

Схема с выбором ведущего элемента

Во избежание необходимости деления на малый ведущий элемент, можно реализовать схему Гаусса с добавкой условия: если $|a_{p,q}| \geq |a_{i,j}|$ для всех $i = k, \dots, n, j = k, \dots, n$, то на очередном шаге поменять местами q -й и k -й столбцы и p -ю и k -ю строки. Так как при перестановке столбцов меняется порядок переменных, необходимо после приведения матрицы к верхнему треугольному (или диагональному) виду восстановить исходный порядок.

Представление данных и результатов:

Требуется написать модуль (на Фортране и других языках, поддерживающих модули) или отдельные файлы (.c/.cpp и .h/.hpp для C/C++ и других языков, не поддерживающих модули) с функцией, получающей матрицу A и вектор B и возвращающей вектор решений. Все три метода должны реализовываться в виде одной функции с дополнительным параметром, задающим тип метода.

Модуль с функцией должен использоваться внешней программой, читающей данные из внешнего файла и записывающей результат в другой внешний файл. Выбор используемого метода должен задаваться ключом командной строки. На экран программа должна выводить модуль вектора невязки (т.е. модуль вектора $AX - B$, где X — найденное решение).

Файл с исходными данными называется *data.dat* и содержит:

- В первой строке — после символа $\#$ и пробела размер системы n (одно натуральное число).
- В последующих n строках — матрицу A . В каждой строке файла содержатся элементы одной строки матрицы, отделенные друг от друга одним или несколькими пробелами.
- В последующих n строках — вектор B , по одному элементу в строке.

Файл для вывода результата называется *result.dat* и содержит в первой строке символ $\#$, пробел и размер системы n , в последующих строках — вектор результата X (по одному элементу в строке). Имейте в виду, что n может быть достаточно большим ($n \sim 10^{(3 \div 4)}$).

Примечания:

При использовании Фортрана рекомендуется использовать встроенные средства языка для работы с матрицами, а также конструкцию **forall**

```
forall (<заголовок>)  
    <операторы>  
end forall
```

или ее же в краткой форме

```
forall (<заголовок>) <оператор>
```

Заголовок имеет вид «триплет, триплет, ..., логическая маска» (где триплет — выражение вида «I=1:15:2»). Важно, что это не обычный цикл, а «параллельное выполнение» — сначала независимо вычисляются все правые части присваиваний, находящихся в блоке, а только потом «параллельно» выполняются присваивания. Ничего, кроме присваиваний, внутри блока быть не должно.

Возможно, полезной окажется также конструкция выборки

```
where (<логическое условие>) <действие>
```

или

```
where (<логическое условие или логический массив>)  
    <действия>  
elsewhere  
    <другие действия>  
end where
```

При реализации схемы с выбором ведущего элемента рекомендуется воспользоваться функциями **maxloc** и **minloc** (поиск максимального и минимального значения в массиве). Аргументы (array,dim,mask) — массив, конкретная размерность (если нужна), логическая маска (также если нужна). Помните, что функции такого типа дают на выходе координаты относительно начала массива-аргумента (считая, что его индексация по каждой из размерностей начинается с 1).

Исследование:

Попробуйте решить разными модификациями метода Гаусса систему с матрицей Гильберта, для которой $a_{ij} = \frac{1}{i+j-1}$. Вектор свободных членов можно взять случайным. Посмотрите, что будет происходить с решением, если вектор свободных членов незначительно изменится (например, отдельные его элементы изменятся на некоторую малую величину). Что получается и как это объяснить?