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 Информатики и систем управления

КАФЕДРА Теоретической информатики и компьютерных технологий

Лабораторная работа № 5

«Метод наименьших квадратов. Аппроксимация алгебраическими многочленами»

по курсу «Численные методы»

Выполнила:

студент группы ИУ9-61Б

Яровикова Анастасия

Проверила:

Домрачева А. Б.

Москва, 2023

1. Цель

Целью данной работы является изучение приближения заданной функции путем аппроксимации алгебраическими многочленами, применяя метод наименьших квадратов, и вычисление значения аппроксимирующего многочлена в серединах частичных отрезков между узловыми точками.

Постановка задачи

Дано: функция $y = f(x)$ задана конечным набором точек

$y_i = f(x_i),\ i = \overline{0, n}$ на отрезке $[a, b], a = x_0,\ b = x_n,\ x_i = a + ih, h = \frac{(b-a)}{n}$

x_i	x_0	x_1	...	x_{n-1}	x_n
y_i	y_0	y_1	...	y_{n-1}	y_n

Задание:

- Аппроксимировать функцию по методу наименьших квадратов (МНК) многочленом третьей степени ($m = 4$);
- Найти матрицу A и столбец b;
- Найти набор коэффициентов $\lambda_1, \lambda_2, \lambda_3, \lambda_4$;
- Найти среднеквадратичное отклонение (СКО) Δ и относительную ошибку δ;
- Найти значения аппроксимирующего многочлена $z(x)$ в средних точках отрезков между узловыми точками.

Индивидуальный вариант:

$y = f(x)$ задана конечным набором точек:

x_i	1	1.5	2	2.5	3	3.5	4	4.5	5
y_i	3.33	2.30	1.60	1.27	1.18	0.99	1.41	0.80	1.12

2. Основные теоретические сведения

Значения приближаемой функции $y = f(x)$ заданы лишь в узлах $(x_i, y_i), i = 0, \dots, n$. Необходимо решить задачу аппроксимации: найти гладкую аналитически заданную функцию $z(x)$, доставляющую наименьшее значение величине

$$\text{СКУ} = \sqrt{\sum_{k=0}^{n} (z(x_k) - y_k)^2} = \sqrt{\sum_{k=0}^{n} \varepsilon_k^2}.$$

Данную величину называют среднеквадратичным уклонением (СКУ) функции $z(x)$ от системы узлов $(x_i, y_i), i = 0, \dots, n$, а описанный подход к решению задачи приближения функции – методом наименьших квадратов (МНК)

Как правило, $z(x)$ отыскивают в виде линейной комбинации заранее заданных функций:

$$z(x) = \lambda_1 \varphi_1(x) + \dots + \lambda_m \varphi_m(x).$$

Параметры λ_i, $i = 1, \dots, m$ являются решениями линейной системы наименьших квадратов

$$A\lambda = b,$$

где λ – столбец параметров λ_i, $A = (a_{ij})$ – симметричная положительно определенная матрица с коэффициентами $a_{ij} = \sum_{k=0}^{n} \varphi_i(x_i)\varphi_j(x_k)$,
b – столбец правой части системы, $b_i = \sum_{k=0}^{n} \varphi_i(x_k) y_k$, $i, j = 1, \dots, m$.

Таким образом, система МНК имеет единственное решение $\lambda_1^*, \dots, \lambda_m^*$, дающее СКУ наименьшее значение (для всех функций данного вида). Система решается методом квадратного корня.

Если приближаемая функция достаточно гладкая, хотя вид ее и неизвестен, аппроксимирующую функцию нередко ищут в виде алгебраического многочлена

$$z(x) = \lambda_1 + \lambda_2 x + \dots + \lambda_m x^{m-1}.$$

Тогда $\varphi_i = x^{i-1}$ и элементы матрицы A получают по формулам:

$$a_{ij} = \sum_{k=0}^{n} x_k^{i+j-1},$$

а свободные члены –

$$b_i = \sum_{k=0}^{n} y_k x_k^{i+j-1}, \qquad i,j = 1,\dots,m.$$

Абсолютной погрешностью аппроксимации выступает СКО:

$$\Delta = \frac{\text{СКУ}}{\sqrt{n}} = \frac{1}{\sqrt{n}} \sqrt{\sum_{k=0}^{n} (y_k - \lambda_1 - \lambda_2 x_k - \dots - \lambda_m x_k^{m-1})^2},$$

относительная ошибка:

$$\delta = \frac{\Delta}{||y||} = \frac{\Delta}{\sqrt{\sum_{k=0}^{n} y_k^2}}$$

3. Реализация

Листинг 1. Метод наименьших квадратов. Аппроксимация алгебраическими многочленами

```
import pandas as pd
import numpy as np
from tabulate import tabulate

m = 4
l = 1
r = 5
n = 8

h = (r - l) / n

def f(x):
    return np.exp(x)

def find_lambd2(A, b):
    return (np.linalg.inv(A).dot(b.T)).T

def find_lambd(A, b):
    T = np.zeros((m,m))
    x = np.empty(m)
    y = np.empty(m)

    # T
    for i in range(m):
        for j in range(i):
            T[i][j] = (A[i][j] - sum([T[i][k] * T[j][k] for k in range(j)])) / T[j][j]
        T[i][i] = np.sqrt(A[i][i] - sum(T[i][k] ** 2 for k in range(i)))

    # прямой ход
    for i in range(m):
```

```
        y[i] = (b[i] - sum([T[i,k]*y[k] for k in range(i)])) / T[i,i]

    # обратный ход
    for i in range(m-1, -1, -1):
        x[i] = (y[i] - sum([T[k,i]*x[k] for k in range(i+1, m)])) / T[i,i]

    return x

xs = np.linspace(l, r, n + 1, True)
mids = np.linspace(l + 0.5 * h, r - 0.5 * h, n, True)
ys = np.array([3.33, 2.30, 1.60, 1.27, 1.18, 0.99, 1.41, 0.80, 1.12])

A = np.empty((m,m))
b = np.empty(m)
A = np.array([[sum(xs[k] ** (i+j) for k in range(0,n+1))  for j in range(0, m)] for i
in range(0,m)])
b = np.array([sum(ys[k] * (xs[k] ** i) for k in range(0,n+1))  for i in range(0, m)])
print("\nA:", A)
print("\nb:", b)

lambd = find_lambd(A,b)
print("\nλ:", lambd)

def z(x):
    return sum([lambd[i] * x**i for i in range(m)])

D = sum([(ys[k] - z(xs)[k]) for k in range(m+1)])**2
D = np.sqrt(D) / np.sqrt(n)
print("\nСКО Δ:", D)

d = sum([ys[k]**2 for k in range(n+1)])
d = D / np.sqrt(d)
print("\nотн. погрешность δ:", d)
print()

for k in range(n + 1):
    print("x:", xs[k], "  f(x):", ys[k], "  z(x):", z(xs)[k], "  |f - z|:", abs(ys[k]
- z(xs)[k]))
    if k != n:
        print("x:", mids[k], "              z(x):", z(mids)[k])
```

4. Результаты

Для заданной функции найдены:

а) матрица A и столбец b;

```
a:
 [[9.00000000e+00 2.70000000e+01 9.60000000e+01 3.78000000e+02]
 [2.70000000e+01 9.60000000e+01 3.78000000e+02 1.58325000e+03]
 [9.60000000e+01 3.78000000e+02 1.58325000e+03 6.90075000e+03]
 [3.78000000e+02 1.58325000e+03 6.90075000e+03 3.09125625e+04]]

b:
 [[ 14.    ]
 [ 35.    ]
 [112.35  ]
 [421.1825]]
```

б) коэффициенты $\lambda_1, \lambda_2, \lambda_3, \lambda_4$;

```
λ:
 [[ 6.755     ]
 [-4.53247475]
 [ 1.1969697 ]
 [-0.1040404 ]]
```

в) СКО и относительная ошибка;

```
СКО Δ: 0.15415294779254587

отн. погрешность δ: 0.029773462710402647
```

г) значения аппроксимирующего многочлена в узловых точках и в серединах отрезков между узловыми точками.

```
x: 1.0   y: 3.33   y*: 3.315454545454374   |y - y*|: 0.014545454546260217
x: 1.25            y*: 2.7564678030295844
x: 1.5   y: 2.3    y*: 2.29833333333269   |y - y*|: 0.001666666673099684
x: 1.75            y*: 1.931297348484268
x: 2.0   y: 1.6    y*: 1.6456060606055303   |y - y*|: 0.045606060605530185
x: 2.25            y*: 1.4315056818176881
x: 2.5   y: 1.27   y*: 1.2792424242419527   |y - y*|: 0.009242424241952651
x: 2.75            y*: 1.1790624999995356
x: 3.0   y: 1.18   y*: 1.1212121212116486   |y - y*|: 0.058787878788351344
x: 3.25            y*: 1.0959374999995029
x: 3.5   y: 0.99   y*: 1.0934848484843096   |y - y*|: 0.10348484848430961
x: 3.75            y*: 1.1041003787872805
x: 4.0   y: 1.41   y*: 1.1180303030296272   |y - y*|: 0.29196969697037267
x: 4.25            y*: 1.125520833332561
x: 4.5   y: 0.8    y*: 1.116818181817294   |y - y*|: 0.31681818181729393
x: 4.75            y*: 1.0821685606050355
x: 5.0   y: 1.12   y*: 1.0118181818169987   |y - y*|: 0.10818181818300143
```

5. Вывод

В ходе выполнения лабораторной работы был изучен и реализован метод приближения функции с помощью аппроксимации алгебраическими многочленами с применением метода наименьших квадратов, был найден аппроксимирующий многочлен и найдены его значения в серединах отрезков между заданными узловыми точками. В результате тестирования была получена гладкая аппроксимирующая функция, доставляющая наименьшее значение среднеквадратичному уклонению, были найдены абсолютная погрешность аппроксимации $\Delta = 0.154153$ и относительная погрешность аппроксимации $\delta = 0.029774$.