается через другое. Таким образом, применительно к ★ можно записать:

$$c_i = \xi_{i+1}c_{i+1} + \eta_{i+1},$$

где ξ_{i+1}, η_{i+1} некоторые неизвестные пока прогоночные коэффициенты;

$$c_{i+1} = \xi_i c_i + \eta_i$$

Подставив последнее выражение в ★ и преобразовав, получим:

$$\xi_{i+1} = -\frac{h_i}{h_{i-1}\xi_i + 2(h_{i-1} + h_i)}, \quad \eta_{i+1} = \frac{f_i - h_{i-1}\eta_i}{h_{i-1}\xi_i + 2(h_{i-1} + h_i)}. \qquad ☆$$

В этих формулах введено обозначение:

$$f_i = 3\left(\frac{y_i - y_{i-1}}{h_i} - \frac{y_{i-1} - y_{i-2}}{h_{i-1}}\right).$$

Из условия $c_1 = 0$ следует $\xi_2 = 0, \eta_2 = 0.$

Тогда алгоритм решения ★ будет выглядеть следующим образом: сначала по формулам ☆ при известных ξ_2, η_2, равных нулю, вычисляют прогоночные коэффициенты ξ_{i+1}, η_{i+1} $(2 \le i \le N)$ (прямой ход); затем по формуле $c_i = \xi_{i+1}c_{i+1} + \eta_{i+1}$ при условии $c_{N+1} = 0$ определяют все c_i (обратный ход).

Описание и тестирование программы

Программа работает как оконное приложение Windows Forms, интерфейс которого имеет следующие области (рис. 1):

- справка о задании
- кнопка выбора и открытия пользователем файла координат с указанием пути к нему в специальном поле
- область для вывода содержимого файла
- кнопки выполнения расчета и очистки всех полей
- область демонстрации результата в виде графиков

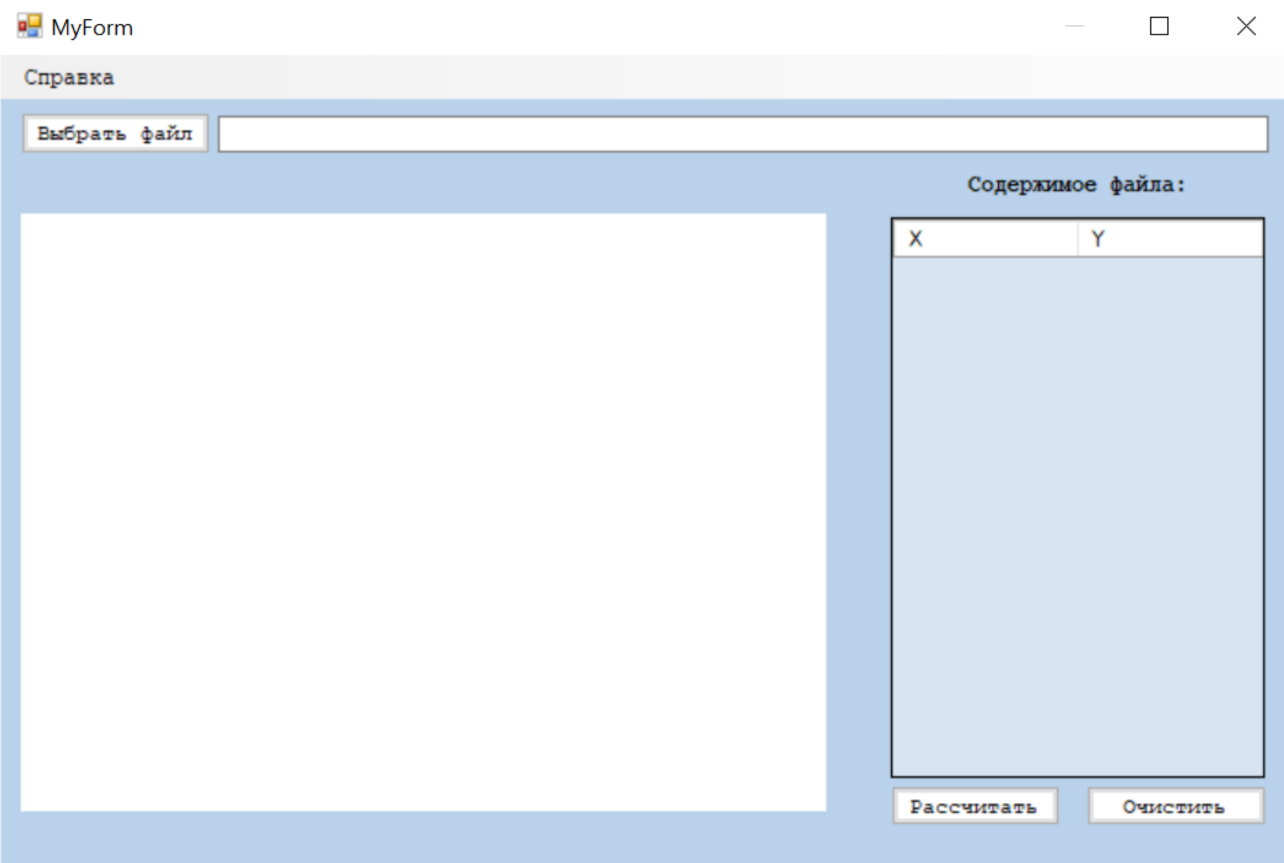

Рис.1

Пример работы программы представлен ниже на двух функциях (рис.2-7).

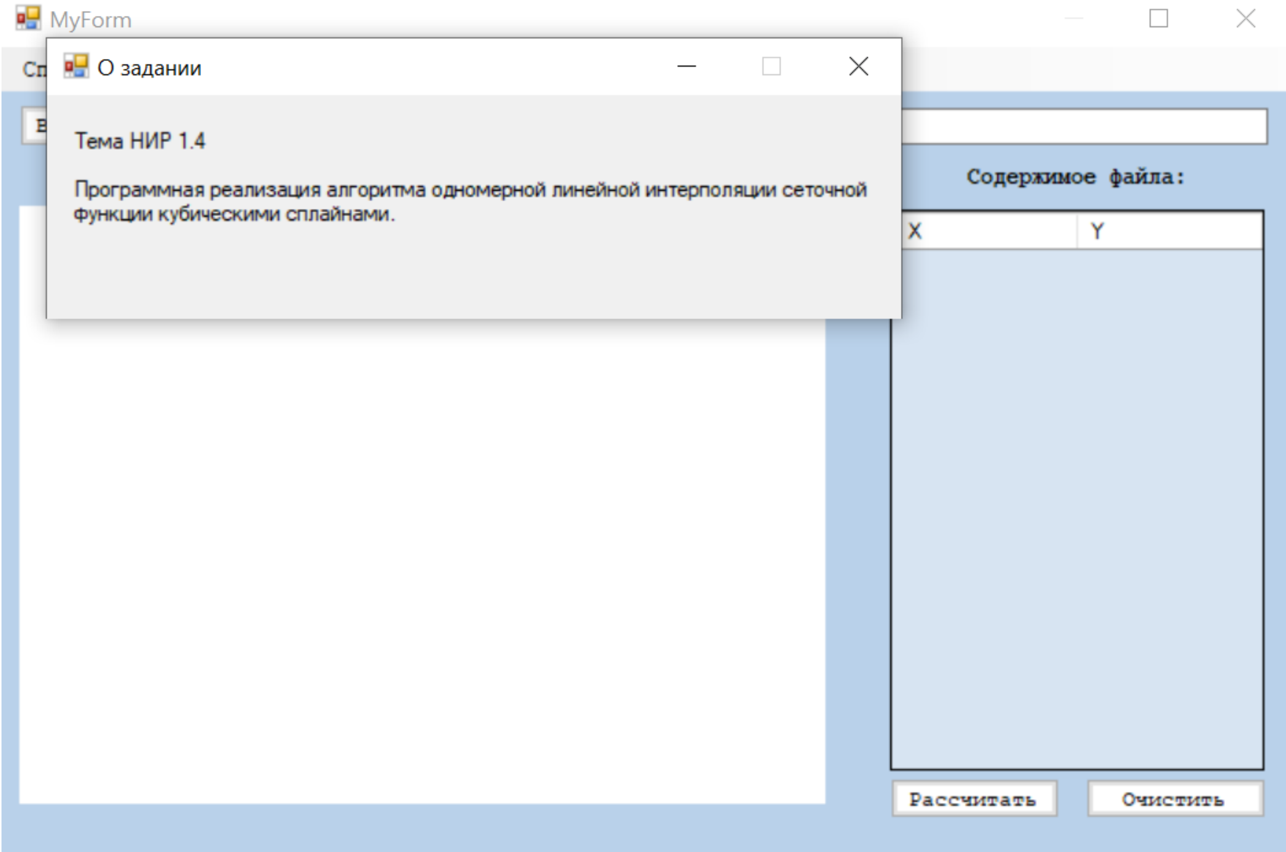

Рис. 2 – Справка о задании

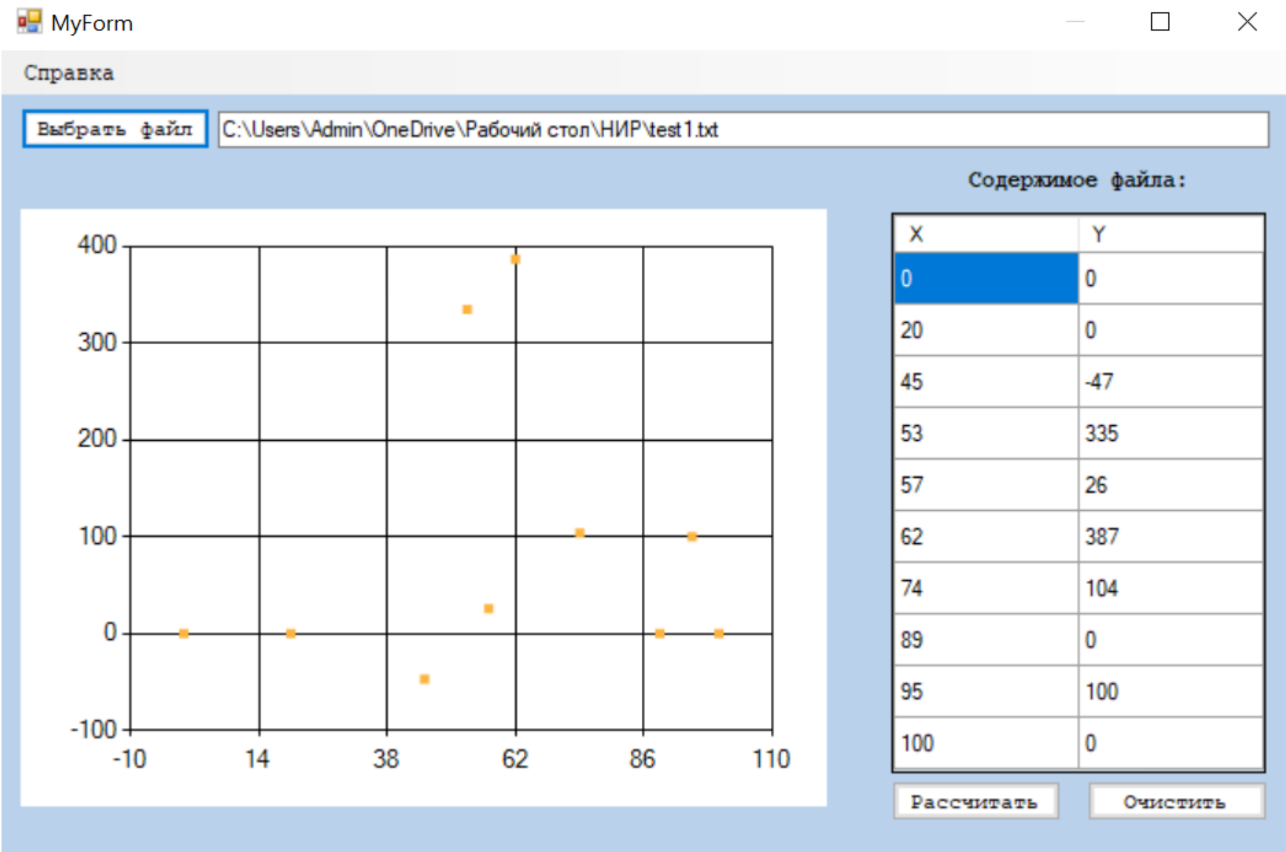

Рис. 3 – Пример расчёта первой функции по координатам

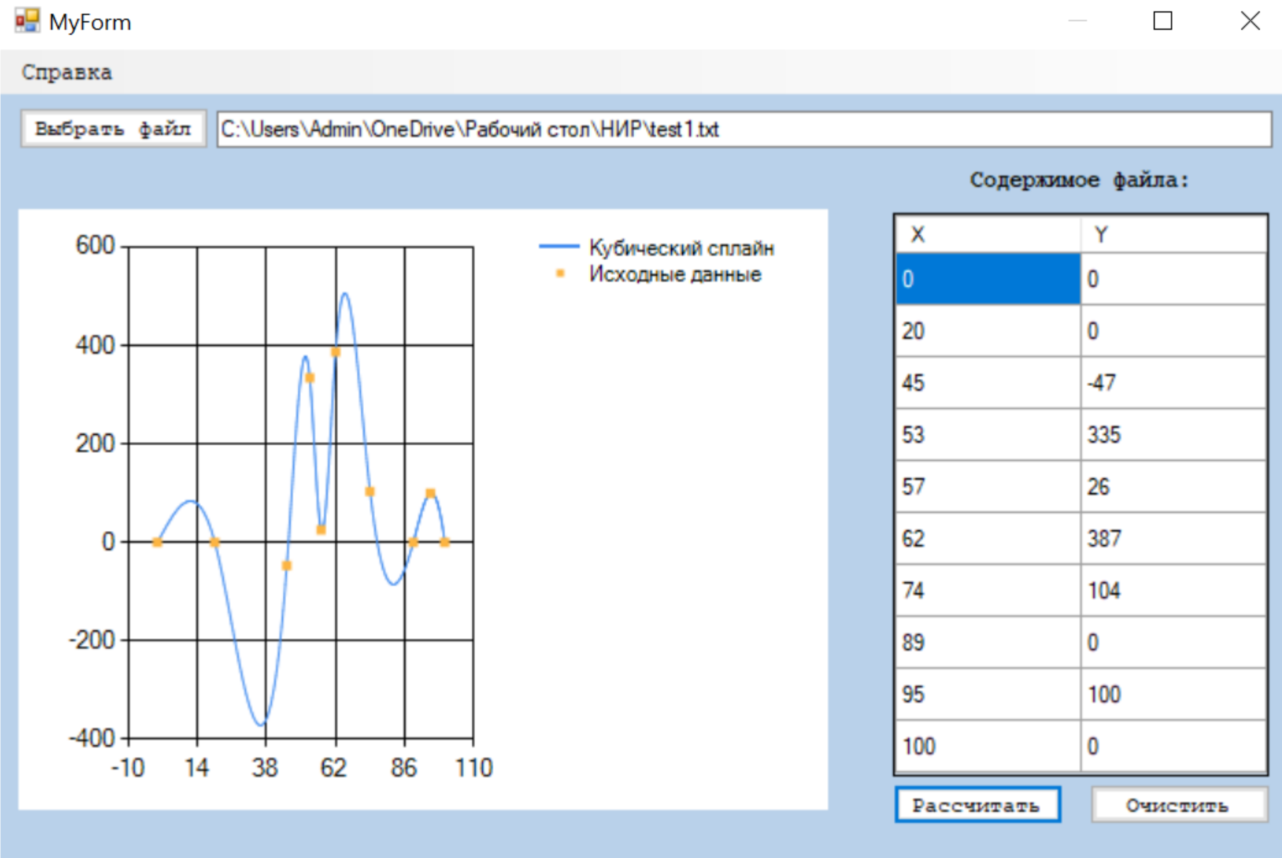

Рис. 4 – Пример расчёта первой функции по координатам

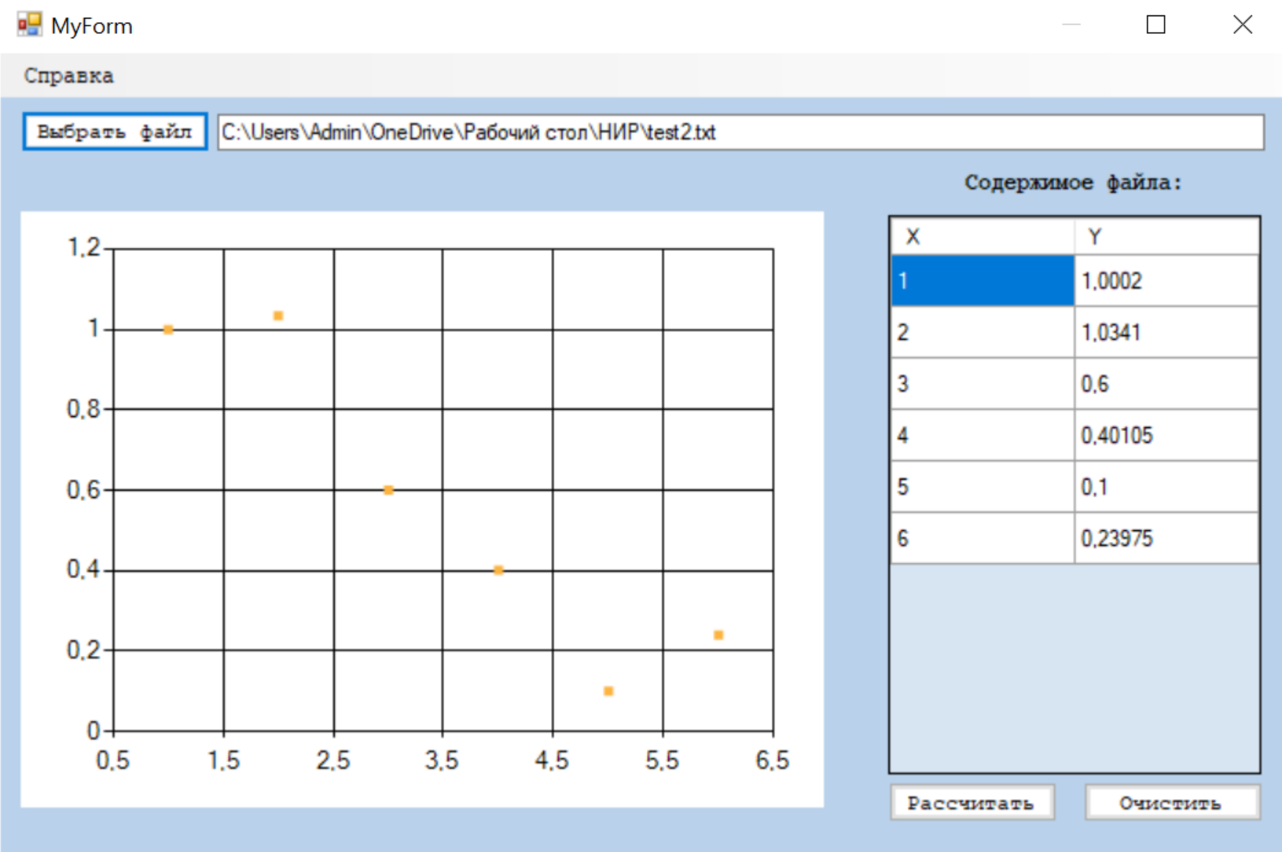

Рис. 5 – Пример расчёта второй функции по координатам

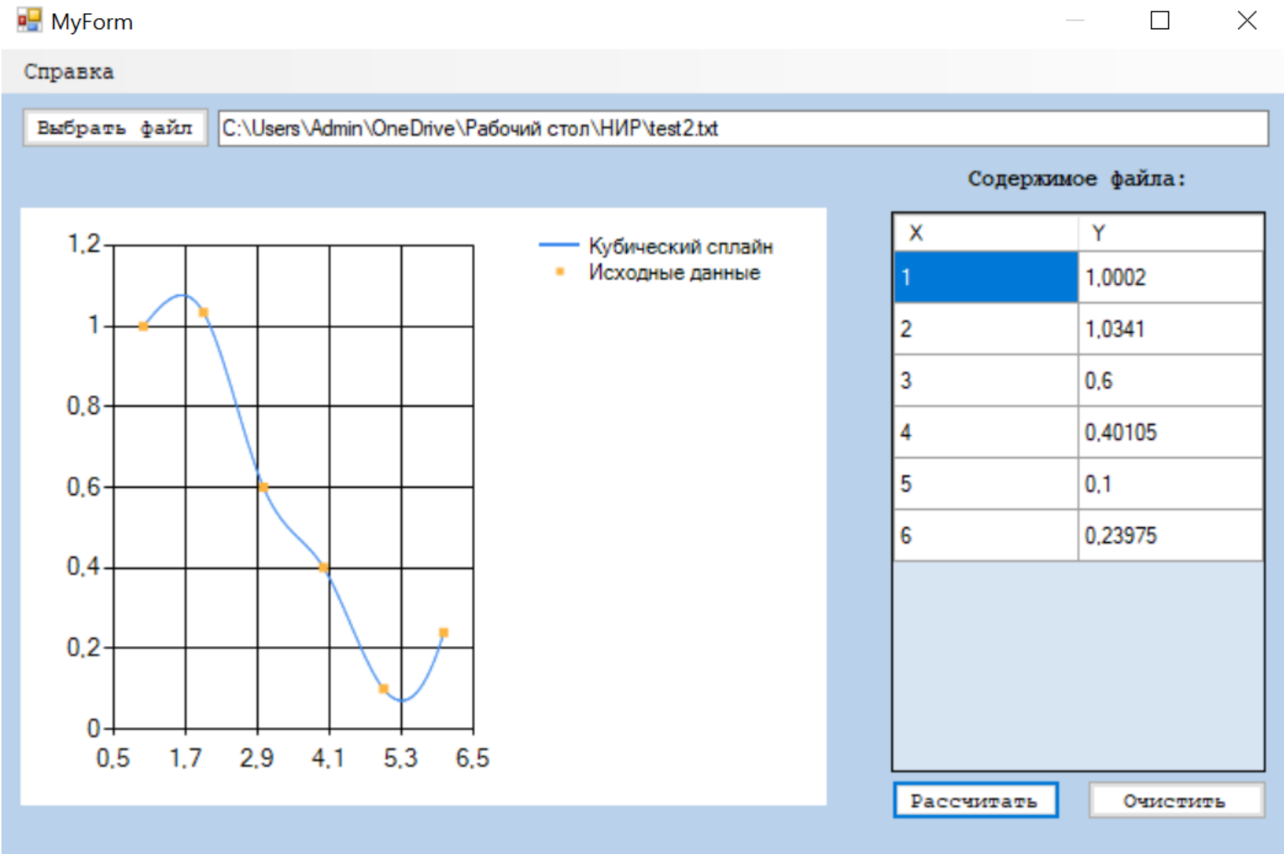

Рис. 6 – Пример расчёта второй функции по координатам

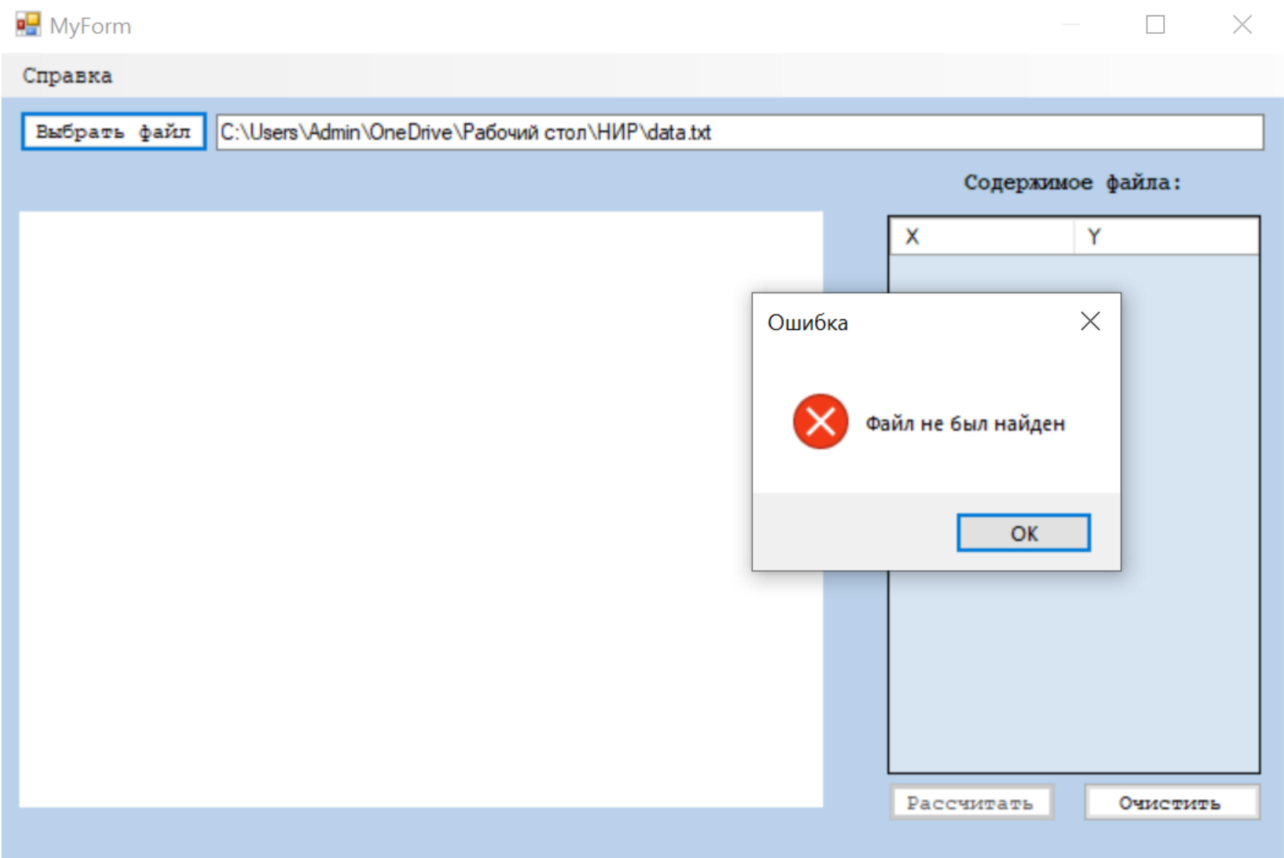

Рис. 7 – Пример ошибки

Код основных функций

```
#pragma once
#include "about.h"
using namespace std;

namespace SplineInterpolation {

	using namespace System;
	using namespace System::ComponentModel;
	using namespace System::Collections;
	using namespace System::Windows::Forms;
	using namespace System::Data;
	using namespace System::Drawing;
	using namespace System::IO;

public ref class MyForm : public System::Windows::Forms::Form
{

…


#pragma endregion
		Generic::List<Double>^ x;
		Generic::List<Double>^ y;

private: System::Void chooseFile_Click(System::Object^ sender, 
System::EventArgs^ e) {
		x = gcnew Generic::List <Double>();
		y = gcnew Generic::List <Double>();
		calculate->Enabled = false;
		chart1->Legends[0]->Enabled = false;

		if (openFileDialog->ShowDialog() == 
Windows::Forms::DialogResult::OK) {

			pathBox->Text = openFileDialog->FileName;

			try {
				StreamReader^ sr = gcnew 
StreamReader(openFileDialog->FileName, 
System::Text::Encoding::Default);

				String^ line;
				dataGridView1->Rows->Clear();
				while (!sr->EndOfStream) {
					line = sr->ReadLine();
					line = line->Replace('.', ',');
					cli::array<String^, 1>^ nums = line-
>Split((gcnew String(" \t"))->ToCharArray(), 
StringSplitOptions::RemoveEmptyEntries);
```

```
                    double xx = Double::Parse(nums[0]);
                    double yy = Double::Parse(nums[1]);
                    x->Add(xx);
                    y->Add(yy);
                    dataGridView1->Rows->Add(xx.ToString(),
yy.ToString());
                }

                chart1->Series[0]->Points->Clear();
                chart1->Series[1]->Points->Clear();
                for (int i = 0; i < x->Count; i++) {
                    chart1->Series[1]->Points->AddXY(x[i],
y[i]);
                }

                calculate->Enabled = true;

                double xMin = x[0];
                double xMax = x[0];

                for (int i = 0; i < x->Count; i++) {
                    if (x[i] < xMin)
                        xMin = x[i];
                    if (x[i] > xMax)
                        xMax = x[i];
                }

                double dx = (xMax - xMin) / 10;
                chart1->ChartAreas[0]->AxisX->Minimum = xMin -
dx;
                chart1->ChartAreas[0]->AxisX->Maximum = xMax +
dx;
                sr->Close();
            }
            catch (Exception^ e) {
                MessageBox::Show(this, "Файл не был найден",
"Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);
            }
        }

}

void CalculateFunction()
{
    int n = x->Count - 1;
    array<Double>^ k = gcnew array<double>(n + 1);
    array<Double>^ c = gcnew array<double>(n + 2);
    k[1] = 0;
    c[1] = 0;
    int i, j, l, m;
    double a, b, q, g, h, d;
    double xi, yi;
    for (i = 2; i <= n; i++) {
```

```
			j = i - 1;
			m = j - 1;
			a = x[i] - x[j];
			b = x[j] - x[m];
			g = 2 * (a + b) - b * c[j];
			c[i] = a / g;
			k[i] = (3.0 * ((y[i] - y[j]) / a - (y[j] - y[m]) / b) - b
* k[j]) / g;
		}
		c[n + 1] = k[n];
		for (i = n; i >= 0; i--) {
			c[i] = k[i] - c[i] * c[i + 1];
		}
		int nx = 100;
		chart1->Series[0]->Points->Clear();
		for (i = 1; i <= n; i++) {
			h = (x[i] - x[i - 1]) / (nx - 1);
			for (l = 0; l < nx; l++) {
				xi = x[i - 1] + l * h;
				q = x[i] - x[i - 1];
				g = xi - x[i - 1];
				b = (y[i] - y[i - 1]) / q - (c[i + 1] + 2 * c[i]) *
q / 3.0;
				d = (c[i + 1] - c[i]) / q * g;
				yi = y[i - 1] + g * (b + g * (c[i] + d / 3.0));
				chart1->Series[0]->Points->AddXY(xi, yi);
			}
		}
		chart1->Legends[0]->Enabled = true;
}

private: System::Void calculate_Click(System::Object^ sender,
System::EventArgs^ e) {
		CalculateFunction();
}

private: System::Void buttonClear_Click(System::Object^ sender,
System::EventArgs^ e) {
		pathBox->Clear();
		dataGridView1->Rows->Clear();
		chart1->Series[0]->Points->Clear();
		chart1->Series[1]->Points->Clear();
		chart1->Legends[0]->Enabled = false;
}
private: System::Void
оЗаданииToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
		Form^ aboutform = gcnew about();
		aboutform->Show();
}
};
}
```

Вывод

В процессе выполнения научно-исследовательской работы была написана программа с графическим пользовательским интерфейсом, которая решает задачу одномерной линейной интерполяции сеточной функции кубическими сплайнами. Результаты работы программы были протестированы на двух примерах.

В результате работы были достигнуты:

1. Реализация программы в соответствии с описанным заданием.
2. Реализация вычисление функции согласно алгоритму.
3. Реализация отображение графиков в зависимости от заданных координат.

Программа написана на языке программирования C++/CLI с использованием:

- среды исполнения CRL
- фреймворка .NET Framework 4.6.1

Список использованных источников

1. Градов В.М., Компьютерные технологии в практике математического моделирования: Учебное пособие. – Ч.1. – М.: Изд-во МГТУ им. Баумана, 2005. – 108 с.
2. Дорофеев А.А., Моделирование и обработка числовых данных с помощью унифицированной технологии построения интерполяционных сплайнов: Дис. … канд. технических наук 05.13.18 / Дорофеев Алексей Анатольевич; Южно-Российский государственный политехнический университет (НПИ) имени М.И. Платова – Новочеркасск. 2016. – 231 с.
3. Линейная интерполяция [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Линейная_интерполяция
4. Spline interpolation [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Spline_interpolation