Національний технічний університет України «КПІ ім. Ігоря Сікорського»

Факультет Інформатики та Обчислювальної Техніки

Кафедра Автоматизованих Систем Обробки Інформації та Управління

Лабораторна робота № 4

з дисципліни «Чисельні методи»

на тему

«**Обчислення власних значень та власних векторів матриць**»

Виконала:

студентка гр. ІП-93

Узунлу Гамзенур

Викладач:

доц. Рибачук Л.В.

Київ – 2021

### Зміст

[Зміст 2](file:///C:\Users\vania\Downloads\Telegram%20Desktop\Zvit_example.doc#_Toc190845267)

[1 Постановка задачі 3](file:///C:\Users\vania\Downloads\Telegram%20Desktop\Zvit_example.doc#_Toc190845268)

[2 Розв’язок 4](file:///C:\Users\vania\Downloads\Telegram%20Desktop\Zvit_example.doc#_Toc190845269)

[3 Розв’язок у Mathcad 5](file:///C:\Users\vania\Downloads\Telegram%20Desktop\Zvit_example.doc#_Toc190845270)

[4 Лістинг програми 6](file:///C:\Users\vania\Downloads\Telegram%20Desktop\Zvit_example.doc#_Toc190845271)

### 1 Постановка задачі:

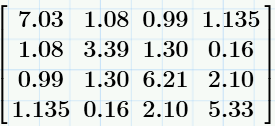
Створити програму, для приведення матриці А до нормальної форми Фробеніюса.

Отримане характеристичне рівняння розв’язати довільним способом у Mathcad і отримати всі власні числа λі, і = 1,…,m з точністю 5 знаків після коми.

Знайти по одному власному вектору для кожного власного числа.

Перевірити точність знайдених результатів, підставляючи у рівняння (1) знайдені власні числа та власні вектори.

Знайти власні числа матриці А виключно за допомогою Mathcad і порівняти з отриманими раніше результатами.



### 2 Розв’язок

Результати виконання програми:

М^-1:

1.0000000 0.0000000 0.0000000 0.0000000

0.0000000 1.0000000 0.0000000 0.0000000

1.1350000 0.1600000 2.1000000 5.3300000

0.0000000 0.0000000 0.0000000 1.0000000

P:

6.4949286 1.0045714 0.4714286 -1.3777143

0.3773810 3.2909524 0.6190476 -3.1395238

2.4627749 3.4031410 12.1741190 -30.7553295

0.0000000 0.0000000 1.0000000 0.0000000

М:

1.0000000 0.0000000 0.0000000 0.0000000

0.0000000 1.0000000 0.0000000 0.0000000

-0.5404762 -0.0761905 0.4761905 -2.5380952

0.0000000 0.0000000 0.0000000 1.0000000

М^-1:

1.0000000 0.0000000 0.0000000 0.0000000

2.4627749 3.4031410 12.1741190 -30.7553295

0.0000000 0.0000000 1.0000000 0.0000000

0.0000000 0.0000000 0.0000000 1.0000000

P:

5.7679433 0.2951895 -3.1222434 7.7009356

7.3845516 16.1920567 -76.4024518 109.4957537

0.0000000 1.0000000 0.0000000 0.0000000

0.0000000 0.0000000 1.0000000 0.0000000

М:

1.0000000 0.0000000 0.0000000 0.0000000

-0.7236770 0.2938462 -3.5773185 9.0373364

0.0000000 0.0000000 1.0000000 0.0000000

0.0000000 0.0000000 0.0000000 1.0000000

М^-1:

7.3845516 16.1920567 -76.4024518 109.4957537

0.0000000 1.0000000 0.0000000 0.0000000

0.0000000 0.0000000 1.0000000 0.0000000

0.0000000 0.0000000 0.0000000 1.0000000

P:

21.9600000 -167.6174750 527.1243980 -574.6973443

1.0000000 0.0000000 0.0000000 0.0000000

0.0000000 1.0000000 0.0000000 0.0000000

0.0000000 0.0000000 1.0000000 0.0000000

М:

0.1354178 -2.1926933 10.3462547 -14.8276780

0.0000000 1.0000000 0.0000000 0.0000000

0.0000000 0.0000000 1.0000000 0.0000000

0.0000000 0.0000000 0.0000000 1.0000000

S:

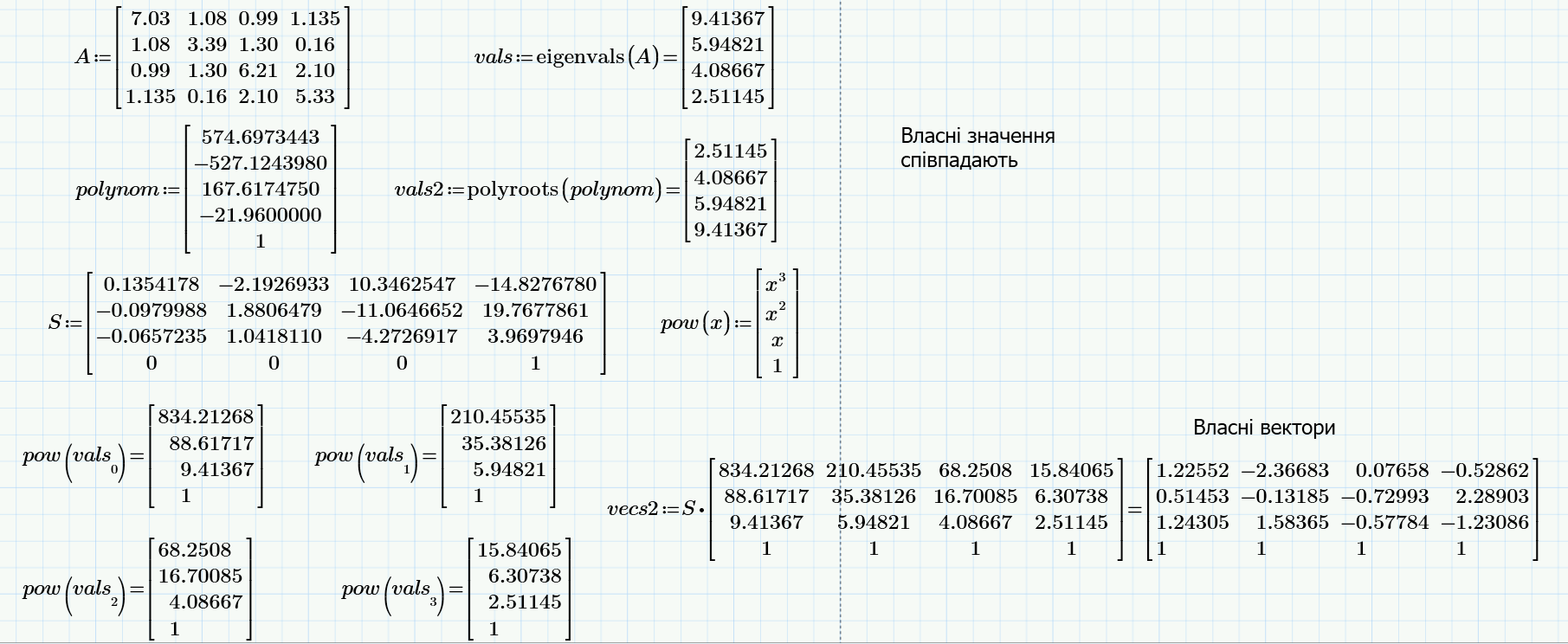
0.1354178 -2.1926933 10.3462547 -14.8276780

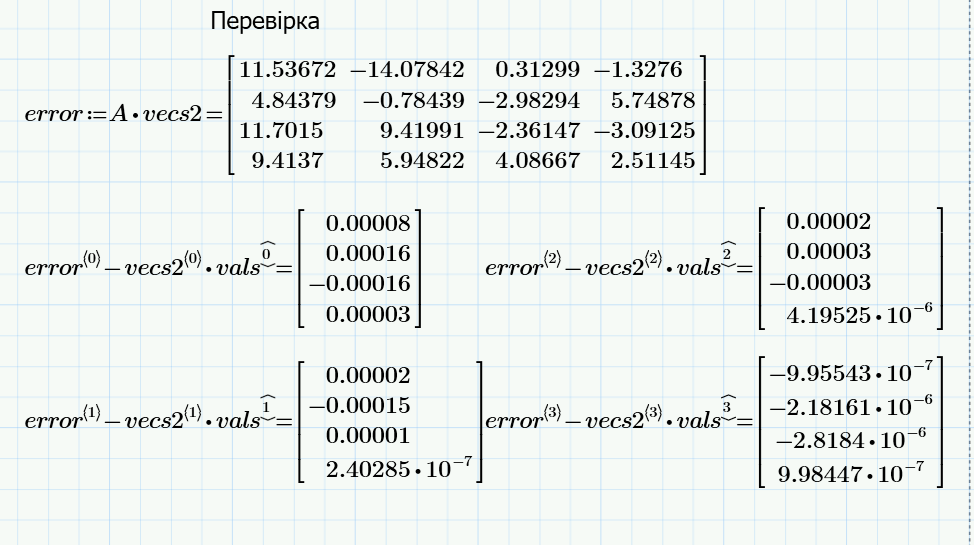
-0.0979988 1.8806479 -11.0646652 19.7677861

-0.0657235 1.0418110 -4.2726917 3.9697946

0.0000000 0.0000000 0.0000000 1.0000000

### 3 Розв’язок у MathCad

****

****

### 4 Лістинг програми

let A = [  
 [7.03, 1.08, 0.99, 1.135],  
 [1.08, 3.39, 1.30, 0.16],  
 [0.99, 1.30, 6.21, 2.10],  
 [1.135, 0.16, 2.10, 5.33]  
]  
let S = [  
 [1, 0, 0, 0],  
 [0, 1, 0, 0],  
 [0, 0, 1, 0],  
 [0, 0, 0, 1]  
]  
let M1 = [  
 [1, 0, 0, 0],  
 [0, 1, 0, 0],  
 [0, 0, 1, 0],  
 [0, 0, 0, 1]  
]  
let M2 = [  
 [1, 0, 0, 0],  
 [0, 1, 0, 0],  
 [0, 0, 1, 0],  
 [0, 0, 0, 1]  
]  
  
for (let i = 1; i < A.length; i++) {  
 const ind = A.length - i  
 const row = A[ind].slice()  
 M1[ind - 1] = row.slice()  
  
 for (let j = 0; j < A.length; j++)  
 row[j] = -row[j]  
 row[ind - 1] = 1  
 for (let j = 0; j < A.length; j++) {  
 row[j] = row[j] / A[ind][ind - 1]  
 }  
 M2[A.length - i - 1] = row.slice()  
  
 S = multiplyMatrix(S, M2)  
 A = multiplyMatrix(M1, A)  
 A = multiplyMatrix(A, M2)  
  
 console.log('\nМ^-1:')  
 for (let j = 0; j < A.length; j++) {  
 console.log(M1[j].map(num => num.toFixed(7)).join(" "))  
 }  
 console.log('\nP:')  
 for (let j = 0; j < A.length; j++) {  
 console.log(A[j].map(num => num.toFixed(7)).join(" "))  
 }  
 console.log('\nМ:')  
 for (let j = 0; j < A.length; j++) {  
 console.log(M2[j].map(num => num.toFixed(7)).join(" "))  
 }  
  
 M1 = [  
 [1, 0, 0, 0],  
 [0, 1, 0, 0],  
 [0, 0, 1, 0],  
 [0, 0, 0, 1]  
 ]  
 M2 = [  
 [1, 0, 0, 0],  
 [0, 1, 0, 0],  
 [0, 0, 1, 0],  
 [0, 0, 0, 1]  
 ]  
}  
console.log('\nS:')  
for (let j = 0; j < A.length; j++) {  
 console.log(S[j].map(num => num.toFixed(7)).join(" "))  
}  
  
function multiplyMatrix(A, B) {  
 let b = []  
 const result = []  
 for (let i = 0; i < A.length; i++)  
 result.push([])  
 for (let i = 0; i < A.length; i++) {  
 for (let j = 0; j < B.length; j++) {  
 b.push(B[j][i])  
 }  
 const a = multiply(A, b)  
 for (let j = 0; j < A.length; j++)  
 result[j].push(a[j])  
 b = []  
 }  
 return result  
}  
  
function multiply(A, b) {  
 const result = []  
 for (let i = 0; i < A.length; i++) {  
 result.push(0)  
 for (let j = 0; j < A.length; j++) {  
 result[i] += b[j] \* A[i][j]  
 }  
 }  
 return result  
}