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»

Лабораторная работа № 4

Тема: Построение и программная реализация алгоритма наилучшего среднеквадратичного приближения.

Студент: Фролов Е.А.

Группа: ИУ7-45Б

Оценка (баллы): _______________

Преподаватель: Градов В.М.

Москва
2021 г

Цель работы

Получение навыков построения алгоритма метода наименьших квадратов с использованием полинома заданной степени при аппроксимации табличных функций с весами

Исходные данные

1) Степень аппроксимирующего полинома - n
2) Таблица функции с количеством узлов N с весами ρ_i, которая формируется случайным образом при каждом запуске программы

Выходные данные

Графики, на которых построены точки (по таблице) и кривые (найденные полиномы).

1) Веса одинаковые: Полиномы степеней 1, 2 и введенной
2) Веса разные: Полиномы степеней 1, 2 и введенной (такой же набор с одинаковыми весами для сравнения)

Анализ алгоритма

1) Выбирается степень полинома n << N (размера таблицы)
2) Составляется СЛАУ следующим образом:

$$\sum_{m=0}^{n}(x^k, x^m)\, a_m = (y, x^k)\ ,\ 0 \le k \le n\ ,$$

где $\quad (x^k, x^m) = \sum_{i=1}^{N} \rho_i x_i^{k+m}, \quad (y, x^k) = \sum_{i=1}^{N} \rho_i y_i x_i^k\ .$

3) Получившаяся система решается методом Гаусса, в результате чего получаются коэффициенты полинома

Код программы

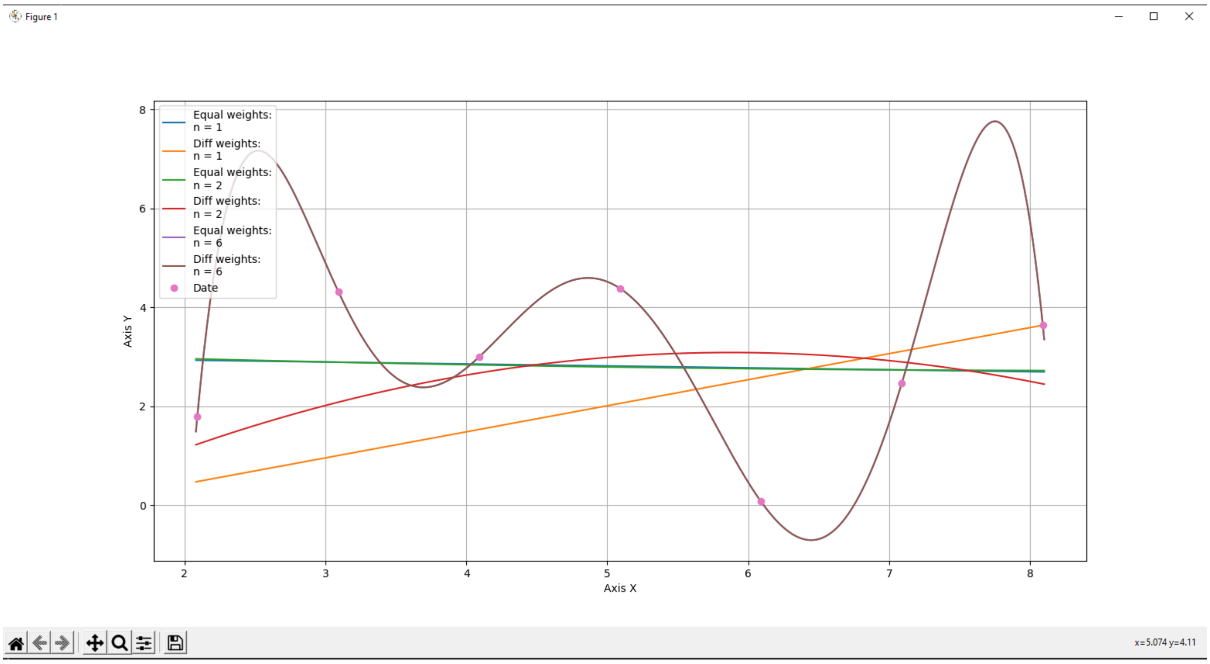
```python
from copy import deepcopy
from random import random
import matplotlib.pyplot as plt

def f(x, n):
    return x ** n

def init_matrix(size):
    return [[0 for i in range(size + 2)] for i in range(size + 1)]

def make_slae_matrix(table, power):
    size_tab = len(table)
    matrix = init_matrix(power)
    for i in range(power + 1):
        for j in range(power + 1):
            a_factor = 0.0
            rsight_factor = 0.0
            for k in range(size_tab):
                weight = table[k][2]
                x = table[k][0]
                y = table[k][1]
                a_factor += weight * f(x, (i + j))
                rsight_factor += weight * y * f(x, i)
            matrix[i][j] = a_factor
            matrix[i][power + 1] = rsight_factor

    return matrix

def solve_matrix_gauss(matrix):
    size_mat = len(matrix)

    for i in range(size_mat):
        for j in range(i + 1, size_mat):
            if (i == j):
                continue
            k = matrix[j][i] / matrix[i][i]
            for q in range(i, size_mat + 1):
                matrix[j][q] -= k * matrix[i][q]

    result = [0 for i in range(size_mat)]

    for i in range(size_mat - 1, -1, -1):
        for j in range(size_mat - 1, i, -1):
            matrix[i][size_mat] -= result[j] * matrix[i][j]
        result[i] = matrix[i][size_mat] / matrix[i][i]

    return result

def generate_table():
    table = []
    x = random() * 5
    for _ in range(SIZE):
        y = random() * 5
        table.append([x, y, BASE_WEIGHT])
```

```python
def change_weight(table):
    changed = True

    try:
        place = int(input("\nВведите номер точки в таблице: "))
        new_weight =  float(input("\nВведите новый вес точки: "))
    except:
        return

    table[place - 1][2] = new_weight

    return table

def print_menu():
    print("\nМеню\n \
    \n1. Распечатать таблицу\
    \n2. Изменить вес точки\
    \n3. Вывести результаты\n\
    \n0. Выйти")

def out_table(table):
    print("\n   Сгенерированная таблица\n")
    print("  №    |     x     |     y    |   w   ")
    print("--------------------------------------")
    for i in range(len(table)):
        print("  %-3d  |   %-5.2f   |   %-4.2f   |   %-5.2f   " %(i + 1, table[i][0], table[i][1], table[i][2]))

def make_equal_weights_table(table):
    for i in range(len(table)):
        table[i][2] = 1

    return table

def get_coords(table):
    array_x = []
    array_y = []
    for i in range(len(table)):
        array_x.append(table[i][0])
        array_y.append(table[i][1])

    return array_x, array_y

def find_graph(table, cur_power):
    matrix = make_slae_matrix(table, cur_power)
    result = solve_matrix_gauss(matrix)

    x, y = [], []
    k = table[0][0] - eps

    while (k <= table[len(table) - 1][0] + eps):
        y_cur = 0
        for j in range(0, cur_power + 1):
            y_cur += result[j] * f(k, j)

        x.append(k)
        y.append(y_cur)
        k += eps

    return x, y
```

```python
def solve(table):
    try:
        power = int(input("\nВведите степень аппроксимирующего многочлена: "))
    except:
        return

    if changed:
        changed_table = deepcopy(table)
        table = make_equal_weights_table(table)

    for cur_power in range(1, power + 1):
        if (cur_power > 2 and cur_power < power):
            continue

        x, y = find_graph(table, cur_power)
        plt.plot(x, y, label = "Equal weights:\nn = %d" %(cur_power))

        if changed:
            x, y = find_graph(changed_table, cur_power)
            plt.plot(x, y, label = "Diff weights:\nn = %d" %(cur_power))

    array_x, array_y = get_coords(table)
    plt.plot(array_x, array_y, 'o', label = "Date")

    plt.legend()
    plt.grid()
    plt.xlabel("Axis X")
    plt.ylabel("Axis Y")
    plt.show()

    if changed:
        return changed_table

    return table

SIZE = 7
BASE_WEIGHT = 1
eps = 0.01
changed = False

if __name__ == "__main__":
    changed = False
    table = generate_table()
    punkt = -1

    while (punkt != 0):
        print_menu()
        try:
            punkt = int(input("\nВведите пункт меню: "))
        except:
            continue
        if (punkt == 1):
            out_table(table)
        elif (punkt == 2):
            table = change_weight(table)
        elif (punkt == 3):
            table = solve(table)
```

Пример работы программы

```
Меню

1. Распечатать таблицу
2. Изменить вес точки
3. Вывести результаты

0. Выйти

Введите пункт меню: 2

Введите номер точки в таблице: 1

Введите новый вес точки: -3

Меню

1. Распечатать таблицу
2. Изменить вес точки
3. Вывести результаты

0. Выйти

Введите пункт меню: 1

   Сгенерированная таблица

  №    |     x     |     y    |   w
--------------------------------------
  1    |   2.09    |   1.80   |   -3.00
  2    |   3.09    |   4.32   |   1.00
  3    |   4.09    |   3.00   |   1.00
  4    |   5.09    |   4.38   |   1.00
  5    |   6.09    |   0.08   |   1.00
  6    |   7.09    |   2.47   |   1.00
  7    |   8.09    |   3.64   |   1.00

Меню

1. Распечатать таблицу
2. Изменить вес точки
3. Вывести результаты

0. Выйти

Введите пункт меню: 3

Введите степень аппроксимирующего многочлена: 6
```



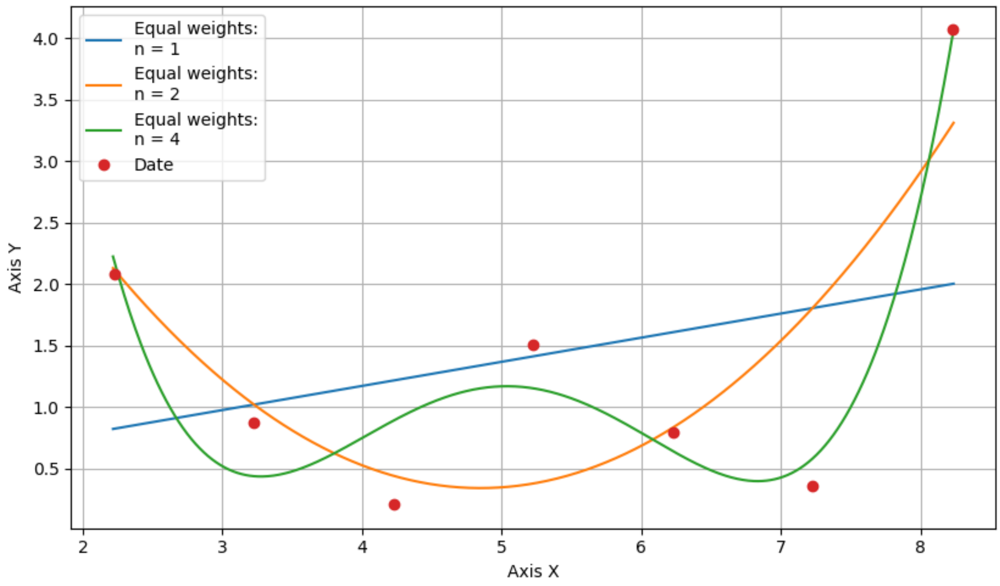
Результаты

1) Одинаковые веса. Входная таблица:

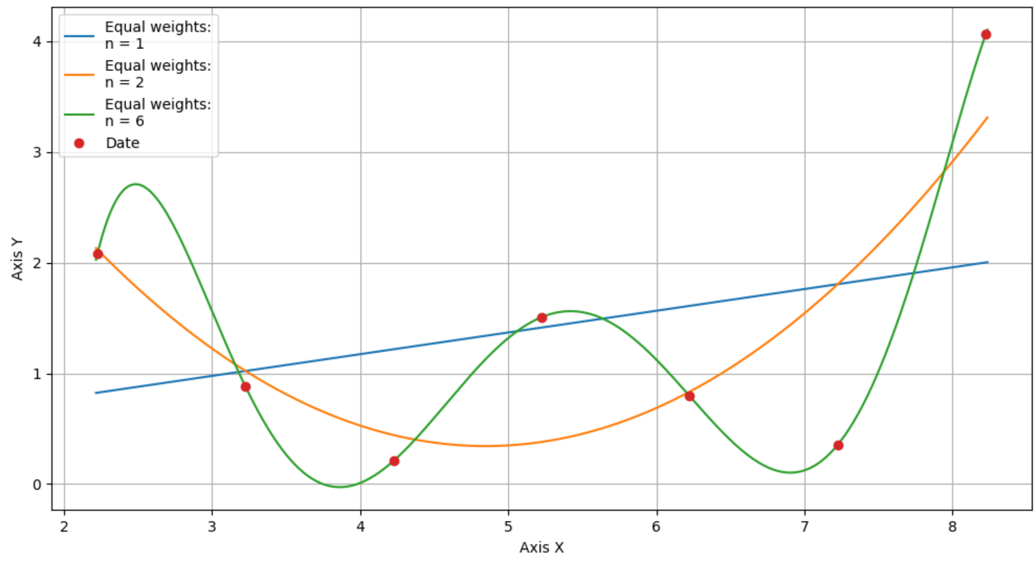
Сгенерированная таблица

№	x	y	w
1	2.23	2.08	1.00
2	3.23	0.88	1.00
3	4.23	0.21	1.00
4	5.23	1.51	1.00
5	6.23	0.79	1.00
6	7.23	0.36	1.00
7	8.23	4.07	1.00

Полиномы степеней 1, 2, 4:



Полиномы степеней 1, 2, 6:

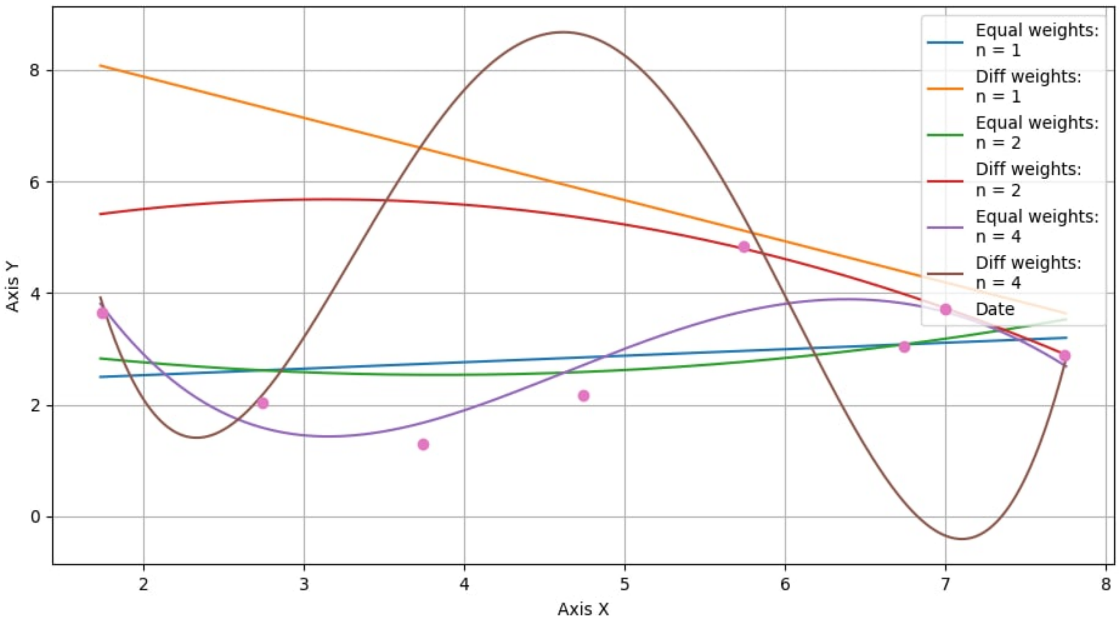


2) Веса разные. Входная таблица:

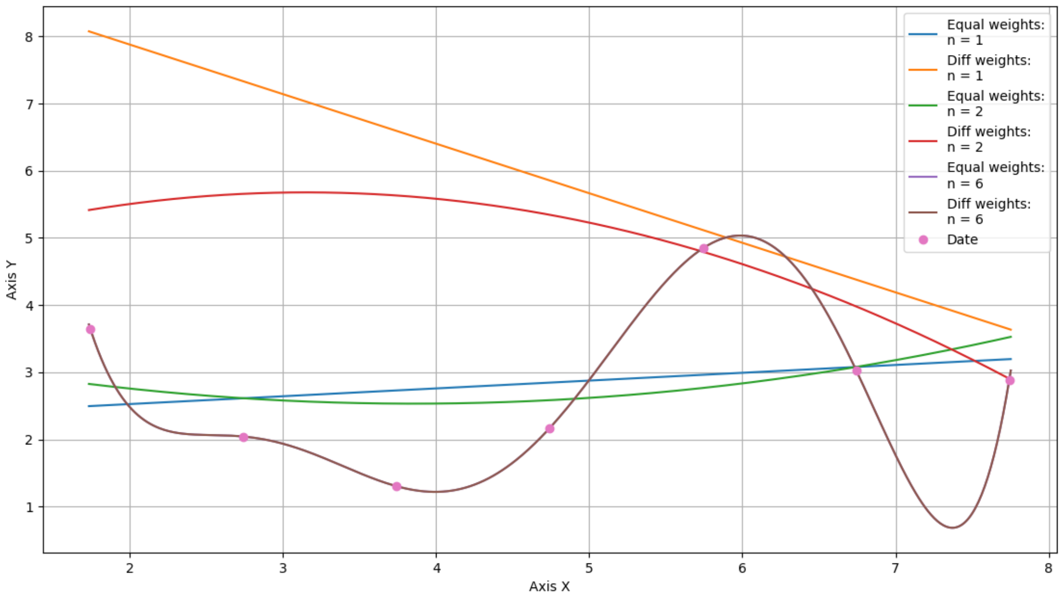
Сгенерированная таблица

№	x	y	w
1	1.74	3.64	-0.70
2	2.74	2.04	0.50
3	3.74	1.30	0.30
4	4.74	2.17	-0.70
5	5.74	4.85	9.99
6	6.74	3.03	1.00
7	7.74	2.89	-3.45

Полиномы степеней 1, 2, 4:



Полиномы степеней 1, 2, 6:



Вопросы при защите лабораторной работы

1. Что произойдет при задании степени полинома n=N-1 (числу узлов таблицы минус 1)?

Если степень полинома n = N - 1, тогда аппроксимирующая функция пройдет по всем заданным точкам.

2. Будет ли работать Ваша программа при n >= N ? Что именно в алгоритме требует отдельного анализа данного случая и может привести к аварийной остановке?

При условии n >= N, невозможно построить полином n степени, из-за недостатка точек, т. к. при этом определитель матрицы будет = 0. Программа работать будет, но в некоторые моменты может возникнуть непредвиденная ситуация. Произойдет деление на ноль.

3. Получить формулу для коэффициента полинома a0 при степени полинома n=0. Какой смысл имеет величина, которую представляет данный коэффициент?

$a0 = \Sigma(p(i)*y(i)) / \Sigma p(i)$, сумма от i = 1 до N

Значение выражения - математическое ожидание

4. Записать и вычислить определитель матрицы СЛАУ для нахождения коэффициентов полинома для случая, когда n=N=2. Принять все i =1.

n = N = 2

Входная таблица:

x	y	ρ
x1	y1	ρ1
x2	y2	ρ2

Выходит система:

(ρ1 + ρ2)a0 + (ρ1x1 + ρ2x2)a1 + (ρ1x1^2 + ρ2x2^2)a2 = ρ1y1 + ρ2y2

(ρ1x + ρ2x)a0 + (ρ1x1^2 + ρ2x2^2)a1 + (ρ1x1^3 + ρ2x2^3)a2 = ρ1x1y1 + ρ2x2y2

(ρ1x^2 + ρ2x^2)a0 + (ρ1x1^3 + ρ2x2^3)a1 + (ρ1x2^4 + ρ2x2^4)a2 = ρ1x2^2y2

Тогда:

(2 x1 + x2 x1^2 + x2^2)

(x1 + x2 x1^2 + x2^2 x1^3 + x2^3) = A

(x1^2 + x2^2 x1^3 + x2^3 x1^4 + x2^4)

|A| = 0 => **нет решений**

5. Построить СЛАУ при выборочном задании степеней аргумента полинома $\varphi(x) = a_0 + a_1 x^m + a_2 x^n$ **, причем степени n и m в этой формуле известны.**

(x^0, x^0)a0 + (x^0, x^0)a1 + (x^0, x^0)a2 = (y, x^0)

(x^m, x^0)a0 + (x^m, x^n)a1 + (x^m, x^n)a2 = (y, x^m)

(x^n, x^0)a0 + (x^n, x^m)a1 + (x^n, x^n)a2 = (y, x^n)

6. Предложить схему алгоритма решения задачи из вопроса 5, если степени n и m подлежат определению наравне с коэффициентами a(k), т.е. количество неизвестных равно 5.

Нужно сделать полный перебор по степеням n и m, чтобы вычислить коэффициенты ***a*** и затем вычислить значение матрицы, и выбрать самое близкое значение к эталонному