Программа gauss.exe

Документация

Алгоритм решения СЛАУ

Рассмотрим систему линейных алгебраических уравнений (СЛАУ) из m уравнений с n неизвестными:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2, \\ \ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m. \end{cases} \qquad (1)$$

Требуется найти все решения данной системы либо определить, что она несовместна (не имеет решений).

Вначале система (1) при помощи элементарных преобразований приводится к ступенчатому виду:

$$\begin{cases} x_{j_1} + a'_{1,j_2}x_{j_2} + a'_{1,j_3}x_{j_3} + \ldots + a'_{1,j_r}x_{j_r} + a'_{1,j_r+1}x_{j_r+1} + \ldots + a'_{1n}x_n = b'_1, \\ \qquad x_{j_2} + a'_{2,j_3}x_{j_3} + \ldots + a'_{2,j_r}x_{j_r} + a'_{2,j_r+1}x_{j_r+1} + \ldots + a'_{2n}x_n = b'_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \qquad x_{j_r} + a'_{r,j_r+1}x_{j_r+1} + \ldots + a'_{rn}x_n = b'_r, \\ \qquad 0 = b'_{r+1}, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \qquad 0 = b'_m. \end{cases}$$

Процесс преобразования системы (1) к ступенчатому виду называется прямым ходом метода Гаусса.

Переменные $x_{j_1}, x_{j_2}, \ldots, x_{j_r}$ называются *главными переменными*, все остальные переменные называются *свободными*.

Если хотя бы одно число $b'_i \neq 0$, где $i > r$, то система несовместна.

Пусть $b'_i = 0$ для всех $i > r$. Рассмотрим обратный ход метода Гаусса. Перенесем свободные переменные в правые части уравнений. Получим

$$\begin{cases} x_{j_1} + a'_{1,j_2}x_{j_2} + a'_{1,j_3}x_{j_3} + \ldots + a'_{1,j_r}x_{j_r} = b'_1 - a'_{1,j_r+1}x_{j_r+1} - \ldots - a'_{1n}x_n, \\ \qquad x_{j_2} + a'_{2,j_3}x_{j_3} + \ldots + a'_{2,j_r}x_{j_r} = b'_2 - a'_{2,j_r+1}x_{j_r+1} - \ldots - a'_{2n}x_n, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \qquad x_{j_r} = b'_r - a'_{r,j_r+1}x_{j_r+1} - \ldots - a'_{rn}x_n. \end{cases}$$

Придадим свободным переменным произвольные значения, а значения главных переменных найдем по формуле

$$x_{j_i} = b'_i - \sum_{j=j_r+1}^{n} a'_{ij} x_j - \sum_{k=i+1}^{r} a'_{i,j_k} x_{j_k}, i = 1, 2, ..., r.$$

Рассмотрим прямой ход метода Гаусса с выбором главного элемента по столбцу.

Сначала находим в 1-м столбце максимальный по модулю элемент (главный элемент). Если все элементы 1-го столбца равны 0, то переменная x_1 уже исключена, переходим к исключению переменной x_2. Если же не все элементы 1-го столбца равны 0, то меняем местами 1-ю строку и строку, в которой находится главный элемент, а затем исключаем переменную x_1 из 2-го, 3-го, …, *m*-го уравнений, домножая 1-е уравнение на коэффициенты $a_{i,1}$ и вычитая его из всех остальных уравнений.

После исключения переменной x_1 аналогичным образом исключаем переменные x_2, x_3 и т.д.

Разработан модуль для решения СЛАУ методом Гаусса в случае, когда матрица системы может быть квадратной вырожденной или прямоугольной.

Программная реализация

Основная функция модуля — `Gauss`, которая объявлена следующим образом:

```
std::vector<std::vector<double> > Gauss(
    std::vector<std::vector<double> > a,
    std::vector<double> b);
```

Функция принимает на вход матрицу системы `a` и вектор правых частей уравнений `b`. Функция возвращает вектор значений переменных, каждый элемент этого вектора, в свою очередь, является вектором, содержащим коэффициенты при свободных переменных и число, которое прибавляется к линейной комбинации свободных переменных. Если система несовместна, то функция возвращает пустой вектор.

Вектор, содержащий значение переменной, устроен так: нулевой элемент вектора содержит число, которое прибавляется к линейной комбинации свободных переменных, а i-й элемент $(i > 1)$ содержит коэффициент при i-й свободной переменной.

Прямой ход метода Гаусса реализован следующим образом. Переменная `i` содержит номер текущего рассматриваемого уравнения, а переменная `j` содержит номер текущей переменной, которую нужно исключить (предполагается, что из уравнений с номерами 1, 2, …, $i - 1$ уже исключены переменные с номерами 1, 2, …, $j - 1$). В процессе прямого хода метода Гаусса вычисляется вектор `jj`, содержащий номера главных переменных. Также вычисляется переменная `r`, содержащая число главных переменных.

```
/* Прямой ход метода Гаусса */

std::vector<int> jj;
int j = 0;
int r = 0;
for (int i = 0; i < m; ++i) {
    double max_abs;
    int k_max;
    while (j <= n-1) {
        //находим максимальный по модулю элемент
        max_abs = 0;
        k_max = i;
        for (int k = i; k < m; ++k) {
            if (fabs(a[k][j]) > max_abs) {
                max_abs = fabs(a[k][j]);
                k_max = k;
            }
        }
```

```
    if (!equal(max_abs, 0))
        break;
    ++j;
}
if (j > n-1)
    break;
++r;
jj.push_back(j);
if (k_max != i) {
    //поменять местами i-ю строчку с k_max-й
    for (int l = j; l < n; ++l)
        swap_double(a[i][l], a[k_max][l]);
    swap_double(b[i], b[k_max]);
}
//делим i-е уравнение на a[i][j]
for (int l = j+1; l < n; ++l) {
    a[i][l] /= a[i][j];
}
b[i] /= a[i][j];
a[i][j] = 1;
//путём элементарных преобразований обнулить
//элементы a[i+1][j], a[i+2][j], ..., a[m-1][j]
for (int k = i+1; k < m; ++k) {
    //умножаем i-е уравнение на a[k][j]
    //и вычитаем из k-го уравнения
    for (int l = j+1; l < n; ++l)
        a[k][l] -= a[i][l]*a[k][j];
    b[k] -= b[i]*a[k][j];
    a[k][j] = 0;
}
```

```
        ++j;
    }
```

Далее система проверяется на совместность.

```
    /* Проверка системы на совместность */

    bool flag = true;
    for (int i = r; i < m; ++i) {
        if (!equal(b[i], 0)) {
            flag = false;
            break;
        }
    }
    if (!flag) {
        //система несовместна
        return ans;
    }
```

После этого определяется, какие переменные будут свободными, и свободным переменным присваивается произвольное значение.

```
    /* Определение свободных переменных */

    int free_vars_count = n-r;
    ans.resize(n);
    for (int j = 0; j < n; ++j)
        ans[j].resize(free_vars_count + 1);
    if (r == 0) {
        for (int j = 0; j < n; ++j)
            ans[j][j+1] = 1;
    }
    else {
        int c = 0;
```

```
        for (int j = 0; j < jj[0]; ++j) {
            ++c;
            ans[j][c] = 1;
        }
        for (int i = 0; i < r-1; ++i) {
            for (int j = jj[i]+1; j < jj[i+1]; ++j) {
                ++c;
                ans[j][c] = 1;
            }
        }
        for (int j = jj[r-1]+1; j < n; ++j) {
            ++c;
            ans[j][c] = 1;
        }
    }
```

Наконец, выполняется обратный ход метода Гаусса, в ходе которого вычисляются значения главных переменных.

```
    /* Обратный ход метода Гаусса */

    for (int i = r-1; i >= 0; --i) {
        ans[jj[i]][0] = b[i];
        for (j = jj[i]+1; j < n; ++j)
            ans[jj[i]] = add(ans[jj[i]],
                mult(ans[j], -a[i][j]));
    }
```

Список литературы

1. Метод Гаусса — Википедия. URL: http://ru.wikipedia.org/wiki/Метод_Гаусса

2. Метод Гаусса: описание алгоритма решения системы линейных

уравнений, примеры, решения. www.cleverstudents.ru. URL: http://www.cleverstudents.ru/solving_systems_Gauss_method.html

3. Метод Гаусса для решения систем линейных уравнений. Математический портал. URL: http://mathportal.net/index.php/linejnaya-algebra/metod-gaussa-metod-zhordana-gaussa?showall=&limitstart=

4. Методы исключения Гаусса. MachineLearning. URL: http://www.machinelearning.ru/wiki/index.php?title=Методы_исключения_Гаусса