**Лабораторная работа**

**«Итерационный степенной метод»**

**Арсений Анищенко**

**Группа 2**

1. **Постановка задачи.**

Разработать программу вычисления наибольшего и второго по величине модуля собственных значений и соответствующих им собственных векторов симметричной матрицы.

Матрицу задать следующим образом (воспользоваться программой лабораторной работы «Решение систем на основе разложения симметричных матриц»):

* недиагональные элементы *ai,j*, *i<j*, выбираются из чисел 0, –1, –2, –3, *–*4 произвольным образом; если *i>j*, то полагается *ai,j*=*aj,i*.
* *ai,i=*, 2≤*i*≤*n*;
* *a*11*=*.

Для вычислений выбрать *n* – одно из чисел в пределах от 10 до 12. В качестве языка программирования выбрать C или C++, использовать тип float.

Для вычисления наибольшего по модулю собственного значения и соответствующего собственного вектора использовать формулы из пункта Случай 1 (файл «Степенной метод»). В формуле λ1≈ (формула вспомогательная, не для счета) выбирать такое *i*, для которого достигается ). В этом случае справедливо (это уже для счета) λ1≈.

Для вычисления наибольшего по модулю собственного значения использовать также формулу λ1≈.

Вывести на печать *uk* на итерациях 46–50 (*k=*46,…,50).

Для обоих случаев: вывести на печать приближенное λ1 для *k=*46,…,50; вычислить вектор *vk+*1–λ1*uk* (это, для проверки насколько оказались хороши вычисленные λ1 и *uk*, вектор «погрешности» *Auk*–λ1*uk*) для *k=*50, вычислить и вывести на печать кубическую норму (максимум-норму) этого вектора.

Для вычисления второго по величине модуля собственного значения использовать формулу λ2≈. Выбирать такое *i*, для которого достигается ).

Рассмотреть три случая:

1) *m=*30, λ1 берется при *k=*50 по формуле для не обязательно симметричной матрицы;

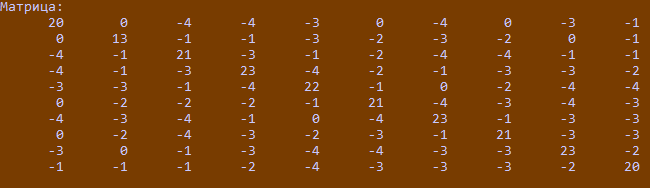
2) *m=*50, λ1 берется при *k=*50 по формуле для не обязательно симметричной матрицы;

3) *m=*50, λ1 берется при *k=*50 по формуле для симметричной матрицы.

Во всех случаях λ1 уже было вычислено ранее.

Для трех случаев: вывести на печать приближенное λ2, вычислить и вывести на печать собственный вектор *u*, соответствующий собственному значению λ2, вычислить вектор *Au*–λ2*u*, вычислить и вывести на печать кубическую норму этого вектора.

1. **Входные данные.**



1. **Листинг программы.**

#include <iostream>

#include <iomanip>

#include <algorithm>

#include <time.h>

using namespace std;

const int ITER = 52;

void func11(float\*\* matr,float\*\* u, float\* l1, float\* l2, float\* vector1,float\* vector2,float\* m1,float\* m2,float\* lambda22,float\*\* sobvector2,float\*\* vector22,float\* normi,int n){

float\*\* v = new float\*[ITER];

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

v[i] = new float[n];

}

u[0][0]=1;

for (int i = 1; i < n; ++i) {

u[0][i] = 0;

}

//нахождение u и v

for (int i = 1; i < ITER; ++i) {

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

float sum = 0;

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

sum += matr[j][k] \* u[i-1][k];

}

v[i][j] = sum;

}

float Max = 0;

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

Max = max(Max, v[i][j]);

}

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

u[i][j] = v[i][j] / Max;

}

}

//нахождение собственных значений первым способом, вектора «погрешности»

for (int i = 0; i < 5; ++i) {

int sign = 1;

if (u[46 + i][0] < 0)

sign = -1;

l1[i] = v[47 + i][0] \* sign;

if(i == 4)

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

vector1[j] = v[47 + i][j] - l1[i] \* u[46 + i][j];

}

//нахождение собственных значений вторым способом, вектора «погрешности»

for (int i = 0; i < 5; ++i) {

l2[i] = 0;

float sum=0;

for (int j = 0; j < 5; ++j)

l2[i]+= v[47 + i][j] \* u[46 + i][j];

for (int j = 0; j < 5; ++j)

sum += u[46 + i][j] \* u[46 + i][j];

l2[i] /= sum;

if (i == 4)

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

vector2[j] = v[47 + i][j] - l2[i] \* u[46 + i][j];

}

\*m1 = 0;

\*m2 = 0;

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

\*m1 = max(\*m1, vector1[i]);

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

\*m2 = max(\*m2, vector2[i]);

//вычисление вторых собственных значений, соответсвующих векторов и тд.

for (int i = 0; i < 3; ++i) {

int m=50;

if (i == 0)

m = 30;

int indx = 0;

float Max = 0;

float l = l1[4];

if (i == 2)

l = l2[4];

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

Max = max(v[m][j] - l \* u[m - 1][j], Max);

if (Max == v[m][j] - l \* u[m - 1][j])

indx = j;

}

Max = 0;

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

Max = max(Max, v[m][j]);

lambda22[i] = (v[m + 1][indx] \* Max - l \* v[m][indx]) / (v[m][indx] - l \* u[m - 1][indx]);

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

sobvector2[i][j] = v[51][j] - l\*u[50][j];

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

float sum = 0;

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

sum += matr[j][k] \* sobvector2[i][k];

}

vector22[i][j] = sum;

}

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

vector22[i][j] -= lambda22[i] \* sobvector2[i][j];

normi[i] = 0;

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

normi[i] = max(vector22[i][j], normi[i]);

}

}

int main(){

int n = 10;

float\*\* matr = new float\*[n];

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

matr[i] = new float[n];

srand(time(0));

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

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

if (i < j)

matr[i][j] = -(rand() % 5);//выбор элементов, не лежащих на главной диагонали

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

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

if (i > j)

matr[i][j] = matr[j][i];//выбор элементов, не лежащих на главной диагонали

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

float sum = 0;

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

if (j != i)

sum += matr[i][j];

if (i != 0)

matr[i][i] = -sum;//выбор элементов, лежащих на главной диагонали

}

float sum = 0;

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

sum += matr[0][i];

matr[0][0] = -sum + 1;

float\* lambda1 = new float[5];

float\* lambda2 = new float[5];

float\* lambda22 = new float[3];

float\* vector1 = new float[n];

float\* vector2 = new float[n];

float\*\* sobvector2 = new float\*[3];

for (int i = 0; i < 3; ++i)

sobvector2[i] = new float[n];

float max1 = 0;

float max2 = 0;

float\*\* vector22 = new float\*[3];

for (int i = 0; i < 3; ++i)

vector22[i] = new float[n];

float\* normi = new float[3];

float\*\* u = new float\*[ITER];

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

u[i] = new float[n];

func11(matr,u, lambda1, lambda2,vector1,vector2, &max1,&max2,lambda22,sobvector2,vector22,normi, n);

//Вывод данных

setlocale(LC\_ALL, ".1251");

cout << "Матрица:" << endl;

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

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

cout << setprecision(4) << setw(8) << matr[i][j];

cout << endl;

}

cout << endl;

cout << "Векторы u:" << endl;

for (int i = 46; i < 51; ++i) {

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

cout << setprecision(4) << setw(8) << u[i][j];

cout << endl;

}

cout << endl;

cout << "Наибольшие по модулю собственные значения в первом случае:" << endl;

for (int i = 0; i < 5; ++i) {

cout << setprecision(4) << setw(8) << lambda1[i];

}

cout << endl;

cout << "Наибольшие по модулю собственные значения во втором случае:" << endl;

for (int i = 0; i < 5; ++i) {

cout << setprecision(4) << setw(8) << lambda2[i];

}

cout << endl;

cout << "Вектор «погрешности» в первом случае:" << endl;

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

cout << setprecision(4) << setw(8) << vector1[i];

}

cout << endl;

cout << "Вектор «погрешности» во втором случае:" << endl;

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

cout << setprecision(4) << setw(8) << vector2[i];

}

cout << endl;

cout << "Кубическая норма вектора «погрешности» в первом случае: ";

cout << max1 << endl;

cout << "Кубическая норма вектора «погрешности» во втором случае: ";

cout << max2 << endl;

cout << "1) m=30, l1 берется при k=50 по формуле для не обязательно симметричной матрицы:" << endl;

cout << "l2: " << lambda22[0] << endl;

cout << "Собственный вектор u:" << endl;

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

cout << sobvector2[0][i] << ' ';

cout << endl;

cout << "Вектор Au–l2u:" << endl;

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

cout << vector22[0][i] << ' ';

cout << endl;

cout << "Кубическая норма Au–l2u: " << normi[0] << endl;

cout << "2) m=50, l1 берется при k=50 по формуле для не обязательно симметричной матрицы:" << endl;

cout << "l2: " << lambda22[1] << endl;

cout << "Собственный вектор u:" << endl;

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

cout << sobvector2[1][i] << ' ';

cout << endl;

cout << "Вектор Au–l2u:" << endl;

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

cout << vector22[1][i] << ' ';

cout << endl;

cout << "Кубическая норма Au–l2u: " << normi[1] << endl;

cout << "3) m=50, l1 берется при k=50 по формуле для симметричной матрицы:" << endl;

cout << "l2: " << lambda22[2] << endl;

cout << "Собственный вектор u:" << endl;

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

cout << sobvector2[2][i]<<' ';

cout << endl;

cout << "Вектор Au–l2u:" << endl;

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

cout << vector22[2][i] << ' ';

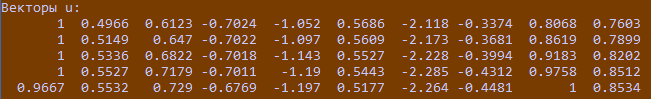
cout << endl;

cout << "Кубическая норма Au–l2u: " << normi[2] << endl;

return 0;

}

1. **Выходные данные**



Наибольшие по модулю собственные значения в первом случае:

28.81 28.83 28.85 28.86 27.92

Наибольшие по модулю собственные значения во втором случае:

29.5 29.53 29.56 29.59 29.61

Вектор <погрешности> в первом случае:

0.9295 1.086 1.723 -0.6208 -2.528 0.245 -3.805 -1.345 2.629 1.718

Вектор <погрешности> во втором случае:

-0.707 0.1494 0.4892 0.5251 -0.501 -0.6314 0.02823 -0.5865 0.9356 0.2732

Кубическая норма вектора <погрешности> в первом случае: 2.629

Кубическая норма вектора <погрешности> во втором случае: 0.9356

1) m=30, λ1 берется при k=50 по формуле для не обязательно симметричной матрицы:

l2: 29.25

Собственный вектор u:

0.9295 1.086 1.723 -0.6208 -2.528 0.245 -3.805 -1.345 2.629 1.718

Вектор Au- λ2u:

0.1947 0.7368 1.139 0.04333 -1.91 -0.2818 -2.154 -1.025 2.076 1.187

Кубическая норма Au-l2u: 2.076

2) m=50, λ1 берется при k=50 по формуле для не обязательно симметричной матрицы:

l2: 29.44

Собственный вектор u:

0.9295 1.086 1.723 -0.6208 -2.528 0.245 -3.805 -1.345 2.629 1.718

Вектор Au- λ2u:

0.01216 0.5236 0.8005 0.1652 -1.414 -0.3299 -1.406 -0.7605 1.56 0.8496

Кубическая норма Au- λ2u: 1.56

3) m=50, λ1 берется при k=50 по формуле для симметричной матрицы:

l2: 28.14

Собственный вектор u:

-0.707 0.1494 0.4892 0.5251 -0.501 -0.6314 0.02823 -0.5865 0.9356 0.2732

Вектор Au- λ2u:

0.005106 0.3056 0.4004 0.1586 -0.8737 -0.2361 -0.7613 -0.4042 0.9088 0.4968

Кубическая норма Au- λ2u: 0.9088

1. **Выводы**

В ходе лабораторной работы была разработана программа вычисления наибольшего и второго по величине модуля собственных значений и соответствующих им собственных векторов симметричной матрицы. При вычислении второго по величине модуля собственного значения, кубическая норма Au– λ2u с использованием формулы для не обязательно симметричной матрицы получилась меньше, чем норма с использованием формулы для симметричной матрицы.