**Министерство науки и высшего образования РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра ИС**

отчет

**по лабораторной работе №7**

**по дисциплине «Конструирование программ»**

Тема: Вычисление собственных значений (чисел) и векторов матриц

|  |  |  |
| --- | --- | --- |
| Студент гр. 8363 |  | Нерсисян А.С. |
| Преподаватель |  | Копыльцов А.В. |

Санкт-Петербург

2020

**Цель работы.**

Написать программу, которая определяет собственные значения и собственные вектора матрицы , затем находит максимальное по модулю собственное число и соответствующий ему собственный вектор с помощью подпрограмм 

**Основные теоретические положения.**

К настоящему времени имеется много различных алгоритмов решения проблемы собственных значений. Все они эффективны, но достаточно сложны и трудоемки. К наиболее простым методам нахождения собственных чисел относится степенной метод. К сожалению, им можно определить лишь два максимальных по модулю собственных значения.

Также довольно простой метод обратных итераций, очень похожий на степенной по алгоритму, позволяет находить собственный вектор, соответствующий заданному собственному числу. Это итеративный метод, он дает приближение к собственному вектору, сходящееся к нему по направлению. Следует заметить, что при вычислении собственных векторов имеется определенный произвол. Собственные векторы линейного оператора, отвечающие определенному собственному числу, образуют линейное пространство. Различные способы вычисления собственных векторов дают разные базисы этого пространства и, следовательно, разные координаты собственных векторов, удовлетворяющих, однако, основному определению собственного вектора .

В среде Mathcad собственные значения и векторы находятся функциями **eigenvals, eigenvec, eigenvecs, genvals** и **genvecs**. Рассмотрим их все по порядку.

Встроенная функция **eigenvals(A)** вычисляет вектор собственных значений матрицы , функция **genvals** – вектор обобщенных собственных значений  матрицы , удовлетворяющий условию



где - матрица с действительными элементами. Если задать , то результаты работы этих двух подпрограмм будут одинаковыми.

Введем с клавиатуры начало программы



Аналогично вычисляются и собственные вектора, соответствующие заданным собственным числам:



Эти вектора удовлетворяют основному определению  (см. ниже). Функции **eigenvecs** и **genvecs** также дают все нормированные собственные векторы заданной матрицы. Координаты собственных векторов расположены по столбцам результирующей матрицы, порядок расположения собственных векторов соответствует порядку собственных чисел, возвращаемых функциями **eigenvals** и **genvals**. Однако в данном случае собственные векторы относятся к иному базису линейного пространства и имеют другие координаты.









Противоположные знаки компонент векторов  в матрицах  объясняются противоположной направленностью собственных векторов, получаемых подпрограммами **eigenvecs** и **genvecs**.

**Экспериментальные результаты.**

**Задание № 1**

Определить собственные значения и собственные вектора матрицы  средствами пакета Mathcad, затем найти максимальное по модулю собственное число и соответствующий ему собственный вектор с помощью подпрограмм 

**Дано:** Вариант 11

**Обработка результатов эксперимента.**

**Задание № 1. решение:**

#include <iostream>

#include <conio.h>

using namespace std;

int main()

{

setlocale(LC\_ALL, "rus");

const double E = 0.0000001;

double A[4][4], x[4], prev, y[4], h[4], temp;

for (int i = 0; i < 4; ++i)

{

cout << "Строка №" << i << endl;

for (int j = 0; j < 4; ++j)

cin >> A[i][j];

x[i] = 0;

}

x[0] = 1;

for (int k = 0; k < 2; ++k)

{

do

{

for (int i = 0; i < 4; ++i)

{

y[i] = 0;

for (int j = 0; j < 4; ++j)

y[i] += A[i][j] \* x[j];

}

prev = h[k];

h[k] = 0;

for (int i = 0; i < 4; ++i)

h[k] += x[i] \* y[i];

temp = 0;

for (int i = 0; i < 4; ++i)

temp += y[i] \* y[i];

temp = sqrt(temp);

for (int i = 0; i < 4; ++i)

x[i] = y[i] / temp;

} while ((h[k] - prev) > 0.001);

if (k == 0) for (int i = 0; i < 4; ++i)

A[i][i] -= h[0];

else for (int i = 0; i < 4; ++i)

A[i][i] += h[0];

}

h[1] += h[0];

cout << h[0] << endl << h[1];

\_getch();

return 0;

}

**Выводы.**

В ходе выполнения данной лабораторной была написана программа, которая определяет собственные значения и собственные вектора матрицы , затем находит максимальное по модулю собственное число и соответствующий ему собственный вектор с помощью подпрограмм 