线性方程组的直接法

殷东生
dyin@math.tsinghua.edu.cn

清华大学数学科学系

2018年秋季学期

I HEAR AND I FORGET,
I SEE AND I REMEMBER,
I DO AND I UNDERSTAND.

“不闻不若闻之,
闻之不若见之,
见之不若知之,
知之不若行之"

荀子《儒孝篇》

ONE MUST LEARN BY DOING THE THING;
FOR THOUGH YOU THINK YOU KNOW IT;
YOU HAVE NO CERTAINTY UNTIL YOU TRY.

$$\frac{1}{n}\sin x = ?$$

$$\frac{1}{\cancel{n}}\mathrm{si}\cancel{n}\, x =$$

$$six = 6$$

After explaining to a student through various lessons and examples that:

$$\lim_{x \to 8} \frac{1}{x-8} = \infty$$

I tried to check if she really understood that, so I gave her a different example. This was the result:

$$\lim_{x \to 5} \frac{1}{x-5} = 5$$

投入产出模型

在经济活动中分析投入多少财力、物力、人力，产出多少社会财富是衡量经济效益高低的主要标志。
投入产出技术正是研究一个经济系统各部门间的“投入”与“产出”关系的数学模型。
该方法最早由美国著名的经济学家瓦.列昂捷夫（W.Leontief）提出，是目前比较成熟的经济分析方法。

- 投入：从事一项经济活动的消耗；
- 产出：从事经济活动的结果；
- 投入产出数学模型：通过编制投入产出表，运用线性代数工具建立数学模型，从而揭示国民经济各部门、再生产各环节之 间的内在联系，并据此进行经济分析、预测和安排预算计划。

投入产出表

流量 产出 投入		消耗部门				最终需求				总产出
		1	2	$\cdots$	n	消费	累计	出口	合计	
生产部门	1	x_{11}	x_{12}	$\cdots$	x_{1n}				y_1	x_1
	2	x_{21}	x_{22}	$\cdots$	x_{2n}				y_2	x_2
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$				$\vdots$	$\vdots$
	n	x_{n1}	x_{n2}		x_{nn}				y_n	x_n
新创价值	工　资	v_1	v_2	$\cdots$	v_n					
	纯收入	m_1	m_2	$\cdots$	m_n					
	合　计	z_1	z_2	$\cdots$	z_n					
总投入		x_1	x_2	$\cdots$	x_n					

投入产出的基本平衡关系

从左到右：　中间需求＋最终需求＝总产出　(1)

从上到下：　中间消耗＋净产值＝总投入　(2)

得：产出平衡方程组(也称分配平衡方程组)：

$$\begin{cases} x_{11}+x_{12}+\cdots+x_{1n}+y_1=x_1 \\ x_{21}+x_{22}+\cdots+x_{2n}+y_2=x_2 \\ \cdots\cdots\cdots\cdots \\ x_{n1}+x_{n2}+\cdots+x_{nn}+y_n=x_n \end{cases} \tag{3}$$

$$\sum_{j=1}^{n} x_{ij}+y_i=x_i(i=1,2,\cdots,n) \tag{4}$$

定义1 第j部门生产单位价值所消耗第i部门的价值称为第j部门对第i部门的直接消耗系数。记作：

$$a_{ij}\left(i,j=1,2,\cdots,n\right)$$

由定义得

$$a_{ij}=\frac{x_{ij}}{x_j}\left(i,j=1,2,\cdots,n\right) \qquad (9)$$

把投入产出表中的各个中间需求x_{ij}换成相应的a_{ij}后得到的数表称为直接消耗系数表，并称n阶矩阵$A=\left(a_{ij}\right)$为直接消耗系数矩阵。

由直接消耗系数的定义得 $x_{ij}=a_{ij}x_j$ ，代入(3)，得

$$\begin{cases} a_{11}x_1+a_{12}x_2+\cdots+a_{1n}x_n+y_1=x_1 \\ a_{21}x_1+a_{22}x_2+\cdots+a_{2n}x_n+y_2=x_2 \\ \cdots\cdots\cdots\cdots \\ a_{n1}x_1+a_{n2}x_2+\cdots+a_{nn}x_n+y_n=x_n \end{cases} \tag{5}$$

令 $X=(x_1 \quad x_2 \quad \cdots \quad x_n)', Y=(y_1 \quad y_2 \quad \cdots \quad y_n)'$，

(5)式可表示为 $AX+Y=X$ ，或

$$\boxed{(E-A)X=Y} \tag{6}$$

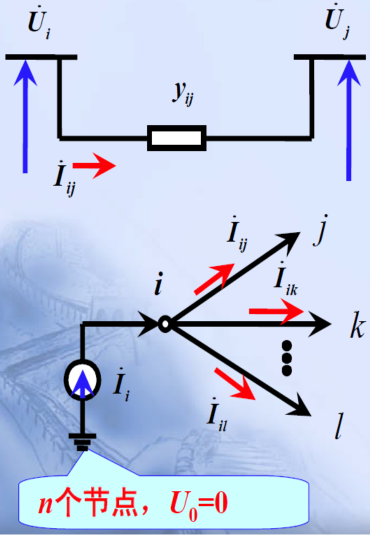
称矩阵E-A为列昂捷夫矩阵。

复杂电力系统潮流计算

- 电力系统潮流计算：是对复杂电力系统正常和故障条件下稳态运行状态的计算。
- 其目的是求取电力系统在给定运行方式下的节点电压和功率分布，用以检查系统各元件是否过负荷、各点电压是否满足要求、功率分布和分配是否合理以及功率损耗等。
- 潮流计算是电力系统计算分析中的一种最基本的计算。
- 潮流计算的计算机算法是以电网络理论为基础的，应用数值计算方法求解一组描述电力系统稳态特性的方程。

原则上，如果已知各节点的电压或各支路的电流，电力网络的运行状态便可唯一确定。节点方程以各节点电压为待求量。

节点导纳矩阵



$$\dot{I}_{ij} = y_{ij}(\dot{U}_i - \dot{U}_j)$$

$$\dot{I}_i = \sum_{j=0,j\neq i}^{n} \dot{I}_{ij} = \sum_{j=0,j\neq i}^{n} y_{ij}(\dot{U}_i - \dot{U}_j)$$

$$= \sum_{j=0,j\neq i}^{n} y_{ij}\dot{U}_i - \sum_{j=0,j\neq i}^{n} y_{ij}\dot{U}_j$$

$$\dot{I}_i = \sum_{j=0,j\neq i}^{n} y_{ij}\dot{U}_i - \sum_{j=1,j\neq i}^{n} y_{ij}\dot{U}_j$$

$$\dot{I}_i = \dot{U}_i \sum_{j=0,j\neq i}^{n} y_{ij} \underbrace{-y_{i1}\dot{U}_1 - y_{i2}\dot{U}_2 \cdots - y_{in}\dot{U}_n}$$

节点 i 自导纳

$$Y_{ii} \triangleq \sum_{j=0,j\neq i}^{n} y_{ij}, \quad Y_{ij} \triangleq -y_{ij}$$

i - j 之间互导纳

$$\dot{I}_i = \sum_{j=1}^{n} Y_{ij}\dot{U}_j$$

$$= Y_{i1}\dot{U}_1 + Y_{i2}\dot{U}_2 + \cdots + Y_{ii}\dot{U}_i + \cdots + Y_{in}\dot{U}_n$$

$$\dot{I}_i = \sum_{j=1}^{n} Y_{ij}\dot{U}_j$$

$$\begin{cases} \dot{I}_1 = Y_{11}\dot{U}_1 + Y_{12}\dot{U}_2 + \cdots + Y_{1i}\dot{U}_i + \cdots + Y_{1n}\dot{U}_n \\ \dot{I}_2 = Y_{21}\dot{U}_1 + Y_{22}\dot{U}_2 + \cdots + Y_{2i}\dot{U}_i + \cdots + Y_{2n}\dot{U}_n \\ \quad\vdots \\ \dot{I}_n = Y_{n1}\dot{U}_1 + Y_{n2}\dot{U}_2 + \cdots + Y_{ni}\dot{U}_i + \cdots + Y_{nn}\dot{U}_n \end{cases}$$

节点导纳矩阵

节点电流列向量

$$\begin{bmatrix} \dot{I}_1 \\ \dot{I}_2 \\ \vdots \\ \dot{I}_n \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} & \cdots & Y_{1i} & \cdots & Y_{1n} \\ Y_{21} & Y_{22} & \cdots & Y_{2i} & \cdots & Y_{2n} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ Y_{n1} & Y_{n2} & \cdots & Y_{ni} & \cdots & Y_{nn} \end{bmatrix} \begin{bmatrix} \dot{U}_1 \\ \dot{U}_2 \\ \vdots \\ \dot{U}_i \\ \vdots \\ \dot{U}_n \end{bmatrix}$$

节点电压列向量

$$\mathbf{I} = \mathbf{Y} \quad \mathbf{U}$$

考虑下三角方程组

$$Ly = b,\ \ b = [b_1, b_2, \cdots, b_n]^T \in \mathbb{R}^n,\ \ y = [y_1, y_2, \cdots, y_n]^T.$$

其中 $\boldsymbol{b}$ 已知，求 $\boldsymbol{y}$，而$L = [l_{ij}] \in \mathbb{R}^{n\times n}$为

$$L = \begin{bmatrix} l_{11} & 0 & \cdots & \cdots & 0 \\ l_{21} & l_{22} & 0 & \cdots & \vdots \\ l_{31} & l_{32} & l_{33} & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ l_{n1} & l_{n2} & l_{n3} & \cdots & l_{nn} \end{bmatrix}, \quad l_{ii} \neq 0$$

第一个方程为

$$l_{11}y_1 = b_1 \Rightarrow y_1 = \frac{b_1}{l_{11}}.$$

第二个方程：$l_{21}y_1 + l_{22}y_2 = b_2 \Rightarrow y_2 = \frac{1}{l_{22}}(b_2 - l_{21}y_1).$

更一般的如果$y_1, y_2, \cdots, y_{i-1}$ 已知，则利用第 i 个方程有

$$y_i = \frac{1}{l_{ii}}\Big(b_i - \sum_{j=1}^{i-1} l_{ij}y_j\Big).$$

这个算法称为向前替换方法。写成算法有

```
function FORSOLVE(L,b)
  % Find the solution to the system Lx = b,
  x_1 = b_1/l_11
  for i = 2:n do
    sum = 0.0
    for j = 1:i-1 do
      sum = sum + l_ij x_j
    end for
    x(i) = (b(i) - sum)/l_ii
  end for
  return x
end function
```

总的计算量

$$1 + \sum_{i=1}^{n-1}(2i+2) = n^2 + n - 1$$

其计算量为 $\mathcal{O}(n^2)$.

求解上三角方程组

$$U\boldsymbol{x} = \boldsymbol{y}, \quad \boldsymbol{x} = [x_1, x_2, \cdots, x_n] \text{未知}$$

而 $U \in \mathbb{R}^{n \times n}$ 如下给出

$$U = \begin{bmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ 0 & u_{22} & u_{23} & \cdots & u_{2n} \\ 0 & 0 & u_{33} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & u_{nn} \end{bmatrix}$$

若 $u_{ii} \neq 0$，则回带(backward substitution)有

$$x_n = \frac{y_n}{u_{nn}}, \quad x_i = \frac{1}{u_{ii}}\Big(y_i - \sum_{j=i+1}^{n} u_{ij}x_j\Big), \quad i = n-1, \cdots, 1.$$

用直接法求解

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

本质上是找到一个矩阵 $M, \det M\neq 0$ 使得 MA 是一个上三角矩阵，这个过程称为消元(elimination)，消元后在求解方程组

$$MA\boldsymbol{x}=M\boldsymbol{b}\quad \text{回带}$$

其中最简单的Gauss 消去法是用一些列的单位下三角阵 $L_i(\boldsymbol{l}_i)$来把A 化成上三角矩阵。

$$L_{n-1}(\boldsymbol{l}_{n-1})\cdots L_i(\boldsymbol{l}_i)\cdots L_1(\boldsymbol{l}_1)A=A^{(n)}=U$$

若令 $L=L_1^{-1}\cdots L_{n-1}^{-1}$，则有

$$A=LU$$

假设方程为

$$
\begin{aligned}
a_{11}^{(1)}x_1 + a_{12}^{(1)}x_2 + a_{13}^{(1)}x_3 + \cdots + a_{1n}^{(1)}x_n &= b_1^{(1)} \\
a_{21}^{(1)}x_1 + a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + \cdots + a_{2n}^{(1)}x_n &= b_2^{(1)} \\
a_{31}^{(1)}x_1 + a_{32}^{(1)}x_2 + a_{33}^{(1)}x_3 + \cdots + a_{3n}^{(1)}x_n &= b_3^{(1)} \\
&\vdots \\
a_{n1}^{(1)}x_1 + a_{n2}^{(1)}x_2 + a_{n3}^{(1)}x_3 + \cdots + a_{nn}^{(1)}x_n &= b_n^{(1)}.
\end{aligned}
$$

若$a_{11}^{(1)} \neq 0$，则可将第1行的$-a_{21}^{(1)}/a_{11}^{(1)}$倍加到第2行上去

$$a_{22}^{(2)}x_2 + a_{23}^{(2)}x_3 + \cdots + a_{2n}^{(2)}x_n = b_2^{(2)}$$

同样可将第1行的$-a_{31}^{(1)}/a_{11}^{(1)}$倍加到第3行上去

$$a_{32}^{(2)}x_2 + a_{33}^{(2)}x_3 + \cdots + a_{3n}^{(2)}x_n = b_3^{(2)}$$

如此一直到第n行

$$a_{n2}^{(2)}x_2+a_{n3}^{(2)}x_3+\cdots+a_{nn}^{(2)}x_n=b_n^{(2)}$$

得到与原方程组等价的一个新的方程组

$$\begin{array}{rcl}
a_{11}^{(1)}x_1+a_{12}^{(1)}x_2+a_{13}^{(1)}x_3+\cdots+a_{1n}^{(1)}x_n & = & b_1^{(1)} \\
a_{22}^{(2)}x_2+a_{23}^{(2)}x_3+\cdots+a_{2n}^{(2)}x_n & = & b_2^{(2)} \\
& \vdots & \\
a_{n2}^{(2)}x_2+a_{n3}^{(2)}x_3+\cdots+a_{nn}^{(2)}x_n & = & b_n^{(2)}.
\end{array}$$

如果$a_{22}^{(2)} \neq 0$，重复消去过程有

$$
\begin{aligned}
a_{11}^{(1)}x_1 + a_{12}^{(1)}x_2 + a_{13}^{(1)}x_3 + \cdots + a_{1n}^{(1)}x_n &= b_1^{(1)} \\
a_{22}^{(2)}x_2 + a_{23}^{(2)}x_2 + \cdots + a_{2n}^{(2)}x_n &= b_2^{(2)} \\
a_{33}^{(3)}x_2 + \cdots + a_{3n}^{(3)}x_n &= b_3^{(3)} \\
&\vdots \\
a_{n3}^{(3)}x_2 + \cdots + a_{nn}^{(3)}x_n &= b_n^{(3)}.
\end{aligned}
$$

同样，如果$a_{33}^{(3)} \neq 0$，上述处理过程可以继续，直到最后一个方程只含有变量 x_n，

$$\begin{aligned}
a_{11}^{(1)}x_1 + a_{12}^{(1)}x_2 + a_{13}^{(1)}x_3 + \cdots + a_{1n}^{(1)}x_n &= b_1^{(1)} \\
a_{22}^{(2)}x_2 + a_{23}^{(2)}x_2 + \cdots + a_{2n}^{(2)}x_n &= b_2^{(2)} \\
a_{33}^{(3)}x_2 + \cdots + a_{3n}^{(3)}x_n &= b_3^{(3)} \\
&\vdots \\
a_{nn}^{(n)}x_n &= b_n^{(n)}.
\end{aligned}$$

这样便将原问题转化为一个系数矩阵为上三角阵的线性代数方程组的求解问题。

算法 (Gauss 消去)

1. For $k = 1 : n-1$ Do:
2. For $i = k+1 : n$ Do:
3. $piv := a_{ik}/a_{kk}$
4. For $j := k+1 : n+1$ Do :
5. $a_{ij} := a_{ij} - piv * a_{kj}$
6. End
7. End
8. End

算法总的计算量为

$$T = \sum_{k=1}^{n-1} \sum_{i=k+1}^{n} [1 + \sum_{j=k+1}^{n+1} 2] = \sum_{k=1}^{n-1} \sum_{i=k+1}^{n} [2(n-k)+3] =?$$

数学家简介 (Johann Carl Friedrich Gauss)

高斯（*1777*年*4*月*30*日—*1855*年*2*月*23*日）德国著名数学家、物理学家、天文学家、大地测量学家。是近代数学奠基者之一，高斯被认为是历史上最重要的数学家之一，并享有“数学王子”之称。高斯和阿基米德、牛顿并列为世界三大数学家。一生成就极为丰硕，以他名字“高斯”命名的成果达*110*个，属数学家中之最。他对数论、代数、统计、分析、微分几何、大地测量学、地球物理学、力学、静电学、天文学、矩阵理论和光学皆有贡献。

现考虑矩阵形式.记

$$
\begin{aligned}
\boldsymbol{A}^{(k)} &= \begin{bmatrix}
a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1k-1}^{(1)} & a_{1k}^{(1)} & a_{1k+1}^{(1)} & \cdots & a_{1n}^{(1)} \\
0 & a_{22}^{(2)} & \cdots & a_{2k-1}^{(2)} & a_{2k}^{(2)} & a_{2k+1}^{(2)} & \cdots & a_{2n}^{(2)} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & a_{k-1k-1}^{(k-1)} & a_{k-1k}^{(k-1)} & a_{k-1k+1}^{(k-1)} & \cdots & a_{k-1n}^{(k-1)} \\
0 & 0 & \cdots & 0 & a_{kk}^{(k)} & a_{kk+1}^{(k)} & \cdots & a_{kn}^{(k)} \\
0 & 0 & \cdots & 0 & a_{k+1k}^{(k)} & a_{k+1k+1}^{(k)} & \cdots & a_{k+1n}^{(k)} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 0 & a_{nk}^{(k)} & a_{nk+1}^{(k)} & \cdots & a_{nn}^{(k)}
\end{bmatrix} \\
\boldsymbol{b}^{(k)} &= [b_1^{(1)}, b_2^{(2)}, \cdots, b_{k-1}^{(k-1)}, b_k^{(k)}, b_{k+1}^{(k)}, \cdots, b_n^{(k)}]^T, \\
\boldsymbol{l}_k &= [0, 0, \cdots, 0, 0, a_{k+1k}^{(k)}/a_{kk}^{(k)}, \cdots, a_{nk}^{(k)}/a_{kk}^{(k)}]^T.
\end{aligned}
$$

则由如上的过程可知

$$A^{(k+1)} = L_k(-\boldsymbol{l}_k)A^{(k)}, \quad \boldsymbol{b}^{(k+1)} = L_k(-\boldsymbol{l}_k)\boldsymbol{b}^{(k)}.$$

其中

$$L_k(-\boldsymbol{l}_k) = \begin{bmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & -a_{k+1k}^{(k)}/a_{kk}^{(k)} & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & -a_{nk}^{(k)}/a_{kk}^{(k)} & 0 & \cdots & 1 \end{bmatrix}$$

则最后有

$$\begin{aligned} \boldsymbol{A}^{(n)} &= L_{n-1}(-\boldsymbol{l}_{n-1})L_{n-2}(-\boldsymbol{l}_{n-2})\cdots L_1(-\boldsymbol{l}_1)\boldsymbol{A}^{(1)}, \\ \boldsymbol{b}^{(n)} &= L_{n-1}(-\boldsymbol{l}_{n-1})L_{n-2}(-\boldsymbol{l}_{n-2})\cdots L_1(-\boldsymbol{l}_1)\boldsymbol{b}^{(1)}. \end{aligned}$$

由$L_j^{-1}(\boldsymbol{l}_j) = L_j(-\boldsymbol{l}_j)$知

$$\boldsymbol{A}^{(1)} = L_1(\boldsymbol{l}_1)L_2(\boldsymbol{l}_2)\cdots L_{n-1}(\boldsymbol{l}_{n-1})\boldsymbol{A}^{(n)}.$$

令$\boldsymbol{L} = L_1(\boldsymbol{l}_1)L_2(\boldsymbol{l}_2)\cdots L_{n-1}(\boldsymbol{l}_{n-1}), \boldsymbol{U} = \boldsymbol{A}^{(n)}$， 则$\boldsymbol{L}$为单位下三角阵(即对角线元素为单位1的下三角阵)，$\boldsymbol{U}$为上三角阵。 有

$$A = LU$$

定理 ($\boldsymbol{LU}$分解)

非奇异矩阵$\boldsymbol{A} \in \mathbb{R}^{n\times n}$，若其顺序主子式

$$\Delta_i \overset{def}{=} \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1i} \\ a_{21} & a_{22} & \cdots & a_{2i} \\ \vdots & \vdots & \ddots & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ii} \end{vmatrix}, \ i = 1, 2, \cdots, n$$

均不为零, 则存在唯一的单位下三角阵$\boldsymbol{L}$和上三角阵$\boldsymbol{U}$，使

$$\boldsymbol{A} = \boldsymbol{LU}.$$

上述分解称为矩阵的三角分解，或$\boldsymbol{LU}$分解，或Doolittle分解。

$$
\begin{aligned}
\boldsymbol{U} &= \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{pmatrix}, \det U \neq 0 \\
\boldsymbol{D} &= \mathbf{diag}(u_{11}, u_{22}, \cdots, u_{nn}) \\
\boldsymbol{U}^* &= \begin{pmatrix} 1 & \frac{u_{12}}{u_{11}} & \cdots & \frac{u_{1n}}{u_{11}} \\ 0 & 1 & \cdots & \frac{u_{2n}}{u_{22}} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}
\end{aligned}
$$

则$\boldsymbol{U} = \boldsymbol{D}\boldsymbol{U}^*$。于是有

$$
\boldsymbol{A} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{U}^*.
$$

其中$\boldsymbol{D}$为对角阵，$\boldsymbol{L}, \boldsymbol{U}^*$分别为单位下三角阵和单位上三角阵。

定理

非奇异矩阵$\boldsymbol{A} \in \mathbb{R}^{n\times n}$，若其顺序主子式 Δ_i, $i=1,2,\cdots,n$ 均不为零，则存在唯一的单位下三角阵$\boldsymbol{L}$，单位上三角阵$\boldsymbol{U}^*$和对角阵$\boldsymbol{D}$，使$\boldsymbol{A}=\boldsymbol{LDU}^*$。

由上可知，若矩阵$\boldsymbol{A}$的所有顺序主子式都非零，则有唯一的三角分解$\boldsymbol{A}=\boldsymbol{LU}$，其中$\boldsymbol{L}$为单位下三角阵，$\boldsymbol{U}$为上三角阵。

$$\underbrace{\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}}_{\boldsymbol{A}} = \underbrace{\begin{pmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & 1 \end{pmatrix}}_{\boldsymbol{L}} \underbrace{\begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{pmatrix}}_{\boldsymbol{U}}$$

两矩阵中的未知变量个数为n^2与矩阵$\boldsymbol{A}$的元素个数一致，因而可以直接由矩阵$\boldsymbol{A}$确定$\boldsymbol{L}$和$\boldsymbol{U}$。

Doolittle分解法

$$
\begin{aligned}
a_{kj} &= \sum_{r=1}^{\min(k,j)} l_{kr}u_{rj}, \\
a_{kj} &= \sum_{r=1}^{k} l_{kr}u_{rj} = \sum_{r=1}^{k-1} l_{kr}u_{rj} + u_{kj},\ j \geq k \\
u_{kj} &= a_{kj} - \sum_{r=1}^{k-1} l_{kr}u_{rj}. \\
a_{jk} &= \sum_{r=1}^{k} l_{jr}u_{rk} = \sum_{r=1}^{k-1} l_{jr}u_{rk} + l_{jk}u_{kk},\ j > k \\
l_{jk} &= \frac{1}{u_{kk}}\left(a_{jk} - \sum_{r=1}^{k-1} l_{jr}u_{rk}\right).
\end{aligned}
$$

Doolite 分解Matlab 函数

```
function [A] = lu_ijk (A)
[n,n]=size(A);
for i=1:n
for j=2:i
A(i,j-1)=A(i,j-1)/A(j-1,j-1);
for k=1:j-1,
    A(i,j)=A(i,j)-A(i,k)*A(k,j);
end
end
for j=i+1:n
for k=1:i-1,
    A(i,j)=A(i,j)-A(i,k)*A(k,j);
end
end
end
```

例

用*Doolittle*分解方法求解如下方程组

$$\begin{aligned} x_1 - x_2 &= 0, \\ -2x_1 + 4x_2 - 2x_3 &= -1, \\ -x_2 + 2x_3 &= 1.5. \end{aligned}$$

解：由于

$$\boldsymbol{A} = \begin{pmatrix} 1 & -1 & 0 \\ -2 & 4 & -2 \\ 0 & -1 & 2 \end{pmatrix}, \boldsymbol{b} = \begin{pmatrix} 0 \\ -1 \\ 1.5 \end{pmatrix}$$

首先对系数矩阵做Doolittle分解

$$\begin{pmatrix} 1 & -1 & 0 \\ -2 & 4 & -2 \\ 0 & -1 & 2 \end{pmatrix} \to \begin{pmatrix} \boxed{1} & \boxed{-1} & \boxed{0} \\ -2 & 4 & -2 \\ 0 & -1 & 2 \end{pmatrix} \to \begin{pmatrix} \boxed{1} & \boxed{-1} & \boxed{0} \\ \boxed{-2} & 4 & -2 \\ \boxed{0} & -1 & 2 \end{pmatrix}$$

$$\to \begin{pmatrix} \boxed{1} & \boxed{-1} & \boxed{0} \\ \boxed{-2} & \boxed{4} & \boxed{-2} \\ \boxed{0} & -1 & 2 \end{pmatrix} \to \begin{pmatrix} \boxed{1} & \boxed{-1} & \boxed{0} \\ \boxed{-2} & \boxed{4} & \boxed{-2} \\ \boxed{0} & \boxed{-1} & 2 \end{pmatrix}$$

$$\to \begin{pmatrix} \boxed{1} & \boxed{-1} & \boxed{0} \\ \boxed{-2} & \boxed{4} & \boxed{-2} \\ \boxed{0} & \boxed{-1} & \boxed{2} \end{pmatrix}$$

其次求解$\boldsymbol{Ly} = \boldsymbol{b}$，得到$\boldsymbol{y} = (0, -1, 1)^T$；最后求解$\boldsymbol{Ux} = \boldsymbol{y}$，得到$\boldsymbol{x} = (1/2, 1/2, 1)^T$。

Matlab 求矩阵 A 的LU 分解的命令为

```
>> A = [ 1    2    3
         4    5    6
         7    8    0 ];
>> [L,U]=lu(A)

L =

    0.1429    1.0000         0
    0.5714    0.5000    1.0000
    1.0000         0         0


U =

    7.0000    8.0000         0
         0    0.8571    3.0000
         0         0    4.5000
```

定理

非奇异矩阵$\boldsymbol{A} \in \mathbb{R}^{n\times n}$，若其顺序主子式都不为零，则$LU$分解计算得到的 $\hat{L}$ 和 $\hat{U}$ 满足

$$\hat{L}\hat{U} = A + H, \qquad 且|A| = \max\left\{|a_{ij}|\right\}_{i,j=1,\cdots,n},$$

其中ϵ是机器精度，而

$$|H| \le 3(n-1)\epsilon(|A| + |\hat{L}||\hat{U}|) + \mathcal{O}(\epsilon^2).$$

通过求解 $\hat{L}\boldsymbol{y} = \boldsymbol{b}$ 和 $\hat{U}\boldsymbol{x} = \boldsymbol{y}$ 得到的解 $\hat{\boldsymbol{x}}$ 满足

$$(A+E)\hat{\boldsymbol{x}} = \boldsymbol{b}$$

此处

$$|E| \le n\epsilon(3|A| + 5|\hat{L}||\hat{U}|) + \mathcal{O}(\epsilon^2).$$

方程组$\boldsymbol{Ax}=\boldsymbol{d}$，其中$\boldsymbol{d}=(d_1,\cdots,d_n)^T$，而$A$为三对角矩阵(tridiagonal matrix)

$$A=\begin{bmatrix} b_1 & c_1 & & & & \\ a_2 & b_2 & c_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & & & c_{n-1} \\ & & & & a_n & b_n \end{bmatrix}$$

若A的顺序主子式非零，则可以用Doolittle分解方法求解。三对角矩阵是稀疏矩阵 (sparse matrix)，因为共有

$$n^2-(n+2(n-1))=n^2-3n+2$$

个零元素。

一维Poisson方程

$$-\frac{d^2u(x)}{dx^2}=f(x),\ \ 0<x<1,$$

其中f是给定的函数。$u(x)$还必须满足边界条件

$$u(0)=u(1)=0.$$

首先把区间$[0,1]$等分$N+1$,有$N+2$个离散的点

$$x_i=ih,\ h=\frac{1}{N+1},\ 0\le i\le N+1.$$

记 $u_i=u(x_i)$, $f_i=f(x_i)$，用中心差分近似

$$-\frac{d^2u(x)}{d^2x}\Big|_{x=x_i}=\frac{2u_i-u_{i-1}-u_{i+1}}{h^2}-\tau_i$$

其中τ_i称为截断误差，

$$\tau_i \approx O(h^2 \cdot \|\frac{d^4u}{dx^4}\|_\infty).$$

则Poisson方程在$x = x_i$处可写为

$$-u_{i-1} + 2u_i - u_{i+1} = h^2 f_i + h^2 \tau_i,\ 1 \le i \le N.$$

由边界条件知$u_0 = u_{N+1} = 0$，因而有下面的方程组

$$T_N \cdot \begin{bmatrix} u_1 \\ \vdots \\ \vdots \\ u_N \end{bmatrix} \equiv \begin{bmatrix} 2 & -1 & & 0 \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 2 \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ \vdots \\ u_N \end{bmatrix} = h^2 \begin{bmatrix} f_1 \\ \vdots \\ \vdots \\ f_N \end{bmatrix} + h^2 \begin{bmatrix} \tau_1 \\ \vdots \\ \vdots \\ \tau_N \end{bmatrix}$$

三对角矩阵可以稀疏存储:

$$\begin{aligned}
\boldsymbol{a} &= [a_1, a_2, \cdots, a_n]^T, \\
\boldsymbol{b} &= [b_1, b_2, \cdots, b_{n-1}, b_n]^T, \\
\boldsymbol{c} &= [c_1, c_2, \cdots, c_{n-1}]^T
\end{aligned}$$

对于三对角阵 L和U有下面的特殊形式:

$$L = \begin{bmatrix} 1 & & & & \\ l_2 & 1 & & & \\ & l_3 & 1 & & \\ & & \ddots & \ddots & \\ & & & l_n & 1 \end{bmatrix}, U = \begin{bmatrix} u_1 & c_1 & & & \\ & u_2 & c_2 & & \\ & & \ddots & \ddots & \\ & & & \ddots & c_{n-1} \\ & & & & u_n \end{bmatrix}$$

$$\Longrightarrow \begin{cases} u_1 = b_1, \\ l_i = \frac{a_i}{u_{i-1}}, i = 2, 3, \cdots, n \\ u_i = b_i - l_i c_{i-1}, i = 2, 3, \cdots, n \end{cases}$$

可求得L和U，求解$A\boldsymbol{x} = \boldsymbol{d}$转化为$L\boldsymbol{y} = \boldsymbol{d}$和 $U\boldsymbol{x} = \boldsymbol{y}$，

$$\begin{cases} y_1 = d_1, \\ y_i = d_i - l_i y_{i-1}, i = 2, 3, \cdots, n \end{cases}$$

$$\begin{cases} x_n = \frac{y_n}{u_n}, \\ x_i = \frac{y_i - c_i x_{i+1}}{u_i}, i = n-1, n-2, \cdots, 1. \end{cases}$$

上面的计算过程称为求解三对角方程组的追赶法。

用追赶法求解三对角方程组的计算量为$\mathcal{O}(n)$。

```
function THOMAS(a,b,c,rhs)
  % The function solves a tridiagonal system of linear equations Ax = rhs
  % using the linear Thomas algorithm. a is the lower diagonal, b the
  % diagonal, and c the upper diagonal.

  % Begin elimination steps, resulting in a bidiagonal matrix
  % with 1s on its diagonal.
   c_1 = c_1/b_1
   rhs_1 = rhs_1/b_1
   for i = 2:n-1 do
      c_i = c_i/(b_i - a_{i-1} c_{i-1})
      rhs_i = (rhs_i - a_{i-1} rhs_{i-1})/(b_i - a_{i-1} c_{i-1})
   end for
   rhs_n = (rhs_n - a_{n-1} rhs_{n-1})/(b_n - a_{n-1} c_{n-1})
   % Now perform back substitution
   x_n = rhs_n
   for i = n-1:-1:1 do
      x_i = rhs_i - c_i x_{i+1}
   end for
   return x
 end function
```

```
>> a = randn(4999,1);
>> b =randn(5000,1);
>> c = randn(4999,1);
>> rhs = randn(5000,1);
>> tic;x1 = thomas(a,b,c,rhs);toc;
Elapsed time is 0.032754 seconds.
>> T = trid(a,b,c);
>> tic;x2 = T\rhs;toc;
Elapsed time is 0.386797 seconds.
```

周期样条插值等问题遇到的循环三对角方程$A\boldsymbol{x}=\boldsymbol{d}$,

$$A=\begin{bmatrix} b_1 & c_1 & & & & a_n \\ a_2 & b_2 & c_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & & & c_{n-1} \\ c_n & & & & a_n & b_n \end{bmatrix}$$

若A的顺序主子式非零，则可以用Doolittle分解方法求解。

此循环三对角阵 L和U有下面的特殊形式：

$$L = \begin{bmatrix} 1 & & & & \\ l_2 & 1 & & & \\ & l_3 & 1 & & \\ & & \ddots & \ddots & \\ \sigma_1 & \sigma_2 & \cdots & \sigma_{n-1}+l_n & 1 \end{bmatrix}, \quad \begin{bmatrix} u_1 & c_1 & & & \rho_1 \\ & u_2 & c_2 & & \rho_2 \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & c_{n-1}+\rho_{n-1} \\ & & & & u_n \end{bmatrix}$$

数学家简介 (L. H. Thomas)

L. H. Thomas(1903-1992) 著名的物理学家，上世纪*50*年代在*IBM*的*Watson*实验室工作，其最著名的工作是 *Thomas-Fermi electron gas model*， 是密度泛函理论的基础。

由矩阵乘法有

$$
\begin{cases}
u_1 = b_1,\ \rho_1 = a_n,\ \sigma_1 = \dfrac{c_n}{u_1}, \\
l_i = \dfrac{a_i}{u_{i-1}},\ u_i = b_i - l_i c_{i-1},\ \rho_i = -l_i \rho_{i-1}, \\
\sigma_i = -\frac{\sigma_{i-1} c_{i-1}}{u_{i-1}},\quad i = 1, s, \cdots, n-1, \\
l_n = \dfrac{a_n}{u_{n-1}}, \\
u_n = b_1 - (l_n + \sigma_{n-1})(c_{n-1} + \rho_{n-1}) - \displaystyle\sum_{i=1}^{n-2} \sigma_i \rho_i
\end{cases}
$$

可求得L和U的所有元素，解原方程组$A\boldsymbol{x}=\boldsymbol{d}$分为两步求解$L\boldsymbol{y}=\boldsymbol{d}$和$U\boldsymbol{x}=\boldsymbol{y}$，

$$\begin{cases} y_1 = d_1, \\ y_i = d_i - l_i y_{i-1}, i = 2, 3, \cdots, n \\ y_n = d_n - \sum\limits_{i=1}^{n-2} \sigma_i y_i - (l_n + \sigma_{n-1}) y_{n-1}, \end{cases}$$

$$\begin{cases} x_n = \dfrac{y_n}{u_n}, \\ x_i = \dfrac{y_i - c_i x_{i+1} - \rho_i x_n}{u_i}, i = n-1, n-2, \cdots, 1. \end{cases}$$

Gauss消去法如果$a_{kk}^{(k)}=0$，则消去法不能继续下去。即便 系数矩阵$\boldsymbol{A}$非奇异，理论上方程组存在唯一解亦是如此。此外，即便在消去过程中，$a_{kk}^{(k)}\neq 0$，但是如果$|a_{kk}^{(k)}|\ll 1$，则会产生大数除小数的情况，导致数值解的精度下降，有时更会完全失真。

$$
\begin{aligned}
x_2+x_3 &= 2\\
x_1-x_2+x_3 &= 1\\
x_1+x_2-x_3 &= 1
\end{aligned}
$$

其准确解为$\boldsymbol{x}=(1,1,1)^T$。但系数矩阵

$$
\begin{pmatrix}
0 & 1 & 1\\
1 & -1 & 1\\
1 & 1 & -1
\end{pmatrix}
$$

Doolittle分解第一步即不能进行。

例

用3位十进制浮点运算求解

$$\begin{aligned} 1.00 \times 10^{-5} x_1 + 1.00 x_2 &= 1.00, \\ 1.00 x_1 + 1.00 x_2 &= 2.00. \end{aligned}$$

解： 这个方程组的准确解用3位十进制浮点数近似为$(1.00, 1.00)^T$；

$$\begin{pmatrix} 1.00 \times 10^{-5} & 1.00 \\ 1.00 & 1.00 \end{pmatrix} \to \begin{pmatrix} \boxed{1.00 \times 10^{-5}} & \boxed{1.00} \\ \boxed{1.00 \times 10^{5}} & 1.00 \end{pmatrix}$$

$$\to \begin{pmatrix} \boxed{1.00 \times 10^{-5}} & \boxed{1.00} \\ \boxed{1.00 \times 10^{5}} & \boxed{-1.00 \times 10^{5}} \end{pmatrix}$$

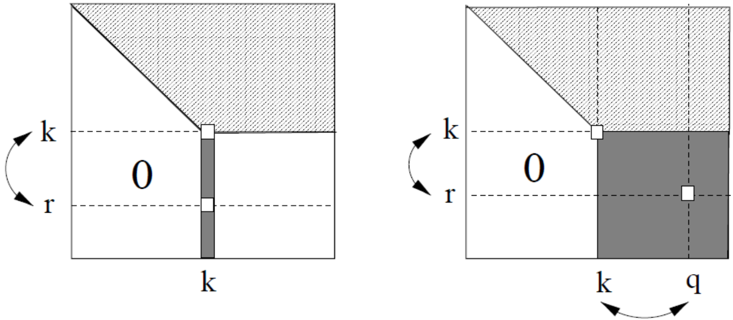
第一步求解$\boldsymbol{Ly} = \boldsymbol{b}$，得到$\boldsymbol{y} = (1.00, -1.00 \times 10^5)^T$。第二步求解$\boldsymbol{Ux} = \boldsymbol{y}$，得到$\boldsymbol{x} = (0.00, 1.00)^T$。

选主元(pivoting)

如上所述困难有一自然的解决方案：即在第k步消去之前，选择

$$a_{kk}^{(k)}, a_{k+1k}^{(k)}, \cdots, a_{nk}^{(k)}$$

模值最大者，将其所在 的行与第k行交换，然后进行消去过程。



设其所在的行数为$i_k \geq k$，则 加入如上处理过程后有

$$\boldsymbol{A}^{(k+1)} = L_k(-\boldsymbol{l}_k)\boldsymbol{I}_{i_k k}\boldsymbol{A}^{(k)}.$$

记$\boldsymbol{U} = \boldsymbol{A}^{(n)}$, 则

$$\boldsymbol{U} = L_{n-1}(-\boldsymbol{l}_{n-1})\boldsymbol{I}_{i_{n-1}n-1}\cdots L_2(-\boldsymbol{l}_2)\boldsymbol{I}_{i_2 2}L_1(-\boldsymbol{l}_1)\boldsymbol{I}_{i_1 1}\boldsymbol{A}$$

定理

设$\boldsymbol{A}$非奇异，则存在排列阵$\boldsymbol{P}$，单位下三角阵$\boldsymbol{L}$和上三角阵$\boldsymbol{U}$，使得

$$\boldsymbol{PA} = \boldsymbol{LU}.$$

```
function [L,U,P,Q] = LUpivtot(A,n)
P=eye(n); Q=P; Minv=P;
for k=1:n-1
  [Pk,Qk]=pivot(A,k,n);    A=Pk*A*Qk;
  [Mk,Mkinv]=MGauss(A,k,n);
  A=Mk*A;    P=Pk*P;       Q=Q*Qk;
  Minv=Minv*Pk*Mkinv;
end
U=triu(A);   L=P*Minv;

function [Mk,Mkinv]=MGauss(A,k,n)
Mk=eye(n);
for i=k+1:n,   Mk(i,k)=-A(i,k)/A(k,k);   end
Mkinv=2*eye(n)-Mk;

function [Pk,Qk]=pivot(A,k,n)
[y,i]=max(abs(A(k:n,k:n)));  [piv,jpiv]=max(y);
ipiv=i(jpiv);   jpiv=jpiv+k-1;   ipiv=ipiv+k-1;
Pk=eye(n); Pk(ipiv,ipiv)=0; Pk(k,k)=0; Pk(k,ipiv)=1; Pk(ipiv,k)=1;
Qk=eye(n); Qk(jpiv,jpiv)=0; Qk(k,k)=0; Qk(k,jpiv)=1; Qk(jpiv,k)=1;
```

Matlab 求LU分解的命令

```
A = [ 1    2    3
      4    5    6
      7    8    0 ];
[L,U,P]=lu(A)
L =

    1.0000         0         0
    0.1429    1.0000         0
    0.5714    0.5000    1.0000

U =

    7.0000    8.0000         0
         0    0.8571    3.0000
         0         0    4.5000

P =

     0     0     1
     1     0     0
     0     1     0
```

定理

非奇异矩阵$\boldsymbol{A} \in \mathbb{R}^{n\times n}$， 若其顺序主子式 Δ_i, $i = 1, 2, \cdots, n$ 均不为零，则存在唯一的单位下三角阵$\boldsymbol{L}$，单位上三角阵$\boldsymbol{U}^*$和对角阵$\boldsymbol{D}$，使得

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{U}^*.$$

推论

设$\boldsymbol{A} \in \mathbb{R}^{n\times n}$，$\boldsymbol{A}^T = \boldsymbol{A}$，$\boldsymbol{A}$的顺序主子式$\Delta_i, i = 1, 2, \cdots, n$都非零， 则存在唯一的单位下三角阵$\boldsymbol{L}$和对角阵$\boldsymbol{D}$，使得

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^T.$$

定理 (Cholesky分解)

设$A \in \mathbb{R}^{n\times n}$为对称正定阵，则存在唯一的对角元素为正的下三角阵$L$，使得

$$A = LL^T.$$

上述分解称为对称正定阵的Cholesky分解。

利用A的Cholesky分解式来求$Ax = b$的方法称为Cholesky方法或平方根法。

数学家简介 (Cholesky)

Cholesky(1875-1918) 法国军官和数学家，在北非和克里特岛进行大地测量。首先用 *Cholesky* 方法求解地图测量中的最小二乘问题 的法方程。

Cholesky 方法

$$
\begin{aligned}
a_{ij} &= \sum_{k=1}^{n} l_{ik}l_{jk} = \sum_{k=1}^{j} l_{ik}l_{jk}, i \geq j, \\
a_{jj} &= \sum_{k=1}^{j} l_{jk}^2 = \sum_{k=1}^{j-1} l_{jk}^2 + l_{jj}^2, \\
l_{jj} &= \left(a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2\right)^{1/2}, \\
a_{ij} &= \sum_{k=1}^{j} l_{ik}l_{jk} = \sum_{k=1}^{j-1} l_{ik}l_{jk} + l_{ij}l_{jj},\ i > j \\
l_{ij} &= \frac{1}{l_{jj}}\left(a_{ij} - \sum_{k=1}^{j-1} l_{ik}l_{jk}\right).
\end{aligned}
$$

```
function [A] = chol2 (A)
[n,n]=size(A);
for k=1:n-1
A(k,k)=sqrt(A(k,k));
A(k+1:n,k)=A(k+1:n,k)/A(k,k);
for j=k+1:n,
    A(j:n,j)=A(j:n,j)-A(j:n,k)*A(j,k);
end
end
A(n,n)=sqrt(A(n,n));
```

$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^T$后，求解$\boldsymbol{L}\boldsymbol{L}^T\boldsymbol{x} = \boldsymbol{b}$，分别求解$\boldsymbol{L}\boldsymbol{y} = \boldsymbol{b}$和$\boldsymbol{L}^T\boldsymbol{x} = \boldsymbol{y}$。由于

$$a_{ii} = \sum_{k=1}^{i} l_{ik}^2,$$

可以推出$|l_{ik}| \leq \sqrt{a_{ii}}, k = 1, 2, \cdots, i$，所以Cholesky方法中，中间量得到控制，不会产生"大数吃掉小数"的情况。

Matlab求 A 的Cholesky分解的函数为chol(A)。

```
>> A = [2 2 18; 2 24 -18; 18 -18 234];
>> b = [65; -62; 819];
>> R = chol(A);
>> y = R' \ b;
>> x = R \ y
x =
   17.0000
   -2.5000
    2.0000
```

注记

Cholesky 分解的算法可以用来检验一个对称矩阵A是否正定。

注记

矩阵$A \in \mathbb{R}^{n\times n}$的行列式可以用$LU$分解$PA = LU$来计算

$$\det A = det(P) \cdot \det(LU) = \mathrm{sgn}(\pi_0) \cdots u_{11} \cdots u_{nn}$$

但是在计算行列式的时候要非常小心； 因为

$$\det(\alpha A) = \alpha^n \det A$$

而这个性质导致在实际就算的时候由于舍入误差的影响会把一个“小”的行列式变得 任意“大”。实际上在上面的这个变换下唯一不变的量是Boole 量

$$\det A = 0, \text{或者} \ \det A \neq 0.$$

而这个量是衡量一个线性方程组可解性的一个理论特征，但是可解性理论并 不是基于行列式的。

可利用LU分解计算矩阵的逆。设 X 是 $A \in \mathbb{R}^{n\times n}$ 的逆矩阵，则 X 的列向量 是下列线性方程组的解：

$$A\boldsymbol{x}_i = \boldsymbol{e}_i, \quad i = 1, \cdots, n.$$

若

$$PA = LU,$$

而P 是选主元的排列阵，则为了求逆需要求$2n$ 个三角形式的方程组

$$\begin{aligned} L\boldsymbol{y}_i &= P\boldsymbol{e}_i, \;\; i = 1, \cdots, n \\ U\boldsymbol{x}_i &= \boldsymbol{y}_i, \;\; i = 1, \cdots, n \end{aligned}$$

在用直接法求解$\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$的过程中，由于舍入误差的存在，必然会导致结果产生误差。因而有必要对 可能产生的误差作一估计。通常假设在数值处理的过程中计算都是精确的。

例

方程组

$$\begin{pmatrix} 3 & 1 \\ 3.0001 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}=\begin{pmatrix} 4 \\ 4.0001 \end{pmatrix}$$

的精确解是$(1,1)^T$。若A和$\boldsymbol{b}$作如下的微小变化，

$$\begin{pmatrix} 3 & 1 \\ 2.9999 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}=\begin{pmatrix} 4 \\ 4.0002 \end{pmatrix}.$$

其精确解是$(-2,10)^T$；数据的微小变化产生解的巨大变化。

方程组

$$\begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 32 \\ 23 \\ 33 \\ 31 \end{pmatrix}$$

其精确解为$(1,1,1,1)^T$。对右端向量做微小的修改

$$\begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix} \begin{pmatrix} x_1+\delta x_1 \\ x_2+\delta x_2 \\ x_3+\delta x_3 \\ x_4+\delta x_4 \end{pmatrix} = \begin{pmatrix} 32.1 \\ 22.9 \\ 33.1 \\ 30.9 \end{pmatrix}$$

其精确解为$(9.2,-12.6,4.5,-1.1)^T$。

若对系数矩阵做微小的修改

$$\begin{pmatrix} 10 & 7 & 8.1 & 7.2 \\ 7.08 & 5.04 & 6 & 5 \\ 8 & 5.98 & 9.89 & 9 \\ 6.99 & 4.99 & 9 & 9.98 \end{pmatrix} \begin{pmatrix} x_1 + \Delta x_1 \\ x_2 + \Delta x_2 \\ x_3 + \Delta x_3 \\ x_4 + \Delta x_4 \end{pmatrix} = \begin{pmatrix} 32 \\ 23 \\ 33 \\ 31 \end{pmatrix}$$

其精确解为$\boldsymbol{x} + \Delta\boldsymbol{x} = (-81, 137, -34, 22)^T$。

定义 (条件数)

设$A \in \mathbb{R}^{n\times n}$为可逆阵，$\|\cdot\|$为一种从属的矩阵范数，则

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

称为A的条件数。

矩阵的条件数的概念是由 Turing 引入的：

Quart. J. Mech. Appl. Math. 1, 287-308, 1948.

数学家简介 (Alan Mathison Turing)

图灵（*1912-1954*），英国数学家、逻辑学家，被称为计算机之父，人工智能之父。图灵在科学、特别在数理逻辑和计算机科学方面，取得了举世瞩目的成就，他的一些科学成果，构成了现代计算机技术的基础：如可计算理论，判定问题、人工智能、图灵实验等。世界级的长跑健将，著名的同性恋者，受到英国政府的迫害，职业生涯尽毁并自杀身亡。

记

$$cond_2(A) = ||A||_2||A^{-1}||_2.$$

类似定义$cond_\infty(A)$和$cond_1(A)$。

定理

设$A \in \mathbb{R}^{n\times n}$, A非奇异，$\boldsymbol{x}$及$\boldsymbol{x}+\delta\boldsymbol{x}$满足

$$A\boldsymbol{x} = \boldsymbol{b}, A(\boldsymbol{x}+\delta\boldsymbol{x}) = \boldsymbol{b}+\delta\boldsymbol{b},$$

且 $\boldsymbol{b} \neq 0$，则

$$\frac{||\delta\boldsymbol{x}||}{||\boldsymbol{x}||} \leq \mathrm{cond}(A)\frac{||\delta\boldsymbol{b}||}{||\boldsymbol{b}||}$$

定理

设$A \in \mathbb{R}^{n\times n}$，$det(A) \neq 0$，$\boldsymbol{x}$和$\boldsymbol{x}+\boldsymbol{\delta x}$分别满足方程组

$$\begin{aligned} A\boldsymbol{x} &= \boldsymbol{b} \\ (A+\delta A)(\boldsymbol{x}+\boldsymbol{\delta x}) &= \boldsymbol{b}+\boldsymbol{\delta b} \end{aligned}$$

其中$\boldsymbol{b} \neq \boldsymbol{0}$，而且$\delta A$充分小，使

$$\frac{\|\delta A\|}{\|A\|} < \frac{1}{cond(A)}$$

则有

$$\frac{\|\boldsymbol{\delta x}\|}{\|\boldsymbol{x}\|} \leq \frac{cond(A)}{1-\|A^{-1}\|\|\delta A\|}\left(\frac{\|\delta A\|}{\|A\|}+\frac{\|\boldsymbol{\delta b}\|}{\|\boldsymbol{b}\|}\right)$$

设$A \in \mathbb{R}^{n\times n}$，$A$非奇异，则有

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

$$cond(\alpha A) = cond(A), \forall \alpha \in \mathbb{R}, \alpha \neq 0$$

2. 若A为正交阵，则

$$cond_2(A) = 1$$

3. 若U为正交阵，则

$$cond_2(A) = cond_2(AU) = cond_2(UA)$$

4. 设λ_1, λ_n分别是按模最大和最小的特征值，则

$$cond(A) \geq \frac{|\lambda_1|}{|\lambda_n|}$$

若对称，则$cond_2(A) = \frac{|\lambda_1|}{|\lambda_n|}$。

矩阵的条件数是否依赖于范数的选取？

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

Matlab 求矩阵条件数的命令为 cond。

```
>>  A = diag(ones(10,1)) + diag(ones(9,1),1);
>> cond(A)

ans =
   13.2320

>> cond(A,'inf')

ans =
    20

>> cond(A,1)

ans =
    20

>> cond(A, 'fro')

ans =
   32.3265
```

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 1 & 0 & \dots & 0 \\ 0 & \vdots & 1 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & 1 & 1 \\ 0 & 0 & 0 & \dots & \dots & 1 \end{bmatrix}$$

定理

设$A \in \mathbb{R}^{n\times n}$，$\boldsymbol{b} \in \mathbb{R}^n$，且$A$非奇异，$\boldsymbol{b} \neq \boldsymbol{0}$，$\boldsymbol{x}$是方程组$A\boldsymbol{x} = \boldsymbol{b}$的精确解，$\tilde{\boldsymbol{x}}$是近似解，$\boldsymbol{r} = \boldsymbol{b} - A\tilde{\boldsymbol{x}}$($\tilde{\boldsymbol{x}}$的残量)，则有

$$\frac{1}{cond(A)}\frac{\|\boldsymbol{r}\|}{\|\boldsymbol{b}\|} \leq \frac{\|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|}{\|\boldsymbol{x}\|} \leq cond(A)\frac{\|\boldsymbol{r}\|}{\|\boldsymbol{b}\|}$$

例

考虑线性方程组

$$Ax = \begin{bmatrix} 0.913 & 0.659 \\ 0.457 & 0.330 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0.254 \\ 0.127 \end{bmatrix} = b$$

考虑两个近似解

$$\tilde{x}_1 = \begin{bmatrix} -0.0827 \\ 0.5 \end{bmatrix} \quad \text{和} \quad \tilde{x}_2 = \begin{bmatrix} 0.999 \\ -1.001 \end{bmatrix}$$

它们的残差范数分别为

$$\|r_1\|_1 = 2.1 \times 10^{-4} \quad \text{及} \quad \|r_2\|_1 = 2.4 \times 10^{-2}.$$

注记

对病态矩阵，由于条件数很大，小的残差并不能说明解的误差小。

例：系数矩阵的所有特征值为

$\lambda_1 \approx 30.2887$，$\lambda_2 \approx 3.858$，$\lambda_3 \approx 0.8431$，$\lambda_4 \approx 0.01015$， 故而

$$cond_2(A) = \frac{\lambda_1}{\lambda_4} \approx 2984$$

若$\boldsymbol{b} = (32, 23, 33, 31)^T$，$\boldsymbol{\delta b} = (0.1, -0.1, 0.1, -0.1)^T$，$\boldsymbol{\delta x} = (8.2, -13.6, 3.5, -2.1)^T$，所以 实际的相对误差为

$$\frac{\|\boldsymbol{\delta x}\|_2}{\|\boldsymbol{x}\|_2} \approx 8.198$$

而用上述定理估计的相对误差界为

$$\frac{\|\boldsymbol{\delta x}\|_2}{\|\boldsymbol{x}\|_2} \leq 2984 \frac{\|\boldsymbol{\delta b}\|_2}{\|\boldsymbol{b}\|_2} = 9.943.$$

相对误差放大了两千多倍。

例

*Hilbert*矩阵是一个著名的病态矩阵，对称正定，

$$H_n = \begin{pmatrix} 1 & \frac{1}{2} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \cdots & \frac{1}{2n-1} \end{pmatrix}$$

n	$\lambda_{\max}$	$\lambda_{\min}$	$\kappa_2(H_n)$
5	1.6	3.3×10^{-6}	4.8×10^{5}
10	1.8	1.1×10^{-13}	1.6×10^{13}
15	1.8	3.0×10^{-21}	6.1×10^{20}
20	1.9	7.8×10^{-29}	2.5×10^{28}
25	2.0	1.9×10^{-36}	1.0×10^{36}

Matlab 求矩阵条件数的命令为 cond。

```
>>  A = diag(ones(10,1)) + diag(ones(9,1),1);
>> cond(A)

ans =
   13.2320

>> cond(A,'inf')

ans =
    20

>> cond(A,1)

ans =
    20

>> cond(A, 'fro')

ans =
   32.3265
```

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 1 & 0 & \dots & 0 \\ 0 & \vdots & 1 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & 1 & 1 \\ 0 & 0 & 0 & \dots & \dots & 1 \end{bmatrix}$$

1. 矩阵的条件数与矩阵的行列式值没有必然的联系。如

$$D_n = diag(10^{-1}, 10^{-1}, \cdots, 10^{-1}) \in \mathbb{R}^{n\times n}$$

其条件数$cond_p(D_n) = 1, p = 1, 2, \infty$。所以$D_n$是一个非常好的矩阵，但是$\det D_n = 10^{-n}$。

2. 对于病态的线性方程组，求解时必须非常小心，否则得不到满意的数值解。

方程组

$$H_4\boldsymbol{x} = (\frac{25}{12}, \frac{77}{60}, \frac{57}{60}, \frac{319}{420})^T$$

的准确解是$\boldsymbol{x} = (1, 1, 1, 1)^T$。如果我们用3位和5位十进制舍入运算的消去法求解，得到的解分别是

$$(0.988, 1.42, -0.428, 2.10)^T$$

$$(1.0000, 0.99950, 1.0017, 0.99900)^T.$$

定理

$A \in \mathbb{R}^{n\times n}$ 非奇异，则

$$\min\left\{\frac{\|\delta A\|_2}{\|A\|_2} : A+\delta A\text{奇异}\right\} = \frac{1}{\mathrm{cond}_2(A)}$$

注记

粗略地说，如果用主元素法或平方根法(A对称正定阵)解方程组$A\boldsymbol{x}=\boldsymbol{b}$，设$A$和$\boldsymbol{b}$的元素准确到 s位数字，且$\mathrm{cond}(A)\approx 10^t$，其中$t<s$，则计算解向量的分量大约有$s-t$位数字的准确度。

- 对原方程作某些预处理，可望降低系数矩阵的条件数。 因为 $\mathrm{cond}(\alpha A) = \mathrm{cond}(A)$，所以不能通过将每一个方程乘上相同的常数$\alpha$来达到这个目标。
  可考虑将矩阵的每一行和每一列分别乘上不同的常数。亦即找可逆的对角阵D_1和D_2，将方程组$A\boldsymbol{x} = \boldsymbol{b}$化为

$$D_1 A D_2 \boldsymbol{y} = D_1 \boldsymbol{b}, \boldsymbol{x} = D_2 \boldsymbol{y}$$

  这称为矩阵的平衡问题。理论上最好选择对角阵$\bar{D}_1$和$\bar{D}_2$满足

$$\mathrm{cond}(\bar{D}_1 A \bar{D}_2) = \min \mathrm{cond}(D_1 A D_2)$$

  其中的min是对所有D_1和D_2属于可逆对角阵集合取的。

预处理矩阵的平衡问题

D_1是平衡每个方程，而D_2是平衡未知数的。令$D_2 = I$，称为行平衡。D_1的选择可使D_1A每一行 的∞-范数大体相等，这可避免消元过程中小数与大数的相加。

如果令 $D_1 = I$， 这时候称为列平衡，D_2的选择可使AD_2的每一列的∞-范数大体相等。

从行平衡和列平衡的想法出发，要求矩阵A的所有行和列的范数大致相等。 但是为了求 D_1和D_2的元素，需要求解未知量为$2n-2$的非线性方程组，所需计算量比原问题本身的计算量要多。

方程组

$$\begin{pmatrix} 10 & 100000 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 100000 \\ 2 \end{pmatrix}$$

的准确解为$x_1 = 1.0010001\cdots, x_2 = 0.99989998\cdots$。
引入$D_1 = \mathrm{diag}(10^{-5}, 1)$，平衡后方程为

$$\begin{pmatrix} 0.0001 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

如果都用3位十进制的列主元消去法求解，原方程组解得$\boldsymbol{x} = (0.00, 1.00)^T$，而后一方程组解得$\boldsymbol{x} = (1.00, 1.00)^T$。前一方程系数矩阵的2-条件数为$1 \times 10^5$，而后一方程系数矩阵的2-条件数为2.6184。

如果Gauss 消去法是向后稳定的，则必须有

$$3n\epsilon\|L\|\cdot\|U\| \approx O(\epsilon)\|A\| \Longrightarrow \frac{\|\delta A\|}{\|A\|} \approx O(\epsilon)$$

从实际的计算结果来看，列主元的Guass 消去几乎总是有

$$\|L\|\cdot\|U\| \approx \|A\|.$$

因为L的每一列的元素的绝对值总是小于1，所以只需要考虑$\|U\|$。定义列主元的Gauss消去的增长因子为

$$\rho_{pp} = \frac{\|U\|_{\max}}{\|A\|_{\max}}, \quad \text{其中} \quad \|A\|_{\max} = \max_{ij}|a_{ij}|.$$

所以LU分解的稳定性等价于ρ_{pp}必须很小或者随着n缓慢增长。
实际计算中，绝大多数情况下$\rho_{pp} \approx n$或者更小。平均来说

$$\rho_{pp} \approx n^{1/2} \sim n^{2/3}.$$

引理

列主元的Gauss 消去法保证$\rho_{pp} \leq 2^{n-1}$。而且这个上界是可以达到的。

下面的矩阵可达上界

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

因为$\|L\|_\infty \leq n$，而$\|U\|_\infty \leq n\rho_{pp}\|A\|_\infty$，结合上面的分析就有

$$\|\delta A\|_\infty \leq 3\rho_{pp}n^3\epsilon\|A\|_\infty.$$

注记

因子$3\rho_{pp}n^3$即使在$\rho_{pp}=1$时也是相当悲观的估计。若采用单精度$\epsilon=10^{-7}$，$n=150$，则$3n^3\epsilon>1$。算法不会有任何精度！

注记 (全主元Gauss消去)

完全主元的*Gauss*消去比列主元的*Gauss*消去稳定，其增长因子ρ_{cp}的上界有

$$\rho_{cp}=\frac{\max_{ij}|u_{ij}|}{\max_{ij}|a_{ij}|}\le\sqrt{n\cdot 1\cdot 3^{1/2}\cdot 4^{1/3}\cdots n^{1/(n-1)}}\approx n^{1/2+\log_e n/4}.$$

计算发现ρ_{cp}平均为$n^{1/2}$。

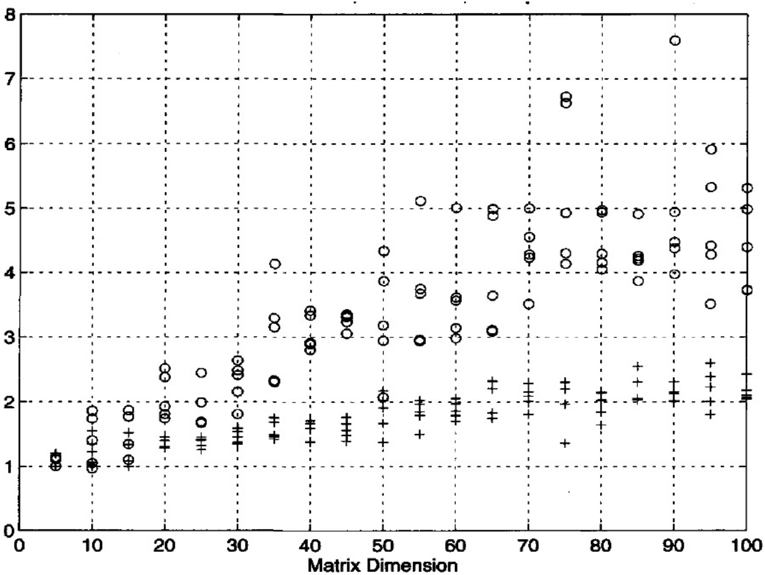


图: 完全主元和部分主元Gauss消去的增长因子 $\circ = \rho_{pp}, + = \rho_{cp}$

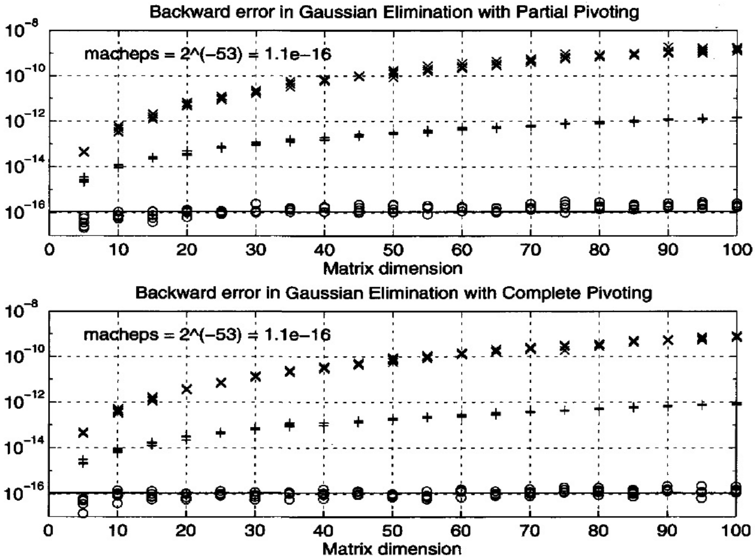


图: 随机矩阵的Gauss消去的向后误差, $\times = 3n^3\epsilon\rho$, $+ = 3n\frac{\||L|\cdot|U|\|_\infty}{\|A\|_\infty}$, $\circ = \frac{\|Ax-b\|_\infty}{\|A\|_\infty\|x\|_\infty}$