# МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Національний університет “Львівська політехніка”**



**Інститут післядипломної освіти**

**ЗВІТ**

**Про виконання лабораторної роботи №3**

**«Ітераційні методи розв’язування системи лінійних рівнянь»**

**з дисципліни «Чисельні методи»**

Виконав:

слухач групи ПЗС-11

Гринчук Тарас

Прийняла:

ст. викл. Мельник Н. Б.

« »\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2014 р.

∑ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

ЛЬВІВ – 2014

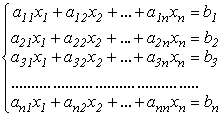
**Тема роботи**: Ітераційні методи розв’язування системи лінійних рівнянь.

**Мета роботи**: Ознайомлення на практиці з ітераційнними методами розв’язування систем лінійних алгебраїчних рівнянь.

## 1. Теоретичні відомості

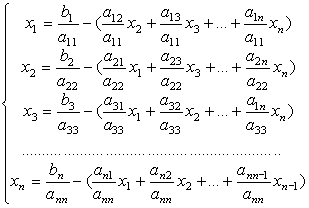
**Метод послідовних наближень (Якобі)**

Розглянемо систему лінійних алгебраїчних рівнянь виду:



Для розв’язання системи методами послідовних наближень необхідно виконати наступні кроки:

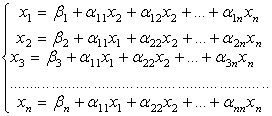
Кожне рівняння системи розділити на діагональний елемент  де k=1,2...n, n – кількість рівнянь в системі, і перетворити кожне рівняння системи відносно координат вектора, індекс якого співпадає з номером рівняння:



Якщо ввести перепозначення

m1_t1_lecture3_image015, m1_t1_lecture3_image017,

де k=1,2…n; i=1,2…n, то система матиме вигляд:



Така система називається зведеною до нормального вигляду.

.

Якщо деяким чином визначити, так званий, вектор початкових значень , який знаходиться в правій частині, то можна отримати певні значення вектора **,,…**

.

Вибір початкового наближення 

В якості вектора початкових наближень  вибирають:

* вектор, в якого всі координати хі дорівнюють 0;
* вектор, в якого всі координати хі дорівнюють 1;
* вектор, координати m1_t1_lecture3_image030якого дорівнюють координатам вектора вільних членів m1_t1_lecture3_image032;
* координати вектору m1_t1_lecture3_image034 вибирають в результаті аналізу особливостей об’єкту дослідження та задачі, яка розв’язується.

Умова закінчення ітераційного процесу

.

Умови збіжності ітераційного процесу (теорема про збіжність)

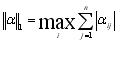
Ітераційний процес пошуку розв’язку системи лінійних алгебраїчних рівнянь виду (1) наближеними методами збігається, якщо будь-яка канонічна норма матриці .

Канонічні норми матриці



Норма матриці - додатне число, яке визначається за такими умовами:

перша канонічна норма – це максимальна з сум модулів елементів матриці коефіцієнтів m1_t1_lecture3_image065 по стрічках:



друга канонічна норма – це максимальна з сум модулів елементів матриці коефіцієнтів m1_t1_lecture3_image065 по стовбцях:

m1_t1_lecture3_image069,

третя канонічна норма – це корінь квадратний з сум квадратів модулів всіх елементів матриці коефіцієнтів m1_t1_lecture3_image065:

m1_t1_lecture3_image071.

**Метод Зейделя**

Метод можна розглядати як модифікацію метода Якобі.

Основна ідея - при обчисленні чергового (n+1)-го наближення до невідомого  xi при i >1 використовують вже знайдені (n+1)-е наближення до  x1, x2, ..., xi - 1, а не n-е наближення, як в методі Якобі.

Розрахункова формула методу:

image034.gif (1729 bytes),

 i = 1, 2, ... m..

Умова збіжності і критерій закінчення ітерацій аналогічні як в методі Якобі.

Якщо матриця A - симетрична і додатньо визначена, то при будь-якому виборі початкового наближення метод Зейделя збіжний.

**2. Хід роботи**

**Завдання 1 (варіант 5).** Розв’язати систему лінійних рівнянь методом ітерацій з точністю до 0,001, наперед оцінивши число необхідних для цього кроків.

Система рівнянь вже перетворена до вигляду:

, зручному для ітерацій. Кількість кроків визначити із співвідношення:





**Завдання 2.** Розв’язати систему лінійних рівнянь методом Зейделя з точністю до 0,001.



Звіт до лабораторної роботи повинен містити такі структурні елементи:

1. Титульний аркуш.
2. Тема.
3. Мета.
4. Короткі теоретичні відомості.
5. Алгоритм розв’язку СЛАР.
6. Текст програми з коментарями.
7. Вигляд реалізованої програми.
8. Висновки.

**3. Текст програми методу Якобі**

#include <iostream>

#include <conio.h>

#include <clocale>

#include <math.h>

#include <cmath>

using namespace std;

int const N = 4;

const double eps = 0.001;

//якщо 0 - то ввід вхідних даних з клавіатури

int const testMode = 1;

double A[N][N], X[N], F[N];

//зчитування масиву з клавіатури

void input(double a[N][N], int n, int m) {

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

for(int j = 0; j < m; j++) {

cout << "a[" << i << "][" << j << "]: ";

cin >> a[i][j];

}

}

//виведення матриці на екран

void display(double a[N][N], int n, int m) {

for(int i=0; i < n; i++) {

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

cout << a[i][j] << "\t";

if(m > 1) cout<<endl;

}

}

void inputVector(double a[N], int m) {

for(int j = 0; j < m; j++) {

cout << "f[" << j << "]: ";

cin >> a[j];

}

}

void displayVector(double X[N], int n) {

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

cout << X[j] << "\t";

}

//перевірка на евклідову норму матриці

int EvklidTest(double a[N][N]) {

double alpha[N][N];

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

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

if(i!=j) alpha[i][j] = (-a[i][j])/(a[i][i]);

else alpha[i][j] = 0;

double Ek = 0;

for(int i=0; i < N; i++) {

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

Ek += alpha[i][j] \* alpha[i][j];

}

Ek = pow(Ek,0.5);

if(Ek > 1.0) {

cout<<"\n\n\nЕвклiдову норма матрицi "<<Ek<<" > 1; Програму буде завершено !\n";

\_getch();

return 0;

}

return 1;

}

void main() {

setlocale(LC\_ALL, "Ukrainian");

if(testMode) {

A[0][0] = -0.82;

A[0][1] = -0.34;

A[0][2] = -0.12;

A[0][3] = 0.15;

A[1][0] = 0.11;

A[1][1] = -0.77;

A[1][2] = -0.15;

A[1][3] = 0.32;

A[2][0] = 0.05;

A[2][1] = -0.12;

A[2][2] = -0.86;

A[2][3] = -0.18;

A[3][0] = 0.12;

A[3][1] = 0.08;

A[3][2] = 0.06;

A[3][3] = -1.0;

F[0] = -1.33;

F[1] = 0.84;

F[2] = -1.16;

F[3] = 0.57;

} else {

cout<<"Введiть матрицю А:\n";

input(A, N, N);

cout<<"\nВведiть стовпець F:\n";

inputVector(F, N);

}

//Вивід a та f

cout<<"А:\n";

display(A, N, N);

cout<<"\nB:\n";

displayVector(F, N);

if(!EvklidTest(A)) return;

double TempX[N];

//норма, розрахована ​​як найбільша різниця компонент стовпця іксів сусідніх ітерацій

double norm;

//обчислення розвязків

do {

for (int i = 0; i < N; i++) {

TempX[i] = F[i];

for (int g = 0; g < N; g++) {

if (i != g)

TempX[i] -= A[i][g] \* X[g];

}

TempX[i] /= A[i][i];

}

norm = fabs(X[0] - TempX[0]);

for (int h = 0; h < N; h++) {

if (fabs(X[h] - TempX[h]) > norm)

norm = fabs(X[h] - TempX[h]);

X[h] = TempX[h];

}

} while (norm > eps);

//виведення результату

cout<<"\n\nX:\n";

displayVector(X, N);

\_getch();

}

**4. Текст програми методу Зейделя**

#include <iostream>

#include <conio.h>

#include <clocale>

#include <math.h>

#include <cmath>

using namespace std;

int const N = 3;

const double eps = 0.001;

//якщо 0 - то ввід вхідних даних з клавіатури

int const testMode = 1;

double A[N][N], X[N], F[N];

//зчитування масиву з клавіатури

void input(double a[N][N], int n, int m) {

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

for(int j = 0; j < m; j++) {

cout << "a[" << i << "][" << j << "]: ";

cin >> a[i][j];

}

}

//виведення матриці на екран

void display(double a[N][N], int n, int m) {

for(int i=0; i < n; i++) {

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

cout << a[i][j] << "\t";

if(m > 1) cout<<endl;

}

}

void inputVector(double a[N], int m) {

for(int j = 0; j < m; j++) {

cout << "f[" << j << "]: ";

cin >> a[j];

}

}

void displayVector(double X[N], int n) {

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

cout << X[j] << "\t";

}

//перевірка на евклідову норму матриці

int EvklidTest(double a[N][N]) {

double alpha[N][N];

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

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

if(i!=j) alpha[i][j] = (-a[i][j])/(a[i][i]);

else alpha[i][j] = 0;

double Ek = 0;

for(int i=0; i < N; i++) {

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

Ek += alpha[i][j] \* alpha[i][j];

}

Ek = pow(Ek,0.5);

if(Ek > 1.0) {

cout<<"\n\n\nЕвклiдова норма матрицi "<<Ek<<" > 1; Програму буде завершено !\n";

\_getch();

return 0;

}

return 1;

}

void main() {

setlocale(LC\_ALL, "Ukrainian");

if(testMode) {

A[0][0] = 1.0;

A[0][1] = 3.9;

A[0][2] = 1.0;

A[1][0] = 1.0;

A[1][1] = 2.7;

A[1][2] = -2.0;

A[2][0] = 2.0;

A[2][1] = -4.4;

A[2][2] = 1.0;

F[0] = -10.2;

F[1] = -11.0;

F[2] = -5.0;

} else {

cout<<"Введiть матрицю А:\n";

input(A, N, N);

cout<<"\nВведiть стовпець F:\n";

inputVector(F, N);

}

//Вивід a та f

cout<<"А:\n";

display(A, N, N);

cout<<"\nB:\n";

displayVector(F, N);

if(!EvklidTest(A)) return;

double norm, v, s;

int i;

for (i = 0; i < N; i++) X[i] = 0;

do {

norm = 0;

for (i = 0; i < N; i++) {

s = 0;

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

if (i != j) s += A[i][j] \* X[j];

v = X[i];

X[i] = (F[i] - s) / A[i][i];

norm = fabs(X[i] - v);

}

} while (norm > eps);

//виведення результату

cout<<"\n\nX:\n";

displayVector(X, N);

\_getch();

}

**5. Результати виконання програм**

Запустимо програму методу Якобі на виконання (рис. 5.1):

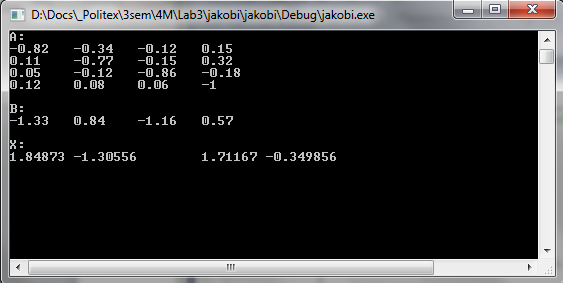


Рис. 5.1. Метод Якобі

Запустимо програму методу Зейделя на виконання (рис. 5.2):

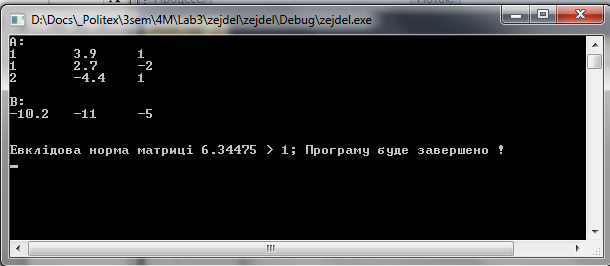


Рис. 5.2. Метод Зейделя

## ВИСНОВКИ

В ході даної лабораторної роботи я оволодів ітераційними методами розв’язування систем лінійних алгебраїчних рівнянь, а саме:

1)методом Якобі;

2)методом Зейделя.