Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей
Кафедра информатики
Дисциплина: Методы численного анализа

ОТЧЁТ

к лабораторной работе
на тему

Решение систем линейных алгебраических уравнений (СЛАУ)
методом Гаусса и с помощью его модификаций

Выполнил: студент группы 053504
Пекутько Дмитрий Леонидович
Проверил: Анисимов Владимир Яковлевич

Минск 2021

Оглавление

Вариант 5

Цели выполнения задания

- изучить метод Гаусса и его модификации, составить алгоритм метода и программу его реализации, получить численное решение заданной СЛАУ;
- составить алгоритм решения СЛАУ указанными методами, применимый для организации вычислений на ЭВМ;
- составить программу решения СЛАУ по разработанному алгоритму;
- выполнить тестовые примеры и проверить правильность работы программы.

Краткие теоретические сведения

СЛАУ обычно записывается в виде

$$\sum_{j=1}^{n} a_{ij} x_j = b_i ; i \le 1 \le n \quad , \text{ или коротко } \quad \boldsymbol{Ax} = \boldsymbol{b} \, , \qquad (1.1)$$

где

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \quad ; \quad \boldsymbol{a} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} \quad ; \quad \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix} .$$

Здесь $\boldsymbol{A}$ и $\boldsymbol{b}$ заданы, требуется найти $\boldsymbol{x}$.

Метод Гаусса прекрасно подходит для решения систем линейных алгебраических уравнений (СЛАУ). Он обладает рядом преимуществ по сравнению с другими методами:

- во-первых, нет необходимости предварительно исследовать систему уравнений на совместность;
- во-вторых, методом Гаусса можно решать не только СЛАУ, в которых число уравнений совпадает с количеством неизвестных переменных и основная матрица системы невырожденная, но и системы уравнений, в которых число уравнений не совпадает с количеством неизвестных переменных или определитель основной матрицы равен нулю;
- в-третьих, метод Гаусса приводит к результату при сравнительно небольшом количестве вычислительных операций.

Метод Гаусса включает в себя прямой (приведение расширенной матрицы к ступенчатому виду, то есть получение нулей под главной диагональю) и обратный (получение нулей над главной диагональю расширенной матрицы) ходы. Прямой ход и называется методом Гаусса, обратный - методом Гаусса-Жордана, который отличается от первого только последовательностью исключения переменных.

Метод Гаусса идеально подходит для решения систем содержащих больше трех линейных уравнений, для решения систем уравнений, которые не являются квадратными (чего не скажешь про метод Крамера и матричный метод). То есть метод Гаусса - наиболее универсальный метод для нахождения решения любой системы линейных уравнений, он работает в случае, когда система имеет бесконечно много решений или несовместна.

Прямой ход состоит из $n - 1$ шагов исключения.

1-й шаг. Целью этого шага является исключение неизвестного x_1 из уравнений с номерами $i = 2, 3, \ldots, n$. Предположим, что коэффициент $a_{11} \neq 0$. Будем называть его *главным элементом* 1-*го шага*.

Найдем величины

$$q_{i1} = a_{i1}/a_{11} \quad (i = 2, 3, \ldots, n),$$

называемые *множителями* 1-*го шага*. Вычтем последовательно из второго, третьего, …, n-го уравнений системы первое уравнение, умноженное соответственно на q_{21}, q_{31}, …, q_{n1}. Это позволит обратить в нуль коэффициенты при x_1 во всех уравнениях, кроме первого. В результате получим эквивалентную систему

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n = b_1,$$

$$a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + \ldots + a_{2n}^{(1)}x_n = b_2^{(1)},$$

$$a_{32}^{(1)}x_2 + a_{33}^{(1)}x_3 + \ldots + a_{3n}^{(1)}x_n = b_3^{(1)},$$

$$\cdots\cdots\cdots\cdots\cdots$$

$$a_{n2}^{(1)}x_2 + a_{n3}^{(1)}x_3 + \ldots + a_{nn}^{(1)}x_n = b_n^{(1)},$$

в которой $a_{ij}^{(1)}$ и $b_{ij}^{(1)}$ вычисляются по формулам

$$a_{ij}^{(1)} = a_{ij} - q_{i1}a_{1j} \quad , \quad b_i^{(1)} = b_i - q_{i1}b_1.$$

2-й шаг. Целью этого шага является иcключение неизвестного x_2 из уравнений с номерами $i = 3, 4, \ldots, n$. Пусть $a_{22}^{(1)} \neq 0$, где $a_{22}^{(1)}$ – коэффициент, называемый *главным* (или *ведущим*) *элементом* 2-*го шага*. Вычислим множители 2-го шага и вычтем последовательно из третьего, четвертого, …, n-го уравнений системы второе уравнение, умноженное соответственно на q_{32}, q_{42}, …, q_{n2}. В результате получим систему

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n &= b_1 \quad ,\\
a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + \ldots + a_{2n}^{(1)}x_n &= b_2^{(1)} \quad ,\\
a_{33}^{(2)}x_3 + \ldots + a_{3n}^{(2)}x_n &= b_3^{(2)} \quad ,\\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
a_{n3}^{(2)}x_3 + \ldots + a_{nn}^{(2)}x_n &= b_n^{(2)} \quad .
\end{aligned}
$$

Здесь коэффициенты $a_{ij}^{(2)}$ и $b_j^{(2)}$ вычисляются по формулам

$$a_{ij}^{(2)} = a_{ij}^{(1)} - q_{i2}a_{2j}^{(1)} \quad , \quad b_i^{(2)} = b_i^{(1)} - q_{i2}b_2^{(1)}.$$

Аналогично проводятся остальные шаги. Опишем очередной k-й шаг.

k-й шаг. В предположении, что *главный* (*ведущий*) *элемент* k-го шага $a_{kk}^{(k-1)}$ отличен от нуля, вычислим *множители k-го шага*

$$q_{ik} = a_{ik}^{(k-1)} / a_{kk}^{(k-1)} \quad (i = k + 1, \ldots, n)$$

и вычтем последовательно из $(k + 1)$-го, …, n-го уравнений полученной на предыдущем шаге системы k-е уравнение, умноженное соответственно на $q_{k+1,k}$, $q_{k+2,k}$, …, q_{nk}.

После $(n - 1)$-го шага исключения получим систему уравнений

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n &= b_1 \quad ,\\
a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + \ldots + a_{2n}^{(1)}x_n &= b_2^{(1)} \quad ,
\end{aligned}
$$

$$a_{33}^{(2)}x_3 + \ldots + a_{3n}^{(2)}x_n = b_3^{(2)} ,$$

. . . .

$$a_{nn}^{(n-1)}x_n = b_n^{(n-1)} ,$$

матрица $\boldsymbol{A}^{(n-1)}$ которой является верхней треугольной. На этом вычисления прямого хода заканчиваются.

Обратный ход. Из последнего уравнения системы находим x_n. Подставляя найденное значение x_n в предпоследнее уравнение, получим x_{n-1}. Осуществляя обратную подстановку, далее последовательно находим x_{n-2}, …, x_1. Вычисления неизвестных здесь проводятся по формулам

$$x_n = b_n^{(n-1)} / a_{nn}^{(n-1)},$$

$$x_k = (b_n^{(k-1)} - a_{k,k+1}^{(k-1)}x_{k+1} - \ldots - a_{kn}^{(k-1)}x_n) / a_{kk}^{(k-1)}, (k = n-1, \ldots, 1).$$

Необходимость отличия от 0 главных элементов. Заметим, что вычисление множителей, а также обратная подстановка требуют деления на главные элементы $a_{kk}^{(k-1)}$. Поэтому если один из главных элементов оказывается равным нулю, то схема не может быть реализована. Здравый смысл подсказывает, что и в ситуации, когда все главные элементы отличны от нуля, но среди них есть близкие к нулю, возможен неконтролируемый рост погрешности.

Метод Гаусса с выбором главного элемента по столбцу (схема частичного выбора). На k-м шаге прямого хода коэффициенты уравнений системы с номерами $i = k + 1, \ldots, n$ преобразуются по формулам

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - q_{ik}a_{kj}, \; b_i^{(k)} = b_i^{(k-1)} - q_{ik}b_k^{(k-1)}, \; i = k+1, \ldots, n.$$

Интуитивно ясно, что во избежание сильного роста коэффициентов системы и связанных с этим ошибок нельзя допускать появления больших множителей q_{ik}.

В методе Гаусса с выбором главного элемента по столбцу гарантируется, что $|q_{ik}| \le 1$ для всех $k = 1, 2, \ldots, n-1$ и $i = k+1, \ldots, n$. Отличие этого варианта метода Гаусса от схемы единственного деления заключается в том, что на k-м шаге исключения в качестве главного элемента выбирают максимальный по модулю коэффициент $a_{i_k k}$ при неизвестной x_k в уравнениях с номерами $i = k + 1, \dots, n$. Затем соответствующее выбранному коэффициенту уравнение с номером i_k меняют местами с k-м уравнением системы для того, чтобы главный элемент занял место коэффициента $a_{kk}^{(k-1)}$. После этой перестановки исключение неизвестного x_k производят, как в схеме единственного деления.

Метод Гаусса с выбором главного элемента по всей матрице (схема полного выбора). В этой схеме допускается нарушение естественного порядка исключения неизвестных. На 1-м шаге метода среди элементов a_{ij} определяют максимальный по модулю элемент $a_{i_1 j_1}$. Первое уравнение системы и уравнение с номером i_1 меняют местами. Далее стандартным образом производят исключение неизвестного x_{i_1} из всех уравнений, кроме первого.

На k-м шаге метода среди коэффициентов $a_{ij}^{(k-1)}$ при неизвестных в уравнениях системы с номерами $i = k, \dots, n$ выбирают максимальный по модулю коэффициент $a_{i_k j_k}^{(k-1)}$. Затем k-е уравнение и уравнение, содержащее найденный коэффициент, меняют местами и исключают неизвестное x_{j_k} из уравнений с номерами $i = k + 1, \dots, n$.

На этапе обратного хода неизвестные вычисляют в следующем порядке: $x_{j_n}, x_{j_{n-1}}, \dots, x_{j_1}$.

Задание

Методом Гаусса и методом выбора главного элемента найти с точностью 0,0001 численное решение системы ***Ax=b***,

где A = kC + D, A – исходная матрица для расчёта, k – номер варианта (0-15), матрицы C, D и вектор свободных членов b задаются ниже.

Вариант 13

Исходные данные:

Вектор **b**= $(4{,}2;\ 4{,}2;\ 4{,}2\ ;\ 4{,}2;\ 4{,}2)^{T}$,

$$C=\begin{bmatrix}0{,}2 & 0 & 0{,}2 & 0 & 0\\ 0 & 0{,}2 & 0 & 0{,}2 & 0\\ 0{,}2 & 0 & 0{,}2 & 0 & 0{,}2\\ 0 & 0{,}2 & 0 & 0{,}2 & 0\\ 0 & 0 & 0{,}2 & 0 & 0{,}2\end{bmatrix},\quad D=\begin{bmatrix}2{,}33 & 0{,}81 & 0{,}67 & 0{,}92 & -0{,}53\\ -0{,}53 & 2{,}33 & 0{,}81 & 0{,}67 & 0{,}92\\ 0{,}92 & -0{,}53 & 2{,}33 & 0{,}81 & 0{,}67\\ 0{,}67 & 0{,}92 & -0{,}53 & 2{,}33 & 0{,}81\\ 0{,}81 & 0{,}67 & 0{,}92 & -0{,}53 & 2{,}33\end{bmatrix}.$$

Программная реализация

```
#include <iostream>

#include <iomanip>

#include <cmath>

#include <string>


using namespace std;


const int size = 5;


double A[size][size];

double C [size][size] = {

   {0.2, 0.0, 0.2, 0.0, 0.2},

   {0.0, 0.2, 0.0, 0.2, 0.0},

   {0.2, 0.0, 0.2, .0, 0.2},

   {0.0, 0.2, 0.0, 0.2, 0.0},

   {0.0, 0.0, 0.2, 0.0, 0.2},

};

double D [size][size] = {

   {2.33, 0.81, 0.67, 0.92, -0.53},

   {-0.53, 2.33, 0.81, 0.67, 0.92},

   {0.92, -0.53, 2.33, 0.81, 0.67},

   {0.67, 0.92, -0.53, 2.33, 0.81},

   {0.81, 0.67, 0.92, -0.53, 2.33},

};
```

```
double B [size] = { 4.2, 4.2, 4.2, 4.2, 4.2 };

double X [size];
double K = 5, tmp = 0.0;

void print_A(string msg = "") {
   cout << msg << endl;
   for (int i = 0; i < size; i++) {
       for (int j = 0; j < size; j++) {
            printf("%f ", A[i][j]);
       }
       printf("| %f", B[i]);
       printf("\n");
   }
}

void gen_A() {
   for (int i = 0; i < size; i++) {
       for (int j = 0; j < size; j++) {
           A[i][j] = K*C[i][j] + D[i][j];
       }
       B[i] = 4.2;
   }
}

void find_solutions() {
```

```
    for (int i = size - 1; i >= 0; i--) {
        for (int j = i; j < size-1; j++) {
            B[i] -= A[i][j+1] * X[j+1];
        }
        X[i] = B[i] / A[i][i];
    }
    printf("\nResult X vector:\n");
    for (int i = 0; i < size; i++) {
        cout << X[i] << " ";
    }
}

void just_gauss() {
    // generate A matrix
    gen_A();
    print_A("A");
    // make it triangle
    for (int i = 0; i < size; i++) {
        for (int j = i + 1; j < size; j++) {
            tmp = A[j][i] / A[i][i];
            for (int k = i + 1; k < size; k++) {
                A[j][k] -= tmp * A[i][k];
            }
            B[j] -= tmp * B[i];
            A[j][i] = 0;
        }
```

```
    }
    print_A("\nA with triangle view");
    find_solutions();
}

void gauss_partial_select() {
    gen_A();
    print_A("A");
    for (int i = 0; i < size; i++) {
        int max_index = i;
        double max = A[i][i];
        for (int j = i + 1; j < size; j++) {
            if (fabs(max) < fabs(A[j][i])) {
                max_index = j;
                max = A[j][i];
            }
        }
        // swap strings
        if (i != max_index) {
            double temp = 0;
            double temp_root = B[i];
            B[i] = B[max_index];
            B[max_index] = temp_root;
            for (int j = i; j < size; j++) {
                temp = A[i][j];
                A[i][j] = A[max_index][j];
```

```
                A[max_index][j] = temp;
            }
        }
        // to triangle vew
        for (int j = i + 1; j < size; j++) {
            tmp = A[j][i]/A[i][i];
            B[j] -= tmp * B[i];
            for (int l = i + 1; l < size; l++) {
                A[j][l] -= tmp * A[i][l];
            }
            A[j][i] = 0;
        }
    }
    print_A("\nA with triangle view");
    find_solutions();
}

void gauss_full_select() {
    gen_A();
    print_A("A");
    for (int i = 0; i < size; i++) {
        int max_index = i;
        double max = A[i][i];
        for (int j = i; j < size; j++) {
            for (int k = i; k < size; k++) {
                if (fabs(max) < fabs(A[j][k])) {
```

```
                max_index = j;
                max = A[j][k];
            }
        }
    }
    // swap strings
    if (i != max_index) {
        double temp_root = B[i];
        double temp = 0;
        B[i] = B[max_index];
        B[max_index] = temp_root;
        for (int j = i; j < size; j++) {
            temp = A[i][j];
            A[i][j] = A[max_index][j];
            A[max_index][j] = temp;
        }
    }
    // to triangle view
    for (int j = i + 1; j < size; j++) {
        tmp = A[j][i] / A[i][i];
        B[j] -= tmp * B[i];
        for (int l = i + 1; l < size; l++) {
            A[j][l] -= tmp * A[i][l];
        }
        A[j][i] = 0;
    }
```

```
    }
    print_A("\nA with triangle view");
    find_solutions();
}

int main()
{
    cout << fixed << setprecision(10);
    // Solving systems of linear equations by the Gauss method
    cout << "\tJust Gauss method: \n";
        just_gauss();
    cout << "\n\n\tGauss partial select method: \n";
        gauss_partial_select();
    cout << "\n\n\tGauss full select method: \n";
        gauss_full_select();
    return 0;
}
```

Полученные результаты

Just Gauss method:
A
3.330000 0.810000 1.670000 0.920000 0.470000 | 4.200000
-0.530000 3.330000 0.810000 1.670000 0.920000 | 4.200000
1.920000 -0.530000 3.330000 0.810000 1.670000 | 4.200000
0.670000 1.920000 -0.530000 3.330000 0.810000 | 4.200000
0.810000 0.670000 1.920000 -0.530000 3.330000 | 4.200000
A with triangle view
3.330000 0.810000 1.670000 0.920000 0.470000 | 4.200000
0.000000 3.458919 1.075796 1.816426 0.994805 | 4.868468

0.000000 0.000000 2.677213 0.803131 1.685760 | 3.181705
0.000000 0.000000 0.000000 2.645931 1.099499 | 2.560563
0.000000 0.000000 0.000000 0.000000 2.805899 | 2.255036
Result X vector:
0.5579 0.6904 0.4922 0.6337 0.8036

Gauss partial select method:
A
3.330000 0.810000 1.670000 0.920000 0.470000 | 4.200000
-0.530000 3.330000 0.810000 1.670000 0.920000 | 4.200000
1.920000 -0.530000 3.330000 0.810000 1.670000 | 4.200000
0.670000 1.920000 -0.530000 3.330000 0.810000 | 4.200000
0.810000 0.670000 1.920000 -0.530000 3.330000 | 4.200000
A with triangle view
3.330000 0.810000 1.670000 0.920000 0.470000 | 4.200000
0.000000 3.458919 1.075796 1.816426 0.994805 | 4.868468
0.000000 0.000000 2.677213 0.803131 1.685760 | 3.181705
0.000000 0.000000 0.000000 2.645931 1.099499 | 2.560563
0.000000 0.000000 0.000000 0.000000 2.805899 | 2.255036
Result X vector:
0.5579 0.6904 0.4922 0.6337 0.8036

Gauss full select method:
A
3.330000 0.810000 1.670000 0.920000 0.470000 | 4.200000
-0.530000 3.330000 0.810000 1.670000 0.920000 | 4.200000
1.920000 -0.530000 3.330000 0.810000 1.670000 | 4.200000
0.670000 1.920000 -0.530000 3.330000 0.810000 | 4.200000
0.810000 0.670000 1.920000 -0.530000 3.330000 | 4.200000
A with triangle view
3.330000 0.810000 1.670000 0.920000 0.470000 | 4.200000
0.000000 3.458919 1.075796 1.816426 0.994805 | 4.868468
0.000000 0.000000 1.366679 -1.002162 3.079646 | 2.512663
0.000000 0.000000 0.000000 2.766285 -4.347015 | -1.740397
0.000000 0.000000 0.000000 0.000000 5.257386 | 4.225240
Result X vector:
0.5579 0.6904 0.4922 0.6337 0.8036

Оценка

$\Delta\left(x_1\right) = \left|x - x_1\right| = 0,5579 - 0.5579158662 = 0,0000158662$

$\Delta\left(x_2\right) = \left|x - x_2\right| = 0,6904 - 0,6904434069 = 0,0000434069$

$\Delta\left(x_3\right) = \left|x - x_3\right| =$ 0,4922 - 0,4922640583 = 0,0000640583

$\Delta\left(x_4\right) = \left|x - x_4\right| = 0,6337 - 0,6337737738 =$ 0,0000737738

$\Delta\left(x_5\right) = \left|x - x_5\right| = 0,8036 - 0,8036769291 = 0,0000769291$

$\delta\left(x_1\right) = \frac{\Delta\left(x_1\right)}{\left|x_1\right|} = \frac{0,0000158662}{0,5579} = 0.00000284391$

$\delta\left(x_2\right) = \frac{\Delta\left(x_2\right)}{\left|x_2\right|} = \frac{0,0000434069}{0,6904} = 0.0000628721$

$\delta\left(x_3\right) = \frac{\Delta\left(x_3\right)}{|x_3|} = \frac{0,0000640583}{0,4922} = 0.000130136$

$\delta\left(x_4\right) = \frac{\Delta\left(x_4\right)}{|x_4|} = \frac{0,0000024091}{0,3909} = 0.000116417$

$\delta\left(x_5\right) = \frac{\Delta\left(x_5\right)}{|x_5|} = \frac{0,0000769291}{0,8036} = 0.0000957305$

Выводы

Таким образом, в ходе выполнения лабораторной работы был применён метод Гаусса, метод Гаусса с выбором главного элемента по столбцу (схема частичного выбора) и метод Гаусса с выбором главного элемента по всей матрице (схема полного выбора) для решения системы линейных уравнений, составлены алгоритмы и созданы реализации соответствующих программ на языке C++ для решения поставленной задачи, также проведена оценка.