

НИЯУ МИФИ

Отчет по курсу СГМ  
«Решения СЛАУ методом Гаусса»

Выполнил: студент группы Б17-511 Чудновец Иван

Проверил: Козин Рудольф Глебович

2019

# Скриншот работы программы:

```
D:\Боты\Создание почт\Уроки\Универ\2 курс (ПМИ)\4 семестр\Численные методы\Метод Гау...
*** Программа находит решение системы  $Ax=b$  методом Гаусса ***
Введите размерность системы: 4
Введите 0/1 - задать систему случайным образом или вручную 1
Введите A(0, 0): 2
Введите A(0, 1): 1
Введите A(0, 2): 4
Введите A(0, 3): 8
Введите b(0): -1
Введите A(1, 0): 1
Введите A(1, 1): 3
Введите A(1, 2): -6
Введите A(1, 3): 2
Введите b(1): 3
Введите A(2, 0): 3
Введите A(2, 1): -2
Введите A(2, 2): 2
Введите A(2, 3): -2
Введите b(2): 8
Введите A(3, 0): 2
Введите A(3, 1): -1
Введите A(3, 2): 2
Введите A(3, 3): 0
Введите b(3): 4
Введенная система:
2,00000 1,00000 4,00000 8,00000 -1,00000
1,00000 3,00000 -6,00000 2,00000 3,00000
3,00000 -2,00000 2,00000 -2,00000 8,00000
2,00000 -1,00000 2,00000 0,00000 4,00000
Система преобразована к виду:
1,00000 -0,66667 0,66667 -0,66667 2,66667
0,00000 1,00000 -1,81818 0,72727 0,09091
0,00000 0,00000 1,00000 1,10526 -0,94737
0,00000 0,00000 0,00000 1,00000 0,50000
Определитель матрицы системы = 23,99999
Решение системы:
2,00000 -3,00000 -1,50000 0,50000
Число переставленных строк - 1
Невязка для полученного решения:
-0,00000 0,00000 0,00000 0,00000
Для завершения программы введите любой символ и ENTER_
```

# Код программы:

```
#include <iostream>
#include <cstdlib>
#include <windows.h>
#include <math.h>
#include <time.h>
#define B(i,j) MM[i*(n+1)+j] //исходная система
#define A(i,j) M[i*(n+1)+j] //приводимая система
using namespace std;
float *M, *MM, *sol, det;
int n, m; //размерность и число перестановок
int n1, n2; //вспомогательные переменные
void metod_G();
void select_Amax(int k);
void b_Ax();
int main()
{
    int i, j, c;
    char s[2];
    setlocale(LC_ALL, "");
    cout << "**** Программа находит решение системы Ax=b методом Гаусса ****";
    cout << "\nВведите размерность системы: ";
    cin >> n; n1 = n - 1; n2 = n + 1;
    M = new float[n*(n + 1)]; //расширенная матрица - (A+b)
    MM = new float[n*(n + 1)];
    cout << "Введите 0/1 - задать систему случайным образом или вручную ";
    cin >> c;
    switch (c) {
        case 0:
            //инициализация датчика п.с. чисел текущим временем
            srand(time(NULL));
            for (i = 0; i < n; i++)for (j = 0; j < n + 1; j++)B(i, j) = 0.5 - rand() / (RAND_MAX +
1.0); //[-0.5;0.5]
            break;
        default:
            for (i = 0; i < n; i++) {
                for (j = 0; j < n; j++) {
                    cout << "Введите A(" << i << ", " << j << "): ";
                    cin >> B(i, j);
                }
                cout << "Введите b(" << i << "): ";
                cin >> B(i, n);
            }
    }
    cout << "Введенная система:\n";
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            printf("%.5f ", B(i, j));
            A(i, j) = B(i, j);
        }
        printf("%.5f\n", B(i, n));
        A(i, n) = B(i, n);
    }
    metod_G();
    printf("Определитель матрицы системы = %.5f", det);
    if (fabs(det) < 1.E-10) {
        printf("\nМатрица системы ВЫРОЖДЕННАЯ !");
    }
    else {
        printf("\nРешение системы:\n");
        sol = new float[n];
        for (i = n - 1; i >= 0; i--) {
            sol[i] = A(i, n);
            for (j = n - 1; j > i; j--) {
                sol[i] -= A(i, j) * sol[j];
            }
        }
    }
}
```

```

    }
    for (i = 0; i < n; i++) {
        printf("%.5f ", sol[i]);
    }
    printf("\nЧисло переставленных строк - %d", m);
    b_Ax();
    printf("\nНевязка для полученного решения:\n");
    for (i = 0; i < n; i++) {
        printf("%.5f ", B(i, n));
    }
}
printf("\nДля завершения программы введите любой символ и ENTER");
cin >> s;
return 0;
}
//п/п вычисляет невязку для полученного решения
void b_Ax() {
    int i, j;
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) B(i, n) = B(i, n) - B(i, j) * sol[j];
    }
}
//п/п реализует метод Гаусса
void metod_G() {
    int i, j, k, t;
    det = 1; m = 0;
    //приводим матрицу к диагональному виду
    for (k = 0; k < n; k++) //цикл по преобразованиям
    {
        select_Amax(k);
        if (fabs(det) < 1.E-10) { return; }
        for (j = n; j > k - 1; j--) {
            A(k, j) = A(k, j) / A(k, k);
        }
        /*
        buf=A(k,k); //можно и так
        for (j=k;j<n+1;j++) A(k,j)=A(k,j)/buf;
        */
        for (i = k + 1; i < n; i++) {
            for (j = n; j > k - 1; j--) {
                A(i, j) = A(i, j) - A(i, k)*A(k, j);
            }
        }
    }
    printf("Система преобразована к виду; \n");
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            printf("%.5f ", A(i, j));
        }
        printf("%.5f\n", A(i, n));
    }
}
//процедура выбора ведущего элемента в k-ом подстолбце
void select_Amax(int k) {
    int i, kmax, j, k1 = k + 1;
    float Amod, Amax, buf;
    Amax = fabs(A(k, k)); kmax = k;
    if (k < n - 1) {
        for (i = k1; i < n; i++) {
            Amod = fabs(A(i, k));
            if (Amod > Amax) { Amax = Amod; kmax = i; }
        }
    }
    if (Amax < 1.E-10) { //матрица системы вырожденная
        det = 0; return;
    }
    if (kmax != k) {
        for (j = k; j < n + 1; j++) { //переставлем строки
            buf = A(k, j);

```

```

        A(k, j) = A(kmax, j);
        A(kmax, j) = buf;
    }
    //printf("ТЕСТ => A(k,k), A(k,n) = %.3f %.3f\n",A(k,k),A(k,n));
    det = -det;
    m++; //число перестановок строк
}
det = det * A(k, k);
}

```