МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «Програмна інженерія та інформаційні технології управління»

Звіт з індивідуального розрахункового завдання №10

З предмету «Числові методи»

Виконав

Студент групи КН-36а

Рубан Ю.Д.

Перевірив:

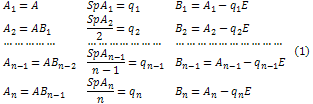
Гужва В.О.

Харків - 2017

Завдання: знайти власні значення та власні вектори матриці методом Левер'є-Фадеева.

Метод Фадєєва також відноситься до точних чисельних методів призначених для відшукання власних значень матриці і являється певною модифікацією [методу Левер'є](http://www.mathros.net.ua/znahodzhennja-vlasnyh-znachen-matryci-vykorystovujuchy-metod-leverje.html). Даний метод вважається більш ефективним, тому що крім спрощень при обчисленні коефіцієнтів характеристичного полінома він дозволяє визначити власні вектори та обернену матрицю до заданої.

Основна ідея методу Фадєєва полягає в тому, що замість послідовності Метод Федєєва, яку ми відшукували використовуючи алгоритм методу Левер'є, обчислюють послідовність Метод Федєєва, побудовану за наступними формулами:



де Метод Фадєєва — одинична матриця того ж самого порядку, що і матриця Метод Фадєєва; Метод Фадєєва сліди матриць Метод Федєєвавідповідно.

Після чого, з (1) коефіцієнти характеристичного многочлена, обернену матрицю та власні значеннавизначають за наступними формулами:

1. Метод Федєєва.
2. Якщо Метод Фадєєва — невироджена матриця, то Метод Федєєва.
3. Кожен стовпець матриці Метод Федєєва містить елементи власного вектора, який належить власному значенню Метод Федєєва.

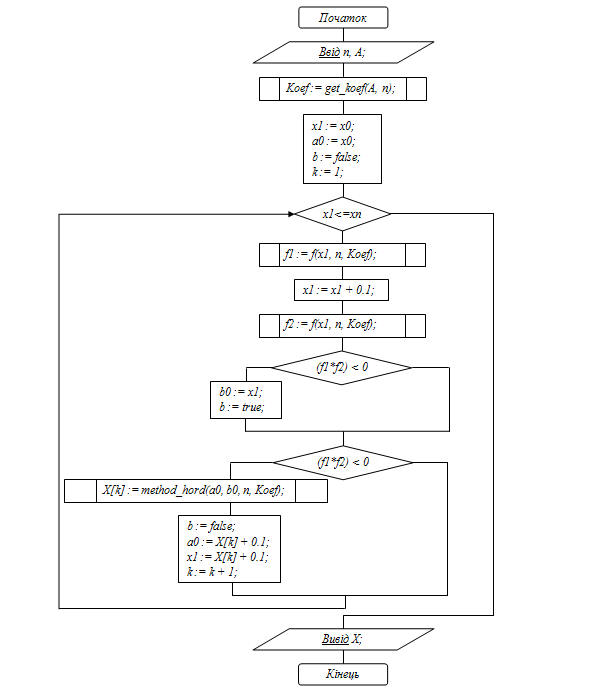


Рисунок 1 – Блок схема алгоритму пошуку власних значень і векторів методом Левер’є-Фадеева

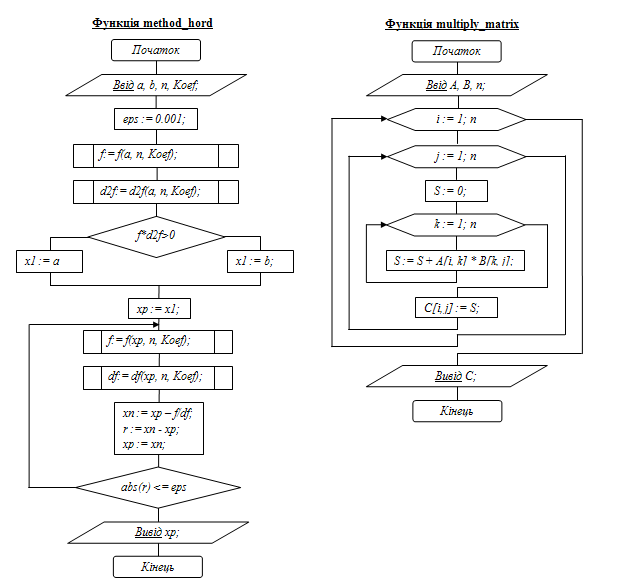


Рисунок 2 – Блок схеми алгоритмів хорд та перемноження матриць

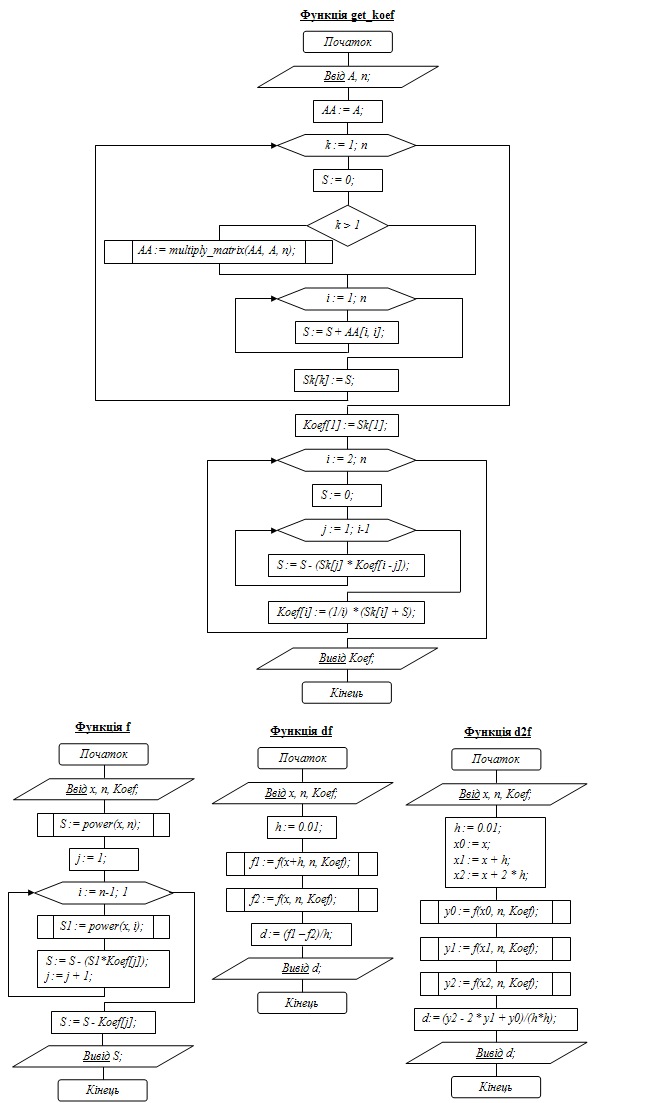


Рисунок 3 – Блок схеми додаткових функцій

Ручне рішення

2 3 2

4 -6 -4

-1 4 7

Начальный вектор

1

0

0

B1

-1 3 2

4 -9 -4

-1 4 4

A2

8 -13 0

-24 50 16

10 -11 10

q2 = SP(A2)/2 = 34

B2

-26 -13 0

-24 16 16

10 -11 -24

A3

-104 0 0

0 -104 0

0 0 -104

q3 = SP(A3)/3 = -104

Характерестический многочлен

(104\*x^0)+(-34\*x^1)+(-3\*x^2)+(1\*x^3)

Собственные значения

x1 = -5.8503

x2 = 3.08162

x3 = 5.76868

Коэфициенты Горнера

1 1 1

-8.8503 0.0816206 2.76868

17.7769 -33.7485 -18.0284

y0

2

4

-1

y1

14

-12

7

y2

6

100

-13

Собственные вектора

14.0763 -19.5852 1.50896

-47.4012 -11.6735 -0.925292

15.8503 6.91838 4.23132

Фрагмент коду програми

#include"Leverye\_Faddeev\_alg.h"

Result Leverye\_Faddeev\_alg::do\_algorithm(vector<vector<double>>matrix, int size, vector<vector<double>>\*arg)

{

vector<vector<double>>E(size);

vector<vector<double>>y(size);

vector<vector<double>>x(size);

vector<vector<double>>b(size - 1);

for (int i = 0; i < size - 1; i++)

{

b[i].resize(size);

}

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

{

y[i].resize(size);

x[i].resize(size);

E[i].resize(size);

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

{

if (i == j)E[i][j] = 1;

else E[i][j] = 0;

}

}

vector<vector<double>>temp = matrix;

vector<vector<double>>B = matrix;

vector<double>q(size);

q[0] = SP(matrix);

string bs = "B";

string as = "A";

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

{

B = matrix\_minus\_matrix(temp, matrix\_multi\_number(E, q[i - 1]));

show(B, bs+to\_string(i));

b[i-1] = B[0];

temp = matrix\_multi(matrix, B);

show(temp,as+to\_string(i+1));

q[i] = SP(temp) / (i + 1);

cout << "q" << to\_string(i + 1) << " = SP(A"+to\_string(i+1)+")/"<<(i+1)<<" = " <<q[i] << endl;

}

vector<double>rev = q;

for(int i =0,k=size-1;i<size;i++,k--)

{

rev[i] = -q[k];

}

rev.push\_back(1);

Polynomal P(rev);

cout << "Характерестический многочлен" << endl;

cout << P << endl;

vector<double>lambdas = \*Half\_div\_alg::solve(P.get\_function());

cout << "Собственные значения" << endl;

show\_row(lambdas);

Result res;

auto qq = gorner\_coefs(lambdas, q);

show(qq, "Коэфициенты Горнера");

vector<vector<double>>X(size);

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

{

X[i].resize(size);

}

vector<vector<double>>Y = \*arg;

vector<vector<double>>\*ys = new vector<vector<double>>[size];

ys[0] = matrix\_multi(matrix, Y);

Algorithms::show(ys[0], "y0");

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

{

cout << "y" << i;

ys[i] = matrix\_multi(matrix, ys[i - 1]);

Algorithms::show(ys[i]);

}

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

{

for (int i = size - 1, b = 0; i >= 0; i--, b++)

{

for (int j = 0, l = size - 1; j < size - 1; j++, l--)

{

X[k][b] += ys[l - 1][b][0] \* qq[j][k];

}

}

}

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

{

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

{

X[i][j] += qq[size - 1][i] \* Y[j][0];

}

}

transpose(X);

res = X;

return res;

}

vector<vector<double>>Leverye\_Faddeev\_alg::gorner\_coefs(vector<double>lambdas, vector <double>p)

{

int n = lambdas.size();

vector<vector<double>>q(n);

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

{

q[i].resize(n);

}

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

{

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

{

if (j == 0) { q[j][i] = 1; }

else

{

q[j][i] = (lambdas[i] \* q[j - 1][i]) - p[j - 1];

}

}

}

return q;

}

Результати виконання програми

n=3

Матриця =

2 3 2

4 -6 -4

-1 4 7

Початковий вектор =

1

0

0

Власні значення =

x1 = -5.8503

x2 = 3.08162

x3 = 5.76868

Власні вектори =

14.0763 -19.5852 1.50896

-47.4012 -11.6735 -0.925292

15.8503 6.91838 4.23132

Висновок

Результати програми співпадають з результатами ручного рішення

***Список використаних джерел***

**1. Островский А.М.** [Решение уравнений и систем уравнений](http://pmpu.ru/vf4/references#островский). М. ИЛ, 1963, c. 137-142

2. **Уилкинсон Дж.Х.** Алгебраическая проблема собственных значений. М.Наука. 1970, с.93-94

3. **Фаддеев Д.К., Фаддеева В.Н.** Вычислительные методы линейной алгебры. М.ГИФМЛ. 1960

4. **Хорн Р.**, **Джонсон Ч.** Матричный анализ. М.Мир.1989