

# Chapter 2

## 矩阵计算

### 2.1 线性方程组问题

#### 2.1.1 模型

科学计算应用经过建模和数值离散(如数据的插值与拟合,微分方程离散等)之后,都归结于求解一个或若干个线性方程组。用矩阵-向量的符号表示,即

$$A\mathbf{x} = \mathbf{b}. \quad (2.1)$$

其中,  $A$ 表示一个 $n \times n$ 的系数矩阵

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

右端项 $\mathbf{b} = [b_1, b_2, \dots, b_n]^T$ 是一个 $n$ 维列向量。关于矩阵的对称, 正定, 相似, 逆, 秩等概念参考线性代数/高等代数。

**定义2.1.1** (Singularity 奇异性). 称 $n \times n$ 矩阵 $A$ 是**非奇异的**, 如过它满足下列条件之一:

1.  $A$ 满秩, 即 $\text{rank}(A) = n$
2. 行列式不为零, 即  $|A| \neq 0$
3. 存在逆矩阵 $A^{-1}$

否则, 称矩阵 $A$ 是**奇异的**。

线性方程组解的存在唯一性与其稀疏矩阵的奇异性有必然的联系。事实上, 在 $A^{-1}$ 存在的情形, 可以用Crame法则获得该解, 即

$$\mathbf{x} = A^{-1}\mathbf{b}.$$

更一般地, 对于给定系数矩阵 $A$ 和右端项 $\mathbf{b}$ 的线性方程组, 其解的情况有如下三种:

1. 唯一解:  $A$ 非奇异,  $\mathbf{b}$ 任意;
2. 无数解:  $A$ 奇异,  $\mathbf{b} \in \text{span}(A)$ ;
3. 无解:  $A$ 奇异,  $\mathbf{b} \notin \text{span}(A)$ .

### 2.1.2 Gauss消元法

求解该问题最经典的方法是Gauss消元法，读者可能已经在线性代数或高等代数课程中学习过。消元法是线性方程组问题的直接求解方法，迭代法是另一类在实用计算中更常用的方法将在下一章中讲。此处仅举一例，目的是为了展示如何完成算法实现。本书中数值算法的实现采用了MATLAB，但是相关算法流程也被给出，因此读者可以根据自身情况采用其他任何一种计算机语言实现。

**例2.1.1** (Gauss消元法). 记矩阵 $A$ 和向量 $b$ 为某个线性方程组的系数矩阵和右端向量，请写出求解线性方程组 $Ax = b$ 的Gauss消元法过程。

```

1 function x = gauss(A, b);
2 n = size(A,1);
3 for k = 1:n-1
4     A(k,:) = A(k, :)/A(k,k);
5     for j = k+1:n
6         factor = -A(j,k)/A(k,k);
7         A(j,k:end) = A(j,k:end) + factor*A(k,k:end);
8         b(j) = b(j) + factor*b(k);
9     end
10 end
11 b(n) = b(n)/A(n,n);
12 for k = n:-1:2
13     b(1:k-1) = b(1:k-1) - A(1:k-1,k)*b(k);
14 end
15 x = b;
```

#### 消元法:条件数与稳定性

时间复杂度：考虑算法执行过程中所做的乘除法次数。一步消元过程所做的乘除适定性

$$\sum_{k=1}^n (n-k+1)^2 = \frac{1}{6}n(n+1)(2n+1).$$

回代过程所做的乘法次数为

$$\sum_{k=1}^n (n-k) = \frac{1}{2}n(n+1).$$

综上,高斯消去法所需总的乘除法次数为

$$\sum_{k=1}^n (n-k+1)^2 + \sum_{k=1}^n (n-k) = \frac{1}{3}n(n^2 + 3n - 1) \approx \frac{1}{3}n^3. \quad (2.2)$$

空间复杂度：不需额外存储空间、原址存储分解结果

稳定性：当 $|a_{kk}| \approx 0$ 时可导致浮点数溢出,用、条件数 $\kappa(A)$ 衡量

系数矩阵的度量是衡量线性方程组求解难易程度的一个重要标识。

常用的矩阵范数的定义是常见的， $\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^m |a_{ij}|$ ， $\|A\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^m |a_{ij}|$ ，和谱范

数 $\|A\|_2 = \sqrt{\lambda_{\max}(A^T A)}$ 。值得指出的是，任何矩阵范数满足 $\|A\| \geq \rho(A)$ 。

谱半径： $\rho(A) = \max |\lambda(A)|$ ，即绝对值最大的特征值。

矩阵的条件数:  $\kappa(A) = \|A\| \cdot \|A^{-1}\|$ 。若取 $L_2$ 范数 $\|\cdot\|_2$ , 则 $\kappa(A) = \frac{\sigma_{max}(A)}{\sigma_{min}(A)}$ , 其中,  $\sigma$ 表示最大和最小奇异值, 我们将在稍后的章节中介绍它们。有如下性质:

- 1、若 $A$ 是正规矩阵( $A^T A = A A^T$ ), 则 $\kappa(A) = \frac{\lambda_{max}(A)}{\lambda_{min}(A)}$ 。
- 2、若 $A$ 是酉矩阵( $A^T A = A A^T = I$ ), 则 $\kappa(A) = 1$ 。

### 2.1.3 直接法: 敏感性与误差分析

定义2.1.2 (矩阵条件数). 非奇异方阵 $A$ 的条件数定义为

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

其中,  $\|\cdot\|$ 是给定的矩阵范数。

可以从定义得到条件数的一些重要性质:

1. 对于任意非奇异方阵,  $\|A\| \geq 1$
2. 对于单位矩阵,  $\|A\| = 1$
3. 对于任意非奇异方阵 $A$ 和 $\gamma \in \mathbb{R}$ ,  $cond(\gamma A) = cond(A)$
4. 对于任何对角阵 $D = diag(d_i)$ ,  $cond(D) = \frac{\max_{1 \leq i \leq n} |d_i|}{\min_{1 \leq i \leq n} |d_i|}$ 。

### 2.1.4 消元法推广: LU分解

如果有一系列具有相同系数矩阵(不同右端项)的线性方程组要求解, 如固定网格下的时间相关偏微分方程的隐格式离散, 显然消元的过程都是重复的。利用系数矩阵的L(下三角)U(上三角)矩阵分解

$$A = LU$$

可以消除这种重复计算。具体地, 让我们以一个实例来说明: 若对矩阵 $A$ 的高斯消去过程时稳定的, 则存在

$$A = \begin{bmatrix} 2 & 4 & 4 & 2 \\ 3 & 3 & 12 & 6 \\ 2 & 4 & -1 & 2 \\ 4 & 2 & 1 & 1 \end{bmatrix} \xrightarrow{k=1} \begin{bmatrix} 2 & 4 & 4 & 2 \\ \frac{3}{2} & -3 & 6 & 3 \\ 1 & 0 & -5 & 0 \\ 2 & -6 & -7 & -3 \end{bmatrix}$$

$$\xrightarrow{k=2} \begin{bmatrix} 2 & 4 & 4 & 2 \\ \frac{3}{2} & -3 & 6 & 3 \\ 1 & 0 & -5 & 0 \\ 2 & 2 & -19 & -9 \end{bmatrix} \xrightarrow{k=3} \begin{bmatrix} 2 & 4 & 4 & 2 \\ \frac{3}{2} & -3 & 6 & 3 \\ 1 & 0 & -5 & 0 \\ 2 & 2 & \frac{19}{5} & -9 \end{bmatrix}$$

由最后一步的结果可以分离出 $L$ 和 $U$ 分别为:

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{3}{2} & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 2 & 2 & \frac{19}{5} & 1 \end{bmatrix} \quad U = \begin{bmatrix} 2 & 4 & 4 & 2 \\ 0 & -3 & 6 & 3 \\ 0 & 0 & -5 & 0 \\ 0 & 0 & 0 & -9 \end{bmatrix}$$

一般地, 对于 $n$ 阶非奇异矩阵 $A$ , 可以用如下脚本实现LU分解:

```

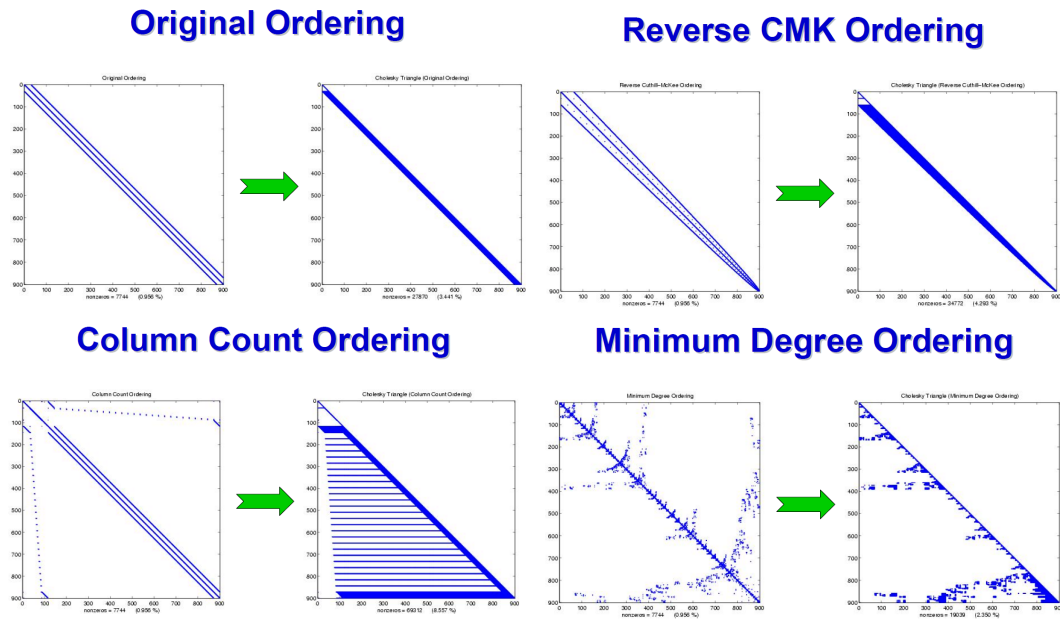
1 function [L, U] = lu_primer(A);
2 n = size(A,1);
3 for k = 1:n-1
4     A(k+1:n,k) = A(k+1:n,k)/A(k,k);
5     for i = k+1:n
6         A(i,k+1:end) = A(i,k+1:end) + A(i,k)*A(k,k+1:end);
7     end
8 end
9 L = tril(A); U = triu(A) + diag(A);

```

当矩阵 $A$ 是对称情形，显然分解所得的两个三角矩阵也是对称的，即 $U = L^T$ 。这也被称为对称矩阵的Dolittle分解。

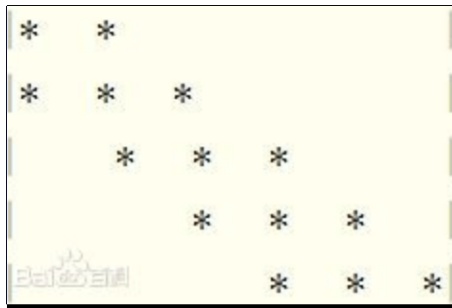
### 2.1.5 稀疏矩阵

稀疏矩阵排序技术主要是通过减少“填零”来实现的。如下所示的是一种减少“带宽”的技术：



### 三对角矩阵追赶法

矩阵中非零元素只占极少部分，带状矩阵与一般稀疏矩阵。可以设计特殊的存储结构以加快运算的速度。两点边值问题有限差分或有限元离散将会得到带三对角系数矩阵的线性方程组问题。所谓“三对角”，意味着只有矩阵的主对角线及其与其平行的上下两条对角线上的元素非零的情形，其结构如下所示：



### 2.1.6 古典迭代法

这就需要将线性方程组进行改写。可以将系数矩阵进行

$$\mathbf{x} = G\mathbf{x} + \mathbf{b}$$

那么,当 $x^{(0)}$ 给定后,可以利用迭代格式

$$\mathbf{x}^{(k)} = G\mathbf{x}^{(k-1)} + \mathbf{b}, \quad \forall k = 1, 2, \dots, n.$$

得到序列 $\{x^{(k)}\}_{k=0}^n$ .

- 序列的收敛只与算子 $G$ 有关,而与初值 $\mathbf{x}^{(0)}$ 的选取无关!
- 收敛性: 谱半径 $\rho(G) < 1$

**定理2.1.1** (一半迭代法收敛性). 迭代矩阵谱半径 $\rho(G) < 1$

#### Jacobi迭代法

为纪念普鲁士著名数学家雅可比

$$x_i^{(k)} = \frac{1}{G_{ii}} \left( b_i - \sum_{j=1}^{i-1} G_{ij} x_j^{(k-1)} - \sum_{j=i+1}^n G_{ij} x_j^{(k-1)} \right)$$

#### Gauss-Seidel迭代法

$$x_i^{(k)} = \frac{1}{G_{ii}} \left( b_i - \sum_{j=1}^{i-1} G_{ij} x_j^{(k)} - \sum_{j=i+1}^n G_{ij} x_j^{(k-1)} \right)$$

矩阵 $G$ 对角占优时两类迭代均收敛。但可以构造不同的例子显示: Gauss-Seidel法收敛时, Jacobi法可能不收敛; 另一方面, Jacobi法收敛时, Gauss-Seidel法也可能不收敛!

**定理2.1.2** (对称正定矩阵收敛). content

$$\begin{aligned}
 \text{对 } \mathbf{Ax}=\mathbf{b} \\
 \mathbf{A} = \begin{bmatrix} 0 & & & & \\ a_{21} & 0 & & & \\ a_{31} & a_{32} & 0 & & \\ \vdots & \vdots & \vdots & \ddots & \\ a_{n1} & a_{n2} & a_{n3} & \cdots & 0 \end{bmatrix} + \begin{bmatrix} a_{11} & & & & \\ & a_{22} & & & \\ & & \ddots & & \\ & & & a_{nn} & \\ & & & & \end{bmatrix} + \begin{bmatrix} 0 & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{23} & \cdots & a_{2n} & \\ & 0 & \cdots & a_{3n} & \\ & & \ddots & \vdots & \\ & & & \ddots & 0 \end{bmatrix} \\
 \text{下三角阵} \qquad \qquad \qquad \text{对角阵} \qquad \qquad \qquad \text{上三角阵}
 \end{aligned}$$

$$=L+D+U$$

$\because a_{ii} \neq 0$  (Jacobi假设)  $\therefore D$ 可逆

进一步, 有:  $(L+D+U)x=b$

$$Dx=-(L+U)x+b$$

迭代求解

$$x=-D^{-1}(L+U)x+b \rightarrow \text{理论分析}$$

### 超松弛(Relaxation)迭代法

在Gauss-Seidel基础上的一种加权修正, 分两步

1.

$$\tilde{x}_i^{(k)} = \frac{1}{G_{ii}} \left( g_i - \sum_{j=1}^{i-1} G_{ij} x_j^{(k)} - \sum_{j=i+1}^n G_{ij} x_j^{(k-1)} \right)$$

2.

$$\begin{aligned}
 x_i^{(k)} &= x_i^{(k-1)} + \omega(\tilde{x}_i^{(k)} - x_i^{(k-1)}) \\
 &= (1-\omega)x_i^{(k-1)} + \omega\tilde{x}_i^{(k)}
 \end{aligned}$$

A对称正定时, 松弛迭代收敛  $0 < \omega < 2$ . 收敛的快慢与松弛因子 $\omega$ 的选择有密切关系.  $0 < \omega < 1$  under-relaxation methods,  $\omega > 1$  over-relaxation methods. 如何选取最佳松弛因子, 使矩阵谱半径达到最小, 是一个尚未很好解决的问题. 经验上可取  $1.4 < \omega < 1.6$ .

在上图所列的算例中, 计算的精度要求误差达到小数点后第7位. 从计算的结果可以知道: Gauss-Seidel迭代法计算了34步, 而SOR迭代法只用了14步. 请使用相同的参数设定, 找出下述例题的SOR迭代解法中最优的 $\omega$ 的数值:

$$\begin{bmatrix} 4 & 3 & 0 \\ 3 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix}$$

**定理2.1.3** (SOR法收敛). content

## 2.2 超定线性方程组的最小二乘解

当矩阵A的行数多于列数, 即  $m > n$  的情形, 我们称问题

$$Ax=b$$

是一个超定线性方程组. 若  $\text{span}(A)$  是图凸集、 $\mathbf{r} = \mathbf{b} - \mathbf{Ax}$  是凸映射, 则存在唯一; A列满秩  $\leftrightarrow$  存在唯一; A秩亏损, 解仍存在, 但不唯一!

**Example**

- The linear system  $A\mathbf{x} = \mathbf{b}$  given by

$$\begin{aligned} 4x_1 + 3x_2 &= 24 \\ 3x_1 + 4x_2 - x_3 &= 30 \\ -x_2 + 4x_3 &= -24 \end{aligned}$$

has the solution  $(3, 4, -5)^t$ .

- Compare the iterations from the Gauss-Seidel method and the SOR method with  $\omega = 1.25$  using  $\mathbf{x}^{(0)} = (1, 1, 1)^t$  for both methods.

**Solution (1/3)**

For each  $k = 1, 2, \dots$ , the equations for the Gauss-Seidel method are

$$\begin{aligned} x_1^{(k)} &= -0.75x_2^{(k-1)} + 6 \\ x_2^{(k)} &= -0.75x_1^{(k)} + 0.25x_3^{(k-1)} + 7.5 \\ x_3^{(k)} &= 0.25x_2^{(k)} - 6 \end{aligned}$$

and the equations for the SOR method with  $\omega = 1.25$  are

$$\begin{aligned} x_1^{(k)} &= -0.25x_1^{(k-1)} - 0.9375x_2^{(k-1)} + 7.5 \\ x_2^{(k)} &= -0.9375x_1^{(k)} - 0.25x_2^{(k-1)} + 0.3125x_3^{(k-1)} + 9.375 \\ x_3^{(k)} &= 0.3125x_2^{(k)} - 0.25x_3^{(k-1)} - 7.5 \end{aligned}$$

**Gauss-Seidel Iterations**

$k$	0	1	2	3	...	7
$x_1^{(k)}$	1	5.250000	3.1406250	3.0878906		3.0134110
$x_2^{(k)}$	1	3.812500	3.8828125	3.9267578		3.9888241
$x_3^{(k)}$	1	-5.046875	-5.0292969	-5.0183105		-5.0027940

**SOR Iterations ( $\omega = 1.25$ )**

$k$	0	1	2	3	...	7
$x_1^{(k)}$	1	6.312500	2.6223145	3.1333027		3.0000498
$x_2^{(k)}$	1	3.5195313	3.9585266	4.0102646		4.0002586
$x_3^{(k)}$	1	-6.6501465	-4.6004238	-5.0966863		-5.0003486

### 2.2.1 正规方程解法

为了求解  $A\mathbf{x} = \mathbf{b}$ ，寻找使  $\|A\mathbf{x} - \mathbf{b}\|_2$  的梯度为零的  $\mathbf{x}$ ，故有正规方程（相对易求解）

$$(A^T A)\mathbf{x} = A^T \mathbf{b}$$

不管  $m$  与  $n$  的大小关系，上述方程组总是具有  $n$  个未知  $n$  个方程。Hessian 矩阵  $A^T A$  是对称的、正定的。

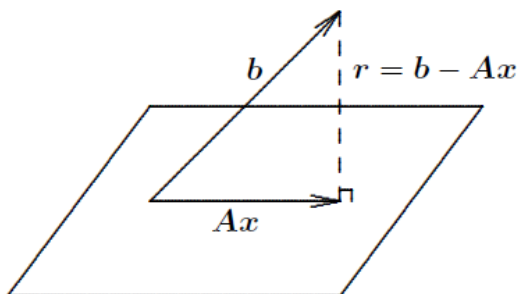
以上述例子为例，容易列出正规方程组 可求得正规方程的解为  $\mathbf{x}^T = [1236, 1943, 2416]^T$ ，将该

$$A^T A \mathbf{x} = \begin{bmatrix} 3 & -1 & -1 \\ -1 & 3 & -1 \\ -1 & -1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -651 \\ 2177 \\ 4069 \end{bmatrix} = A^T \mathbf{b}.$$

数值解代入原问题，可得到原方程残量的范数为  $\|\mathbf{r}\|_2^2 = 35$ 。

#### 几何的观点：正交投影法

也可以直接考虑求解  $A\mathbf{x} = \mathbf{b}$ 。对于  $m > n$  的问题，一般  $\mathbf{b} \notin \text{span}(A)$ ，则由投影关系 可知  $\mathbf{r}$  垂



直于  $A\mathbf{x} - \mathbf{b}$ ，即（等同于正规方程组）

$$A^T(A\mathbf{x} - \mathbf{b}) = 0$$

上述正规方程组方法即求  $A$  的广义逆

$$A^{-*} = (A^T A)^{-1} A^T$$

其中， $P = A^T A$  是在  $\text{span}(A)$  上的正交投影阵， $\text{cond}_2(A) = \|A\|_2 \|A^{-*}\|_2$  这个量体现了与秩亏损矩阵的接近程度，也容易因此知道正规方程组的条件数与原问题条件数几乎成平方关系。

#### 扰动分析

求解  $A^T A \mathbf{x} = A^T \mathbf{b}$ ，此时考虑右端项引起的扰动

$$A^T A(\mathbf{x} + \delta \mathbf{x}) = A^T (\mathbf{b} + \delta \mathbf{b})$$

$$\frac{\|\delta \mathbf{x}\|_2}{\|\mathbf{x}\|_2} \leq \dots \leq \text{cond}(A) \left( \frac{\|\mathbf{b}\|_2}{\|A\mathbf{x}\|_2} \right) \left( \frac{\|\delta \mathbf{b}\|_2}{\|\mathbf{b}\|_2} \right)$$



对于例1的分析:

$$\text{cond}(A) = \|A\|_2 \cdot \|A^{-*}\|_2 = 2 \cdot 1 = 2$$

$$\cos(\theta) = \frac{\|\mathbf{b}\|_2}{\|A\mathbf{x}\|_2} \approx 0.99999868$$

此例的条件数和角度 $\theta$ 都足够小, 问题良态! 忽略高阶项并简化之后, 有 因此,

对于矩阵  $\mathbf{A}$  的扰动  $\mathbf{A} + \mathbf{E}$ , 解的扰动由正规方程

$$(\mathbf{A} + \mathbf{E})^T (\mathbf{A} + \mathbf{E}) (\mathbf{x} + \Delta \mathbf{x}) = (\mathbf{A} + \mathbf{E})^T \mathbf{b}$$

$$\Delta \mathbf{x} \approx (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{E}^T \mathbf{r} - (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{E} \mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{E}^T \mathbf{r} - \mathbf{A}^+ \mathbf{E} \mathbf{x},$$

$$\begin{aligned} \frac{\|\Delta \mathbf{x}\|_2}{\|\mathbf{x}\|_2} &\lesssim \|(\mathbf{A}^T \mathbf{A})^{-1}\|_2 \cdot \|\mathbf{E}\|_2 \cdot \frac{\|\mathbf{r}\|_2}{\|\mathbf{x}\|_2} + \|\mathbf{A}^+\|_2 \cdot \|\mathbf{E}\|_2 \\ &= [\text{cond}(\mathbf{A})]^2 \cdot \frac{\|\mathbf{E}\|_2}{\|\mathbf{A}\|_2} \cdot \frac{\|\mathbf{r}\|_2}{\|\mathbf{A}\|_2 \cdot \|\mathbf{x}\|_2} + \text{cond}(\mathbf{A}) \cdot \frac{\|\mathbf{E}\|_2}{\|\mathbf{A}\|_2} \\ &\leq \left( [\text{cond}(\mathbf{A})]^2 \cdot \frac{\|\mathbf{r}\|_2}{\|\mathbf{A}\|_2} + \text{cond}(\mathbf{A}) \right) \frac{\|\mathbf{E}\|_2}{\|\mathbf{A}\|_2} \\ &= ([\text{cond}(\mathbf{A})]^2 \tan \theta + \text{cond}(\mathbf{A})) \frac{\|\mathbf{E}\|_2}{\|\mathbf{A}\|_2}. \end{aligned}$$

### 一个反例

考虑到数值稳定性, 也可以将最小二乘问题转换为其对于的增广 (Augment) 方程组或者三角最小二乘问题求解。

## 2.2.2 其他常用的解法

### Augment 方法

特点是化矩为方, 设解向量  $\mathbf{x}$  和残差向量  $\mathbf{r}$ , 让它们满足

$$\begin{aligned} \mathbf{r} + A\mathbf{x} &= \mathbf{b} \\ A^T \mathbf{r} &= \mathbf{0} \end{aligned}$$

写成矩阵形式, 有

$$\begin{bmatrix} I & A \\ A^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}$$

为了调节相对误差(rescale技巧), 有

$$\begin{bmatrix} \alpha I & A \\ A^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{r}/\alpha \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}$$

一般地, 取参数  $\alpha = \max_{i,j} |a_{i,j}|/1000$ 。会导致计算量增加, 但是对于大型问题可利用其结构的特殊性获得简化。在MATLAB中实际计算很有效!

**例 3.6 条件数平方效应** 考虑如下矩阵及扰动

$$A = \begin{bmatrix} 1 & 1 \\ \epsilon & -\epsilon \\ 0 & 0 \end{bmatrix}, \quad E = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -\epsilon & \epsilon \end{bmatrix},$$

其中  $\epsilon \ll 1$ , 接近  $\sqrt{\epsilon_{\text{mach}}}$ , 计算

$$\text{cond}(A) = 1/\epsilon, \quad \|E\|_2 / \|A\|_2 = \epsilon.$$

对右端向量  $b = [1, 0, \epsilon]^T$ , 有  $\|\Delta x\|_2 / \|x\|_2 = 0.5$ , 因此解的相对扰动大约为  $\text{cond}(A)$  乘以  $A$  的相对扰动. 对这个右端向量不存在条件数平方效应, 这是因为此时残差较小, 且  $\tan\theta \approx \epsilon$ , 在条件数估计中扰动平方项被忽略了.

另一方面, 对于右端向量  $b = [1, 0, 1]^T$ , 有  $\|\Delta x\|_2 / \|x\|_2 = 0.5/\epsilon$ , 因此解的相对扰动等于  $\text{cond}(A)$  的平方乘以  $A$  的相对扰动. 对这样的右端向量, 残差的范数大约为 1, 且  $\tan\theta \approx 1$ , 此时在扰动估计式中条件数平方项不能忽略, 这时解的敏感性非常高. ■

### 三角最小二乘问题

- 正交变换保欧氏距离
- 等价的三角最小二乘问题
- QR分解实现矩阵的正交分解;

### 2.2.3 矩阵的正交变换

Gram-Schmidt正交化是一般技巧, 上述两种实现方式是等价的, 只有在形式上的一些区别:

**算法 3.2 古典格拉姆-施密特正交化**

```
for k=1 to n           {列循环}
     $q_k = a_k$ 
    for j=1 to k-1      {从当前列}
         $r_{jk} = q_j^T a_k$ 
         $q_k = q_k - r_{jk} q_j$ 
    end
     $r_{kk} = \|q_k\|_2$ 
    if  $r_{kk} = 0$  then stop {如果线性}
     $q_k = q_k / r_{kk}$       {将当前列}
end
```

**算法 3.3 改进的格拉姆-施密特正交化**

```
for k=1 to n           {列循环}
     $r_{kk} = \|a_k\|_2$ 
    if  $r_{kk} = 0$  then stop {如果线性}
     $q_k = a_k / r_{kk}$       {将当前列}
    for j=k+1 to n      {减去后续}
         $r_{kj} = q_k^T a_j$ 
         $a_j = a_j - r_{kj} q_k$ 
    end
end
```

被动减与主动消。另外, 改进版本可以选主元, 以增加稳定性! 此外, 常见的正交化方法还有Householder变换和Givens旋转

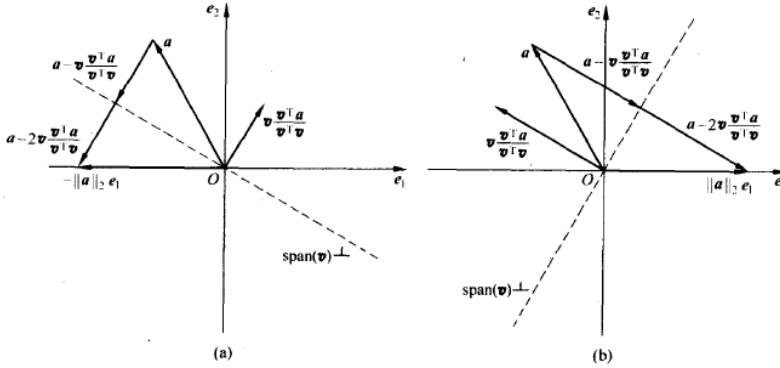
### Householder变换

$$Ha = \alpha e_1$$

其中,

$$H = I - \frac{2}{v^T v} v v^T$$

$$Ha = a - 2 \frac{\mathbf{v}^T \mathbf{a}}{\mathbf{v}^T \mathbf{v}} \mathbf{v} = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} - 2 \times \frac{15}{30} \times \begin{bmatrix} 5 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} -3 \\ 0 \\ 0 \end{bmatrix}$$



根据几何意义，可取

$$\mathbf{v} = \mathbf{a} - \|\mathbf{a}\|_2 \mathbf{e}_1$$

Household矩阵性质

- symmetric
- orthogonal

例：将向量  $\mathbf{a} = [2, 1, 2]^T$  实施household变换。第一步，取

$$\mathbf{v} = \mathbf{a} - \alpha \mathbf{e}_1 = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} - (-3) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \\ 2 \end{bmatrix}$$

第二步，实施变换，即矩阵向量乘法 思考：一次H变换完成一个列向量单位化，那么，如何利用这个算法实现矩阵的QR分解？

## Givens 变换

记旋转矩阵

$$G = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}$$

如：可把向量  $[a_1, a_2]^T$  旋转成  $[\sqrt{a_1^2 + a_2^2}, 0]^T$ ，此时，

$$\cos(\theta) = \frac{a_1}{\sqrt{a_1^2 + a_2^2}}, \quad \sin(\theta) = \frac{a_2}{\sqrt{a_1^2 + a_2^2}}$$

需要指出的是：变换矩阵G是正交矩阵，也不改变范数  $\|\cdot\|_2$ ；G变换一次得一个0元，H变换一次只剩一个非0元；也可用于实现QR分解；G变换工作量、存储量均比H变换多。

### 2.2.4 矩阵分解

在求解线性方程组的直接法中，我们已经注意到Gauss消元法可以用矩阵的LU分解法实现。事实上，对于更一般的矩阵还有其他的分解方式，它们适用于不同

#### QR分解

**定理2.2.1.** 设 $A$ 是 $m \times n$ 阶矩阵， $m \geq n$ .假如 $A$ 是列满秩的（非奇异），则存在一个唯一的正交阵 $Q(Q^T Q = I)$ 和唯一的具有正对角元 $r_{ii} > 0$ 的 $n \times n$ 阶上三角阵 $R$ ，使得 $A = QR$

证明 记 $A = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n]$ 的列向量形式，将这些列向量做正交化即得到正交矩阵 $Q[\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n]$ ，记 $r_{ji} = \mathbf{q}_j^T \mathbf{a}_i$ ，即有

$$\mathbf{a}_i = \sum_{j=1}^i r_{ji} \mathbf{q}_j,$$

当然，QR分解是可以通过Gram-Schmit正交化来实现的：

```

1 for  $i = 1, 2, \dots, n$  do
2    $\mathbf{q}_i = \mathbf{a}_i$ ;
3   for  $j = 1, 2, \dots, i$  do
4      $r_{ji} = \mathbf{q}_j^T \mathbf{a}_i$  或  $\mathbf{q}_j^T \mathbf{a}_i$ ;
5      $\mathbf{q}_i = \mathbf{a}_i - r_{ji} \mathbf{q}_j$ ;
6   end
7    $r_{ii} = \|\mathbf{q}_i\|_2$ ;
8   if  $r_{ii} == 0$  then
9     quit
10  else
11     $\mathbf{q}_i = \mathbf{q}_i / r_{ii}$ 
12  end
13 end
```

当然在实际应用中，Givens 变换、Householder 变换以及 Gram-Schmidt 正交化等都能实现QR分解;  $[Q,R]=qr(A)$ ;

**例2.2.1** (QR分解实例).

$$\begin{aligned}
 A &= \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} & \frac{1}{\sqrt{3}} \\ 0 & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} \frac{2}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & \frac{\sqrt{3}}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ 0 & 0 & \frac{2}{\sqrt{3}} \end{bmatrix} \\
 &:= QR
 \end{aligned}$$

#### 奇异值分解

**定理2.2.2.** 对于任意给定的 $m \times n$ 矩阵 $A$ 具有如下形式所谓SVD分解

$$A = U \Sigma V^T$$

其中,  $U$  是  $m \times m$  正交矩阵 ( $U^H U = I$ ),  $V$  是  $n \times n$  正交矩阵 ( $V^H V = I$ ),  $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$  是一个  $m \times n$  对角矩阵。

其中,  $\Sigma$  对角上的数值称为矩阵  $A$  的奇异值:

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0.$$

而  $U$  和  $V$  的列向量分别称为对应的左、右奇异向量。

**例2.2.2** (SVD分解的一个例子).

$$A = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} 2^{-\frac{1}{2}} & -2^{-\frac{1}{2}} \\ 2^{-\frac{1}{2}} & 2^{-\frac{1}{2}} \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 2^{-\frac{1}{2}} & -2^{-\frac{1}{2}} \\ 2^{-\frac{1}{2}} & 2^{-\frac{1}{2}} \end{bmatrix} := U \Sigma V^T$$

当极小化  $\|A\mathbf{x} - \mathbf{b}\|_2$  时, 若  $A$  秩亏或 "接近" 于秩亏, 常称为秩亏的最小二乘问题。SVD 可计算其低维近似解。

SVD 在现代科学应用中具有广泛的应用, 在求解诸如数字图像恢复、某些积分方程的解或从噪声数据中提取信号等病态问题中, 常用的正则化 (regularization) 来解决秩亏问题。下面是几个参考链接, 在图像分析中的应用, 感兴趣的读者可以自行参考:

- <http://zh.wikipedia.org/wiki/奇异值分解>
- <http://www.cnblogs.com/LeftNotEasy/archive/2011/01/19/svd-and-applications.html>
- SVD 的图像压缩

## 2.3 矩阵特征值问题

计算矩阵一个或多个特征值的数值方法是科学计算中关心的重要问题。

**定义2.3.1** (矩阵特征值). 给定一个代表  $n$  维向量空间上的线性变化的  $n \times n$  矩阵  $A$ , 希望找某个非零向量  $\mathbf{x}$  和标量  $\lambda$ , 满足

$$A\mathbf{x} = \lambda\mathbf{x}$$

则称  $\lambda$  为  $A$  的一个特征值,  $\mathbf{x}$  为相应的 (右) 特征向量。

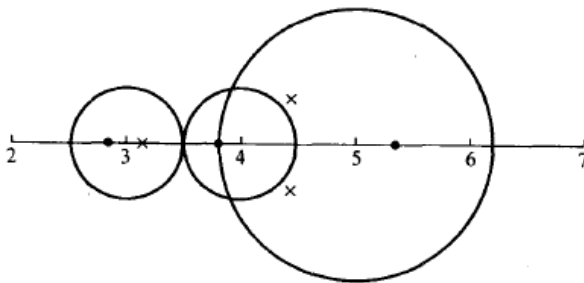
### 直接估算特征值

**谱半径**: 特征值集合  $\lambda(A)$  中模最大, 记为  $\rho(A)$

**定理2.3.1** (Gershgorin 圆盘定理). 设  $A$  是一个任意的矩阵, 则  $A$  的特征值  $\lambda$  位于由  $|\lambda - a_{ii}| \leq \sum_{j \neq i} |a_{ij}|$ , ( $i = 1, \dots, n$ ) 所定义的  $n$  个如下图所示的圆盘的并中:

**例2.3.1.** 请估算下列矩阵特征值所在的区间范围。

$$A_1 = \begin{bmatrix} 4.0 & -0.5 & 0.0 \\ 0.6 & 5.0 & -0.6 \\ 0.0 & 0.5 & 3.0 \end{bmatrix} \quad A_2 = \begin{bmatrix} 4.0 & 0.5 & 0.0 \\ 0.6 & 5.0 & 0.6 \\ 0.0 & 0.5 & 3.0 \end{bmatrix}$$



### 2.3.1 敏感性分析

考虑特征值的扰动问题

$$(A + E)(\mathbf{x} + \delta\mathbf{x}) = (\lambda + \delta\lambda)(\mathbf{x} + \delta\mathbf{x})$$

忽略二阶项，再化简、左乘左特征向量  $\mathbf{y}^H$ ，可得：

$$\delta\lambda \approx \frac{\mathbf{y}^H E \mathbf{x}}{\mathbf{y}^H \mathbf{x}},$$

那么，运用范数不等式可得估计(其中  $\theta$  是左、右特征向量夹角)

$$|\delta\lambda| \leq \frac{\|\mathbf{y}\|_2 \|\mathbf{x}\|_2}{|\mathbf{y}^H \mathbf{x}|} \|E\|_2 = \frac{1}{\cos \theta} \|E\|_2$$

实对称矩阵或复共轭矩阵的左右特征向量相同，良态！

求解特征值可粗略地分为：直接法和迭代法。

计算特征值的直接法可以从其定义出发，通过其与特征多项式零点的关系，将问题转换为方程求根问题：特征值问题  $A\mathbf{x} = \lambda\mathbf{x}$  等价于求解方程

$$(A - \lambda\mathbf{I})\mathbf{x} = \mathbf{0}$$

$\mathbf{x}$  有非零解的充要条件是系数矩阵奇异，即特征多项式的零点

$$\det(A - \lambda I) = 0$$

就是  $A$  的特征值。特征值问题  $\Leftrightarrow$  方程求根问题

**例2.3.2.** 算例：特征多项式零点

**例 4.3 特征多项式** 例 4.2 中的第 3 个矩阵的特征多项式为

$$\begin{aligned} \det\left(\begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) &= \det\left(\begin{bmatrix} 3-\lambda & 1 \\ 1 & 3-\lambda \end{bmatrix}\right) \\ &= (3-\lambda)(3-\lambda) - 1 \times 1 = \lambda^2 - 6\lambda + 8 = 0, \end{aligned}$$

利用求根公式，这个多项式的根为

$$\lambda = \frac{6 \pm \sqrt{36 - 32}}{2} = \frac{6 \pm 2}{2},$$

因此，特征值为  $\lambda_1 = 4, \lambda_2 = 2$ 。

迭代法是求解矩阵特征值的重要方法，迭代法通常用于稀疏矩阵，或能方便地执行矩阵向量乘法的隐式算子的情形。注意直接法中也需要用到迭代；如果其中用一个固定的迭代次数收敛几乎不失败，也称为直接法。幂法和反幂法是最简单的求解矩阵特征值的迭代法。

**幂法**

幂法只能求矩阵 $A$ 绝对值最大的特征值

```

1  给定 $\mathbf{x}_0$ ;
2  for  $i = 0, 1, \dots$  until convergence do
3      计算 $\mathbf{y}_{i+1} = A\mathbf{x}_i$ ;
4      计算近似特征向量 $\mathbf{x}_{i+1} = \mathbf{y}_{i+1} / \|\mathbf{y}_{i+1}\|_2$ ;
5      计算近似特征值 $\lambda_{i+1} = \mathbf{x}_{i+1}^T A \mathbf{x}_{i+1}$ ;
6  end

```

**反幂法**

相反，反幂法将找到最小特征值，但可以通过“位移”的技巧计算任意数 $\sigma$ 附近的那个特征值，其算法如下：

```

1  给定 $\mathbf{x}_0$ ;
2  for  $i = 0, 1, \dots$  until convergence do
3      计算 $\mathbf{y}_{i+1} = (A - \sigma I)^{-1} \mathbf{x}_i$ ;
4      计算近似特征向量:  $\mathbf{x}_{i+1} = \mathbf{y}_{i+1} / \|\mathbf{y}_{i+1}\|_2$ ;
5      计算近似特征值:  $\lambda_{i+1} = \mathbf{x}_{i+1}^T A \mathbf{x}_{i+1}$ ;
6  end

```

正交化是一类十分高效的计算技巧，将正交技巧用于矩阵特征值计算的迭代法中也有十分良好的效果。基于QR正交分解，人们提出了如下的QR算法：

```

1  给定 $A_0$ ;
2  for  $i = 0, 1, \dots$  until convergence do
3      分解 $A_i = Q_i R_i$  (QR分解);
4      计算 $A_{i+1} = R_i Q_i$ ;
5  end

```

利用该方法，我们将能同时得到矩阵 $A$ 的所有特征值和特征向量，最后  $\{A_i\}_{i=1}^{\infty}$  将收敛到一个上三角矩阵，其对角元素即为所求所有特征值。这是因为：

$$A_{i+1} = R_i Q_i = (Q_i^T Q_i) R_i Q_i = Q_i^T (Q_i R_i) Q_i = Q_i^T A_i Q_i$$

在这里， $A_i$ 等同于用正交迭代隐式计算矩阵 $Z_i^T Z_i$ ，且数值稳定。此外，带位移的QR迭代可加快收敛（当选择的位移接近特征值时二次收敛）

**瑞利Releigh商迭代**

```

1  给定 $\mathbf{x}_0$ 满足 $\|\mathbf{x}_0\|_2 = 1$ 以及精度 $TOL$ ;
2   $\rho_0 = \frac{\mathbf{x}_0^T A \mathbf{x}_0}{\mathbf{x}_0^T \mathbf{x}_0}$ ;
3  for  $i = 0, 1, \dots$  until  $\|A\mathbf{x}_{i+1} - \rho_{i+1}\mathbf{x}_{i+1}\|_2 < TOL$  do
4       $\mathbf{y}_{i+1} = (A - \rho_i I)^{-1} \mathbf{x}_i$ ;
5       $\mathbf{x}_{i+1} = \mathbf{y}_{i+1} / \|\mathbf{y}_{i+1}\|_2$ ;
6       $\rho_{i+1} = \frac{\mathbf{x}_{i+1}^T A \mathbf{x}_{i+1}}{\mathbf{x}_{i+1}^T \mathbf{x}_{i+1}}$ ;
7  end

```

A对称情形可用。该算法等同于逆迭代算法中位移取为瑞利商 $\rho(\mathbf{x}, A) := \frac{\mathbf{x}^T A \mathbf{x}}{\mathbf{x}^T \mathbf{x}}$ 。采用初值 $\mathbf{x}_0 = [0, \dots, 0, 1]^T$ 时与QR迭代得到的序列相同，是分析QR迭代的基础。Releigh商迭代法对单重特征值的计算是局部立方收敛的。

此外，三对角QR迭代是目前求对称三对角矩阵所有特征值最快( $O(n)^2$ )的方法( $n = 25$ 以内); Matlab指令eig, LAPACK程序ssyev(稠密)和sstev(三对角阵)。二分法和逆迭代(Bisection and inverse iteration): 二分法只求对称三对角阵特征值的一个子集(给定区间); 逆迭代可求相应的特征向量。最坏的情形(许多特征值很接近)不能保证精度; LAPACK程序ssyevx。分而治之(Divide-and-conquer): 求解 $n > 25$ 对称三对角矩阵所有特征值和特征向量的最快方法, (平均 $O(n)^{2.3}$ , 最坏 $O(n)^3$ ); LAPACK程序sstevd。Jacobi方法是求特征值问题最古老的方法(1846年); 可反复利用Givens变换实现; 通常比任何方法都慢( $O(n^3)$ )变换实现; 但结果可以更精确。

## 2.4 软件包与参考教材

### blas

提供矩阵-向量以及矩阵-矩阵运算得软件包, 支持不同精度(如float, double)等类型的运算, 是其他很多软件包的基础。

### lapack

集成了大量基本数值线性代数运算, 由计算数学领域多位专家主持开发工作. 是数值代数研究人员的必备工具。

### spooles

求解线性方程组的直接解法。开源库安装案例: Spooles 2.2

1. 下载最新版本，目前为2.2版本，更新慢;
2. 解压缩: `tar -xvf spooles.2.2.tar.gz`;
3. 修改目录下Make.inc中的编译命令，最简单是gcc;
4. 构建库: `make lib`.

使用案例-  $A$ 为14922阶的稀疏矩阵求解线性方程

$$AX = Y$$

where  $A$  is square, large and sparse, and  $X$  and  $Y$  are dense matrices with one or more columns. 在Spooles 2.2 中

1. 内存中建立线性方程组
  - (a) Constructing an InpMtx object that holds the entries of  $A$ ;
  - (b) Constructing a DenseMtx object that holds the entries of  $Y$ ;
  - (c) Constructing a DenseMtx object to hold the entries of  $X$ .
2. 利用Spooles中的Bridge类进行求解:



- (a) Initialization and setup step;
- (b) Factorization step;
- (c) Solution step.

### **mumps**

求解线性方程组的直接解法的另一个常用软件包。

### **superLU**

基于LU分解实现的线性方程求解器，有MPI/OpenMP并行版本以及串行版本

### **Intel MKL**

非开源，但是Linux系统下可以免费使用。针对Intel的CPU优化了编译指令，具有比其他开源软件快得多得执行速度。

数值线性代数算法是高性能科学计算的重要基础，有不少经典教材提供了更为详细的算法细节和拓展介绍，感兴趣的读者可以通过这些教材获得更多有用的信息：

1. Lloyd N. Trefethen & David Bau III: Numerical Linear Algebra
2. Gene H Golub & van Loan: Matrix Computation
3. James W. Demmel: Applied Numerical Linear Algebra

## **Exercises**

1. write gs elimination
2. perform dolittle decomposition
3. do the least square calculation
4. write gs elimination
5. perform dolittle decomposition
6. do the least square calculation
7. write gs elimination
8. perform dolittle decomposition
9. do the least square calculation
10. write gs elimination
11. perform dolittle decomposition
12. do the least square calculation