

Лабораторная работа №8

«Сглаживание табличных данных»

Задание. В файле данных, полученном в результате весового аэродинамического эксперимента, в четырех столбцах размещены угол атаки α (в градусах), безразмерные коэффициенты сил в связанной системе координат, коэффициент момента. Например:

```
"Al" "Cx" "Cy" "mz"
-9 0.584 -0.117 0.167
-8 0.594 -0.064 0.184
...
```

Считать данные из файла, преобразовать в скоростную систему координат и сгладить многочленом заданной степени m . Сглаживание проводить с помощью самописного кода, а также модуля `scipy.optimize` и Excel. Результаты сравнить между собой. По исходным и сглаженным данным построить графики зависимостей от α коэффициентов лобового сопротивления c_{xa} , подъемной силы c_{ya} и момента m_z , аэродинамического качества K . С помощью сглаживающих зависимостей найти параметр галопирования A . Варианты исходных данные приведены в таблице:

Вариант	1	2	3	4	5	6	7	8	9
m	3	4	5	3	4	5	3	4	5
Вариант	10	11	12	13	14	15	16	17	18
m	3	4	5	3	4	5	3	4	5
Вариант	20	21	22	23	24	25	26	27	28
m	3	4	5	3	4	5	3	4	5
Вариант	29	30							
m	3	4							

Содержание отчета:

1. Постановка задачи (что задано, что нужно найти).
2. Краткое описание используемого метода.
3. Текст расчетной программы.
4. Листинг и графический вывод результатов.
5. Формулы аппроксимирующих зависимостей.
6. Решение задачи в Excel.
7. Выводы.

Указания. Переход от связанной к скоростной системе координат осуществляется по формулам

$$c_{xa} = c_x \cos \alpha + c_y \sin \alpha,$$

$$c_{ya} = c_y \cos \alpha - c_x \sin \alpha,$$

где α — угол атаки (столбец "Al" файла исходных данных), c_x и c_y — аэродинамические коэффициенты в связанной системе координат, c_{xa} и c_{ya} — коэффициенты лобового сопротивления и подъемной силы соответственно. Аэродинамическое качество и параметр галопирования поперечного сечения

рассчитываются по формулам $K = c_{ya}/c_{xa}$ и $A = \left(\frac{dc_{ya}}{d\alpha} + c_{xa}\right)\Big|_{\alpha=0}$ соответственно. Для определения $\frac{dc_{ya}}{d\alpha}(\alpha = 0)$ и $c_{xa}(\alpha = 0)$ используются соответствующие сглаживающие зависимости.

В задаче сглаживания (аппроксимации) табличных данных $(x_i, f_i)_{i=1,\dots,N}$ строится сглаживающая функция, например, многочлен 3-й степени,

$$P(x) = a_1 + a_2x + a_3x^2 + a_4x^3, \quad (1)$$

имеющий в узлах таблицы x_i минимальное отклонение от заданных значений f_i . В i -й точке функция $P(x)$ отклоняется от значения f_i на величину $P(x_i) - f_i$. Суммируя квадраты отклонений по всем точкам $i = 1, \dots, N$, находим суммарную невязку:

$$\Phi(a_1, a_2, a_3, a_4) = \sum_{i=1}^n (P(x_i) - f_i)^2 = \sum_{i=1}^n (a_1 + a_2x_i + a_3x_i^2 + a_4x_i^3 - f_i)^2.$$

Потребуем, чтобы $\Phi(a_1, a_2, a_3, a_4) \rightarrow \min$ (метод наименьших квадратов, МНК). Это будет выполняться, если

$$\frac{\partial \Phi}{\partial a_k} = 2 \sum_{i=1}^n x_i^{k-1} (a_1 + a_2x_i + a_3x_i^2 + a_4x_i^3 - f_i) = 0, \quad k = 1, \dots, 4.$$

Отсюда получаем систему линейных алгебраических уравнений (СЛАУ) относительно искомых коэффициентов a_k :

$$\begin{cases} N \cdot a_1 + \sum_{i=1}^N x_i \cdot a_2 + \sum_{i=1}^N x_i^2 \cdot a_3 + \sum_{i=1}^N x_i^3 \cdot a_4 = \sum_{i=1}^N f_i \\ \sum_{i=1}^N x_i \cdot a_1 + \sum_{i=1}^N x_i^2 \cdot a_2 + \sum_{i=1}^N x_i^3 \cdot a_3 + \sum_{i=1}^N x_i^4 \cdot a_4 = \sum_{i=1}^N f_i \cdot x_i \\ \sum_{i=1}^N x_i^2 \cdot a_1 + \sum_{i=1}^N x_i^3 \cdot a_2 + \sum_{i=1}^N x_i^4 \cdot a_3 + \sum_{i=1}^N x_i^5 \cdot a_4 = \sum_{i=1}^N f_i \cdot x_i^2 \\ \sum_{i=1}^N x_i^3 \cdot a_1 + \sum_{i=1}^N x_i^4 \cdot a_2 + \sum_{i=1}^N x_i^5 \cdot a_3 + \sum_{i=1}^N x_i^6 \cdot a_4 = \sum_{i=1}^N f_i \cdot x_i^3, \end{cases} \quad (2)$$

или

$$B\vec{a} = \vec{c},$$

где B – симметричная положительно определенная матрица. Решение (2) можно получить с помощью одного из методов решения СЛАУ. Аналогично определяются коэффициенты сглаживающего многочлена 4-й и т.д. степеней.

Ниже приведен текст и результаты выполнения Python-программы, реализующей МНК на основе многочлена (1), при этом СЛАУ (2) решается с помощью функции `solve` модуля `np.linalg`. Для проверки и сравнения также используется функция `curve_fit` модуля `scipy.optimize`.

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
```

```

xf=np.loadtxt('tab.dat',skiprows=1) # skiprows=1 – для сброса заголовков 1-й строки
print('xf=',xf)
x=xf[:,0]
f=xf[:,1]
print('x=',x)
print('f=',f)
N=len(x)
print('N=',N)

#B=np.array([[N, np.sum(x), np.sum(x**2), np.sum(x**3)], - самостоятельно!
print('B=',B)
c=np.array([np.sum(f), np.sum(f*x), np.sum(f*x**2), np.sum(f*x**3)])
print('c=',c)
a=np.linalg.solve(B,c) # Решение СЛАУ
print('МНК:')
print('a=',a)
y=np.polyval([a[3],a[2],a[1],a[0]], x)
resid=np.sum((y-f)**2)
print(f'невязка={resid:.3e}')

def func1(x,a0,a1,a2,a3):
    return a3*x**3+a2*x**2+a1*x+a0
popt1,pcov1=curve_fit(func1,x,f)
print('SCIPY:')
print('a=',popt1)
resid=np.sum((func1(x,*popt1)-f)**2)
print(f'невязка={resid:.3e}')

dpi = 300
plt.figure(dpi=dpi)
plt.plot(x,f,'ro',x,y)
plt.xlabel('x')
plt.ylabel('y')
plt.grid(True)
plt.legend(['Данные', 'МНК'])
plt.savefig('MNK.png',dpi=dpi)
plt.show()

xf= [[-2.   -0.276 ]
 [-1.8  -0.291 ]
 [-1.6  -0.282 ]
 [-1.4  -0.293 ]
 [-1.2  -0.26  ]
 [-1.   -0.265 ]
 [-0.8  -0.207 ]
 [-0.6  -0.188 ]
 [-0.4  -0.124 ]
 [-0.2  -0.0553]
 [ 0.   0.007 ]
 [ 0.2  0.0558]
 [ 0.4  0.139 ]
 [ 0.6  0.162 ]
 [ 0.8  0.235 ]]
x=[-2. -1.8 -1.6 -1.4 -1.2 -1. -0.8 -0.6 -0.4 -0.2  0.  0.2  0.4  0.6
 0.8]
f=[-0.276 -0.291 -0.282 -0.293 -0.26 -0.265 -0.207 -0.188 -0.124
-0.0553 0.007 0.0558 0.139 0.162 0.235 ]
N= 15
B= [[ 15.   -9.   16.6  -23.4 ]
 [-9.   16.6  -23.4  41.0992]
 [ 16.6  -23.4  41.0992 -70.248 ]
 [-23.4  41.0992 -70.248  126.93088]]

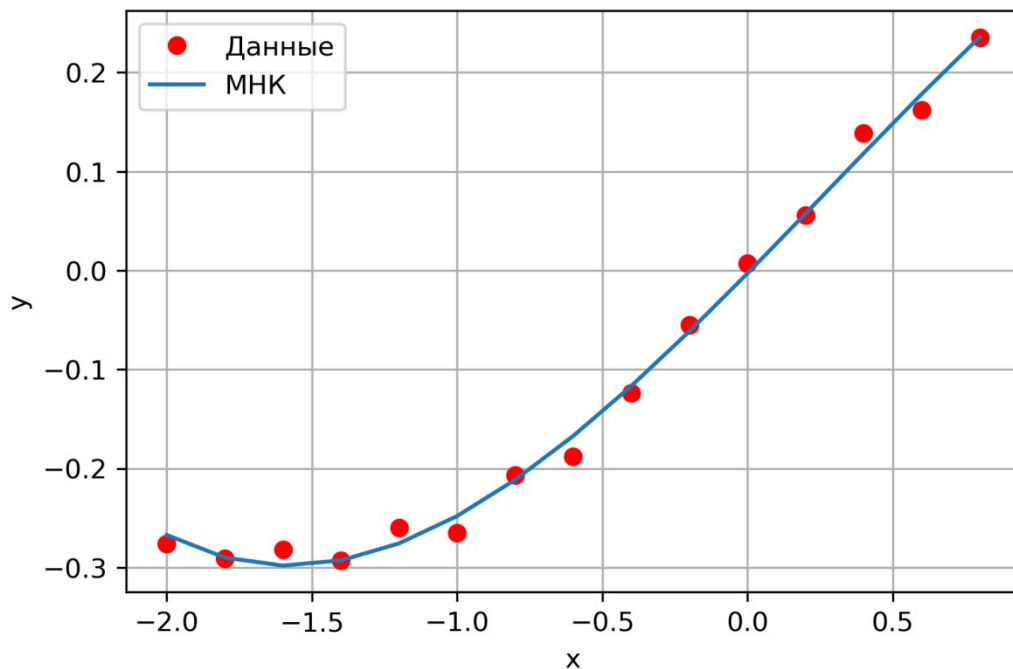
```

$c = [-1.6425 \quad 3.20522 \quad -3.97146 \quad 6.8980808]$
 МНК:
 $a = [-0.00300869 \quad 0.29851937 \quad 0.0236193 \quad -0.02979591]$
 невязка = 2.210×10^{-3}
 SCIPY:
 $a = [-0.0030087 \quad 0.29851937 \quad 0.02361931 \quad -0.0297959]$
 невязка = 2.210×10^{-3}

tab.dat – Блокнот

Файл Правка Формат

x	f
-2	-0.276
-1.8	-0.291
-1.6	-0.282
-1.4	-0.293
-1.2	-0.260
-1	-0.265
-0.8	-0.207
-0.6	-0.188
-0.4	-0.124
-0.2	-0.0553
0	0.007
0.2	0.0558
0.4	0.139
0.6	0.162
0.8	0.235



Как видно, приведенные в листинге данные могут быть сглажены зависимостью

$$P(x) = -0.003 + 0.2985x + 0.0236x^2 - 0.0298x^3$$

Решение в Excel (инструмент диаграммы «Добавить линию тренда...»):

