УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Вычислительная математика

Лабораторная работа №1

Решение систем линейных алгебраических уравнений

Преподаватель: Перл Ольга Вячеславовна
Выполнил: Ле Чонг Дат
Группа: P3231

2021 г.

1 Задание

Вариант: Метод Гаусса

- Размерность n<=20 (задается из файла или с клавиатуры - по выбору конечного пользователя)
- Должно быть предусмотрено чтение исходных данных как из файла, так и ввод с клавиатуры.
- Должна быть реализована возможность ввода коэффициентов матрицы как с клавиатуры, так и из файла. Также предусмотреть случайные коэффициенты.
- Обязательно: Тестовые данные на матрице большого размера (5*5 / 6*6...) + в отчёт с решением.
- Для точных методов(Гаусс и главные элементы) должно быть реализовано:
  - Вычисление определителя
  - Вывод треугольной матрицы (включая преобразованный столбец B)
  - Столбец неизвестных
  - Столбец невязок

2 Описание метода

2.1 Problem

Дана система n линейных алгебраических уравнений (СЛАУ) с m неизвестными. Вам предлагается решить систему: определить, нет ли у нее решения, ровно одно решение или бесконечное количество решений. И если есть хоть одно решение, найдите любое из них.

Формально задача формулируется так: решить систему:

$$\begin{gathered} a_{11}x_1 + a_{12}x_2 + ... + a_{1m}x_m = b_1 \\ a_{12}x_1 + a_{22}x_2 + ... + a_{2m}x_m = b_2 \\ ... \\ a_{n1}x_1 + a_{n2}x_2 + ... + a_{nm}x_m = b_n \end{gathered}$$

где коэффициенты a_{ij} (для i от 1 до n, для j от 1 до m) и b_i (i от 1 до n) известны и переменные x_i (i от 1 до m) неизвестны.

Эта задача также имеет простое матричное представление:

$$Ax = B$$

где A - это матрица размера $n*m$ коэффициентов a_{ij}, а B - вектор-столбец размера n.

2.2 Метод Гаусса

Алгоритм представляет собой последовательное исключение переменных в каждом уравнении, пока в каждом уравнении не останется только одна оставшаяся переменная.

Исключение Гаусса основано на двух простых преобразованиях:

- Можно поменять местами два уравнения
- Любое уравнение можно заменить линейной комбинацией этой строки (с ненулевым коэффициентом) и некоторых других строк (с произвольными коэффициентами).

На первом этапе алгоритм Гаусса-Жордана делит первую строку на a_{11}. Затем алгоритм добавляет первую строку к оставшимся строкам, так что все коэффициенты в первом столбце становятся нулями. Для этого в i-th строку мы должны добавить первую строку, умноженную на $-a_{11}$. Обратите внимание, что эта операция также должна выполняться с вектором B. В некотором смысле он ведет себя так, как если бы вектор B был m+1-th столбцом матрицы A.

В результате после первого шага первый столбец матрицы A будет состоять из 1 в первой строке и 0 в остальных строках.

Аналогично выполняем второй шаг алгоритма, где рассматриваем второй столбец второй строки. Сначала строка делится на a_{22} , затем он вычитается из других строк, так что весь второй столбец становится 0 (кроме второй строки).

Продолжим этот процесс для всех столбцов матрицы A. Если $n = m$, то A станет единичной матрицей.

2.3 Поиск поворотного элемента

В описанной схеме не учтены многие детали. На i-м шаге, если a_{ii} равно нулю, мы не можем напрямую применить описанный метод. Вместо этого мы должны сначала выбрать поворотную строку: найти одну строку матрицы, в которой i-й столбец не равен нулю, а затем поменять местами две строки.

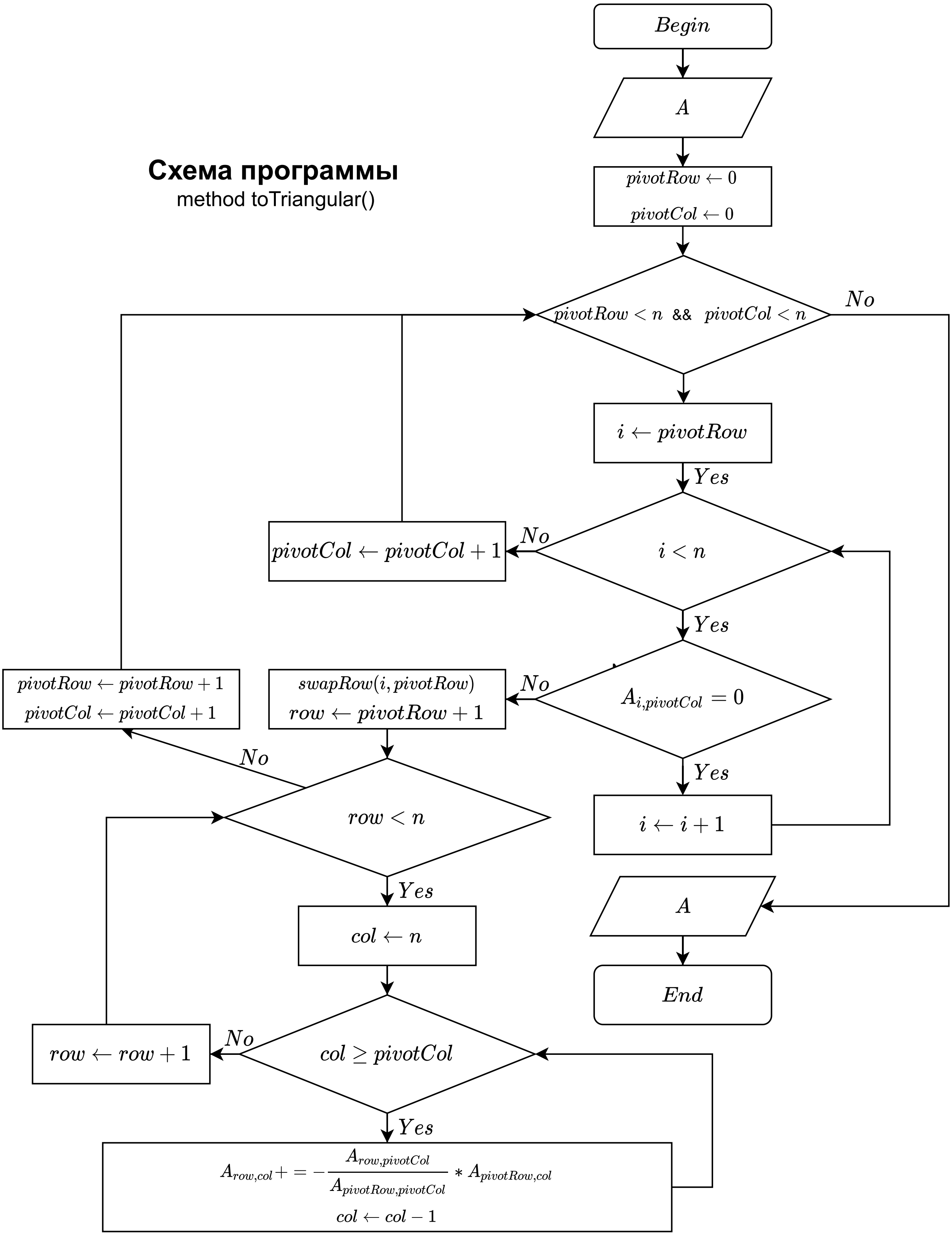
2.4 Complexity

Теперь мы должны оценить сложность этого алгоритма. Алгоритм состоит из m фаз, в каждой фазе:

Найти и перетасовать поворотную строку. Это требует $O(n + m)$ при использовании эвристики, упомянутой выше. Если ключевой элемент в текущем столбце найден - тогда мы должны добавить это уравнение ко всем остальным уравнениям, что требует времени $O(nm)$. Таким образом, окончательная сложность алгоритма составляет $O(min(n, m).nm)$. В случае $n = m$ сложность будет просто $O(n^3)$.

3 Flowchart

Схема программы
method toTriangular()



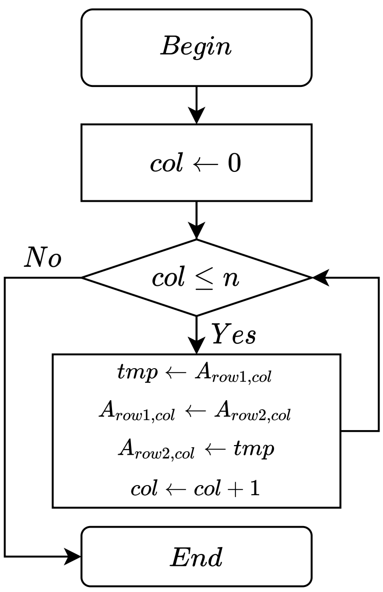


Схема подпрограммы
method swapRow(row1, row2)

4 Вывод

Во многих реализациях, когда $a_{ii} \neq 0$, вы можете видеть, что люди попрежнему меняют местами i-ю строку какой-нибудь поворотной строкой, используя некоторые эвристики, такие как выбор поворотной строки с максимальным абсолютным значением a_{ji}. Эта эвристика используется для уменьшения диапазона значений матрицы на более поздних этапах. Без этой эвристики даже для матриц размером около 20 ошибка будет слишком большой и может вызвать переполнение для типов данных с плавающей запятой.