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 «Спеціальні розділи математики»

на тему

«Обчислення власних значень та власних векторів матриць»

Виконав:

студент гр. ІС-02

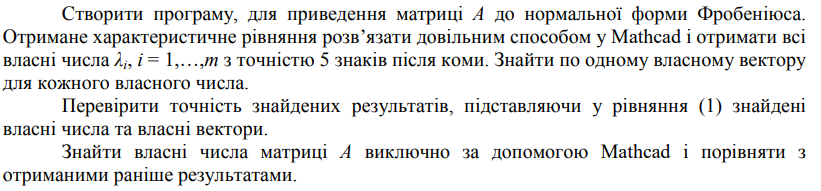
Плостак Ілля

Викладач:

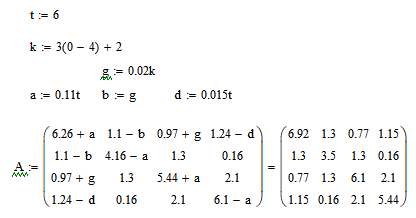
доц. Рибачук Л.В.

Київ – 2021

**1. Завдання**

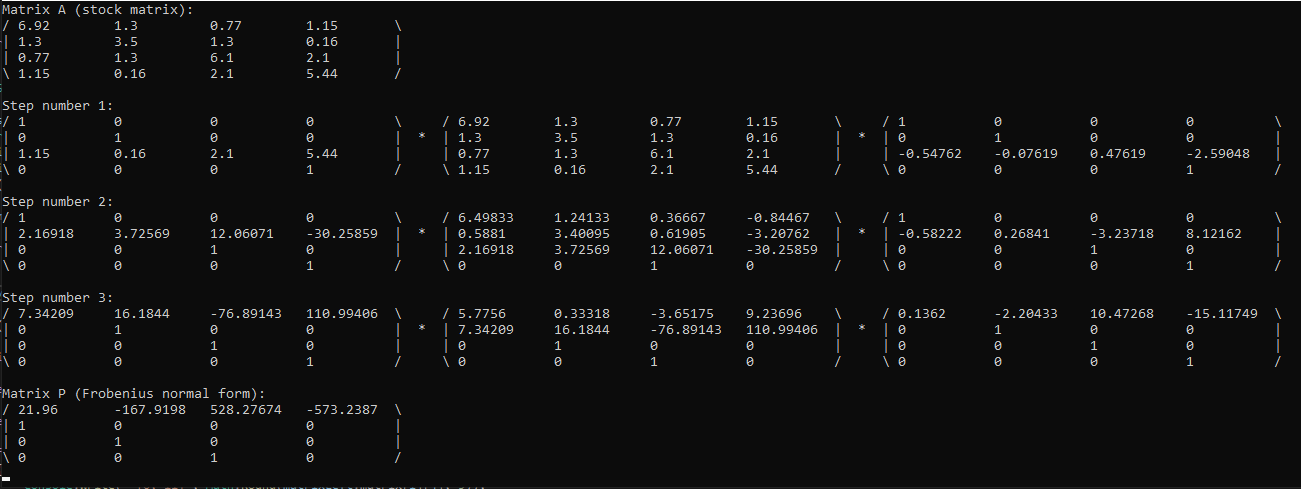


**2. Система рівнянь**



**3. Проміжні матриці Mi-1 та Mi, результуючу матрицю Р у нормальній формі**

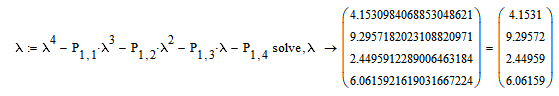
**Фробеніюса у програмі:**



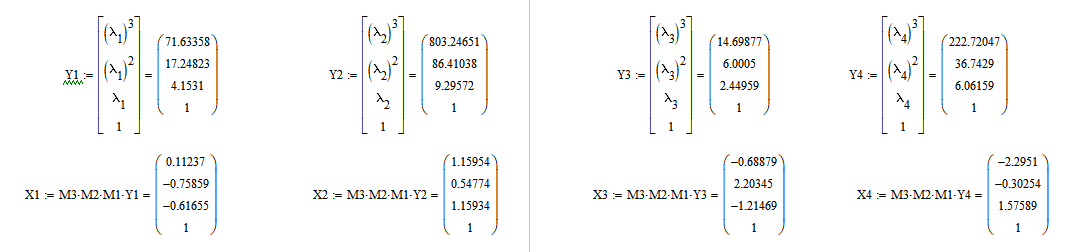
**4. Отримане характеристичне рівняння у MathCad:**



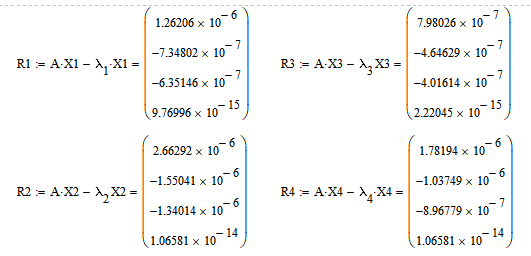
**5. Власні числа – корені характеристичного рівняння у Mathcad:**



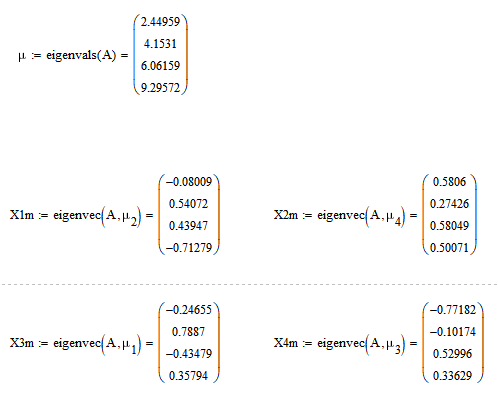
**6. Власний вектор для кожного власного числа у MathCad:**



**7. Oцінка точності обчислень (підстановка результатів у вихідне рівняння (1)):**

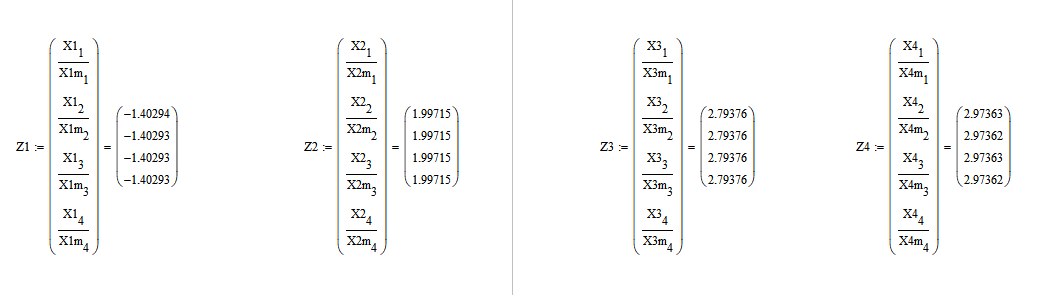


**8.** **Копія розв’язку задачі у Mathcad:**



**9.** **Порівняння власного розв’язку та розв’язку, отриманого у Mathcad:**

Усі значення векторів повинні відрізнятися на однакову константу:



Отримане рішення правильне.

**10. Лістинг програми**

[Program.cs](https://github.com/feedblackg44/kpilabs2/blob/master/CHM/Lab4/Lab4/Program.cs):

using System;

using MyClass;

namespace Lab4csh

{

class Program

{

static void Main(string[] args)

{

int size = 4;

SquareMatrix matrix = new SquareMatrix(size);

matrix.matrix4x4Init();

Console.WriteLine("Matrix A (stock matrix):");

matrix.WriteToConsole();

Console.WriteLine();

Console.WriteLine();

matrix = matrix.toFrobenius();

Console.WriteLine("Matrix P (Frobenius normal form):");

matrix.WriteToConsole();

Console.WriteLine();

Console.ReadKey();

}

}

}

[SquareMatrix.cs](https://github.com/feedblackg44/kpilabs2/blob/master/CHM/Lab4/MyClass/SquareMatrix.cs):

using System;

namespace MyClass

{

public class SquareMatrix

{

private double[][] matrix;

private int size;

public SquareMatrix(int size)

{

this.size = size;

matrix = new double[this.size][];

for (int i = 0; i < this.size; i++)

{

matrix[i] = new double[this.size];

}

}

public void matrix4x4Init()

{

matrix[0][0] = 6.92; matrix[0][1] = 1.3; matrix[0][2] = 0.77; matrix[0][3] = 1.15;

matrix[1][0] = 1.3; matrix[1][1] = 3.5; matrix[1][2] = 1.3; matrix[1][3] = 0.16;

matrix[2][0] = 0.77; matrix[2][1] = 1.3; matrix[2][2] = 6.1; matrix[2][3] = 2.1;

matrix[3][0] = 1.15; matrix[3][1] = 0.16; matrix[3][2] = 2.1; matrix[3][3] = 5.44;

}

public void matrixEInit()

{

for (int i = 0; i < this.size; i++)

{

for (int j = 0; j < this.size; j++)

{

if (i == j)

matrix[i][j] = 1;

else

matrix[i][j] = 0;

}

}

}

public void DeepCopy(in SquareMatrix matrixCopyFrom)

{

this.size = matrixCopyFrom.size;

for (int i = 0; i < this.size; i++)

{

for (int j = 0; j < this.size; j++)

{

this.matrix[i][j] = matrixCopyFrom.matrix[i][j];

}

}

}

public SquareMatrix toFrobenius()

{

int sizeRes = this.size;

SquareMatrix matrixRes = new SquareMatrix(sizeRes);

matrixRes.DeepCopy(this);

for (int i = 1; i < sizeRes; i++)

{

Console.WriteLine("Step number {0}:", i);

matrixRes.WriteStep(matrixRes.reversedM(i), matrixRes.M(i));

Console.WriteLine();

Console.WriteLine();

matrixRes = matrixRes.reversedM(i) \* matrixRes \* matrixRes.M(i);

}

return matrixRes;

}

public static SquareMatrix operator\*(SquareMatrix matrix1, SquareMatrix matrix2)

{

int sizeRes = matrix1.size;

SquareMatrix matrixRes = new SquareMatrix(sizeRes);

for (int i = 0; i < sizeRes; i++)

{

for (int j = 0; j < sizeRes; j++)

{

matrixRes.matrix[i][j] = 0;

for (int k = 0; k < sizeRes; k++)

{

matrixRes.matrix[i][j] += matrix1.matrix[i][k] \* matrix2.matrix[k][j];

}

}

}

return matrixRes;

}

public void WriteToConsole()

{

for (int i = 0; i < this.size; i++)

{

if (i == 0)

Console.Write("/");

else if (i == this.size - 1)

Console.Write("\\");

else

Console.Write("|");

for (int j = 0; j < this.size; j++)

{

Console.Write(" {0,-11}", Math.Round(this.matrix[i][j], 5));

}

if (i == 0)

Console.Write("\\");

else if (i == this.size - 1)

Console.Write("/");

else

Console.Write("|");

if (i != this.size - 1)

Console.WriteLine();

}

}

private SquareMatrix M(int iteration)

{

int sizeRes = this.size;

SquareMatrix matrixRes = new SquareMatrix(sizeRes);

matrixRes.matrixEInit();

int i = sizeRes - iteration;

for (int j = 0; j < sizeRes; j++)

{

if (j == i - 1)

matrixRes.matrix[i - 1][j] = 1 / this.matrix[i][i - 1];

else

matrixRes.matrix[i - 1][j] = - this.matrix[i][j] / this.matrix[i][i - 1];

}

return matrixRes;

}

private SquareMatrix reversedM(int iteration)

{

int sizeRes = this.size;

SquareMatrix matrixRes = new SquareMatrix(sizeRes);

matrixRes.matrixEInit();

int i = sizeRes - iteration;

for (int j = 0; j < sizeRes; j++)

{

matrixRes.matrix[i - 1][j] = this.matrix[i][j];

}

return matrixRes;

}

private void WriteStep(SquareMatrix matrixLeft, SquareMatrix matrixRight)

{

for (int i = 0; i < this.size; i++)

{

if (i == 0)

Console.Write("/");

else if (i == this.size - 1)

Console.Write("\\");

else

Console.Write("|");

for (int j = 0; j < this.size; j++)

{

Console.Write(" {0,-11}", Math.Round(matrixLeft.matrix[i][j], 5));

}

if (i == 0)

{

Console.Write("\\");

Console.Write(" ");

Console.Write("/");

}

else if (i == 1)

{

Console.Write("|");

Console.Write(" \* ");

Console.Write("|");

}

else if (i == this.size - 1)

{

Console.Write("/");

Console.Write(" ");

Console.Write("\\");

}

else

{

Console.Write("|");

Console.Write(" ");

Console.Write("|");

}

for (int j = 0; j < this.size; j++)

{

Console.Write(" {0,-11}", Math.Round(this.matrix[i][j], 5));

}

if (i == 0)

{

Console.Write("\\");

Console.Write(" ");

Console.Write("/");

}

else if (i == 1)

{

Console.Write("|");

Console.Write(" \* ");

Console.Write("|");

}

else if (i == this.size - 1)

{

Console.Write("/");

Console.Write(" ");

Console.Write("\\");

}

else

{

Console.Write("|");

Console.Write(" ");

Console.Write("|");

}

for (int j = 0; j < this.size; j++)

{

Console.Write(" {0,-11}", Math.Round(matrixRight.matrix[i][j], 5));

}

if (i == 0)

Console.Write("\\");

else if (i == this.size - 1)

Console.Write("/");

else

Console.Write("|");

if (i != this.size - 1)

Console.WriteLine();

}

}

}

}