Метод Гаусса выбора

Задача: решить систему линейных уравнений методом Гаусса выбора.

Решение поиск главного элемента по строкам.

Код программы:

#include <stdio.h>

#include <math.h>

#define \_i 4

#define \_j 5

void solve(float \*a, float \*b){

    //создание одной матрицы

    float array[\_i][\_j];

    int count\_a = 0, count\_b = 0;

    for(int i = 0; i < \_i; i++){

        for(int j = 0; j < \_j-1; j++){

            array[i][j] = a[count\_a];

            count\_a++;

        }

        array[i][\_j-1]=b[count\_b];

        count\_b++;

    }

    //решение задачи

    float a\_max, v, max, tmp;

    int max\_i;

    for(int i = 0; i < \_i-1; i++)

    {

        //Выбор максимального элемента и замена строк

        max = array[i][i];

        for (int m = i; m < \_i; m++)

        {

            if(fabs(max) < fabs(array[m][i]))

            {

                max = array[m][i];

                max\_i = m;

            }

        }

        for (int r = i; r <\_j; r++){

            tmp = array[i][r];

            array[i][r]= array[max\_i][r];

            array[max\_i][r] = tmp;

        }

        //прямой ход

        a\_max = array[i][i];

        for(int j = i; j < \_j; j++)

        {

            array[i][j] = array[i][j] / a\_max;

        }

        for(int k = i + 1; k <\_i; k++)

        {

            v = array[k][i];

            for(int j = 0; j < \_j; j++)

            {

                array[k][j] -= v \* array[i][j];

            }

        }

    }

    // v -> копит сумму

    v = 0;

    for(int i = \_i-1; i >= 0; i--)

    {

        for(int j = \_j-2; j > i; j--)

        {

            v += array[i][j]\*array[j][j];

        }

        array[i][i] = (array[i][\_j-1]-v)/array[i][i];

        v = 0;

        printf("x%d = %f\n", i+1, array[i][i]);

    }

    //вывод матрицы

    for (int i=0; i<\_i; i++){

        for (int j=0; j<\_j; j++)

            printf("%f ", array[i][j]);

        printf("\n");

    }

}

int main(){

    float a[\_i][\_j-1]={{5, 7, 6, 5},

                 {7, 10, 8, 7},

                 {6, 8, 10, 9},

                 {5, 7, 9, 10}};

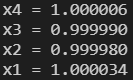
    float b[\_i][1]={{23}, {32}, {33}, {31}};

    solve(&a[0][0], &b[0][0]);

    return 0;

}

Результат работы программы:



Решение поиск главного элемента по столбцам.

Код программы:

#include <stdio.h>

#include <math.h>

#define \_i 4

#define \_j 5

void solve(float \*a, float \*b){

    //создание одной матрицы

    float array[\_i][\_j];

    int count\_a = 0, count\_b = 0;

    for(int i = 0; i < \_i; i++){

        for(int j = 0; j < \_j-1; j++){

            array[i][j] = a[count\_a];

            count\_a++;

        }

        array[i][\_j-1]=b[count\_b];

        count\_b++;

    }

    //решение задачи

    float a\_max, v, max, tmp;

    int max\_j;

    //Выбор максимального элемента и замена столбцов

    for (int j = 0; j < \_j-2; j++)

    {

        max\_j = j;

        max = array[0][max\_j];

        for(int s = j + 1; s<\_j-2;s++)

        {

            if (fabs(max) < fabs(array[0][s]))

            {

                max\_j = s;

                max = array[0][max\_j];

            }

        }

        for (int i = 0; i < \_i; i++)

        {

            tmp = array[i][j];

            array[i][j] = array[i][max\_j];

            array[i][max\_j] = tmp;

        }

    }

    for(int i = 0; i < \_i-1; i++)

    {

        //прямой ход

        a\_max = array[i][i];

        for(int j = i; j < \_j; j++)

        {

            array[i][j] = array[i][j] / a\_max;

        }

        for(int k = i + 1; k <\_i; k++)

        {

            v = array[k][i];

            for(int j = 0; j < \_j; j++)

            {

                array[k][j] -= v \* array[i][j];

            }

        }

    }

    // v -> копит сумму

    v = 0;

    for(int i = \_i-1; i >= 0; i--)

    {

        for(int j = \_j-2; j > i; j--)

        {

            v += array[i][j]\*array[j][j];

        }

        array[i][i] = (array[i][\_j-1]-v)/array[i][i];

        v = 0;

        printf("x%d = %f\n", i+1, array[i][i]);

    }

    //вывод матрицы

    for (int i=0; i<\_i; i++){

        for (int j=0; j<\_j; j++)

            printf("%f ", array[i][j]);

        printf("\n");

    }

}

int main(){

    float a[\_i][\_j-1]={{5, 7, 6, 5},

                 {7, 10, 8, 7},

                 {6, 8, 10, 9},

                 {5, 7, 9, 10}};

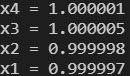
    float b[\_i][1]={{23}, {32}, {33}, {31}};

    solve(&a[0][0], &b[0][0]);

    return 0;

}

Результат работы программы:



Вывод: программа решила систему линейных уравнений и выдала приближенные к реальным значениям корни.