Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Інститут комп’ютерних наук та інформаційних технологій
Кафедра програмного забезпечення

ЗВІТ

Про виконання лабораторної роботи № 4

«РОЗВ’ЯЗУВАННЯ СИСТЕМ ЛІНІЙНИХ АЛГЕБРАЇЧНИХ РІВНЯНЬ МЕТОДОМ ГАУСА ТА МЕТОДОМ LUРОЗКЛАДУ»

з дисципліни «Чисельні методи»

Лектор:
доцент кафедри ПЗ
Мельник Н.Б.

Виконав:
студ. групи ПЗ-15
Бабіля О.О.

Прийняв:
асистент кафедри ПЗ
Гарматій Г.Ю.

«___» ______ 2022 р.

∑ = ____ ___________

Львів – 2022

Мета: ознайомлення на практиці з методом Гауса та методом LUрозкладу розв’язування систем лінійних алгебраїчних рівнянь..

Теоретичні відомості:

Метод Гауса

Формулювання задачі. Задано систему лінійних алгебраїчних рівнянь

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m, \end{cases} \qquad (1)$$

для якої необхідно знайти точний розв’язок.

У матричній формі цю систему записують у вигляді

$$A \cdot X = B. \qquad (2)$$

Найвідомішим точним методом розв’язування систем лінійних алгебраїчних рівнянь є метод Гауса, суть якого полягає в тому, що систему рівнянь, яку необхідно розв’язати, зводять до еквівалентної системи з верхньою (або нижньою) трикутною матрицею. Невідомі знаходять послідовними підстановками, починаючи з останнього рівняння перетвореної системи. Точність результату та витрачений на його отримання час у більшості випадків залежить від алгоритму формування трикутної матриці системи. У загальному випадку алгоритм методу Гауса складається з двох етапів – прямого та зворотного ходу.

Під час ***прямого ходу*** СЛАР (1) перетворюють до еквівалентної системи з верхньою трикутною матрицею коефіцієнтів у вигляді

Зворотній хід дає змогу визначити елементи вектора невідомих, починаючи з останнього рівняння системи , підставляючи послідовно відповідні елементи цього вектора, отримані на попередньому кроці.

Метод LU-розкладу

Розв’язуючи систему лінійних алгебраїчних рівнянь даним методом, матрицю A коефіцієнтів системи розкладають на добуток двох матриць – нижньої трикутної матриці L, елементи головної діагоналі якої не дорівнюють нулеві та верхньої трикутної U, на головній діагоналі якої містяться одиниці, тобто

$$A = LU,$$

де

$$L = \begin{pmatrix} l_{11} & 0 & 0 & \dots & 0 \\ l_{21} & l_{22} & 0 & \dots & 0 \\ l_{31} & l_{32} & l_{33} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ l_{n1} & l_{n2} & l_{n3} & \dots & l_{nn} \end{pmatrix}, \qquad U = \begin{pmatrix} 1 & u_{12} & u_{13} & \dots & u_{1n} \\ 0 & 1 & u_{23} & \dots & u_{2n} \\ 0 & 0 & 1 & \dots & u_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix}.$$

У результаті такого розкладу матриці A систему запишемо у вигляді

$$LUX = B.$$

Введемо допоміжний вектор Y, такий що

$$UX = Y.$$

Тоді матричне рівняння подамо у вигляді

$$LY = B.$$

Розв’язування матричного рівняння виконуємо за два етапи: спочатку розв’язуємо матричне рівняння , а потім Такий підхід суттєво спрощує отримання розв’язку порівняно з методом Гауса для випадку, коли маємо кілька систем рівнянь з однаковою матрицею коефіцієнтів A, оскільки матриці L та U визначають один раз.

Розв’язування систем $LY = B$ та $UX = Y$ називають прямим та оберненим ходом відповідно.

Спочатку розглянемо ***прямий хід*** методу. Завдяки трикутній формі матриці L вектор Y легко визначають. Для цього матричне рівняння перепишемо у розгорнутому вигляді

$$\begin{cases} l_{11}y_1 & = b_1 \\ l_{21}y_1 + l_{22}y_2 & = b_2, \\ l_{31}y_1 + l_{32}y_2 + l_{33}y_3 & = b_3, \\ \dots\dots\dots\dots\dots\dots\dots\dots & \\ l_{n1}y_1 + l_{n2}y_2 + l_{n3}y_3 + \dots + l_{nn}y_n & = b_n, \end{cases}$$

звідки отримуємо компоненти вектора Y у вигляді

$$y_1 = b_1 / l_{11},$$

$$y_i = \frac{1}{l_{ii}}\left(b_i - \sum_{m=1}^{i-1} l_{im}y_m\right), \qquad i = \overline{2,n}.$$

При виконанні ***оберненого ходу*** компоненти вектора X визначають зі системи рівнянь

$$\begin{cases} x_1 + u_{12}x_2 + u_{13}x_3 + \dots + u_{1n}x_n = y_1, \\ \qquad x_2 + u_{23}x_3 + \dots + u_{2n}x_n = y_2, \\ \qquad\qquad x_3 + \dots + u_{3n}x_n = y_3, \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ \qquad\qquad\qquad\qquad x_n = y_n. \end{cases}$$

Починаючи з останнього рівняння, послідовно знаходимо компоненти вектора X за співвідношеннями

$$x_n = y_n, \;\; x_i = y_i - \sum_{m=i+1}^{n} u_{im}x_m, \quad i < n.$$

Розглянемо LU – розклад матриці A. Елементи матриці L та U визначаємо за такими формулами

$$l_{i1} = a_{i1}, \quad l_{ij} = a_{ij} - \sum_{k=1}^{j-1} l_{ik}u_{kj}, \quad i = \overline{1,n}, \quad j = \overline{2,i},$$

$$u_{ii} = 1, \quad u_{1j} = \frac{a_{1j}}{a_{11}}, \; u_{ij} = \frac{1}{l_{ii}}\left(a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj}\right), \quad i = \overline{2,j}, \quad j = \overline{2,n}.$$

Формули записують у зручнішому вигляді (метод Краута)

$$l_{ik} = a_{ik} - \sum_{m=1}^{k-1} l_{im} u_{mk}, \quad u_{kk} = 1,$$

$$u_{kj} = \frac{1}{l_{kk}} \left(a_{kj} - \sum_{m=1}^{k-1} l_{km} u_{mj} \right), \quad i = \overline{k,n}, \quad j = \overline{k+1,n}.$$

Наприклад, при $k = 1$

$$l_{i1} = a_{i1} \quad \text{для всіх } i = \overline{1,n},$$

$$u_{1j} = \frac{a_{1j}}{a_{11}} \quad \text{для всіх } j = \overline{2,n}.$$

Індивідуальне завдання

Скласти програму розв'язування системи лінійних алгебраїчних рівнянь методами Гауса з вибором головного елемента та LU -розкладу. Розв'язати систему лінійних алгебраїчних рівнянь відповідно доваріанту

$$1. \begin{cases} 0{,}65x_1 - 0{,}93x_2 + 0{,}45x_3 = -0{,}72 \\ 1{,}15x_1 + 0{,}43x_2 - 0{,}72x_3 = 1{,}24 \\ 0{,}56x_1 - 0{,}18x_2 + 1{,}03x_3 = 2{,}15 \end{cases}$$

Хід роботи:

Метод Гауса з вибіром головного елемента:

Запишемо матрицю A та B:

A=

$$\begin{vmatrix} 0.65 & -0.93 & 0.45 \\ 1.15 & 0.43 & -0.72 \\ 0.56 & -0.18 & 1.03 \end{vmatrix}$$

B=

$$\begin{vmatrix} -0.72 \\ 1.24 \\ 2.25 \end{vmatrix}$$

В стовпцю 1 найбільший елемент по модулю = 1.15,отже він і є головним.
Зміним перший рядок та другий місцями:

A= B=

$$\begin{pmatrix} 1.15 & 0.43 & -0.72 \\ 0.65 & -0.93 & 0.45 \\ 0.56 & -0.18 & 1.03 \end{pmatrix} \quad \begin{pmatrix} 1.24 \\ -0.72 \\ 2.25 \end{pmatrix}$$

Домножим 1 рядок на -0.56 та додамо його до 2 рядка, домножим 1 рядок на -0.48 та додамо його до 3 рядка, отримаємо:

A= B=

$$\begin{pmatrix} 1.15 & 0.43 & -0.72 \\ 0 & -1.17 & 0.85 \\ 0 & -0.38 & 1.3 \end{pmatrix} \ \begin{matrix} -0.56 \\ -0.48 \\ \ \end{matrix} \quad \begin{pmatrix} 1.24 \\ -1.42 \\ 1.64 \end{pmatrix} \ \begin{matrix} -0.56 \\ -0.48 \\ \ \end{matrix}$$

В стовпцю 2 найбільший елемент по модулю = 1.17,отже він і є головним.
Ніякі заміни не потрібно проводити.

Домножим 2 рядок на -0.32 та додамо його до 3 рядка , отримаємо:

A= B=

$$\begin{pmatrix} 1.15 & 0.43 & -0.72 \\ 0 & -1.17 & 0.85 \\ 0 & 0 & 1.09 \end{pmatrix} \ \begin{matrix} \ \\ -0.32 \\ \ \end{matrix} \quad \begin{pmatrix} 1.24 \\ -1.42 \\ 2.11 \end{pmatrix} \ \begin{matrix} - \\ -0.32 \\ \ \end{matrix}$$

Ми звели матрицю до остаточного варіанту
Знайдемо розв’язки СЛАР:

$X_3=B_1/A_{33}=1.93$

$X_2=(B_2-X_3*A_{23})/A_{22}=2.62$

$X_1=(B_1-X_3*A_{13}-X_2*A_{12})/A_{11}=1.3$

Метод Гауса з LU розкладом:

Запишемо матрицю A та B:

A=

$$\begin{vmatrix} 0.65 & -0.93 & 0.45 \\ 1.15 & 0.43 & -0.72 \\ 0.56 & -0.18 & 1.03 \end{vmatrix}$$

B=

$$\begin{vmatrix} -0.72 \\ 1.24 \\ 2.25 \end{vmatrix}$$

Знайдемо матрицицю L та U:

Використовуючи формли:

$l_{11}=a_{11},u_{12}=a_{12}/l_{11},u_{13}=a_{13}/l_{11}$
$l_{21}=a_{21},l_{22}=a_{22}-l_{21}*u_{12},u_{23}=(a_{23}-l_{21}*u_{13})/l_{22}$
$l_{31}=a_{31}, l_{32}=a_{32}-l_{31}*u_{12},l_{33}=a_{33}-l_{31}*u_{13}- l_{32}*u_{23},u_{23}=(a_{23}-l_{21}*u_{13})/l_{22}$

Проведем заміну:

$l_{11}=0.65$
$u_{12}=-0.93/0.65=-1.43$
$u_{13}=0.45/0.65=0.69$
$l_{21}=1.15$
$l_{22}=0.43-1.15*1.43=2.07$
$u_{23}=(-0.72-1.15*0.69)/2.07=-0.73$
$l_{31}=0.56$
$l_{32}=-0.18-1.43*0.56=0.62$
$l_{33}=1.03-0.56*0.69- 0.62*-0.73=1.09$

Складемо матрицю L та U:

L=

$$\begin{pmatrix} 0.65 & 0 & 0 \\ 1.15 & 2.07 & 0 \\ 0.56 & 0.62 & 1.09 \end{pmatrix}$$

U=

$$\begin{pmatrix} 1 & -1.43 & 0.69 \\ 0 & 1 & -0.73 \\ 0 & 0 & 1 \end{pmatrix}$$

Обчислим вектори Y:

$y_1=b_1/l_{11}$
$y_2=1/l_{22}*(b_2-l_{21}*y_1)$
$y_3=1/l_{33}*(b_3-l_{31}*y_1-l_{32}*y_2)$

$y_1=-0.72/0.65=-1.1$
$y_2=1/2.07*(1.24-1.15*-1.1)=1.21$
$y_3=1/1.09*(2.25-0.56*-1.1-0.62*1.21)=1.93$

Знайдемо X
X3=Y3= 1.93

$X2=Y2-u_{23}*x3= 2.62$
$X1=y1-u_{13}*x3-u_{12}*x2= 1.3$

Код програми:

Source.c

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <conio.h>
#include <malloc.h>
void showMatrix(float** array, int n, int m)
{
	printf("\n");
	for (int i = 0; i < n; i++)
	{
		for (int j = 0; j < m; j++)
			printf("%f ", array[i][j]);
		printf("\n");
	}
	printf("\n\n---------------------------------------\n");
}
void swapRow(float** array, int m, int k, int l)
{
	float* temp = (float*)malloc(m * sizeof(float));
	if (k != l) {
		for (int i = 0; i < m; i++)
		{
			temp[i] = array[k][i];
		}
		for (int i = 0; i < m; i++)
		{
			array[k][i] = array[l][i];
		}
		for (int i = 0; i < m; i++)
		{
			array[l][i] = temp[i];
		}
	}
	free(temp);
}
float sumOfVectors(float* y, float** l, int n)
{
	float s = 0;
	for (int i = 0; i < n; i++)
	{
		s += l[n][i] * y[i];
		if (l[n][i] * y[i] < 0)
		{
			printf("-(%f*%f)", l[n][i], y[i]);
		}
	}
	return s;
}
float sumOfVectors2(float* x, float** u, int k, int n)
{
	float s = 0;
	for (int i = n - k/**/; i > 0; i--)
	{
```

```
			s += u[k][n - i] * x[n - i];
			printf("-(%f*%f)", u[k][n - i], x[n - i]);
		}
		return s;
}
void chooseElementMatrix(float** array, int n, int j)
{
		float max = fabs(array[j][j]);
		for (int i = j; i < n; i++)
		{
			if (fabs(array[i][j]) > max)
			{
				max = fabs(array[i][j]);
				swapRow(array, n + 1, j, i);
			}
		}
}
float sumOfRowS(float** array, int m, int k)
{
		int p = 0;
		float res = 0;
		for (int i = 0; i < m; i++)
		{
			if (i > 1 + k && i < m)
			{
				printf("%f", array[k][i]);
				p++;
			}
			if ((i != k) && (i != m - 1))
				res += array[k][i];
		}
		if (p == 0)
		{
			printf("0");
		}
		//printf("\n%f\n",res);
		return res;
}
void subtractionRow(float** array, int n, int m, int l, int k, float coefficient)
{
		for (int i = 0; i < m; i++) {
			//printf("\n-%f %f- \n", array[l][i], array[k][i]);
			if (k != l)
				array[l][i] = array[l][i] - array[k][i] * coefficient;
		}
}
void methodGauss(float** array, int n, int m)
{
		printf("\nGauss method:\n");
		float** system = (float**)malloc(n * sizeof(float*));
		for (int i = 0; i < n; i++)
		{
			system[i] = (float*)malloc(m * sizeof(float));
			for (int j = 0; j < m; j++)
			{
				system[i][j] = array[i][j];
			}
		}
		for (int i = 0; i < n - 1; i++)
		{
			chooseElementMatrix(system, n, i);
```

```c
            printf("\nThe %ith row:", i);
            for (int j = i; j < n; j++)
            {
                  subtractionRow(system, n, m, j, i, system[j][i] / system[i][i]);
                  //printf("\n|%i %i|\n",j,i);
            }
            showMatrix(system, n, m);
      }
      float* X = (float*)malloc(n * sizeof(float));
      printf("Answers:\n");
      for (int i = 1; i <= n; i++)
      {
            float b = system[n - i][m - 1];
            printf("X%i=(%f-", n - i + 1, b);
            X[n - i] = (b - sumOfRowS(system, m, n - i)) / system[n - i][n - i];
            for (int j = 0; j < n - i; j++)
                  system[j][n - i] *= X[n - i];
            printf(")/%f=%f\n", system[n - i][n - i], X[n - i]);
      }
      free(system);
      free(X);
}
void methodLU(float** array, int n, int m)
{
      printf("\nLU method:\n");
      float** A = (float**)malloc(n * sizeof(float*));
      float** L = (float**)malloc(n * sizeof(float*));
      float** U = (float**)malloc(n * sizeof(float*));
      for (int i = 0; i < n; i++)
      {
            A[i] = (float*)malloc(n * sizeof(float));
            U[i] = (float*)malloc(n * sizeof(float));
            L[i] = (float*)malloc(n * sizeof(float));
            for (int j = 0; j < n; j++)
            {
                  A[i][j] = array[i][j];
                  L[i][j] = 0;
                  if (i == j)
                        U[i][j] = 1;
                  else
                        U[i][j] = 0;
            }
      }
      for (int i = 0; i < n; i++)
      {
            L[i][0] = A[i][0];
            U[0][i] = A[0][i] / L[0][0];
      }
      float sum;
      for (int i = 1; i < n; i++)
      {
            for (int j = 1; j < n; j++)
            {
                  if (i >= j)
                  {
                        sum = 0;
                        for (int k = 0; k < j; k++)
                              sum += L[i][k] * U[k][j];

                        L[i][j] = A[i][j] - sum;
                  }
```

```
				else
				{
					sum = 0;
					for (int k = 0; k < i; k++)
						sum += L[i][k] * U[k][j];

					U[i][j] = (A[i][j] - sum) / L[i][i];
				}
			}
		}
		printf("\nL matrix:\n");
		showMatrix(L, n, n);
		printf("\nU matrix:\n");
		showMatrix(U, n, n);
		float* Y = (float*)malloc(n * sizeof(float));
		Y[0] = array[0][n] / L[0][0];
		printf("\nY1=%f/%f=%f\n", array[0][n], L[0][0], Y[0]);
		for (int i = 1; i < n; i++)
		{
			printf("\nY%i=1/%f*(%f)", i + 1, L[i][i], array[i][n]);
			Y[i] = 1 / L[i][i] * (array[i][n] - sumOfVectors(Y, L, i));
			printf("=%f\n", Y[i]);
		}
		float* X = (float*)malloc(n * sizeof(float));
		for (int i = 0; i < n; i++)
		{
			X[i] = 0;
		}
		X[n - 1] = Y[n - 1];
		printf("\n---------------------------------------\n");
		printf("\nAnswers:\n");
		printf("\nX%i=Y%i=%f\n", n, n, X[n - 1]);
		for (int i = n - 2; i >= 0; i--)
		{
			printf("\nX%i=%f", i + 1, Y[i]);
			X[i] = Y[i] - sumOfVectors2(X, U, i, n);
			printf("=%f\n", X[i]);
		}
}
int main()
{
		int n = 3, m = 4;
		float** systemS = NULL; int flag;
		printf("What do you prefer?\nStandart matrix(0)\nInput from console(1)\nInput from
file(2)\n");
		scanf_s("%i", &flag);
		if (flag == 0)
		{
			systemS = (float**)malloc(n * sizeof(float*));
			for (int i = 0; i < n; i++)
			{
				systemS[i] = (float*)malloc(m * sizeof(float));
				/*for (int j = 0;j < m;j++)
				{
					system[i][j] = i * m + j+1;
				}*/
			}
			systemS[0][0] = 0.65;
			systemS[0][1] = -0.93;
			systemS[0][2] = 0.45;
			systemS[0][3] = -0.72;
```

```
		systemS[1][0] = 1.15;
		systemS[1][1] = 0.43;
		systemS[1][2] = -0.72;
		systemS[1][3] = 1.24;
		systemS[2][0] = 0.56;
		systemS[2][1] = -0.18;
		systemS[2][2] = 1.03;
		systemS[2][3] = 2.25;
	}
	if (flag == 1)
	{
		printf("Enter the amount of X: ");
		scanf("%i", &n);
		m = n + 1;
		systemS = (float**)malloc(n * sizeof(float*));
		for (int i = 0; i < n; i++)
		{
			systemS[i] = (float*)malloc(m * sizeof(float));
			for (int j = 0; j < m; j++)
			{
				scanf("%f ", &systemS[i][j]);
			}
		}
	}
	if (flag == 2)
	{
		char fln[25];
		printf("\nEnter the name of file: ");
		scanf("%s", &fln);
		FILE* fl;
		fl = fopen(fln, "r");
		printf("Enter the amount of X: ");
		scanf("%i", &n);
		m = n + 1;
		systemS = (float**)malloc(n * sizeof(float*));
		for (int i = 0; i < n; i++)
		{
			systemS[i] = (float*)malloc(m * sizeof(float));
			for (int j = 0; j < m; j++)
			{
				if (fscanf(fl, "%f", &systemS[i][j]) != EOF)
				{
					fscanf(fl, "%f", &systemS[i][j]);
				}
				else
				{
					printf("Invalid format. Check your matrix");
					return 0;
				}
			}
		}
	}
	printf("\nYour matrix is:\n");
	showMatrix(systemS, n, m);
	printf("\n=======================================\n");
	methodGauss(systemS, n, m);
	printf("\n=======================================\n");
	methodLU(systemS, n, m);
	printf("\n=======================================\n");
}
```

Вигляд виконаної програми:

```
What do you prefer?
Standart matrix(0)
Input from console(1)
Input from file(2)
0

Your matrix is:

0.650000 -0.930000 0.450000 -0.720000
1.150000 0.430000 -0.720000 1.240000
0.560000 -0.180000 1.030000 2.250000


---------------------------------------

=======================================

Gauss method:

The 0th row:
1.150000 0.430000 -0.720000 1.240000
0.000000 -1.173043 0.856956 -1.420870
0.000000 -0.389391 1.380609 1.646174


---------------------------------------

The 1th row:
0.000000 0.430000 -0.720000 1.240000
0.000000 -1.173043 0.856956 -1.420870
0.000000 0.000000 1.096142 2.117831


---------------------------------------
Answers:
X3=(2.117831-0)/1.096142=1.932077
X2=(-1.420870--1.420870)/-1.173043=2.622729
X1=(1.240000--1.3910951.240000)/1.150000=1.307236

=======================================

LU method:

L matrix:

0.650000 0.000000 0.000000
1.150000 2.075385 0.000000
0.560000 0.621231 1.096142


---------------------------------------

U matrix:

1.000000 -1.430769 0.692308
0.000000 1.000000 -0.730541
0.000000 0.000000 1.000000


---------------------------------------

Y1=-0.720000/0.650000=-1.107692

Y2=1/2.075385*(1.240000)-(1.150000*-1.107692)=1.211268

Y3=1/1.096142*(2.250000)-(0.560000*-1.107692)=1.932077

---------------------------------------

Answers:

X3=Y3=1.932077

X2=1.211268-(1.000000*0.000000)-(-0.730541*1.932077)=2.622729

X1=-1.107692-(1.000000*0.000000)-(-1.430769*2.622729)-(0.692308*1.932077)=1.307236

=======================================

C:\Users\home\source\repos\ЧМ04 C\Debug\ЧМ04 C.exe (процесс 4132) завершил работу с кодом 0.
```

Висновки:

На даній лабораторній роботі я ознайомився на практиці з методом Гауса з вибором головного елемента та методом Гауса з LU розкладом для розв’язування систем лінійних алгебраїчних рівнянь. Розв’язав СЛАР,

згідно до індивідуального завдання:

$$1. \begin{cases} 0,65x_1 - 0,93x_2 + 0,45x_3 = -0,72 \\ 1,15x_1 + 0,43x_2 - 0,72x_3 = 1,24 \\ 0,56x_1 - 0,18x_2 + 1,03x_3 = 2,15 \end{cases}$$

Розв’язками стали x_1= 1.307236, x_2= 2.622729, x_3= 1.932077.