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

|  |  |  |
| --- | --- | --- |
| **x** | **y** | **pi** |
| 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

1. *Веса точек разные.*

|  |  |  |
| --- | --- | --- |
| **x** | **y** | **pi** |
| 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

1. **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:

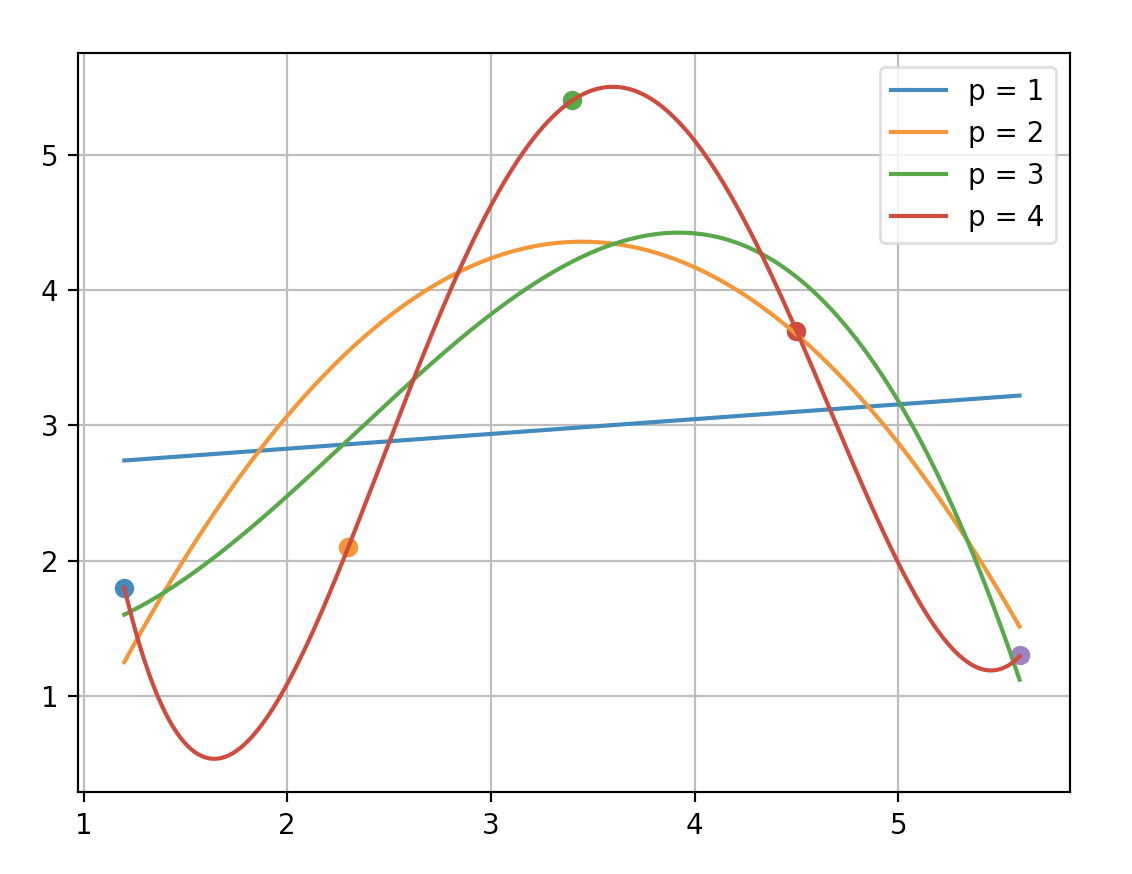
generate\_file()

t = read\_file()

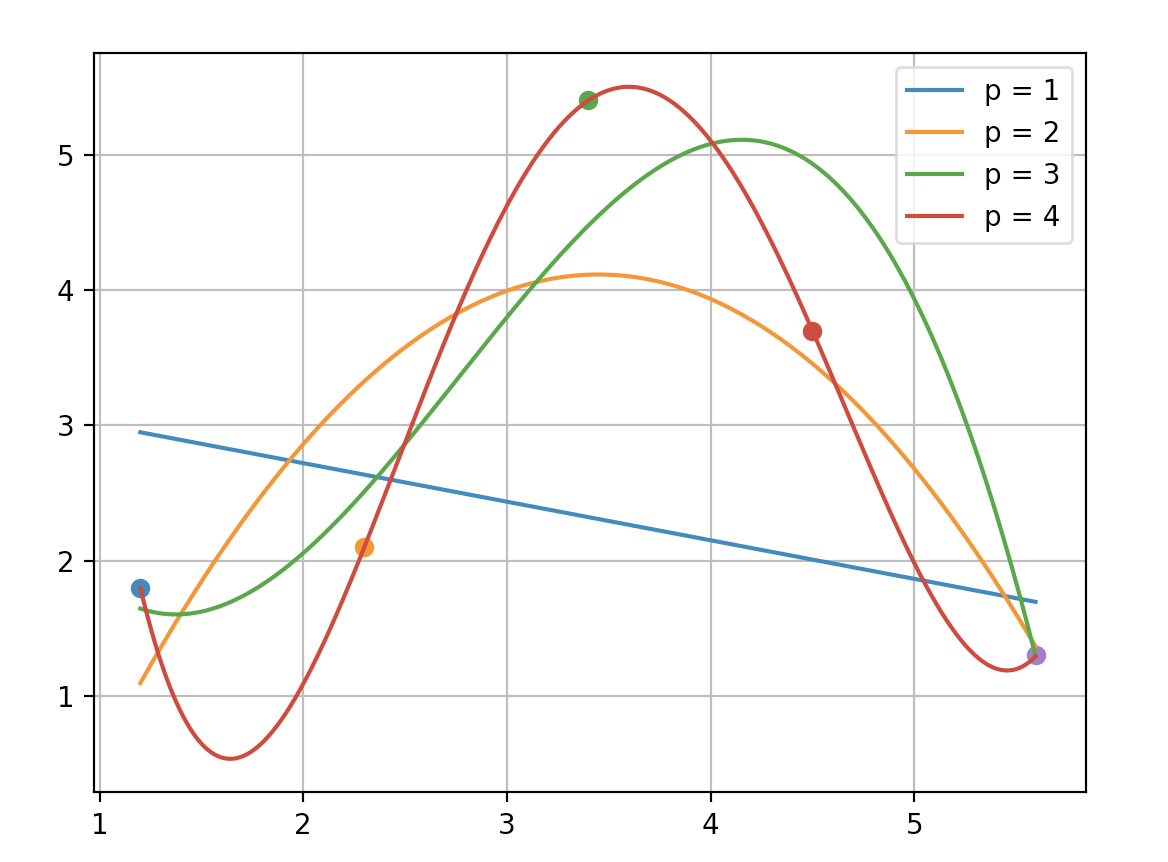
#n = int(input("степень полинома: "))

print\_graph(t[0], t[1], t[2])

1. **Результат работы программы**
2. *Таблица входных данных с одинаковыми весами* (**Таблица 1**):

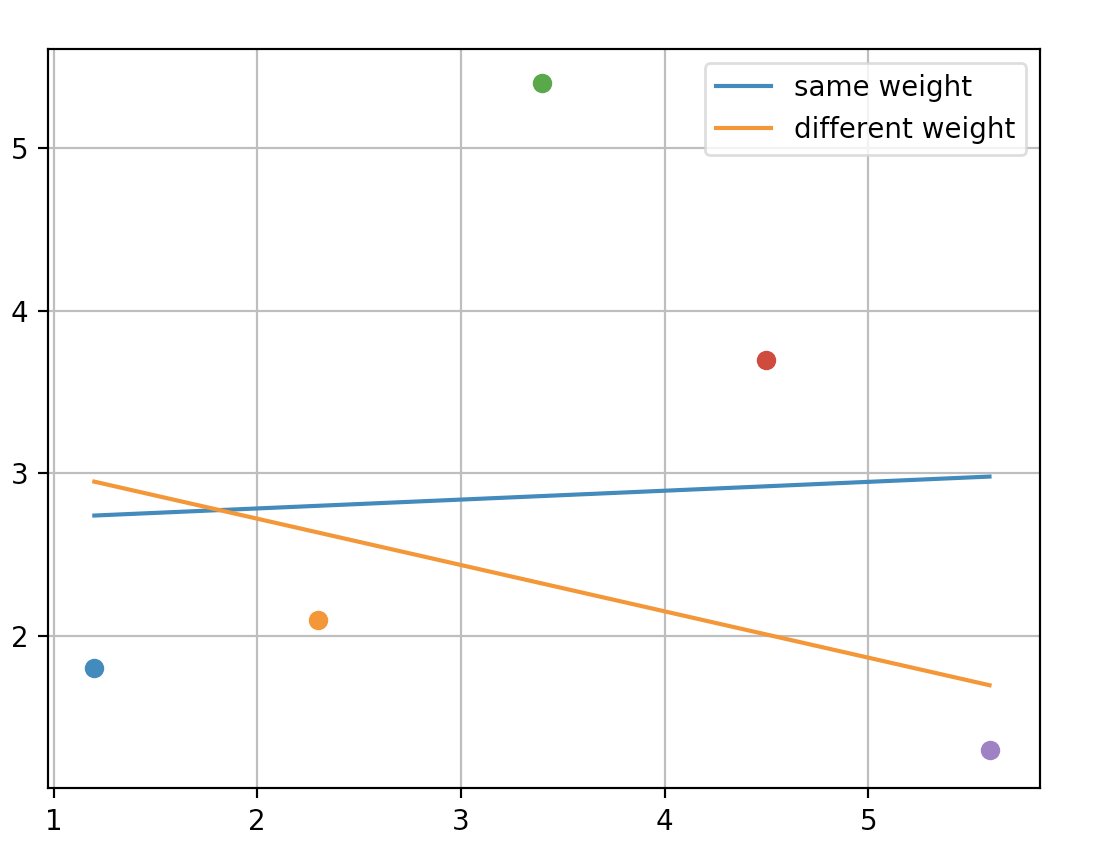
****

1. *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):

****

1. **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?***

(x0, x0)a = (y, x0)

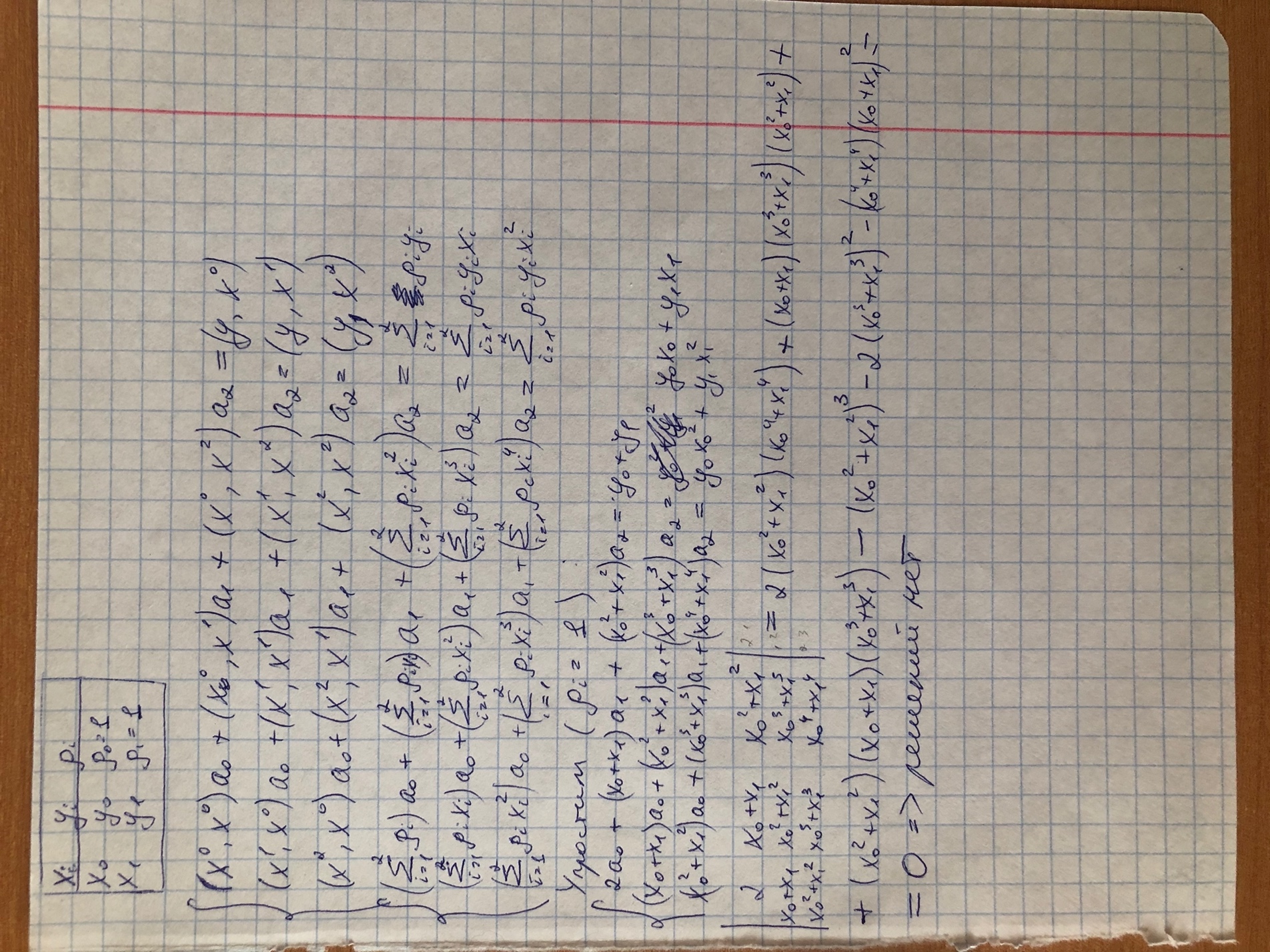
Sum(pi) \* a = Sum(pi \* yi)

a = Sum(pi \* yi) / Sum(pi), 0 <= i < N

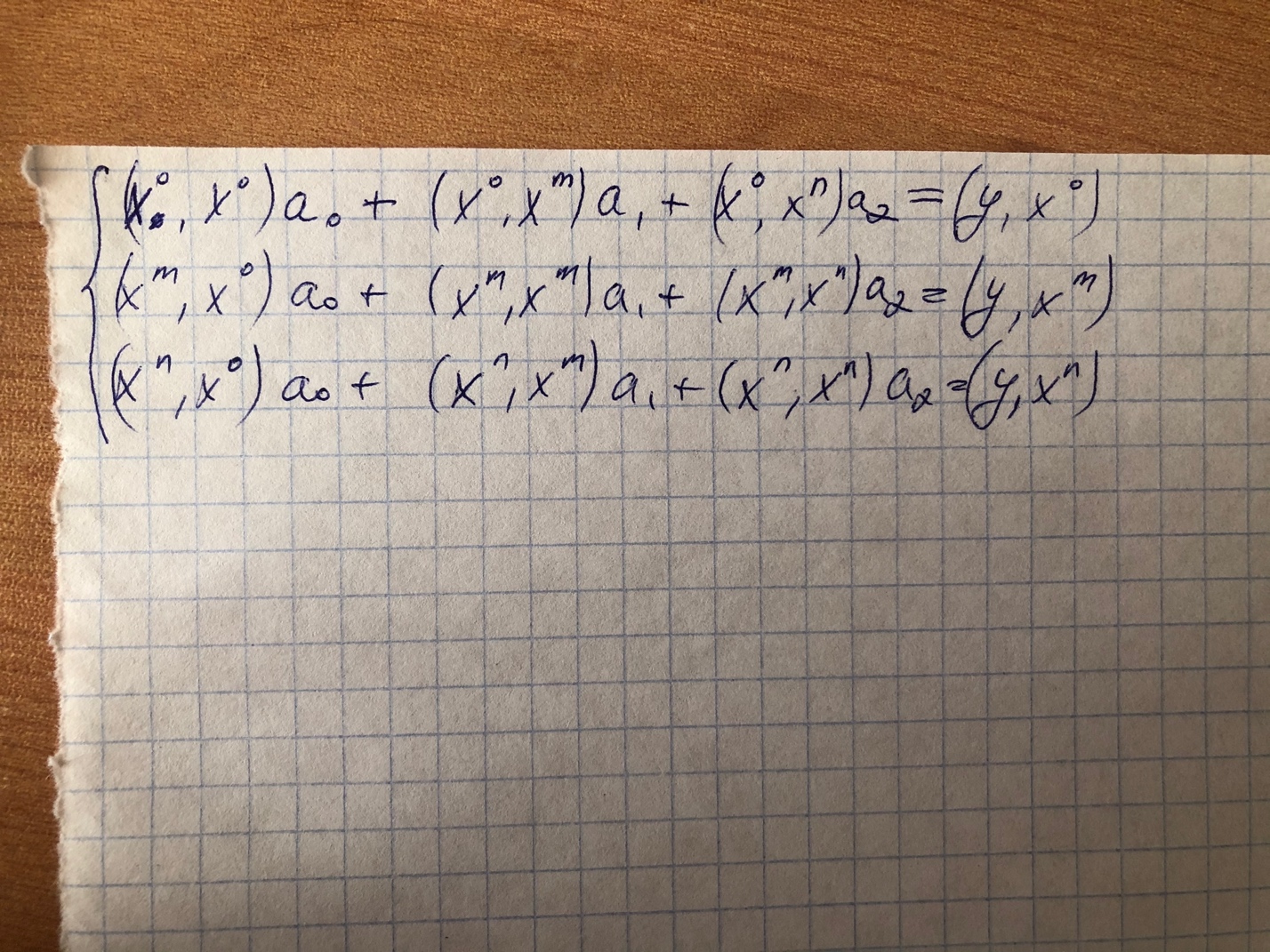
pi – вес точки, N – количество точек

Коэффициент представляет собой математическое ожидание.

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

**

***5. Построить СЛАУ при выборочном задании степеней аргумента полинома (x) = a0 + a1 xm + a2 xn, причем степени n и m в этой формуле известны.***



***6. Предложить схему алгоритма решения задачи из вопроса 5, если степени n и m подлежат определению наравне с коэффициентами ak, 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 , а затем выбрать ту пару, для которой справедливо:

