Министерство образования и науки 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 (ИУ)

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии (ИУ7)

Лабораторная работа №4

Тема: Построение и программная реализация алгоритма наилучшего среднеквадратичного приближения.

Студент Сучкова Т.М.
Группа ИУ7-42Б
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. Таблица функции с весами p_i с количеством узлов N. Сформировать таблицу самостоятельно со случайным разбросом точек.

x	y	p_i

2. Степень аппроксимирующего полинома - n.

Алгоритм наилучшего среднеквадратичного приближения

Пусть имеется множество функций $\varphi(x)$, принадлежащих линейному пространству функций. Под близостью в среднем исходной y и аппроксимирующей φ функций будем понимать результат оценки суммы

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

где p_i - вес точки. Суммирование выполняется по всем N узлам заданной функции.
Такой вид аппроксимации называют среднеквадратичным приближением.

Алгоритм:
1. Выбирается степень полинома n<<N. Обычно степень полинома не превышает 5-6.
2. Составляется система линейных алгебраических уравнений типа (6).

$$\sum_{m=0}^{n} (x^k, x^m)\, a_m = (y, x^k)\ ,\ 0 \le k \le n\ , \qquad (6)$$

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

3. В результате решения СЛАУ находятся коэффициенты полинома a_k.

В качестве исходных данных используется произвольная табличная функция, для каждого узла i которой пользователь задает вес p_i по своему усмотрению.

Код программы

main.py:

```
import matplotlib.pyplot as plt
import numpy as np

def f(x_arr, coef):
    res = np.zeros(len(x_arr))

    n = len(coef)
    for i in range(n):
        res += coef[i] * (x_arr**i)

    return res

def file_read(filename):
    f = open(filename, "r")
    x = []
    y = []
    p = []

    for line in f:
        line = line.split(" ")
        x.append(float(line[0]))
        y.append(float(line[1]))
        p.append(float(line[2]))

    return x, y, p

def table_print(x, y, p):
    n = len(x)
    print("%8s |%8s |%8s " % ("x    ", "y    ", "p    "))

    for i in range(n):
        print("%.6f |%.6f |%.6f" % (x[i], y[i], p[i]))
    print()

def matrix_print(matrix):
    for el in matrix:
        print(el)

def gauss_method(matrix):
    n = len(matrix)

    for k in range(n):
        for i in range(k + 1, n):
            coef = -(matrix[i][k] / matrix[k][k])
            for j in range(k, n + 1):
                matrix[i][j] += coef * matrix[k][j]

    #print("\ntriangled:")
    #matrix_print(matrix)
```

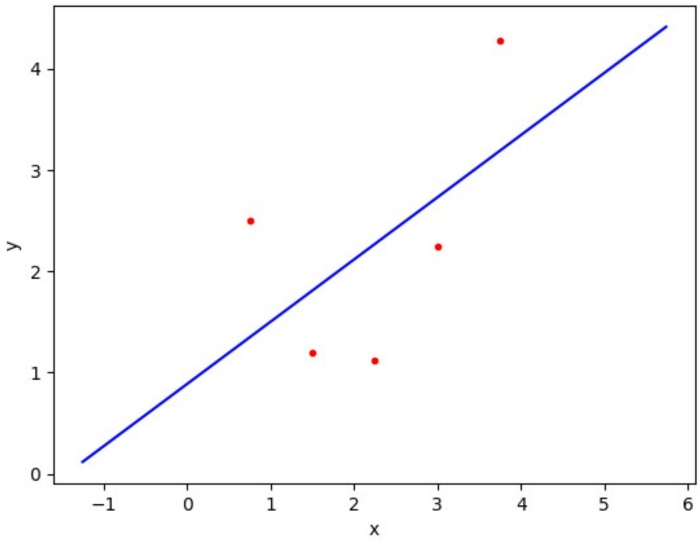
```
    # находим неизвестные
    a = [0 for i in range(n)]
    for i in range(n - 1, -1, -1):
        for j in range(n - 1, i, -1):
            matrix[i][n] -= a[j] * matrix[i][j]
        a[i] = matrix[i][n] / matrix[i][i]

    return a

def solve(x, y, p, n): #n - кол-во искомых коэффициентов
    l = len(x)

    sum_x_n = [sum([x[i]**j * p[i] for i in range(l)]) for j in range(n*2 -1)]
    sum_x_n = [round(el, 6) for el in sum_x_n]
    sum_yx_n = [sum([x[i]**j * p[i] * y[i] for i in range(l)]) for j in range(n)]

    matrix = [sum_x_n[i:i+n] for i in range(n)]

    for i in range(n):
        matrix[i].append(round(sum_yx_n[i], 6))
    matrix_print(matrix)

    return gauss_method(matrix)

def figure_show(a, x, y, p):
    t = np.arange(x[0] - 2, x[len(x) - 1] + 2, 0.01)

    plt.figure(1)

    plt.xlabel("x")
    plt.ylabel("y")
    plt.plot(t, f(t, a), 'b')

    for i in range(len(x)):
        plt.plot(x[i], y[i], 'ro', markersize = p[i] + 2)

    plt.show()

if __name__ == '__main__':
    x, y, p = file_read("data.txt")
    #N = len(x) - 1
    n = int(input("Введите порядок полинома: "))
    table_print(x, y, p)
    a = solve(x, y, p, n + 1)
    #print("\na:", a)
    figure_show(a, x, y, p)
```

Результаты работы программы.

1) Веса всех точек одинаковы и равны, например, единице.

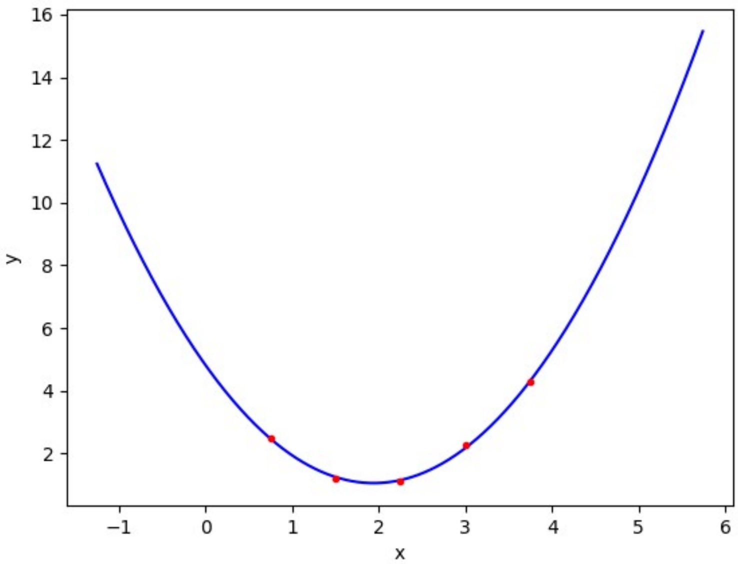
n = 1 (прямая)

```
Введите порядок полинома: 1
    x     |     y     |     p
0.750000 |2.500000 |1.000000
1.500000 |1.200000 |1.000000
2.250000 |1.120000 |1.000000
3.000000 |2.250000 |1.000000
3.750000 |4.280000 |1.000000

[5.0, 11.25, 11.35]
[11.25, 30.9375, 28.995]
```



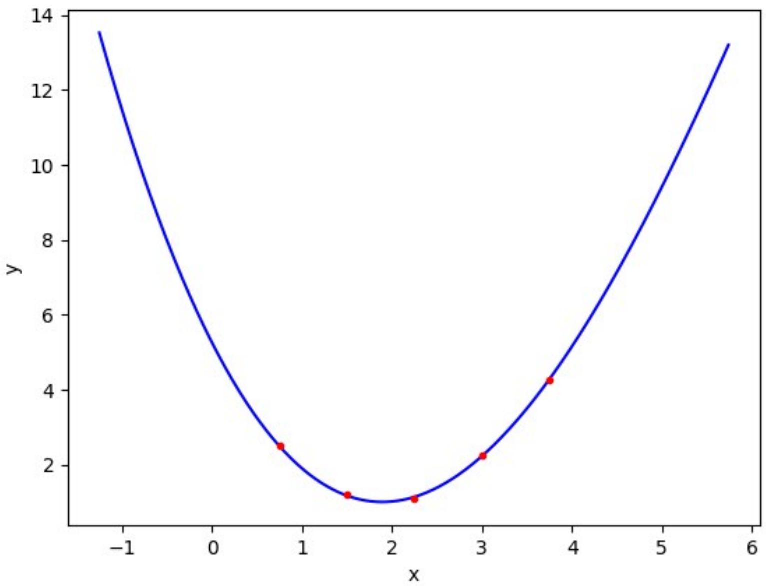
n = 2 (парабола)

```
Введите порядок полинома: 2
    x     |     y     |     p
0.750000 |2.500000 |1.000000
1.500000 |1.200000 |1.000000
2.250000 |1.120000 |1.000000
3.000000 |2.250000 |1.000000
3.750000 |4.280000 |1.000000

[5.0, 11.25, 30.9375, 11.35]
[11.25, 30.9375, 94.921875, 28.995]
[30.9375, 94.921875, 309.761719, 90.21375]
```



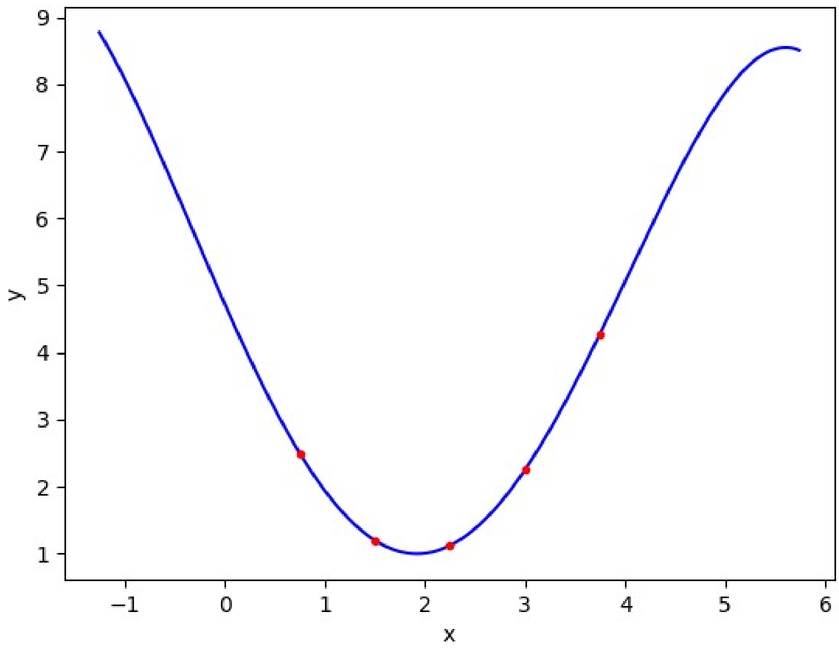
n = 3

```
Введите порядок полинома: 3
    x     |    y     |    p
0.750000 |2.500000 |1.000000
1.500000 |1.200000 |1.000000
2.250000 |1.120000 |1.000000
3.000000 |2.250000 |1.000000
3.750000 |4.280000 |1.000000

[5.0, 11.25, 30.9375, 94.921875, 11.35]
[11.25, 30.9375, 94.921875, 309.761719, 28.995]
[30.9375, 94.921875, 309.761719, 1050.073242, 90.21375]
[94.921875, 309.761719, 1050.073242, 3651.229248, 304.315313]
```

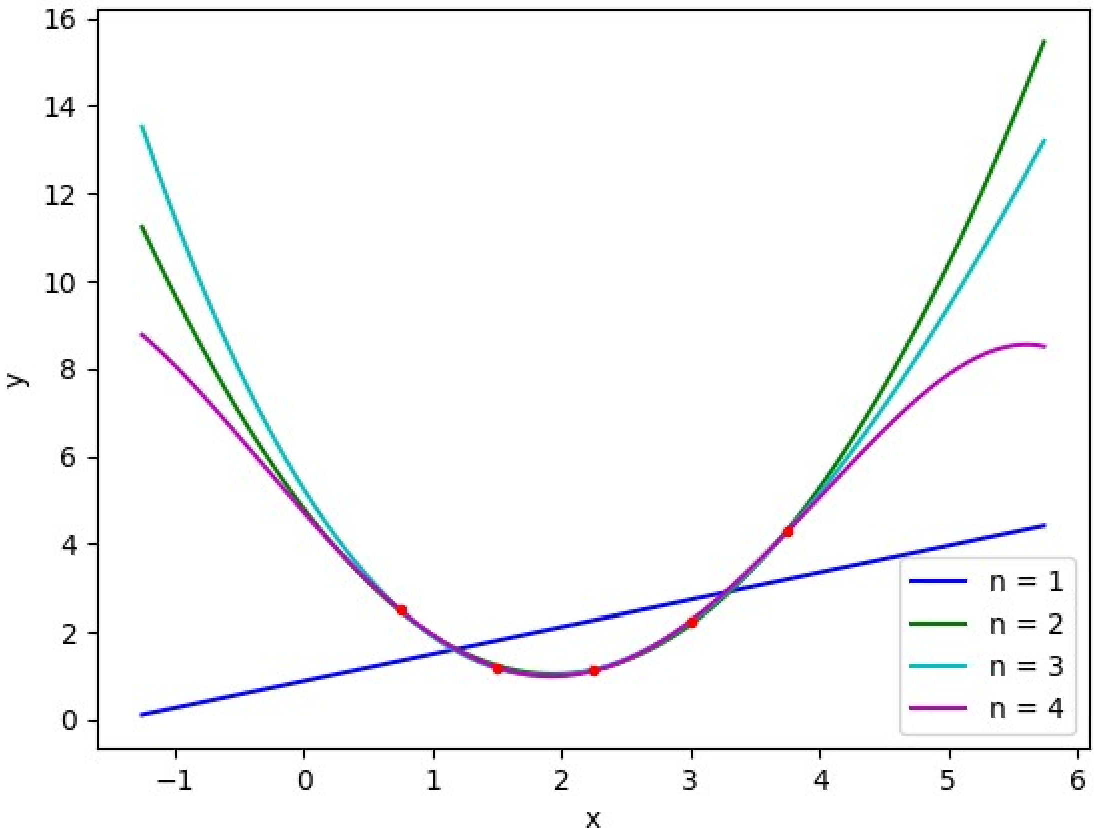


n = 4

```
Введите порядок полинома: 4
    x     |    y     |    p
0.750000 |2.500000 |1.000000
1.500000 |1.200000 |1.000000
2.250000 |1.120000 |1.000000
3.000000 |2.250000 |1.000000
3.750000 |4.280000 |1.000000

[5.0, 11.25, 30.9375, 94.921875, 309.761719, 11.35]
[11.25, 30.9375, 94.921875, 309.761719, 1050.073242, 28.995]
[30.9375, 94.921875, 309.761719, 1050.073242, 3651.229248, 90.21375]
[94.921875, 309.761719, 1050.073242, 3651.229248, 12924.577332, 304.315313]
[309.761719, 1050.073242, 3651.229248, 12924.577332, 46350.177292, 1064.207109]
```



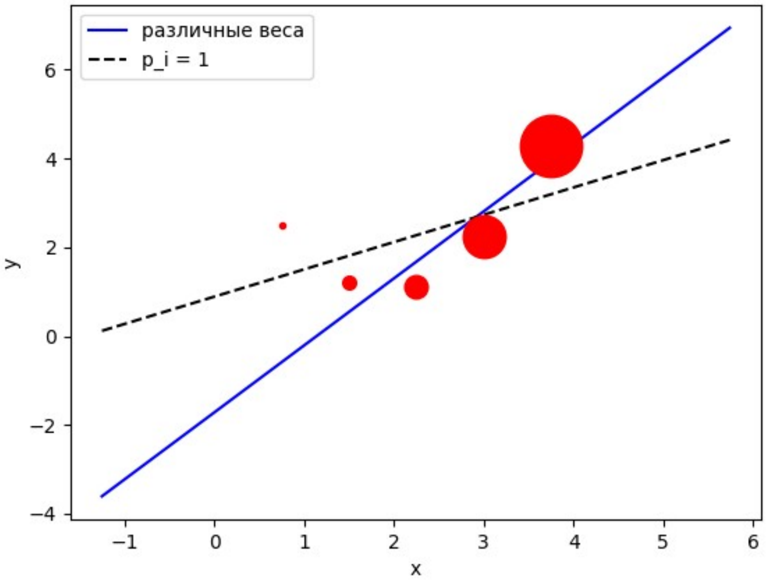
И общий график для наглядности
(p_i = 1, степени полинома меняются от 1 до n (вводимый параметр); n = 4)**:**



2) Веса точек разные. Продемонстрировать, как за счет назначения весов точкам можно изменить положение на плоскости прямой линии (полином первой степени), аппроксимирующей один и тот же набор точек (одну таблицу y(x)).

n = 1

```
Введите порядок полинома: 1
    x    |    y    |    p
0.750000 |2.500000 |1.000000
1.500000 |1.200000 |5.000000
2.250000 |1.120000 |10.000000
3.000000 |2.250000 |20.000000
3.750000 |4.280000 |30.000000

[66.0, 203.25, 193.1]
[203.25, 664.3125, 652.575]
[5.0, 11.25, 11.35]
[11.25, 30.9375, 28.995]
```



Веса представлены в таблице входных данных. Точки на графике имеют соответствующий их весам размер.
Вес определяет «значимость» точки. Чем больше вес точки, тем ближе к точке проходит аппроксимирующая кривая. Под весом можно понимать, например, величину, обратную относительной погрешности задания функции, т.е. чем более точное значение имеет табличная функция в некоторой точке, тем больше ее вес и тем ближе к ней пройдет график аппроксимирующей функции. Что является вполне логичным.

Вопросы при защите лабораторной работы.

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

В этом случае кривая пройдет по всем точкам независимо от весов.

На N точках может быть определен единственный полином степени n-1, проходящий через эти точки.
Тогда в формуле $\sum_{i=1}^{N} \rho_i [\, y(x_i) - \varphi(x_i)]^2 = min$

выражение в скобках примет значение 0 для любого из N узлов, поэтому независимо от весов будет построена кривая, проходящая по всем 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?

По идее построение кривой n-й степени по n точкам невозможно, т. к. определитель равен нулю. Но программа все считает из-за погрешности расчетов.

Из-за определителя, равного нулю, при решении методом Гаусса происходит деление на 0 при приведении к единичной матрице (на главной диагонали матрицы коэффициентов СЛАУ встретится нулевой элемент).
Следует проверять условие n < N на этапе ввода степени полинома.

3. Получить формулу для коэффициента полинома a_0 при степени полинома n=0. Какой смысл имеет величина, которую представляет данный коэффициент?

В данном случае система представляет собой одно уравнение с одной неизвестной.

$$\sum_{i=0}^{N} p_i \cdot a_0 = \sum_{i=0}^{N} p_i y_i \Rightarrow a_0 = \frac{\sum_{i=0}^{N} p_i y_i}{\sum_{i=0}^{N} p_i} \quad \Big| : \sum_{i=0}^{N} p_i$$

Получаем выражение, равное математическому ожиданию: $a_0 = \sum_{i=0}^{N} p_i y_i$, где p_i – отношение веса p_i к сумме всех весов.

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

4. $n = N = 2$; $p_i = 1$.

$$|A| = \begin{vmatrix} p_0 + p_1 & p_0 x_0 + p_1 x_1 & p_0 x_0^2 + p_1 x_1^2 \\ p_0 x_0 + p_1 x_1 & p_0 x_0^2 + p_1 x_1^2 & p_0 x_0^3 + p_1 x_1^3 \\ p_0 x_0^2 + p_1 x_1^2 & p_0 x_0^3 + p_1 x_1^3 & p_0 x_0^4 + p_1 x_1^4 \end{vmatrix} = [p_i = 1] =$$

$$= \begin{vmatrix} 1+1 & x_0 + x_1 & x_0^2 + x_1^2 \\ x_0 + x_1 & x_0^2 + x_1^2 & x_0^3 + x_1^3 \\ x_0^2 + x_1^2 & x_0^3 + x_1^3 & x_0^4 + x_1^4 \end{vmatrix} = 2 \cdot ((x_0^2 + x_1^2)(x_0^4 + x_1^4) - (x_0^3 + x_1^3)^2) -$$

$$- (x_0 + x_1) \cdot ((x_0 + x_1)(x_0^4 + x_1^4) - (x_0^2 + x_1^2)(x_0^3 + x_1^3)) +$$

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

$$- 2x_0^3 x_1^3) - (x_0 + x_1) \cdot (x_0 x_1^4 + x_0^4 x_1 - x_0^2 x_1^3 - x_0^3 x_1^2) +$$

$$+ (x_0^2 + x_1^2)(x_0 x_1^3 + x_0^3 x_1 - 2x_0^2 x_1^2) = \cancel{2x_0^2 x_1^4} + \cancel{2x_0^4 x_1^2} -$$

$$- \cancel{4x_0^3 x_1^3} - \cancel{x_0^2 x_1^4} - \cancel{x_0^5 x_1} + \cancel{x_0^3 x_1^3} + \cancel{x_0^4 x_1^2} - \cancel{x_0 x_1^5} - \cancel{x_0^4 x_1^2} +$$

$$+ \cancel{x_0^2 x_1^4} + \cancel{x_0^3 x_1^3} + \cancel{x_0^3 x_1^3} + \cancel{x_0^5 x_1} - \cancel{2x_0^4 x_1^2} + \cancel{x_0 x_1^5} +$$

$$+ \cancel{x_0^3 x_1^3} - \cancel{2x_0^2 x_1^4} = 0$$

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.

5. $\varphi(x) = a_0 + a_1 x^m + a_2 x^n$; n, m - известны

По ЛНЗ функции x^0, x^m, x^n можно построить СЛАУ:

$$(x^0,x^0)a_0 + (x^0,x^m)a_1 + (x^0,x^n)a_2 = (y,x^0)$$
$$(x^m,x^0)a_0 + (x^m,x^m)a_1 + (x^m,x^n)a_2 = (y,x^m)$$
$$(x^n,x^0)a_0 + (x^n,x^m)a_1 + (x^n,x^n)a_2 = (y,x^n)$$

Вывести можно из усл. $\frac{\partial I}{\partial a_k} = 0$, где $I = ((y-\varphi),(y-\varphi))$.

$$((y-\varphi),(y-\varphi)) = (y,y) - 2\sum_{k=0}^{n} a_k (y,\varphi_k) + \sum_{k=0}^{n}\sum_{m=0}^{n} a_k a_m (\varphi_k,\varphi_m) = \min$$

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.

Подойдет перебор всех возможных комбинаций m и n (их конечное число, т. к. m и n ограничены сверху количеством заданных узлов).
Для каждого ищем значение I. В

Т.к. I = $\sum_{i=1}^{N} \rho_i [y(x_i) - \varphi(x_i)]^2 = min$, выбираем m и n, при которых значение I наименьшее.