

计算方法

Computational Methods

Lecture 6: 线性方程组、算子范数与条件数

Linear Systems, Operator Norms, and Conditioning

刘景钺

南京大学 · 计算机学院

课程信息

计算方法 Fall 2026

授课教师 刘景铎 liu@nju.edu.cn

上课 周四 5-6 节 · 1-18 周 · 仙 I-319

答疑 周四下午 4 点

助教 周宇恒 · 侯哲

作业 nm_nju_2026@163.com

课程主页 TCS Wiki · Fall 2026

QQ 群 1054896504

加入时注明姓名、专业、学号

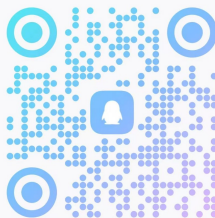
教材: Timothy Sauer, 《数值分析》(原书第 2 版)。

最新公告以课程主页为准。



计算方法 Fall 2026

群号: 1054896504



扫一扫二维码, 加入群聊



从第五讲到今天：一般线性系统还有一个更难的问题

第五讲里，结构化矩阵告诉我们：**选对表示，可以让计算快很多**。对于一般稠密矩阵，最通用的起点仍然是

$$Ax = b \xrightarrow{\text{高斯消元}} \text{上三角系统.}$$

LU 分解以及需要换行时出现的置换矩阵，接下来再正式写出。

先解决：怎么算？

消元、 LU 、选主元，把一般系统化
成三角系统。

算法结构

但今天真正要追问：答案可信吗？

输入稍微变化，精确解会变化多少？浮
点算法又额外带来多少误差？

条件性 + 稳定性

第一讲的主线重新出现：**后向误差** $\times$ **条件数** $\implies$ **前向误差**.

高斯消元：用等价行变换逐列消去

给定方阵 $A \in \mathbb{R}^{n \times n}$ 、 $b \in \mathbb{R}^n$ 。消元可以按行或按列组织；这里采用最常见的**逐列主元**描述。用第 k 行消去下面元素时，核心操作是

$$R_i \leftarrow R_i - \underbrace{\ell_{ik}}_{\text{消元乘子}} R_k,$$

必要时先**交换两行**选择合适的主元。两种操作都不会改变方程组的解集。每一步都可以写成左乘一个可逆矩阵 E ：

$$A \mapsto EA, \quad b \mapsto Eb,$$

所以

$$\boxed{Ax = b \iff EAx = Eb.}$$

一行乘非零常数也是等价行变换；但标准 LU 消元并不需要把主元行先归一化。

算法改变的是表示，不是解集。

消元的“历史”就是 $L: A = LU$

先假设过程中不需要换行。消元把 A 变成上三角矩阵 U :

$$U = E_k \cdots E_2 E_1 A.$$

因此

$$\boxed{A = LU}, \quad L := (E_k \cdots E_2 E_1)^{-1}.$$

L : 记录消元乘子

$$L = \begin{pmatrix} 1 & 0 & 0 \\ \ell_{21} & 1 & 0 \\ \ell_{31} & \ell_{32} & 1 \end{pmatrix}.$$

U : 上三角系统

$$Ax = b \Rightarrow Ly = b, \quad Ux = y.$$

前代、回代各需 $O(n^2)$ 。

一次 LU 分解，可以复用于很多不同的右端项 b 。

稠密矩阵分解约需 $O(n^3)$ 次算术运算；已有分解后，每个新右端项只需 $O(n^2)$ 。

一个 2×2 例子：消元乘子怎样变成 L

考虑

$$A = \begin{pmatrix} 2 & 1 \\ 4 & 3 \end{pmatrix}.$$

用第一行消去第二行：

$$R_2 \leftarrow R_2 - \underbrace{2}_{\ell_{21}} R_1 \quad \Rightarrow \quad U = \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix}.$$

把刚才使用的消元乘子放回下三角矩阵：

$$L = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}, \quad \boxed{A = LU}.$$

L 记录“怎样消掉下面的元素”； U 记录“消完以后剩下什么”。

选主元：控制消元乘子，而不只是避免 0

在当前第 k 列，用主元 $p = a_{kk}$ 消去下面的 a_{ik} 时，乘子是

$$\ell_{ik} = \frac{a_{ik}}{p}.$$

小主元只有**相对同列待消去元素过小时**才危险：此时乘子过大，舍入误差容易被放大。

部分选主元（常用）

在当前列剩余行中选择绝对值最大的元素；用**置换矩阵** P 记录行交换：

$$PA = LU.$$

于是每一步都有 $|\ell_{ik}| \leq 1$.

完全选主元

还允许交换列；**置换矩阵** 记为 Q ：

$$PAQ = LU.$$

若当前剩余子矩阵找不到非零主元，则方阵必为奇异矩阵。

列交换会重排未知量，求解后须按 Q 还原； $|\ell_{ik}| \leq 1$ 控制了乘子，**但不单独保证数值稳定。**

一个很小的主元，为什么会放大舍入误差？

考虑增广矩阵 ($\varepsilon = 0$ 时第一步必须换行；现在看 $0 < \varepsilon \ll 1$ 为什么也危险)

$$\left(\begin{array}{cc|c} \varepsilon & 1 & 1 \\ 1 & 1 & 2 \end{array} \right), \quad 0 < \varepsilon \ll 1.$$

若不换行，第一步乘子是 $\ell_{21} = 1/\varepsilon$ ，于是

$$R_2 \leftarrow R_2 - \frac{1}{\varepsilon} R_1 \quad \Rightarrow \quad \left(\begin{array}{cc|c} \varepsilon & 1 & 1 \\ 0 & 1 - \frac{1}{\varepsilon} & 2 - \frac{1}{\varepsilon} \end{array} \right).$$

原来都是 $O(1)$ 的数据，中间量却膨胀到 $O(1/\varepsilon)$ ；同时，主元行里已有的微小舍入误差也会被乘上 $1/\varepsilon$ 。

交换两行以后，第一个主元变成 1，乘子只有 ε ，中间量仍保持 $O(1)$ 。

选主元控制的不仅是“能不能除”，还有计算过程中的中间量与误差放大。

直接法给出 $\hat{x}$ 以后，我们真正关心什么？

设精确问题是

$$Ax = b,$$

计算机返回 $\hat{x}$ 。有两个自然问题：

前向误差

$$\hat{x} - x$$

答案离真正的解有多远？

这最直接，但通常不知道 x 。

残差

$$r = b - A\hat{x}$$

把 $\hat{x}$ 代回原方程，差了多少？

可以代回原式计算其近似值；计算残差本身也会受到舍入影响。

一个小残差，是否一定意味着一个小前向误差？

第一讲的框架重新出现：条件性与算法误差要分开

第一讲已经给出数值分析最重要的一条主线：

$$\text{前向误差} \lesssim \text{条件数} \times \text{后向误差}$$

今天把它落实到线性系统

$$b \mapsto x = A^{-1}b.$$

需要依次回答：

1. 怎样衡量一个线性映射“最多能放大多少”？
2. b 的小相对扰动，最坏会把 x 的相对误差放大多少？

这会自然逼出算子范数和矩阵条件数。

回顾：向量范数是一把“长度/误差”的尺子

第三讲已经见过常见向量范数：

$$\|x\|_1 = \sum_i |x_i|, \quad \|x\|_2 = \sqrt{\sum_i x_i^2}, \quad \|x\|_\infty = \max_i |x_i|.$$

同一个向量空间可以有不同的“长度”；**选择范数，就是选择怎样衡量误差。**

现在矩阵 $A \in \mathbb{R}^{m \times n}$ 不再只是 mn 个数，而是一个线性映射

$$A : \mathbb{R}^n \rightarrow \mathbb{R}^m, \quad x \mapsto Ax.$$

因此最自然的问题是：A 最多能把向量拉长多少倍？

两种“矩阵大小”：逐项大小与线性映射的放大倍数

一个直接的想法是把矩阵摊平成向量：

$$\|A\|_F := \sqrt{\sum_{i,j} a_{ij}^2} \quad (\text{Frobenius 范数}).$$

它很有用，但今天研究的是 $Ax = b$ 的扰动传播，所以更关心 A 怎样作用在向量上。

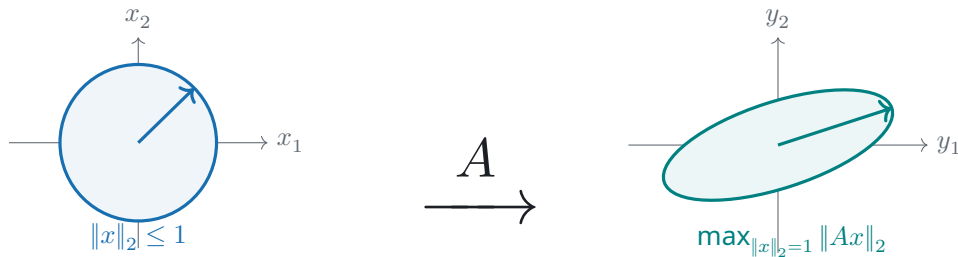
给定输入、输出空间上的向量范数，定义**算子范数 (operator norm)**

$$\|A\|_{\text{in} \rightarrow \text{out}} := \max_{x \neq 0} \frac{\|Ax\|_{\text{out}}}{\|x\|_{\text{in}}} = \max_{\|x\|_{\text{in}}=1} \|Ax\|_{\text{out}}.$$

若输入、输出都用同一个 p -范数，就简写成 $\|A\|_p$ ；一般的 $p \rightarrow q$ 记号是同一个定义。

它直接回答：“最坏方向上， A 会把长度放大多少倍？”

2-算子范数的几何：看单位圆被 A 拉成什么形状



当输入、输出都用 2-范数时，单位圆经线性映射变成椭圆（若 A 不可逆，则可能退化）； $\|A\|_2$ 就是最长半轴的长度。

如何从矩阵条目算出最长半轴？等正式引入特征值 / SVD 后再回答；这里先保留几何意义。

定义立刻给出三个重要性质

对向量范数诱导的算子范数（复合时，前一步的输出范数须与后一步的输入范数一致）：

1. 作用在向量上

$$\|Ax\| \leq \|A\| \|x\|.$$

2. 复合线性映射

$$\|AB\| \leq \|A\| \|B\|.$$

3. 恒等映射：当输入、输出使用同一个范数时

$$\|I\| = 1.$$

复合变换的放大倍数，至多为两步放大倍数上界的乘积。

1-算子范数：看“最大列绝对值和”

定义

$$\|A\|_1 := \max_{x \neq 0} \frac{\|Ax\|_1}{\|x\|_1}.$$

把 A 的列写成 $A_1, \dots, A_n$, 则

$$Ax = x_1 A_1 + \dots + x_n A_n.$$

因此

$$\|Ax\|_1 \leq \sum_j |x_j| \|A_j\|_1 \leq \left(\max_j \|A_j\|_1 \right) \|x\|_1.$$

而取第 j_* 个标准基向量 $x = e_{j_*}$ 可以达到这个上界, 所以

$$\|A\|_1 = \max_j \sum_i |a_{ij}|.$$

结论：最大的“列绝对值和”。

∞ -算子范数：看“最大行绝对值和”

同理，

$$\|A\|_{\infty} := \max_{x \neq 0} \frac{\|Ax\|_{\infty}}{\|x\|_{\infty}}.$$

对任意第 i 个分量，

$$|(Ax)_i| = \left| \sum_j a_{ij} x_j \right| \leq \sum_j |a_{ij}| \|x\|_{\infty}.$$

这个上界可以通过适当选择 x_j 的符号达到（构造见补充），因此

$$\|A\|_{\infty} = \max_i \sum_j |a_{ij}|.$$

$\|A\|_1$ ：最大列和

$\|A\|_{\infty}$ ：最大行和

精确取等的构造放在补充材料；今天更重要的是“算子范数 = 最坏方向放大”。

回到第一讲：条件数就是相对误差的最坏放大

固定可逆矩阵 A ，用同一种向量范数衡量 b 与 x 的相对误差：

$$b \mapsto x = A^{-1}b, \quad \Delta x = A^{-1}\Delta b.$$

第一讲的**条件数**，就是相对解误差相对于相对输入误差的最坏放大：

$$\text{cond}(A) := \max_{b \neq 0, \Delta b \neq 0} \frac{\|\Delta x\|/\|x\|}{\|\Delta b\|/\|b\|}.$$

写成 $b = Ax$ 。基准方向 x 与扰动方向 Δb **可以独立选择**，于是

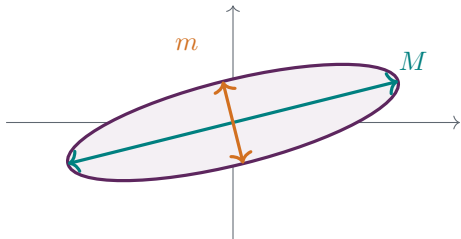
$$\max_{\Delta b \neq 0} \frac{\|\Delta x\|}{\|\Delta b\|} = \max_{\Delta b \neq 0} \frac{\|A^{-1}\Delta b\|}{\|\Delta b\|} = \|A^{-1}\|,$$

$$\max_{b \neq 0} \frac{\|b\|}{\|x\|} = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \|A\|.$$

$$\boxed{\text{cond}(A) = \|A\| \|A^{-1}\|} \quad (\text{也记作 } \kappa(A))$$

几何上: $\text{cond}_2(A)$ 就是“最长半轴 / 最短半轴”

以 2-范数为例, 单位圆经过可逆线性映射 A 后变成椭圆:



记 $M = \max_{\|x\|_2=1} \|Ax\|_2$, $m = \min_{\|x\|_2=1} \|Ax\|_2$. 由定义

$$\|A\|_2 = M, \quad \|A^{-1}\|_2 = \frac{1}{m}, \quad \boxed{\text{cond}_2(A) = \frac{M}{m}}.$$

条件数大 $\iff$ 椭圆很“扁” $\iff$ 存在几乎被 A 看不见的方向。

两个最简单的例子：同样可逆，敏感性完全不同

考虑 2-范数下的对角矩阵：

尺度相近

$$A = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}.$$

最大拉伸为 2，逆映射最大拉伸为 1：

$$\text{cond}_2(A) = 2.$$

一个方向几乎消失 ($0 < \varepsilon \ll 1$)

$$A = \begin{pmatrix} 1 & 0 \\ 0 & \varepsilon \end{pmatrix}.$$

逆映射必须把第二个方向放大 $1/\varepsilon$ ：

$$\text{cond}_2(A) = 1/\varepsilon.$$

“可逆”只回答能不能恢复；条件数回答恢复是否稳定。

$\varepsilon \rightarrow 0$ 时 $\text{cond}_2(A) \rightarrow \infty$ ； $\varepsilon = 0$ 时矩阵奇异，通常约定 $\text{cond}(A) = \infty$ 。

条件数的三个基本事实

对可逆方阵和诱导范数：

1. 至少为 1

$$1 = \|I\| = \|AA^{-1}\| \leq \|A\| \|A^{-1}\| = \text{cond}(A).$$

2. 整体缩放不改变条件数

$$\boxed{\text{cond}(cA) = \text{cond}(A)} \quad (c \neq 0).$$

3. 2-范数下的正交矩阵条件数为 1

$$Q^{\top}Q = I \quad \Rightarrow \quad \|Qx\|_2 = \|x\|_2 \quad \Rightarrow \quad \boxed{\text{cond}_2(Q) = 1}.$$

第四讲为什么一直偏爱正交变换？这里终于有了一个定量答案。

残差其实就是一个“附近问题”的输入扰动

计算机给出 $\hat{x}$ ，定义残差

$$r = b - A\hat{x}.$$

把式子重排：

$$A\hat{x} = b - r.$$

所以 $\hat{x}$ 是下面这个**附近线性系统**的精确解：

$$A\hat{x} = \underbrace{(b - r)}_{\text{把右端项扰动了 } -r}.$$

若固定 A 、只允许扰动 b ，要让 $\hat{x}$ 成为精确解只能取 $\Delta b = -r$ 。因此

$$\frac{\|r\|}{\|b\|} = \text{相对后向误差（只扰动 } b \text{）}.$$

从残差到前向误差：第一讲的框架变成精确不等式

设 x 是 $Ax = b$ 的精确解。上一页说明 $\Delta b = -r$ ，于是

$$\underbrace{\frac{\|\hat{x} - x\|}{\|x\|}}_{\text{前向误差}} \leq \underbrace{\text{cond}(A)}_{\text{问题的条件性}} \cdot \underbrace{\frac{\|r\|}{\|b\|}}_{\text{后向误差}}.$$

相对残差小

$$\|r\|/\|b\| \ll 1$$

说明计算结果是一个**附近问题**的精确解。

条件数适中

$\text{cond}(A)$ 不大

说明附近问题的精确解与原解**相差**
不远。

两件事同时成立，才能把“小残差”解释成“答案可信”。

小残差并不自动等于“答案很准”

令

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 1 + 10^{-6} \end{pmatrix}, \quad x = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad b = Ax = \begin{pmatrix} 2 \\ 2 + 10^{-6} \end{pmatrix}.$$

现在取候选解

$$\hat{x} = x + \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}.$$

它已经有 100% 的相对解误差：

$$\frac{\|\hat{x} - x\|_2}{\|x\|_2} = 1.$$

但沿 $(1, -1)^\top$ ，矩阵几乎“看不见”变化：

$$A(\hat{x} - x) = \begin{pmatrix} 0 \\ -10^{-6} \end{pmatrix}, \quad \boxed{\frac{\|b - A\hat{x}\|_2}{\|b\|_2} \approx 3.5 \times 10^{-7}.}$$

残差看起来精确到 10^{-7} ，答案本身却已经错了一个常数量级。

回扣第四讲：法线方程为什么会把“弱方向”变得更弱？

先看一个现在已经能完全量化的例子：

$$A = \begin{pmatrix} 1 & 0 \\ 0 & \varepsilon \end{pmatrix}, \quad 0 < \varepsilon \ll 1.$$

第二个方向经过 A 后只剩 ε 倍，因此

$$\text{cond}_2(A) = \frac{1}{\varepsilon}.$$

而法线方程使用

$$A^\top A = \begin{pmatrix} 1 & 0 \\ 0 & \varepsilon^2 \end{pmatrix}, \quad \boxed{\text{cond}_2(A^\top A) = \frac{1}{\varepsilon^2} = \text{cond}_2(A)^2.}$$

第四讲的“弱方向被平方放大”，现在有了**条件数的定量解释**。

这一平方关系对任意可逆矩阵也成立；一般证明留待 SVD。

条件性与数值稳定性：两个问题，不要混在一起

条件性：问题的性质

输入有一个很小的相对扰动，**精确解**会改变多少？

$$\text{cond}(A)$$

和你选择哪一种算法无关。

后向稳定性：算法的性质

浮点计算得到的结果，是否可以解释成**附近输入的精确答案**？

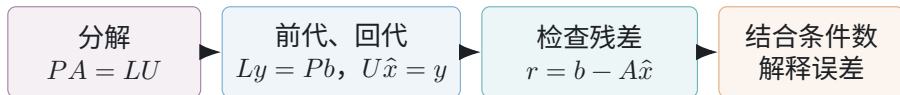
后向误差是否足够小？

取决于具体的计算路径。

好算法不能治愈一个病态问题；但坏算法可以毁掉一个条件良好的问题。

残差提供的是**某个答案**的后验误差检查；分析**整个算法**的稳定性时，通常还允许 A 受到微小扰动。

一个实际求解器的思维流程



实际高性能软件会把 LU 的核心更新组织成块矩阵运算，并调用优化过的 BLAS/LAPACK；但从数学上看，答案是否可信仍然落在两件事上：

后向误差是否小？

和

条件数是否可接受？

实际不必显式求 A^{-1} ：可利用分解**估计**条件数。残差检查也不能替代对整个算法稳定性的分析。

本讲主线：从“解方程”到“解释答案是否可信”

1. **一般矩阵：**高斯消元配合选主元，得到 $PA = LU$ ；控制乘子与中间量。

2. **矩阵作为线性映射：**

$$\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|}$$

衡量最坏方向的放大。

3. **线性系统的条件数：**

$$\text{cond}(A) = \|A\| \|A^{-1}\|$$

衡量输入相对误差到解相对误差的最坏放大。

4. **残差 + 条件数：**

$$\frac{\|\hat{x} - x\|}{\|x\|} \leq \text{cond}(A) \frac{\|b - A\hat{x}\|}{\|b\|}.$$

下一讲：如果 $O(n^3)$ 太贵呢？

高斯消元很通用，但大规模问题常常有额外结构：

- ▶ A 很**稀疏**，只有 $O(n)$ 或接近线性的非零元素；直接消元却可能把许多 0 变成非零；
- ▶ 已经有一个不错的初始猜测，或者 A, b 随时间缓慢变化；
- ▶ 只需要一个足够准确的近似解。

这时自然想到：不要一次性完成消元，而是反复做便宜的更新

$$x_{k+1} = Gx_k + c.$$

什么时候这样的迭代会收敛？

下一讲：**Jacobi**、**Gauss-Seidel** 与不动点收敛。

更一般的特征值判据，等正式引入特征值后再回来统一解释。

迭代改进：已经有 LU ，还能继续修正答案

设当前近似解是 $\hat{x}$ ，残差为

$$r = b - A\hat{x}.$$

真正的误差 $e = x - \hat{x}$ 满足

$$Ae = b - A\hat{x} = r.$$

所以可以用已有的浮点 LU 因子近似求解修正方程：

$$Ad \approx r, \quad \hat{x} \leftarrow \hat{x} + d.$$

迭代改进 (iterative refinement)：反复“算残差 $\rightarrow$ 解修正方程 $\rightarrow$ 更新”。

若修正方程精确求解，一步即可得到真解；需要迭代是因为 LU 分解和三角求解也有舍入误差。
高精度计算残差是常用的改进。

从逐行解方程到雅可比迭代 (Jacobi)

不先做完整的 LU 分解：只要 $a_{ii} \neq 0$ ，第 i 个方程就可以改写为

$$x_i = \frac{b_i - \sum_{j \neq i} a_{ij} x_j}{a_{ii}}.$$

这启发一种**不动点迭代**：用上一轮的估计代替右侧未知量，得到

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j \neq i} a_{ij} x_j^{(k)} \right), \quad i = 1, \dots, n.$$

雅可比迭代的规则

同一轮全部使用旧值 $x^{(k)}$ ，算完后统一写入 $x^{(k+1)}$ 。

为什么便宜？

每步只涉及各行的乘加和一次对角除法；
稀疏矩阵每轮通常很省。

先定义算法；它**未必收敛**，更不会因为每步便宜就一定更快。

雅可比迭代：同样的更新，有时收敛，有时发散

均从 $x^{(0)} = (0, 0)^\top$ 出发；以下每轮都用**上一轮**的两个坐标。

收敛的例子

$$\begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} x = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad x_* = \begin{pmatrix} 2/3 \\ 1/3 \end{pmatrix}.$$

$$x_1^{(k+1)} = \frac{1 + x_2^{(k)}}{2}, \quad x_2^{(k+1)} = \frac{x_1^{(k)}}{2}.$$

k	$x^{(k)}$
0	(0, 0)
1	(1/2, 0)
2	(1/2, 1/4)
3	(5/8, 1/4)

发散的例子

$$\begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} x = \begin{pmatrix} 3 \\ 3 \end{pmatrix}, \quad x_* = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

$$x_1^{(k+1)} = 3 - 2x_2^{(k)}, \quad x_2^{(k+1)} = 3 - 2x_1^{(k)}.$$

k	$x^{(k)}$
0	(0, 0)
1	(3, 3)
2	(-3, -3)
3	(9, 9)

雅可比迭代的同一轮内，各行计算互不依赖。若按 $i = 1, 2, \dots, n$ 顺序更新，**已经算出的新坐标就立刻用上**，得到

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j < i} a_{ij} x_j^{(k+1)} - \sum_{j > i} a_{ij} x_j^{(k)} \right).$$

写成矩阵形式：令 $A = D + L + U$ ，其中 D 是对角部分， L 和 U 分别是严格下、上三角部分，

$$\text{Jacobi: } Dx^{(k+1)} = b - (L + U)x^{(k)},$$

$$\text{Gauss-Seidel: } (D + L)x^{(k+1)} = b - Ux^{(k)}.$$

雅可比：同一轮同时更新

高斯-赛德尔：同一轮依次更新

同一个线性系统：雅可比和高斯-赛德尔比较

回到收敛的例子 $2x_1 - x_2 = 1$ 、 $-x_1 + 2x_2 = 0$ ，取 $x^{(0)} = (0, 0)^\top$ 。

$$\text{精确解 } x_* = \left(\frac{2}{3}, \frac{1}{3}\right)^\top.$$

雅可比：两式都用旧值

$$\begin{cases} x_1^{(k+1)} = (1 + x_2^{(k)})/2, \\ x_2^{(k+1)} = x_1^{(k)}/2. \end{cases}$$

k	$x^{(k)}$
1	$(1/2, 0)$
2	$(1/2, 1/4)$
3	$(5/8, 1/4)$

高斯-赛德尔：第二式用新值

$$\begin{cases} x_1^{(k+1)} = (1 + x_2^{(k)})/2, \\ x_2^{(k+1)} = x_1^{(k+1)}/2. \end{cases}$$

k	$x^{(k)}$
1	$(1/2, 1/4)$
2	$(5/8, 5/16)$
3	$(21/32, 21/64)$

什么时候收敛？回到第一讲的不动点与范数

两种算法都可以写成 $x^{(k+1)} = Gx^{(k)} + c$ ；若 x_* 是精确解（也是不动点），令 $e^{(k)} = x^{(k)} - x_*$ ，则

$$e^{(k+1)} = Ge^{(k)}, \quad e^{(k)} = G^k e^{(0)}.$$

因此如果在某个**诱导算子范数**下 $\|G\| < 1$,

$$\|e^{(k)}\| \leq \|G\|^k \|e^{(0)}\| \rightarrow 0.$$

分别是哪一个矩阵？

$$G_J = -D^{-1}(L + U), \quad G_{GS} = -(D + L)^{-1}U.$$

充分但非必要；准确判据将在正式引入特征值后给出。

若 A 在每一行都**严格对角占优**：

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}| \quad (i = 1, \dots, n),$$

那么

$$\|G_J\|_{\infty} = \max_i \frac{\sum_{j \neq i} |a_{ij}|}{|a_{ii}|} =: q < 1.$$

因此 $\|e^{(k)}\|_{\infty} \leq q^k \|e^{(0)}\|_{\infty} \rightarrow 0$ 。

顺便得到：严格对角占优的方阵一定可逆。

否则 $Ax = 0$ 有非零解。选 $|x_i| = \|x\|_{\infty} > 0$ ，由第 i 行得

$$|a_{ii}| \|x\|_{\infty} \leq \left(\sum_{j \neq i} |a_{ij}| \right) \|x\|_{\infty} < |a_{ii}| \|x\|_{\infty},$$

矛盾。

仍令 $e^{(k)} = x^{(k)} - x_*$ 。由逐行更新式得到

$$|e_i^{(k+1)}| \leq \frac{1}{|a_{ii}|} \left(\sum_{j<i} |a_{ij}| |e_j^{(k+1)}| + \sum_{j>i} |a_{ij}| |e_j^{(k)}| \right).$$

$$\text{令 } M = \|e^{(k)}\|_\infty, \quad q = \max_i \sum_{j \neq i} |a_{ij}|/|a_{ii}| < 1.$$

沿 $i = 1, 2, \dots, n$ 归纳。

前面已更新的坐标满足 $|e_j^{(k+1)}| \leq qM$ ，尚未更新的坐标满足 $|e_j^{(k)}| \leq M$ 。所以

$$|e_i^{(k+1)}| \leq \frac{q \sum_{j<i} |a_{ij}| + \sum_{j>i} |a_{ij}|}{|a_{ii}|} M \leq qM.$$

$\|e^{(k+1)}\|_\infty \leq q\|e^{(k)}\|_\infty$ ：从而高斯-赛德尔迭代收敛。

考虑

$$(A + \Delta A)(x + \Delta x) = b + \Delta b.$$

若扰动足够小, 使 $\|A^{-1}\| \|\Delta A\| < 1$, 可以得到标准估计

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{\text{cond}(A)}{1 - \text{cond}(A) \|\Delta A\|/\|A\|} \left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|} \right).$$

一阶近似下仍然是同一个故事:

$$\text{相对解误差} \lesssim \text{cond}(A) \times \text{相对输入扰动}.$$

我们已经有上界

$$\|Ax\|_{\infty} \leq \left(\max_i \sum_j |a_{ij}| \right) \|x\|_{\infty}.$$

取达到最大行和的某一行 i_* ，并令

$$x_j = \begin{cases} \operatorname{sgn}(a_{i_*j}), & a_{i_*j} \neq 0, \\ 1, & a_{i_*j} = 0. \end{cases}$$

那么 $\|x\|_{\infty} = 1$ ，而

$$(Ax)_{i_*} = \sum_j a_{i_*j} x_j = \sum_j |a_{i_*j}|.$$

因此

$$\boxed{\|A\|_{\infty} = \max_i \sum_j |a_{ij}|.}$$

交换两行

$$A \mapsto PA$$

P 为置换矩阵。

重排方程

一行乘常数

$$A \mapsto DA$$

D 为可逆对角矩阵。

重新缩放方程

消去一个元素

$$R_j \leftarrow R_j - cR_i, \quad i \neq j$$

$$A \mapsto (I - ce_j e_i^\top) A$$

其逆为 $I + ce_j e_i^\top$ 。

初等消元矩阵

旧课件还提到一个更一般的观点：矩阵范数不一定非要由向量范数诱导。
若矩阵范数 $\|\cdot\|_M$ 与所用向量范数相容，例如满足

$$\|Ax\| \leq \|A\|_M \|x\|,$$

一些教材也定义

$$\text{cond}_M(A) := \|A\|_M \|A^{-1}\|_M.$$

它同样给出扰动上界

$$\frac{\|\Delta x\|}{\|x\|} \leq \text{cond}_M(A) \frac{\|\Delta b\|}{\|b\|}.$$

诱导算子范数的优势：它直接等于线性映射的最坏放大，因此得到的是与所选向量误差尺度匹配的最自然、最紧的最坏情形放大。

核心定义已经允许输入、输出使用不同的“尺子”：

$$\|A\|_{p \rightarrow q} = \max_{x \neq 0} \frac{\|Ax\|_q}{\|x\|_p}.$$

例如

$$\|A\|_{1 \rightarrow \infty} = \max_{i,j} |a_{ij}|.$$

因为对每个 i ,

$$|(Ax)_i| \leq \sum_j |a_{ij}| |x_j| \leq \left(\max_j |a_{ij}| \right) \|x\|_1,$$

而取到最大元素所在列的标准基向量即可达到上界。

同一个矩阵，用不同输入/输出误差尺度衡量，会得到不同的“放大倍数”。