МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «Програмна інженерія та інформаційні технології управління»

Звіт з індивідуального розрахункового завдання №6

З предмету «Числові методи»

Виконав

Студент групи КН-36а

Рубан Ю.Д.

Перевірив:

Гужва В.О.

Харків - 2017

Завдання: розв'язати рівняння методом Гаусса-Зейделя.

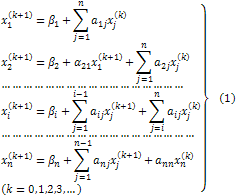
**Метод Зейделя** представляє собою модифікацію [**методу ітерації**](http://www.mathros.net.ua/nablyzhenyj-rozvjazok-systemy-linijnyh-rivnjan-metodom-prostoi-iteracii.html), основна ідея якого полягає в тому, що при обчисленні http://www.mathros.net.ua/wp-content/uploads/2012/07/metod_zejdela18.gif-го наближення невідомої http://www.mathros.net.ua/wp-content/uploads/2012/07/metod_zejdela19.gif, враховуються вже обчислені http://www.mathros.net.ua/wp-content/uploads/2012/07/metod_zejdela18.gif-ші наближення невідомих http://www.mathros.net.ua/wp-content/uploads/2012/07/metod_zejdela20.gif.

Розглянемо даний процес більш детально. Нехай дано**систему лінійних рівнянь** виду:

http://www.mathros.net.ua/wp-content/uploads/2012/07/metod_zejdela21.gif

Довільним чином, виберемо початкові наближення розв'язку http://www.mathros.net.ua/wp-content/uploads/2012/07/metod_zejdela22.gif, намагаючись, щоб вони якоюсь мірою відповідали шуканим невідомим http://www.mathros.net.ua/wp-content/uploads/2012/07/metod_zejdela23.gif.

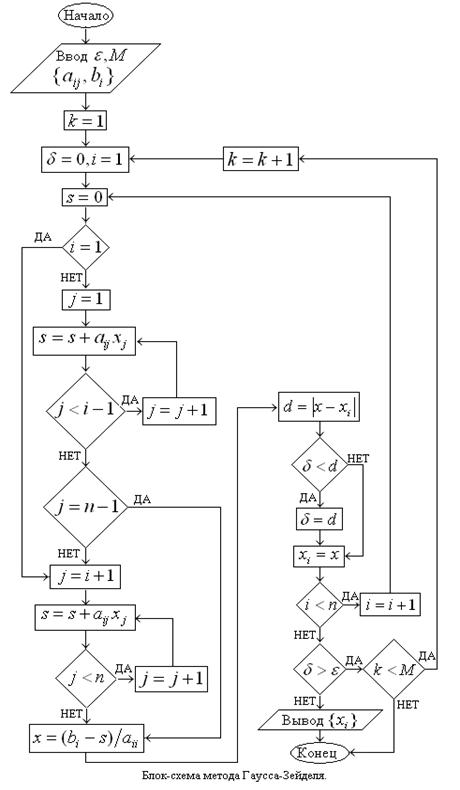
Далі, припускаючи, що http://www.mathros.net.ua/wp-content/uploads/2012/07/metod_zejdela24.gif-те наближення коренів http://www.mathros.net.ua/wp-content/uploads/2012/07/metod_zejdela25.gif відоме, відповідно до **методу Зейделя**, будемо шукати http://www.mathros.net.ua/wp-content/uploads/2012/07/metod_zejdela18.gif-е наближення за наступними формулами:



**Зауваження:** теорема відносно збіжності **методу простої ітерації** залишається актуальною і для ітерації **методом Зейделя**.

Одним з плюсів **методу Зейделя** є те, що він дає кращу збіжність ніж **метод простої ітерації**, а недоліком є більш громіздкий процес обчислень.  **Метод Зейделя** може збігатися навіть у тому випадку, коли процес ітерації розбіжний. Однак це буває не завжди. Бувають також випадки, коли процес Зейделя збігається повільніше від процесу простої ітерації. Більше того, бувають випадки, коли процес ітерації збігається, а **процес Зейделя** розбіжний.

Блок-схема алгоритму:



Ручне рішення

-0.88 -0.23 0.25 -0.16 -1.24

0.14 -0.66 -0.18 0.24 0.89

0.33 0.03 -0.54 -0.32 -1.15

0.12 -0.05 0 -0.85 0.57

Система з діагональною перевагою

-0.88 -0.23 0.25 -0.16 -1.24

0.14 -0.66 -0.18 0.24 0.89

-4.32 0.13 5.97 1.36 9.37

0.12 -0.05 0 -0.85 0.57

Выразим x1 в 1 строке

Поделим строку 1 на 0.88 (d1 на -0.88)

Выразим x2 в 2 строке

Поделим строку 2 на 0.66 (d2 на -0.66)

Выразим x3 в 3 строке

Поделим строку 3 на -5.97 (d3 на 5.97)

Выразим x4 в 4 строке

Поделим строку 4 на 0.85 (d4 на -0.85)

1 -0.261364 0.284091 -0.181818 1.40909

0.212121 1 -0.272727 0.363636 -1.34848

0.723618 -0.0217755 1 -0.227806 1.56951

0.141176 -0.0588235 0 1 -0.670588

Итерация = 1

x(1)1 = -0.261364 \* -1.34848 + 0.284091 \* 1.56951 + -0.181818 \* -0.670588 + 1.40909 = 2.32935

x(1)2 = 0.212121 \* 2.32935 + -0.272727 \* 1.56951 + 0.363636 \* -0.670588 + -1.34848 = -1.52628

x(1)3 = 0.723618 \* 2.32935 + -0.0217755 \* -1.52628 + -0.227806 \* -0.670588 + 1.56951 = 3.44107

x(1)4 = 0.141176 \* 2.32935 + -0.0588235 \* -1.52628 + 0 \* 3.44107 + -0.670588 = -0.251958

Итерация = 2

x(2)1 = -0.261364 \* -1.52628 + 0.284091 \* 3.44107 + -0.181818 \* -0.251958 + 1.40909 = 2.83139

x(2)2 = 0.212121 \* 2.83139 + -0.272727 \* 3.44107 + 0.363636 \* -0.251958 + -1.34848 = -1.77798

x(2)3 = 0.723618 \* 2.83139 + -0.0217755 \* -1.77798 + -0.227806 \* -0.251958 + 1.56951 = 3.71448

x(2)4 = 0.141176 \* 2.83139 + -0.0588235 \* -1.77798 + 0 \* 3.71448 + -0.670588 = -0.166275

Итерация = 3

x(3)1 = -0.261364 \* -1.77798 + 0.284091 \* 3.71448 + -0.181818 \* -0.166275 + 1.40909 = 2.95927

x(3)2 = 0.212121 \* 2.95927 + -0.272727 \* 3.71448 + 0.363636 \* -0.166275 + -1.34848 = -1.79426

x(3)3 = 0.723618 \* 2.95927 + -0.0217755 \* -1.79426 + -0.227806 \* -0.166275 + 1.56951 = 3.78785

x(3)4 = 0.141176 \* 2.95927 + -0.0588235 \* -1.79426 + 0 \* 3.78785 + -0.670588 = -0.147264

Итерация = 4

x(4)1 = -0.261364 \* -1.79426 + 0.284091 \* 3.78785 + -0.181818 \* -0.147264 + 1.40909 = 2.98091

x(4)2 = 0.212121 \* 2.98091 + -0.272727 \* 3.78785 + 0.363636 \* -0.147264 + -1.34848 = -1.80277

x(4)3 = 0.723618 \* 2.98091 + -0.0217755 \* -1.80277 + -0.227806 \* -0.147264 + 1.56951 = 3.79936

x(4)4 = 0.141176 \* 2.98091 + -0.0588235 \* -1.80277 + 0 \* 3.79936 + -0.670588 = -0.143708

Итерация = 5

x(5)1 = -0.261364 \* -1.80277 + 0.284091 \* 3.79936 + -0.181818 \* -0.143708 + 1.40909 = 2.98576

x(5)2 = 0.212121 \* 2.98576 + -0.272727 \* 3.79936 + 0.363636 \* -0.143708 + -1.34848 = -1.80359

x(5)3 = 0.723618 \* 2.98576 + -0.0217755 \* -1.80359 + -0.227806 \* -0.143708 + 1.56951 = 3.80208

x(5)4 = 0.141176 \* 2.98576 + -0.0588235 \* -1.80359 + 0 \* 3.80208 + -0.670588 = -0.142975

|x(k+1)i - x(k)i|< 0.001

x1 = 2.98576

x2 = -1.80359

x3 = 3.80208

x4 = -0.142975

Фрагмент коду програми:

#include"Gauss\_Seidel\_alg.h"

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

{

express\_x(matrix, size);

vector<double>d(size);

vector<double>d\_prev(size);

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

{

d[i] = matrix[i][size];

d\_prev[i] = matrix[i][size];

}

show(matrix);

double eps = 0.001;

bool flag[4] = { true,true,true,true };

bool iterate = true;

int iteration = 1;

do

{

cout << "Итерация = " << iteration << endl;

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

{

d[i] = 0;

cout << "x(" << iteration << ")" << i + 1 << " = ";

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

{

if (j == i)continue;

if (j == size)

{

d[i] += matrix[i][j];

cout << matrix[i][j] << " = " << d[i] << endl;

}

else

{

d[i] += matrix[i][j]\*d[j];

cout << matrix[i][j] << " \* " << d[j] << " + ";

}

}

}

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

{

if (abs(d[i] - d\_prev[i]) < eps)

{

flag[i] = false;

}

else

{

flag[i] = true;

}

d\_prev[i] = d[i];

}

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

{

if (flag[i] == false) { iterate = false; break; }

else { iterate = true; }

}

iteration++;

}while (iterate);

cout << "|x(k+1)i - x(k)i|< " << eps << endl;

Result res;

res = d;

return res;

}

Результати виконання програми

n=4

Матриця =

0.14 -0.66 -0.18 0.24 0.89

-4.32 0.13 5.97 1.36 9.37

0.12 -0.05 0 -0.85 0.57

x1 = 2.98658

x2 = -1.80388

x3 = 3.80226

x4 = -0.142897

Висновок:

Результати програми співпадають з ручним рішенням

***Список використаних джерел***

1. Щуп Т. Решение инженерных задач на ЕВМ. – М.: Мир, 1982. – 235с.

2. Демидович Б. П., Марон И. А. Основы вычислительной математики. – М.: Наука, 1970. – 664 с.

3. Демидович Б. П., Марон И. А., Шувалова Е. З. Численные методы анализа. – М.: Мир, 1967Волков Е. А. Численные методы. – М.: Наука, 1988.