

**Міністерство освіти і науки України  
Національний технічний університет України «КПІ» імені Ігоря  
Сікорського  
Факультет обчислювальної техніки  
Кафедра обчислювальної техніки**

**ЗВІТ**  
з лабораторної роботи 2  
з навчальної дисципліни «Основи комп'ютерного моделювання»

Тема: Ідентифікація об'єкта за даними спостережень

Виконав  
Студент 4 курсу ІП-94  
Рекечинський Дмитро

Перевірів  
Іваніщев Б. В.

Київ 2022

**Завдання до роботи:** Скласти програму ідентифікації об'єкта за даними спостережень функціями, які дані у варіанті завдання. Перевірити програму на даних, які точно або приблизно відповідають наперед відомій моделі. Ідентифікувати об'єкт за даними спостережень, які наведені у варіанті завдання. Проаналізувати результати. Оформити звіт та здати лабораторну роботу.

**Варіант завдання:**  $9423 \bmod 4 + 1 = 4$

4) Використати функції виду  $y = b_0 \cdot x^{b_1} \cdot z^{b_2}$  для ідентифікації об'єкта за такими даними спостережень:

y	6.0	8.5	11	14.5	18.5	23	28.5	35
x	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8
z	5.5	6	6.5	7	7.5	8	8.5	9
y	42	50.5	60	70.5	82.5	96	110.5	127
x	1.9	2.0	2.1	2.2	2.3	2.4	2.5	2.6
z	9.5	10	10.5	11	11.5	12	12.5	13

## Лістинг програми

Примітка: Програма написана за допомогою компільованої мови програмування Nim.

Команда для компіляції та запуску: `nim c -r solution.nim`

Вміст програми у файлі `solution.nim`:

```
#####  
# Import statements  
#####  
  
# ln()  
import std/math  
  
# sequence.mapIt(), newSeqWith()  
import sequtils  
  
# float.formatFloat()  
import strutils  
  
#####  
# Types  
#####  
  
# Matrix, which contains elements with generic type  
type matrix[T] = seq[seq[T]]  
  
#####  
# Global constants  
#####  
  
const xArgs = [
```

```

    1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8,
    1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6
]

const zArgs = [
    5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 8.5, 9.0,
    9.5, 10.0, 10.5, 11.0, 11.5, 12.0, 12.5, 13.0
]

const yValues = [
    6.0, 8.5, 11.0, 14.5, 18.5, 23.0, 28.5, 35.0,
    42.0, 50.5, 60.0, 70.5, 82.5, 96.0, 110.5, 127.0
]

# Nim supports scientific notation
const EPSILON = 1e-10

#####
# Functions
#####

# func is a procedure (function) with no side-effects

func initializeMatrix[T] (rowAmount, columnAmount: int): matrix[T] =
    newSeqWith(rowAmount, newSeqWith(columnAmount, T(0)))

func `$`[T] (self: matrix[T]): string =
    for row in self:
        result &= row.mapIt(it.formatFloat(ffDecimal, 4)).join("\t") & "\n"

func transpose[T] (matrix: matrix[T]): matrix[T] =
    # Dimentions of new matrix are inverted

```

```

let rowAmount = matrix[0].len
let columnAmount = matrix.len

# Create a matrix, fill with zeros
# T(0) means zero depending on generic type
result = initializeMatrix[T](rowAmount, columnAmount)

for i in 0..<rowAmount:
    for j in 0..<columnAmount:
        result[i][j] = matrix[j][i]

func multiply[T] (a, b: matrix[T]): matrix[T] =
    let intermediateLength = a[0].len

    if intermediateLength != b.len:
        raise Exception.newException(
            "Column amount of matrix A should be equal to row amount of
matrix B"
        )

    let rowAmount = a.len
    let columnAmount = b[0].len

    result = initializeMatrix[T](rowAmount, columnAmount)

    for i in 0..<rowAmount:
        for j in 0..<columnAmount:
            for k in 0..<intermediateLength:
                result[i][j] += a[i][k] * b[k][j]

# Transform a matrix to reduced row echelon form.

```

```

func transformToRref[T] (matrix: var matrix[T]) =
  var lead = 0

  for rowNumber in 0..

```

```

func inverse[T] (matrix: matrix[T]): matrix[T] =
    let matrixSize = matrix.len

    if matrixSize != matrix[0].len:
        raise Exception.newException(
            "Column amount of square matrix should be equal to row amount"
        )

    result = initializeMatrix[T](matrixSize, matrixSize)

    # Build augmented matrix.
    var augmat: matrix[T] = initializeMatrix[T](matrixSize, matrixSize *

2)

    for i in 0..

```

```
#####
```

```
let lnXArgs = xArgs.mapIt(ln(it))
```

```
let lnZArgs = zArgs.mapIt(ln(it))
```

```
let lnYValues = yValues.mapIt(ln(it))
```

```
var argMatrix: matrix[float64]
```

```
for i in 0..
```

```
    argMatrix.add @[1.0, lnXArgs[i], lnZArgs[i]]
```

```
let transposedArgMatrix = transpose(argMatrix)
```

```
let argProduct = multiply(transposedArgMatrix, argMatrix)
```

```
let invertedArgProduct = inverse(argProduct)
```

```
let yMatrix = lnYValues.mapIt(@[it])
```

```
let partialResult = multiply(invertedArgProduct, transposedArgMatrix)
```

```
# Get array of bN values
```

```
var bValues = multiply(partialResult, yMatrix).mapIt(it[0])
```

```
# First value in bValues is ln-ed, thus, we need to
```

```
# return back b0 initial value by performing exponential
```

```
# operation
```

```
bValues[0] = exp(bValues[0])
```

```
let (bZero, bOne, bTwo) = (bValues[0], bValues[1], bValues[2])
```

```
echo "Found function: ",
```

```
    bZero.formatFloat(ffDecimal, 4), " * x^(",
```



```
bOne.formatFloat(ffDecimal, 4), ") * z^(",  
bTwo.formatFloat(ffDecimal, 4), ")"
```

```
echo "\n"
```

```
proc foundFunction (x, z: float64): float64 =  
  bZero * pow(x, bOne) * pow(z, bTwo)
```

```
var smallestSquareCriteria = 0.0
```

```
echo "x\tz\ty\tf(x, z)"
```

```
for i in 0..  echo xArgs[i].formatFloat(ffDecimal, 2), "\t",  
    zArgs[i].formatFloat(ffDecimal, 2), "\t",  
    yValues[i].formatFloat(ffDecimal, 2), "\t",  
    foundFunction(xArgs[i], zArgs[i]).formatFloat(ffDecimal, 2)
```

```
echo "\n"
```

```
for i in 0..  let value = foundFunction(xArgs[i], zArgs[i])  
  let expected = yValues[i]  
  smallestSquareCriteria += (value - expected)^2
```

```
echo "Smallest square criteria value: ", smallestSquareCriteria
```

## Перевірка роботи

Перш за все, необхідно привести дану формулу у вигляд суми:

$$y = b_0 \cdot x^{b_1} \cdot z^{b_2}$$

Якщо логарифмувати обидві частини виразу:

$$\ln(y) = \ln(b_0 \cdot x^{b_1} \cdot z^{b_2})$$

То можна розкласти добуток аргументів логарифмічної функції на суму логарифмів:

$$\ln(y) = \ln(b_0) + \ln(x^{b_1}) + \ln(z^{b_2})$$

А степені аргументів логарифмічної функції винести як коефіцієнти:

$$\ln(y) = \ln(b_0) + b_1 \cdot \ln(x) + b_2 \cdot \ln(z)$$

Таким чином, ми отримали форму, з якою більш зручно працювати.

При отриманні шуканих коефіцієнтів  $b_i$  лише  $b_0$  доведеться приводити до нормальної форми (так як результатом буде не  $b_0$ , а  $\ln(b_0)$  ).

В кодї програми це передбачено:

```
bValues[0] = exp(bValues[0])
```

Тобто, взяти експоненту від  $\ln(b_0)$ , і вийде  $b_0$ .

### 1. Функція з визначеними коефіцієнтами

Програма виводить на екран знайдену функцію з визначеними коефіцієнтами:

Found function:  $4.3921 * x^{(3.5231)} * z^{(0.0000)}$

Або, в простішому вигляді (враховуючи те, що число в нульовому степені дорівнює одиниці за умови, що число не нуль):

$$4.3921 * x^{3.5231}$$

Тобто, за даними розрахунку, вплив  $z$  на результат вкрай мізерний.

2. Таблиця заданих значень відгуку моделі та значень відгуку моделі, розрахованих за функцією

Таблиця була побудована за допомогою Excel:

x	y	f(x, z)		z
1.1	6	6.14		5.5
1.2	8.5	8.35		6
1.3	11	11.07		6.5
1.4	14.5	14.37		7
1.5	18.5	18.33		7.5
1.6	23	23		8
1.7	28.5	28.48		8.5
1.8	35	34.84		9
1.9	42	42.15		9.5
2	50.5	50.49		10
2.1	60	59.97		10.5
2.2	70.5	70.64		11
2.3	82.5	82.62		11.5
2.4	96	95.99		12
2.5	110.5	110.83		12.5
2.6	127	127.26		13

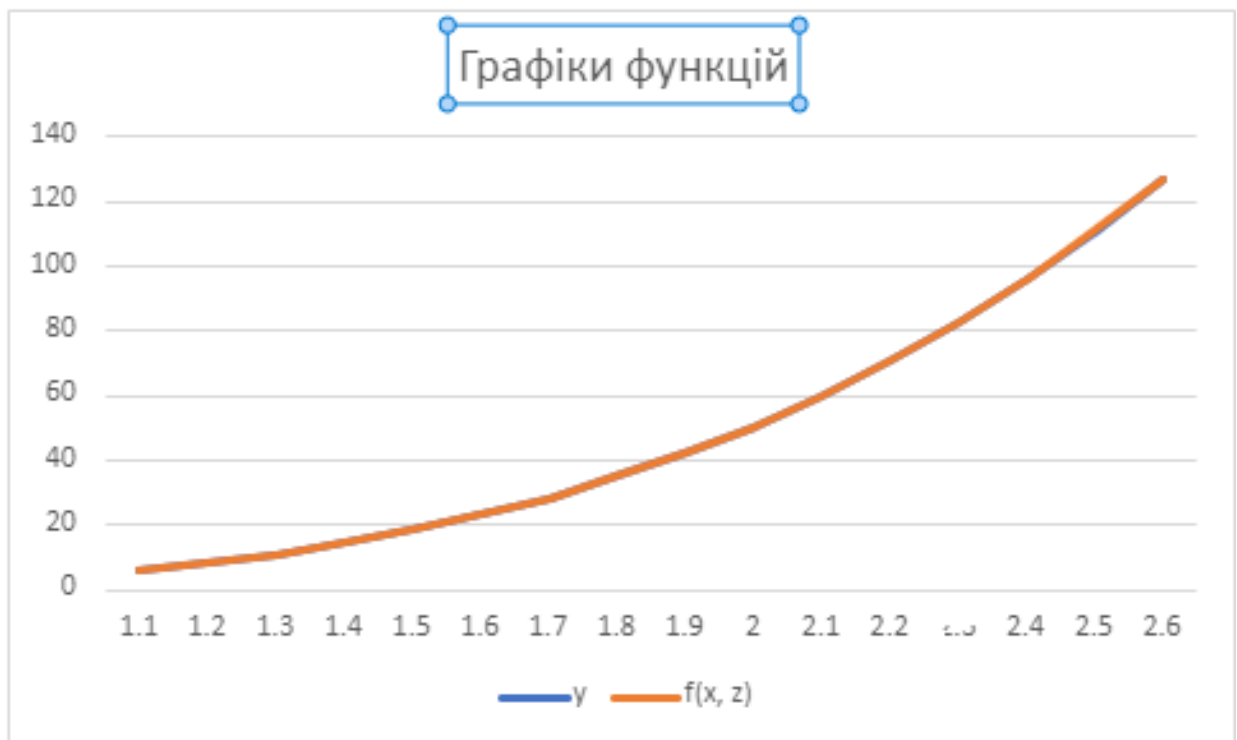
3. Значення критерію найменших квадратів

Результат програми:

Smallest square criteria value: 0.3585521738982345

Тобто, критерій найменших квадратів показав невелику різницю між результатами знайденої моделі та виміряними значеннями  $y$ .

#### 4. Графік функції



За графіками функцій ми можемо побачити, що в деяких точках ледве помітна різниця між вимірним значенням  $y$  та знайденою моделлю  $f(x, z)$ .

Що свідчить про достатню точність моделі.

#### Висновки про результати ідентифікації

Ідентифікована модель показує дуже слабку залежність від змінної  $z$ .

Із таблиці даних та графіків функцій ми можемо помітити, що різниця між вимірними значеннями та значеннями моделі невелика.

Критерій найменших квадратів менший, ніж 1, що показує достатню точність.

Вищенаведені фактори свідчать про те, що знайдена модель відповідає критеріям.

## **Висновки**

Під час виконання даної роботи було проведено ідентифіковано модель об'єкта. Усі критерії підтверджують можливість створення моделі на основі заданої формули, в тому числі критерій найменших квадратів. Для роботи було використано компільовану мову програмування Nim та програму для роботи із табличними даними Excel.