

Лекции по Линейной Алгебре

Дима Трушин

2021 — 2022

Содержание

1 Системы линейных уравнений	2
1.1 Системы линейных уравнений и связанная с ними терминология	2
1.2 Матрицы связанные со СЛУ	2
1.3 Элементарные преобразования	3
1.4 Алгоритм Гаусса	3

1 Системы линейных уравнений

1.1 Системы линейных уравнений и связанная с ними терминология

Наша задача научиться решать Системы Линейных Уравнений (СЛУ), то есть находить все их решения или доказывать, что решений нет. Общий вид СЛУ и ее однородная версия (ОСЛУ):

$$\begin{cases} a_{11}x_1 + \dots + a_{1n}x_n = b_1 \\ \dots \\ a_{m1}x_1 + \dots + a_{mn}x_n = b_m \end{cases} \quad \begin{cases} a_{11}x_1 + \dots + a_{1n}x_n = 0 \\ \dots \\ a_{m1}x_1 + \dots + a_{mn}x_n = 0 \end{cases}$$

Коэффициенты Где живут коэффициенты a_{ij} и b_j ? Варианты:

- Вещественные числа \mathbb{R}
- Комплексные числа \mathbb{C}
- Рациональные числа \mathbb{Q}

Для решения СЛУ **НЕ** имеет значения откуда берутся коэффициенты, так как решения будут лежать там же. Потому мы будем работать с числами из \mathbb{R} .

Решение Решением системы линейных уравнений называется набор чисел (c_1, \dots, c_n) , $c_i \in \mathbb{R}$ такой, что при подстановке c_i вместо x_i , все уравнения системы превращаются в верные равенства. Введем обозначение $\mathbb{R}^n = \mathbb{R} \times \dots \times \mathbb{R} = \{(c_1, \dots, c_n) \mid c_i \in \mathbb{R}\}$. То есть элемент \mathbb{R}^n – это набор из n вещественных чисел. Потому любое решение $c = (c_1, \dots, c_n)$ является элементом \mathbb{R}^n .

1.2 Матрицы связанные со СЛУ

Для каждой СЛУ введем следующие обозначения:

$$A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix} \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} \quad x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \quad (A|b) = \left(\begin{array}{ccc|c} a_{11} & \dots & a_{1n} & b_1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{m1} & \dots & a_{mn} & b_m \end{array} \right)$$

Названия:

- A – матрица системы
- b – вектор правой части
- $(A|b)$ – расширенная матрица системы
- x – вектор решений

Будем кратко записывать СЛУ и ее однородную версию так: $Ax = b$ и $Ax = 0$. Также для краткости будем обозначать системы буквами Σ .

При решении системы линейных уравнений приходится много раз переписывать кучу данных, чтобы сократить эти записи целесообразно сократить количество записываемой на бумаге информации. Расширенная матрица системы $(A|b)$ является необходимым минимумом такой информации. Потому сейчас к такой записи можно относиться как к удобному способу компактно записать систему.

Количество решений Случай одного уравнения и одной неизвестной $ax = b$, где $a, b \in \mathbb{R}$:

- При $a \neq 0$ – одно решение $x = b/a$.
- При $a = 0$, $b \neq 0$ – нет решений.
- При $a = 0$, $b = 0$ – любое число является решением, т.е. бесконечное число решений.

Что значит решить систему Решить систему значит описать множество ее решений, то есть либо доказать, что система не имеет решений вовсе, либо описать все наборы, которые являются решениями. Если система не имеет решений, она называется несовместной, в противном случае – совместной.

Эквивалентные системы Пусть даны две системы линейных уравнений с одинаковым числом неизвестных (но быть может разным числом уравнений) Σ_1 и Σ_2 . Будем говорить, что эти системы эквивалентны и писать $\Sigma_1 \sim \Sigma_2$, если множества решений этих систем совпадают. Если $E_i \subseteq \mathbb{R}^n$ – множество решений i -ой системы, то системы эквивалентны, если $E_1 = E_2$.

Вот полезный пример эквивалентных систем:

$$\begin{cases} x + y = 1 \\ x - y = 0 \end{cases} \sim \begin{cases} 2x = 1 \\ 2y = 1 \end{cases}$$

Как решать систему Пусть нам надо решить систему Σ . Идея состоит в том, чтобы постепенно менять ее на эквивалентную до тех пор, пока она не упростится до такого состояния, что все ее решения становятся легко описываемые.

$$\Sigma = \Sigma_1 \mapsto \Sigma_2 \mapsto \dots \mapsto \Sigma_n \leftarrow \text{легко решается}$$

Теперь надо объяснить две вещи: (1) какого сорта преобразования над системами мы будем делать и (2) к какому замечательному виду мы их приводим и как в нем выглядят все решения. Ответам на эти два вопроса и будет посвящена оставшаяся часть лекции.

1.3 Элементарные преобразования

Мы разделим все преобразования на три типа¹:

$$\begin{aligned} \text{I тип: } & \left(\begin{array}{ccc|c} a_{11} & \dots & a_{1n} & b_1 \\ a_{i1} & \dots & a_{in} & b_i \\ a_{j1} & \dots & a_{jn} & b_j \\ a_{m1} & \dots & a_{mn} & b_m \end{array} \right) \mapsto \left(\begin{array}{ccc|c} a_{11} & \dots & a_{1n} & b_1 \\ a_{i1} & \dots & a_{in} & b_i \\ a_{j1} + \lambda a_{i1} & \dots & a_{jn} + \lambda a_{in} & b_j + \lambda b_i \\ a_{m1} & \dots & a_{mn} & b_m \end{array} \right) \quad i \neq j \\ \\ \text{II тип: } & \left(\begin{array}{ccc|c} a_{11} & \dots & a_{1n} & b_1 \\ a_{i1} & \dots & a_{in} & b_i \\ a_{j1} & \dots & a_{jn} & b_j \\ a_{m1} & \dots & a_{mn} & b_m \end{array} \right) \mapsto \left(\begin{array}{ccc|c} a_{11} & \dots & a_{1n} & b_1 \\ a_{j1} & \dots & a_{jn} & b_j \\ a_{i1} & \dots & a_{in} & b_i \\ a_{m1} & \dots & a_{mn} & b_m \end{array} \right) \\ \\ \text{III тип: } & \left(\begin{array}{ccc|c} a_{11} & \dots & a_{1n} & b_1 \\ a_{i1} & \dots & a_{in} & b_i \\ a_{m1} & \dots & a_{mn} & b_m \end{array} \right) \mapsto \left(\begin{array}{ccc|c} a_{11} & \dots & a_{1n} & b_1 \\ \lambda a_{i1} & \dots & \lambda a_{in} & \lambda b_i \\ a_{m1} & \dots & a_{mn} & b_m \end{array} \right) \quad \lambda \neq 0 \end{aligned}$$

Поясним словами, что делают преобразования:

1. Прибавляем к j -ой строке i -ю, умноженную на константу $\lambda \in \mathbb{R}$.
2. Меняем местами i -ю и j -ю строки.
3. Умножаем i -ю строку на ненулевую константу $\lambda \neq 0$, $\lambda \in \mathbb{R}$.

1.4 Алгоритм Гаусса

Этот метод заключается в приведении СЛУ к некоторому «ступенчатому виду», где множество решений очевидно.² Разберем типичный ход алгоритма Гаусса на примере 3 уравнений и 4 неизвестных.³

¹Стоит отметить, что нумерация типов преобразования не является общепринятой и отличается от учебника к учебнику.

²Данный метод является самым быстрым возможным как для написания программ, так и для ручного вычисления. При вычислениях руками, однако, полезно местами пользоваться «локальными оптимизациями», то есть, если вы видите, что какая-то хитрая комбинация строк сильно упростит вид системы, то сделайте ее.

³При переходе от одной матрицы к другой я новым коэффициентам даю старые имена, чтобы не захламлять текст новыми обозначениями.

Прямой ход алгоритма Гаусса

$$\begin{aligned}
 & \left(\begin{array}{cccc|c} a_{11} & a_{12} & a_{13} & a_{14} & b_1 \\ a_{21} & a_{22} & a_{23} & a_{24} & b_2 \\ a_{31} & a_{32} & a_{33} & a_{34} & b_3 \end{array} \right) \quad \begin{array}{l} \text{2-я строка} \\ \text{3-я строка} \end{array} \quad \begin{array}{l} - \frac{a_{21}}{a_{11}} \cdot \text{1-я строка} \\ - \frac{a_{31}}{a_{11}} \cdot \text{1-я строка} \end{array} \\
 & \left(\begin{array}{cccc|c} a_{11} & a_{12} & a_{13} & a_{14} & b_1 \\ 0 & a_{22} & a_{23} & a_{24} & b_2 \\ a_{31} & a_{32} & a_{33} & a_{34} & b_3 \end{array} \right) \quad \begin{array}{l} \text{3-я строка} \\ \text{3-я строка} \end{array} \quad \begin{array}{l} - \frac{a_{31}}{a_{11}} \cdot \text{1-я строка} \\ - \frac{a_{32}}{a_{22}} \cdot \text{2-я строка} \end{array} \\
 & \left(\begin{array}{cccc|c} a_{11} & a_{12} & a_{13} & a_{14} & b_1 \\ 0 & a_{22} & a_{23} & a_{24} & b_2 \\ 0 & a_{32} & a_{33} & a_{34} & b_3 \end{array} \right) \\
 & \left(\begin{array}{cccc|c} a_{11} & a_{12} & a_{13} & a_{14} & b_1 \\ 0 & a_{22} & a_{23} & a_{24} & b_2 \\ 0 & 0 & a_{33} & a_{34} & b_3 \end{array} \right)
 \end{aligned}$$

В результате данного хода какие-то коэффициенты, например a_{33} , могли занулиться, потому возможны следующие принципиально другие случаи⁴

$$\left(\begin{array}{cccc|c} \underline{a_{11}} & a_{12} & a_{13} & a_{14} & b_1 \\ 0 & \underline{a_{22}} & a_{23} & a_{24} & b_2 \\ 0 & 0 & 0 & \underline{a_{34}} & b_3 \end{array} \right) \quad \left(\begin{array}{cccc|c} \underline{a_{11}} & a_{12} & a_{13} & a_{14} & b_1 \\ 0 & 0 & \underline{a_{23}} & a_{24} & b_2 \\ 0 & 0 & 0 & \underline{a_{34}} & b_3 \end{array} \right) \quad \left(\begin{array}{cccc|c} \underline{a_{11}} & a_{12} & a_{13} & a_{14} & b_1 \\ 0 & \underline{a_{22}} & a_{23} & a_{24} & b_2 \\ 0 & 0 & 0 & 0 & \underline{b_3} \end{array} \right) \quad \left(\begin{array}{cccc|c} \underline{a_{11}} & a_{12} & a_{13} & a_{14} & b_1 \\ 0 & \underline{a_{22}} & a_{23} & a_{24} & b_2 \\ 0 & 0 & 0 & 0 & 0 \end{array} \right)$$

Главные и неглавные переменные Подчеркнутые элементы считаются не равными нулю. В ступенчатом виде все переменные (и соответственно коэффициенты перед ними) делятся на главные и неглавные. Главные коэффициенты – это первые ненулевые коэффициенты в строке (подчеркнутые). Переменные при них называются главными, остальные ненулевые коэффициенты и переменные – неглавные.

Обратный ход алгоритма Гаусса Разберем типичный обратный ход алгоритма Гаусса. Подчеркнутые элементы считаются не равными нулю.

$$\begin{aligned}
 & \left(\begin{array}{cccc|c} \underline{a_{11}} & a_{12} & a_{13} & a_{14} & b_1 \\ 0 & \underline{a_{22}} & a_{23} & a_{24} & b_2 \\ 0 & 0 & \underline{a_{33}} & a_{34} & b_3 \end{array} \right) \quad \begin{array}{l} \text{разделить } i\text{-ю строку на } a_{ii} \\ \text{2-я строка} \end{array} \\
 & \left(\begin{array}{cccc|c} 1 & a_{12} & a_{13} & a_{14} & b_1 \\ 0 & 1 & a_{23} & a_{24} & b_2 \\ 0 & 0 & 1 & a_{34} & b_3 \end{array} \right) \quad \begin{array}{l} \text{2-я строка} \\ \text{1-я строка} \end{array} \quad \begin{array}{l} - a_{23} \cdot \text{3-я строка} \\ - a_{13} \cdot \text{3-я строка} \end{array} \\
 & \left(\begin{array}{cccc|c} 1 & a_{12} & a_{13} & a_{14} & b_1 \\ 0 & 1 & 0 & a_{24} & b_2 \\ 0 & 0 & 1 & a_{34} & b_3 \end{array} \right) \\
 & \left(\begin{array}{cccc|c} 1 & a_{12} & 0 & a_{14} & b_1 \\ 0 & 1 & 0 & a_{24} & b_2 \\ 0 & 0 & 1 & a_{34} & b_3 \end{array} \right) \quad \begin{array}{l} \text{1-я строка} \\ \text{1-я строка} \end{array} \quad \begin{array}{l} - a_{12} \cdot \text{2-я строка} \\ \end{array} \\
 & \left(\begin{array}{cccc|c} 1 & 0 & 0 & a_{14} & b_1 \\ 0 & 1 & 0 & a_{24} & b_2 \\ 0 & 0 & 1 & a_{34} & b_3 \end{array} \right)
 \end{aligned}$$

В специальных случаях приведенных выше, получим

$$\left(\begin{array}{cccc|c} 1 & 0 & a_{13} & 0 & b_1 \\ 0 & 1 & a_{23} & 0 & b_2 \\ 0 & 0 & 0 & 1 & b_3 \end{array} \right) \quad \left(\begin{array}{cccc|c} 1 & a_{12} & 0 & 0 & b_1 \\ 0 & 0 & 1 & 0 & b_2 \\ 0 & 0 & 0 & 1 & b_3 \end{array} \right) \quad \left(\begin{array}{cccc|c} 1 & 0 & a_{13} & a_{14} & 0 \\ 0 & 1 & a_{23} & a_{24} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array} \right) \quad \left(\begin{array}{cccc|c} 1 & 0 & a_{13} & a_{14} & b_1 \\ 0 & 1 & a_{23} & a_{24} & b_2 \\ 0 & 0 & 0 & 0 & 0 \end{array} \right)$$

Полученный в результате обратного хода вид расширенной матрицы называется улучшенным ступенчатым видом, т.е., это ступенчатый вид, где все коэффициенты при главных неизвестных – единицы, и все коэффициенты над ними равны нулю.

⁴Это не полный список всех случаев.

Удобный формализм Пока мы подробно не говорили о матрицах, введем некие удобные обозначения, которые упростят запись решений СЛУ.

$$a = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} \in \mathbb{R}^n \text{ и } b = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} \in \mathbb{R}^n. \text{ Тогда } a + b = \begin{pmatrix} a_1 + b_1 \\ \vdots \\ a_n + b_n \end{pmatrix} \in \mathbb{R}^n \text{ и } \lambda a = \begin{pmatrix} \lambda a_1 \\ \vdots \\ \lambda a_n \end{pmatrix} \in \mathbb{R}^n \text{ для любого } \lambda \in \mathbb{R}.$$

Получение решений В системе ниже, выберем переменную x_4 как параметр

$$\left(\begin{array}{cccc|c} 1 & 0 & 0 & a_{14} & b_1 \\ 0 & 1 & 0 & a_{24} & b_2 \\ 0 & 0 & 1 & a_{34} & b_3 \end{array} \right)$$

Тогда решения имеют вид⁵

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} - x_4 \begin{pmatrix} a_{14} \\ a_{24} \\ a_{34} \end{pmatrix}$$

Специальные случаи:

$$\begin{array}{ll} \left(\begin{array}{cccc|c} 1 & 0 & a_{13} & 0 & b_1 \\ 0 & 1 & a_{23} & 0 & b_2 \\ 0 & 0 & 0 & 1 & b_3 \end{array} \right) & \text{Решения: } \begin{pmatrix} x_1 \\ x_2 \\ x_4 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} - x_3 \begin{pmatrix} a_{13} \\ a_{23} \\ 0 \end{pmatrix} \\ \left(\begin{array}{cccc|c} 1 & a_{12} & 0 & 0 & b_1 \\ 0 & 0 & 1 & 0 & b_2 \\ 0 & 0 & 0 & 1 & b_3 \end{array} \right) & \text{Решения: } \begin{pmatrix} x_1 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} - x_2 \begin{pmatrix} a_{12} \\ 0 \\ 0 \end{pmatrix} \\ \left(\begin{array}{cccc|c} 1 & 0 & a_{13} & a_{14} & 0 \\ 0 & 1 & a_{23} & a_{24} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array} \right) & \text{Решения: Нет решений, т.к. последнее уравнение } 0 = 1 \\ \left(\begin{array}{cccc|c} 1 & 0 & a_{13} & a_{14} & b_1 \\ 0 & 1 & a_{23} & a_{24} & b_2 \\ 0 & 0 & 0 & 0 & 0 \end{array} \right) & \text{Решения: } \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} - x_3 \begin{pmatrix} a_{13} \\ a_{23} \end{pmatrix} - x_4 \begin{pmatrix} a_{14} \\ a_{24} \end{pmatrix} \end{array}$$

Количество решений в ступенчатом виде Если во время прямого хода алгоритма Гаусса в расширенной матрице системы вам встретилась строка вида $(0 \dots 0 \mid b)$, где b – произвольное ненулевое число, то данная система решений не имеет. В этом случае нет необходимости переходить к обратному ходу. Если же таких строк не встретилось, то система обязательно имеет решения. При этом, если есть свободные переменные, то решений бесконечное число, а если их нет, то решение единственное.

Технические рекомендации Работая с целочисленными матрицами, старайтесь во время прямого хода алгоритма Гаусса не выходить за рамки целых чисел.

- Используйте элементарные преобразования I типа только с целым параметром.
- Полезно не злоупотреблять умножением на ненулевое целое, умножайте только на ± 1 . Иначе придется работать с большими числами.

На этапе обратного хода алгоритма Гаусса избавиться от деления уже не возможно.

⁵Операция умножения матрицы на число покомпонентная (умножаем каждый элемент на число). Сумма и разность двух матриц покомпонентная (складываем или вычитаем числа на одних и тех же позициях).