

Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут ім. Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки

## ЛАБОРАТОРНА РОБОТА № 6

з дисципліни «Методи планування експерименту»

на тему «Проведення трьохфакторного експерименту при використанні  
рівняння регресії з квадратичними членами»

ВИКОНАВ:  
студент II курсу ФІОТ  
групи ІВ-92  
Чередник Віталій Юрійович  
Номер у списку групи: 26

ПЕРЕВІРИВ:  
ас. Регіда П.Г.

## Хід роботи

**Мета:** Провести трьохфакторний експеримент і отримати адекватну модель – рівняння регресії, використовуючи ротатабельний композиційний план.

### Завдання на лабораторну роботу:

1. Ознайомитися з теоретичними відомостями.
2. Вибрати з таблиці варіантів і записати в протокол інтервали значень  $x_1$ ,  $x_2$ ,  $x_3$ . Обчислити і записати значення, відповідні кодованим значенням факторів +1; -1; + ; - ; 0 для  $x_1$ ,  $x_2$ ,  $x_3$ .
3. Значення функції відгуку знайти за допомогою підстановки в формулу:  
 $y_i = f(x_1, x_2, x_3) + \text{random}(10) - 5$ ,  
де  $f(x_1, x_2, x_3)$  вибирається по номеру в списку в журналі викладача.
4. Провести експерименти і аналізуючи значення статистичних перевірок, отримати адекватну модель рівняння регресії. При розрахунках використовувати натуральні значення факторів.
5. Зробити висновки по виконаній роботі.

### Варіант завдання

№ варіанту	$x_1$		$x_2$		$x_3$		$f(x_1, x_2, x_3)$
	min	max	min	max	min	max	
226	10	60	-25	10	10	15	$1,9+7,1*x_1+0,4*x_2+3,6*x_3+3,8*x_1*x_1+0,9*x_2*x_2+2,8*x_3*x_3+7,4*x_1*x_2+0,2*x_1*x_3+2,7*x_2*x_3+3,0*x_1*x_2*x_3$

### Код програми

```
from math import fabs
from random import randrange
import numpy as np
from numpy.linalg import solve
from scipy.stats import f, t

m = 3
n = 15

x1min = 10
x1max = 60
x2min = -25
x2max = 10
x3min = 10
x3max = 15

x01 = (x1max + x1min) / 2
x02 = (x2max + x2min) / 2
x03 = (x3max + x3min) / 2
deltax1 = x1max - x01
deltax2 = x2max - x02
deltax3 = x3max - x03

xn = [[-1, -1, -1, +1, +1, +1, -1, +1, +1, +1],
       [-1, -1, +1, +1, -1, -1, +1, +1, +1, +1],
       [-1, +1, -1, -1, +1, -1, +1, +1, +1, +1],
```

```

[-1, +1, +1, -1, -1, +1, -1, +1, +1, +1],
[+1, -1, -1, -1, -1, +1, +1, +1, +1, +1],
[+1, -1, +1, -1, +1, -1, -1, +1, +1, +1],
[+1, +1, -1, +1, -1, -1, -1, +1, +1, +1],
[+1, +1, +1, +1, +1, +1, +1, +1, +1, +1],
[-1.73, 0, 0, 0, 0, 0, 0, 0, 2.9929, 0, 0],
[+1.73, 0, 0, 0, 0, 0, 0, 0, 2.9929, 0, 0],
[0, -1.73, 0, 0, 0, 0, 0, 0, 2.9929, 0, 0],
[0, +1.73, 0, 0, 0, 0, 0, 0, 2.9929, 0, 0],
[0, 0, -1.73, 0, 0, 0, 0, 0, 0, 2.9929, 0],
[0, 0, +1.73, 0, 0, 0, 0, 0, 0, 2.9929, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]

x1 = [x1min, x1min, x1min, x1min, x1max, x1max, x1max, x1max, -1.73 * deltax1 + x01,
1.73 * deltax1 + x01, x01, x01,
      x01, x01, x01]
x2 = [x2min, x2min, x2max, x2max, x2min, x2min, x2max, x2max, x02, x02, -1.73 *
deltax2 + x02, 1.73 * deltax2 + x02,
      x02, x02, x02]
x3 = [x3min, x3max, x3min, x3max, x3min, x3max, x3min, x3max, x03, x03, x03, x03, -
1.73 * deltax3 + x03,
      1.73 * deltax3 + x03, x03]
x1x2 = [0] * 15
x1x3 = [0] * 15
x2x3 = [0] * 15
x1x2x3 = [0] * 15
x1kv = [0] * 15
x2kv = [0] * 15
x3kv = [0] * 15
for i in range(15):
    x1x2[i] = x1[i] * x2[i]
    x1x3[i] = x1[i] * x3[i]
    x2x3[i] = x2[i] * x3[i]
    x1x2x3[i] = x1[i] * x2[i] * x3[i]
    x1kv[i] = x1[i] ** 2
    x2kv[i] = x2[i] ** 2
    x3kv[i] = x3[i] ** 2

list_for_a = list(zip(x1, x2, x3, x1x2, x1x3, x2x3, x1x2x3, x1kv, x2kv, x3kv))

print("Матриця планування з натуралізованими коефіцієнтами X:")
print("      X1      X2      X3      X1X2      X1X3      X2X3
X1X2X3      X1X1"
      "      X2X2      X3X3")
for i in range(15):
    print(end=' ')
    for j in range(len(list_for_a[0])):
        print("{:^12.3f}".format(list_for_a[i][j]), end=' ')
    print("")

def function(X1, X2, X3):
    y = 1.9 + 7.1 * X1 + 0.4 * X2 + 3.6 * X3 + 3.8 * X1 * X1 + 0.9 * X2 * X2 + 2.8 *
X3 * X3 + 7.4 * X1 * X2 + \
        0.2 * X1 * X3 + 2.7 * X2 * X3 + 3 * X1 * X2 * X3 + randrange(0, 10) - 5
    return y

Y = [[function(list_for_a[j][0], list_for_a[j][1], list_for_a[j][2]) for i in
range(m)] for j in range(15)]
print("Матриця планування Y:")
print("      Y1      Y2      Y3")

```

```

for i in range(15):
    print(end=' ')
    for j in range(len(Y[0])):
        print("{:^12.3f}".format(Y[i][j]), end=' ')
    print("")

Y_average = []
for i in range(len(Y)):
    Y_average.append(np.mean(Y[i], axis=0))
print("Середні значення відгуку за рядками:")
for i in range(15):
    print("{:.3f}".format(Y_average[i]), end=" ")

dispersions = []
for i in range(len(Y)):
    a = 0
    for k in Y[i]:
        a += (k - np.mean(Y[i], axis=0)) ** 2
    dispersions.append(a / len(Y[i]))

def find_known(num):
    a = 0
    for j in range(15):
        a += Y_average[j] * list_for_a[j][num - 1] / 15
    return a

def a(first, second):
    a = 0
    for j in range(15):
        a += list_for_a[j][first - 1] * list_for_a[j][second - 1] / 15
    return a

my = sum(Y_average) / 15
mx = []
for i in range(10):
    number_lst = []
    for j in range(15):
        number_lst.append(list_for_a[j][i])
    mx.append(sum(number_lst) / len(number_lst))

det1 = [
    [1, mx[0], mx[1], mx[2], mx[3], mx[4], mx[5], mx[6], mx[7], mx[8], mx[9]],
    [mx[0], a(1, 1), a(1, 2), a(1, 3), a(1, 4), a(1, 5), a(1, 6), a(1, 7), a(1, 8),
a(1, 9), a(1, 10)],
    [mx[1], a(2, 1), a(2, 2), a(2, 3), a(2, 4), a(2, 5), a(2, 6), a(2, 7), a(2, 8),
a(2, 9), a(2, 10)],
    [mx[2], a(3, 1), a(3, 2), a(3, 3), a(3, 4), a(3, 5), a(3, 6), a(3, 7), a(3, 8),
a(3, 9), a(3, 10)],
    [mx[3], a(4, 1), a(4, 2), a(4, 3), a(4, 4), a(4, 5), a(4, 6), a(4, 7), a(4, 8),
a(4, 9), a(4, 10)],
    [mx[4], a(5, 1), a(5, 2), a(5, 3), a(5, 4), a(5, 5), a(5, 6), a(5, 7), a(5, 8),
a(5, 9), a(5, 10)],
    [mx[5], a(6, 1), a(6, 2), a(6, 3), a(6, 4), a(6, 5), a(6, 6), a(6, 7), a(6, 8),
a(6, 9), a(6, 10)],
    [mx[6], a(7, 1), a(7, 2), a(7, 3), a(7, 4), a(7, 5), a(7, 6), a(7, 7), a(7, 8),
a(7, 9), a(7, 10)],
    [mx[7], a(8, 1), a(8, 2), a(8, 3), a(8, 4), a(8, 5), a(8, 6), a(8, 7), a(8, 8),
a(8, 9), a(8, 10)],
    [mx[8], a(9, 1), a(9, 2), a(9, 3), a(9, 4), a(9, 5), a(9, 6), a(9, 7), a(9, 8),

```

```

a(9, 9), a(9, 10)],
    [mx[9], a(10, 1), a(10, 2), a(10, 3), a(10, 4), a(10, 5), a(10, 6), a(10, 7),
a(10, 8), a(10, 9), a(10, 10)]]

det2 = [my, find_known(1), find_known(2), find_known(3), find_known(4),
find_known(5), find_known(6), find_known(7),
    find_known(8), find_known(9), find_known(10)]

beta = solve(det1, det2)
print("\nОтримане рівняння регресії:")
print("{:.3f} + {:.3f} * X1 + {:.3f} * X2 + {:.3f} * X3 + {:.3f} * X1X2 + {:.3f} *
X1X3 + {:.3f} * X2X3"
    "+ {:.3f} * X1X2X3 + {:.3f} * X11^2 + {:.3f} * X22^2 + {:.3f} * X33^2 = ŷ"
    .format(beta[0], beta[1], beta[2], beta[3], beta[4], beta[5], beta[6],
beta[7], beta[8], beta[9], beta[10]))
y_i = [0] * 15
print("Експериментальні значення:")
for k in range(15):
    y_i[k] = beta[0] + beta[1] * list_for_a[k][0] + beta[2] * list_for_a[k][1] +
beta[3] * list_for_a[k][2] + \
        beta[4] * list_for_a[k][3] + beta[5] * list_for_a[k][4] + beta[6] *
list_for_a[k][5] + beta[7] * \
        list_for_a[k][6] + beta[8] * list_for_a[k][7] + beta[9] *
list_for_a[k][8] + beta[10] * list_for_a[k][9]
for i in range(15):
    print("{:.3f}".format(y_i[i]), end=" ")
print("\n----- Перевірка за критерієм Кохрена -----
-----")
Gp = max(dispersions) / sum(dispersions)
Gt = 0.3346
print("Gp =", Gp)
if Gp < Gt:
    print("Дисперсія однорідна")
else:
    print("Дисперсія неоднорідна")

print("----- Перевірка значущості коефіцієнтів за критерієм Стюдента --
-----")
sb = sum(dispersions) / len(dispersions)
sbs = (sb / (15 * m)) ** 0.5

F3 = (m - 1) * n
coefs1 = []
coefs2 = []
d = 11
res = [0] * 11
for j in range(11):
    t_pract = 0
    for i in range(15):
        if j == 0:
            t_pract += Y_average[i] / 15
        else:
            t_pract += Y_average[i] * xn[i][j - 1]
    res[j] = beta[j]
    if fabs(t_pract / sbs) < t.ppf(q=0.975, df=F3):
        coefs2.append(beta[j])
        res[j] = 0
        d-=1
    else:
        coefs1.append(beta[j])
print("Значущі коефіцієнти регресії:", [round(i, 3) for i in coefs1])
print("Незначущі коефіцієнти регресії:", [round(i, 3) for i in coefs2])

```

```

y_st = []
for i in range(15):
    y_st.append(res[0] + res[1] * x1[i] + res[2] * x2[i] + res[3] * x3[i] + res[4] *
x1x2[i] + res[5] *
                x1x3[i] + res[6] * x2x3[i] + res[7] * x1x2x3[i] + res[8] * x1kv[i] +
res[9] *
                x2kv[i] + res[10] * x3kv[i])
print("Значення з отриманими коефіцієнтами:")
for i in range(15):
    print("{:.3f}".format(y_st[i]), end=" ")

print("\n----- Перевірка адекватності за критерієм Фішера -----
-----")
Sad = m * sum([(y_st[i] - Y_average[i]) ** 2 for i in range(15)]) / (n - d)
Fp = Sad / sb
F4 = n - d
print("Fp =", Fp)
if Fp < f.ppf(q=0.95, dfn=F4, dfd=F3):
    print("Рівняння регресії адекватне при рівні значимості 0.05")
else:
    print("Рівняння регресії неадекватне при рівні значимості 0.05")

```

## Результати роботи програми

Матриця планування з натуралізованими коефіцієнтами X:

X1	X2	X3	X1X2	X1X3	X2X3	X1X2X3	X1X1	X2X2	X3X3
10.000	-25.000	10.000	-250.000	100.000	-250.000	-2500.000	100.000	625.000	100.000
10.000	-25.000	15.000	-250.000	150.000	-375.000	-3750.000	100.000	625.000	225.000
10.000	10.000	10.000	100.000	100.000	100.000	1000.000	100.000	100.000	100.000
10.000	10.000	15.000	100.000	150.000	150.000	1500.000	100.000	100.000	225.000
60.000	-25.000	10.000	-1500.000	600.000	-250.000	-15000.000	3600.000	625.000	100.000
60.000	-25.000	15.000	-1500.000	900.000	-375.000	-22500.000	3600.000	625.000	225.000
60.000	10.000	10.000	600.000	600.000	100.000	6000.000	3600.000	100.000	100.000
60.000	10.000	15.000	600.000	900.000	150.000	9000.000	3600.000	100.000	225.000
-8.250	-7.500	12.500	61.875	-103.125	-93.750	773.438	68.062	56.250	156.250
78.250	-7.500	12.500	-586.875	978.125	-93.750	-7335.938	6123.062	56.250	156.250
35.000	-37.775	12.500	-1322.125	437.500	-472.188	-16526.562	1225.000	1426.951	156.250
35.000	22.775	12.500	797.125	437.500	284.688	9964.062	1225.000	518.701	156.250
35.000	-7.500	8.175	-262.500	286.125	-61.313	-2145.938	1225.000	56.250	66.831
35.000	-7.500	16.825	-262.500	588.875	-126.188	-4416.562	1225.000	56.250	283.081
35.000	-7.500	12.500	-262.500	437.500	-93.750	-3281.250	1225.000	56.250	156.250

Матриця планування Y:

Y1	Y2	Y3
-8687.600	-8688.600	-8688.600
-12389.100	-12390.100	-12390.100
4890.900	4894.900	4895.900
6904.900	6906.900	6903.900
-41678.600	-41678.600	-41678.600
-64091.100	-64093.100	-64084.100
37349.900	37347.900	37342.900
46911.900	46908.900	46910.900
3240.525	3237.525	3233.525
-2052.950	-2052.950	-2054.950
-53892.773	-53889.773	-53890.773
42512.909	42514.909	42510.909
-3317.051	-3321.051	-3317.051
-9612.898	-9612.898	-9607.898
-6519.350	-6519.350	-6517.350

```

Середні значення відгуку за рядками:
-8688.267 -12389.767 4893.900 6905.233 -41678.600 -64089.433 37346.900 46910.567 3237.192 -2053.617 -53891.107 42512.909 -3318.384 -9611.231 -6518.683
Отримане рівняння регресії:

$$9.520 + 7.069 * X_1 + 1.343 * X_2 + 2.332 * X_3 + 7.383 * X_1X_2 + 0.196 * X_1X_3 + 2.631 * X_2X_3 + 3.001 * X_1X_2X_3 + 3.801 * X_1^2 + 0.904 * X_2^2 + 2.850 * X_3^2 = \hat{y}$$

Експериментальні значення:
-8686.933 -12389.781 4894.891 6904.876 -41677.417 -64089.599 37347.739 46910.057 3236.465 -2053.993 -53892.055 42512.755 -3320.494 -9610.224 -6518.676
----- Перевірка за критерієм Кохрена -----
Gr = 0.2701612903225807
Дисперсія однорідна
----- Перевірка значущості коефіцієнтів за критерієм Стюдента -----
Значущі коефіцієнти регресії: [9.52, 7.069, 1.343, 2.332, 7.383, 0.196, 2.631, 3.001, 3.801, 0.904, 2.85]
Незначущі коефіцієнти регресії: []
Значення з отриманими коефіцієнтами:
-8686.933 -12389.781 4894.891 6904.876 -41677.417 -64089.599 37347.739 46910.057 3236.465 -2053.993 -53892.055 42512.755 -3320.494 -9610.224 -6518.676
----- Перевірка адекватності за критерієм Фішера -----
Fr = 2.5189961538766346
Рівняння регресії адекватне при рівні значимості 0.05

Process finished with exit code 0

```

**Висновок:** у ході виконання лабораторної роботи я провів трьохфакторний експеримент і отримав адекватну модель — рівняння регресії, використовуючи рототабельний композиційний план. Кінцева мета роботи досягнута.