МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

**«Московский технический университет связи и информатики»**

Кафедра «Информатика»

**Отчет по заданию №3**

**по дисциплине**

**«Численные методы»**

Выполнил: студент гр. БЭИ2202

Тогузов А. А.

Вариант 24.

Проверил: доц. каф. «Информатика»

Мацкевич А. Г.

Москва, 2023 г.

* 1. **Задание для решения задачи аппроксимации**

Для решения задачи аппроксимации методом наименьших квадратов выберем функцию y(x), заданную следующей таблицей:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| x | -0.1 | 0.1 | 0.3 | 0.5 | 0.7 | 0.9 |
| y | -0.356 | 1,376 | 1,03 | -0,316 | -1,296 | -0,91 |

* 1. **Линейная аппроксимация:**

Вычислить и записать в таблицу элементы матрицы Грамма и столбец свободных членов:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| 0 | -0.1 | -0.356 | 0.01 | 0.036 |
| 1 | 0.1 | 1,376 | 0.01 | 0.138 |
| 2 | 0.3 | 1,03 | 0.09 | 0.309 |
| 3 | 0.5 | -0,316 | 0.25 | -0.158 |
| 4 | 0.7 | -1,296 | 0.49 | -0.907 |
| 5 | 0.9 | -0,91 | 0.81 | -0.819 |

Составить системы нормальных уравнений:

Для линейной функции **P1(x) = А0+А1\*x** система нормальных уравнений примет вид (линейная аппроксимация):

**6\*А0+2.4\*А1 = -0.472**

**2.4\*А0+1.66\*А1 = -1.401**

Решив систему уравнений, получим коэффициенты **А0 = 0.614** и **А1 = -1.309**, тогда полином первой степени будет таким:

**P1(x) = -1.309x +0.61**

* 1. **Линейная аппроксимация:**

import numpy as np

import matplotlib.pyplot as plt

# Задаем исходные данные

x = np.array([-0.1, 0.1, 0.3, 0.5, 0.9])

y = np.array([-0.356, 1.376, 1.03, -0.316, -0.91])

# Создаем список степеней многочлена

degrees = [1, 2, 3, 4, 5]

# Выполняем аппроксимацию наименьших квадратов для каждой степени многочлена

for degree in degrees:

    coefficients = np.polyfit(x, y, degree)

    poly = np.poly1d(coefficients)

    print(coefficients)

    # Генерируем значения для построения графика

    x\_vals = np.linspace(x.min(), x.max(), 100)

    y\_vals = poly(x\_vals)

    # Вычисляем среднеквадратичное отклонение

    y\_pred = poly(x)

    rmse = np.sqrt(np.mean((y - y\_pred) \*\* 2))

    # Строим график и выводим RMSE

    plt.plot(x\_vals, y\_vals, label=f'Степень: {degree}, СКО: {rmse:.2f}')

plt.scatter(x, y, label='Исходные данные')

plt.xlabel('x')

plt.ylabel('y')

plt.legend()

plt.show()

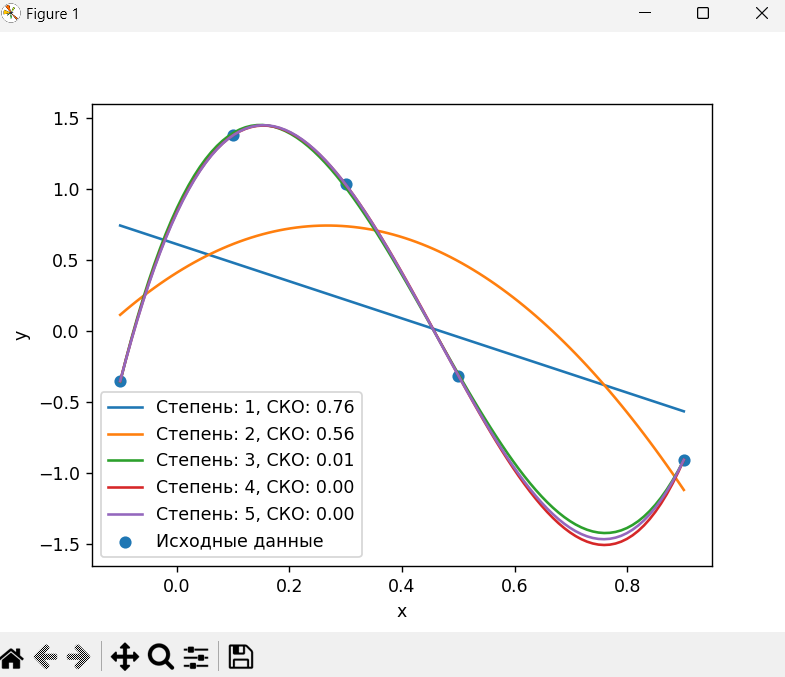


Рисунок 1 – Полученные полиномы и их СКО

Полученные в ходе аппроксимации многочлены:

;