第四章 解线性方程组的直接法

中国科学技术大学 数学学院

chenxjin@ustc.edu.cn

问题：解线性方程组 $\begin{cases} a_{11}x_1 + \cdots + a_{1n}x_n = b_1 \\ \vdots \\ a_{n1}x_1 + \cdots + a_{nn}x_n = b_n \end{cases}$

$\Longleftrightarrow Ax = b$，其中$b = (b_1, b_2, \ldots, b_n)^T$,$A$ 为系数矩阵。

问题：解线性方程组 $\begin{cases} a_{11}x_1 + \cdots + a_{1n}x_n = b_1 \\ \vdots \\ a_{n1}x_1 + \cdots + a_{nn}x_n = b_n \end{cases}$

$\Longleftrightarrow Ax = b$，其中$b = (b_1, b_2, \ldots, b_n)^T$,$A$ 为系数矩阵。

实际中，存在大量的解线性方程组的问题。很多数值方法本身到最后也会归结为解线性方程组，如：

- 样条插值的M和m关系式
- 曲线拟合的法方程
- 非线性方程组的Newton迭代法
- ……

理论上我们有Cramer法则：当且仅当$\det(A) \neq 0$时，方程组有唯一的解；而且解为$x_i = \frac{D_i}{D}$. 其中

$$D = \det(A), D_i = \det \begin{pmatrix} a_{11} & \cdots & a_{1,i-1} & b_1 & a_{1,i+1} & \cdots & a_{1n} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ a_{n1} & \cdots & a_{n,i-1} & b_n & a_{n,i+1} & \cdots & a_{nn} \end{pmatrix}$$

理论上我们有Cramer法则：当且仅当$\det(A) \neq 0$时，方程组有唯一的解；而且解为$x_i = \frac{D_i}{D}$. 其中

$$D = \det(A), D_i = \det \begin{pmatrix} a_{11} & \cdots & a_{1,i-1} & b_1 & a_{1,i+1} & \cdots & a_{1n} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ a_{n1} & \cdots & a_{n,i-1} & b_n & a_{n,i+1} & \cdots & a_{nn} \end{pmatrix}$$

但在实际计算中，Cramer法则往往无法执行（使用）！例如$n = 100$，需做约$101 \cdot 100! \cdot 99$次乘除法!

理论上我们有Cramer法则：当且仅当$\det(A) \neq 0$时，方程组有唯一的解；而且解为$x_i = \frac{D_i}{D}$. 其中

$$D = \det(A), D_i = \det \begin{pmatrix} a_{11} & \cdots & a_{1,i-1} & b_1 & a_{1,i+1} & \cdots & a_{1n} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ a_{n1} & \cdots & a_{n,i-1} & b_n & a_{n,i+1} & \cdots & a_{nn} \end{pmatrix}$$

但在实际计算中，Cramer法则往往无法执行（使用）！例如$n = 100$， 需做约$101 \cdot 100! \cdot 99$次乘除法! 国产的超级计算机的“神威-太湖之光”, 其浮点运算速度为每秒9.3亿亿次， 其运算一秒约相当于全中国人用计算器算两千年。够厉害!!!

理论上我们有Cramer法则：当且仅当$\det(A) \neq 0$时，方程组有唯一的解；而且解为$x_i = \frac{D_i}{D}$. 其中

$$D = \det(A), D_i = \det \begin{pmatrix} a_{11} & \cdots & a_{1,i-1} & b_1 & a_{1,i+1} & \cdots & a_{1n} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ a_{n1} & \cdots & a_{n,i-1} & b_n & a_{n,i+1} & \cdots & a_{nn} \end{pmatrix}$$

但在实际计算中，Cramer法则往往无法执行（使用）！例如$n = 100$， 需做约$101 \cdot 100! \cdot 99$次乘除法! 国产的超级计算机的“神威-太湖之光”, 其浮点运算速度为每秒9.3亿亿次， 其运算一秒约相当于全中国人用计算器算两千年。够厉害!!! 但是，即使用速度为10^{33}次/秒的超级计算机来计算它，需要算

$$\frac{101 \cdot 100! \cdot 99}{365 \cdot 24 \cdot 3600 \cdot 10^{33}}$$

理论上我们有Cramer法则：当且仅当$\det(A) \neq 0$时，方程组有唯一的解；而且解为$x_i = \frac{D_i}{D}$. 其中

$$D = \det(A), D_i = \det \begin{pmatrix} a_{11} & \cdots & a_{1,i-1} & b_1 & a_{1,i+1} & \cdots & a_{1n} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ a_{n1} & \cdots & a_{n,i-1} & b_n & a_{n,i+1} & \cdots & a_{nn} \end{pmatrix}$$

但在实际计算中，Cramer法则往往无法执行（使用）！例如$n = 100$， 需做约$101 \cdot 100! \cdot 99$次乘除法! 国产的超级计算机的“神威-太湖之光”, 其浮点运算速度为每秒9.3亿亿次， 其运算一秒约相当于全中国人用计算器算两千年。够厉害!!! 但是，即使用速度为10^{33}次/秒的超级计算机来计算它，需要算

$$\frac{101 \cdot 100! \cdot 99}{365 \cdot 24 \cdot 3600 \cdot 10^{33}} \approx 10^{120}\text{年}.$$

解线性方程组的方法可以分为2类：

解线性方程组的方法可以分为2类：

1. 直接法：准确，可靠，理论上得到的解是精确的;
2. 迭代法：速度快，但有误差.

本章讲解直接法.

线性方程组$Ax = b$：三种特例

线性方程组$Ax=b$：三种特例

下面3种特殊线性代数方程组$Ax=b$的解，可以直接求出:

1. $A=\mathrm{diag}(a_{11},a_{22},\cdots,a_{nn})\Rightarrow x_i=\frac{b_i}{a_{ii}}(i=1,\cdots,n)$ $\longmapsto$ n次乘(除)法

线性方程组$Ax = b$：三种特例

下面3种特殊线性代数方程组$Ax = b$的解，可以直接求出:

1. $A = \mathrm{diag}(a_{11}, a_{22}, \cdots, a_{nn}) \Rightarrow x_i = \frac{b_i}{a_{ii}}(i = 1, \cdots, n) \longmapsto n$次乘(除)法

2. $A = \begin{pmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{pmatrix}$

线性方程组$Ax = b$：三种特例

下面3种特殊线性代数方程组$Ax = b$的解，可以直接求出:

1. $A = \mathrm{diag}(a_{11}, a_{22}, \cdots, a_{nn}) \Rightarrow x_i = \frac{b_i}{a_{ii}} (i = 1, \cdots, n) \longmapsto n$次乘(除)法

2. $$A = \begin{pmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{pmatrix} \Rightarrow x_i = \frac{b_i - \sum\limits_{j=1}^{i-1} l_{ij} x_j}{l_{ii}} \longmapsto \frac{n(n+1)}{2}$$ 次乘法

线性方程组$Ax=b$：三种特例

下面3种特殊线性代数方程组$Ax=b$的解，可以直接求出:

1. $A=\mathrm{diag}(a_{11},a_{22},\cdots,a_{nn})\Rightarrow x_i=\frac{b_i}{a_{ii}}(i=1,\cdots,n)\longmapsto n$次乘(除)法

2. $$A=\begin{pmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{pmatrix}\Rightarrow x_i=\frac{b_i-\sum\limits_{j=1}^{i-1} l_{ij}x_j}{l_{ii}}\longmapsto\frac{n(n+1)}{2}\text{ 次乘法}$$

3. $$A=\begin{pmatrix} u_{11} & u_{12} & \cdots & \cdots & u_{1n} \\ & u_{22} & \cdots & \cdots & u_{2n} \\ & & \ddots & \cdots & \vdots \\ & & & u_{(n-1)(n-1)} & u_{(n-1)n} \\ & & & & u_{nn} \end{pmatrix}$$

线性方程组$Ax=b$：三种特例

下面3种特殊线性代数方程组$Ax=b$的解，可以直接求出:

1. $A=\mathrm{diag}(a_{11},a_{22},\cdots,a_{nn})\Rightarrow x_i=\frac{b_i}{a_{ii}}(i=1,\cdots,n)\longmapsto n$次乘(除)法

2. $$A=\begin{pmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{pmatrix}\Rightarrow x_i=\frac{b_i-\sum\limits_{j=1}^{i-1} l_{ij}x_j}{l_{ii}}\longmapsto\frac{n(n+1)}{2}\text{ 次乘法}$$

3. $$A=\begin{pmatrix} u_{11} & u_{12} & \cdots & \cdots & u_{1n} \\ & u_{22} & \cdots & \cdots & u_{2n} \\ & & \ddots & \cdots & \vdots \\ & & & u_{(n-1)(n-1)} & u_{(n-1)n} \\ & & & & u_{nn} \end{pmatrix}\Rightarrow x_i=\frac{b_i-\sum\limits_{j=i+1}^{n} u_{ij}x_j}{u_{ii}}$$

$\mapsto\frac{n(n+1)}{2}$次乘法

消元法

消元法就是对增广矩阵(A, b)作行变换，变为上述的3种类型之一后再求解.

消元法

消元法就是对增广矩阵(A, b)作行变换，变为上述的3种类型之一后再求解.

例如，Gauss消元法即化为上三角形式.

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{pmatrix} \Longrightarrow \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ 0 & 0 & a_{33}^{(3)} & \cdots & a_{3n}^{(3)} & b_3^{(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn}^{(n)} & b_n^{(n)} \end{pmatrix}$$

具体步骤如下.

Gauss消元法

第一步： 第1行$\times\left(-\frac{a_{i1}}{a_{11}}\right)$ +第i行，$i=2,\cdots,n$.

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{pmatrix} \Longrightarrow \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} & b_n^{(2)} \end{pmatrix}$$

运算量： $(n-1)\cdot(n+1)$

第二步： 第2行$\times\left(-\frac{a_{i2}^{(2)}}{a_{22}^{(2)}}\right)$+第$i$行，$i=3,\cdots,n$.

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} & b_n^{(2)} \end{pmatrix} \Longrightarrow \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & b_1 \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ 0 & 0 & a_{33}^{(3)} & \cdots & a_{3n}^{(3)} & b_3^{(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & a_{n3}^{(3)} & \cdots & a_{nn}^{(3)} & b_n^{(3)} \end{pmatrix}$$

运算量：$(n-2)\cdot(1+n-1)=(n-2)n$

类似的做下去，第k步：第k行$\times\left(-\frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}\right)$+第$i$行，$i=k+1,\cdots,n$.

运算量：$(n-k)\cdot(1+n-k+1)=(n-k)(n-k+2)$.

类似的做下去，第k步： 第k行$\times\left(-\frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}\right)$+第$i$行，$i=k+1,\cdots,n$.

运算量：$(n-k)\cdot(1+n-k+1)=(n-k)(n-k+2)$.

$n-1$ 步以后，可得增广矩阵：

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & b_1 \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ 0 & 0 & a_{33}^{(3)} & \cdots & a_{3n}^{(3)} & b_3^{(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn}^{(n)} & b_n^{(n)} \end{pmatrix}$$

化为上三角方程组的运算量为$\sum_{k=1}^{n-1}(n-k)(n-k+2)$，加上 解上述上三角阵的运算量 $\frac{(n+1)n}{2}$，总共为

$$\frac{n^3}{3}+n^2-\frac{n}{3}=O(n^3)$$

在Gauss消元过程中，须确保 $a_{kk}^{(k)} \neq 0$，

在Gauss消元过程中，须确保 $a_{kk}^{(k)} \neq 0$，
等价于 A的所有顺序主子式均不为0，即

$$\det \begin{pmatrix} a_{11} & \cdots & a_{1i} \\ \vdots & \ddots & \vdots \\ a_{i1} & \cdots & a_{ii} \end{pmatrix} \neq 0, \quad i = 1, \cdots, n$$

在Gauss消元过程中，须确保 $a_{kk}^{(k)} \neq 0$，
等价于 A的所有顺序主子式均不为0，即

$$\det \begin{pmatrix} a_{11} & \cdots & a_{1i} \\ \vdots & \ddots & \vdots \\ a_{i1} & \cdots & a_{ii} \end{pmatrix} \neq 0, \quad i = 1, \cdots, n$$

这是Gauss消元法理论上可行的充要条件.

在Gauss消元过程中，须确保 $a_{kk}^{(k)} \neq 0$，
等价于 A的所有顺序主子式均不为0，即

$$\det \begin{pmatrix} a_{11} & \cdots & a_{1i} \\ \vdots & \ddots & \vdots \\ a_{i1} & \cdots & a_{ii} \end{pmatrix} \neq 0, \quad i = 1, \cdots, n$$

这是Gauss消元法理论上可行的充要条件.

注：对某些有解的问题，我们不能用Gauss消元法去求解； 另外，如果某个$a_{kk}^{(k)}$ 很小的话，可能会导致比较大的计算误差。

例：用单精度（每次运算结果保留4位小数）解二元方程组

$$\begin{cases} 0.0003x_1 + 3.0000x_2 = 2.0001 \\ 1.0000x_1 + 1.0000x_2 = 1.0000 \end{cases}$$

例：用单精度（每次运算结果保留4位小数）解二元方程组

$$\begin{cases} 0.0003x_1 + 3.0000x_2 = 2.0001 \\ 1.0000x_1 + 1.0000x_2 = 1.0000 \end{cases}$$

按Gauss消元法得

$$\begin{cases} 0.0003x_1 + 3.0000x_2 = 2.0001 \\ -9999.0000x_2 = -6666.0000 \end{cases}$$

例：用单精度（每次运算结果保留4位小数）解二元方程组

$$\begin{cases} 0.0003x_1 + 3.0000x_2 = 2.0001 \\ 1.0000x_1 + 1.0000x_2 = 1.0000 \end{cases}$$

按Gauss消元法得

$$\begin{cases} 0.0003x_1 + 3.0000x_2 = 2.0001 \\ -9999.0000x_2 = -6666.0000 \end{cases}$$

$$\Longrightarrow \quad x_2 = 0.6667, \quad x_1 = 0.0000$$

例：用单精度（每次运算结果保留4位小数）解二元方程组

$$\begin{cases} 0.0003x_1 + 3.0000x_2 = 2.0001 \\ 1.0000x_1 + 1.0000x_2 = 1.0000 \end{cases}$$

按Gauss消元法得

$$\begin{cases} 0.0003x_1 + 3.0000x_2 = 2.0001 \\ \quad -9999.0000x_2 = -6666.0000 \end{cases}$$

$$\Longrightarrow \quad x_2 = 0.6667, \quad x_1 = 0.0000$$

真实解为 $x_2 = \frac{2}{3} \approx 0.6667, \quad x_1 = \frac{1}{3} \approx 0.3333$

列主元消元法

在Gauss消元第k步消元之前，曾广矩阵(A,b)为

$$
\begin{pmatrix}
a_{11} & \cdots & \cdots & \cdots & a_{1n} & b_1 \\
\vdots & \ddots & \cdots & \cdots & \cdots & \vdots \\
0 & 0 & a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} & b_k^{(k)} \\
0 & 0 & a_{(k+1)k}^{(k)} & \cdots & a_{(k+1)n}^{(k)} & b_{(k+1)}^{(k)} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & a_{nk}^{(k)} & \cdots & a_{nn}^{(k)} & b_n^{(k)}
\end{pmatrix}
$$

列主元消元法

在Gauss消元第k步消元之前，曾广矩阵(A,b)为

$$\begin{pmatrix} a_{11} & \cdots & \cdots & \cdots & a_{1n} & b_1 \\ \vdots & \ddots & \cdots & \cdots & \cdots & \vdots \\ 0 & 0 & a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} & b_k^{(k)} \\ 0 & 0 & a_{(k+1)k}^{(k)} & \cdots & a_{(k+1)n}^{(k)} & b_{(k+1)}^{(k)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & a_{nk}^{(k)} & \cdots & a_{nn}^{(k)} & b_n^{(k)} \end{pmatrix}$$

- 列主元消元的基本思路： 先选列主元，再消元； 若$\max\limits_{k\leqslant i\leqslant n}|a_{ik}^{(k)}|=|a_{jk}^{(k)}|$，交换$k$行和$j$行， 即将第$k$到$n$列的模最大元通过行交换移到对角位置，再进行消元.

例：单精度解方程组（每次运算结果保留4位小数）

$$\begin{cases} 0.0003x_1 + 3.0000x_2 = 2.0001 \\ 1.0000x_1 + 1.0000x_2 = 1.0000 \end{cases}$$

按列主元消元法得

$$\begin{cases} 1.0000x_1 + 1.0000x_2 = 1.0000 \\ \qquad\qquad 2.9997x_2 = 1.9998 \end{cases}$$

例：单精度解方程组（每次运算结果保留4位小数）

$$\begin{cases} 0.0003x_1 + 3.0000x_2 = 2.0001 \\ 1.0000x_1 + 1.0000x_2 = 1.0000 \end{cases}$$

按列主元消元法得

$$\begin{cases} 1.0000x_1 + 1.0000x_2 = 1.0000 \\ \qquad\qquad 2.9997x_2 = 1.9998 \end{cases}$$

$$\Longrightarrow \quad x_2 = 0.6667, \quad x_1 = 0.3333$$

Gauss-Jordan消元法

Gauss-Jordan消元法

将Gauss消元第k步变为：

$$第k行\times\left(-\frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}\right)+第i行，\ i=1,\cdots,k-1,k+1,\cdots,n.$$

最后系数矩阵被消元成一对角阵，即称为Gauss-Jordan消元法.

也可这样认为：Gauss-Jordan消元即在Gauss消元完成后（此时系数矩阵为上三角阵），继续用对角元进行向上消元（打洞），直至变成对角阵，再求解.

Gauss-Jordan消元法计算复杂度比Gauss消元法略多，量级一样.

Gauss消元法回顾

Gauss消元法回顾

第1步： 第1行$\times\left(-\frac{a_{i1}}{a_{11}}\right)$+第$i$行，$i=2,\cdots,n$.

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{pmatrix} \Longrightarrow \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} & b_n^{(2)} \end{pmatrix}$$

$n-1$ 步以后，可得增广矩阵：

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & b_1 \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ 0 & 0 & a_{33}^{(3)} & \cdots & a_{3n}^{(3)} & b_3^{(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn}^{(n)} & b_n^{(n)} \end{pmatrix}$$

Gauss消元法回顾

第1步： 第1行$\times\left(-\frac{a_{i1}}{a_{11}}\right)$+第$i$行，$i=2,\cdots,n.$

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{pmatrix} \Longrightarrow \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} & b_n^{(2)} \end{pmatrix}$$

$n-1$ 步以后，可得增广矩阵：

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & b_1 \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ 0 & 0 & a_{33}^{(3)} & \cdots & a_{3n}^{(3)} & b_3^{(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn}^{(n)} & b_n^{(n)} \end{pmatrix}$$

- 问题：Gauss消元法本质上到底在做什么？

直接分解法

Gauss消元法第k步为:

$$第k行\times\left(-\frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}\right)+第i行，\quad i=k+1,\cdots,n$$

直接分解法

Gauss消元法第k步为:

$$\text{第}k\text{行}\times\left(-\frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}\right)+\text{第}i\text{行}，\quad i=k+1,\cdots,n$$

从矩阵理论来看，第k步消元等价于将曾广矩阵(A,b)左乘一个单位下三角矩阵

$$P_k=\begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & l_{k+1,k}^{(k)} & 1 & & \\ & & \vdots & & \ddots & \\ & & l_{nk}^{(k)} & & & 1 \end{pmatrix},\quad \text{其中}l_{ik}^{(k)}=\frac{-a_{ik}^{(k)}}{a_{kk}^{(k)}},i=k+1,\cdots,n$$

Gauss消元法实际上就是不断地用单位下三角阵P_k左乘(A, b)，最后得到一个上三角阵$P_{n-1} \cdot \cdots \cdot P_1 \cdot A$.

Gauss消元法实际上就是不断地用单位下三角阵P_k左乘(A, b)，最后得到一个上三角阵$P_{n-1} \cdot \cdots \cdot P_1 \cdot A$. 因此，整个Gauss消元法相当于左乘了一个单位下三角阵

$$\tilde{L} \triangleq P_{n-1} \cdots P_1 = \begin{pmatrix} 1 & & & & & \\ l_{21} & \ddots & & & & \\ & & 1 & & & \\ \vdots & & l_{k+1,k}^{(k)} & 1 & & \\ & & \vdots & & \ddots & \\ l_{n1} & \cdots & l_{nk}^{(k)} & \cdots & l_{n,n-1}^{(n)} & 1 \end{pmatrix}$$

最后使矩阵A最后变成上三角阵U，即$\tilde{L}A = U$.

Gauss消元法实际上就是不断地用单位下三角阵P_k左乘(A,b)，最后得到一个上三角阵$P_{n-1}\cdot\cdots\cdot P_1\cdot A$. 因此，整个Gauss消元法相当于左乘了一个单位下三角阵

$$\tilde{L} \triangleq P_{n-1}\cdots P_1 = \begin{pmatrix} 1 & & & & & \\ l_{21} & \ddots & & & & \\ & & 1 & & & \\ \vdots & & l_{k+1,k}^{(k)} & 1 & & \\ & & \vdots & & \ddots & \\ l_{n1} & \cdots & l_{nk}^{(k)} & \cdots & l_{n,n-1}^{(n)} & 1 \end{pmatrix}$$

最后使矩阵A最后变成上三角阵U，即$\tilde{L}A=U$.

结论：若A的各阶顺序主子式不为0，则 有$A=LU$ $(L=\tilde{L}^{-1})$. 注：Gauss消元法的实质就是要对矩阵 A 作 LU 分解！

直接分解法

要解方程组$Ax=b$，可先将A进行 LU分解，再转化为两个容易求解的方程组.

$$Ax=b \Longleftrightarrow LUx=b \Longleftrightarrow \begin{cases} Ly=b & (1) \\ Ux=y & (2) \end{cases}$$

先由(1)求y，再代入(2)求x.

注： 能进行直接分解法的条件与Gauss消元法相同，即各阶顺序主子式不为0.

LU Factorizations (分解)

- 问题：如何对矩阵A作 LU分解？ 若分解存在，分解是否唯一？

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

LU Factorizations (分解)

- 问题：如何对矩阵A作 LU分解？ 若分解存在，分解是否唯一？

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

- Check— equations：？； unknowns：？

LU Factorizations (分解)

- 问题：如何对矩阵A作 LU分解？ 若分解存在，分解是否唯一？

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

- Check— equations：？； unknowns：？
- 若要求L为单位下三角，则称为 Doolittle 分解.

LU Factorizations (分解)

- 问题：如何对矩阵A作 LU分解？ 若分解存在，分解是否唯一？

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

- Check— equations：？； unknowns：？
- 若要求L为单位下三角，则称为 Doolittle 分解.
- 若要求U为单位上三角，则称为 Crout's Factorization.

LU Factorizations (分解)

- 问题：如何对矩阵A作 LU分解？ 若分解存在，分解是否唯一？

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

- Check— equations：？； unknowns：？
- 若要求L为单位下三角，则称为 Doolittle 分解.
- 若要求U为单位上三角，则称为 Crout's Factorization.
- 若要求$l_{ij} = u_{ji}$ $(i, j = 1, \dots, n)$(要求A为对称正定矩阵)，则称为Cholesky 分解 (i.e., $A = LL^T \Longleftrightarrow \tilde{L}D\tilde{L}^T$ 分解).

Doolittle's Factorization

- 如何进行 LU 分解？

Doolittle's Factorization

- 如何进行 LU 分解?

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

Doolittle's Factorization

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & \mathbf{a_{22}} & \cdots & \mathbf{a_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} = \begin{pmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ & \mathbf{u_{22}} & \cdots & \mathbf{u_{2n}} \\ & & \ddots & \vdots \\ & & & u_{nn} \end{pmatrix}$$

Doolittle's Factorization

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & \mathbf{a_{22}} & \cdots & \mathbf{a_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} = \begin{pmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ & \mathbf{u_{22}} & \cdots & \mathbf{u_{2n}} \\ & & \ddots & \vdots \\ & & & u_{nn} \end{pmatrix}$$

1. 利用A第1行： $a_{1j} = u_{1j},\ j = 1, \cdots, n \Rightarrow u_{1j} = a_{1j}$

Doolittle's Factorization

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

1. 利用A第1行： $a_{1j} = u_{1j},\ j = 1, \cdots, n \Rightarrow u_{1j} = a_{1j}$

2. 利用A第1列： $a_{i1} = l_{i1} u_{11},\ i = 2, \cdots, n \Rightarrow l_{i1} = \frac{a_{i1}}{u_{11}}$

Doolittle's Factorization

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

1. **利用A第1行：** $a_{1j} = u_{1j},\ j = 1, \cdots, n \Rightarrow u_{1j} = a_{1j}$
2. **利用A第1列：** $a_{i1} = l_{i1} u_{11},\ i = 2, \cdots, n \Rightarrow l_{i1} = \frac{a_{i1}}{u_{11}}$
3. **利用第2行：** $a_{2j} = l_{21} u_{1j} + u_{2j},\ j = 2, \cdots, n \Rightarrow u_{2j} = a_{1j} - l_{21} u_{1j}$

Doolittle's Factorization

$$\begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & \mathbf{a_{22}} & \cdots & \mathbf{a_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} = \begin{pmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & \ldots & u_{1n} \\ & \mathbf{u_{22}} & \cdots & \mathbf{u_{2n}} \\ & & \ddots & \vdots \\ & & & u_{nn} \end{pmatrix}$$

1. **利用A第1行：** $a_{1j} = u_{1j},\ j = 1, \cdots, n \Rightarrow u_{1j} = a_{1j}$
2. **利用A第1列：** $a_{i1} = l_{i1} u_{11},\ i = 2, \cdots, n \Rightarrow l_{i1} = \frac{a_{i1}}{u_{11}}$
3. **利用第2行：** $a_{2j} = l_{21} u_{1j} + u_{2j},\ j = 2, \cdots, n \Rightarrow u_{2j} = a_{1j} - l_{21} u_{1j}$
4. **利用第2列：** $a_{i2} = l_{i1} u_{12} + l_{i2} u_{22},\ i = 3, \cdots, n \Rightarrow l_{i2} = \frac{a_{i2} - l_{i1} u_{12}}{u_{22}}$
5. $\cdots\cdots$

Doolittle's Factorization

$$\begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & \mathbf{a_{22}} & \cdots & \mathbf{a_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} = \begin{pmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & \ldots & u_{1n} \\ & \mathbf{u_{22}} & \cdots & \mathbf{u_{2n}} \\ & & \ddots & \vdots \\ & & & u_{nn} \end{pmatrix}$$

1. 利用A第1行： $a_{1j} = u_{1j},\ j = 1, \cdots, n \Rightarrow u_{1j} = a_{1j}$

2. 利用A第1列： $a_{i1} = l_{i1} u_{11},\ i = 2, \cdots, n \Rightarrow l_{i1} = \frac{a_{i1}}{u_{11}}$

3. 利用第2行： $a_{2j} = l_{21} u_{1j} + u_{2j},\ j = 2, \cdots, n \Rightarrow u_{2j} = a_{1j} - l_{21} u_{1j}$

4. 利用第2列： $a_{i2} = l_{i1} u_{12} + l_{i2} u_{22},\ i = 3, \cdots, n \Rightarrow l_{i2} = \frac{a_{i2} - l_{i1} u_{12}}{u_{22}}$

5. $\cdots\cdots$

计算顺序：

U的第一行$\longrightarrow$$L$的第一列$\longrightarrow$$U$的第二行$\longrightarrow$ L的第二列$\longrightarrow\cdots\cdots$

Crout's (Courant) Factorization

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

Crout分解与Doolittle分解类似.

Crout's (Courant) Factorization

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

Crout分解与Doolittle分解类似.

计算顺序：L的第一列$\longrightarrow U$的第一行$\longrightarrow L$的第二列$\longrightarrow$ U的第二行$\longrightarrow\cdots\cdots$

Crout's (Courant) Factorization

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

Crout分解与Doolittle分解类似.

计算顺序： L的第一列$\longrightarrow$$U$的第一行$\longrightarrow$$L$的第二列$\longrightarrow$ U的第二行$\longrightarrow\cdots\cdots$

注： 1. 直接分解法与Gauss消元法复杂度相同.

2. 多个系数矩阵相同的方程组 $Ax = b_i (i = 1, 2 \cdots, m)$，矩阵$A$只需分解一次，其中的 L, U 可保存备用.

Example (in class)

Find the Doolittle's Factorization for the following matrix A.

$$A = \begin{pmatrix} 4 & -2 & 1 \\ -2 & 4 & 1 \\ 1 & 1 & 3 \end{pmatrix} = L\,U$$

Example (in class)

Find the Doolittle's Factorization for the following matrix A.

$$A = \begin{pmatrix} 4 & -2 & 1 \\ -2 & 4 & 1 \\ 1 & 1 & 3 \end{pmatrix} = L\,U$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{pmatrix} \begin{pmatrix} 4 & -2 & 1 \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix}$$

Example (in class)

Find the Doolittle's Factorization for the following matrix A.

$$A = \begin{pmatrix} 4 & -2 & 1 \\ -2 & 4 & 1 \\ 1 & 1 & 3 \end{pmatrix} = L\,U$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{pmatrix} \begin{pmatrix} 4 & -2 & 1 \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ \frac{1}{4} & \frac{1}{2} & 1 \end{pmatrix} \begin{pmatrix} 4 & -2 & 1 \\ 0 & 3 & \frac{3}{2} \\ 0 & 0 & 2 \end{pmatrix}$$

Example (in class)

Find the Crout's Factorization for the following matrix A

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

Example (in class)

Find the Crout's Factorization for the following matrix A

$$A = \begin{pmatrix} 4 & -2 & 1 \\ -2 & 4 & 1 \\ 1 & 1 & 3 \end{pmatrix} = \begin{pmatrix} 4 & 0 & 0 \\ -2 & 3 & 0 \\ 1 & \frac{3}{2} & 2 \end{pmatrix} \begin{pmatrix} 1 & -\frac{1}{2} & \frac{1}{4} \\ 0 & 1 & \frac{1}{2} \\ 0 & 0 & 1 \end{pmatrix}$$

三对角阵的追赶法

将Crout's Factorization应用于三对角阵 ($A = LU$)

$$\begin{pmatrix} a_1 & b_1 & & \\ c_2 & a_2 & \ddots & \\ & \ddots & \ddots & b_{n-1} \\ & & c_n & a_n \end{pmatrix} = \begin{pmatrix} \alpha_1 & & & \\ \gamma_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \gamma_n & \alpha_n \end{pmatrix} \begin{pmatrix} 1 & \beta_1 & & \\ & 1 & \ddots & \\ & & \ddots & \beta_{n-1} \\ & & & 1 \end{pmatrix}$$

三对角阵的追赶法

将Crout's Factorization应用于三对角阵 ($A = LU$)

$$\begin{pmatrix} a_1 & b_1 & & \\ c_2 & a_2 & \ddots & \\ & \ddots & \ddots & b_{n-1} \\ & & c_n & a_n \end{pmatrix} = \begin{pmatrix} \alpha_1 & & & \\ \gamma_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \gamma_n & \alpha_n \end{pmatrix} \begin{pmatrix} 1 & \beta_1 & & \\ & 1 & \ddots & \\ & & \ddots & \beta_{n-1} \\ & & & 1 \end{pmatrix}$$

对方程组 $Ax = f$, 其计算过程为(这里$\gamma_i = c_i,\ i = 2, \ldots, n;\ f = (f_1, \ldots, f_n)^T$):

$$\begin{cases} \alpha_i = a_i - c_i\beta_{i-1} \qquad (c_1 = 0) & \\ \beta_i = b_i/\alpha_i & i = 1, \cdots, n \\ & \end{cases}$$

三对角阵的追赶法

将Crout's Factorization应用于三对角阵 ($A = LU$)

$$\begin{pmatrix} a_1 & b_1 & & \\ c_2 & a_2 & \ddots & \\ & \ddots & \ddots & b_{n-1} \\ & & c_n & a_n \end{pmatrix} = \begin{pmatrix} \alpha_1 & & & \\ \gamma_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \gamma_n & \alpha_n \end{pmatrix} \begin{pmatrix} 1 & \beta_1 & & \\ & 1 & \ddots & \\ & & \ddots & \beta_{n-1} \\ & & & 1 \end{pmatrix}$$

对方程组 $Ax = f$, 其计算过程为(这里$\gamma_i = c_i,\ i = 2, \ldots, n;\ f = (f_1, \ldots, f_n)^T$):

$$\begin{cases} \alpha_i = a_i - c_i\beta_{i-1} \qquad (c_1 = 0) & \\ \beta_i = {}^{b_i}/_{\alpha_i} & i = 1, \cdots, n \\ y_i = (f_i - c_i y_{i-1})/\alpha_i & i = 1, \cdots, n \qquad (\text{追}) \\ x_k = y_k - \beta_k x_{k+1}, \qquad k = n, \cdots, 1. & (\beta_n = x_{n+1} = 0) \qquad (\text{赶}) \end{cases}$$

Here, $y \equiv (y_1, \cdots, y_n)^T = Ux$.

三对角阵的追赶法

将Crout's Factorization应用于三对角阵 ($A = LU$)

$$\begin{pmatrix} a_1 & b_1 & & \\ c_2 & a_2 & \ddots & \\ & \ddots & \ddots & b_{n-1} \\ & & c_n & a_n \end{pmatrix} = \begin{pmatrix} \alpha_1 & & & \\ \gamma_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \gamma_n & \alpha_n \end{pmatrix} \begin{pmatrix} 1 & \beta_1 & & \\ & 1 & \ddots & \\ & & \ddots & \beta_{n-1} \\ & & & 1 \end{pmatrix}$$

对方程组 $Ax = f$, 其计算过程为(这里$\gamma_i = c_i,\ i = 2, \ldots, n;\ f = (f_1, \ldots, f_n)^T$)：

$$\begin{cases} \alpha_i = a_i - c_i\beta_{i-1} \qquad (c_1 = 0) & \\ \beta_i = {b_i}/{\alpha_i} & i = 1, \cdots, n \\ y_i = (f_i - c_i y_{i-1})/\alpha_i & i = 1, \cdots, n \qquad (\text{追}) \\ x_k = y_k - \beta_k x_{k+1}, \qquad k = n, \cdots, 1. & (\beta_n = x_{n+1} = 0) \qquad (\text{赶}) \end{cases}$$

Here, $y \equiv (y_1, \cdots, y_n)^T = Ux$.　计算复杂度：$5n - 4 = O(n)$.

对称正定矩阵的LDL^T分解

问题：对称正定阵 $A=(a_{ij})$，求其 LDL^T分解

$$A=\begin{pmatrix}1&&&\\l_{21}&1&&\\\vdots&\vdots&\ddots&\\l_{n1}&l_{n2}&\cdots&1\end{pmatrix}\begin{pmatrix}d_1&&&\\&d_2&&\\&&\ddots&\\&&&d_n\end{pmatrix}\begin{pmatrix}1&l_{21}&\cdots&l_{n1}\\&1&\cdots&l_{n2}\\&&\ddots&\vdots\\&&&1\end{pmatrix}$$

由A正定，则A可以进行Doolittle分解$A=L\cdot U$. 设

$$L=(l_{ij}),\ U=(u_{ij}),\ D=\mathrm{diag}(u_{11},\cdots,u_{nn}),\ \tilde{U}=D^{-1}U$$

则 $A=LU=LDD^{-1}U=LD\tilde{U}$，其中$\tilde{U}$为单位上三角阵，且这种分解是唯一的. 由$A$对称知 $A=A^T=\tilde{U}^TDL^T$；故有$L=\tilde{U}^T$, 即：$A=LDL^T$.

对称正定矩阵的LDL^T分解

问题：对称正定阵 $A=(a_{ij})$，求其 LDL^T分解

$$A=\begin{pmatrix}1 & & & \\ l_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & 1\end{pmatrix}\begin{pmatrix}d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n\end{pmatrix}\begin{pmatrix}1 & l_{21} & \cdots & l_{n1} \\ & 1 & \cdots & l_{n2} \\ & & \ddots & \vdots \\ & & & 1\end{pmatrix}$$

由A正定，则A可以进行Doolittle分解$A=L\cdot U$. 设

$$L=(l_{ij}),\ U=(u_{ij}),\ D=\mathrm{diag}(u_{11},\cdots,u_{nn}),\ \tilde{U}=D^{-1}U$$

则 $A=LU=LDD^{-1}U=LD\tilde{U}$，其中$\tilde{U}$为单位上三角阵，且这种分解是唯一的. 由$A$对称知 $A=A^T=\tilde{U}^TDL^T$；故有$L=\tilde{U}^T$, 即：$A=LDL^T$.

算法：先计算A的Doolittle分解（或Crout分解）$A=LU$，D取为U（L）的对角元即可.

向量范数

定义：若映射 $\|\cdot\|:\mathbb{R}^n\to\mathbb{R}^+\cup\{0\}$ 满足：

- 非负性 $\|X\|\geq 0$, 且 $\|X\|=0\Leftrightarrow X=0$
- 齐次性 $\forall a\in\mathbb{R}, \|aX\|=|a|\cdot\|X\|$
- 三角不等式 $\|X+Y\|\leq\|X\|+\|Y\|$

则称该映射$\|\cdot\|$为向量空间 $\mathbb{R}^n$上的一种向量范数.

向量范数

定义：若映射 $\|\cdot\|:\mathbb{R}^n\to\mathbb{R}^+\cup\{0\}$ 满足：

- 非负性　$\|X\|\geq 0$, 且 $\|X\|=0\Leftrightarrow X=0$
- 齐次性　$\forall a\in\mathbb{R}, \|aX\|=|a|\cdot\|X\|$
- 三角不等式　$\|X+Y\|\leq\|X\|+\|Y\|$

则称该映射$\|\cdot\|$为向量空间 $\mathbb{R}^n$上的一种向量范数.

设 $X=(x_1,x_2,\cdots,x_n)^T\in\mathbb{R}^n$. 称 $\|X\|_p\triangleq(\sum\limits_{i=1}^{n}|x_i|^p)^{\frac{1}{p}}$ $(p\geq 1)$ 为向量 X的$p-$范数.

向量范数

定义：若映射 $\|\cdot\|:\mathbb{R}^n\to\mathbb{R}^+\cup\{0\}$ 满足：

- 非负性 $\|X\|\geq 0$, 且 $\|X\|=0\Leftrightarrow X=0$
- 齐次性 $\forall a\in\mathbb{R}, \|aX\|=|a|\cdot\|X\|$
- 三角不等式 $\|X+Y\|\leq\|X\|+\|Y\|$

则称该映射$\|\cdot\|$为向量空间 $\mathbb{R}^n$上的一种向量范数.

设 $X=(x_1,x_2,\cdots,x_n)^T\in\mathbb{R}^n$. 称 $\|X\|_p\triangleq(\sum\limits_{i=1}^n|x_i|^p)^{\frac{1}{p}}$ $(p\geq 1)$ 为向量 X的$p-$范数. 常见的p范数有：

1. $\|X\|_1=\sum\limits_{i=1}^n|x_i|\longrightarrow$ 1-范数
2. $\|X\|_2=\sqrt{\sum\limits_{i=1}^n x_i^2}\longrightarrow$ 2-范数
3. $\|X\|_\infty=\max\{|x_i|\}\longrightarrow\infty$- 范数

例

例：计算向量$X = \begin{pmatrix} 1 \\ -2 \\ 1.5 \end{pmatrix}$的1范数，2范数和 ∞范数。

例

例：计算向量$X = \begin{pmatrix} 1 \\ -2 \\ 1.5 \end{pmatrix}$的1范数，2范数和 ∞范数。

解：由向量范数定义：

$$||X||_1 = \sum_{k=1}^{3} |x_k| = 1 + 2 + 1.5 = 4.5$$

$$||X||_2 = \sqrt{\sum_{k=1}^{3} x_k^2} = \sqrt{1 + (-2)^2 + (1.5)^2} = \sqrt{7.25} = 2.69258$$

$$||X||_\infty = \max_{1 \le k \le 3} \{|x_k|\} = 2$$

矩阵范数

定义：设$\|\cdot\|$是以n 阶方阵为变量的实值函数(映射)，且满足条件

（1） 非负性：$\|A\| \geq 0$，$\|A\| = 0$当且仅当$A = 0.$

（2） 齐次性：$\|\alpha A\| = |\alpha| \cdot \|A\|, \ \forall \alpha \in \mathbb{R}$

（3） 三角不等式: $\|A + B\| \leq \|A\| + \|B\|$

则映射$\|\cdot\|$定义了一种矩阵范数；

矩阵范数

定义：设$\|\cdot\|$是以n 阶方阵为变量的实值函数(映射)，且满足条件

（1） 非负性：$\|A\| \geq 0$，$\|A\| = 0$当且仅当$A = 0$.

（2） 齐次性：$\|\alpha A\| = |\alpha| \cdot \|A\|, \ \forall \alpha \in \mathbb{R}$

（3） 三角不等式: $\|A + B\| \leq \|A\| + \|B\|$

则映射$\|\cdot\|$定义了一种矩阵范数；若映射$\|\cdot\|$还满足相容性条件, 即

（4） 相容性: $\|AB\| \leq \|A\|\|B\|$, $\forall$ n 阶方阵 A, B

则称映射$\|\cdot\|$为一种相容矩阵范数.

矩阵范数

定义：设$\|\cdot\|$是以n阶方阵为变量的实值函数(映射)，且满足条件

（1） 非负性：$\|A\| \geq 0$，$\|A\| = 0$当且仅当$A = 0$.

（2） 齐次性：$\|\alpha A\| = |\alpha| \cdot \|A\|, \ \forall \alpha \in \mathbb{R}$

（3） 三角不等式: $\|A + B\| \leq \|A\| + \|B\|$

则映射$\|\cdot\|$定义了一种矩阵范数；若映射$\|\cdot\|$还满足相容性条件, 即

（4） 相容性: $\|AB\| \leq \|A\|\|B\|, \ \forall$ n 阶方阵 A, B

则称映射$\|\cdot\|$为一种相容矩阵范数.

定义：设$\|\cdot\|$ 是一种向量范数，可以由它定义矩阵的范数

$$\|A\| \triangleq \sup_{x \in R^n, x \neq 0} \frac{\|Ax\|}{\|x\|} = \sup_{x \in R^n, \|x\|=1} \|Ax\|$$

称之为向量范数$\|\cdot\|$的诱导(从属)矩阵范数$\|\cdot\|$.

矩阵范数

定义：设$\|\cdot\|$是以n 阶方阵为变量的实值函数(映射)，且满足条件

（1） 非负性：$\|A\| \geq 0$，$\|A\| = 0$当且仅当$A = 0$.

（2） 齐次性：$\|\alpha A\| = |\alpha| \cdot \|A\|, \ \forall \alpha \in \mathbb{R}$

（3） 三角不等式: $\|A + B\| \leq \|A\| + \|B\|$

则映射$\|\cdot\|$定义了一种矩阵范数；若映射$\|\cdot\|$还满足相容性条件, 即

（4） 相容性: $\|AB\| \leq \|A\|\|B\|, \ \forall$ n 阶方阵 A, B

则称映射$\|\cdot\|$为一种相容矩阵范数.

定义：设$\|\cdot\|$ 是一种向量范数，可以由它定义矩阵的范数

$$\|A\| \triangleq \sup_{x \in R^n, x \neq 0} \frac{\|Ax\|}{\|x\|} = \sup_{x \in R^n, \|x\|=1} \|Ax\|$$

称之为向量范数$\|\cdot\|$的诱导(从属)矩阵范数$\|\cdot\|$.

注： 不难验证**诱导(从属)矩阵范数必是相容矩阵范数。**

与3种常见的向量$p-$范数 ($\|\cdot\|_p$)对应的诱导矩阵范数分别为

- $\|A\|_1 = \max\limits_{1\le j\le n} \sum\limits_{i=1}^{n} |a_{ij}| \longrightarrow$ 列模(绝对值)和的最大值
- $\|A\|_\infty = \max\limits_{1\le i\le n} \sum\limits_{j=1}^{n} |a_{ij}| \longrightarrow$ 行模(绝对值)和的最大值
- $\|A\|_2 = \sqrt{\rho(A^TA)}$，也称为谱范数

与3种常见的向量$p-$范数 $(\|\cdot\|_p)$对应的诱导矩阵范数分别为

- $\|A\|_1 = \max\limits_{1\le j\le n}\sum\limits_{i=1}^{n}|a_{ij}| \longrightarrow$ 列模(绝对值)和的最大值
- $\|A\|_\infty = \max\limits_{1\le i\le n}\sum\limits_{j=1}^{n}|a_{ij}| \longrightarrow$ 行模(绝对值)和的最大值
- $\|A\|_2 = \sqrt{\rho(A^TA)}$，也称为谱范数

定义： 设 $\{\lambda_i\}_{i=1}^{n}$为A的所有特征值. $\rho(A) \triangleq \max\limits_{1\le i\le n}|\lambda_i|$ 表示A的模最大的特征值的模，称为A的谱半径.

例 ： $\|A\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{ij}|$

例：$\|A\|_1 = \max\limits_{1\le j\le n} \sum\limits_{i=1}^{n} |a_{ij}|$

Pf：

$$
\begin{aligned}
\|A\|_1 &= \sup_{\|x\|_1=1} \|Ax\|_1 \\
&= \sup_{\|x\|_1=1} \|(a_{11}x_1+\cdots+a_{1n}x_n,\cdots,a_{n1}x_1+\cdots+a_{nn}x_n)^T\|_1 \\
&= \sup_{\|x\|_1=1} \left(|a_{11}x_1+\cdots+a_{1n}x_n|+\cdots+|a_{n1}x_1+\cdots+a_{nn}x_n|\right) \\
&\le \sup_{\|x\|_1=1} \left((|a_{11}|+\cdots+|a_{n1}|)|x_1|+\cdots+(|a_{1n}|+\cdots+|a_{nn}|)|x_n|\right) \\
&\le \sup_{\|x\|_1=1} \left(\max_{1\le j\le n} \sum_{1\le i\le n} |a_{ij}|\right)\cdot\left(|x_1|+\cdots+|x_n|\right) = \max_{1\le j\le n} \sum_{1\le i\le n} |a_{ij}|
\end{aligned}
$$

设第k列列和最大，取$x_k = 1$和其余$x_i = 0$能使等号成立.

同理可证：$\|A\|_\infty = \max\limits_{1\le i\le n} \sum\limits_{j=1}^{n} |a_{ij}|$.

例 ： $\|A\|_2 = \sqrt{\rho(A^T A)}$.

例 ：$\|A\|_2 = \sqrt{\rho(A^TA)}$.

Pf：A^TA为对称半正定阵，故其有$\lambda_1, \cdots, \lambda_n$个非负特征根，且与其对应的$n$个单位长特征向量$u_1, \cdots, u_n$可构成$\mathbb{R}^n$的一 组标准正交基.

$\forall x \in \mathbb{R}^n, \|x\|_2 = 1$， 设$x = x_1u_1 + \cdots + x_nu_n$. 则：

$$\begin{aligned}
{\|Ax\|_2}^2 &= (Ax)^TAx = \left(A(x_1u_1 + \cdots + x_nu_n)\right)^T \cdot A(x_1u_1 + \cdots + x_nu_n) \\
&= (x_1u_1 + \cdots + x_nu_n)^T \cdot (A^TAx_1u_1 + \cdots + A^TAx_nu_n) \\
&= (x_1u_1 + \cdots + x_nu_n)^T \cdot (\lambda_1x_1u_1 + \cdots + \lambda_nx_nu_n) \\
&= \lambda_1x_1^2 + \cdots + \lambda_nx_n^2 \\
&\leq \max_i \lambda_i \cdot \sum_{1\leq i\leq n} |x_i|^2 = \max_i \lambda_i = \rho(A^TA)
\end{aligned}$$

且等号可以取到. 故 $\|A\|_2 = \sup_{\|x\|_2=1} \|Ax\|_2 = \sqrt{\rho(A^TA)}$.

定理：λ为A的特征值，$\|\cdot\|$为任一诱导的矩阵范数. 则$|\lambda| \leq \|A\|$.

定理：λ为A的特征值，$\|\cdot\|$为任一诱导的矩阵范数. 则$|\lambda| \leq \|A\|$.

Pf： 设λ对应的特征向量为x，则$Ax = \lambda x$. 由

$$\|A\| = \sup_{x \in R^n, x \neq 0} \frac{\|Ax\|}{\|x\|} = \sup_{x \in R^n, \|x\|=1} \|Ax\|$$

知对相应的向量范数有$\|Ax\| \leq \|A\| \, \|x\|$.

$$\|\lambda \cdot x\| = |\lambda| \cdot \|x\| = \|A \cdot x\| \leq \|A\| \cdot \|x\| \Longrightarrow |\lambda| \leq \|A\|$$

由定理知： $\rho(A) \leq \|A\|$.

注：等号有可能成立，例如：当A为实对称矩阵时，$\rho(A) = \|A\|_2$.

例

例：计算矩阵$A=\begin{pmatrix} 1.1 & -2 \\ 2.5 & -3.5 \end{pmatrix}$的1范数，2范数和 ∞范数和Euclid范数。

解：由矩阵范数定义：

$$||A||_1 =$$

例

例：计算矩阵$A = \begin{pmatrix} 1.1 & -2 \\ 2.5 & -3.5 \end{pmatrix}$的1范数，2范数和 ∞范数和Euclid范数。

解：由矩阵范数定义：

$$||A||_1 = Max\{1.1 + 2.5, |-2| + |-3.5|\} = 5.5$$

$$||A||_\infty =$$

例

例：计算矩阵$A = \begin{pmatrix} 1.1 & -2 \\ 2.5 & -3.5 \end{pmatrix}$的1范数，2范数和 ∞范数和Euclid范数。

解：由矩阵范数定义：

$$||A||_1 = Max\{1.1 + 2.5, |-2| + |-3.5|\} = 5.5$$

$$||A||_\infty = Max\{1.1 + |-2|, 2.5 + |-3.5|\} = 6$$

$$||A||_E = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}|a_{ij}|^2} =$$

例

例：计算矩阵$A=\begin{pmatrix}1.1 & -2\\ 2.5 & -3.5\end{pmatrix}$的1范数，2范数和 ∞范数和Euclid范数。

解：由矩阵范数定义：

$$||A||_1 = Max\{1.1+2.5, |-2|+|-3.5|\} = 5.5$$

$$||A||_\infty = Max\{1.1+|-2|, 2.5+|-3.5|\} = 6$$

$$||A||_E = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}|a_{ij}|^2} = \sqrt{1.1^2+2^2+2.5^2+3.5^2} = 4.86929$$

例

例：计算矩阵$A=\begin{pmatrix}1.1 & -2\\ 2.5 & -3.5\end{pmatrix}$的1范数，2范数和 ∞范数和Euclid范数。

解：由矩阵范数定义：

$$||A||_1 = Max\{1.1+2.5, |-2|+|-3.5|\} = 5.5$$

$$||A||_\infty = Max\{1.1+|-2|, 2.5+|-3.5|\} = 6$$

$$||A||_E = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}|a_{ij}|^2} = \sqrt{1.1^2+2^2+2.5^2+3.5^2} = 4.86929$$

由 $A^TA=\begin{pmatrix}7.46 & -10.95\\ -10.95 & 16.25\end{pmatrix}$得 $\lambda_1=23.6541,\quad \lambda_2=0.05591.$

例

例：计算矩阵$A=\begin{pmatrix}1.1 & -2\\ 2.5 & -3.5\end{pmatrix}$的1范数，2范数和 ∞范数和Euclid范数。

解：由矩阵范数定义：

$$||A||_1 = Max\{1.1+2.5, |-2|+|-3.5|\} = 5.5$$

$$||A||_\infty = Max\{1.1+|-2|, 2.5+|-3.5|\} = 6$$

$$||A||_E = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}|a_{ij}|^2} = \sqrt{1.1^2+2^2+2.5^2+3.5^2} = 4.86929$$

由 $A^TA=\begin{pmatrix}7.46 & -10.95\\ -10.95 & 16.25\end{pmatrix}$得 $\lambda_1=23.6541, \quad \lambda_2=0.05591$. 故

$$||A||_2 = \sqrt{\rho(A^TA)} = \sqrt{\lambda_1} = 4.86355$$

条件数和病态矩阵

定义：$\mathrm{Cond}_p(A) \triangleq \|A\|_p \times \|A^{-1}\|_p$ 称为$p-$ 范数意义下的矩阵 A的条件数.

条件数和病态矩阵

定义：$\text{Cond}_p(A) \triangleq \|A\|_p \times \|A^{-1}\|_p$ 称为$p-$ 范数意义下的矩阵 A的条件数.

例如，当A为对称阵且$p=2$时，$\text{Cond}_2(A) = \|A\|_2 \times \|A^{-1}\|_2 = \left|\frac{\lambda_1}{\lambda_n}\right|$，其中$\lambda_1, \lambda_n$分别为$A$的模最大与最小特征值.

引入条件数的概念，主要是为了讨论解随误差的变化，解的稳定性等. 一般说来，条件数大的方阵对应的线性方程组是病态的.

解随误差的变化

问题一：当b有小误差(扰动)δb时，会对方程组 $Ax = b$解有多大影响？

解随误差的变化

问题一：当b有小误差(扰动)δb时，会对方程组 $Ax=b$解有多大影响？

设b有小误差δb 时，方程的解为$x+\delta x$. 则：

解随误差的变化

问题一：当b有小误差(扰动)δb时，会对方程组 $Ax=b$解有多大影响？

设b有小误差δb 时，方程的解为$x+\delta x$. 则：

$$\begin{cases} Ax = b \\ A(x+\delta x) = b+\delta b \end{cases} \Longrightarrow \begin{cases} \|b\| \le \|A\|\|x\| \\ \|\delta x\| \le \|A^{-1}\| \cdot \|\delta b\| \end{cases}$$

$$\Longrightarrow \frac{\|\delta x\|}{\|x\|} \leqslant \left\|A^{-1}\right\| \cdot \|A\| \cdot \frac{\|\delta b\|}{\|b\|} = \mathrm{Cond}(A) \cdot \frac{\|\delta b\|}{\|b\|}$$

解随误差的变化

问题一：当b有小误差(扰动)δb时，会对方程组 $Ax = b$解有多大影响？

设b有小误差δb 时，方程的解为$x + \delta x$. 则：

$$\begin{cases} Ax = b \\ A(x + \delta x) = b + \delta b \end{cases} \qquad \Longrightarrow \begin{cases} \|b\| \leq \|A\|\|x\| \\ \|\delta x\| \leq \|A^{-1}\| \cdot \|\delta b\| \end{cases}$$

$$\Longrightarrow \frac{\|\delta x\|}{\|x\|} \leqslant \left\|A^{-1}\right\| \cdot \|A\| \cdot \frac{\|\delta b\|}{\|b\|} = \mathrm{Cond}(A) \cdot \frac{\|\delta b\|}{\|b\|}$$

这里Cond(A)描述了解的相对误差随b的相对误差的放大率.

例：$A=\begin{pmatrix} \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \end{pmatrix}$， $b=\begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$. 方程$Ax=b$的解为：

$$\begin{cases} x_1 = 72b_1 - 240b_2 + 180b_3 \\ x_2 = -240b_1 + 900b_2 - 720b_3 \\ x_3 = 180b_1 - 720b_2 + 600b_3 \end{cases}$$

例：$A=\begin{pmatrix} \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \end{pmatrix}$，$b=\begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$. 方程$Ax=b$的解为：

$$\begin{cases} x_1 = 72b_1 - 240b_2 + 180b_3 \\ x_2 = -240b_1 + 900b_2 - 720b_3 \\ x_3 = 180b_1 - 720b_2 + 600b_3 \end{cases}$$

不难看出该方程组的解对b的误差(扰动)比较敏感，即方程是病态的. 此时：

$$\mathrm{Cond}_1(A) = \mathrm{Cond}_\infty(A) = 2015,\ \mathrm{Cond}_2(A) = 1353.28\cdots$$

解随误差的变化

问题二：当A有小误差 δA时，会对方程组$Ax = b$解有多大影响？

解随误差的变化

问题二：当A有小误差 δA时，会对方程组$Ax=b$解有多大影响？

答案：$\frac{\|\delta x\|}{\|x\|} \le \frac{\left\|A^{-1}\right\| \cdot \|A\| \frac{\|\delta A\|}{\|A\|}}{1-\left\|A^{-1}\right\| \cdot \|A\| \frac{\|\delta A\|}{\|A\|}} = \frac{\mathrm{Cond(A)} \frac{\|\delta A\|}{\|A\|}}{1-\mathrm{Cond(A)} \cdot \frac{\|\delta A\|}{\|A\|}} \approx \mathrm{Cond}(A) \cdot \frac{\|\delta A\|}{\|A\|}$