МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «Програмна інженерія та інформаційні технології управління»

Звіт з індивідуального розрахункового завдання №8

З предмету «Числові методи»

Виконав

Студент групи КН-36а

Рубан Ю.Д.

Перевірив:

Гужва В.О.

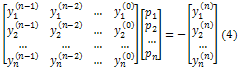
Харків - 2017

Завдання: знайти власні значення та власні вектори матриці методом Крилова.

Розглянемо метод призначений для знаходження власних значень матриці, алгоритм якого дещо відрізняється від [методу Данилевського](http://www.mathros.net.ua/znahodzhennja-vlasnyh-znachen-matryci-za-metodom-danylevskogo.html). Нехай Метод Крилова характеристичний многочлен матриці Метод Крилова. Виходячи з того, що всяка матриця перетворює в нуль свій характеристичний многочлен, будемо мати Метод Крилова.

Візьмемо тепер довільний ненульовий вектор Метод Крилова, розмірність якого співпадає з розмірністю матриці Метод Крилова і помножимо обидві частини рівності (1) з правої сторони на даний вектор, отримаємо: Метод Крилова.

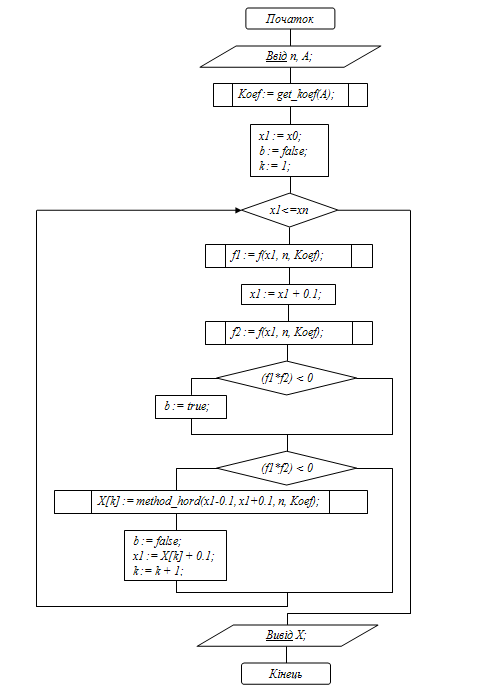
Поклавши Метод Крилова рівність (2) можна переписати в наступному вигляді: Метод Крилова, або

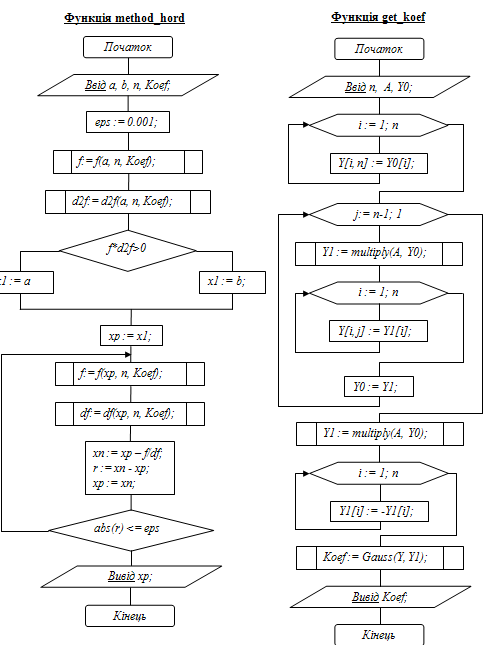


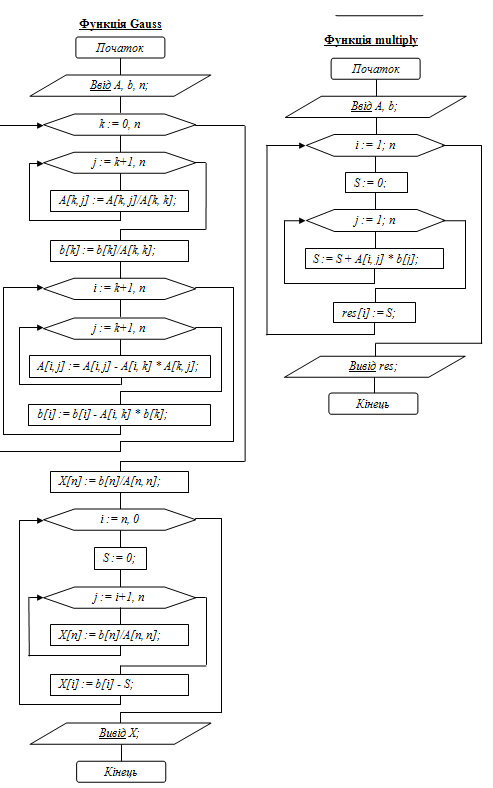
де координати векторів Метод Крилова визначаються за наступною формулою Метод Крилова. Тобто, ми отримуємо систему лінійних рівнянь Метод Крилова, розв'язавши яку, отримуємо коефіцієнти многочлена.

Якщо система (5) має єдиний розв'язок, то її корені Метод Крилова являються коефіцієнтами характеристичного многочлена. Даний розв'язок можна знайти будь-яким методом призначеним для знаходження рішення систем лінійних рівнянь ([метод Гаусса](http://www.mathros.net.ua/metod-gaussa-rozvjazok-systemy-linijnyh-rivnjan-metodom-gaussa.html), [метод простої ітерації](http://www.mathros.net.ua/nablyzhenyj-rozvjazok-systemy-linijnyh-rivnjan-metodom-prostoi-iteracii.html), [метод Зейделя](http://www.mathros.net.ua/nablyzhene-rozvjazannja-systemy-linijnyh-rivnjan-metodom-zejdelja.html) та інші). Якщо ж система (5) не має єдиного розв'язку, то в такому випадку рекомендується вибрати інший початковий вектор Метод Криловаі заново виконати зазначені дії.

## Блок-схема програмної реалізації методу Крилова для знаходження власних значень матриці:

****





Ручне рішення

2 3 2

4 -6 -4

-1 4 7

Початковий вектор

1

0

0

y0

2

4

-1

y1

14

-12

7

y2

6

100

-13

Прямой ход

14 2 1 -6

-12 4 0 -100

7 -1 0 13

Делим строку 1 на 14

Домножим строку 1 на 12 и прибавим эту строку к строке 2

Домножим строку 1 на -7 и прибавим эту строку к строке 3

1 0.142857 0.0714286 -0.428571

0 5.71429 0.857143 -105.143

0 -2 -0.5 16

Делим строку 2 на 5.71429

Домножим строку 2 на 2 и прибавим эту строку к строке 3

1 0.142857 0.0714286 -0.428571

0 1 0.15 -18.4

0 0 -0.2 -20.8

Делим строку 3 на -0.2

Треугольная матрица

1 0.142857 0.0714286 -0.428571

0 1 0.15 -18.4

0 0 1 104

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

Собственные вектора

14.0763 -19.5852 1.50896

-47.4012 -11.6735 -0.925292

15.8503 6.91838 4.23132

Фрагмент коду програми:

#include"Krulov\_alg.h"

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

{

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

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]);

}

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

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

{

res[i].resize(size + 1);

}

int k = size-1;

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

{

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

{

res[i][j] = ys[k-1][i][0];

}

k = size - 1;

}

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

{

res[i][size-1] = Y[i][0];

res[i][size] = -ys[size - 1][i][0];

}

Algorithms\*alg = new Gauss\_alg();

Result result;

vector<double>p = (alg->do\_algorithm(res, size, NULL));

vector<double>temp(p.size());

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

{

p[k] = -p[k];

temp[i] = -p[k];

}

temp.push\_back(1);

Polynomal P(temp);

cout << "Характерестический многочлен\n" << P << endl;

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

cout << "Собственные значения" << endl;

Algorithms::show\_row(lambdas,"x");

vector<vector<double>>q = gorner\_coefs(lambdas, p);

Algorithms::show(q, "Коэфициенты Горнера");

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

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

{

X[i].resize(size);

}

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] \* q[j][k];

}

}

}

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

{

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

{

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

}

}

transpose(X);

result = X;

return result;

}

vector<vector<double>>Krulov\_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