数值计算方法：原理、算法和应用
Numerical Methods: Principles, Algorithms and Applications

授课教师：周铁

北京大学数学科学学院

2021 年 9 月 23 日

1. 线性方程组的直接解法
    - Gauss 消元法
    - 矩阵的 LU 分解
    - 带状矩阵与稀疏矩阵
    - 计算解的误差分析
        - 向量和矩阵的范数
        - 线性方程组的稳定性与条件数
        - LU 分解的数值稳定性

Gauss 消元法

Gauss 消元法 (Gaussian Elimination Method)

Gauss 消元法：增广矩阵化成阶梯形，初等行变换，主元 (pivot)

$$Ax = b: \quad \begin{bmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7 \\ 4 \\ 6 \end{bmatrix}$$

$$\begin{bmatrix} 10 & -7 & 0 & 7 \\ -3 & 2 & 6 & 4 \\ 5 & -1 & 5 & 6 \end{bmatrix} \to \begin{bmatrix} 10 & -7 & 0 & 7 \\ 0 & -0.1 & 6 & 6.1 \\ 0 & 2.5 & 5 & 2.5 \end{bmatrix} \to \begin{bmatrix} 10 & -7 & 0 & 7 \\ 0 & 2.5 & 5 & 2.5 \\ 0 & -0.1 & 6 & 6.1 \end{bmatrix} \to$$

$$\to \begin{bmatrix} 10 & -7 & 0 & 7 \\ 0 & 2.5 & 5 & 2.5 \\ 0 & 0 & 6.2 & 6.2 \end{bmatrix}$$

注意中间做了一次两行交换，一定要做行交换吗？

行交换的必要性

有些时候一定需要行交换！否则 Gauss 消元法都不一定能进行.

$$\begin{bmatrix} 0 & 3 & 1 & 9 \\ 1 & -2 & 1 & 0 \\ 3 & 3 & -1 & 6 \end{bmatrix}, \qquad \begin{bmatrix} 1 & 1 & 3 & 5 \\ 2 & 2 & 2 & 6 \\ 3 & 6 & 4 & 13 \end{bmatrix}$$

定理

如果 n 阶方阵 A 的前 $n-1$ 阶主子式都非零，则 Gauss 消元法一定可以进行，最后得到一个对角元素非零的上三角阵 U.

下面三类矩阵就满足定理条件：

- 对角占优矩阵.
- 实对称正定矩阵.
- 复正定矩阵.

但这不是主要原因！
主要原因：小主元会带来很大的计算误差！

在浮点数系统 $\mathbb{F}$ 中用 Gauss 消元法解方程组:

$$\begin{bmatrix} \epsilon & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \tag{1}$$

其中 ϵ 是很小的正数，小到浮点运算时 $1 \pm \epsilon = 1\,(\epsilon^{-1} \pm 1 = \epsilon^{-1})$.

不难看出方程组(1)的准确解为:

$$x = \begin{bmatrix} 1/(1-\epsilon) \\ (1-2\epsilon)/(1-\epsilon) \end{bmatrix} \approx \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

用 Gauss 消元法 (主元为 ϵ) 解方程组(1)得到

$$\begin{bmatrix} \epsilon & 1 \\ 0 & 1-\epsilon^{-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2-\epsilon^{-1} \end{bmatrix} \Longrightarrow \tilde{x} \approx \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

计算解的相对误差很大!

如果先交换一下方程组(1)的两个方程的位置

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

再用 Gauss 消元法 (主元为 1) 得到:

$$\begin{bmatrix} 1 & 1 \\ 0 & 1-\epsilon \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 1-2\epsilon \end{bmatrix} \Longrightarrow \tilde{x} \approx \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \text{ 很准!}$$

由于用非零数乘方程的两端不改变方程的解, 可以用 ϵ^{-1} 乘方程组(1)的第一个方程得到等价的方程组

$$\begin{bmatrix} 1 & \epsilon^{-1} \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \epsilon^{-1} \\ 2 \end{bmatrix} \tag{2}$$

这样就不用选主元?

用 Gauss 消元法 (主元为 1) 解方程组(2)

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

由于 $1(2) - \epsilon^{-1} = -\epsilon^{-1}$，得到的计算解还是:

$$\tilde{x} \approx \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

误差还是很大！这是因为在系数矩阵

$$\begin{bmatrix} 1 & \epsilon^{-1} \\ 1 & 1 \end{bmatrix}$$

中，$a_{11}(=1)$ 在第一行中仍然是相对很小的元素！

Gauss 消元法应该选比较大的数做主元–列选主元和全选主元.

矩阵的 LU 分解

Gauss 消元法和矩阵的 LU 分解

看一个没有行交换的例子： $Ax = b$

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 6 \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad b = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$

消元的第一步等价于矩阵乘法

$$L_1 A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & -1 & -2 \\ 0 & -2 & -3 \end{bmatrix}, \quad L_1 b = \begin{bmatrix} 4 \\ -3 \\ -6 \end{bmatrix}$$

其中

$$L_1 = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -3 & 0 & 1 \end{bmatrix}$$

消元的第二步等价于矩阵乘法

$$L_2(L_1A) = U = \begin{bmatrix} 1 & 2 & 3 \\ 0 & -1 & -2 \\ 0 & 0 & 1 \end{bmatrix}, \quad L_2(L_1b) = \begin{bmatrix} 4 \\ -3 \\ 0 \end{bmatrix}$$

其中

$$L_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix}$$

把两步消元合在一起，就有

$$L_2L_1A = U \quad \Rightarrow \quad A = (L_2L_1)^{-1}U = LU$$

其中 L 为单位下三角矩阵，U 为上三角矩阵.
并不是所有的矩阵都可以做 LU 分解，例如：

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}, \quad \begin{bmatrix} 2 & 2 & 1 \\ 1 & 1 & 1 \\ 3 & 2 & 1 \end{bmatrix} \quad \text{都不可能等于 } LU.$$

从 Gauss 消元法的过程可以看出：只要不遇到主元为零的情况，理论上就一定可以实现 LU 分解.

定理

任给可逆方阵 A，都存在排列阵 P，使得 $PA = LU$，其中 L 为单位下三角阵，U 为上三角阵.

- 即使不出现零主元，选主元也是必要的
- 太小的主元是数值不稳定的主要原因
- 选主元计算量不小

用 Scientific Computing with MATLAB and Octave 一书 149 页的 lugauss 程序 (不选主元 LU 分解) 做个试验.

$$A = \begin{bmatrix} 1 & 1+0.5e-15 & 3 \\ 2 & 2 & 20 \\ 3 & 6 & 4 \end{bmatrix}$$

```
>> A = [1  1+0.5e-15  3;  2  2  20;  3  6  4]
A =
 1.000000000000000    1.000000000000000    3.000000000000000
 2.000000000000000    2.000000000000000   20.000000000000000
 3.000000000000000    6.000000000000000    4.000000000000000

>> A(1,2)-1
 ans = 4.440892098500626e-16

>> B = lugauss(A)

>> L = eye(size(A)) + tril(B,-1)

>> U = triu(B)

>> L*U - A
 ans =
 0        0        0
 0        0        0
 0        0       -4
```

MATLAB 的 lu 命令是选主元的，数值稳定性比较好.

```
>> A = [1  1+0.5e-15  3;  2  2  20;  3  6  4]
>> [L,U,P] = lu(A)
L =
 1.000000000000000                    0                    0
 0.666666666666667    1.000000000000000                    0
 0.333333333333333    0.500000000000000    1.000000000000000
U =
 3.000000000000000    6.000000000000000    4.000000000000000
 0                   -2.000000000000000   17.333333333333332
 0                    0                   -6.999999999999996
P =
 0       0       1
 0       1       0
 1       0       0
>> L*U - P*A
 ans =
 0       0       0
 0       0       0
 0       0       0
```

LU 分解的 Doolittle 算法

$$\text{可逆阵} A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

$$u_{11} = a_{11}, u_{12} = a_{12}, u_{13} = a_{13}$$
$$l_{21} u_{11} = a_{21} \Rightarrow l_{21} = a_{21} / u_{11}$$
$$l_{31} u_{11} = a_{31} \Rightarrow l_{31} = a_{31} / u_{11}$$
$$l_{21} u_{12} + u_{22} = a_{22} \Rightarrow u_{22} = a_{22} - l_{21} u_{12}$$
$$l_{31} u_{12} + l_{32} u_{22} = a_{32} \Rightarrow l_{32} = (a_{32} - l_{31} u_{12}) / u_{22}$$
$$l_{31} u_{13} + l_{32} u_{23} + u_{33} = a_{33} \Rightarrow u_{33} = a_{33} - l_{31} u_{13} - l_{32} u_{23}$$

u_{11} 和 u_{22} 不能为零，也不能太小，否则需要选主元.

n 阶方阵做选主元 Doolittle 分解的计算量 $\approx \dfrac{2n^3}{3}$.

一旦有了列主元 LU 分解 $PA = LU$，解方程组 $PAx = Pb$ 就等价于解方程组 $LUx = Pb$，就等价于依次解方程组：

$$
\begin{aligned}
Ly &= Pb \\
Ux &= y
\end{aligned}
$$

解 n 阶上 (下) 三角形线性方程组的计算量是 n^2.

LU 分解还可用于：

- 计算行列式： $A = LU, \quad \det(A) = \det(L)\det(U)$.
- 计算矩阵的逆： 就是解矩阵方程 $AX = I$.

对称矩阵的 LDL^T 分解

设 $A \in \mathbb{K}^{n,n}$，并有 LU 分解：$A = LU$.
如果 $A^T = A$(即 A 是对称阵)，则必有

$$LU = A = A^T = U^T L^T$$

由于 L 的对角元素都为 1，它必可逆，于是有

$$U(L^T)^{-1} = L^{-1} U^T \tag{3}$$

利用

- 两个同阶上 (下) 三角阵的乘积仍然是同阶的上 (下) 三角矩阵.
- 可逆上 (下) 三角阵的逆矩阵仍然是上 (下) 三角矩阵.

(3)式左边为上三角阵，右边为下三角阵，所以两边都只能是对角阵，设为 D. 由此可推出 $U = DL^T$，最后就得到

$$A = LDL^T$$

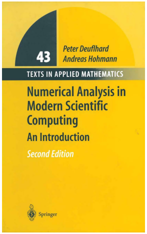
这就是对称矩阵 A 的 LDL^T 分解.

实对称正定矩阵

定理，the following book, pp.14

设 $A=(a_{ij})\in\mathbb{R}^{n,n}$ 为对称正定阵，则有

- A 必可逆.
- A 的对角元 a_{ii} 都是正数.
- $\max\limits_{i,j=1,\dots,n}|a_{ij}|=\max\limits_{i=1,\dots,n}a_{ii}$.
- 做完一次不选主元 Gauss 消元法后，剩下的低一阶的方阵还是对称正定矩阵.



实对称正定矩阵的 Cholesky 分解

Cholesky 分解定理

设 $A \in \mathbb{R}^{n,n}$ 为对称正定阵，则存在惟一的上三角阵 $U \in \mathbb{R}^{n,n}$(对角元全为正数)，使得

$$A = U^T U.$$

如果记 $\tilde{L} = U^T$，则有惟一分解

$$A = \tilde{L}\tilde{L}^T.$$

其中 $\tilde{L}$ 为对角元全为正数的下三角矩阵.

- 在 LDL^T 分解后取 $U = D^{1/2}L^T$，或者 $\tilde{L} = LD^{1/2}$ 即可得到上述 Cholesky 分解.
- 直接用矩阵乘法实现的效率比较高.
- 不用选主元，数值稳定性比较好.

Cholesky 分解的矩阵乘法实现

写成 $A = LL^T$ 形式，以 3 阶为例.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{bmatrix}$$

$$\begin{aligned}
l_{11}^2 &= a_{11} \Rightarrow l_{11} = \sqrt{a_{11}} \\
l_{11} l_{21} &= a_{12} \Rightarrow l_{21} = a_{12}/l_{11} \\
l_{11} l_{31} &= a_{13} \Rightarrow l_{31} = a_{13}/l_{11} \\
l_{21}^2 + l_{22}^2 &= a_{22} \Rightarrow l_{22} = \sqrt{a_{22} - l_{21}^2} \\
l_{21} l_{31} + l_{22} l_{32} &= a_{23} \Rightarrow l_{32} = (a_{23} - l_{21} l_{31})/l_{22} \\
l_{31}^2 + l_{32}^2 + l_{33}^2 &= a_{33} \Rightarrow l_{33} = \sqrt{a_{33} - l_{31}^2 - l_{32}^2}
\end{aligned}$$

写成 $A = U^T U$ 形式，U 的元素可由下列公式顺序计算.

$$u_{11} = \sqrt{a_{11}}$$

对 $j = 2, \ldots, n$

$$u_{ij} = \frac{1}{u_{ii}}\left(a_{ij} - \sum_{k=1}^{i-1} u_{ki}u_{kj}\right), \quad i = 1, \ldots, j-1$$

$$u_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{i-1} u_{kj}^2}$$

计算量为 $\frac{1}{3}n^3$.

- Cholesky 分解是判别实对称矩阵是否正定的方法.

$$\exists j,\ a_{jj} - \sum_{k=1}^{i-1} u_{kj}^2 \leq 0 \Longrightarrow A \text{ 不正定}$$

- 可以推广到复矩阵，设 $A \in \mathbb{C}^{n,n}$ 且正定，则有惟一分解 $A = LL^H$.

```
>> A=gallery('moler',5)
A =
  1     -1     -1     -1     -1
 -1      2      0      0      0
 -1      0      3      1      1
 -1      0      1      4      2
 -1      0      1      2      5

>> chol(A)
 ans =
 1     -1     -1     -1     -1
 0      1     -1     -1     -1
 0      0      1     -1     -1
 0      0      0      1     -1
 0      0      0      0      1
```

用 Cholesky 分解判断对称矩阵是否正定

```
>> A = [7,5,5,8;5,6,9,7;5,9,9,0;8,7,0,1]
A =
7      5      5      8
5      6      9      7
5      9      9      0
8      7      0      1

>> [U,p] = chol(A)
U =
2.6458    1.8898
0         1.5584
p = 3
```

```
>> B = gallery('moler',5)
>> [U p]=chol(B)
U =
1    -1    -1    -1    -1
0     1    -1    -1    -1
0     0     1    -1    -1
0     0     0     1    -1
0     0     0     0     1
p = 0
>> [L U] = lu(sym(B))
```

带状矩阵与稀疏矩阵的 LU 分解

带状矩阵

只在主对角线和少数几条次对角线上有非零元素的矩阵称为**带状矩阵**. 如果主对角线上方有 p 条非零次对角线，则称其上带宽为 p. 类似地，有下带宽为 q 的说法，和总带宽为 $p+q+1$ 的说法.

$$A = \begin{pmatrix} 2 & 1 & -1 & 0 & 0 & 0 & 0 \\ -4 & 2 & 3 & 0 & 0 & 0 & 0 \\ 0 & -12 & 3 & 1 & 2 & 0 & 0 \\ 0 & 0 & -24 & 4 & -7 & 0 & 0 \\ 0 & 0 & 0 & -40 & 5 & 1 & 4 \\ 0 & 0 & 0 & 0 & -60 & 6 & -23 \\ 0 & 0 & 0 & 0 & 0 & -84 & 7 \end{pmatrix}.$$

A 的上带宽 $p=2$，下带宽为 $q=1$，总带宽为 $p+q+1=4$.

带状矩阵 LU 分解

定理

设 $A \in \mathbb{K}^{n,n}$ 为上带宽为 p，下带宽为 q 的带状方阵，$A = LU$ 为不选主元 LU 分解，则 L 的下带宽为 q，U 的上带宽为 p. 此时 LU 分解的计算量为 $\mathcal{O}(pqn)$. 如果做列主元 LU 分解，则 L 为每一列最多有 $q+1$ 个非零元的下三角阵，U 为上带宽为 $p+q$ 的带状上三角阵.

- 解 $Ly = b$ 的计算量为 $\mathcal{O}(pn)$，解 $Ux = y$ 的计算量为 $\mathcal{O}(qn)$.
- 解 $n \times n$ 带状线性方程组的工作量为 $\mathcal{O}(pqn)$.

$$A = \begin{pmatrix} 2 & 1 & -1 & 0 & 0 & 0 & 0 \\ -4 & 2 & 3 & 0 & 0 & 0 & 0 \\ 0 & -12 & 3 & 1 & 2 & 0 & 0 \\ 0 & 0 & -24 & 4 & -7 & 0 & 0 \\ 0 & 0 & 0 & -40 & 5 & 1 & 4 \\ 0 & 0 & 0 & 0 & -60 & 6 & -23 \\ 0 & 0 & 0 & 0 & 0 & -84 & 7 \end{pmatrix}.$$

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -3 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -5 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -6 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -7 & 1 \end{pmatrix} \quad U = \begin{pmatrix} 2 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 4 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 6 & 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 8 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 1 & 4 \\ 0 & 0 & 0 & 0 & 0 & 12 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 14 \end{pmatrix}.$$

```
>> A = diag([-1,0,2,0,4],2)  ...
 + diag([1,3,1,-7,1,-23],1)  ...
 + diag([2,2,3,4,5,6,7],0)...
 + diag([-4,-12,-24,-40,-60,-84],-1)

>> A = sym(A)

>> [L,U] = lu(A)

>> B = StoreBandMatrix(A,1,2)
```

带状矩阵的存储

为节省存储空间，只要把总带宽为 $p+q+1$ 的带状矩阵 A 的非零元存储在一个 $n \times (p+q+1)$ 矩阵 B 里面. 参考书

> W. Gander, M. J. Gander, F. Kwok, Scientific Computing: an introduction using Maple and MATLAB,

里给出了一个 Matlab 程序 StoreBandMatrix.

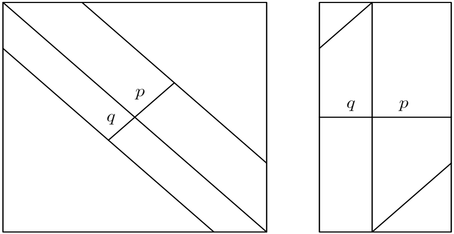


FIGURE 3.5. *Storing of banded matrices*

```
function B = StoreBandMatrix(A,q,p)
% StoreBandMatrix stores the band of a matrix in a
% rectangular matrix B = StoreBandMatrix(A) stores
% the band of A  (with lower bandwidth p and upper
% bandwidth q) in the rectangular matrix B of
% dimensions n*p+q+1.

n = length(A);
B = zeros(n,p+q+1); % reserve space
for i=1:n
for j=max(1,i-q):min(n,i+p)
B(i,j-i+q+1)=A(i,j);
end
end
```

$$A = \begin{pmatrix} 2 & 1 & -1 & 0 & 0 & 0 & 0 \\ -4 & 2 & 3 & 0 & 0 & 0 & 0 \\ 0 & -12 & 3 & 1 & 2 & 0 & 0 \\ 0 & 0 & -24 & 4 & -7 & 0 & 0 \\ 0 & 0 & 0 & -40 & 5 & 1 & 4 \\ 0 & 0 & 0 & 0 & -60 & 6 & -23 \\ 0 & 0 & 0 & 0 & 0 & -84 & 7 \end{pmatrix}.$$

```
B =
     0     2     1    -1
    -4     2     3     0
   -12     3     1     2
   -24     4    -7     0
   -40     5     1     4
   -60     6   -23     0
   -84     7     0     0
```

三对角矩阵

$p = q = 1$ 的带状矩阵称为三对角矩阵，比如

$$A = \begin{bmatrix} a_1 & c_1 & & & \\ b_2 & a_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & b_{n-1} & a_{n-1} & c_{n-1} \\ & & & b_n & a_n \end{bmatrix}$$

这种矩阵的不选主元 LU 分解为：

$$A = LU = \begin{bmatrix} 1 & & & \\ l_2 & 1 & & \\ & \ddots & \ddots & \\ & & l_n & 1 \end{bmatrix} \begin{bmatrix} u_1 & c_1 & & \\ & u_2 & \ddots & \\ & & \ddots & c_{n-1} \\ & & & u_n \end{bmatrix}$$

对角占优三对角矩阵

如果上述三对角阵 A 的元素还满足:

1. $|a_1| > |c_1| > 0$,
2. $|a_i| \geqslant |b_i| + |c_i|, \quad b_i, c_i \neq 0, \quad i = 2, 3, \ldots, n-1$,
3. $|a_n| > |b_n| > 0$.

则称它是对角占优的 (diagonally dominant). 对角占优阵的 LU 分解不用选主元.

n 阶对角占优三对角阵的 LU 分解计算量很小, 只有 $\mathcal{O}(n)$, 不用选主元就是数值稳定的, 其计算公式 (Llewellyn Thomas) 为

$$u_1 = a_1, \ l_i = b_i / u_{i-1}, \ u_i = a_i - l_i c_{i-1}, \ i = 2, \ldots, n.$$

稀疏矩阵

稀疏矩阵的定义:

$A \in \mathbb{R}^{n,n}$ 的 n^2 个元素中，如果只有 $\mathcal{O}(n)$ 个是非零的，就称它是一个稀疏矩阵 (sparse matrix).

1. 带宽比较小的带状矩阵是稀疏阵，其非零元分布有明显规律. 一般的稀疏矩阵，其非零元素分布没有规律.
2. 只存储和运算非零元素，可大大降低存储和计算复杂度，但必须同时存储它在矩阵中的位置，即如果 $a_{ij} \neq 0$，则要存储三元组 (a_{ij}, i, j)，导致复杂数据结构.
3. 设计只对非零元素进行计算的算法，稀疏性和数值稳定性往往是一对矛盾.

matlab 稀疏矩阵操作

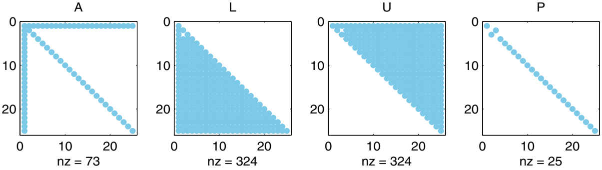
```
>> A = [0 0 11; 22 0 0; 0 33 0]
        0       0      11
       22       0       0
        0      33       0
>> S = sparse(A)
     (2,1)          22
     (3,2)          33
     (1,3)          11
>> B = full(S)

>> nnz(A)
ans = 3
>> nzmax(A)
ans = 9
>> nzmax(S)
ans = 3
```

```
>> load west0479
>> A = west0479;
>> size(A)
ans = 479     479
>> nnz(A)
ans = 1887
>> density = nnz(A)/prod(size(A))
density = 0.0082
>> spy(A)
>> spy(A,10)
>> [L U P] = lu(A);
>> spy(L,10)
>> spy(U,10)
```

稀疏矩阵 LU 分解的填充 (fill-in) 现象

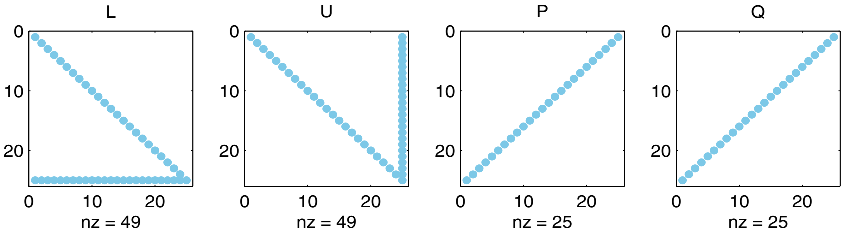
```
>> n=25; e = ones(n,1); A = spdiags(e,0,n,n);
>> A(1,:) = e'; A(:,1) = e;
>> figure
>> spy(A)
>> [L,U,P] = lu(A);
>> figure
>> spy(L)
>> figure
>> spy(U)
```



填充现象的克服

矩阵元素位置重排 (reordering) 可以克服填充现象.
位置重排 = 行交换 + 列交换. 需要找到合适的置换矩阵 P 和 Q,

$$PAQ = LU$$



$$A = \begin{bmatrix} 1 & & & & .1 \\ & 1 & & & .1 \\ & & 1 & & .1 \\ & & & 1 & .1 \\ .1 & .1 & .1 & .1 & 1 \end{bmatrix} = LU$$

$$= \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ .1 & .1 & .1 & .1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & & & & .1 \\ & 1 & & & .1 \\ & & 1 & & .1 \\ & & & 1 & .1 \\ & & & & .96 \end{bmatrix}.$$

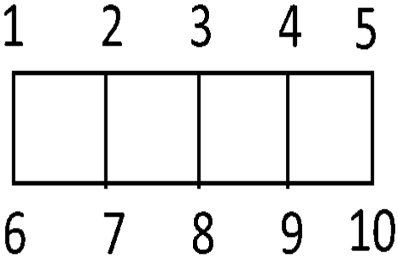
$$A' = \begin{bmatrix} 1 & .1 & .1 & .1 & .1 \\ .1 & 1 & & & \\ .1 & & 1 & & \\ .1 & & & 1 & \\ .1 & & & & 1 \end{bmatrix} = L'U'$$

$$= \begin{bmatrix} 1 & & & & \\ .1 & 1 & & & \\ .1 & -.01 & 1 & & \\ .1 & -.01 & -.01 & 1 & \\ .1 & -.01 & -.01 & -.01 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & .1 & .1 & .1 & .1 \\ & .99 & -.01 & -.01 & -.01 \\ & & .99 & -.01 & -.01 \\ & & & .99 & -.01 \\ & & & & .99 \end{bmatrix}.$$

以稀疏对称正定矩阵 A 的 Cholesky 分解为例，最需要解决的问题是：找到置换矩阵 P 使得 $PAP^T = LL^T$ 产生最少的填充.

利用图论方法可以证明：寻找产生最少填充的置换矩阵 P 是一个 NP-hard 问题!

	1	2	3	4	5	6	7	8	9	10
1	x	x				x				
2	x	x	x				x			
3		x	x	x				x		
4			x	x	x				x	
5				x	x					x
6	x					x	x			
7		x				x	x	x		
8			x				x	x	x	
9				x				x	x	x
10					x				x	x




```
>> doc Sparse Matrix Reordering
```

计算解的误差分析

误差与残差

设 $A \in \mathbb{R}^{n,n}$，当 $|A| \neq 0$ 时，将线性方程组 $Ax = b$ 的准确解记为 $x = A^{-1}b$，计算解记为 $\tilde{x}$，定义：

$$\text{误差 error：}\quad e(\tilde{x}) = x - \tilde{x}$$

$$\text{残差 residual：}\quad r(\tilde{x}) = b - A\tilde{x} = Ae(\tilde{x})$$

- 在实际问题中，准确解 x 和误差 $e(\tilde{x})$ 都是未知的量；而残差 $r(\tilde{x})$ 则是容易计算的量. 所以经常用残差来度量计算解的好坏.
- 误差和残差都是向量，需要用它们的范数 (norm) 衡量它们的大小.
- 残差很小的计算解，其误差可能很大！

```
>> A = [1 1.0001; 1.0001 1];
>> b = [1; 1];
>> x = A\b
x =
4.999750012497856e-01
4.999750012500894e-01

>> x1 = [-4.499775; 5.5002249]   %computing solution
x1 =
-4.499775000000000
5.500224900000000

>> A*x1 - b
9.999224900001380e-04
-7.749999930695140e-08
```

残差很小，误差可能很大！

原因：比较“病态”的系数矩阵，残差小不能保证误差也小！

```
>> det(A)
 ans =   -2.0001e-04

>> cond(A)
 ans = 2.0001e+04
```

定理

列主元 LU 分解方法可得到残量小的计算解，但不一定得到误差小的计算解.

向量的范数

定义 (向量的范数)

$\|x\|$ 称为 $x \in \mathbb{K}^n$ 的范数，如果 $\|x\| \in \mathbb{R}$ 满足：

1. 正定性：$\|x\| \geq 0, \forall\, x \in \mathbb{K}^n$，$\|x\| = 0$ 当且仅当 $x = 0$.
2. 齐次性：$\forall\, x \in \mathbb{K}^n, \forall\, \alpha \in \mathbb{K}$，都有 $\|\alpha x\| = |\alpha|\|x\|$.
3. 三角不等式：$\forall\, x \in \mathbb{K}^n,\ y \in \mathbb{K}^n$，都有 $\|x + y\| \leq \|x\| + \|y\|$.

对于 $x = (x_1, x_2, \cdots, x_n)^T \in \mathbb{K}^n$，常用的范数有：

$$\|x\|_2 = \left(\sum_{i=1}^{n} |x_i|^2 \right)^{1/2}, \quad \|x\|_1 = \sum_{i=1}^{n} |x_i|, \quad \|x\|_\infty = \max_i |x_i|$$

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}, \quad 0 < p < \infty$$

向量的范数 (norm) 就是其“长度”.

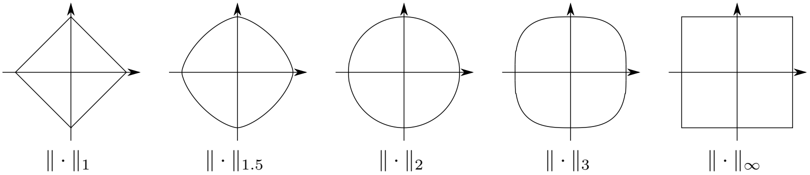
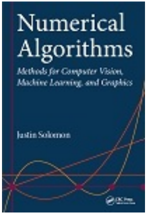


Figure 4.7 The set $\{\vec{x} \in \mathbb{R}^2 : \|\vec{x}\| = 1\}$ for different vector norms $\|\cdot\|$.

page 82 of



矩阵的范数

矩阵的范数可以跟向量范数一样定义，但常用向量范数来导出.
设 $A \in \mathbb{K}^{n,n}, x \in \mathbb{K}^n$，已有向量范数 $\|x\|$，矩阵 A 的范数定义为:

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

容易验证 $\|A\|$ 满足正定性，齐次性和三角不等式，并且还满足:

$$\|I\| = 1, \quad \|Ax\| \leq \|A\|\|x\|, \quad \|AB\| \leq \|A\|\|B\| (B \in \mathbb{K}^{n,n})$$

不同的向量范数会导出不同的矩阵范数，如果 $A = (a_{ij}) \in \mathbb{R}^{n,n}$，则有:

$$\|A\|_1 = \max_j \sum_{i=1}^n |a_{ij}|, \quad \|A\|_\infty = \max_i \sum_{j=1}^n |a_{ij}|, \quad \|A\|_2 = \sqrt{\lambda_{max}(A^TA)}$$

其中 $\lambda_{max}(A^TA)$ 为实对称矩阵 A^TA 的最大特征值.

线性方程组的稳定性与条件数

一个线性方程组 $Ax = b$ 由其系数矩阵和右端向量确定. 当 A 是可逆矩阵时，其准确解为 $x = A^{-1}b$.
当给定的数据 A 和 b 带有误差时，要解的方程组可以写成

$$(A + \Delta A)\tilde{x} = b + \Delta b$$

即使计算的过程不产生误差，由这个方程组得到的解 $\tilde{x}$ 也只是原方程组准确解 $x = A^{-1}b$ 的一个近似. 记近似解为 $\tilde{x} = x + \Delta x$，则有

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

问题：当数据误差 $\Delta A, \Delta b$ 很小时，解的误差 Δx 是不是也很小？

先假设系数矩阵 A 没有误差，右端向量 b 有误差 Δb，

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

在 A^{-1} 存在且 b 不是零向量的条件下，

$$\begin{aligned} \Delta x &= A^{-1}\Delta b, \\ \Rightarrow \frac{\|\Delta x\|}{\|x\|} &\le \frac{\|A^{-1}\|\|Ax\|\|\Delta b\|}{\|x\|\|b\|}, \\ \Rightarrow \frac{\|\Delta x\|}{\|x\|} &\le \|A^{-1}\|\|A\|\frac{\|\Delta b\|}{\|b\|} \end{aligned}$$

定义 (条件数，A.M. Turing, 1948 年)

$\kappa(A) = \|A^{-1}\|\|A\|$ 称为矩阵 A 的**条件数** (condition number).

- 显然条件数与矩阵的范数有关.

如果 A 可逆，b 不是零向量，A 和 b 都有误差，现在要解的方程组为

$$\tilde{A}\tilde{x} = \tilde{b}, \quad \text{或写成 } (A + \Delta A)(x + \Delta x) = b + \Delta b.$$

由于

$$\tilde{b} - b = \tilde{A}\tilde{x} - Ax = (\tilde{A} - A)\tilde{x} + A(\tilde{x} - x)$$

所以

$$\|\tilde{x} - x\| \le \|A^{-1}\|(\|\Delta A\|\|\tilde{x}\| + \|\Delta b\|)$$

由于 $\|\tilde{x}\| \le \|\tilde{x} - x\| + \|x\|$ 和 $\|b\| \le \|A\|\|x\|$，所以

$$\|\Delta x\| \le \|A^{-1}\|\|A\| \left(\frac{\|\Delta A\|}{\|A\|}(\|x\| + \|\Delta x\|) + \frac{\|\Delta b\|}{\|b\|}\|x\| \right)$$

当 $\kappa(A)\|\Delta A\|/\|A\| < 1$ 时，就有：

$$\frac{\|\Delta x\|}{\|x\|} \le \frac{\kappa(A)}{1 - \kappa(A)\frac{\|\Delta A\|}{\|A\|}} \left(\frac{\|\Delta b\|}{\|b\|} + \frac{\|\Delta A\|}{\|A\|} \right)$$

如果

$$\kappa(A)\frac{\|\Delta A\|}{\|A\|} \ll 1$$

(这只要 $\|\Delta A\|$ 相对 $\|A\|$ 充分小)，就有

$$\frac{\kappa(A)}{1-\kappa(A)\frac{\|\Delta A\|}{\|A\|}} \approx \kappa(A)$$

则有

$$\frac{\|\Delta x\|}{\|x\|} \lessapprox \kappa(A)\left(\frac{\|\Delta b\|}{\|b\|}+\frac{\|\Delta A\|}{\|A\|}\right)$$

再一次看到条件数 $\kappa(A)$ 就是数据中带有的误差传播到线性方程组解里的放大倍数.

再看看残差与误差的关系.

设用某种计算方法解线性方程组 $Ax=b$(b 不是零向量) 的计算解为 $\tilde{x}$, 准确解为 $x=A^{-1}b$, 计算解的残差为:

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

则有

$$r=Ax-A\tilde{x}=A(x-\tilde{x})$$

于是

$$\|x-\tilde{x}\|=\|A^{-1}r\|\leq\|A^{-1}\|\|r\|$$

再注意

$$\|b\|=\|Ax\|\leq\|A\|\|x\|\Rightarrow\frac{1}{\|x\|}\leq\frac{\|A\|}{\|b\|}$$

可得

$$\frac{\|x-\tilde{x}\|}{\|x\|}\leq\|A^{-1}\|\|A\|\frac{\|r\|}{\|b\|}=\kappa(A)\frac{\|r\|}{\|b\|}$$

- 条件数 $\kappa(A)$ 的计算一般比解线性方程组计算量大.

Hilbert 矩阵： $H = (h_{ij})$, $h_{ij} = 1/(i+j-1), i, j = 1, 2, \cdots, n$

Vandermonde 矩阵： $A = (a_{ij})$, $a_{ij} = v_i^{n-j}, i, j = 1, 2, \cdots, n$

```
>> H3 = hilb(3); cond(H3,2)
>> H4 = hilb(4); cond(H4,2)
ans = 1.5514e+04

>> H8 = hilb(8); cond(H8,2)
ans = 1.5258e+10

>> v = [1:8]
v =    1    2    3    4    5    6    7    8
>> A = vander(v); cond(A)
ans = 9.5211e+08
>> B = inv(A); det(B)
```

经验法则 (Rule of Thumb)

如果 $\kappa(A) = 10^k$，则数值求解线性方程组 $Ax = b$ 就会丢失 k 位有效数字.

LU 分解的数值稳定性

前面讨论的是线性方程组的稳定性，现在讨论算法的稳定性．前面已经见到了这样的例子：一个很简单的方阵，在浮点数系统中做 LU 分解的误差非常大！其原因是分解后的三角矩阵 L 或者 U 中含有绝对值非常大的元素 (跟 A 的元素相比)．记有舍入误差的 LU 分解为 $\tilde{L}\tilde{U}$，则可以证明存在 ΔA 使得

$$\tilde{L}\tilde{U} = A + \Delta A, \quad \text{且} \quad \frac{\|\Delta A\|}{\|L\|\|U\|} = \mathcal{O}(\epsilon_{mach}).$$

显然，如果 $\|L\|\|U\| = \mathcal{O}(\|A\|)$，则 LU 分解就是数值稳定的．

如果不选主元，很容易找到矩阵 A 使得 L 和 U 都含有非常大的元素．

$$A = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 1 \end{bmatrix}, \quad L = \begin{bmatrix} 1 & 0 \\ 10^{20} & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 10^{-20} & 1 \\ 0 & 1 - 10^{20} \end{bmatrix}$$

如果采用列选主元，L 的元素绝对值都不超过 1，所以有 $\|L\| = \mathcal{O}(1)$. 于是

$$\frac{\|\Delta A\|}{\|U\|} = \mathcal{O}(\epsilon_{mach})$$

设 $A \in \mathbb{R}^{n,n}$ 且 LU 分解为 $A = LU$，定义放大因子 ρ_n 为

$$\rho_n(A) = \frac{\max_{i,j} |u_{ij}|}{\max_{i,j} |a_{ij}|}, \qquad U = (u_{ij})$$

有下列结论：

- 一般可逆矩阵，列选主元 $\rho_n(A) \leq 2^{n-1}$；
- 上 Hessenberg 矩阵，列选主元，$\rho_n(A) \leq n$；
- 三对角矩阵，列选主元 $\rho_n(A) \leq 2$；
- 对角占优阵，不用选主元，$\rho_n(A) \leq 2$；
- 实对称正定阵，不用选主元，$\rho_n(A) \leq 1$；
- 随机生成的矩阵，列选主元 $\rho_n(A) \leq n^{2/3}$(平均).

一个 5 阶方阵的 LU 分解:

$$\begin{bmatrix} 1 & & & & 1 \\ -1 & 1 & & & 1 \\ -1 & -1 & 1 & & 1 \\ -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & & & & \\ -1 & 1 & & & \\ -1 & -1 & 1 & & \\ -1 & -1 & -1 & 1 & \\ -1 & -1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & & & & 1 \\ & 1 & & & 2 \\ & & 1 & & 4 \\ & & & 1 & 8 \\ & & & & 16 \end{bmatrix}$$

$$\rho_5(A) = 16$$

$$\rho_n(A) = 2^{n-1}$$

参考书

1. Gene H. Golub, Charles F. Van Loan, Matrix Computations (4th Edition), Johns Hopkins University Press, 2012.
2. James W. Demmel, Applied Numerical Linear Algebra, SIAM, 1997.
3. Alfio Quarteroni, Riccardo Sacco, Fausto Saleri, Numerical Mathematics (2nd Edition), Springer, 2007.
4. Walter Gander, Martin J. Gander, Felix Kwok, Scientific Computing: an introduction using Maple and MATLAB, Springer, 2014.
5. Justin Solomon, Numerical Algorithms, Textbook published by AK Peters/CRC Press, 2015.