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
На тему:
«Розв’язування систем лінійних алгебраїчних рівнянь методом Гауса та методом LU-розкладу»
з дисципліни «Чисельні методи»

Лектор:
доцент каф. ПЗ
Мельник Н. Б.
Виконав:
ст. гр. ПЗ-11
Морозов О. Р.
Прийняла:
доцент каф. ПЗ
Мельник Н. Б.

« __ » ________ 2022 р.

∑ = _____ _________ .

Львів – 2022

Тема: розв’язування систем лінійних алгебраїчних рівнянь методом Гауса та методом LU-розкладу
Мета: ознайомлення на практиці з методом Гауса та методом LU-розкладу розв’язування систем лінійних алгебраїчних рівнянь.

Теоретичні відомості

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 перетворюють до еквівалентної системи з верхньою трикутною матрицею. 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. 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. При виконанні оберненого ходу компоненти вектора X визначають зі системи рівнянь.

Індивідуальне завдання

Варіант 2

Написати програму розв’язку матриці
0.62*a + 0.56*b - 0.43*c = 1.16
1.32*a - 0.88*b + 1.76*c = 2.07
0.73*a + 1.42*b - 0.34*c = 2.18

Хід роботи

Код програми

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <math.h>

double f(double x);
double fp(double x);
double fpp(double x);
double fi(double x, double a, double b);

void retryInput();

void NewtonMethod(double n, double m, double eps);
void IterationMethod(double a, double b, double eps);

int main() {

	std::cout << "Here we have an Expression: x^3 - 6x - 8 = 0\n" <<
"Now you have to enter a limits and eps\n" << "Expression has 1 root
between (2,5; 3)\n";

	double a = 0, b = 0, eps = 0;
	while (1){
		printf("Enter Eps:");
		while (scanf("%lf", &eps) != 1 && eps <= 0){
			retryInput();
			printf("Wrong input\ntry again:");
		}
		printf("Enter A:");
		while (scanf("%lf", &a) != 1){
			retryInput();
			printf("Wrong input\nTry again:");
		}
		printf("Enter B:");
```

```
		while (scanf("%lf", &b) != 1){
			retryInput();
			printf("Wrong input\nTry again:");
		}
		if ((f(a) * f(b)) >= 0 || fabs(a - b) < eps){
			printf("Limits not correct\nTry again\n");
		}
		else{
			break;
		}
	}
	if (f(a) * fpp(a) > 0) {
		NewtonMethod(a, b, eps);
	}
	else if(f(b) * fpp(b) > 0) {
		NewtonMethod(b, a, eps);
	}
	IterationMethod(a, b, eps);
	return 0;
}

double f(double x) {
	return ((x * x * x) - (6 * x) - 8);
}

double fp(double x) {
	return ((3 * x * x) - 6);
}

double fpp(double x) {
	return 6 * x;
}

void retryInput(){
	while (getchar() != '\n');
}

void NewtonMethod(double n, double m, double eps) {
	double x = n;
	double xi = x;
	x = xi - (f(xi)) / (fp(xi));
	int i = 1;
	while (fabs(xi - x) > eps) {
		xi = x;
		x = xi - (f(xi)) / (fp(xi));
		i++;
	}
```

```
	printf("\tNewton Method\n\tSolution\n\n  X  = \t%.10lf\nf(x) =\t%.10lf \nIteration = %d\n", x, f(x), i);
}

void IterationMethod(double a, double b, double eps) {
	double x = b, xi = fi(x, a, b);
	int i = 1;
	while (fabs(xi - x) >= eps) {
		x = xi;
		xi = fi(x, a, b);
		i++;
	}
	printf("\tNewton Method\n\tSolution\n\n  X  = \t%.10lf\nf(x) =\t%.10lf \nIteration = %d\n", x, f(x), i);
}

double fi(double x, double a, double b) {

	double k;
	if ( a >= 0 && b >= 0) {
		k = fp(b) / (4./3);
	}
	else if(a <= 0 && b <= 0) {
		k = -1 * fp(b) / (4./3);
	}
	else {
		if (a <= 0) {
			a = 0;
		}
		else {
			b = 0;

		}
		k = fp(b) / 6;
	}
	return (x - (f(x) / k));
}
```

Результат роботи

Консоль отладки Microsoft Visual Studio

```
Custom matrix - U
Standard 3x3 matrix - S
Close - E
S
Standart Matrix
--------------------------------------------------
 0.62 +  0.56  -0.43 =  1.16
 1.32  -0.88 +  1.76 =  2.07
 0.73 +  1.42  -0.34 =  2.18
--------------------------------------------------

Gauss method


The 1th row
--------------------------------------------------
 0.62 +  0.56  -0.43 =  1.16
    0  -2.072 + 2.675 = -0.3997
    0 + 0.7606 + 0.1663 = 0.8142
--------------------------------------------------

The 2th row
--------------------------------------------------
 0.62 +  0.56  -0.43 =  1.16
    0  -2.072 + 2.675 = -0.3997
    0 +     0 + 1.148 = 0.6675
--------------------------------------------------

The solution is:
x[ 1] =  1.422
x[ 2] = 0.9433
x[ 3] = 0.5813

LU method

A
--------------------------------------------------
 0.62 +  0.56  -0.43
 1.32  -0.88 +  1.76
 0.73 +  1.42  -0.34
```

Консоль отладки Microsoft Visual Studio

```
    0 +     0 + 1.148 = 0.6675
--------------------------------------------------

The solution is:
x[ 1] =  1.422
x[ 2] = 0.9433
x[ 3] = 0.5813

LU method

A
--------------------------------------------------
 0.62 +  0.56  -0.43
 1.32  -0.88 +  1.76
 0.73 +  1.42  -0.34
--------------------------------------------------
L
--------------------------------------------------
 0.62 +     0 +     0
 1.32  -2.072 +     0
 0.73 + 0.7606 + 1.148
--------------------------------------------------
U
--------------------------------------------------
    1 + 0.9032  -0.6935
    0 +     1  -1.291
    0 +     0 +     1
--------------------------------------------------

Y-matrix
Y[ 1] =  1.871
Y[ 2] = 0.1929
Y[ 3] = 0.5813
0.5813 - 00.1929 - -0.75051.871 - 0.4489
The solution is:
X[ 1] =  1.422
X[ 2] = 0.9433
X[ 3] = 0.5813
E
```

Висновок

Виконуючи лабораторну роботу №4, я навчився програмувати розв’язок матриці методом Гаусса та методом LU - розкладу.