线性方程组的直接解法

- 线性方程组求解是一个基本的数学问题
  - 挑战: 方程的规模大，例如变量数>10^8
  - 困难: 存储量、计算时间、准确度
- 内容
  - 基本概念与问题敏感性
  - 高斯消去法
  - 矩阵的LU分解
  - 选主元技术与算法稳定性
  - 对称正定矩阵的Cholesky分解
  - 带状矩阵、稀疏矩阵

Cramer法则

n 元线性方程组可简写成矩阵形式 $Ax = b$

Cramer法则表明: 如果 n 阶线性方程组 $Ax = b$ 的系数矩阵 A 的行列式不为零, 则此方程组有唯一的解, 并且其解可以通过系数表示为

$$x_i = \frac{d_i}{d},\ i = 1, \cdots, n,$$

其中 $d = \det A, d_i = \det A_i$, 这里 A_i 是将 A 的第 i 列换为 b 而得到的矩阵. 它把线性方程组的求解问题归结为计算 $n+1$ 个 n 阶行列式的问题.

对于行列式的计算, 理论上有著名的 Laplace 展开定理:

$$d = \det A = a_{i1}A_{i1} + a_{i2}A_{i2} + \cdots + a_{in}A_{in},$$

其中 A_{ij} 表示元素 a_{ij} 的代数余子式. 按照这一定理就可从二阶行列式出发逐步递推地计算出任意阶行列式的值. 这样, 理论上就有了一种非常漂亮的求解线性方程组的方法.

设计算 k 阶行列式所需要的乘法运算的次数为 m_k, 则容易推出

$$m_k = k + k m_{k-1}$$

于是, $m_n = n + n m_{n-1} = n + n[(n-1) + (n-1) m_{n-2}] = \cdots$

$= n + n(n-1) + n(n-1)(n-2) + \cdots + n(n-1)\cdots 3\cdot 2 > n!$

这样, 利用 Cramer 法则和 Laplace 展开定理来求解一个 n 阶线性方程组, 所需要的乘法运算的次数就大于 $(n+1)n! = (n+1)!$.

因此, 若在一个百亿次计算机上求解一个 25 阶线性方程组, 则至少需要

$$\frac{26!}{10^{10}\times 3600\times 24\times 365} \approx \frac{4.0329\times 10^{26}}{3.1536\times 10^{17}} \approx 13 \text{ 亿 (年)},$$

如果改用将要介绍的消元法, 则在不到一秒钟之内即可完成这一计算任务.

按系数矩阵阶数的高低和含零元素的多少, 线性方程组可以分为两类:

(1) 低阶稠密方程组, 即系数矩阵阶数不高, 含零元素很少;

(2) 高阶稀疏方程组, 即系数矩阵阶数高, 零元素极多.

线性方程组的数值解法主要有两大类.

(1) 直接法

指理论上在没有舍入误差影响的条件下, 经过有限次四则运算可以求得方程组的准确解的一些方法. 但由于实际计算时舍入误差是不可避免的, 也只能求得近似解.

(2) 迭代法

用某种极限过程去逐步逼近方程组的准确解的一些方法. 这类方法编程较容易, 但要考虑迭代过程的收敛性、收敛速度等问题. 由于实际计算时, 只能做有限步, 得到的也是近似解. 当系数矩阵是高阶稀疏矩阵时, 一般优先考虑迭代法.

基本概念

线性方程组与矩阵的基本概念

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \cdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{cases}$$

- 若$m > n$, 超定方程组, 一般无解 (最小二乘解)
- 若$m < n$, 一般有无穷多解, 与其他条件构成优化问题
- 讨论$m = n$的情况，记为:

$$\boldsymbol{Ax} = \boldsymbol{b},\ \boldsymbol{A} \in \mathbb{R}^{n\times n}, \boldsymbol{b} \in \mathbb{R}^n$$

(假设实数矩阵、向量)

非奇异矩阵

矩阵 $A \in \mathbb{R}^{n\times n}$, 若它满足下列等价条件之一:

(1) 存在逆矩阵 $\boldsymbol{A}^{-1}$, 即存在一个矩阵, 记为 $\boldsymbol{A}^{-1}$, 满足 $\boldsymbol{AA} = \mathbf{A}^{-1}\boldsymbol{A} = \boldsymbol{I}$ 。

(2) $\det(\boldsymbol{A}) \neq 0$, 其中 $\det(\boldsymbol{A})$ 表示 $\boldsymbol{A}$ 的行列式 (determinant)。

(3) $\boldsymbol{A}$ 的秩 $\operatorname{rank}(\boldsymbol{A}) = n$ (矩阵的秩为其包含的线性无关的行或列的最多个数)。

(4) 对任意向量 $z \neq 0$, 有 $\boldsymbol{Az} \neq \mathbf{0}$ 。或者 由$\boldsymbol{Az} = \mathbf{0}$ 推出 向量 $\boldsymbol{z} = \mathbf{0}$

(5) $\boldsymbol{A}$ 的所有特征值都不等于0.

则矩阵 $\boldsymbol{A}$ 为**非奇异矩阵** (nonsingular matrix), 否则为**奇异矩阵** (singular matrix)。

线性方程组解的存在性与唯一性

$\boldsymbol{Ax} = \boldsymbol{b}, \boldsymbol{A} \in \mathbb{R}^{n\times n}, \boldsymbol{b} \in \mathbb{R}^n$

(1) 若 $\boldsymbol{A}$ 为非奇异矩阵, 则方程组 $\boldsymbol{Ax} = \boldsymbol{b}$ 有唯一的解 $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$。

(2) 若 $\boldsymbol{A}$ 为奇异矩阵, 且 $b \in \text{span}(\boldsymbol{A})$, 集合 $\text{span}(\boldsymbol{A})$ 表示 $\boldsymbol{A}$ 的各个列向量张成的线性空间, 则方程组 $\boldsymbol{Ax} = \boldsymbol{b}$ 有无穷多个解。

(3) 若 $\boldsymbol{A}$ 为奇异矩阵, 且 $\boldsymbol{b} \notin \text{span}(\boldsymbol{A})$, 则方程组 $\boldsymbol{Ax} = \boldsymbol{b}$ 没有解。

特殊矩阵

- 稀疏矩阵："×"标记非零矩阵元素的位置

$$\begin{bmatrix} \times & & & \\ & \times & & \\ & & \times & \\ & & & \times \end{bmatrix} \quad \begin{bmatrix} \times & \times & & \\ \times & \times & \times & \\ & \times & \times & \times \\ & & \times & \times \end{bmatrix} \quad \begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & & \times \end{bmatrix} \quad \begin{bmatrix} \times & & & \\ \times & \times & & \\ \times & \times & \times & \\ \times & \times & \times & \times \end{bmatrix}$$

表示稀疏矩阵的Wilkinson图

- 对称正定矩阵:

  若$\boldsymbol{A}^T=\boldsymbol{A}$, 且对任意非零向量$\boldsymbol{x}\in\mathbb{R}^n$, 二次型$\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}>0$

- 正交矩阵: $\boldsymbol{A}^{-1}=\boldsymbol{A}^T$, 矩阵行(列)构成一组单位正交基
- $\boldsymbol{A}=(a_{ij})\in\mathbb{R}^{n\times n}$, $\boldsymbol{A}$的k阶顺序主子阵$(k=1,2,\cdots,n)$

$$\boldsymbol{A}_k=\begin{bmatrix} a_{11} & \cdots & a_{1k} \\ \vdots & \ddots & \vdots \\ a_{k1} & \cdots & a_{kk} \end{bmatrix},$$

行列式$\det(\boldsymbol{A}_k)$称为顺序主子式

向量的范数

- 一般线性空间中范数的定义, 记为$\|\cdot\|$ (以$\mathbb{R}^n$为例)
  - $\forall \boldsymbol{x} \in \mathbb{R}^n, \|\boldsymbol{x}\| \geq 0$; 当且仅当$\boldsymbol{x} = \boldsymbol{0}$时, $\|\boldsymbol{x}\| = 0$; (正定性)
  - $\forall \alpha \in \mathbb{R}, \ \|\alpha \boldsymbol{x}\| = |\alpha| \cdot \|\boldsymbol{x}\|$; (正齐次性)
  - $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n, \|\boldsymbol{x} + \boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$. (三角不等式)

  数域$\mathbb{K}$上的线性空间S, 定义了范数, 称为赋范线性空间
- 向量的p-范数与内积范数
  - $\boldsymbol{x} = [x_1, x_2, \cdots, x_n]^T$, $\|\boldsymbol{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$, $p \geq 1$
  - 内积范数: 针对内积$\langle \boldsymbol{x}, \boldsymbol{y} \rangle$, 定义范数$\|\boldsymbol{x}\| = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$

    ($\langle \boldsymbol{x}, \boldsymbol{x} \rangle = \boldsymbol{x}^T \boldsymbol{x}$)

向量的范数

- 三种常用范数 　　常写作l_1-norm, l_2-norm, l_∞-norm
  - 1-范数：$\|\boldsymbol{x}\|_1 = \sum_{i=1}^{n}|x_i|$ 　　(曼哈顿范数)
  - 2-范数：$\|\boldsymbol{x}\|_2 = \sqrt{\sum_{i=1}^{n}|x_i|^2}$ 　　(欧氏范数=内积范数)
  - ∞-范数：$\|\boldsymbol{x}\|_\infty = \max_{1\le i\le n}|x_i|$ 　　(“最大”范数)

例子

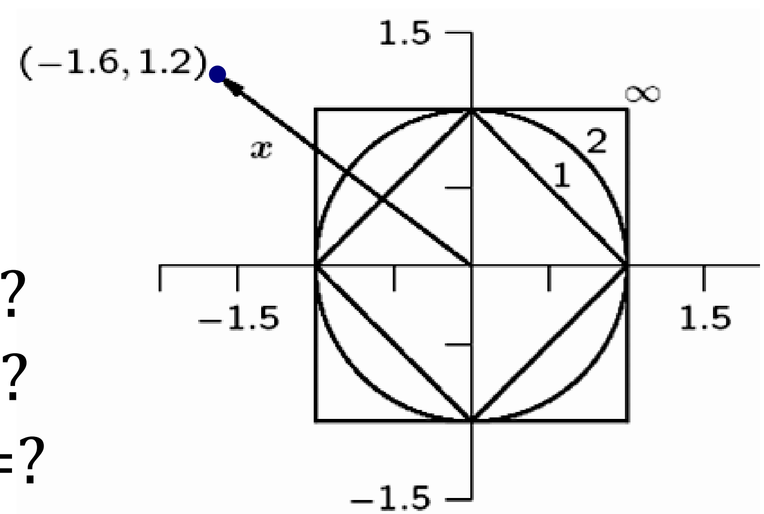
在二维坐标系中，绘出单位长度向量的端点集合(按三种不同范数)

若$\boldsymbol{x} = [-1.6, 1.2]^T$,

$\|\boldsymbol{x}\|_1 = ?$

$\|\boldsymbol{x}\|_2 = ?$

$\|\boldsymbol{x}\|_\infty = ?$



向量有关的定理

- **定义3.8:** 序列$\{\boldsymbol{x}^{(k)} \in \mathbb{R}^n\}$的极限$\lim\limits_{k\to\infty} \boldsymbol{x}^{(k)} = \boldsymbol{x}^*$的含义

  $\lim\limits_{k\to\infty} x_i^{(k)} = x_i^*$， i=1, 2, …, n
- **定理3.6** $\|\boldsymbol{x}\| = f(x_1, x_2, \cdots, x_n)$, f为连续函数
- **定理3.7** 不同范数的等价性：$c_1\|\boldsymbol{x}\|_s \le \|\boldsymbol{x}\|_t \le c_2\|\boldsymbol{x}\|_s$

  其中$c_1, c_2 > 0$为常数, 其值不依赖于具体$\boldsymbol{x}$
- **定理3.8** $\lim\limits_{k\to\infty} \boldsymbol{x}^{(k)} = \boldsymbol{x}^* \iff \lim\limits_{k\to\infty} \left\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\right\| = 0$
- 说明
  - 由定理3.7,在某种范数下的有些结论(如向量序列收敛到0)在其他范数下也成立
  - 除p-范数外的其他范数: 如$\|\boldsymbol{x}\|_A = \sqrt{\boldsymbol{x}^T A \boldsymbol{x}}$, A对称正定
  - 不严格的l_0-norm: $\|\boldsymbol{x}\|_0$

矩阵的范数

- 矩阵构成线性空间$\mathbb{R}^{n\times n}$
  - 矩阵加法、矩阵与实数乘法
  - “矩阵乘”运算

(不满足交换律)

$$\begin{bmatrix}\mathbf{0} & \mathbf{1}\\ \mathbf{3} & \mathbf{2}\end{bmatrix}\begin{bmatrix}\mathbf{0} & \mathbf{3}\\ \mathbf{1} & \mathbf{2}\end{bmatrix}=\begin{bmatrix}\mathbf{1} & \mathbf{2}\\ \mathbf{2} & \mathbf{13}\end{bmatrix}\qquad \begin{bmatrix}\mathbf{0} & \mathbf{3}\\ \mathbf{1} & \mathbf{2}\end{bmatrix}\begin{bmatrix}\mathbf{0} & \mathbf{1}\\ \mathbf{3} & \mathbf{2}\end{bmatrix}=?$$

一般矩阵相乘的条件: $\underset{m\times \boldsymbol{s}}{\boldsymbol{A}}\ \underset{\boldsymbol{s}\times n}{\boldsymbol{B}} = \underset{m\times n}{\boldsymbol{AB}}$ $\quad [\mathbf{1}\ \mathbf{2}]\begin{bmatrix}\mathbf{0}\\ \mathbf{3}\end{bmatrix}=?\quad \begin{bmatrix}\mathbf{0}\\ \mathbf{3}\end{bmatrix}[\mathbf{1}\ \mathbf{2}]=?$

- $\mathbb{R}^{n\times n}$空间的矩阵范数
  - 增加对矩阵乘法的要求: $\|\boldsymbol{AB}\| \le \|\boldsymbol{A}\|\|\boldsymbol{B}\|$
  - 矩阵常与向量相乘, 与向量范数相联系, 增加相容性条件: $\|\boldsymbol{Ax}\| \le \|\boldsymbol{A}\|\|\boldsymbol{x}\|$

矩阵的范数

若不加说明，默认算子范数

- 矩阵的**算子范数**（向量诱导范数）

  根据某种向量范数$\|\boldsymbol{x}\|_v$，定义矩阵范数：

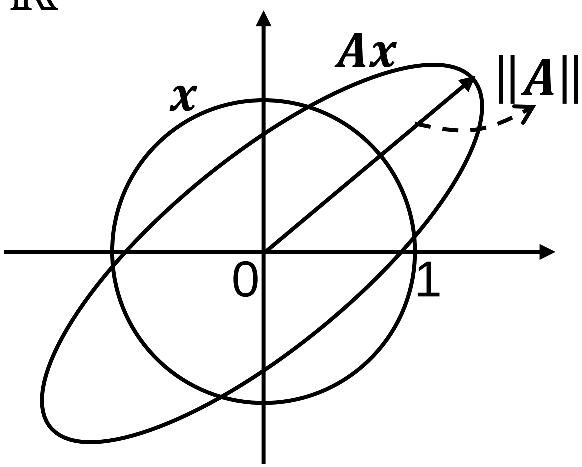
  $\|\boldsymbol{A}\|_v = \max\limits_{\boldsymbol{x}\neq\boldsymbol{0}} \frac{\|\boldsymbol{A}\boldsymbol{x}\|_v}{\|\boldsymbol{x}\|_v}$，其中$\boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{A} \in \mathbb{R}^{n\times n}$

  $\boldsymbol{A}\boldsymbol{x}$对向量$\boldsymbol{x}$的最大拉伸倍数(可能<1)

  形象描述: 不妨设$\|\boldsymbol{x}\|_v = 1$, $\boldsymbol{x}$的端点轨迹为二维空间中的单位圆

  $\boldsymbol{A}\boldsymbol{x}$向量端点的轨迹, 为椭圆

  $\|\boldsymbol{A}\|_v$为这个椭圆半长轴的长度



- 算子范数是满足相容性条件的范数
  - 满足一般范数定义;相容性条件;$\|\boldsymbol{AB}\| \le \|\boldsymbol{A}\|\|\boldsymbol{B}\|$

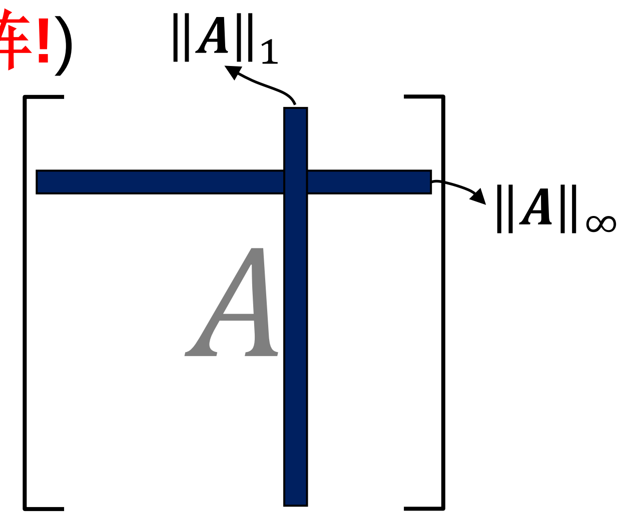
矩阵的范数

列范数、行范数

- 三种常用的矩阵范数 (不局限于方阵!)
  - 1-范数: $\|\boldsymbol{A}\|_1 = \max\limits_{1\le j\le n} \sum_{i=1}^{n} |a_{ij}|$
  - ∞-范数: $\|\boldsymbol{A}\|_\infty = \max\limits_{1\le i\le n} \sum_{j=1}^{n} |a_{ij}|$
  - 2-范数: $\|\boldsymbol{A}\|_2 = \sqrt{\lambda_{max}(\boldsymbol{A}^T\boldsymbol{A})}$,

    $\lambda_{max}(\cdot)$表示取矩阵的最大特征值

    2-范数的计算复杂, 常常避免计算
  - Frobenius范数: $\|\boldsymbol{A}\|_\mathrm{F} = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}^2}$



$\|\boldsymbol{A}\|_\mathrm{F}=\sqrt{\mathrm{tr}(\boldsymbol{A}^T\boldsymbol{A})}$

问题的敏感性

线性方程组求解问题的敏感性

$$\boldsymbol{Ax} = \boldsymbol{b} \Longrightarrow \boldsymbol{A}(\boldsymbol{x} + \Delta\boldsymbol{x}) = \boldsymbol{b} + \Delta\boldsymbol{b}$$

- 先考虑右端项发生扰动的情况，需求$\|\Delta\boldsymbol{x}\|/\|\boldsymbol{x}\|$与$\|\Delta\boldsymbol{b}\|/\|\boldsymbol{b}\|$的关系(不等式)
- 问题的条件数

$$\text{cond} = \frac{\|\Delta\boldsymbol{x}\|/\|\boldsymbol{x}\|}{\|\Delta\boldsymbol{b}\|/\|\boldsymbol{b}\|} = \frac{\|\Delta\boldsymbol{x}\|\|\boldsymbol{b}\|}{\cancel{\|\Delta\boldsymbol{b}\|}\cancel{\|\boldsymbol{x}\|}} \le \|\boldsymbol{A}\|\|\boldsymbol{A}^{-1}\|$$

$$\boldsymbol{A}\Delta\boldsymbol{x} = \Delta\boldsymbol{b} \Longrightarrow \Delta\boldsymbol{x} = \boldsymbol{A}^{-1}\Delta\boldsymbol{b} \Longrightarrow \|\Delta\boldsymbol{x}\| \le \|\boldsymbol{A}^{-1}\|\|\Delta\boldsymbol{b}\|$$

$$\boldsymbol{Ax} = \boldsymbol{b} \Longrightarrow \|\boldsymbol{b}\| \le \|\boldsymbol{A}\|\|\boldsymbol{x}\| \Longrightarrow \|\Delta\boldsymbol{x}\|\|\boldsymbol{b}\| \le \|\boldsymbol{A}^{-1}\|\cancel{\|\Delta\boldsymbol{b}\|}\|\boldsymbol{A}\|\cancel{\|\boldsymbol{x}\|}$$

- 与函数求值问题一样，条件数随输入数据不同而不同，上述结论说明: 其上限为矩阵$\boldsymbol{A}$与其逆的范数的乘积

矩阵的条件数

- 设$\boldsymbol{A}$为非奇异矩阵，

$$\operatorname{cond}(\boldsymbol{A}) = \|\boldsymbol{A}\|\left\|\boldsymbol{A}^{-1}\right\|$$

称为矩阵$\boldsymbol{A}$的条件数

- 1.矩阵条件数是反映线性方程组求解问题的敏感性的条件数上界

  2.其值随矩阵范数的不同而变化

  3.若考虑矩阵发生扰动的情况，也得出问题的条件数上界近似为矩阵$\boldsymbol{A}$的条件数

  4.矩阵条件数大则问题很病态, 也称该矩阵为病态矩阵

Matlab命令: norm, inv, cond

矩阵的条件数

例 求解方程 $\boldsymbol{Ax}=\boldsymbol{b}$： $\begin{bmatrix}1 & 1\\ 1 & 1.0001\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}=\begin{bmatrix}2\\ 2\end{bmatrix}$

考虑右端项扰动 $\Delta\boldsymbol{b}=\begin{bmatrix}0\\ 0.0001\end{bmatrix}$ 对解的影响

解: 原方程的解为 $\boldsymbol{x}=\begin{bmatrix}2\\ 0\end{bmatrix}$, 受扰动后, 解为 $\boldsymbol{x}+\Delta\boldsymbol{x}=\begin{bmatrix}1\\ 1\end{bmatrix}$

$\Delta\boldsymbol{x}=\begin{bmatrix}-1\\ 1\end{bmatrix}$

∞范数意义下

$$\mathrm{cond}=\frac{\|\Delta\boldsymbol{x}\|/\|\boldsymbol{x}\|}{\|\Delta\boldsymbol{b}\|/\|\boldsymbol{b}\|}=\frac{1/2}{0.0001/2}=10000$$

$$\mathrm{cond}(\boldsymbol{A})_\infty=\|\boldsymbol{A}\|_\infty\left\|\boldsymbol{A}^{-1}\right\|_\infty$$

$$=2.0001\times\left\|10^4\times\begin{bmatrix}1.0001 & -1.0000\\ -1.0000 & 1.0000\end{bmatrix}\right\|_\infty$$

$$=2.0001\times10^4\times2.0001\approx40000$$

符合“上界”的说法

矩阵的条件数

良态问题！

例 求解方程$\boldsymbol{Ax}=\boldsymbol{b}$：

$$\begin{bmatrix}1 & -1\\ 1 & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}=\begin{bmatrix}0\\ 2\end{bmatrix}$$

考虑右端项扰动$\Delta\boldsymbol{b}=\begin{bmatrix}0\\ 0.0001\end{bmatrix}$对解的影响

解: 类似前一个例子, 算出 $\text{cond}_\infty=1$, $\text{cond}(\boldsymbol{A})_\infty=2$

例 Hilbert矩阵

$$\boldsymbol{H}_n=\begin{bmatrix}1 & \frac{1}{2} & \cdots & \frac{1}{n}\\ \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n+1}\\ \vdots & \vdots & \ddots & \vdots\\ \frac{1}{n} & \frac{1}{n+1} & \cdots & \frac{1}{2n-1}\end{bmatrix}$$

$\text{cond}(\boldsymbol{H}_3)_\infty=748$

$\text{cond}(\boldsymbol{H}_4)_\infty=28375$

随n增大，$\boldsymbol{H}_n$变得很病态!

矩阵条件数的性质

- 在任一算子范数下:

$$\mathrm{cond}(\boldsymbol{A}) = \max_{\boldsymbol{x}\neq 0}\frac{\|\boldsymbol{A}\boldsymbol{x}\|}{\|\boldsymbol{x}\|}\Big/\min_{\boldsymbol{x}\neq 0}\frac{\|\boldsymbol{A}\boldsymbol{x}\|}{\|\boldsymbol{x}\|}$$

证明:

$$\|\boldsymbol{A}^{-1}\| = \max_{\boldsymbol{y}\neq 0}\frac{\|\boldsymbol{A}^{-1}\boldsymbol{y}\|}{\|\boldsymbol{y}\|} = \max_{\boldsymbol{x}\neq 0}\frac{\|\boldsymbol{x}\|}{\|\boldsymbol{A}\boldsymbol{x}\|} = \frac{1}{\min_{\boldsymbol{x}\neq 0}\frac{\|\boldsymbol{A}\boldsymbol{x}\|}{\|\boldsymbol{x}\|}}$$

…… ……

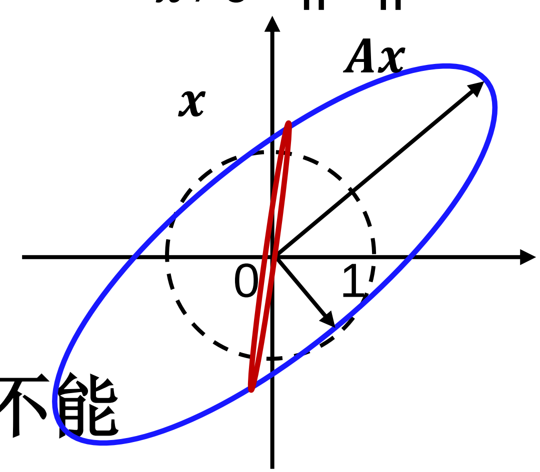
说明: 1. 反映了$\mathrm{cond}(\boldsymbol{A})$的几何意义:

$\boldsymbol{A}\boldsymbol{x}$对单位圆的扭曲程度

2. $\mathrm{cond}(\boldsymbol{A}) \geq 1$

3. cond(奇异矩阵) = +∞,

4.条件数反映矩阵近奇异程度, 行列式则不能



矩阵条件数的性质

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

- 矩阵条件数满足如下性质:

(1) $\mathrm{cond}(\boldsymbol{A}) \geq 1$, $\mathrm{cond}(\boldsymbol{A}^{-1}) = \mathrm{cond}(\boldsymbol{A})$,

$\mathrm{cond}(c\boldsymbol{A}) = \mathrm{cond}(\boldsymbol{A}), \forall c \neq 0$ 与det($\boldsymbol{A}$)性质不同

(2) $\mathrm{cond}(\boldsymbol{I})=1$ (在p-范数下)

(3) 设$\boldsymbol{D}$为对角阵, $\mathrm{cond}(\boldsymbol{D}) = \dfrac{\max_i |d_{ii}|}{\min_i |d_{ii}|}$, d_{ii}为$\boldsymbol{D}$的对角元

(4) 采用2-范数, 采用一般范数时, 应该是≥

$$\mathrm{cond}(\boldsymbol{A})_2 = \sqrt{\frac{\lambda_{\max}(\boldsymbol{A}^T\boldsymbol{A})}{\lambda_{\min}(\boldsymbol{A}^T\boldsymbol{A})}}$$

(5) 若$\boldsymbol{Q}$为正交矩阵

$\mathrm{cond}(\boldsymbol{Q})_2 = 1$, 正交变换不改条件数!

$\mathrm{cond}(\boldsymbol{QA})_2 = \mathrm{cond}(\boldsymbol{AQ})_2 = \mathrm{cond}(\boldsymbol{A})_2$

高斯消去法

容易求解的方程组:**对角方程组**

$$
\begin{cases}
a_{11}x_1 & & & = b_1 \\
& a_{22}x_2 & & = b_2 \\
& & \cdots & \\
& & & a_{nn}x_n = b_n
\end{cases}
$$

$$
\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b} = \left(\frac{b_1}{a_{11}}, \frac{b_2}{a_{22}}, \cdots, \frac{b_n}{a_{nn}}\right)^{\mathrm{T}}
$$

时间复杂度： O(n)

容易求解的方程组:**下三角形方程组**

$$\begin{cases} a_{11}x_1 & = b_1 \\ a_{21}x_1 + a_{22}x_2 & = b_2 \\ \cdots & \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n & = b_n \end{cases}$$

$$x_1 = \frac{1}{a_{11}} b_1$$

$$x_2 = \frac{1}{a_{22}} (b_2 - a_{21}x_1)$$

$$x_3 = \frac{1}{a_{33}} (b_3 - a_{31}x_1 - a_{32}x_2)$$

$\cdots\cdots\cdots$

$$x_n = \frac{1}{a_{nn}} (b_n - a_{n1}x_1 - a_{n2}x_2 - \cdots - a_{n,n-1}x_{n-1})$$

前代法

时间复杂度： O(n^2)

容易求解的方程组:**上三角形方程组**

算法 求解上三角形方程组$Ux = b$的回代过程

输入: U, n, b

输出: $\boldsymbol{x}$

For $i = n, n-1, n-2, \cdots, 1$

If $u_{ii} = 0$ then 停止;

$x_i := b_i$;

For $j = n, n-1, \dots, i+1$

$x_i := x_i - u_{ij}x_j$;

End

$x_i := x_i / u_{ii}$

End

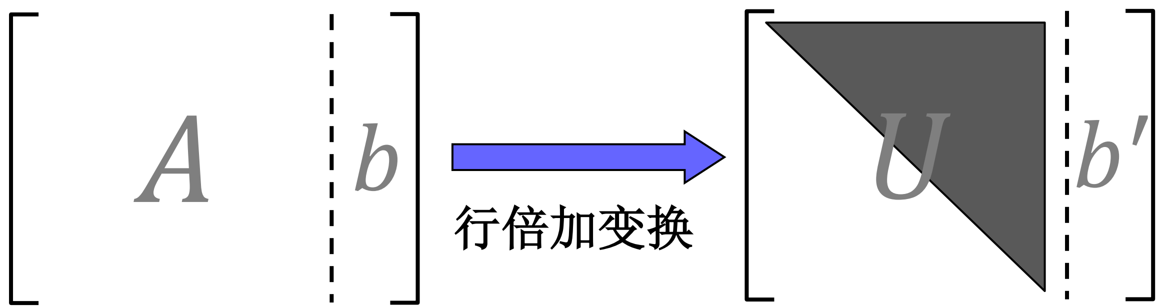
$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ \qquad\quad a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \qquad\qquad\qquad \cdots \\ \qquad\qquad\qquad\quad a_{nn}x_n = b_n \end{cases}$$

回代法

时间复杂度: O(n²)

高斯消去法

- 高斯消去法解线性方程组
  - 消去过程使系数矩阵变为上三角矩阵



  - 回代过程解<u>上三角型</u>方程组
  - 第k步消去过程可执行的条件：主元$a_{kk}^{(k)} \neq 0$

例 3.5 (高斯消去法求解线性方程组): 求解线性方程组

$$\begin{cases} 10x_1 - 7x_2 = 7 \\ -3x_1 + 2x_2 + 6x_3 = 4 \\ 5x_1 - x_2 + 5x_3 = 6 \end{cases}$$

$$\begin{cases} x_1 - 0.7x_2 = 0.7 \\ x_2 - 60x_3 = -61 \\ 155x_3 = 155 \end{cases}$$

$$x_1 = 0, x_2 = -1, x_3 = 1$$

高斯消去法

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix}$$

- 算法3.1 解线性方程组的高斯消去过程

输入：$\boldsymbol{A}$, n, $\boldsymbol{b}$; 输出：$\boldsymbol{A}$, $\boldsymbol{b}$.
For k=1, 2, …, n-1
　If $a_{kk} = 0$ **then** 停止；
　For i=k+1, k+2,…, n
　　$c := -a_{ik}/a_{kk}$; {倍乘因子}
　　For j=k+1, k+2, …, n
　　　$a_{ij} := a_{ij} + c a_{kj}$; {更新矩阵元素}
　　End
　　$b_i := b_i + c b_k$; {更新右端项}
　End
End

a_{ik}

a_{kk}是主元, 不同于原始$\boldsymbol{A}$的对角线元素

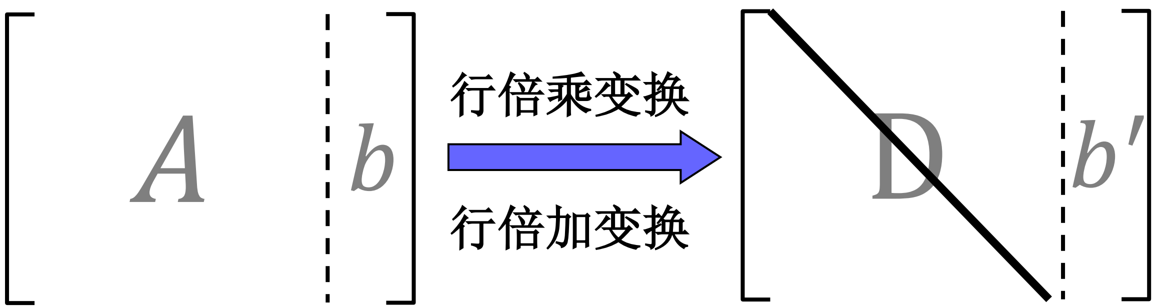
“原地工作”存储方式
无需额外存储空间

计算复杂度:
$\approx \boldsymbol{n^3/3}$

乘法 $(n)(n-1)+(n-1)(n-2)+\cdots+2\times 1$

高斯-约当(Gauss-Jordan)消去法

- 高斯消去法解线性方程组
  - 消去过程使系数矩阵变为对角矩阵



  - 用初等变换将A变为单位阵, 可以用于算逆矩阵
    - 时间复杂度: $\boldsymbol{n^3/2}$
    - 用到$\boldsymbol{A^{-1}}$时首先应想到“解方程组”, 而不是“求逆”

计算$\boldsymbol{A^{-1}b}$并不需要真正计算 $\boldsymbol{A^{-1}}$, 求解方程 $\boldsymbol{Ax=b}$ 即得到 $\boldsymbol{A^{-1}b}$ 的结果

例3.6(高斯-约当消去法求解线性方程组): 求解线性方程组

$$\begin{cases} 10x_1 - 7x_2 & = 7 \\ -3x_1 + 2x_2 + 6x_3 & = 4 \\ 5x_1 - x_2 + 5x_3 & = 6 \end{cases}$$

$$x_1 = 0, x_2 = -1, x_3 = 1$$

例3.7(矩阵求逆): 求矩阵的逆矩阵。

$$A = \begin{bmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{bmatrix}$$

$$\left[\begin{matrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{matrix}\right.\left.\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}\right] \rightarrow \left[\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}\right.\left.\begin{matrix} -0.1032 & -0.2258 & 0.2710 \\ -0.2903 & -0.3226 & 0.3871 \\ 0.0452 & 0.1613 & 0.0065 \end{matrix}\right]$$

$$\boldsymbol{A} \qquad \boldsymbol{I} \qquad \boldsymbol{I} \qquad \boldsymbol{A}^{-1}$$

矩阵的LU分解

高斯消去法

- 初等行(列)变换与矩阵
  - 倍乘变换、倍加变换、 交换变换，及其对应的矩阵

$$E_c^{(1)} = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & c & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix} \quad E^{(2)} = \begin{bmatrix} \ddots & & & & \\ & 1 & & & \\ & & \ddots & & \\ & c & & 1 & \\ & & & & \ddots \end{bmatrix} \quad E^{(3)} = \begin{bmatrix} \ddots & & & & \\ & 0 & & 1 & \\ & & \ddots & & \\ & 1 & & 0 & \\ & & & & \ddots \end{bmatrix}$$

  - 倍加变换矩阵$\boldsymbol{E}^{(2)}$是对单位阵$\boldsymbol{I}$的某一行乘以c加到一行的结果
  - $\boldsymbol{E}^{(2)}$左乘矩阵$\boldsymbol{A}$的结果$\boldsymbol{E}^{(2)}\boldsymbol{A}$，是对$\boldsymbol{A}$实施相应的行倍加变换得到的矩阵
  - 右乘初等变换阵则相当于对矩阵的列实施初等变换

矩阵的LU分解

- 高斯消去过程对矩阵$\boldsymbol{A}$的变换 (3阶矩阵为例)

$$\boldsymbol{A}^{(1)} \triangleq \boldsymbol{A} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & a_{23}^{(1)} \\ a_{31}^{(1)} & a_{32}^{(1)} & a_{33}^{(1)} \end{bmatrix} \xrightarrow[\text{行倍加变换}]{a_{11}^{(1)} \neq 0} \boldsymbol{A}^{(2)} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} \\ 0 & a_{32}^{(2)} & a_{33}^{(2)} \end{bmatrix}$$

等价于左乘消去矩阵:

两次倍加变换

乘数

是单位下三角阵

$$\boldsymbol{M}_1 = \begin{bmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & 0 & 1 \end{bmatrix}, \text{其中} m_{i1} = -\frac{a_{i1}^{(1)}}{a_{11}^{(1)}}, (i = 2,3)$$

$\boldsymbol{M}_1\boldsymbol{A} = \boldsymbol{A}^{(2)}$, 更新的矩阵元素$a_{ij}^{(2)} = a_{ij}^{(1)} + m_{i1}a_{1j}^{(1)}$,

$(i = 2,3, j = 1,2,3)$

矩阵的LU分解

- 第2步消去

$$A^{(2)}=\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} \\ 0 & a_{32}^{(2)} & a_{33}^{(2)} \end{bmatrix} \xrightarrow[\text{行倍加变换}]{a_{22}^{(2)}\neq 0} A^{(3)}=\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} \\ 0 & 0 & a_{33}^{(3)} \end{bmatrix}$$

等价于左乘消去矩阵

$$M_2=\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & m_{32} & 1 \end{bmatrix},\text{其中} m_{32}=-\frac{a_{32}^{(2)}}{a_{22}^{(2)}}$$

一次倍加变换

$$M_2A^{(2)}=A^{(3)},\text{更新元素} a_{33}^{(3)}=a_{33}^{(2)}+m_{32}a_{23}^{(2)}$$

$$\Rightarrow M_2M_1A=A^{(3)}\triangleq U \Rightarrow A=M_1^{-1}M_2^{-1}U$$

矩阵的LU分解

$$A = M_1^{-1} M_2^{-1} U$$

- 消去矩阵的逆矩阵

$$M_1 = \begin{bmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & 0 & 1 \end{bmatrix}, \quad M_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -m_{21} & 1 & 0 \\ -m_{31} & 0 & 1 \end{bmatrix}$$

逆矩阵也属于消去矩阵

$$M_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & m_{32} & 1 \end{bmatrix}, \quad M_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -m_{32} & 1 \end{bmatrix}$$

注: 下标指示消去矩阵的列号(类型数)

  - 只需改变非对角元的符号

- 类型数从小到大的消去阵依次相乘,其乘积为单位下三角阵, 且其非零元为各消去阵的“并”

上例中, $A = M_1^{-1} M_2^{-1} U$, 其中$M_1^{-1} M_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -m_{21} & 1 & 0 \\ -m_{31} & -m_{32} & 1 \end{bmatrix}$

- $A = LU$, L为单位下三角阵

矩阵L

矩阵的LU分解

- **A=LU**
  - L为单位下三角阵，U为上三角阵 (Doolittle分解) 默认
  - L为下三角阵，U为单位上三角阵 (Crout分解)
- **定理3.14** 对方程$\boldsymbol{Ax}=\boldsymbol{b}$, 其中$\boldsymbol{A}\in\mathbb{R}^{n\times n}$, 若执行高斯消去过程中的主元$a_{kk}^{(k)}\neq 0, (k=1,2,\cdots,n-1)$, $\Leftrightarrow$系数矩阵$\boldsymbol{A}$存在唯一的LU分解　(充要条件)

只证 ⇨

- 前面的推导已说明存在LU分解, 用反证法证明唯一性

设$\boldsymbol{A}=\boldsymbol{L}_1\boldsymbol{U}_1=\boldsymbol{L}_2\boldsymbol{U}_2$, $\boldsymbol{L}_1$和$\boldsymbol{L}_2$为单位下三角阵(非奇异)　矛盾!

若$\boldsymbol{A}$非奇异, $\boldsymbol{U}_1$也非奇异 $\Longrightarrow$ $\boldsymbol{L}_2^{-1}\boldsymbol{L}_1=\boldsymbol{U}_2\boldsymbol{U}_1^{-1}=\boldsymbol{I}$ $\Longrightarrow$ $\begin{cases}\boldsymbol{L}_1=\boldsymbol{L}_2\\ \boldsymbol{U}_1=\boldsymbol{U}_2\end{cases}$

矛盾! 若$\boldsymbol{A}$奇异, 则U(n,n)=0

$$\begin{bmatrix}\boldsymbol{L}_1' & \boldsymbol{0}\\ \boldsymbol{\alpha}_1^T & 1\end{bmatrix}\begin{bmatrix}\boldsymbol{U}_1' & \boldsymbol{\beta}_1\\ \boldsymbol{0}^T & 0\end{bmatrix}=\begin{bmatrix}\boldsymbol{L}_2' & \boldsymbol{0}\\ \boldsymbol{\alpha}_2^T & 1\end{bmatrix}\begin{bmatrix}\boldsymbol{U}_2' & \boldsymbol{\beta}_2\\ \boldsymbol{0}^T & 0\end{bmatrix}$$

$\Longrightarrow \boldsymbol{L}_1'\boldsymbol{U}_1'=\boldsymbol{L}_2'\boldsymbol{U}_2' \Longrightarrow \begin{matrix}\boldsymbol{L}_1'=\boldsymbol{L}_2'\\ \boldsymbol{U}_1'=\boldsymbol{U}_2'\end{matrix} \Longrightarrow \begin{matrix}\boldsymbol{L}_1=\boldsymbol{L}_2\\ \boldsymbol{U}_1=\boldsymbol{U}_2\end{matrix}$

矩阵的LU分解

例3.9矩阵**LU**分解: $\boldsymbol{A}=\begin{bmatrix}1&2&2\\4&4&2\\4&6&4\end{bmatrix} \Rightarrow \begin{bmatrix}1&2&2\\0&-4&-6\\0&-2&-4\end{bmatrix}$

1. 乘数$m_{21}=m_{31}=-4$
2. 乘数$m_{32}=-1/2$

$\Rightarrow \begin{bmatrix}1&0&0\\4&1&0\\4&1/2&1\end{bmatrix}$ L $\quad \begin{bmatrix}1&2&2\\0&-4&-6\\0&0&-1\end{bmatrix}$ U

矩阵的LU分解

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix}$$

- **算法3.5** 用高斯消去过程做LU分解

输入：$\boldsymbol{A}$, n; 输出：$\boldsymbol{A}$.
For k=1, 2, …, n-1
    If $a_{kk}=0$ **then** 停止；
    For i=k+1, k+2, …, n
        $a_{ik}:=a_{ik}/a_{kk}$； {$\boldsymbol{L}$矩阵的元素}
        For j=k+1, k+2, …, n
            $a_{ij}:=a_{ij}-a_{ik}a_{kj}$；
        End
    End
End

乘数的相反数

原地存储

- 执行结束后, $\boldsymbol{A}$的上三角部分成为矩阵$\boldsymbol{U}$, 而对角线以下部分是矩阵$\boldsymbol{L}$的元素(其对角元1不存储)
- 计算量: $\approx n^3/3$

矩阵的LU分解

- 另一种计算LU分解的思路

$$\begin{bmatrix} 1 & & \\ l_{21} & 1 & \\ l_{31} & l_{32} & 1 \end{bmatrix}\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ & u_{22} & u_{23} \\ & & u_{33} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

根据矩阵乘法列9个方程, 解9个未知量

- ***A***的第一行: $u_{1j} = a_{1j}, (j = 1,2,3)$
- ***A***的第一列: $a_{i1} = l_{i1}u_{11} \Rightarrow l_{i1} = a_{i1}/u_{11}, (i = 2,3)$
- ***A***的第二行: $a_{2j} = l_{21}u_{1j} + u_{2j} \Rightarrow u_{2j} = a_{2j} - l_{21}u_{1j}, (j=2,3)$
- ***A***的第二列: $a_{32} = l_{31}u_{12} + l_{32}u_{22} \Rightarrow l_{32}=(a_{32} - l_{31}u_{12})/u_{22}$
- ***A***最后一个元素: $a_{33} = \sum_{i=1}^{2} l_{3i}u_{i3} + u_{33}$, …

按一定顺序列方程，可逐个逐个求出L和U的元素!

直接LU分解算法 (算法3.6)

$$
\begin{bmatrix}
1 & & & & & \\
\vdots & \ddots & & & & \\
\vdots & & \ddots & & & \\
l_{k1} & \cdots & l_{k,k-1} & 1 & & \\
\vdots & & \vdots & l_{ik} & \ddots & \\
l_{n1} & \cdots & l_{n,k-1} & & \cdots & 1
\end{bmatrix}
\begin{bmatrix}
u_{11} & \cdots & \cdots & u_{1k} & \cdots & u_{1n} \\
 & \ddots & & \vdots & & \vdots \\
 & & \ddots & u_{k-1,k} & \cdots & u_{k-1,n} \\
 & & & u_{kk} & \cdots & u_{kn} \\
 & & & & \ddots & \vdots \\
 & & & & & u_{nn}
\end{bmatrix}
$$

输入：***A***, n; 输出：***A***.

For k=1, 2, …, n-1

  If $a_{kk} = 0$ then 停止；

  For i=k+1, k+2, …, n {算***L***的第k列}

    For j=1, 2, …, k-1   {用***U***的第k列}

      $a_{ik} := a_{ik} - \boldsymbol{u}_{jk}\, \boldsymbol{l}_{ij}$;

    End

    $\boldsymbol{l}_{ik} := a_{ik} / \boldsymbol{u}_{kk}$;

  End   {***U***的第1行已知!}

  For j=k+1, k+2, …, n {算 ***U*** 的第k+1行}

    For i=1, 2, …, k   {用***L***的第k+1行}

      $\boldsymbol{u}_{k+1,j} := a_{k+1,j} - \boldsymbol{l}_{k+1,i}\boldsymbol{u}_{ij}$;

    End

  End

End

- 数学上等价算法3.5
- 算法描述不同，存取、计算的顺序不同
- 对于稠密矩阵(二维数组)计算量一样；对于稀疏矩阵(特殊数据结构)，效率可能有较大差别

LU分解的用途

$\boldsymbol{x}=\boldsymbol{A}^{-1}\boldsymbol{b}$, 并非要算$\boldsymbol{A}^{-1}$才能得$\boldsymbol{x}$

实际上应避免算$\boldsymbol{A}^{-1}$

- 单个方程求解

  $\boldsymbol{Ax}=\boldsymbol{b} \quad \Longrightarrow \boldsymbol{x}=(\boldsymbol{LU})^{-1}\boldsymbol{b}=\boldsymbol{U}^{-1}\boldsymbol{L}^{-1}\boldsymbol{b}$

  - **1.**解单位下三角型方程组$\boldsymbol{Ly}=\boldsymbol{b}$ $\quad (\boldsymbol{y}=\boldsymbol{L}^{-1}\boldsymbol{b})$
  - **2.**解上三角型方程组$\boldsymbol{Ux}=\boldsymbol{y}$
  - 计算量: n^2次乘除法
- 右端项变化的问题 (多右端方程组)
  - $\boldsymbol{Ax}_i=\boldsymbol{b}_i,\ i=1,\dots,m$
  - 对每个右端项执行上述两步，不必重复算**LU**分解
- 用途：表达式简洁; 有效解决多右端项等问题; 将复杂的计算独立出来, 便于程序包的开发和应用

选主元技术与稳定性

如何解决主元为零的问题

- 选主元技术 (pivoting)
  - 若当前主元$a_{kk}^{(k)}=0$, 看它同一列下面的元素. 若$a_{jk}^{(k)}\neq 0, j>k$,交换第j、第k行(这不改变方程的解)
  - 可证明: 只要A非奇异,一定能找到$a_{jk}^{(k)}\neq 0$.   <u>这种交换矩阵行/列来改变主元的操作称为选主元</u>   <u>选主元还能减小数值误差</u>
  - 乘子$m_{ik}=-a_{ik}^{(k)}/a_{kk}^{(k)}$, 用它乘以当前行加其它行上, 若使$|m_{ik}|$较小, 则减小操作数上误差的放大、传播
  - 通过选主元, 可使主元尽可能大, 增强算法稳定性

$$A^{(k)}\quad \begin{bmatrix} a_{11}^{(1)} & \cdots & a_{1,k}^{(1)} & \cdots & a_{1,n}^{(1)} \\ & \ddots & \vdots & \cdots & \cdots \\ & & a_{k,k}^{(k)} & \cdots & a_{k,n}^{(k)} \\ & & \vdots & \cdots & \vdots \\ & & a_{n,k}^{(k)} & \cdots & a_{n,n}^{(k)} \end{bmatrix}$$

例 (不能做 LU 分解的非奇异矩阵): 试求下述线性方程组系数矩阵的 LU 分解, 并判断该线性方程组的可解性:

$$\begin{cases} 2x_2 = 2 \\ 3x_1 + x_2 = 4 \end{cases}$$

【解】系数矩阵为 $\boldsymbol{A} = \begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix}$, 第一个主元 $a_{11} = 0$, 无法执行高斯消去过程, 而且即使采用待定系数法, 也无法求出 $\boldsymbol{L}$ 和 $\boldsymbol{U}$, 因此矩阵 $\boldsymbol{A}$ 不能进行 LU 分解。

$\det(\boldsymbol{A}) = -6$,矩阵 $\boldsymbol{A}$ 非奇异, 该方程组有唯一的解: $x_1 = 1, x_2 = 1$ 。

例(奇异矩阵做 LU 分解) 奇异矩阵 $\boldsymbol{A} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$, $\boldsymbol{A} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} = LU$ 。这种奇异矩阵做 LU 分解得到的 U 矩阵最后一个对角元为零,

矩阵 A 能不能做 LU 分解 与 它的奇异性无关。

例(小主元带来的误差危害)：解线性方程组

$$\boldsymbol{Ax}=\begin{bmatrix}\varepsilon & 1\\ 1 & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}=\begin{bmatrix}1\\ \varepsilon\end{bmatrix},$$

其中，ε为一个很小的正数（小于浮点系统机器精度$\varepsilon_{\mathrm{mach}}$的一半）

这个方程的准确解为 $x_2^*=1+\varepsilon\approx 1, x_1^*=-1$ 。

如果不做行交换，得到消去过程的乘数为 $-1/\varepsilon$, 做一次消去后, 增广矩阵为

$$\begin{bmatrix}\varepsilon & 1 & 1\\ 0 & 1-1/\varepsilon & \varepsilon-1/\varepsilon\end{bmatrix},$$

由于 1 和 $1/\varepsilon$ 的值相差悬殊,做浮点数加减时 1 会被 $1/\varepsilon$ “吞没”。同理, ε 也会被$1/\varepsilon$ 吞没。因此，在浮点计算中增广矩阵为

$$\begin{bmatrix}\varepsilon & 1 & 1\\ 0 & -1/\varepsilon & -1/\varepsilon\end{bmatrix}$$

解出 $x_2=1, x_1=(1-1)/\varepsilon=0$ 。可见, 数值解 x_1 的误差非常大。

上述高斯消去过程对应的 LU 分解为

$$\boldsymbol{L}=\begin{bmatrix}1 & 0\\ 1/\varepsilon & 1\end{bmatrix},\ \boldsymbol{U}=\begin{bmatrix}\varepsilon & 1\\ 0 & 1-1/\varepsilon\end{bmatrix}=\begin{bmatrix}\varepsilon & 1\\ 0 & -1/\varepsilon\end{bmatrix}。$$

最后一个等号是由于 1 和 $1/\varepsilon$ 的值相差悬殊。

$$\boldsymbol{LU}=\begin{bmatrix}1 & 0\\ 1/\varepsilon & 1\end{bmatrix}\begin{bmatrix}\varepsilon & 1\\ 0 & -1/\varepsilon\end{bmatrix}=\begin{bmatrix}\varepsilon & 1\\ 1 & 0\end{bmatrix}\neq\boldsymbol{A},$$

这也说明小主元造成了不可恢复的信息损失。

如果**交换行**, 主元变为较大的 1 , 则对应得到增广矩阵 $\begin{bmatrix}1 & 1 & \varepsilon\\ \varepsilon & 1 & 1\end{bmatrix}$,

利用乘数为 ε 做一次消去后, 得到 $\begin{bmatrix}1 & 1 & \varepsilon\\ 0 & 1-\varepsilon & 1-\varepsilon^2\end{bmatrix}$,

考虑浮点计算的舍入, 实际计算的增广矩阵为 $\begin{bmatrix}1 & 1 & \varepsilon\\ 0 & 1 & 1\end{bmatrix}$,

由此解出 $x_2=1, x_1=\varepsilon-1=-1$, 它与准确解非常接近。

使用部分主元技术的LU分解

- 部分主元(列主元)高斯消去法
  - 在第k步，选第k列未消去部分绝对值最大元素
  - 选i_k, ($i_k \geq k$), 使得$\left|a_{i_k,k}^{(k)}\right| = \max\limits_{k\leq i\leq n}\left|a_{i,k}^{(k)}\right|$
  - 若$i_k \neq k$, 交换$\boldsymbol{A}$的第i_k行与第k行，再进行消去操作

  这样保证算法可执行，且乘子$|m_{ik}| = \left|a_{ik}^{(k)}/a_{kk}^{(k)}\right| \leq 1$
- 初等交换阵$\boldsymbol{P}_k$
  - 记$\boldsymbol{P}_k$为交换$\boldsymbol{I}$的第k行和第i_k,(i_k>k)行得到
  - $\boldsymbol{P}_k^T = \boldsymbol{P}_k$, $\boldsymbol{P}_k^{-1} = \boldsymbol{P}_k$
  - 可将$\boldsymbol{I}$视为特殊的交换阵，$i_k = k$

$\boldsymbol{P}_k$

$$\begin{array}{cc} & \begin{array}{ccc} k & & i_k \\ \downarrow & & \downarrow \end{array} \\ \begin{array}{c} \\ k\rightarrow \\ \\ i_k\rightarrow \\ \\ \end{array} & \begin{bmatrix} \ddots & & & & \\ & 0 & & 1 & \\ & & \ddots & & \\ & 1 & & 0 & \\ & & & & \ddots \end{bmatrix} \end{array}$$

使用部分主元技术的LU分解

- 部分主元消去过程: $M_{n-1}P_{n-1}\cdots M_2P_2M_1P_1A=U$
  - $P_k\ (k=1,\dots n-1)$实现第k与第i_k行($i_k\geq k$)的交换(P_k可能为I)
  - 两边同乘M_{n-1}^{-1}, $\Longrightarrow$ $P_{n-1}M_{n-2}P_{n-2}\cdots M_1P_1A=M_{n-1}^{-1}U$

$\Longrightarrow$ $\underbrace{P_{n-1}M_{n-2}(P_{n-1}}_{\text{第n-1行与}i_{n-1}\text{行}(i_{n-1}\geq \text{n-1})\text{交换后，再做同样的列交换}}P_{n-1})P_{n-2}\cdots M_1P_1A=M_{n-1}^{-1}U$

  - 记$\bar{M}_{n-2}=P_{n-1}M_{n-2}P_{n-1}$，它也是消去阵 $\Longrightarrow$

$$\bar{M}_{n-2}=\begin{bmatrix}1&&&&\\&\ddots&&&\\&&1&&\\&&m_{n,n-2}&1&\boxed{1}\\&&m_{n-1,n-2}&\boxed{1}&1\end{bmatrix}$$

$P_{n-1}P_{n-2}M_{n-3}P_{n-3}\cdots M_1P_1A=\bar{M}_{n-2}^{-1}M_{n-1}^{-1}U$

$\Longrightarrow$ $\underbrace{P_{n-1}P_{n-2}M_{n-3}(P_{n-2}P_{n-1}}P_{n-1}P_{n-2})P_{n-3}\cdots M_1P_1A=\cdots$

使用部分主元技术的LU分解

$$\underbrace{(P_{n-1}P_{n-2}M_{n-3}P_{n-2}P_{n-1})}_{\text{记为}\bar{M}_{n-3}\text{, 也是消去阵}}P_{n-1}P_{n-2}P_{n-3}\cdots M_1P_1A=\bar{M}_{n-2}^{-1}M_{n-1}^{-1}U$$

$\Rightarrow P_{n-1}P_{n-2}P_{n-3}M_{n-4}P_{n-4}\cdots M_1P_1A = \bar{M}_{n-3}^{-1}\bar{M}_{n-2}^{-1}M_{n-1}^{-1}U$

…… ……, $\Rightarrow P_{n-1}P_{n-2}\cdots P_1A = \bar{M}_1^{-1}\cdots\bar{M}_{n-2}^{-1}M_{n-1}^{-1}U$

记L=$\bar{M}_1^{-1}\cdots\bar{M}_{n-2}^{-1}M_{n-1}^{-1}$, 是单位下三角阵(消去阵乘积)

记$P = P_{n-1}P_{n-2}\cdots P_1$, 称为排列阵 (*permutation*), 是单位阵I行重排的结果

$$\begin{bmatrix} & 1 & \\ 1 & & \\ & & 1 \end{bmatrix}$$

例: 一个3阶排列阵

- 部分主元的LU分解: $PA = LU$
- 矩阵L: $\bar{M}_k$(k=1, …, n-2)及M_{n-1}元素取相反数再合并
  $\bar{M}_k = P_{n-1}\cdots P_{k+1}M_kP_{k+1}\cdots P_{n-1}$, $(k \le n-2)$, 将后续的行交换作用于M_k的第k列元素便得到$\bar{M}_k$

使用部分主元技术的LU分解

- 计算部分主元LU分解
  - 例: $A = \begin{bmatrix} 1 & 2 & 2 \\ 4 & 4 & 2 \\ 4 & 6 & 4 \end{bmatrix} \xrightarrow{\text{行交换}} \begin{bmatrix} 4 & 4 & 2 \\ 1 & 2 & 2 \\ 4 & 6 & 4 \end{bmatrix} \xrightarrow[\text{消去}]{m_{*1}= -1/4,\ -1} \begin{bmatrix} 4 & 4 & 2 \\ 1/4\ \cancel{0} & 1 & 3/2 \\ 1\ \cancel{0} & 2 & 2 \end{bmatrix}$

$$\xrightarrow[\text{行交换}]{\bar{M}_1^{-1}} \begin{bmatrix} 4 & 4 & 2 \\ 1\ \cancel{1/4} & 2 & 2 \\ 1/4\ \cancel{1} & 1 & 3/2 \end{bmatrix} \xrightarrow[\text{消去}]{m_{32}=-1/2} \begin{bmatrix} 4 & 4 & 2 \\ 1 & 2 & 2 \\ 1/4 & 1/2\ \cancel{0} & 1/2 \end{bmatrix} \qquad \underbrace{\begin{bmatrix} 1 & & \\ 1 & 1 & \\ 1/4 & 1/2 & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} 4 & 4 & 2 \\ & 2 & 2 \\ & & 1/2 \end{bmatrix}}_{U}$$

$$\text{排列阵}\boldsymbol{P}: \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix} \rightarrow \begin{bmatrix} & 1 & \\ 1 & & \\ & & 1 \end{bmatrix} \rightarrow \begin{bmatrix} & 1 & \\ & & 1 \\ 1 & & \end{bmatrix} \quad \text{验证} \quad \underbrace{\begin{bmatrix} 4 & 4 & 2 \\ 4 & 6 & 4 \\ 1 & 2 & 2 \end{bmatrix}}_{PA}$$

  - $\boldsymbol{P}$是对$\boldsymbol{I}$行重排的结果, 只需用一个长为n的数组p[ ]表示, p[i]的值指示$\boldsymbol{P}$的第i行是$\boldsymbol{I}$的第几行　　***PA*则对*A*做相同重排**
  - p的初始值[1,2,..., n], 每次行交换即交换p对应两个单元

例(不显式进行行交换的部分主元消去过程)：不显式进行行交换, 用部分主元高斯消去法分解矩阵 $\boldsymbol{A}=\begin{bmatrix}1 & 2 & 2\\ 4 & 4 & 2\\ 4 & 6 & 4\end{bmatrix}$

第一步消去过程, $p=[2,1,3]$, 乘数 $m_{21}=-A(p[2],1)/A(p[1],1)=-1/4$, $m_{31}=-A(p[3],1)/A(p[1],1)=-1$, 这里利用 $p[i]$ 来索引行交换后矩阵的第 i 行, 使用同样的方法更新其他元素, 得到

$$\boldsymbol{A}=\begin{bmatrix}1/4 & 1 & 3/2\\ 4 & 4 & 2\\ 1 & 2 & 2\end{bmatrix}$$

第二步消去过程, 考查 $A(p[2],2)$ 和 $A(p[3],2)$, 需交换第二行、第三行, 得到 $p=[2,3,1]$, 乘数 $m_{32}=-A(p[3],2)/A(p[2],2)=-1/2$ 。类似地, 更新 A未消去部分的其他元素,得到

$$\boldsymbol{A}=\begin{bmatrix}1/4 & 1/2 & 1/2\\ 4 & 4 & 2\\ 1 & 2 & 2\end{bmatrix}$$

最后, 根据 A 和 p 的值, 可以得到矩阵 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 。

$$\boldsymbol{PA}=\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ 1 & 0 & 0\end{bmatrix}\begin{bmatrix}1 & 2 & 2\\ 4 & 4 & 2\\ 4 & 6 & 4\end{bmatrix}=\begin{bmatrix}1 & 0 & 0\\ 1 & 1 & 0\\ 1/4 & 1/2 & 1\end{bmatrix}\begin{bmatrix}4 & 4 & 2\\ 0 & 2 & 2\\ 0 & 0 & 1/2\end{bmatrix}=\boldsymbol{LU}$$

采用部分主元的LU分解算法 (算法3.9)

输入：$\boldsymbol{A}$, n; 输出：$\boldsymbol{A}$, 一维数组p.
p= [1, 2, 3,…, n];
For k=1, 2, …, n-1
    确定满足$|a_{sk}| = \max_{k\le i\le n} |a_{ik}|$的$s$;
    If $s \neq k$ **then**
        交换矩阵$\boldsymbol{A}$的第k行与第s行;
        交换p[k]与p[s];
    End
    For i=k+1, k+2, …, n
        $a_{ik} := a_{ik}/a_{kk}$;
        For j=k+1, k+2, …, n
            $a_{ij} := a_{ij} - a_{ik}a_{kj}$;
        End
    End
End

- 只比算法3.5增加了选主元的操作
- n-1次求最大值不影响总乘法量$n^3/3$
- 交换行的操作并不意味着移动矩阵元素
- 额外存储量为一维整型数组
- 算法不中断, 稳定性好

部分主元技术与其他主元技术

- 部分主元LU分解的应用
  - $Ax = b \Longrightarrow PAx = Pb \Longrightarrow x = (LU)^{-1}Pb = U^{-1}L^{-1}Pb$
  - 先对右端项重排, 再执行前代、回代过程
- 其他选主元技术
  - 数值更稳定的全选主元：未消去子矩阵中选最大元素, 通过行、列交换到主元位置

$$A^{(k)} = \begin{bmatrix} a_{11}^{(1)} & \cdots & a_{1,k}^{(1)} & \cdots & a_{1,n}^{(1)} \\ & \ddots & \vdots & \cdots & \cdots \\ & & a_{k,k}^{(k)} & \cdots & a_{k,n}^{(k)} \\ & & \vdots & \cdot & \vdots \\ & & a_{n,k}^{(k)} & \cdots & a_{n,n}^{(k)} \end{bmatrix}$$

  - 乘数更小, 但开销增大
  - 矩阵形式: $PAQ = LU$, Q为排列阵
- 通过选主元, 得到实用的高斯消去法、高斯-约当消去法

Matlab

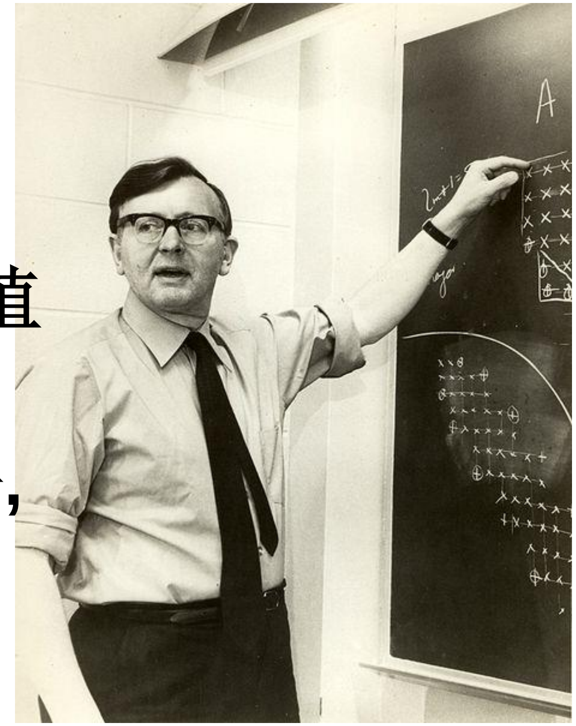
(1) Partial Pivoting $\boldsymbol{PA = LU}$

$$
\begin{aligned}
[L, U, P] &= \mathrm{lu}(A); && \rightarrow \text{cost } O(n^3) \\
z &= P * b; && \rightarrow \text{cost } O(n) \\
y &= L \setminus z; && \rightarrow \text{cost } O(n^2) \\
x &= U \setminus y; && \rightarrow \text{cost } O(n^2)
\end{aligned}
$$

(2) Complete Pivoting $\boldsymbol{PAQ = LU}$

$$
\begin{aligned}
[L, U, P, Q] &= \mathrm{lu}(A); && \rightarrow \text{cost } O(n^3) \\
c &= P * b && \rightarrow \text{cost } O(n) \\
y &= L \setminus c; && \rightarrow \text{cost } O(n^2) \\
z &= U \setminus y; && \rightarrow \text{cost } O(n^2) \\
x &= Q * z && \rightarrow \text{cost } O(n)
\end{aligned}
$$

算法的稳定性



- **Wilkinson与向后误差分析**
  - 高斯消去法主要受舍入误差影响
  - 向后误差分析: 设方程$\boldsymbol{Ax} = \boldsymbol{b}$的数值解为$\widehat{\boldsymbol{x}}$, 等效地, $(\boldsymbol{A} + \Delta\boldsymbol{A})\widehat{\boldsymbol{x}} = \boldsymbol{b}$
  - 需分析$\|\Delta\boldsymbol{A}\|/\|\boldsymbol{A}\|$.对各种LU分解算法, 有结论: $\frac{\|\Delta\boldsymbol{A}\|_\infty}{\|\boldsymbol{A}\|_\infty} \lesssim n\rho\varepsilon_{\mathrm{mach}}$
    ρ为$\boldsymbol{A}^{(k)}$ (或者$\boldsymbol{U}$)与$\boldsymbol{A}$最大元素之比
- **增长因子ρ的上限**
  - 部分选主元: $\rho \le 2^{n-1}$, 一般**<10**, 是实用的稳定算法
  - 不选主元: ρ可能任意大

对称正定矩阵的Cholesky分解

对称矩阵的LU分解

- LDL^T分解
  - 将LU分解中的U矩阵写成对角阵与单位上三角阵之积

$$U=\begin{bmatrix} u_{11} & & & \\ & u_{22} & & \\ & & \ddots & \\ & & & u_{nn} \end{bmatrix}\begin{bmatrix} 1 & u_{12}/u_{11} & \cdots & u_{1n}/u_{11} \\ & 1 & \cdots & u_{2n}/u_{22} \\ & & \ddots & \vdots \\ & & & 1 \end{bmatrix}=DU_0$$

  - 对称矩阵: $A=LDU_0=A^T=U_0^TDL^T$
  - 定理(LDL^T分解) 若高斯消去不中断, LU分解唯一(L=U_0^T), 则 $A=LDL^T$, 其中, L 为单位下三角矩阵, D 为对角矩阵。
  - 定理 对称正定矩阵(SPD) $A=LDL^T$唯一地存在 $D=L^{-1}AL^{-T}$, 也是对称正定阵 $\Rightarrow$ D的对角元u_{ii}>0

$\boldsymbol{A}=\left(a_{ij}\right)_{n\times n}$ 是对称正定矩阵, 则对角元素 $a_{ii}>0(i=1,2,\cdots,n)$

证明　由于 $\boldsymbol{A}$ 对称正定,故 $a_{ii}=e_i{}^{\mathrm{T}}\boldsymbol{A}e_i>0,\ i=1,2,\cdots,n$,

其中 $\boldsymbol{e}_i=(0,\cdots,0,i,1,0,\cdots,0)^{\mathrm{T}}$ 为第 i 个单位向量.

对称正定矩阵 $\boldsymbol{A}$ 的$\boldsymbol{LDL}^{\mathrm{T}}$分解中，对角矩阵 $\boldsymbol{D}$ 的对角线元素 $u_{ii}>0,(i=1,2,\cdots,n)$

证明　$\forall\boldsymbol{x}\neq\boldsymbol{0}$, 设 $\boldsymbol{x}=\boldsymbol{L}^{\mathrm{T}}\boldsymbol{y}$, 由 $\boldsymbol{L}^{\mathrm{T}}$的非奇异性可推出 $\boldsymbol{y}\neq\boldsymbol{0}$，则$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{x}=\boldsymbol{y}^{\mathrm{T}}\boldsymbol{LDL}^{\mathrm{T}}\boldsymbol{y}=\boldsymbol{y}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{y}>\boldsymbol{0}$，因此$\boldsymbol{D}$为对称正定矩阵，其对角线元素一定大于 0 。

对称正定矩阵的LU分解

- **Cholesky分解**
  - 对称正定矩阵, $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^T$, $\boldsymbol{D}$的对角元$u_{ii} > 0$
  - 记$\boldsymbol{D}^{\frac{1}{2}} = \begin{bmatrix} \sqrt{u_{11}} & & \\ & \ddots & \\ & & \sqrt{u_{nn}} \end{bmatrix}$
  - $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{D}^{\frac{1}{2}}\boldsymbol{D}^{\frac{1}{2}}\boldsymbol{L}^T = \boldsymbol{L}_1\boldsymbol{L}_1^T$, 其中$\boldsymbol{L}_1$为下三角阵

  定理3.18 若$\boldsymbol{A}$为SPD阵, $\boldsymbol{L}$为对角元>0的下三角阵, 则$\boldsymbol{A}$可唯一地分解为$\boldsymbol{A}=\boldsymbol{L}\boldsymbol{L}^T$的形式 (也写作$\boldsymbol{A}=\boldsymbol{R}^T\boldsymbol{R}$)

  - 上述矩阵分解称为Cholesky分解, 可通过LU分解求Cholesky因子, 计算量同LU分解 (~ $n^3/3$ 乘法)
  - 对称阵存储量可省一半, 分解的计算量能减少吗?

Cholesky分解算法

- 也叫*平方根法*. 下面按**直接LU分解**思想推导算法

$$\begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & \cdots & l_{n1} \\ 0 & l_{22} & \cdots & l_{n2} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & l_{nn} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

$L \quad L^T$

  - 算法3.10 (按从1到n的顺序逐列算出L的元素值)

若 $A = LL^T$, $a_{jj} = l_{j1}^2 + \cdots + l_{jj}^2$

For j =1 to n

$$a_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} a_{jk}^2} \qquad \{\text{对角元}\}$$

For $i = j+1$ to n {当前列}

$$a_{ij} = (a_{ij} - \sum_{k=1}^{j-1} a_{ik}a_{jk}) / a_{jj}$$

End

End

- 仅读取A的下三角部分
- 原地存L的结果
- 计算复杂度: (不计开方) $\approx n^3/6$乘除法

例(Cholesky分解)：计算下面这个对称正定矩阵的 Cholesky 分解:

$$\boldsymbol{A}=\begin{bmatrix}5 & -1 & -1\\ -1 & 3 & -1\\ -1 & -1 & 5\end{bmatrix}$$

【解】 首先将第一列除以第一个对角元的平方根, $\sqrt{5}\approx 2.2361$, 得

$$\begin{bmatrix}\color{red}{2.2361} & 0 & 0\\ -0.4472 & 3 & 0\\ -0.4472 & -1 & 5\end{bmatrix}$$

计算第二个对角元, $a_{22}=\sqrt{a_{22}-a_{21}^2}=\sqrt{3-0.4472^2}\approx 1.6733$ 。注意, 这里的 a_{ij} 表示改变了的矩阵 $\boldsymbol{A}$ 的元素。然后计算第二列余下部分, 得

$$\begin{bmatrix}2.2361 & 0 & 0\\ -0.4472 & \color{red}{1.6733} & 0\\ -0.4472 & -0.7171 & 5\end{bmatrix}$$

计算第三个对角元, $a_{33}=\sqrt{a_{33}-a_{31}^2-a_{32}^2}=\sqrt{5-0.4472^2-0.7171^2}\approx 2.0702$,

得到所要的矩阵 $\boldsymbol{L}=\begin{bmatrix}2.2361 & 0 & 0\\ -0.4472 & 1.6733 & 0\\ -0.4472 & -0.7171 & \color{red}{2.0702}\end{bmatrix}$。

Cholesky分解算法

$$\frac{\|\Delta A\|_\infty}{\|A\|_\infty} \lesssim n\rho\varepsilon_{\mathrm{mach}}$$

- 通过增长因子分析算法稳定性

  考虑根据LDL^T分解推导的算法

$$A = L_0U = L_0D^{1/2}D^{1/2}L_0^T = LL^T \Longrightarrow \begin{aligned} U &= D^{1/2}L^T \\ U^T &= LD^{1/2} \end{aligned}$$

$D^{1/2}$与L对角元相同

增长因子 $$\rho = \frac{max\{|U^T(i,j)|\}}{max\{|a_{ij}|\}} = \frac{max\{|l_{ij}l_{jj}|\}}{max\{|a_{ij}|\}} \le \frac{max\{l_{ij}^2\}}{max\{a_{ii}\}} \le 1$$

A的对角元是L一行元素的平方和

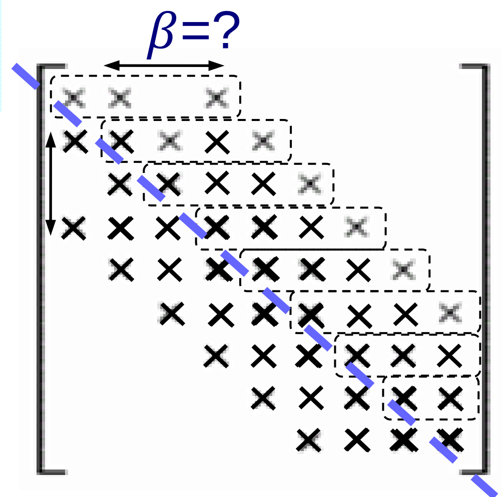
- 小结
  - 对于SPD矩阵, 可做Cholesky分解 (主元非0, 可开方)
  - 存储量、计算量都为一般的LU分解的一半
  - 算法数值稳定，不需要考虑选主元问题
  - **Matlab命令chol** (可检查对称矩阵的正定性)

带状矩阵解法与稀疏矩阵简介

带状线性方程组

- 带状矩阵
  - 三对角阵的扩展 β=?

$$\begin{bmatrix} \times & \times & & \\ \times & \times & \times & \\ & \times & \times & \times \\ & & \times & \times \end{bmatrix}$$



  - **定义3.13** 矩阵$A=(a_{ij})_{n\times n}$, 若$\forall i,j$, $|i-j|>\beta$时都有a_{ij}=0,且$\exists k$, $a_{k,k-\beta}\neq 0$或$a_{k,k+\beta}\neq 0$, 则称A为半带宽为β的带状矩阵(band matrix)
  - 非零元分布在主对角线及邻近的副对角线上, 最远副对角线到主对角线的“距离”就是 β
- 带状矩阵的LU分解
  - 结果L, U矩阵非零元仍分布在原始带宽范围内

带状线性方程组

- 三对角矩阵的LU分解(不考虑选主元)

$$\begin{bmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & a_n & b_n \end{bmatrix} = \begin{bmatrix} 1 & & & \\ m_2 & 1 & & \\ & \ddots & \ddots & \\ & & m_n & 1 \end{bmatrix} \begin{bmatrix} d_1 & c_1 & & \\ & \ddots & \ddots & \\ & & d_{n-1} & c_{n-1} \\ & & & d_n \end{bmatrix}$$

$\begin{bmatrix} a_{11} \\ a_{21} \end{bmatrix}$

For k=1, 2, …, n-1
    For i=k+1, ~~k+2, …, n~~
        $a_{ik} := a_{ik}/a_{kk}$; {L矩阵的元素}
        For j=k+1, ~~k+2, …, n~~
            $a_{ij} := a_{ij} - a_{ik}\underline{a_{kj}}$;
        End
    End
End

计算量: M.D.= 2n-2

算法3.11: 三对角矩阵的LU分解

输入: n, 向量$\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$; 输出: 向量$\boldsymbol{m}, \boldsymbol{d}$.

$d_1 := b_1$;
For i=2, 3, …, n
    $m_i := a_i/d_{i-1}$;
    $d_i := b_i - m_i c_{i-1}$;
End

- 用向量$\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$表示$\boldsymbol{A}$
- 线性复杂度$\ll O(n^3)$

算法3.12 三对角线性方程组$\boldsymbol{Ax}=\boldsymbol{f}$的"**追赶法**"解法

输入： n, 向量 a,b,c,f

输出： x

For $i=2,3,\cdots,n$

$m_i:=a_i/b_{i-1}$;

$b_i:=b_i-m_ic_{i-1}$

$f_i:=f_i-m_if_{i-1}$

"追"

End

$x_n:=f_n/b_n$

For $i=n-1,n-2,\cdots,1$

$x_i:=(f_i-c_ix_{i+1})/b_i$

"赶"

End

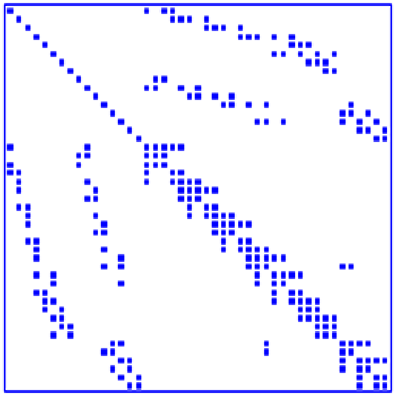
带状线性方程组

- 一般的带状矩阵(不考虑选主元)
  - 类似于算法3.11推导LU分解算法, 当$\beta \ll n$时效率很高! 时间复杂度$O(\beta^2 n)$, 空间复杂度$O(\beta n)$
  - $\boldsymbol{A}^{-1}, \boldsymbol{L}^{-1}, \boldsymbol{U}^{-1}$均稠密, 再次说明应避免算$\boldsymbol{A}^{-1}\boldsymbol{b}$
- 有些矩阵作LU分解不必选主元
  - 对称正定矩阵
  - 按行对角占优矩阵: (弱对角占优) 例: $\begin{bmatrix} 4 & 3 & 0 \\ 1 & 4 & 3 \\ 0 & 4 & 4 \end{bmatrix}$

    $$|a_{ii}| \geq \sum_{j=1, j\neq i}^{n} |a_{ij}|, i = 1,2, \dots, n$$ 且至少有一个大于号成立
  - 按列对角占优, 按行严格对角占优, 按列严格 … 不选主元也稳定
  - 定理3.20 按列严格对角占优阵, 列主元LU分解不需交换行

带状线性方程组

- 一般的带状矩阵(<u>需考虑选主元</u>)
  - 若采用部分选主元, L和U矩阵非零元分布将超出原始的带宽范围, 但它们到主对角线距离不超过2β
  - 当$\beta \ll n$, 选主元LU分解复杂度仍较低
- 一般的稀疏矩阵
  - 存在大量零元素的矩阵
  - **Wilkinson's definition: "matrices that allow special techniques to take advantage of the large number of zero elements."**
  - 强调用特殊技术节约计算量和存储空间(不处理零元素)
  - 若用二维数组来存, 即使有些零元素也不是稀疏矩阵



稀疏矩阵

$$A=\begin{bmatrix}1 & 0 & 2 & 0 & 0\\ 3 & 4 & 0 & 5 & 0\\ 6 & 0 & 7 & 0 & 8\\ 0 & 9 & 0 & 10 & 0\\ 0 & 0 & 0 & 0 & 11\end{bmatrix}$$

- 存储稀疏矩阵的数据结构
  - 三元组 (**COO格式**)

aa	1	2	3	4	5	6	7	8	9	10	11
row	1	1	2	2	2	3	3	3	4	4	5
col	1	3	1	2	4	1	3	5	2	4	5

    - 非零元在数组中顺序可以任意
    - 有点冗余: 一些连续存储的元素有相同的行(列)编号
  - 压缩稀疏行 (**CSR格式**)

aa	1	2	3	4	5	6	7	8	9	10	11
col	1	3	1	2	4	1	3	5	2	4	5
prow	1	3	6	9	11	12					

节省$N_{nz}-n$个整数

    - 非零元按第**1**行, 第**2**行, …, 第**n**行的顺序存储
    - **prow**为各行在数组中的开始位置

如何得每行非零元数目?

稀疏矩阵

- 存储稀疏矩阵的数据结构 (续)
  - 压缩稀疏行的变种: 压缩稀疏列(CSC); 用指针数组表示prow, 等等
  - 结构化稀疏阵
    - 若干个一维数组: 带状矩阵
    - 分块压缩稀疏行: 分块矩阵

广泛用于非

$$\left[\begin{array}{cc:cc:cc} 1 & 2 & 0 & 0 & 3 & 4 \\ 5 & 6 & 0 & 0 & 7 & 8 \\ \hdashline 0 & 0 & 9 & 10 & 11 & 12 \\ 0 & 0 & 13 & 14 & 15 & 16 \\ \hdashline 17 & 18 & 0 & 0 & 20 & 21 \\ 22 & 23 & 0 & 0 & 24 & 25 \end{array}\right]$$

例(稀疏矩阵的存储量) : 假设一个十万 (10^5) 阶矩阵 $\boldsymbol{A}$ 包含一百万 (10^6) 个非零元素 (平均每行 10 个非零元), 矩阵元素采用 IEEE 双精度浮点数存储, 试分析使用稠密矩阵存储和压缩稀疏行存储分别需要多少内存量。

一个 IEEE 双精度浮点数占 8B (字节), 一个 IEEE 整型数据占 4B 。

【解】

- 采用稠密矩阵格式, 矩阵 $\boldsymbol{A}$ 的存储量为 $8 \times 10^5 \times 10^5 = 8 \times 10^{10} \approx 80\text{GB}$;
- 采用 CSR 格式, 三个数组的长度分别为 10^6、10^6 和 10^5, 而且后两个数组为整型数组, 因此总存储量为

  $$8 \times 10^6 + 4 \times 10^6 + 4 \times 10^5 = 12.4 \times 10^6 \approx 12.4\text{MB}\ 。$$

稀疏矩阵

- 有关稀疏矩阵的计算
  - 基本技巧: 不存储零元素, 避免与零作加、减、乘、除

    例: 计算$\boldsymbol{A}+\boldsymbol{B}$, 运算量为$O(\mathrm{nnz}(\boldsymbol{A})+\mathrm{nnz}(\boldsymbol{B}))$次, 而非$O(n^2)$, 因为只需遍历所有非零元
- 对稀疏矩阵做高斯消去过程
  - 填入, 填入元 (fill-in)
  - 造成稀疏矩阵存储结构变化
  - 降低稀疏度, 增大存储/计算量
  - 大规模稀疏线性方程组的直接解法涉及很复杂的算法

$$\begin{bmatrix} \mathrm{X} & \mathrm{X} & \mathrm{X} & \mathrm{X} \\ \mathrm{X} & \mathrm{X} & \mathrm{X} & \mathrm{X} \\ 0 & \mathrm{X} & \mathrm{X} & \mathrm{X} \\ 0 & \mathrm{X} & \mathrm{X} & \mathrm{X} \end{bmatrix}$$

Matlab “\”的内部算法

矩阵A (按如下顺序)	求解算法
稀疏的对角阵	右端项元素除以矩阵对角元
较稠密的带状方阵	部分主元的带状矩阵LU分解 (LAPACK软件包)
上三角或下三角矩阵	回代法或前代法
对三角矩阵作行排列形成的矩阵	重排序后用回代或前代法
对称矩阵, 且对角线元素大于零	尝试Cholesky分解算法(稠密矩阵: LAPACK, 稀疏矩阵: CHOLMOD), 若稠密、且不正定使用选主元的对称矩阵求解算法(LAPACK)
稠密上Hessenberg矩阵	消为上三角矩阵再回代求解
一般的稀疏方阵	针对稀疏矩阵的直接解法(UMFPACK)
一般的稠密方阵	部分主元LU分解(LAPACK)
不是方阵	通过矩阵的QR分解得到最小二乘解