**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

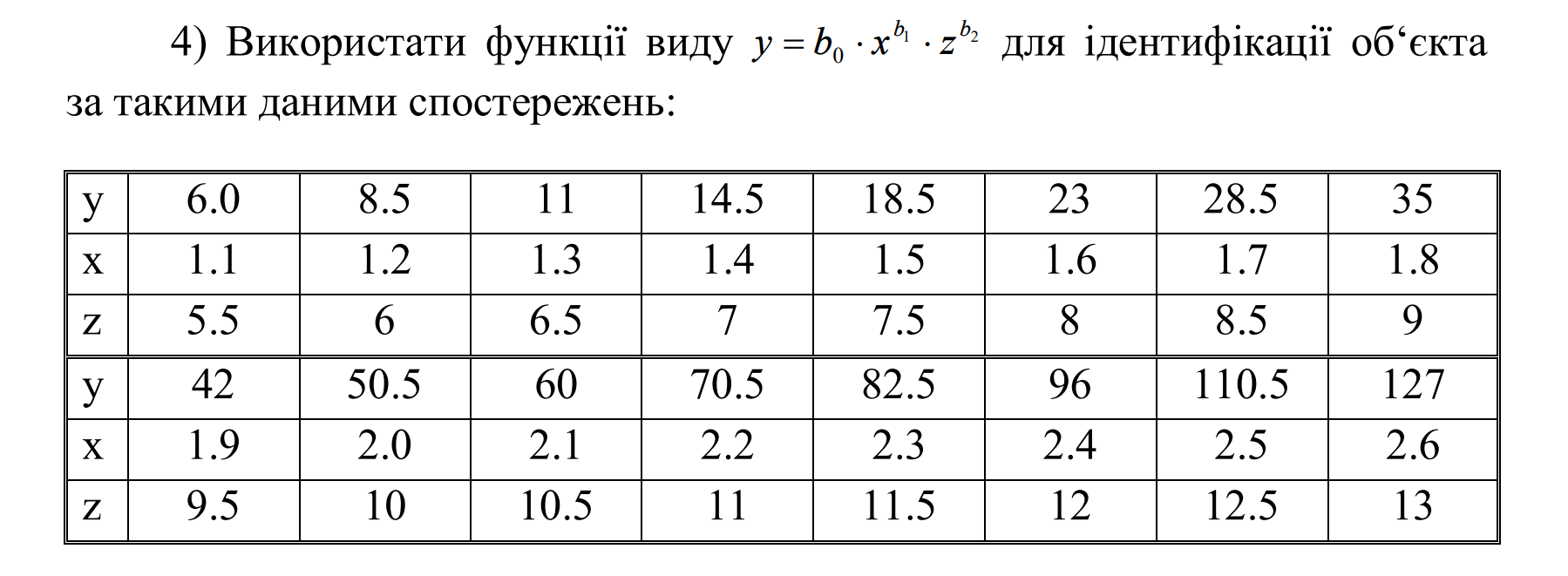
Перевірив

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 mod 4 + 1 = 4



**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 scentific 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..<matrix.len:

if lead >= matrix[0].len: return

var i = rowNumber

while matrix[i][lead] == 0:

inc i

if i == matrix.len:

i = rowNumber

inc lead

if lead == matrix[0].len: return

swap matrix[i], matrix[rowNumber]

let delimeter = matrix[rowNumber][lead]

# Checking "delimeter != 0" will give wrong results in some cases.

if abs(delimeter) > EPSILON:

for item in matrix[rowNumber].mitems:

item /= delimeter

for i in 0..<matrix.len:

if i != rowNumber:

let multiplier = matrix[i][lead]

for columnNumber in 0..<matrix[0].len:

matrix[i][columnNumber] -=

matrix[rowNumber][columnNumber] \* multiplier

inc lead

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..<matrixSize:

augmat[i][0..<matrixSize] = matrix[i]

augmat[i][matrixSize + i] = 1

# Transform it to reduced row echelon form.

augmat.transformToRref()

# Extract second half.

for i in 0..<matrixSize:

for j in 0..<matrixSize:

result[i][j] = augmat[i][matrixSize + j]

func calculateError[T](expected, actual: T): T =

abs((expected - actual) / expected)

#############################

# Implementation

#############################

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..<lnXArgs.len:

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..<xArgs.len:

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..<xArgs.len:

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:



Якщо логарифмувати обидві частини виразу:



То можна розкласти добуток аргументів логарифмічної функції на суму логарифмів:



А степені аргументів логарифмічної функції винести як коефіцієнти:



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

При отриманні шуканих коефіцієнтів bi лише b0 доведеться приводити до нормальної форми (так як результатом буде не b0, а ln(b0) ).

В коді програми це передбачено:

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

Тобто, взяти експоненту від ln(b0), і вийде b0.

1. Функція з визначеними коефіцієнтами

Програма виводить на екран знайдену функцію з визначеними коефіцієнтами:

Found function: 4.3921 \* x^(3.5231) \* z^(0.0000)

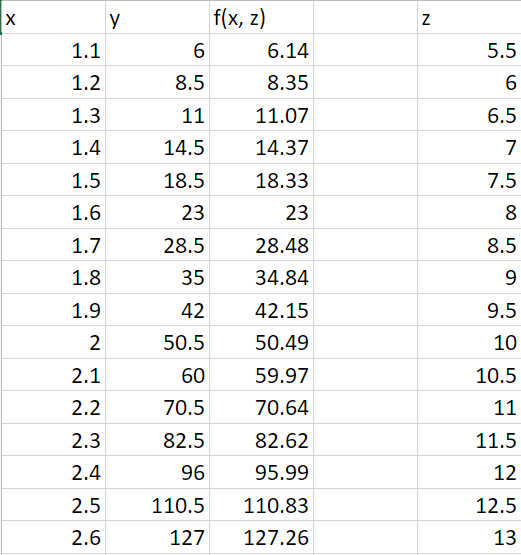
Або, в простішому вигляді (враховуючи те, що число в нульовому степені дорівнює одиниці за умови, що число не нуль):

4.3921 \* x3.5231

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:



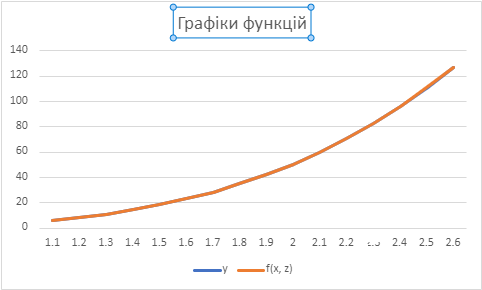
3. Значення критерію найменших квадратів

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 виміряним 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 виміряними 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.