	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: Наилучшее среднеквадратичное приближение

Студент: Миронов Г.А.

Группа: ИУ7-43Б

Оценка (баллы): _________

Преподаватель: Градов В.М.

Москва.
2020 г.

Задание

1. Задана матрица значений функции вида X, F(X). С помощью интерполяции, используя метод полинома сплайнов, найти приближенное значение функции от введенного X

Входные данные

1. Таблица координат
2. Степень искомого полинома

Описание алгоритма

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

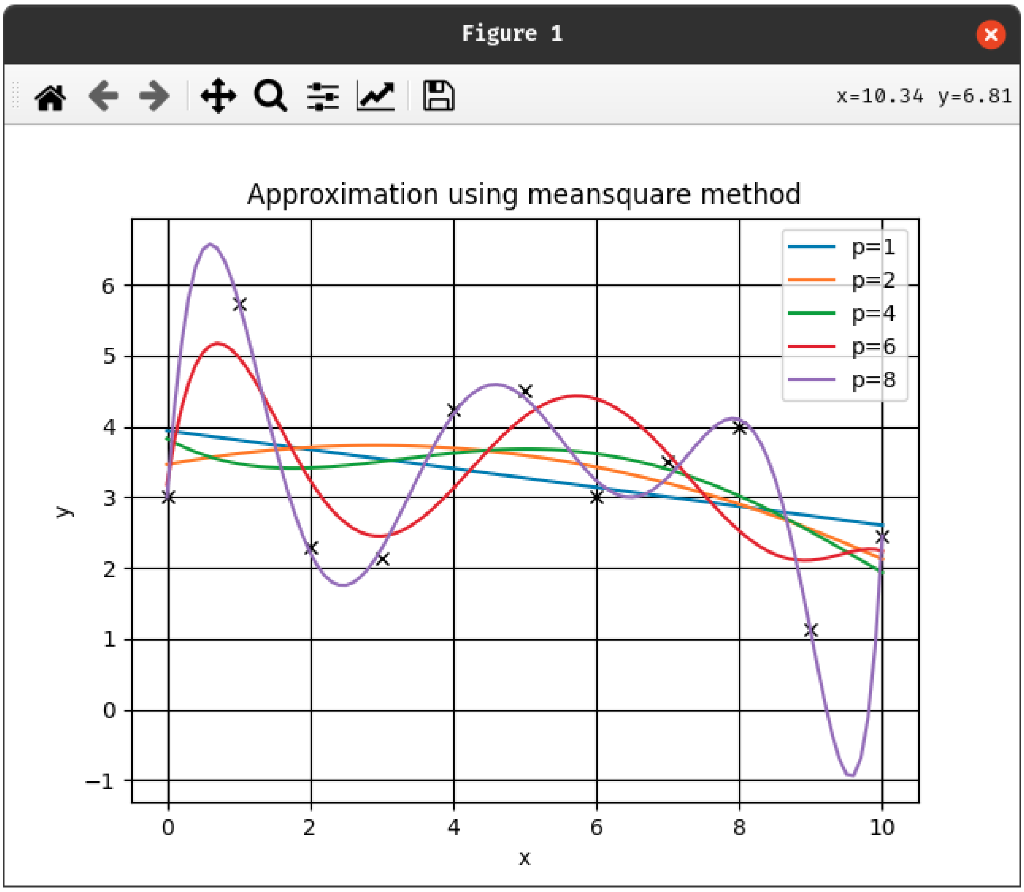
$$\sum_{m=0}^{n}(x^k, x^m)\ a_m = (y, x^k)\ ,\ 0 \le k \le n\ ,\ \text{где}\quad (x^k, x^m) = \sum_{i=1}^{N} \rho_i x_i^{k+m},\quad (y, x^k) = \sum_{i=1}^{N} \rho_i y_i x_i^k$$

3. В результате решения СЛАУ находятся коэффициенты полинома

Результаты аппроксимации

Исходная таблица (Таблица 1)

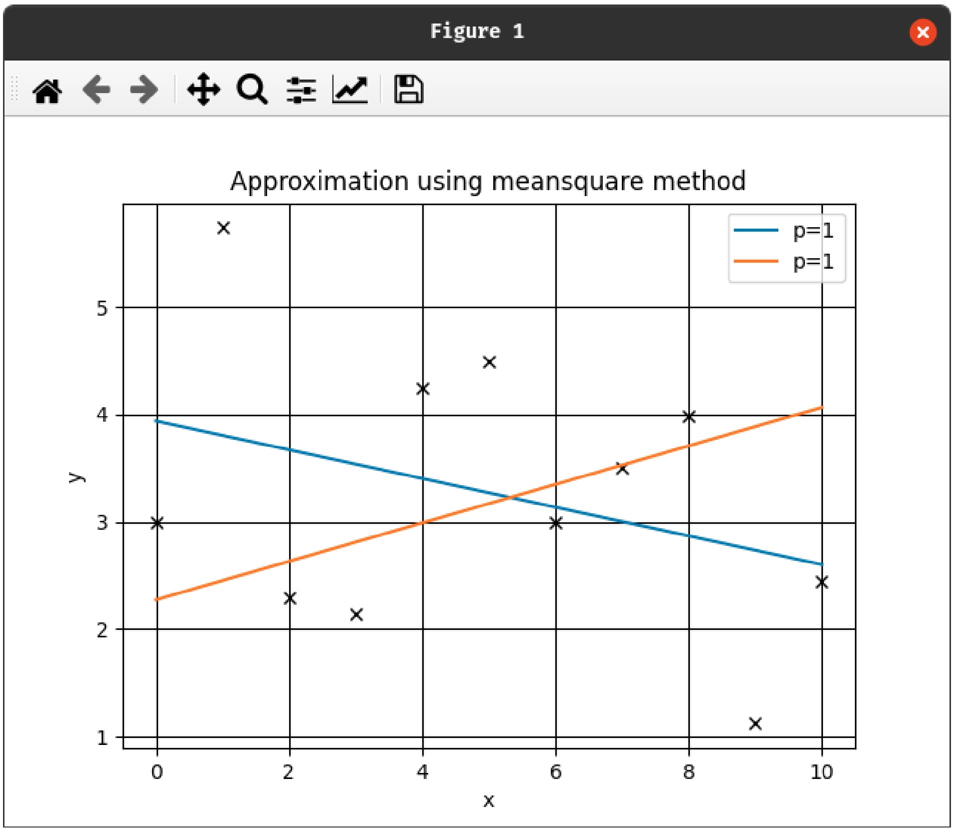
X	Y	Weight
0.00	3.00	1
1.00	5.74	1
2.00	2.30	1
3.00	2.14	1
4.00	4.24	1
5.00	4.50	1
6.00	3.00	1
7.00	3.50	1
8.00	3.98	1
9.00	1.12	1
10.00	2.45	1



Преобразованная таблица (Таблица 2)

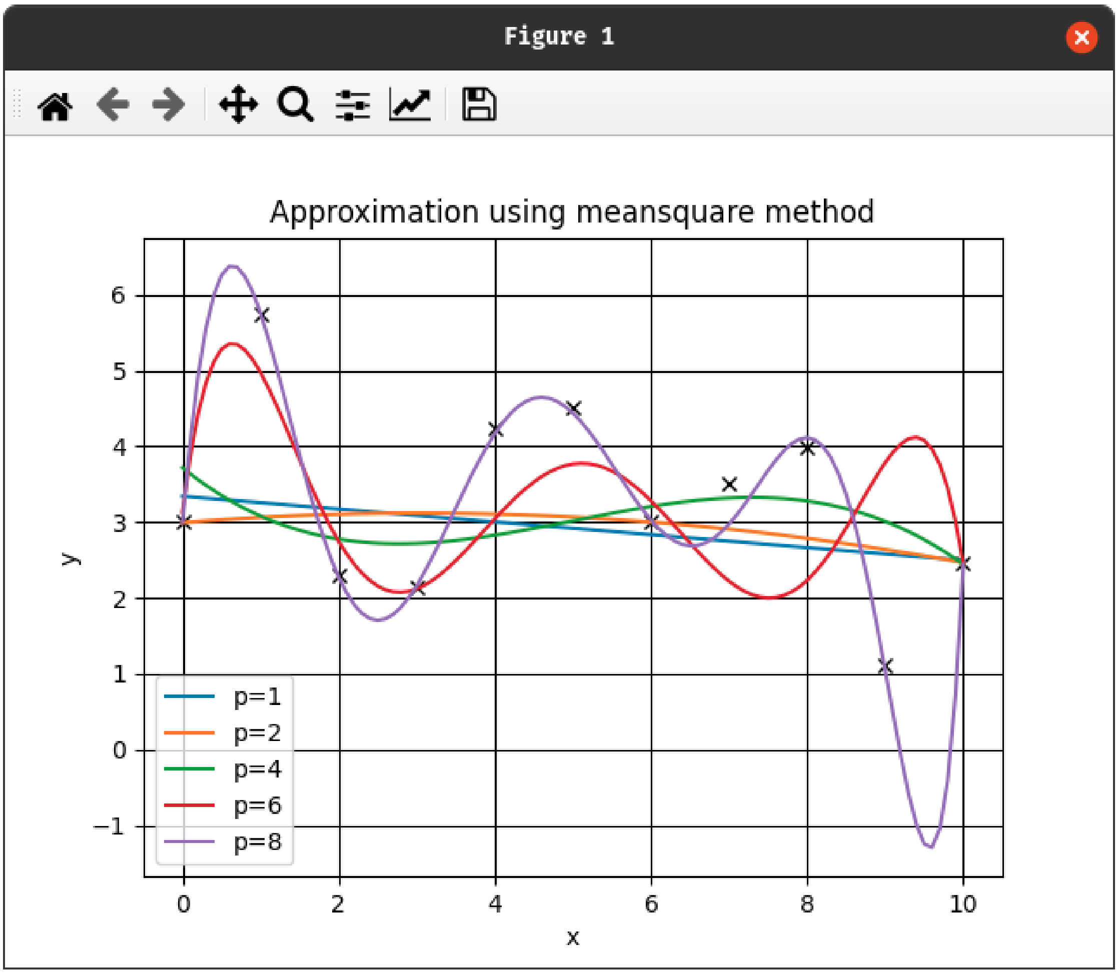
X	Y	Weight 1 (blue)	Weight 2 (yellow)
0.00	3.00	1	1
1.00	5.74	1	1
2.00	2.30	1	20
3.00	2.14	1	1
4.00	4.24	1	1
5.00	4.50	1	1
6.00	3.00	1	1
7.00	3.50	1	1
8.00	3.98	1	20
9.00	1.12	1	1
10.00	2.45	1	1

Графическое представление сравнения аппроксимации



Исходная таблица (Таблица 3)

X	Y	Weight
0.00	3.00	4
1.00	5.74	3
2.00	2.30	7
3.00	2.14	11
4.00	4.24	1
5.00	4.50	5
6.00	3.00	23
7.00	3.50	1
8.00	3.98	2
9.00	1.12	1
10.00	2.45	99



Код программы

Листинг 1. `utils.py`

```
from meansquare import Dot


def read_dots(fname: str) -> list[Dot]:
    dots = []

    with open(fname) as fin:
        for line in fin.readlines():
            dots += [Dot(*list(map(float, line.split()[:3])))]

    return dots


def print_dots(dots: list[Dot]) -> None:
    print("{:^8} {:^8} {:^8}".format("X", "Y", "Weight"))
    for i in dots:
        print("{:<8.2f} {:<8.2f} {:<8.2f}".format(i.x, i.y, i.weight))


def read_polynom_degree() -> int:
    return int(input())


def print_matrix(mat: list[list[float]]) -> None:
    for line in mat:
        print(('[' + ', '.join(["{:8.2f}"] * len(line)) + ']').format(*line))
```

Листинг 2. meansquare.py

```python
from __future__ import annotations

import numpy as np


class Dot(object):
   x: float
   y: float
   weight: float

   def __init__(self, _x: float, _y: float, _w: float):
       self.x, self.y, self.weight = _x, _y, _w


class SLAE(object):
   mat: list[list[float]]
   n: int

   def build(self, ds: list[Dot], _n: int) -> SLAE:
       self.n = _n
       self.mat = [[0 for i in range(self.n + 2)] for i in range(self.n + 1)]

       for i in range(self.n + 1):
           for j in range(self.n + 1):
               slae_coeffs = 0.0
               expanded_coeff = 0.0
               for k in range(len(ds)):
                   slae_coeffs += ds[k].weight * \
                                  (ds[k].x ** i) * \
                                  (ds[k].x ** j)
                   expanded_coeff += ds[k].weight * ds[k].y * (ds[k].x ** i)

               self.mat[i][j] = slae_coeffs
               self.mat[i][self.n + 1] = expanded_coeff

       return self

   def solve(self) -> list[list[float]]:
       for i in range(self.n + 1):
           for j in range(self.n + 1):
               if i == j:
                   continue

               sub_coeff = self.mat[j][i] / self.mat[i][i]
               for k in range(self.n + 2):
                   self.mat[j][k] -= sub_coeff * self.mat[i][k]

       for i in range(self.n + 1):
           divider = self.mat[i][i]
```

```python
            for j in range(self.n + 2):
                self.mat[i][j] /= divider

        return self.mat


class Approx(object):
    def __init__(self):
        self.coeffs = []

    def get_coeffs(self, mat: list[list[float]]) -> Approx:
        self.coeffs = [mat[i][len(mat)] for i in range(len(mat))]

        return self

    def build(self, ds: list[Dot]) -> list[Dot]:
        dots = []

        for i in np.arange(ds[0].x, ds[-1].x, 0.1):
            d = Dot(i, 0, 0)

            for j in range(len(self.coeffs)):
                d.y += d.x ** j * self.coeffs[j]

            dots += [d]

        return dots
```

Листинг 3. `main.py`

```
from sys import argv

import matplotlib.pyplot as plt

from utils import *
from meansquare import *


def plot(dots: list[Dot], approx: list[tuple[int, list[Dot]]]) -> None:
   x, y = [p.x for p in dots], [p.y for p in dots]

   plt.clf()

   plt.title("Approximation using meansquare method")
   plt.xlabel("x")
   plt.ylabel("y")
   plt.grid(which='minor', color='k', linestyle=':')
   plt.grid(which='major', color='k')

   plt.plot(x, y, "xk")

   for a in approx:
       plt.plot([p.x for p in a[1]], [p.y for p in a[1]],
                label="p=" + str(a[0]))

   plt.legend()
   plt.show()


def main():
   dots = read_dots(argv[1])

   print("Table loaded from file\n")
   print_dots(dots)

   print("\nEnter polynom degree")

   approxs = []
   for deg in [1, 2, 4, 6, 8]:
       slae = SLAE().build(dots, deg)

       print("\nSLAE to solve\n")
       print_matrix(slae.mat)

       slae = slae.solve()

       print("\nSolved SLAE\n")
       print_matrix(slae)
       print()
```

```
        approxs.append((deg, Approx().get_coeffs(slae).build(dots)))

    plot(dots, approxs)


if __name__ == "__main__":
    main()
    # cmp_main()
```

Контрольные вопросы

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

   Для однозначного определения полинома N − 1 степени достаточно N точек, что означает, что полином будет построен таким образом, что его график будет проходить через все табличные точки. При такой конфигурации в выражении

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

   часть выражения, находящаяся в скобках, обращается в 0, что означает, что нет зависимости от весов (при любых заданных весах, значение полинома будет минимальным в случае прохода через табличные 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?

   Программа работать будет, но её работа будет некорректна, так как начиная со случая n = N определитель СЛАУ, которую необходимо решить, будет тождественно равен 0 (уравнения СЛАУ не будут линейно-независимыми). Из-за применения метода Гаусса-Жордана, аварийная остановка программы может произойти при приведении диагональной матрицы к единичной, где может произойти деление на ноль. Анализ можно проводить при решении СЛАУ или же на начальном этапе (ввод степени полинома).

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

   Полученная формула:

$$a_0 = \frac{\sum_{i=1}^{N} \rho_i y_i}{\sum_{i=1}^{N} \rho_i}$$

   Данную формулу можно преобразовать делением числителя и знаменателя на сумму весов, из чего получим ***математическое ожидание***:

$$M[X] = \sum_{i=1}^{\infty} x_i \rho_i$$

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

   Зададим таблицу точек:

x_i	y_i	ρ_i
x_0	y_0	1
x_1	y_1	1

   Тогда имеем СЛАУ вида:

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

$$\Delta = (x_0^2 + x_1^2)(x_0^4 + x_1^4) + (x_0 + x_1)(x_0^3 + x_1^3)(x_0^2 + x_1^2) +$$
$$(x_0^2 + x_1^2)(x_0 + x_1)(x_0^3 + x_1^3) - (x_0^2 + x_1^2)(x_0^2 + x_1^2)(x_0^2 + x_1^2) -$$
$$(x_0^3 + x_1^3)(x_0^3 + x_1^3) - (x_0 + x_1)(x_0 + x_1)(x_0^4 + x_1^4) = 0$$

   Так как Δ = 0, система решений не имеет, что было упомянуто в ответе на вопрос 2.

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

   Можно поступить следующим образом.
   1. Перебрать все возможные пары n и m
   2. Для каждой ищем все коэффициенты a_i, а также ошибку.
   3. Среди всех таких наборов выбираем тот, у которого ошибка будет минимальной