# 数值分析(3)

计算机系 软件所 喻文健

# 第三章 线性方程组的直接解法

- 线性方程组求解是一个经典的数学问题
  - 新的挑战: 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
  - 数值线性代数的基础问题之一
- 本章内容
  - 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、及Matlab相关技术 (简介)

分为直接解法、迭代解法。
前者是理论上经过有限步计算能得到准确解的方法。

# 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*约束优化问题*

本章讨论$m = n$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)

复习: 线性空间, 线性相关(无关), 基, 维数, 内积;
非奇异矩阵, 线性方程组解的存在性与唯一性

# 线性方程组与矩阵的基本概念

- 特殊矩阵 (定义3.3) *主要关注非奇异方阵*

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

表示稀疏矩阵的Wilkinson图

  - 对称矩阵, 对称正定矩阵, 对称半正定矩阵

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

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

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

行列式$\det(A_k)$, 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)

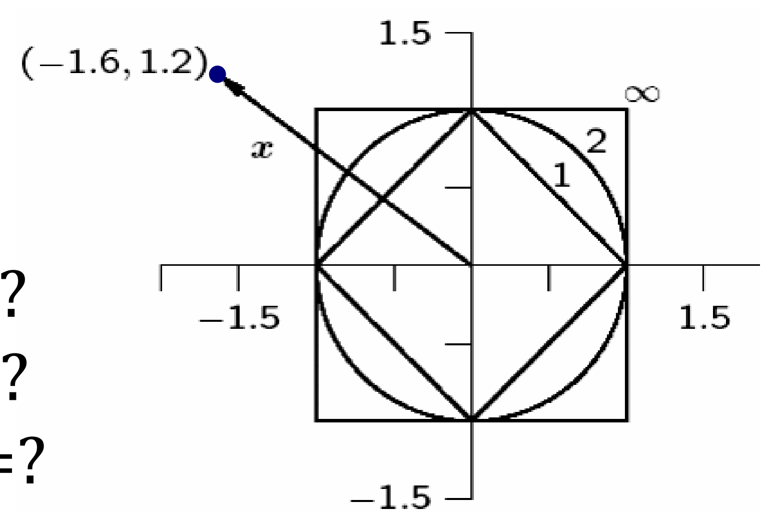
  数域$\mathbb{K}$上的线性空间$S$, 定义了范数, 称为赋范线性空间
- 向量的p-范数与内积范数
  - $\boldsymbol{x} = [x_1, x_2, \cdots, x_n]^T$, $\|\boldsymbol{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$ , $p \geq 1$
  - 内积范数: 针对内积$\langle \boldsymbol{x}, \boldsymbol{y} \rangle$, 定义范数$\|\boldsymbol{x}\| = \sqrt{\langle \boldsymbol{x},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

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

# 向量的范数

- 三种常用范数 常写作$l_1$-norm, $l_2$-norm, $l_\infty$-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：$\|\boldsymbol{x}\|_\infty=\max\limits_{1\le i\le n}|x_i|$ (“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$\boldsymbol{x}=[-1.6,1.2]^T$，

$\|\boldsymbol{x}\|_1=?$
$\|\boldsymbol{x}\|_2=?$
$\|\boldsymbol{x}\|_\infty=?$




(演示3.1)

# 向量有关的定理

- 定义3.8: 序列$\{\boldsymbol{x}^{(k)} \in \mathbb{R}^n\}$的极限$\lim_{k\to\infty} \boldsymbol{x}^{(k)} = \boldsymbol{x}^*$的含义
  - $\lim_{k\to\infty} x_i^{(k)} = x_i^*$ , $i$=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 $\lim_{k\to\infty} \boldsymbol{x}^{(k)} = \boldsymbol{x}^* \Leftrightarrow \lim_{k\to\infty} \left\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\right\| = 0$
- 说明 (Hint: 利用∞范数，课后思考)
  - 由定理3.7, 在某种范数下的有些结论(如向量序列范数收敛到0)在其他范数下也成立 不严格的$l_0$-norm: $\|\boldsymbol{x}\|_0$
  - 除p-范数外的其他范数: 如$\|\boldsymbol{x}\|_A = \sqrt{\boldsymbol{x}^T A \boldsymbol{x}}$, $\boldsymbol{A}$对称正定

# 矩阵的范数

- 矩阵也构成线性空间$\mathbb{R}^{n\times n}$
  - 矩阵加法、矩阵与实数乘法
  - 特殊性: 自我封闭的“矩阵乘”运算
  - 小测试: (不满足交换律)

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

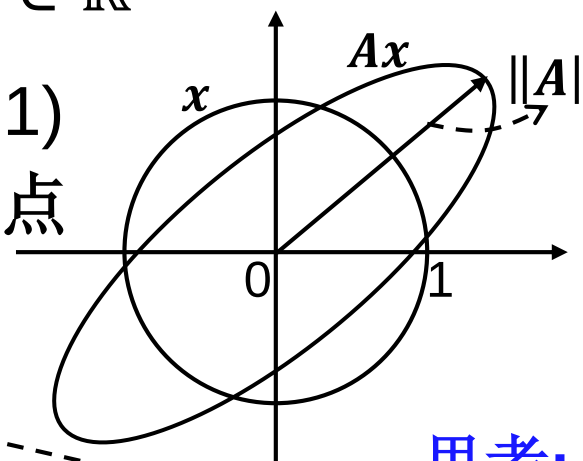
一般矩阵相乘的条件: $\underset{m\times s}{\boldsymbol{A}} \ \underset{s\times n}{\boldsymbol{B}} = \underset{m\times n}{\boldsymbol{AB}}$ $\quad [1\ 2]\begin{bmatrix} 0 \\ 3 \end{bmatrix} = ? \quad \begin{bmatrix} 0 \\ 3 \end{bmatrix}[1\ 2] = ?$

- $\mathbb{R}^{n\times n}$空间的矩阵范数 **(定义3.9)**
  - 增加对矩阵乘法的要求: $\|\boldsymbol{AB}\| \le \|\boldsymbol{A}\|\|\boldsymbol{B}\|$
  - 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)
  - 定义3.10 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




思考:
如何证明?

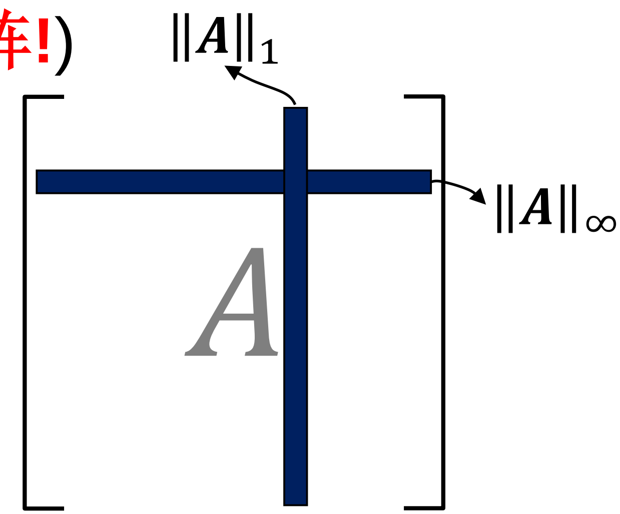
- 定理3.9 算子范数是满足相容性条件的范数
  - 证明满足一般范数定义;相容性条件;$\|\boldsymbol{AB}\|\le\|\boldsymbol{A}\|\|\boldsymbol{B}\|$

# 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: $\|\boldsymbol{A}\|_\infty = \max\limits_{1\le i\le n} \sum_{j=1}^{n} |a_{ij}|$
  - 1-, ∞-范数的公式请课后思考
  - 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-范数公式的证明自己看书 (对称半正定矩阵特征值分解的性质)
  - 2-范数的计算复杂得多, 常常避免计算
  - Frobenius范数: $\|\boldsymbol{A}\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}^2}$

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




思考: 它满足哪些范数的要求?

# 问题的敏感性

Wenjian Yu

# 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

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

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

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

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

# 矩阵的条件数

- 定义3.11 设$\boldsymbol{A}$为非奇异矩阵，称

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

为矩阵$\boldsymbol{A}$的条件数

- 说明: 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 (见课本定义3.11下面内容)

  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3.2 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

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

1-范数下cond=40000

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

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

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

符合“上界”的说法

# 矩阵的条件数

- 例3.3 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$\mathrm{cond}_\infty$ =1, $\mathrm{cond}(\boldsymbol{A})_\infty$ =2

良态问题！

- 例3.4 Hilbert矩阵

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

见课本

$$\mathrm{cond}(\boldsymbol{H}_3)_\infty=748$$
$$\mathrm{cond}(\boldsymbol{H}_4)_\infty=28375$$

随n增大, $\boldsymbol{H}_n$变得很病态!

# 矩阵条件数的性质

- **定理3.11** 在任一算子范数下:

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

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

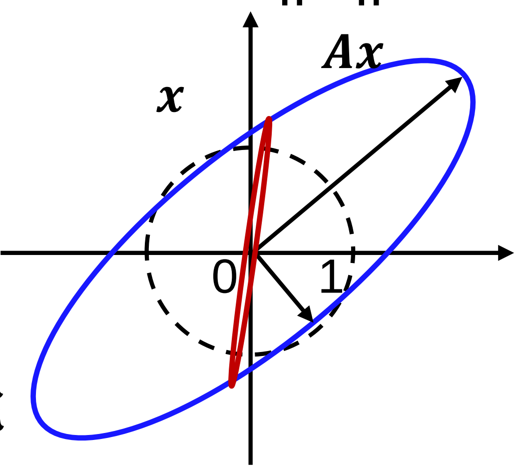
  …… ……

- 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. 定义cond(奇异矩阵) = +∞, 条件数

(演示3.2) 反映矩阵近奇异程度, 行列式则不能



# 矩阵条件数的性质

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

- 定理3.12 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! $\|\boldsymbol{QA}\| = \|\boldsymbol{A}\|$

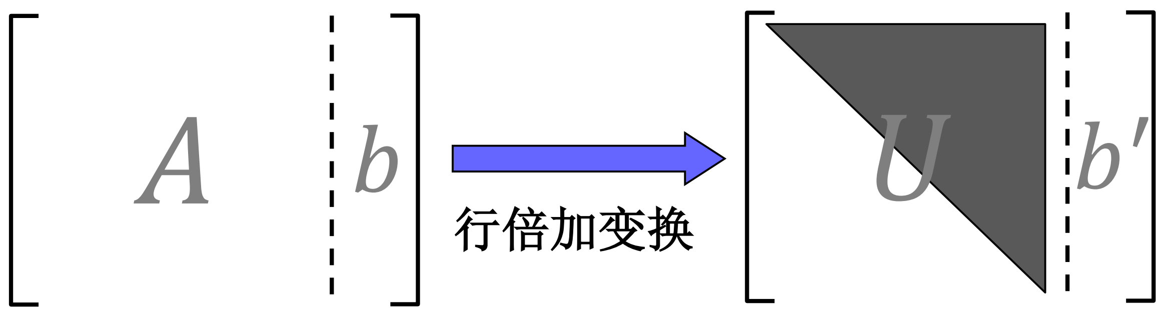
  $\mathrm{cond}(\boldsymbol{QA})_2 = \mathrm{cond}(\boldsymbol{AQ})_2 = \mathrm{cond}(\boldsymbol{A})_2$ 哪些范数下成立?

# 高斯消去法

Wenjian Yu

# 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(算法3.2)
  - 第k步消去过程可执行的条件：主元$a_{kk}^{(k)} \neq 0$
- 高斯-约当(Gauss-Jordan)消去法　　***(此小节不要求)***
  - 用初等变换将$A$变为单位阵，用于算逆矩阵(算法3.3)

# 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：A, n, b; 输出：A, 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:

**M.D.≈ $n^3/3$ ≈A.S.**

$(n-1)(n+1)+(n-2)n+(n-3)(n-1)+\cdots 1\times 3$

# 矩阵的LU分解

Wenjian Yu

# 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

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

  - 倍加变换矩阵$\boldsymbol{E}^{(2)}$是对单位阵$\boldsymbol{I}$的某一行乘以c加到另一行上得到的结果
  - $\boldsymbol{E}^{(2)}$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

是单位下三角阵

乘数

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

$\boldsymbol{M}_1\boldsymbol{A} = \boldsymbol{A}^{(2)}$, 更新的矩阵元素$a_{ij}^{(2)} = a_{ij}^{(1)} + m_{i1}a_{1j}^{(1)}, (i, j = 2,3)$

# 矩阵的LU分解

- 第2步消去

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

  - 等价于左乘消去矩阵

$$\boldsymbol{M}_2=\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & m_{32} & 1 \end{bmatrix}, \text{其中} m_{32}=-\frac{a_{32}^{(2)}}{a_{22}^{(2)}}$$

一次倍加变换

$\boldsymbol{M}_2\boldsymbol{A}^{(2)}=\boldsymbol{A}^{(3)}$, 更新元素$a_{33}^{(3)}=a_{33}^{(2)}+m_{32}a_{23}^{(2)}$

$\Rightarrow \boldsymbol{M}_2\boldsymbol{M}_1\boldsymbol{A}=\boldsymbol{A}^{(3)}\triangleq\boldsymbol{U} \Rightarrow \boldsymbol{A}=\boldsymbol{M}_1^{-1}\boldsymbol{M}_2^{-1}\boldsymbol{U}$ $\quad$ $\boldsymbol{U}$的对角元是?

# 矩阵的LU分解

$$\boldsymbol{A} = \boldsymbol{M}_1^{-1}\boldsymbol{M}_2^{-1}\boldsymbol{U}$$

- 消去矩阵的逆矩阵

定理 3.13

$$\boldsymbol{M}_1 = \begin{bmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & 0 & 1 \end{bmatrix}, \quad \boldsymbol{M}_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -m_{21} & 1 & 0 \\ -m_{31} & 0 & 1 \end{bmatrix}$$

逆矩阵也属于消去矩阵

$$\boldsymbol{M}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & m_{32} & 1 \end{bmatrix}, \quad \boldsymbol{M}_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -m_{32} & 1 \end{bmatrix}$$

下标反映了消去矩阵的类型数(定义3.12)

  - 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, $\boldsymbol{A} = \boldsymbol{M}_1^{-1}\boldsymbol{M}_2^{-1}\boldsymbol{U}$, 其中$\boldsymbol{M}_1^{-1}\boldsymbol{M}_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -m_{21} & 1 & 0 \\ -m_{31} & -m_{32} & 1 \end{bmatrix}$
- $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U}$, $\boldsymbol{L}$为单位下三角阵

矩阵$\boldsymbol{L}$

# 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$\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$, 其中$\boldsymbol{A}\in\mathbb{R}^{n\times n}$, 若执行高斯消去过程中的主元$a_{kk}^{(k)}\neq 0,(k=1,2,\cdots,n-1)$,

  只证 ⇒ ⇔系数矩阵$\boldsymbol{A}$存在唯一的LU分解　(充要条件)
- 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)

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!

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算矩阵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$m_{32} = -1/2$

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

# 矩阵的LU分解

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

Right-looking算法

- **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

    $a_{ik} := a_{ik}/a_{kk}$ ；  {$L$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:**

**M.D.**$\approx n^3/3$ $\approx$**A.S.**

$2n^3/3$ **flops**

# 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
  - $\boldsymbol{A}$的第一行: $u_{1j} = a_{1j}, (j = 1,2,3)$
  - $\boldsymbol{A}$的第一列: $a_{i1} = l_{i1}u_{11} \Rightarrow l_{i1} = a_{i1}/u_{11}, (i = 2, 3)$
  - $\boldsymbol{A}$的第二行: $a_{2j} = l_{21}u_{1j} + u_{2j} \Rightarrow u_{2j} = a_{2j} - l_{21}u_{1j}, (j=2,3)$
  - $\boldsymbol{A}$的第二列: $a_{32} = l_{31}u_{12} + l_{32}u_{22} \Rightarrow l_{32}=(a_{32} - l_{31}u_{12})/u_{22}$
  - $\boldsymbol{A}$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： $\boldsymbol{A}$, 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 {算$\boldsymbol{L}$的第k列}
    For j=1, 2, …, k-1 {用$\boldsymbol{U}$的第k列}
      $a_{ik} := a_{ik} - \boldsymbol{u}_{jk}\, \boldsymbol{l}_{ij}$;
    End
    $\boldsymbol{l}_{ik} := a_{ik} / \boldsymbol{u}_{kk}$ ;
  End {$\boldsymbol{U}$的第1行已知!}
  For j=k+1, k+2, …, n {算 $\boldsymbol{U}$ 的第k+1行}
    For i=1, 2, …, k {用$\boldsymbol{L}$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

$x = A^{-1}b$, 并非要算$A^{-1}$才能得$x$

实际上应避免算$A^{-1}$

- 单个方程求解
  - $Ax = b \Longrightarrow x = (LU)^{-1}b = U^{-1}L^{-1}b$
  - 1.解单位下三角型方程组$Ly = b$ $(y = L^{-1}b)$
  - 2.解上三角型方程组$Ux = y$
  - 计算量: $n^2$次乘除法
- 右端项变化的问题 (多右端方程组)
  - $Ax_i = b_i,\ i = 1, \dots, m$
  - 对每个右端项执行上述两步，不必重复算LU分解
- 三点用途：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

Wenjian Yu

# 主元为零的可能性

- 什么情况下高斯消去过程中不出现零主元?
  - 定理3.15 对矩阵$A \in \mathbb{R}^{n\times n}$，高斯消去过程中不出现零主元的充要条件是，$A$的前n-1个顺序主子式均不为零，即$\det(A_k) \neq 0, (k = 1, \dots, n-1)$.
  - 证明: ⟹ 已知不出现零主元，证明$\det(A_k) \neq 0$
  - $A^{(k)} = M_{k-1} \cdots M_2 M_1 A$
  - $A = \underbrace{M_1^{-1} M_2^{-1} \cdots M_{k-1}^{-1}}_{\text{单位下三角阵}} A^{(k)} = \begin{bmatrix} L_{11} & \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} a_{11}^{(1)} & \cdots & a_{1,k}^{(1)} & \cdots & a_{1,n}^{(1)} \\ & \ddots & \vdots & \cdots & \cdots \\ & & a_{k,k}^{(k)} & \cdots & a_{k,n}^{(k)} \\ & & \vdots & \cdots & \vdots \\ & & a_{n,k}^{(k)} & \cdots & a_{n,n}^{(k)} \end{bmatrix}$

    $A^{(k)}$
  - $A$的k阶顺序主子阵$A_k = L_{11} A_k^{(k)}$，$\det(A_k) = a_{11}^{(1)} \cdots a_{kk}^{(k)} \neq 0$
  - ⟸ 反证法. 设前k-1步主元非零, 而第k个主元为0…

# 主元为零的可能性

- 定理3.15的意义 (即是否出现零主元)
  - 根据前n-1个顺序主子式, 看高斯消去过程是否中断
  - 对$\det(A)=0$的奇异阵,也可能完成消去、LU分解 $\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix}$
  - 是LU分解存在且唯一的充要条件, 若不满足, LU分解<u>不存在</u>或<u>有不止一个</u>

$\begin{bmatrix}0 & 2\\ 3 & 1\end{bmatrix}$ 无LU分解　　$\begin{bmatrix}0 & 2\\ 0 & 1\end{bmatrix}$ 无穷多LU分解

  - **推论:** 对对称正定(负定)矩阵, 高斯消去不会中断!

高斯消去算法不中断

前n-1个顺序主子阵非奇异 → 矩阵可LU分解

**注意:**与矩阵$A$是否非奇异关系不大

# 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)

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

定理 3.16
  - 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

# 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
  - 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$i_k$,($i_k$>$k$)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, $i_k = k$

$$\begin{array}{c|ccccc|}
 & & k & & i_k & \\
 & & \downarrow & & \downarrow & \\
 & \ddots & & & & \\
k \rightarrow & & 0 & & 1 & \\
 & & & \boldsymbol{P}_k \ \ddots & & \\
i_k \rightarrow & & 1 & & 0 & \\
 & & & & & \ddots
\end{array}$$

# 使用部分主元技术的LU分解

- 部分主元消去过程: $\boldsymbol{M}_{n-1}\boldsymbol{P}_{n-1}\cdots\boldsymbol{M}_2\boldsymbol{P}_2\boldsymbol{M}_1\boldsymbol{P}_1\boldsymbol{A}=\boldsymbol{U}$
  - $\boldsymbol{P}_k,(k=1,\dots n-1)$实现第$k$与第$i_k$行($i_k\geq k$)的交换 ($\boldsymbol{P}_k$可能为$\boldsymbol{I}$)
  - 两边同乘$\boldsymbol{M}_{n-1}^{-1}$, $\Rightarrow$ $\boldsymbol{P}_{n-1}\boldsymbol{M}_{n-2}\boldsymbol{P}_{n-2}\cdots\boldsymbol{M}_1\boldsymbol{P}_1\boldsymbol{A}=\boldsymbol{M}_{n-1}^{-1}\boldsymbol{U}$

$\Rightarrow$ $\underbrace{\boldsymbol{P}_{n-1}\boldsymbol{M}_{n-2}(\boldsymbol{P}_{n-1}}_{\text{第n-1行与}i_{n-1}\text{行}(i_{n-1}\geq\text{n-1})\text{交换后，再做同样的列交换}}\boldsymbol{P}_{n-1})\boldsymbol{P}_{n-2}\cdots\boldsymbol{M}_1\boldsymbol{P}_1\boldsymbol{A}=\boldsymbol{M}_{n-1}^{-1}\boldsymbol{U}$

第n-1行与$i_{n-1}$行($i_{n-1}$≥n-1)交换后，再做同样的列交换

  - 记$\bar{\boldsymbol{M}}_{n-2}=\boldsymbol{P}_{n-1}\boldsymbol{M}_{n-2}\boldsymbol{P}_{n-1}$，它也是消去阵 $\Rightarrow$

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

$\boldsymbol{P}_{n-1}\boldsymbol{P}_{n-2}\boldsymbol{M}_{n-3}\boldsymbol{P}_{n-3}\cdots\boldsymbol{M}_1\boldsymbol{P}_1\boldsymbol{A}=\bar{\boldsymbol{M}}_{n-2}^{-1}\boldsymbol{M}_{n-1}^{-1}\boldsymbol{U}$

$\Rightarrow$ $\underbrace{\boldsymbol{P}_{n-1}\boldsymbol{P}_{n-2}\boldsymbol{M}_{n-3}(\boldsymbol{P}_{n-2}\boldsymbol{P}_{n-1}}\boldsymbol{P}_{n-1}\boldsymbol{P}_{n-2})\boldsymbol{P}_{n-3}\cdots\boldsymbol{M}_1\boldsymbol{P}_1\boldsymbol{A}=\cdots$

# 使用部分主元技术的LU分解

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

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

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

  记$L = \overline{M}_1^{-1}\cdots \overline{M}_{n-2}^{-1}M_{n-1}^{-1}$，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

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

- 部分主元的LU分解: $PA = LU$ 　　例: 一个3阶排列阵
- 矩阵$L$: $\overline{M}_k$,($k$=1, …, n-2)及$M_{n-1}$元素取相反数再合并

  $\overline{M}_k = P_{n-1}\cdots P_{k+1}M_kP_{k+1}\cdots P_{n-1}, (k \le n-2)$，将后续的行交换作用于$M_k$的第$k$列元素便得到$\overline{M}_k$

# 使用部分主元技术的LU分解

*(Java演示)*

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

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

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

  - $P$是对$I$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$P$的第$i$行是$I$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

# 部分选主元的LU分解算法 (算法3.9)

```
输入：A, n; 输出：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≤i≤n} |a_ik|的s;
    If s ≠ k then
        交换矩阵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计算量$2n^3/3$ flops
- 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

*(Matlab演示: lu)*

# 部分主元技术与其他主元技术

- 部分主元LU分解的应用
  - $Ax=b \Longrightarrow PAx=Pb \Longrightarrow \boxed{x=(LU)^{-1}Pb=U^{-1}L^{-1}Pb}$
  - 先对右端项重排, 再执行前代、回代过程

不要算$A^{-1}$！

- 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

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

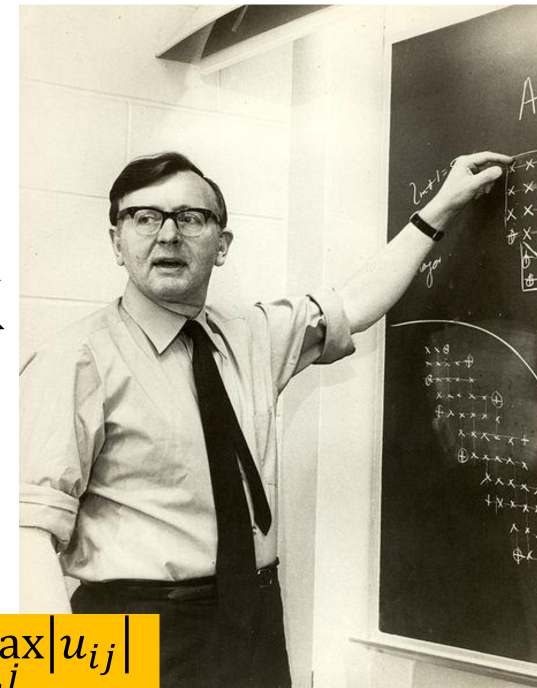
  - 乘数更小, 但开销增大
  - 矩阵形式: $PAQ=LU$, $Q$为排列阵

(Java演示)

- 通过选主元, 得到实用的高斯消去法、高斯-约当消去法

# 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$\hat{\boldsymbol{x}}$, 等效地, $(\boldsymbol{A}+\Delta\boldsymbol{A})\hat{\boldsymbol{x}}=\boldsymbol{b}$
  - 需分析$\|\Delta\boldsymbol{A}\|/\|\boldsymbol{A}\|$.对各种LU分解算法, 有结论: $\frac{\|\Delta\boldsymbol{A}\|_\infty}{\|\boldsymbol{A}\|_\infty}\lesssim n\rho\varepsilon_{\mathrm{mach}}$
  - $\rho$为$\boldsymbol{A}^{(k)}$(或者$\boldsymbol{U}$)与$\boldsymbol{A}$最大元素之比 $\frac{\max_{i,j}|u_{ij}|}{\max_{i,j}|a_{ij}|}$



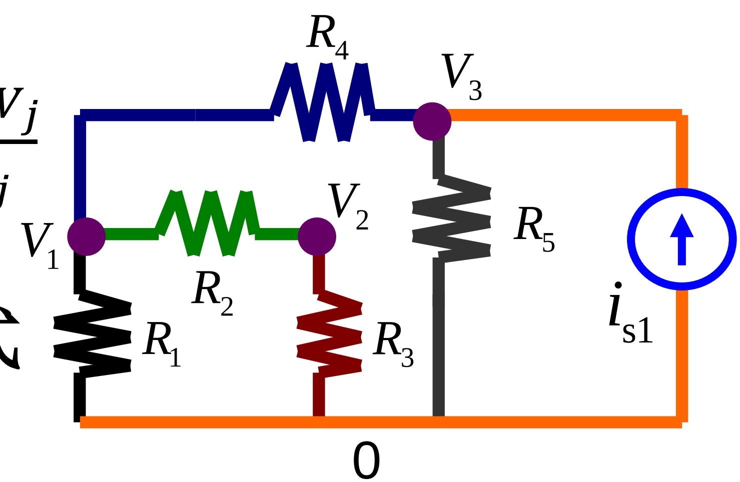
- **增长因子$\rho$的上限**
  - 部分选主元: $\rho\le 2^{n-1}$, 但一般**<10**, 是实用的稳定算法 **(思考题)**
  - 不选主元: $\rho$可能任意大

# 对称正定矩阵的Cholesky分解

Wenjian Yu

# 应用实例

- 稳态电路方程的求解
  - 是求解动态电路响应的基础, 仅考虑电阻元件
  - 节点分析法(nodal analysis)
  - 节点电压为变量, 电流$i_{kj}=\frac{V_k-V_j}{R_{kj}}$
  - 基尔霍夫电流定律: 节点流出电流总和为0, 列出电压关系式




$$\begin{bmatrix} \frac{1}{R_1}+\frac{1}{R_2}+\frac{1}{R_4} & -\frac{1}{R_2} & -\frac{1}{R_4} \\ -\frac{1}{R_2} & \frac{1}{R_2}+\frac{1}{R_3} & 0 \\ -\frac{1}{R_4} & 0 & \frac{1}{R_4}+\frac{1}{R_5} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ i_{s1} \end{bmatrix}$$

- 系数矩阵$A$对称
- 若电压$V\neq 0$, $V^TAV$为电路总功率, 则必定>0
- 矩阵$A$对称正定

# 对称矩阵的LU分解

*希望快速求解对称或对称正定矩阵*

- $LDL^T$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: $A=LDU_0$, $=A^T=U_0^TDL^T$
  - 若高斯消去不中断, LU分解唯一($L$=$U_0^T$), 则$A=LDL^T$
- Cholesky分解
  - 对称正定矩阵, $A=LDL^T$唯一地存在
  - $D=L^{-1}AL^{-T}$, 所以也是对称正定阵 ⇒ $D$的对角元$u_{ii}$>0

# 对称矩阵的LU分解

- **Cholesky分解 (续)**
  - 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, 但计算量同LU分解 ( ~ 2n³/3 flops)
  - 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

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

$L$ $L^T$

- 算法3.10 (按从1到n的顺序逐列算出L的元素值)

For $j$ =1 to n

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

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

若$A = LL^T$,

$$a_{jj} = l_{j1}^2 + \cdots + l_{jj}^2$$

- 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)

M.D.≈ $n^3/6$ ≈A.S.

$\sim n^3/3$ flops

$$(n-1)\times 1 + (n-2)2 + (n-3)3 + \cdots 1\times(n-1)$$

# Cholesky分解算法

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

- 通过LU分解算法的稳定性来分析
  - 对对称正定阵$A$做LU分解

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

$D^{1/2}$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

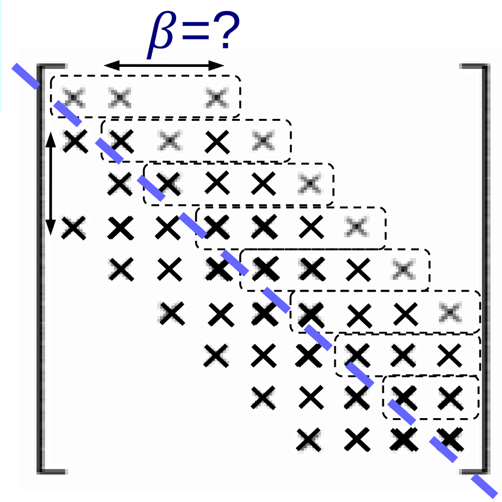
- 小结
  - 对于SPD矩阵, 可做Cholesky分解 (主元非0, 可开方)
  - 存储量、计算量都为一般的LU分解的一半
  - 算法数值稳定，不需要考虑选主元问题　*(Matlab演示)*
  - **Matlab命令chol**　(可检查对称矩阵的正定性)

# 带状矩阵解法与稀疏矩阵

Wenjian Yu

# 带状线性方程组

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




- 带状矩阵
  - □ 三对角阵的扩展　$\beta$=?
  - □ 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$\beta$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 $\beta$
- 带状矩阵的LU分解　　思考: $L^{-1}$, $U^{-1}$的非零元分布呢?
  - □ 结果$L$, $U$矩阵非零元仍分布在<u>原始带宽范围内</u>

# 带状线性方程组

- 三对角矩阵的LU分解(不考虑选主元)

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

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 (*追赶法*)

# 带状线性方程组

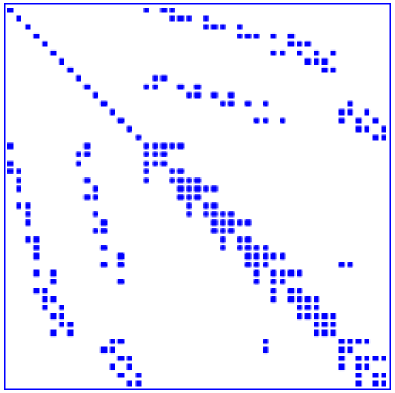
- 一般的带状矩阵(不考虑选主元)
  - 类似于算法3.11推导LU分解算法, 当$\beta \ll n$时效率很高! 时间复杂度$O(\beta^2 n)$, 空间复杂度$O(\beta n)$
  - $\boldsymbol{A}^{-1}, \boldsymbol{L}^{-1}, \boldsymbol{U}^{-1}$均稠密, 再次说明应避免算 $\boldsymbol{A}^{-1} \cdot \boldsymbol{b}$
- 有些矩阵作LU分解不必选主元
  - 对称正定矩阵
  - 定义3.14 按行对角占优矩阵: (弱对角占优) 例: $\begin{bmatrix} 4 & 3 & 0 \\ 1 & 4 & 3 \\ 0 & 4 & 4 \end{bmatrix}$

    $$|a_{ii}| \geq \sum_{j=1, j\neq i}^{n} |a_{ij}|, i = 1,2,\ldots,n$$ 且至少有一个大于号成立

    (加不可约则保证矩阵非奇异, 见 § 4.2)
  - 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(需考虑选主元)
  - 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$2\beta$
  - 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

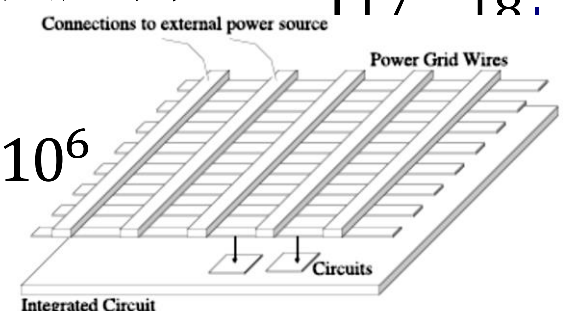
- 存储稀疏矩阵的数据结构 (续)

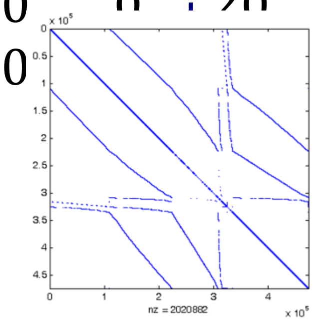
广泛用于非结构化稀疏阵

  - 压缩稀疏行的变种: 压缩稀疏列(CSC); 用指针数组表示prow, 等等
  - 若干个一维数组: 带状矩阵
  - 分块压缩稀疏行: 分块矩阵

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

  - 例: 供电网络分析




  - $n \approx 4.7\times10^5, N_{nz} \approx 2.0\times10^6$
  - CSR格式的数据量: $\approx 8N_{nz}+4N_{nz}+4n \quad \approx 26\times10^6 \approx 26$ **MB**
  - 若用二维数组, 需要约$1.8\times10^{12}$字节, 即**1.8TB**

*不便于访问单个矩阵元素; 但实际中往往访问一系列元素(各种矩阵算法)*

# 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

思考: 按稀疏矩阵存储格式写相应算法

- 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

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

# Matlab中的相关技术

Wenjian Yu

# Matlab中的矩阵

- 存储方式
  - 稠密矩阵(二维数组)、稀疏矩阵(压缩稀疏列)
  - 以相同方式支持矩阵操作: 通过whos看两者区别

| Name | Size | Bytes | Class | Attributes |
|---|---|---|---|---|
| A | 4x3 | 96 | double | |
| sA | 4x3 | 48 | double | sparse |

- 稠密矩阵/常规命令
  - 生成矩阵: ones(m,n), zeros(m,n), eye(m,n), rand(m,n)
  - 矩阵维数等: size(A), diag(A)/diag(v), tril(A), triu(A)
  - 载入/导出数据文件 (ascii或binary格式): load, save
- 稀疏矩阵的常用命令
  - 非零元的信息: nnz(A), spy(A), find(A)
  - 生成与转换: sparse(X) or sparse(i,j,s,m,n), X = full(A)

# Matlab中的矩阵

- 稀疏矩阵的命令 (续)
  - 生成特殊矩阵: speye, spones(S), spdiags(B, d, m, n)
  - 生成随机矩阵: sprand(m, n, density), sprandn
- 与线性方程组直接解法有关的命令
  - lu, chol
  - 反斜线运算符 “\” （mldivide）
  - >> x= A \ B;    % 求解$\boldsymbol{Ax} = \boldsymbol{B}$，$\boldsymbol{B}$为向量或矩阵
  - “\” 会根据矩阵A的特点自动选择合适的直接解法
  - linsolve与“\”功能类似, 可设置矩阵特点 (免去检测), 但不支持稀疏矩阵

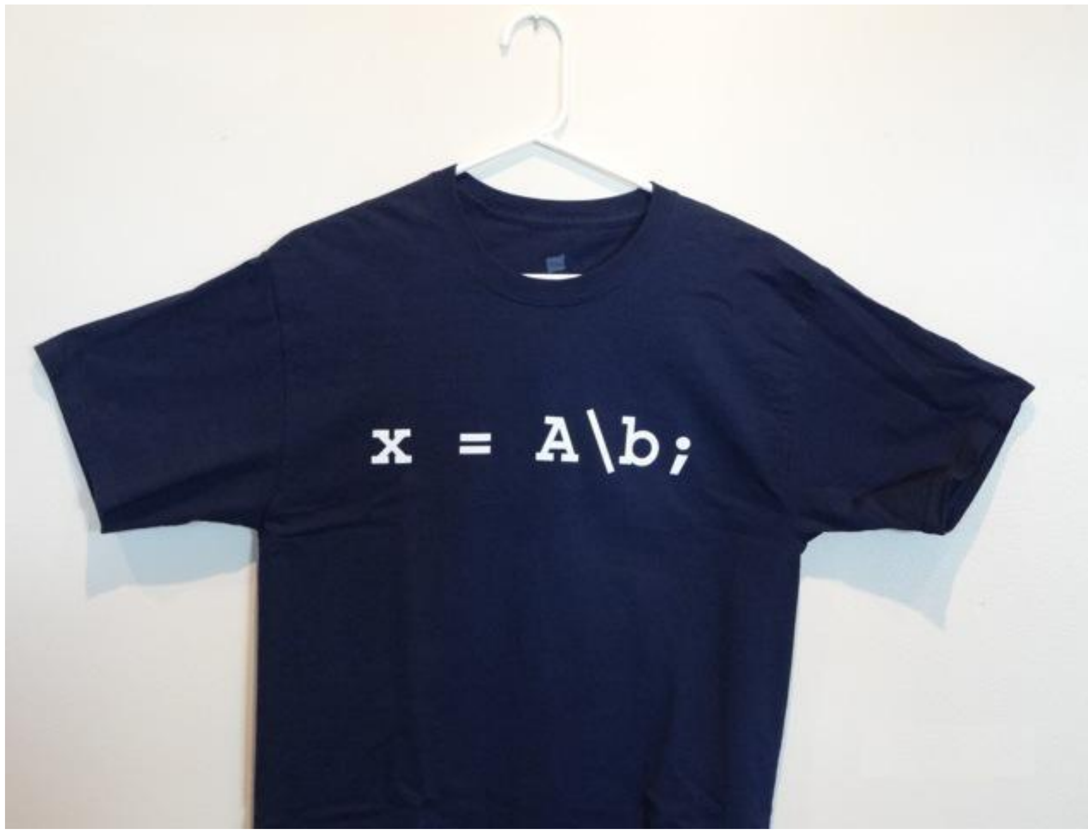
# Matlab “\”的内部算法

| 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 |

*美国TAMU, Prof. Tim Davis, http://faculty.cse.tamu.edu/davis/*

Cleve's Corner: Cleve Moler on Mathematics and Computing

Scientific computing, math & more




"It's amazing how much numerical linear algebra is contained in that single character"

# 非线性方程组求解

（§2.7）

Wenjian Yu

# 非线性方程组的求解

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0} \qquad \boldsymbol{x} \in \mathbb{R}^n, \text{函数}\boldsymbol{f}: \mathbb{R}^n \to \mathbb{R}^n$$

- 不动点迭代法仍是可行思路 $\boldsymbol{x} = \boldsymbol{g}(\boldsymbol{x})$
- 收敛性如何判断? $\boldsymbol{x}_{k+1} = \boldsymbol{g}(\boldsymbol{x}_k)$, k= 0,1,2 …
  - 对单个方程问题, (局部)收敛性由$|g'(x^*)|$决定
- 定义2.4 多元向量函数$\boldsymbol{g}: \mathbb{R}^n \to \mathbb{R}^n$的雅可比矩阵$\boldsymbol{J}_g(\boldsymbol{x})$为n阶方阵, 元素值为$\{\boldsymbol{J}_g(\boldsymbol{x})\}_{ij} = \frac{\partial g_i(\boldsymbol{x})}{\partial x_j}, (i, j = 1,2, \cdots, n)$
- 定理2.10 不动点迭代法$\boldsymbol{x}_{k+1} = \boldsymbol{g}(\boldsymbol{x}_k)$, 设$\boldsymbol{x}^*$为准确解, 若雅可比矩阵$\boldsymbol{J}_g(\boldsymbol{x}^*)$的特征值λ都满足$|\lambda| < 1$, 则该迭代法局部收敛

最大特征值<1

# 牛顿法解非线性方程组

解方程组$\boldsymbol{f}(\boldsymbol{x}) = \mathbf{0}$

- 思路
  - 对可微函数$\boldsymbol{f}: \mathbb{R}^n \to \mathbb{R}^n$, 在任一点$\boldsymbol{x}$处作一阶泰勒展开
    $\boldsymbol{f}(\boldsymbol{x}+\boldsymbol{s}) \approx \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{J}_f(\boldsymbol{x})\boldsymbol{s}$，当$\boldsymbol{x}=\boldsymbol{x}_k$，解$\boldsymbol{f}(\boldsymbol{x}_k)+\boldsymbol{J}_f(\boldsymbol{x}_k)\boldsymbol{s}=\mathbf{0}$
    $\Rightarrow \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{s} = \boldsymbol{x}_k - \left[\boldsymbol{J}_f(\boldsymbol{x}_k)\right]^{-1}\boldsymbol{f}(\boldsymbol{x}_k)$

单个方程牛顿法

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

**算法2.6** 解非线性方程组的牛顿法

**输入**：$\boldsymbol{x}_0$, n维多元函数$\boldsymbol{f}(\boldsymbol{x})$; **输出**：$\boldsymbol{x}$ .

$k := 0$ ;

**While** 不满足收敛条件 **do**

  解线性方程组$\boldsymbol{J}_f(\boldsymbol{x}_k)\boldsymbol{s}_k = \boldsymbol{f}(\boldsymbol{x}_k)$，求$\boldsymbol{s}_k$ ;

  $\boldsymbol{x}_{k+1} := \boldsymbol{x}_k - \boldsymbol{s}_k$ ;

  $k := k+1$ ;

**End**

$\boldsymbol{x} := \boldsymbol{x}_k$

Java演示

# 牛顿法解非线性方程组

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \left[J_f(\boldsymbol{x}_k)\right]^{-1} \boldsymbol{f}(\boldsymbol{x}_k)$$

- 例2.9
  - 用牛顿法解 $\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} x_1 + 2x_2 - 2 \\ x_1^2 + 4x_2^2 - 4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$，初值$\boldsymbol{x}_0 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$
- 解:

该方程的雅可比矩阵$J_f(\boldsymbol{x}) = \begin{bmatrix} 1 & 2 \\ 2x_1 & 8x_2 \end{bmatrix}$

算$J_f(\boldsymbol{x}_0) = \begin{bmatrix} 1 & 2 \\ 2 & 16 \end{bmatrix}$, $\boldsymbol{f}(\boldsymbol{x}_0) = \begin{bmatrix} 3 \\ 13 \end{bmatrix}$, 解方程 $J_f(\boldsymbol{x}_0)\boldsymbol{s}_0 = \boldsymbol{f}(\boldsymbol{x}_0)$

得$\boldsymbol{s}_0 = [1.83,\ 0.58]^T$, $\boldsymbol{x}_1 = \boldsymbol{x}_0 - \boldsymbol{s}_0 = \begin{bmatrix} -0.83 \\ 1.42 \end{bmatrix}$

算$J_f(\boldsymbol{x}_1) = \begin{bmatrix} 1 & 2 \\ -1.67 & 11.3 \end{bmatrix}$, $\boldsymbol{f}(\boldsymbol{x}_1) = \begin{bmatrix} 0 \\ 4.72 \end{bmatrix}$, 解方程$J_f(\boldsymbol{x}_1)\boldsymbol{s}_1 = \boldsymbol{f}(\boldsymbol{x}_1)$

得$\boldsymbol{s}_1 = [-0.64,\ 0.32]^T$, $\boldsymbol{x}_2 = \begin{bmatrix} -0.19 \\ 1.10 \end{bmatrix}$ …, $\boldsymbol{x}_3 = \begin{bmatrix} -0.02 \\ 1.01 \end{bmatrix}$, $\boldsymbol{x}_4 = \begin{bmatrix} -0.00 \\ 1.00 \end{bmatrix}$