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 Варламова Е.А.

Группа ИУ7-41Б

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.

1. Исходные данные

1. *Веса всех точек одинаковы и равны единице.*

x	**y**	**p_i**
1.2	1.8	1
2.3	2.1	1
3.4	5.4	1
4.5	3.7	1
5.6	1.3	1

Таблица 1

2. *Веса точек разные.*

x	**y**	**p_i**
1.2	1.8	2
2.3	2.1	3
3.4	5.4	2
4.5	3.7	1
5.6	1.3	10

Таблица 2

2. Код программы

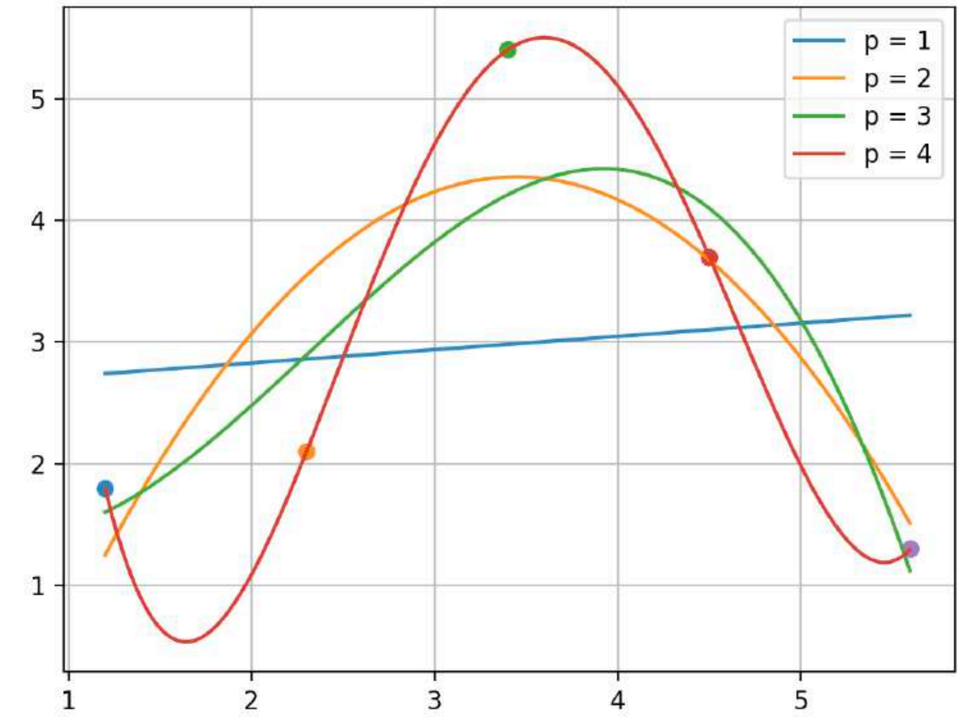
```
import random
import matplotlib.pyplot as plt

# решение СЛАУ методом Гаусса
def gauss(matrix, n):
    for k in range(n):
        for i in range(k + 1, n):
            coeff = -(matrix[i][k] / matrix[k][k])
            for j in range(k, n + 1):
                matrix[i][j] += coeff * matrix[k][j]

    a = [0 for i in range(n)]
    for i in range(n - 1, -1, -1):
        for j in range(n - 1, i, -1):
            matrix[i][n] -= a[j] * matrix[i][j]
        a[i] = matrix[i][n] / matrix[i][i]
    return a

#метод наименьших квадратов
def least_squares_method(x, y, ro, n):
    N = len(x)
    matrix = []
    for k in range(n + 1):
        array = []
        for m in range(n + 1):
            s = 0
            for i in range(N):
                s += ro[i] * (x[i])**(k + m)
            array.append(s)
        s = 0
        for i in range(N):
            s += ro[i] * y[i] * (x[i])**k
        array.append(s)
        matrix.append(array)
    res = gauss(matrix, n + 1)
    return res

def read_file():
    f = open("data.txt")
    x = []
    y = []
    ro = []

    for line in f:
        try:
            xp, yp, rop = map(float, line.split())
        except:
```

```
            return ()
        x.append(xp)
        y.append(yp)
        ro.append(rop)
    f.close()
    return (x, y, ro)

def print_graph(x, y, ro):
    for i in range (len(x)):
        plt.scatter(x[i], y[i])
    mi = min(x)
    ma = max(x)
    for n in [1, 2, 3, 5]:
        if n >= N:
            continue
        coefs = least_squares_method(x, y, ro, n)
        x_pr = []
        y_pr = []
        dx = (ma - mi) / 1000
        i = mi
        while i < ma:
            s = 0
            for j in range(len(coefs)):
                s += (i**j) * coefs[j]
            x_pr.append(i)
            y_pr.append(s)
            i += dx
        plt.plot(x_pr, y_pr, label='p = {}'.format(n))

    plt.grid(True)
    plt.legend(loc='best')
    plt.show()
    return


def generate_file():
    n = int(input("количество точек: "))
    flag = int (input("0 - разные веса\n1 - одинаковые веса\n"))
    same = True
    if flag == 0:
        same = False
    f = open("data.txt", "w")
    for i in range(n):
        x = random.randint(0, 30) / 10
        y = random.randint(0, 30) / 10
        ro = 1
        if same == False:
            ro = random.randint(0, 5)
        f.write("{} {} {}\n".format(x, y, ro))
```

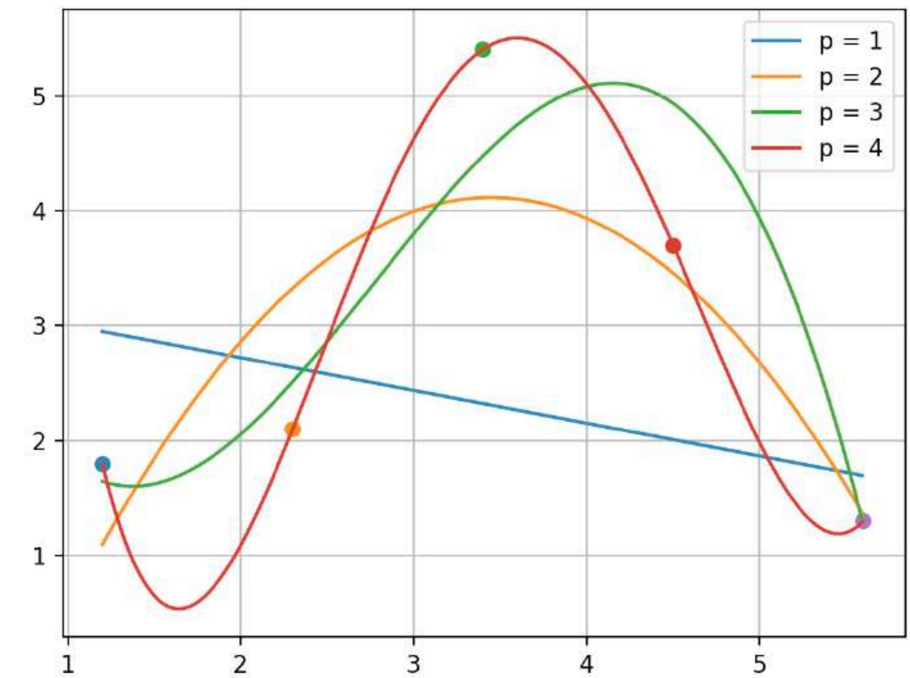
```
    f.close()

#main
choise = int(input("0 - использовать существующий файл\n\
1 - сгенерировать новый\n"))
t = ()
if choise == 0:
    t = read_file()
    if t == ():
        print("произошла ошибка чтения, сгенерируйте новый файл")
        generate_file()
        t = read_file()
if choise == 1:
    generate_file()
    t = read_file()
#n = int(input("степень полинома: "))
print_graph(t[0], t[1], t[2])
```

3. Результат работы программы

1. *Таблица входных данных с одинаковыми весами* (**Таблица 1**):

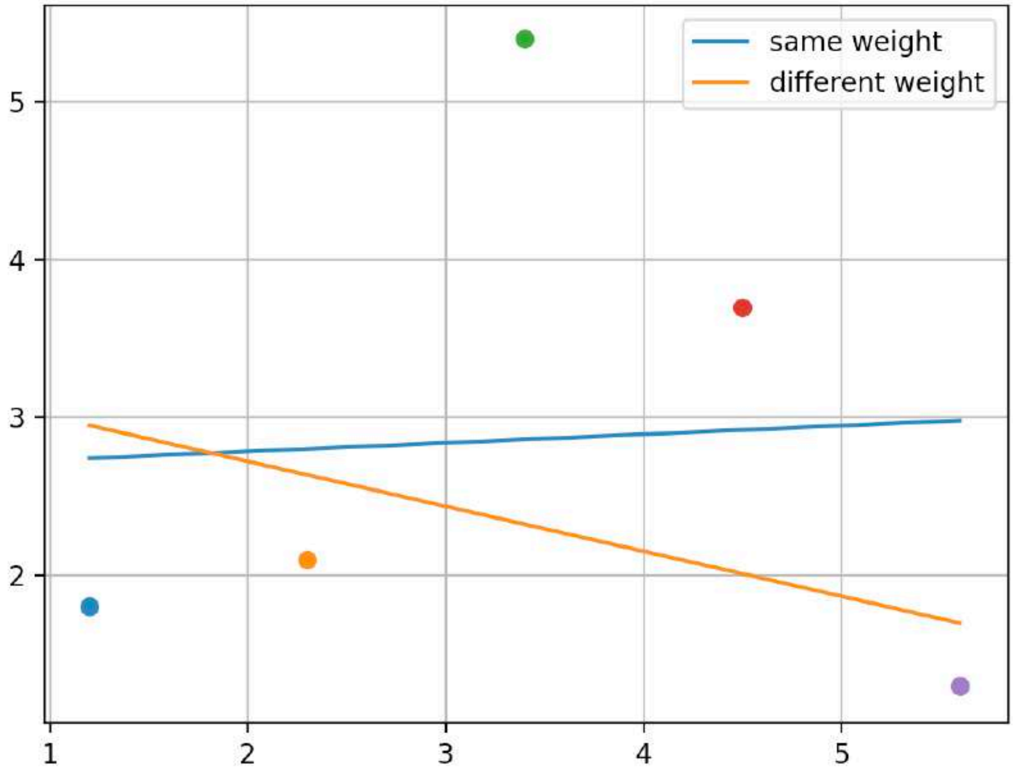


2. *Таблица входных данных с разными весами (**Таблица 2**)*



Изменив веса точек, мы видим, что наклон прямой (полинома первой степени) изменился.

Более очевидно на одном графике (данные взяты из тех же 2-х таблиц):



4. Ответы на вопросы

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

Кривая пройдёт через все заданные точки.

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

Формально программа работать не будет, если не предусмотреть условие n < N, так как при n >= N определитель будет равен нулю, а потому коэффициенты не могут быть определены однозначно. На практике программа может выдавать результат из-за вычислений с числами с плавающей точкой, но при определённых данных она в любом случае завершится аварийно.

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

$(x^0, x^0)a = (y, x^0)$

$Sum(p_i) * a = Sum(p_i * y_i)$

$a = Sum(p_i * y_i) / Sum(p_i), 0 <= i < N$

p_i – вес точки, N – количество точек

Коэффициент представляет собой математическое ожидание.

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

x_i	y_i	p_i
x_0	y_0	$p_0 = 1$
x_1	y_1	$p_1 = 1$

$$\begin{cases} (x^0, x^0)a_0 + (x^0, x^1)a_1 + (x^0, x^2)a_2 = (y, x^0) \\ (x^1, x^0)a_0 + (x^1, x^1)a_1 + (x^1, x^2)a_2 = (y, x^1) \\ (x^2, x^0)a_0 + (x^2, x^1)a_1 + (x^2, x^2)a_2 = (y, x^2) \end{cases}$$

$$\begin{cases} \left(\sum_{i=1}^{2} p_i\right)a_0 + \left(\sum_{i=1}^{2} p_i x_i\right)a_1 + \left(\sum_{i=1}^{2} p_i x_i^2\right)a_2 = \sum_{i=1}^{2} p_i y_i \\ \left(\sum_{i=1}^{2} p_i x_i\right)a_0 + \left(\sum_{i=1}^{2} p_i x_i^2\right)a_1 + \left(\sum_{i=1}^{2} p_i x_i^3\right)a_2 = \sum_{i=1}^{2} p_i y_i x_i \\ \left(\sum_{i=1}^{2} p_i x_i^2\right)a_0 + \left(\sum_{i=1}^{2} p_i x_i^3\right)a_1 + \left(\sum_{i=1}^{2} p_i x_i^4\right)a_2 = \sum_{i=1}^{2} p_i y_i x_i^2 \end{cases}$$

Упростим ($p_i = 1$):

$$\begin{cases} 2a_0 + (x_0+x_1)a_1 + (x_0^2+x_1^2)a_2 = y_0 + y_1 \\ (x_0+x_1)a_0 + (x_0^2+x_1^2)a_1 + (x_0^3+x_1^3)a_2 = \cancel{y_0^2+y_1^2}\ y_0x_0 + y_1x_1 \\ (x_0^2+x_1^2)a_0 + (x_0^3+x_1^3)a_1 + (x_0^4+x_1^4)a_2 = y_0x_0^2 + y_1x_1^2 \end{cases}$$

$$\begin{vmatrix} 2 & x_0+x_1 & x_0^2+x_1^2 \\ x_0+x_1 & x_0^2+x_1^2 & x_0^3+x_1^5 \\ x_0^2+x_1^2 & x_0^5+x_1^3 & x_0^4+x_1^4 \end{vmatrix} = 2(x_0^2+x_1^2)(x_0^4+x_1^4) + (x_0+x_1)(x_0^3+x_1^3)(x_0^2+x_1^2) +$$

$$+ (x_0^2+x_1^2)(x_0+x_1)(x_0^3+x_1^3) - (x_0^2+x_1^2)^3 - 2(x_0^3+x_1^3)^2 - (x_0^4+x_1^4)(x_0+x_1)^2 =$$

$= 0 =>$ решений нет

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.

$$\begin{cases} (x^0, x^0)a_0 + (x^0, x^m)a_1 + (x^0, x^n)a_2 = (y, x^0) \\ (x^m, x^0)a_0 + (x^m, x^m)a_1 + (x^m, x^n)a_2 = (y, x^m) \\ (x^n, x^0)a_0 + (x^n, x^m)a_1 + (x^n, x^n)a_2 = (y, x^n) \end{cases}$$

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

Для каждой пары n и m (при условии, что степень полинома меньше количества точек) вычислить коэффициенты a0, a1, a2 функции φ, а затем выбрать ту пару, для которой справедливо:

$$\sum_{i=1}^{N} \rho_i [y(x_i) - \varphi(x_i)]^2 = min$$