

20. Метод Гаусса и явление плохой
обусловленности. LU-разложение матрицы.
Подпрограммы DECOMP и SOLVE

Андрей Бареков Ярослав Пылаев
По лекциям Устинова С.М.

January 9, 2020

1 Явление плохой обусловленности матриц

Дана линейная алгебраическая система:

$$Ax = b, \quad \det(A) \neq 0, \quad (1)$$

A – квадратная матрица,

b – заданный вектор-столбец,

x – искомый вектор-столбец.

Как погрешность в исходных данных влияет на точность решения?

Если малые изменения в исходной информации приводят к сильному изменению решения, то такая матрица (и сама система (1)) называется *плохо обусловленной*.

Получим *количественные характеристики* плохой обусловленности.

I.

$$\begin{aligned} b &\rightarrow b + \Delta b \Rightarrow x \rightarrow x + \Delta x, \\ A(x + \Delta x) &= b + \Delta b, \quad Ax + A\Delta x = b + \Delta b, \\ A\Delta x &= \Delta b, \quad \Delta x = A^{-1}\Delta b, \end{aligned}$$

$$\|\Delta x\| \leq \|A^{-1}\| \cdot \|\Delta b\|, \quad \|b\| \leq \|A\| \cdot \|x\|,$$

перемножим эти два неравенства и разделим результат на $\|x\| \cdot \|b\|$:

$$\frac{\|\Delta x\|}{\|x\|} \leq \underbrace{\|A\| \cdot \|A^{-1}\|}_{\text{cond}(A)} \cdot \frac{\|\Delta b\|}{\|b\|}.$$

$$\begin{aligned} \frac{\|\Delta x\|}{\|x\|} &- \text{относительная погрешность результата,} \\ \frac{\|\Delta b\|}{\|b\|} &- \text{относительная погрешность исходных данных.} \end{aligned}$$

$\text{cond}(A)$ - *число обусловленности матрицы* или коэффициент усиления погрешности. Если $\text{cond}(A) \gg 1$, то система - плохо обусловлена.

II.

$$\begin{aligned}
A \rightarrow A + \Delta A &\Rightarrow x \rightarrow x + \Delta x, \\
(A + \Delta A)(x + \Delta x) &= b, \\
\cancel{Ax} + A\Delta x + \Delta A(x + \Delta x) &= b, \\
A^{-1} \Big| \quad A\Delta x &= -\Delta A(x + \Delta x), \\
\Delta x &= A^{-1} \Delta A(x + \Delta x), \\
\text{берем норму и делим на } \|x + \Delta x\| : \\
\frac{\|\Delta x\|}{\|x + \Delta x\|} &\leq \|A^{-1}\| \cdot \|\Delta A\| \cdot \frac{\|A\|}{\|A\|}, \\
\frac{\|\Delta x\|}{\|x + \Delta x\|} &\leq \underbrace{\|A\| \cdot \|A^{-1}\|}_{\text{cond}(A)} \cdot \frac{\|\Delta A\|}{\|A\|}.
\end{aligned}$$

Пример:

$$\begin{aligned}
10^{-5} &\leq \underline{10^7} \cdot 10^{-12} - \text{погрешность размерной сетки,} \\
10 &\leq \underline{10^4} \cdot 10^{-3} - \text{инженерная погрешность}
\end{aligned}$$

Таким образом, все зависит от задачи.

Существуют и другие числа обусловленности, так плохо обусловленной матрицей является матрица с большим разбросом собственных значений:

$$k(A) = \frac{|\lambda_k|_{\max}}{|\lambda_k|_{\min}}.$$

Легко показать, что $k(A) \leq \text{cond}(A)$:

$$\begin{aligned}
\frac{|\lambda_k|_{\max}}{|\lambda_k|_{\min}} &\leq \|A\| \cdot \|A^{-1}\|, \\
\text{по Следствию 2 к Теореме 5 } |\lambda_k| &\leq \|A\|, \quad \frac{1}{|\lambda_k|} \geq \|A^{-1}\|.
\end{aligned}$$

Покажем, что матрица с большим разбросом собственных значений является плохо обусловленной, что приводит к большой погрешности.

Утверждение 1. Максимальные по модулю элементы матрицы A имеют величину порядка $|\lambda_k|_{\max}$, а могут и значительно ее превышать. Это следует из неравенства $|\lambda_k| \leq \|A\|$.

Утверждение 2. Собственные значения λ_k могут изменяться на величину порядка элементов матрицы возмущения ΔA .

$$\begin{aligned}\Delta A &= \varepsilon E, & A + \Delta A &\rightarrow A + \varepsilon E, \\ A &\rightarrow \lambda_k, & A + \varepsilon E &\rightarrow \lambda_k + \varepsilon.\end{aligned}$$

Пусть относительная погрешность элементов матрицы A $\delta \ll 1$.

Шаг 1.

$$\begin{aligned}\Delta A &\rightarrow \delta |a_{ik}|_{\max}, \\ \text{по утверждению 1: } \Delta A &\rightarrow \delta |\lambda_k|_{\max}.\end{aligned}$$

Шаг 2. В соответствии с утверждением 2 $\lambda_k \pm \delta |\lambda_k|_{\max}$. $|\lambda_k|_{\max}$ изменится крайне незначительно, а $|\lambda_k|_{\min}$ может измениться достаточно сильно.

Шаг 3.

$$\begin{aligned}A &\rightarrow \lambda_k, A^{-1} \rightarrow \frac{1}{\lambda_k}, \\ \frac{1}{|\lambda_k|_{\min}} &\text{ — самые большие собственные значения } A^{-1}.\end{aligned}$$

Если $|\lambda_k|_{\min}$ изменилось сильно, то сильно изменились элементы обратной матрицы, а, следовательно, сильно изменилось решение.

Врем в *min* собственном значении, значит врем в обратной матрице.

$$\begin{aligned}Ax &= b, \\ x &= A^{-1}b,\end{aligned}$$

Описанные неприятности будут появляться тем быстрее, чем *больше разброс* элементов матрицы A .

2 Метод Гаусса

Методы решения системы (1) делятся на две большие группы:

- точные,
- итерационные.

Точные методы в отсутствии ошибок округления позволяют получить точные решения за конечное число арифметических операций. В ходе применения *итерационных* методов рождается последовательность векторов, сходящихся к решению.

Типичный точный метод - **метод Гаусса**. Его "грубая" схема выглядит следующим образом:

Прямой ход. В первом уравнении делим все на a_{11} , выразим x_1 , подставляем во все остальные уравнения и приводим подобные. Аналогично из второго уравнения *исключаем* x_2 и т.д. В результате получаем верхнюю треугольную матрицу.

Обратный ход.

$$\begin{pmatrix} * & * & * \\ & * & * \\ 0 & & * \end{pmatrix}$$

Из последнего уравнения находим x_N , из предпоследнего - x_{N-1} , и т.д.

Недостаток "грубой" схемы - возможное деление на нулевой или близкий к нулю элемент. Для его устранения используется *выбор ведущего элемента*.

Вариант 1. На k -м шаге в оставшейся матрице находят самый большой по модулю элемент. Затем строки и столбцы меняют так, чтобы этот элемент поменялся местами с a_{kk} ,

$$\begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & a_{kk} & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & * & * & * \end{pmatrix}.$$

Делим уже на него и далее по алгоритму. *Но* в этом варианте при перестановке столбцов меняется нумерация компонент вектора x .

Вариант 2. На k -м шаге максимальный по модулю элемент разыскивают только в k -м столбце, в таком случае переставляются только строки. На практике этот вариант используют чаще.

2.1 Требования к хорошей программе метода Гаусса

1. Должен быть реализован выбор ведущего элемента.

2. Программа должна оценивать число обусловленности матрицы - $cond(A)$.
3. Программа должна эффективно решать несколько систем уравнений с одной и той же матрицей A и различными векторами b .

Реализовать 3-е требование помогает LU -разложение матриц.

3 LU -разложение матрицы

LU -разложение матрицы - представление матрицы в следующем виде:

$$A = LU, \text{ где}$$

L — левая треугольная матрица с единичной диагональю,

U — правая треугольная матрица, на диагонали - все, что угодно.

Если LU -разложение построено, то решение исходной системы $Ax = b$ сводится к решению двух систем с треугольными матрицами:

$$Ly = b, \tag{2}$$

$$Ux = y. \tag{3}$$

ввели
обознач.

LU -разложение выполняется однократно, а системы (2) и (3) решаются столько раз, сколько имеется различных векторов x .

3.1 Пример - матрица 4×4

$$A = A_0 = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix},$$

Рассмотрим левую треугольную матрицу M_1 , которая отличается от единичной первым столбцом:

$$M_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -a_{21}/a_{11} & 1 & 0 & 0 \\ -a_{31}/a_{11} & 0 & 1 & 0 \\ -a_{41}/a_{11} & 0 & 0 & 1 \end{pmatrix},$$

Умножим исходную матрицу на матрицу M_1 слева:

$$A_1 = M_1 A_0 = \begin{pmatrix} a_{11} & \dots & \dots & a_{14}^* \\ 0 & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots \end{pmatrix}.$$

На втором шаге строим матрицу M_2 , которая отличается от единичной вторым столбцом (* - элементы матрицы A_1):

$$M_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -a_{32}^*/a_{22}^* & 1 & 0 \\ 0 & -a_{42}^*/a_{22}^* & 0 & 1 \end{pmatrix},$$

$$A_2 = M_2 A_1 = \begin{pmatrix} + & + & \dots & a_{14}^{**} \\ 0 & + & \dots & \dots \\ 0 & 0 & \dots & \dots \\ 0 & 0 & \dots & \dots \end{pmatrix},$$

+ - элементы не обязательно нулевые.

На последнем шаге умножаем A_2 на M_3 , где

$$M_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -a_{43}^{**}/a_{33}^{**} & 1 \end{pmatrix}.$$

Тогда

$$A_3 = M_3 A_2 = \underbrace{M_1 M_2 M_3}_M A = U,$$

$$M^{-1} \Big| \quad MA = U,$$

$$A = M^{-1}U, \quad M^{-1} = L,$$

$$A = LU.$$

Здесь учитывались "Задачи на матрицы" 6 и 7.

4 Подпрограммы DECOMP и SOLVE

Программное обеспечение состоит из двух программ:

- *DECOMP*(*NDIM*, *N*, *A*, *COND*, *IPVT*, *WORK*)

Выполняет *LU* разложение матрицы,

NDIM - длина столбца матрицы *A* в операторе описания,

N - порядок матрицы (системы уравнений),

A - исходная матрица, в ней же будут размещены матрицы *L* и *U*,

COND - оценка числа обусловленности, *выходной*

IPVT - вектор индексов ведущих элементов, его *k*-я компонента указывает, какое уравнение используется для исключения x_k на *k*-м шаге,

WORK - одномерный рабочий массив (размерность *N*).

- $SOLVE(NDIMgit, N, A, B, IPVT)$
Решает системы (2) и (3) с треугольными матрицами,
 B - вектор решения x , сначала в нем правые части системы (1).

4.1 Нахождение обратной матрицы с помощью программ DECOMP и SOLVE

$X = A^{-1}$, x_k - k -й столбец обратной матрицы.

$$AX = E,$$

$\underbrace{Ax_k = e_k}_{k=1,2,\dots,N}$, e_k - k -й столбец единичной матрицы.

Однократно вызывается программа *DECOMP* и строится *LU*-разложение матрицы.

Цикл по k от 1 до N :

$$e_k \rightarrow B,$$

$$SOLVE(\dots, A, B, \dots),$$

$$B \rightarrow k\text{-й столбец } A^{-1}.$$