Lecture 5

第四章 LU及QR因式分解

大綱

- LU因式分解 (Factorization)
- 矩陣轉換 (Matrix Transformation)
- QR因式分解
- 進階問題

矩陣之因式分解

- 線性代數的問題
  - LU因式分解：求出一個下三角矩陣L和一個上三角矩陣U，以使得A=LU。
  - QR因式分解：求一個正交(Orthogonal)矩陣Q和一個右(上)三角矩陣R，以使得A=QR。如果一個矩陣的轉置，與它的逆矩陣相等，則此矩陣稱為正交矩陣。

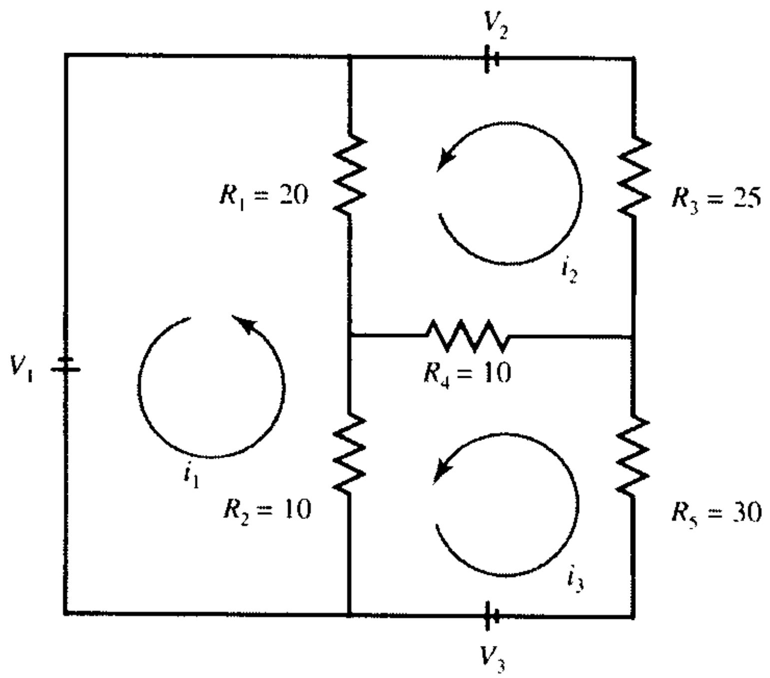
矩陣之因式分解

- LU因式分解的方法
  - 高斯消去法：在消去的過程中產生一個上三角矩陣；而與其相對應的下三角矩陣，則是由消去過程中所用的各個乘數所構成，且對角線元素均為1。
  - 直接計算法：直接分解，可能是更常用的方法，此一方法允許一個矩陣可以有不同的LU因式分解；有三種最常見的形式，對應於三種關於L和U之對角線元素的設定。

矩陣之因式分解

- QR因式分解的方法
  - ➢ *利用Householder轉換。*
  - ➢ *依據Givens旋轉。*
  - ➢ *在某些情形下，Householder轉換也可用於將一個矩陣轉換成(上)Hessenberg矩陣的相似轉換(Similarity transformation)。在相似轉換中，特徵值不變。在Hessenberg矩陣中，所有在第一下對角線(Subdiagonal)以下的元素全為零。*

問題4-A：電路分析



問題4-A：電路分析

左側迴路：$20(i_1 - i_2) + 10(i_1 - i_3) = 0$

右上迴路：$25i_2 + 10(i_2 - i_3) + 20(i_2 - i_1) = 0$

右下迴路：$30i_3 + 10(i_3 - i_2) + 10(i_3 - i_1) = 200$

化簡成線性方程組

$$30i_1 - 20i_2 - 10i_3 = 0$$

$$-20i_1 + 55i_2 - 10i_3 = 0$$

$$-10i_1 - 10i_2 + 50i_3 = 200$$

LU因式分解

- 對一個矩陣A做LU因式分解，是要找出一個下三角矩陣L和一個上三角矩陣U，以使得A=LU。

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

LU因式分解-高斯消去法

- 基本高斯消去將係數矩陣A的一列乘上一個適當的數，再將結果加到另一列而使得某一項之係數變成零。此消去過程，會逐漸將原矩陣A轉換成一個上三角矩陣U。

例題4.1 四乘四方程組

- 求四乘四矩陣的LU因式分解(使用高斯消去法)

$$A = \begin{bmatrix} 4 & 12 & 8 & 4 \\ 1 & 7 & 18 & 9 \\ 2 & 9 & 20 & 20 \\ 3 & 11 & 15 & 14 \end{bmatrix}$$

例題4.1 四乘四方程組

- 初始化

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} 4 & 12 & 8 & 4 \\ 1 & 7 & 18 & 9 \\ 2 & 9 & 20 & 20 \\ 3 & 11 & 15 & 14 \end{bmatrix}$$

步驟 1： $m(2,1) = -u(2,1)/u(1,1) = -1/4;$

$m(3,1) = -u(3,1)/u(1,1) = -2/4;$

$m(4,1) = -u(4,1)/u(1,1) = -3/4;$

針對此矩陣使用
消去法

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1/4 & 1 & 0 & 0 \\ 1/2 & 0 & 1 & 0 \\ 3/4 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} 4 & 12 & 8 & 4 \\ 0 & 4 & 16 & 8 \\ 0 & 3 & 16 & 18 \\ 0 & 2 & 9 & 11 \end{bmatrix}$$

步驟 2： $m(3,2) = -u(3,2)/u(2,2) = -3/4;$

$m(4,2) = -u(4,2)/u(2,2) = -2/4;$

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1/4 & 1 & 0 & 0 \\ 1/2 & 3/4 & 1 & 0 \\ 3/4 & 1/2 & 0 & 1 \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} 4 & 12 & 8 & 4 \\ 0 & 4 & 16 & 8 \\ 0 & 0 & 4 & 12 \\ 0 & 0 & 1 & 7 \end{bmatrix}$$

步驟 3： $m(4,3) = -u(4,3)/u(3,3) = -1/4;$

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1/4 & 1 & 0 & 0 \\ 1/2 & 3/4 & 1 & 0 \\ 3/4 & 1/2 & 1/4 & 1 \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} 4 & 12 & 8 & 4 \\ 0 & 4 & 16 & 8 \\ 0 & 0 & 4 & 12 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

將 **L** 乘以 **U** 以驗證結果：

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1/4 & 1 & 0 & 0 \\ 1/2 & 3/4 & 1 & 0 \\ 3/4 & 1/2 & 1/4 & 1 \end{bmatrix} \begin{bmatrix} 4 & 12 & 8 & 4 \\ 0 & 4 & 16 & 8 \\ 0 & 0 & 4 & 12 \\ 0 & 0 & 0 & 4 \end{bmatrix} = \begin{bmatrix} 4 & 12 & 8 & 4 \\ 1 & 7 & 12 & 9 \\ 2 & 9 & 20 & 20 \\ 3 & 11 & 15 & 14 \end{bmatrix}$$

例題4.1 LU因式分解程式

- **主程式：Main_LU_Factor.m**
- **LU因式分解程式:LU_Factor.m**

```
A=[4 12 8 4;1 7 18 9;2 9 20 20;3 11 15 14];

[L,U]=LU_Factor(A);

L

U
```

例題4.1 LU因式分解程式

- 結果：

```
L =
    1.0000         0         0         0
    0.2500    1.0000         0         0
    0.5000    0.7500    1.0000         0
    0.7500    0.5000    0.2500    1.0000


U =
     4    12     8     4
     0     4    16     8
     0     0     4    12
     0     0     0     4
```


例題4.1 LU因式分解程式

- 為更有效的使用電腦記憶體，L和U可以儲存在A原來所佔用的位置上。此時，矩陣L之對角線的1將不會被儲存。

三對角線矩陣的LU因式分解

- 一個三對角線矩陣T的LU因式分解所需的計算量(所需的電腦記憶體也一樣)，遠少於分解一個相同大小但完整的矩陣A所需的計算量。

三對角線矩陣的LU因式分解

- 對於以下向量所定義之三對角線矩陣

$$\begin{bmatrix} d_1 & a_1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & d_2 & a_2 & 0 & \cdots & 0 & 0 \\ 0 & b_1 & d_3 & a_3 & 0 & \cdots & 0 \\ 0 & 0 & b_2 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & 0 & b_3 & \ddots & a_{n-1} & 0 \\ 0 & 0 & \vdots & \ddots & \ddots & d_{n-1} & 0 \\ 0 & 0 & 0 & \cdots & 0 & b_{n-1} & d_n \end{bmatrix}$$

三對角線矩陣的LU因式分解

- 將三對角線矩陣以三個列矩陣表示可節省記憶體空間

$$d = [d_1 \quad d_2 \quad \ldots \quad d_{n-1} \quad d_n]\text{(主對角線)}$$
$$a = [a_1 \quad a_2 \quad \ldots \quad a_{n-1} \quad 0]\text{(對角線之上)}$$
$$b = [0 \quad b_2 \quad \ldots \quad b_{n-1} \quad b_n]\text{(對角線之下)}$$

三對角線矩陣的LU因式分解

第一階段的乘數爲 $m_{2,1} = -b_2/d_1$ 。

新的對角線元素爲 $D_2 = d_2 + a_1 m_{2,1}$ 。

- **MT=U**

$$\begin{bmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} d_1 & a_1 & 0 \\ b_2 & d_2 & a_2 \\ 0 & b_3 & d_3 \end{bmatrix} = \begin{bmatrix} d_1 & a_1 & 0 \\ 0 & D_2 & a_2 \\ 0 & b_3 & d_3 \end{bmatrix}$$

三對角線矩陣的LU因式分解

$$\boxed{\begin{aligned} &B_1 = 0,\ D_1 = d_1 \\ &B_i = b_i / D_{i-1} \\ &D_i = d_i - B_i a_{i-1} \end{aligned}}$$

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ B_2 & 1 & 0 & 0 \\ 0 & B_3 & 1 & 0 \\ 0 & 0 & B_4 & 1 \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} D_1 & a_1 & 0 & 0 \\ 0 & D_2 & a_2 & 0 \\ 0 & 0 & D_3 & a_3 \\ 0 & 0 & 0 & D_4 \end{bmatrix}$$

例題4.2 三對角線方程組的LU因式分解

$$M = \begin{bmatrix} 2 & 2 & 0 & 0 \\ 2 & 4 & 4 & 0 \\ 0 & 1 & 3 & 3 \\ 0 & 0 & 2 & 5 \end{bmatrix}$$

$d = [2 \quad 4 \quad 3 \quad 5]$ (主對角線)

$a = [2 \quad 4 \quad 3 \quad 0]$ (對角線之上)

$b = [0 \quad 2 \quad 1 \quad 2]$ (對角線之下)

例題4.2 三對角線方程組的LU因式分解

- $D_1=d_1=2$

對 $i=2$

$$B_2 = b_2/D_1 \quad = 2/2 \quad = 1$$
$$D_2 = d_2 - B_2 a_1 = 4-(1)(2) = 2$$

對 $i=3$

$$B_3 = b_3/D_2 \quad = 1/2 \quad = 1/2$$
$$D_3 = d_3 - B_3 a_2 = 3-(1/2)(4) = 1$$

例題4.2 三對角線方程組的LU因式分解

對 $i=4$

$$B_4 = b_4/D_3 \quad = 2/1 \quad = 2$$
$$D_4 = d_4 - B_4 a_3 = 5-(2)(3) = -1$$

以通式表示，此因式分解爲

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ B_2 & 1 & 0 & 0 \\ 0 & B_3 & 1 & 0 \\ 0 & 0 & B_4 & 1 \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} D_1 & a_1 & 0 & 0 \\ 0 & D_2 & a_2 & 0 \\ 0 & 0 & D_3 & a_3 \\ 0 & 0 & 0 & D_4 \end{bmatrix}$$

例題4.2 三對角線方程組的LU因式分解

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1/2 & 1 & 0 \\ 0 & 0 & 2 & 1 \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} 2 & 2 & 0 & 0 \\ 0 & 2 & 4 & 0 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

三對角線矩陣的LU因式分解之程式

- 主程式：Main_LU_Tridiag.m
- LU因式分解程式:LU_Tridiag.m

```
clear all;
close all;
d=[2 4 3 5];
a=[2 4 3 0];
b=[0 2 1 2];

[D, B] = LU_Tridiag(a, d, b);
D
B
```

三對角線矩陣的LU因式分解之程式

- 結果：

```
D =
        2        2        1      -1


B =
        0     1.0000     0.5000     2.0000
```


LU因式分解帶樞軸變換

- 對某些問題，在進行高斯消去法時，必須對係數矩陣A做列樞軸變換，對此種問題，可以求置換後(Permuted)之矩陣PA的LU因式分解，其中P是一個置換矩陣，它代表樞軸變換過程中所做的列對調。

例題4.3 LU因式分解帶樞軸變換

$$A = \begin{bmatrix} 2 & 6 & 10 \\ 1 & 3 & 3 \\ 3 & 14 & 28 \end{bmatrix}$$

- 初始化

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},\ U = \begin{bmatrix} 2 & 6 & 10 \\ 1 & 3 & 3 \\ 3 & 14 & 28 \end{bmatrix},\ P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

例題4.3 LU因式分解帶樞軸變換

將矩陣 **U** 和 **P** 的列 1 與列 3 對調：

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},\quad \mathbf{U} = \begin{bmatrix} 3 & 14 & 28 \\ 1 & 3 & 3 \\ 2 & 6 & 10 \end{bmatrix},\quad \mathbf{P} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

第一次消去得

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 1/3 & 1 & 0 \\ 2/3 & 0 & 1 \end{bmatrix},\quad \mathbf{U} = \begin{bmatrix} 3 & 14 & 28 \\ 0 & -5/3 & -19/3 \\ 0 & -10/3 & -26/3 \end{bmatrix},\quad \mathbf{P} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

例題4.3 LU因式分解帶樞軸變換

將矩陣 **U** 和 **P** 的列 2 和列 3 對調，以及 **L** 的第一行的列 2 和列 3：

$$\mathbf{L}=\begin{bmatrix}1&0&0\\2/3&1&0\\1/3&0&1\end{bmatrix}\quad \mathbf{U}=\begin{bmatrix}3&14&28\\0&-10/3&-26/3\\0&-5/3&-19/3\end{bmatrix}\quad \mathbf{P}=\begin{bmatrix}0&0&1\\1&0&0\\0&1&0\end{bmatrix}$$

第二次消去得到

$$\mathbf{L}=\begin{bmatrix}1&0&0\\2/3&1&0\\1/3&1/2&1\end{bmatrix},\quad \mathbf{U}=\begin{bmatrix}3&14&28\\0&-10/3&-26/3\\0&0&-2\end{bmatrix},\quad \mathbf{P}=\begin{bmatrix}0&0&1\\1&0&0\\0&1&0\end{bmatrix}$$

例題4.3之MATLAB程式

- **主程式：Main_LU_Pivot.m**
- **LU因式分解程式:LU_Pivot.m**

```
clear all;
close all;

A=[2 6 10;1 3 3;3 14 28];
[L,U,P]= LU_Pivot(A);
L
U
P
```

例題4.3之MATLAB程式

- 結果：

```
L =
    1.0000         0         0
    0.6667    1.0000         0
    0.3333    0.5000    1.0000

U =
    3.0000   14.0000   28.0000
         0   -3.3333   -8.6667
         0         0   -2.0000

P =
     0     0     1
     1     0     0
     0     1     0
```


LU因式分解帶樞軸變換

- 如果將L的元素，存放在矩陣A之對角線以下被化簡為零的位置，則即使有使用樞軸變換，L各元素的位置仍然正確。
- 如果將L存成獨立的矩陣，那麼在進行樞軸變換時，L中所有已儲存的元素，要和矩陣A一起做置換。但是L的對角線元素不置換。

LU因式分解-高斯消去法之分析

- 將A的第一列乘c，再將結果加到第二列以得到新的矩陣，可以表示成CA=B

$$\begin{bmatrix} 1 & 0 & 0 \\ c & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

其中

$$\alpha_{21} = ca_{11} + a_{21}, \alpha_{22} = ca_{12} + a_{22}, \alpha_{23} = ca_{13} + a_{23}$$

LU因式分解-高斯消去法之分析

- 將數個基本列運算結合在一個矩陣中。例如，高斯消去法的第一步可以寫成 $M_1A=A_1$:

$$\begin{bmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & 0 & 1 \end{bmatrix} \times \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & \alpha_{22} & \alpha_{23} \\ 0 & \alpha_{32} & \alpha_{33} \end{bmatrix}$$

LU因式分解-高斯消去法之分析

$$M_1 = \begin{bmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & 0 & 1 \end{bmatrix}, M_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & m_{32} & 1 \end{bmatrix}$$

- 其中$\mathbf{m_{21}}$和$\mathbf{m_{31}}$的值是用來將A轉換成$\mathbf{A_1}$(第一行對角線以下元素為零)，而$\mathbf{m_{32}}$是用來將$\mathbf{A_1}$轉換成$\mathbf{A_2}$(其第二行對角線以下元素也是零)。因此$\mathbf{A_2=U=M_2\ (M_1A)}$。

LU因式分解-高斯消去法之分析

- 對A的LU因式分解的驗證(使用高斯消去法的步驟但不含樞軸變換)

$$(M_1^{-1}M_2^{-1})M_2(M_1A) = M_1^{-1}(M_1A) = A$$

又 $\mathrm{U} = M_2(M_1A)$

$$(M_1^{-1}M_2^{-1})U = A$$

即$L = M_1^{-1}M_2^{-1}$

LU因式分解-高斯消去法之分析

$$\begin{bmatrix} 1 & 0 & 0 \\ -m_{21} & 1 & 0 \\ -m_{31} & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -m_{32} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -m_{21} & 1 & 0 \\ -m_{31} & -m_{32} & 1 \end{bmatrix}$$

- 因此，一個對角線元素均為1的下三角矩陣，其各個元素是消去過程中所用的乘數變號後置於適當位置，再加上消去過程所得到的上三角矩陣，就構成A的LU因式分解式。

因式分解加樞軸變換

- 在高斯消去法中加入列樞軸變換，得到的矩陣L和U是A經置換後的因式分解，此置換對應於樞軸變換時所做的列對調。
- 對矩陣A進行列對調，相當於在A的左側乘上一個置換矩陣，此置換矩陣的所有元素不是0就是1，且在每一行及每一列都恰僅有一個1。行對調，則是在A的右側乘上置換矩陣。

因式分解加樞軸變換

- 將A的第二列和第三列對調

$$\begin{bmatrix}1&0&0\\0&0&1\\0&1&0\end{bmatrix}\times\begin{bmatrix}a_{11}&a_{12}&a_{13}\\a_{21}&a_{22}&a_{23}\\a_{31}&a_{32}&a_{33}\end{bmatrix}=\begin{bmatrix}a_{11}&a_{12}&a_{13}\\a_{31}&a_{32}&a_{33}\\a_{21}&a_{22}&a_{23}\end{bmatrix}$$

因式分解加樞軸變換

- 假設在消去的第一階段沒有用到樞軸變換，但在第二階段出現樞軸變換。在建構$\mathbf{M_2}$之前，矩陣$\mathbf{M_1A}$的第二及第三列先對調；將此動作表示為乘上**置換矩陣P**。因為P的逆矩陣等於它本身，有恆等式

$$M_1^{-1}PPM_1A=A$$

代入M_2可得

$$(M_1^{-1}PM_2^{-1})(M_2PM_1A)=A$$

因式分解加樞軸變換

- 其中 $(M_2PM_1A)=A$ 是由高斯消去法所形成的上三角矩陣。但 $M_1^{-1}PM_2^{-1}$ 並不是下三角矩陣。

將矩陣兩邊皆乘以P，得

$(\mathrm{P}M_1^{-1}PM_2^{-1})(M_2PM_1A)=PA$

即$LU=PA$

$$PM_1^{-1}P=\begin{bmatrix}1&0&0\\0&0&1\\0&1&0\end{bmatrix}\begin{bmatrix}1&0&0\\-m_{21}&1&0\\-m_{31}&0&1\end{bmatrix}\begin{bmatrix}1&0&0\\0&0&1\\0&1&0\end{bmatrix}=\begin{bmatrix}1&0&0\\-m_{31}&1&0\\-m_{21}&0&1\end{bmatrix}$$

因式分解加樞軸變換

- 求下三角矩陣：

$(PM_1^{-1}P)M_2^{-1}=$

$$\begin{bmatrix}1&0&0\\-m_{31}&1&0\\-m_{21}&0&1\end{bmatrix}\begin{bmatrix}1&0&0\\0&1&0\\0&-m_{32}&1\end{bmatrix}=\begin{bmatrix}1&0&0\\-m_{31}&1&0\\-m_{21}&-m_{32}&1\end{bmatrix}$$

$=L$(下三角矩陣)

- 因此，L之元素的位置，反應出消去過程中的對調動作；由L和U相乘所得的矩陣，是原矩陣A置換後的結果。

因式分解加樞軸變換

- 結論

$$LU = A$$

$$\text{其中L} = (PM_1^{-1}P)M_2^{-1}(\text{下三角矩陣})$$

$$U = M_2PM_1A(\text{上三角矩陣})$$

LU因式分解-直接法

- 除了高斯消去法之外，另外一種可以求矩陣A之LU因式分解的方法，是以系統的方式建立起LU乘積的元素與A之元素的關係。最常用的三種LU因式分解形式對應到三種L和U之對角線元素的選擇方式。
  - *Doolittle法：L的對角線元素均為1。*
  - *Crout法：U的對角線元素均為1。*
  - *Cholesky法：L和U的對角線元素相等。*

Doolittle法

- **Doolittle形式的LU因式分解設定L的對角線元素均為1。因此，對一個三乘三矩陣A，問題就成為找出矩陣L和U以使得LU=A：**

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

Doolittle法

- **步驟1：先求$u_{11}=a_{11}$，然後求解U之第一列和L的第一行中的其它元素。**
- **步驟2：求u_{22}，然後是U的第二列與L的第二行中其餘的元素。**
- **持續此一過程，可以找出U及L所有的元素。**
- **Crout因式分解則使用類似的程序，只不過它令U的對角線元素為1而不是L的。**

Doolittle法

- 解題步驟：

(1) $u_{11} = a_{11}$

$u_{12} = a_{12}$

$u_{13} = a_{13}$

(2) $l_{21}u_{11} = a_{21} \Rightarrow u_{11} = \dfrac{a_{21}}{l_{21}}$

$l_{21}u_{12} + u_{22} = a_{22} \Rightarrow u_{22} = a_{22} - l_{21}u_{12}$

$l_{21}u_{13} + u_{23} = a_{23} \Rightarrow u_{23} = a_{23} - l_{21}u_{13}$

(3) $l_{31}u_{11} = a_{31} \Rightarrow l_{31} = \dfrac{a_{31}}{u_{11}}$

$l_{31}u_{12} + l_{32}u_{22} = a_{32} \Rightarrow l_{32} = \dfrac{a_{32} - l_{31}u_{12}}{u_{22}}$

$l_{31}u_{13} + l_{32}u_{23} + u_{33} = a_{33} \Rightarrow u_{33} = a_{33} - (l_{31}u_{13} + l_{32}u_{23}) = a_{33} - \begin{bmatrix} l_{31} & l_{32} \end{bmatrix} \begin{bmatrix} u_{13} \\ u_{23} \end{bmatrix}$

Doolittle法

- 通式：

$$u_{k,k} = a_{k,k} - l_{k,k-1}u_{k-1,k}$$

$$u_{k,j} = a_{k,j} - l_{k,k-1}u_{k-1,j}$$

$$l_{j,k} = \frac{a_{j,k} - l_{k,k-1}u_{k-1,j}}{u_{k,k}}$$

例題4.4 Doolittle之LU因式分解

- 矩陣A

$$A = \begin{bmatrix} 1 & 4 & 5 \\ 4 & 20 & 32 \\ 5 & 32 & 64 \end{bmatrix}$$

例題4.4 Doolittle之LU因式分解

$$\begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix}\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} = \begin{bmatrix} 1 & 4 & 5 \\ 4 & 20 & 32 \\ 5 & 32 & 64 \end{bmatrix}$$

一開始先解 **U** 的第一列和 **L** 的第一行：

$$u_{11} = a_{11} = 1, \quad u_{12} = a_{12} = 4, \quad u_{13} = a_{13} = 5;$$

$$\ell_{21}u_{11} = a_{21} = 4 \quad \Rightarrow \ell_{21} = 4$$

$$\ell_{31}u_{11} = a_{31} = 5 \quad \Rightarrow \ell_{31} = 5$$

例題4.4 Doolittle之LU因式分解

然後再用這些值求 **U** 的第二列和 **L** 的第二行：

$$\begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 5 & \ell_{32} & 1 \end{bmatrix}\begin{bmatrix} 1 & 4 & 5 \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} = \begin{bmatrix} 1 & 4 & 5 \\ 4 & 20 & 32 \\ 5 & 32 & 64 \end{bmatrix}$$

$$(4)(4) + u_{22} = 20 \quad \Rightarrow u_{22} = 20 - 16 = 4$$

$$(4)(5) + u_{23} = 32 \quad \Rightarrow u_{23} = 32 - 20 = 12$$

$$(5)(4) + \ell_{32} u_{22} = 32 \quad \Rightarrow \ell_{32} = (32 - 20)/4 = 3$$

例題4.4 Doolittle之LU因式分解

- 代入求出其他未知數

$$\begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 5 & 3 & 1 \end{bmatrix}\begin{bmatrix} 1 & 4 & 5 \\ 0 & 4 & 12 \\ 0 & 0 & u_{33} \end{bmatrix} = \begin{bmatrix} 1 & 4 & 5 \\ 4 & 20 & 32 \\ 5 & 32 & 64 \end{bmatrix}$$

$$(5)(5) + (3)(12) + u_{33} = 64 \quad \Rightarrow u_{33} = 64 - 25 - 36 = 3$$

$$\text{即} \mathrm{L} = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 5 & 3 & 1 \end{bmatrix}, \ U = \begin{bmatrix} 1 & 4 & 5 \\ 0 & 4 & 12 \\ 0 & 0 & 3 \end{bmatrix}$$

例題4.4 Doolittle之MATLAB程式

- 主程式：Main_LU_Doolittle.m
- LU因式分解程式:LU_Doolittle.m

```
clear all;
close all;

A=[1 4 5;4 20 32;5 32 64];

[L, U] = LU_Doolittle(A);
L
U
```


例題4.4 Doolittle之MATLAB程式

- 結果：

```
L =
       1       0       0
       4       1       0
       5       3       1

U =
       1       4       5
       0       4      12
       0       0       3
```

Crout法之因式分解

- Crout形式的LU因式分解設定U的對角線元素均為1。因此，對一個三乘三矩陣A，問題就成為找出矩陣L和U以使得LU=A：

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

例題4.4 Crout之LU因式分解

- 矩陣A

$$A = \begin{bmatrix} 1 & 4 & 5 \\ 4 & 20 & 32 \\ 5 & 32 & 64 \end{bmatrix}$$

$$\begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix}\begin{bmatrix} 1 & u_{12} & u_{13} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 4 & 5 \\ 4 & 20 & 32 \\ 5 & 32 & 64 \end{bmatrix}$$

例題4.4 Crout之MATLAB程式

- 主程式：Main_LU_Crout.m
- LU因式分解程式:LU_Crout.m

```
clear all;
close all;

A=[1 4 5;4 20 32;5 32 64];

[L, U] = LU_ Crout(A);
L
U
```


例題4.4 Crout之MATLAB程式

- 結果：

```
L =
      1     0     0
      4     4     0
      5    12     3

U =
      1     4     5
      0     1     3
      0     0     1
```

Homework 5

- 推導Crout法之因式分解的通式，並撰寫成Matlab程式：LU_Crout.m

對稱矩陣的因式分解

- 如果矩陣A是對稱正定(Symmetric positive definite)，則Cholesky因式分解是一種非常簡便的LU因式分解法，此種形式中的上三角矩陣U恰好是下三角矩陣L的轉置$U=L^T$；亦即$A=LL^T$。除此之外，它只需要$n(n+1)/2$個儲存位置，而不是n^2個。因此，Cholesky因式分解所需的計算量，只有非對稱矩陣的一半。

Cholesky法之因式分解

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

其中L和U對應的對角線元素必須相等。

- 若A為正定，則$U=L^T$，即$A=LL^T$

Cholesky法之因式分解

- 如果矩陣A是對稱且正定(見1.2.2節)，進行Cholesky因式分解時無需做樞軸變換及尺度化運算。
- 如果A不為正定，在計算中可能會出現對負數開根號的情形(參考Atkinson,1989)。
- Cholesky因式分解，其形式為$A=LDL^T$，它會產生一個單位下三角矩陣L(對角線都是1)並避免基本Cholesky法中開根號的運算。

Cholesky法之因式分解

- 解題步驟：

$$(1)\ x_{11}^2 = a_{11} \Rightarrow x_{11} = \pm\sqrt{a_{11}}$$

$$x_{11}u_{12} = a_{12} \Rightarrow u_{12} = \frac{a_{12}}{x_{11}}$$

$$x_{11}u_{13} = a_{13} \Rightarrow u_{13} = \frac{a_{13}}{x_{11}}$$

$$(2)\ l_{21}x_{11} = a_{21} \Rightarrow l_{21} = \frac{a_{21}}{x_{11}}$$

$$l_{31}x_{11} = a_{31} \Rightarrow l_{31} = \frac{a_{31}}{x_{11}}$$

$$l_{21}u_{12} + x_{22}^2 = a_{22} \Rightarrow x_{22} = \pm\sqrt{a_{22} - l_{21}u_{12}}$$

$$l_{21}u_{13} + x_{22}u_{23} = a_{23} \Rightarrow u_{23} = \frac{a_{23} - l_{21}u_{13}}{x_{22}}$$

$$(3)\ l_{31}u_{12} + l_{32}x_{22} = a_{32} \Rightarrow l_{32} = \frac{a_{32} - l_{31}u_{12}}{x_{22}}$$

$$l_{31}u_{13} + l_{32}u_{23} + x_{33}^2 = a_{33} \Rightarrow x_{33} = \pm\sqrt{a_{33} - (l_{31}u_{13} + l_{32}u_{23})} = \pm\sqrt{a_{33} - \begin{bmatrix} l_{31} & l_{32} \end{bmatrix}\begin{bmatrix} u_{13} \\ u_{23} \end{bmatrix}}$$

Cholesky法之因式分解

- 通式：

$$x_{k,k} = \pm\sqrt{a_{k,k} - l_{k,k-1}u_{k-1,k}}$$

$$l_{i,k} = \frac{a_{i,k} - l_{i,k-1}u_{k-1,k}}{x_{k,k}}$$

$$u_{k,i} = \frac{a_{k,i} - l_{k,k-1}u_{k-1,i}}{x_{k,k}}$$

Cholesky法之因式分解

- 通式：(若A為對稱正定)

$$x_{k,k} = \pm\sqrt{a_{k,k} - l_{k,k-1} l^T_{k,k-1}}$$

$$l_{i,k} = \frac{a_{i,k} - l_{i,k-1} l^T_{i,k-1}}{l_{k,k}}$$

例題4.5 Cholesky法之LU因式分解

- 有一對稱矩陣

$$A = \begin{bmatrix} 1 & 4 & 5 \\ 4 & 20 & 32 \\ 5 & 32 & 64 \end{bmatrix}$$

例題4.5 Cholesky法之LU因式分解

$$A=\begin{bmatrix} x_{11} & 0 & 0 \\ l_{21} & x_{22} & 0 \\ l_{31} & l_{32} & x_{33} \end{bmatrix}\begin{bmatrix} x_{11} & u_{12} & u_{13} \\ 0 & x_{22} & u_{23} \\ 0 & 0 & x_{33} \end{bmatrix}=\begin{bmatrix} 1 & 4 & 5 \\ 4 & 20 & 32 \\ 5 & 32 & 64 \end{bmatrix}$$

- **步驟1：**

$$x_{11}x_{11}=a_{11}=1 \Rightarrow x_{11}=1$$
$$x_{11}u_{12}=a_{12}=4 \Rightarrow u_{12}=4/1=4$$
$$x_{11}u_{13}=a_{13}=5 \Rightarrow u_{13}=5/1=5$$
$$\ell_{21}x_{11}=a_{21}=4 \Rightarrow \ell_{21}=4/1=4$$
$$\ell_{31}x_{11}=a_{31}=5 \Rightarrow \ell_{31}=5/1=5$$

若 **A** 爲對稱，$a_{1j}=a_{j1}$，則自動有 $\ell_{j1}=u_{1j}$。

例題4.5 Cholesky法之LU因式分解

- **步驟2：**

$$\begin{bmatrix} 1 & 0 & 0 \\ 4 & x_{22} & 0 \\ 5 & \ell_{32} & x_{33} \end{bmatrix}\begin{bmatrix} 1 & 4 & 5 \\ 0 & x_{22} & u_{23} \\ 0 & 0 & x_{33} \end{bmatrix}=\begin{bmatrix} 1 & 4 & 5 \\ 4 & 20 & 32 \\ 5 & 32 & 64 \end{bmatrix}$$

$$(4)(4)+(x_{22})(x_{22})=20 \Rightarrow x_{22}=(20-16)^{1/2}=2$$
$$(4)(5)+(x_{22})u_{23}=32 \Rightarrow u_{23}=(32-20)/2=6$$
$$(5)(4)+\ell_{32}(x_{22})=32 \Rightarrow \ell_{32}=(32-20)/2=6$$

例題4.5 Cholesky法之LU因式分解

- **步驟3：**

$$\begin{bmatrix} 1 & 0 & 0 \\ 4 & 2 & 0 \\ 5 & 6 & x_{33} \end{bmatrix} \times \begin{bmatrix} 1 & 4 & 5 \\ 0 & 2 & 6 \\ 0 & 0 & x_{33} \end{bmatrix} = \begin{bmatrix} 1 & 4 & 5 \\ 4 & 20 & 32 \\ 5 & 32 & 64 \end{bmatrix}$$

$$(5)(5)+(6)(6)+(x_{33})(x_{33})=64$$

$$x_{33}=\sqrt{64-25-36}=\sqrt{3}$$

- **LU因式分解滿足L=U'**

$$即\mathrm{L}=\begin{bmatrix} 1 & 0 & 0 \\ 4 & 2 & 0 \\ 5 & 6 & \sqrt{3} \end{bmatrix},\ U=\begin{bmatrix} 1 & 4 & 5 \\ 0 & 2 & 6 \\ 0 & 0 & \sqrt{3} \end{bmatrix}$$

Cholesky法之MATLAB程式

- **主程式：Main_LU_Cholesky.m**
- **LU因式分解程式:LU_Cholesky.m**

```
clear all;
close all;

A=[1 4 5;4 20 32;5 32 64];

[L, U] = LU_Cholesky(A);

L

U
```

Cholesky法之MATLAB程式

- 結果：

```
L =
    1.0000         0         0
    4.0000    2.0000         0
    5.0000    6.0000    1.7321

U =
    1.0000    4.0000    5.0000
         0    2.0000    6.0000
         0         0    1.7321
```


LU因式分解之應用

- 解線性方程組
- 解三對角線方程組
- 求矩陣的行列式值

應用1：解線性方程組

- 如果一個線性方程組的係數矩陣A，是寫成一個下三角矩陣L和一個上三角矩陣U的乘積，那用兩個步驟就可輕易的解出這個方程組。
- 原來的矩陣-向量方程式Ax=b，可以用A的LU因式分解寫成LUx=b。引入未知向量y，定義為Ux=y。
- 先用「前向代換」由Ly=b解y；也就是先解y_1然後y_2，並依此類推。其次再用「後向代換」由方程組Ux=y解x，先求x_n、再求x_{n-1}、餘依此類推。

例題4.6 解電路問題

線性方程組

$$30i_1 - 20i_2 - 10i_3 = 0$$
$$-20i_1 + 55i_2 - 10i_3 = 80$$
$$-10i_1 - 10i_2 + 50i_3 = 0$$

$$Ax = b$$

$$A = \begin{bmatrix} 30 & -20 & -10 \\ -20 & 55 & -10 \\ -10 & -10 & 50 \end{bmatrix}, b = \begin{bmatrix} 0 \\ 80 \\ 0 \end{bmatrix}$$

例題4.6 解電路問題

A= **LU** 在此

$$\mathbf{L}=\begin{bmatrix}1 & 0 & 0\\ -2/3 & 1 & 0\\ -1/3 & -2/5 & 1\end{bmatrix} \text{ 及 } \mathbf{U}=\begin{bmatrix}30 & -20 & -10\\ 0 & 125/3 & -50/3\\ 0 & 0 & 40\end{bmatrix}$$

我們先解 **Ly** = **b**，即

$$\begin{bmatrix}1 & 0 & 0\\ -2/3 & 1 & 0\\ -1/3 & -2/5 & 1\end{bmatrix}\begin{bmatrix}y_1\\ y_2\\ y_3\end{bmatrix}=\begin{bmatrix}0\\ 80\\ 0\end{bmatrix}$$

經由前向代換得

$$y_1=0,\quad y_2=80 \text{ 及 } y_3=32$$

例題4.6 解電路問題

接著解 **Ux** = **y**，或

$$\begin{bmatrix}30 & -20 & -10\\ 0 & 125/3 & -50/3\\ 0 & 0 & 40\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}=\begin{bmatrix}0\\ 80\\ 32\end{bmatrix}$$

並由後向代換得

$$x_3=4/5,\quad x_2=56/25 \text{ 及 } x_1=44/25$$

三個迴路中的電流值分別爲

$$i_1=1.76,\quad i_2=2.24 \text{ 及 } i_3=0.80$$

例題4.6 MATLAB程式

- 主程式：Main_LU_Solve.m
- LU因式分解程式:LU_Solve.m

```
clear all;
close all;

A=[30 -20 -10;-20 55 -10;-10 -10
50];
b=[0 80 0];

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

x = LU_Solve(L, U, b)
```


例題4.6 MATLAB程式

- 結果：

```
L =
    1.0000         0         0
   -0.6667    1.0000         0
   -0.3333   -0.4000    1.0000

U =
   30.0000  -20.0000  -10.0000
         0   41.6667  -16.6667
         0         0   40.0000

P =
     1     0     0
     0     1     0
     0     0     1

x =
    1.7600
    2.2400
    0.8000
```

用樞軸變換之因式分解

- 如果分解A的時候有用到樞軸變換，則所要求解的方程組不再是Ax=b，而是PAx=Pb，其中PA=LU，P為樞軸變換指標矩陣(置換矩陣)。在此情形下，原來的右側項也要轉換成c=Pb。

例3.5 用樞軸變換之因式分解

$$
\begin{aligned}
2x + 6y + 10z &= 0 \\
x + 3y + 3z &= 2 \\
3x + 14y + 28z &= -8
\end{aligned}
$$

MATLAB程式

```
clear all;
close all;

A=[2 6 10;1 3 3;3 14 28];
b=[0 2 -8];

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

c=P*b'
y=L\c
x=U\y
```


MATLAB程式

結果：

```
L =
    1.0000         0         0
    0.6667    1.0000         0
    0.3333    0.5000    1.0000

U =
    3.0000   14.0000   28.0000
         0   -3.3333   -8.6667
         0         0   -2.0000

P =
     0     0     1
     1     0     0
     0     1     0
```

```
c =
    -8
     0
     2

y =
   -8.0000
    5.3333
    2.0000

x =
    2.0000
    1.0000
   -1.0000
```

應用2：解三對角線方程組

- 一個三對角線方程組，如果其係數矩陣已經過前述的因式分解，則此方程組也可用MATLAB函數求解。

例題4.7 用LU因式分解求解三對角線方程組

爲求解方程組 $\mathbf{A}\mathbf{x} = \mathbf{r}$，其中

$$\mathbf{A} = \begin{bmatrix} 1 & 4 & 0 & 0 & 0 \\ 4 & 17 & 1 & 0 & 0 \\ 0 & 2 & 3 & 2 & 0 \\ 0 & 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 4 & 21 \end{bmatrix} \quad \text{且} \quad \mathbf{r} = \begin{bmatrix} -7 \\ -27 \\ -3 \\ 21 \\ 89 \end{bmatrix}$$

求解三對角線方程組

$$\mathbf{a} = [\ 4 \quad 1 \quad 2 \quad 5 \quad 0\ ]$$
$$\mathbf{d} = [\ 1 \quad 17 \quad 3 \quad 1 \quad 21\ ]$$
$$\mathbf{b} = [\ 0 \quad 4 \quad 2 \quad 0 \quad 4\ ]$$

- 使用LU_Tridiag.m及LU_TridiagSolve.m

$$\mathbf{B} = [\ 0 \quad 4 \quad 2 \quad 0 \quad 4\ ]$$
$$\mathbf{D} = [\ 1 \quad 1 \quad 1 \quad 1 \quad 1\ ]$$
$$\mathbf{x} = [\ 1 \quad -2 \quad 3 \quad -4 \quad 5\ ]$$

求解三對角線方程組

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ B_2 & 1 & 0 & 0 \\ 0 & B_3 & 1 & 0 \\ 0 & 0 & B_4 & 1 \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} D_1 & a_1 & 0 & 0 \\ 0 & D_2 & a_2 & 0 \\ 0 & 0 & D_3 & a_3 \\ 0 & 0 & 0 & D_4 \end{bmatrix}$$

MATLAB程式

- 主程式：Main_LU_TridiagSolve.m
- LU因式分解程式:LU_TridiagSolve.m

```
clear all;
close all;

d=[1 17 3 1 21];
a=[4 1 2 5 0];
b=[0 4 2 0 4];
r=[-7 -27 -3 21 89];
[D, B] = LU_Tridiag(a, d, b)
x = LU_TridiagSolve(a,D, B, r)
```


MATLAB程式

- 結果：

```
D =
      1      1      1      1      1

B =
      0      4      2      0      4

x =
      1     -2      3     -4      5
```

應用3：求矩陣的行列式值 (Determinant)

- 在許多的數學應用中都會用到矩陣的行列式值，包括解特徵值問題及多重積分的變數轉換等。

$$A = LU$$

$$\det(A) = \prod_{i=1}^{n} l_{ii} \prod_{i=1}^{n} u_{ii}$$

若使用樞軸轉換，$A = P^{-1}LU$

$$\det(A) = (-1)^k \prod_{i=1}^{n} l_{ii} \prod_{i=1}^{n} u_{ii}$$

其中k為LU因式分解過程中列對調的次數

例題4.8 求矩陣的行列式值

$$A = LU$$

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

$$L = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -1 & -1 & 1 \end{bmatrix}, U = \begin{bmatrix} 1 & -1 & 2 \\ 0 & -1 & 5 \\ 0 & 0 & 8 \end{bmatrix}$$

無樞軸轉換，則

$$\det(A) = u_{11}u_{22}u_{33} = (1)(-1)(8) = -8$$

4.2 正交矩陣之矩陣轉換

- 正交矩陣：其逆矩陣等於原矩陣的轉置矩陣。
- 轉換方法
  - ➢*Householder轉換*
  - ➢*Givens旋轉*

Householder(豪斯霍爾德)轉換

- **Householder矩陣為**

$$H = I - 2ww^T$$

其中w為$n \times 1$單位行向量(即$\|w\|_2 = w^T w = 1$)

$$H = I - \frac{2}{v'v}vv', \quad w = \frac{v}{\|v\|_2}$$

其中v為非零行向量

- 經由適當的選取v，可以組成H以使得

$$\mathrm{Hx} = \pm\|\mathrm{x}\|e_1$$

- 亦即，除了第一個之外，乘積的所有分量均為零。

Householder轉換

- **Householder矩陣的性質**
  - ➢*it is Hermitian : $H=H^T$*
  - ➢*it is unitary(單位矩陣): $H^{-1}=H^T$*
  - ➢*hence it is involutary(對稱矩陣): $HH=I$*

Givens旋轉

- 第二種轉換叫做Givens旋轉(或轉換)。可以將它描述為乘上一個正交矩陣，此矩陣是一個單位矩陣再加上四個非零元素：

$$g_{ii} = g_{jj} = c \text{ 且 } g_{ij} = -s, g_{ji} = s$$

$$\text{其中 } i \le j \text{ 且 } c^2 + s^2 = 1$$

- 藉由選取適當的c和s的值，Givens旋轉可將矩陣中某一特定元素化為零。

正交矩陣之矩陣轉換

- **Householder矩陣和Givens旋轉矩陣的一個關鍵特性是，它們都是正交的，所以有$HH^T=I$和$GG^T=I$。但是，Householder矩陣同時也是對稱的，所以$HH=I$。**

Householder轉換

滿足$Hx=\pm\|x\|e_1$的Householder矩陣可寫成

$$H=I-\frac{2}{v'v}vv'$$

其中 $v=x+\mathrm{sign}(x_1)\|x\|_2 e_1$

Householder轉換

- 演算法：

$$g = norm(x) = x\_1\hat{}2 + \ldots + x\_n\hat{}2)\hat{}(1/2) \quad \%g = \|x\|$$
$$p = sign(x\_1)$$
$$v = [x\_1 + p * g, x\_2, \ldots, x\_n]$$
$$s = norm(v) = (2 * g * (g + p * x\_1))\hat{}(1/2) \quad \%\text{代數運算之後}$$
$$w = v / s$$

- 要將一個向量由k+1到n的元素都化為零。為此目的，要處理x的k到n分量；經由將向量w初始化為0，前面k-1個分量可被忽略。若s=0則所有需要處理的分量都已經是零了，且向量w應被設為零向量，所以H=I。

Householder轉換

- 為快速求得HA，並不會實際建構出矩陣H。相反的，定義$u=A^Tw$，以使得$u^T=w^TA$，並計算

$$\mathrm{HA} = (\mathrm{I} - 2\mathrm{ww}^{\mathrm{T}})\mathrm{A}$$
$$= \mathrm{A} - 2\mathrm{ww}^{\mathrm{T}}\mathrm{A}$$
$$= \mathrm{A} - 2\mathrm{wu}^{\mathrm{T}}$$

- 計算u需要矩陣與向量相乘，計算wu^T需要求向量外積，而$A-wu^T$則為兩矩陣相減。

Householder轉換

- 同理，要快速求得AH

$$AH = A(I - 2ww^T) = A - 2Aww^T$$

轉換成上三角矩陣形式

- 經由計算B=HA以將一般的矩陣A轉換成上三角矩陣B(並不實際建構出Householder矩陣)。Householder轉換被用於矩陣A的第1行到第n-1行。在第k行中，將位置k+1到n化為零。此函數將x的第一個分量化為零，以取代將向量w初始化為零的做法，並且限制在計算g=norm(x)時使用多少個x的分量。

Householder轉換

- 這些矩陣示意如下，其中空白位置代表零。

x	x	x	x	x		x	x	x	x	x
x	x	x	x	x			x	x	x	x
x	x	x	x	x				x	x	x
x	x	x	x	x					x	x
x	x	x	x	x						x

A　　　　　　　　　　B

例題4.9　轉換為上三角形式

- 要將一個對稱矩陣A轉換成上三角形式，用Householder轉換，逐行將對角線以下元素化為零

$$A=\begin{bmatrix} 1.72 & 1.04 & 0.32 & 0.24 \\ 1.04 & 3.28 & 0.24 & -0.32 \\ 0.32 & 0.24 & 2.92 & -0.56 \\ 0.24 & -0.32 & -0.56 & 2.08 \end{bmatrix}$$

例題4.9　轉換為上三角形式

要在 **A** 的第一行對角線以下加入零：

$$\mathbf{x} = [1.7200 \quad 1.0400 \quad 0.3200 \quad 0.2400]^T$$

$$g = 2.0494$$

$$s = 3.9306$$

$$\mathbf{w} = [0.9590 \quad 0.2646 \quad 0.0814 \quad 0.0611]^T$$

$$\mathbf{u} = [3.9306 \quad 3.7304 \quad 1.1478 \quad 0.4538]^T$$

$$\mathbf{B} = \begin{bmatrix} -2.0494 & -2.5373 & -0.7807 & -0.1952 \\ 0.0000 & 2.2930 & -0.0637 & -0.4401 \\ 0.0000 & -0.0637 & 2.8266 & -0.5969 \\ 0.0000 & -0.5478 & -0.6301 & 2.0523 \end{bmatrix}$$

例題4.9　轉換為上三角形式

要在 **B** 的第二行對角線以下加入零：

$$\mathbf{x} = [-2.5373 \quad 2.2930 \quad -0.0637 \quad -0.5478]^T$$

$$\mathbf{w} = [\ 0.0000 \quad 0.9930 \quad -0.0136 \quad -0.1169]^T$$

$$\mathbf{u} = [\ 0.0000 \quad 4.6839 \quad -0.0560 \quad -1.3378]^T$$

$$\mathbf{B} = \begin{bmatrix} -2.0494 & -2.5373 & -0.7807 & -0.1952 \\ -0.0000 & -2.3584 & -0.0081 & 0.8884 \\ 0.0000 & -0.0000 & 2.8258 & -0.6151 \\ 0.0000 & -0.0000 & -0.6366 & 1.8958 \end{bmatrix}$$

例題4.9 轉換為上三角形式

在 B 的第三行對角線以下加入零：

$$\mathbf{x} = \begin{bmatrix} -0.7807 & -0.0081 & 2.8258 & -0.6366 \end{bmatrix}^T$$

$$\mathbf{w} = \begin{bmatrix} 0.0000 & 0.0000 & 0.9939 & -0.1106 \end{bmatrix}^T$$

$$\mathbf{u} = \begin{bmatrix} 0.0000 & -0.0000 & 5.7577 & -1.6420 \end{bmatrix}^T$$

$$\mathbf{B} = \begin{bmatrix} -2.0494 & -2.5373 & -0.7807 & -0.1952 \\ -0.0000 & -2.3584 & -0.0081 & 0.8884 \\ -0.0000 & -0.0000 & -2.8966 & 1.0168 \\ 0.0000 & -0.0000 & -0.0000 & 1.7143 \end{bmatrix}$$

例題4.9 MATLAB程式

- **主程式：Main_House_to_Triang.m**
- **轉換為上三角形式程式：House_to_Triang.m**

```
clear all;
close all;

A=[1.72 1.04 0.32 0.24
   1.04 3.28 0.24 -0.32
   0.32 0.24 2.92 -0.56
   0.24 -0.32 -0.56 2.08];
B = House_to_triang(A)
```

例題4.9MATLAB程式

- 結果：

```
B =

  -2.0494   -2.5373   -0.7807   -0.1952

  -0.0000   -2.3584   -0.0081    0.8884

  -0.0000    0.0000   -2.8966    1.0168

   0.0000    0.0000   -0.0000    1.7143
```


轉成Hessenberg形式的相似轉換

- 因為Householder矩陣H是正交且對稱的，所以**B=HAH轉換**是相似轉換。B的特徵值和A的一樣。
- 為求得HAH，定義$C=ww^T$及**$d=w^TAw$(純量)**，以使得

$$\begin{aligned}HAH &= (I-2ww^T)A(I-2ww^T)\\ &= (I-2ww^T)A-2(I-2ww^T)Aww^T\\ &= A-2ww^TA-2Aww^T+4ww^TAww^T\\ &= A-2CA-2AC+4dC\end{aligned}$$

$\boxed{4(wAW^T)ww^T}$

轉成Hessenberg形式的相似轉換

- Hessenberg 矩陣的第一下對角線(subdiagonal)以及上三角區域的元素可以不為零。這些矩陣示意如下，其中空白位置代表零。

$$\begin{matrix} x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{matrix} \qquad \begin{matrix} x & x & x & x & x \\ x & x & x & x & x \\ & x & x & x & x \\ & & x & x & x \\ & & & x & x \end{matrix}$$

A　　　　　　　　B

轉成Hessenberg形式的相似轉換

- 使用QR因式分解求矩陣特徵值的問題，必須產生出一序列的矩陣，每個矩陣都需要做因式分解。
- 如果矩陣是屬於Hessenberg形式，則可用更有效率的方式來做 QR因式分解。
- 如果原矩陣是Hessenberg形式，其轉換矩陣都是Hessenberg。

例題 4.10 Hessenberg 形式的相似轉換

- 將矩陣經由Hessenberg相似轉換，使成為Hessenberg矩陣

$$A=\begin{bmatrix} 11 & -26 & 3 & -12 \\ 3 & -12 & 3 & -6 \\ 31 & -99 & 15 & -44 \\ 9 & -10 & -3 & -4 \end{bmatrix}$$

K=1

- Hessenberg 形式B=HAH

例題 4.10 Hessenberg 形式的相似轉換

當$k=1$時

$$x=[0 \quad 3 \quad 31 \quad 9]^T$$

$$g=32.4191$$

$$s=47.9220$$

$$w=[0 \quad 0.7391 \quad 0.6469 \quad 0.1878]^T$$

$$d=-54.2491$$

$$C=\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.5463 & 0.4781 & 0.1388 \\ 0 & 0.4781 & 0.4185 & 0.1215 \\ 0 & 0.1388 & 0.1215 & 0.0353 \end{bmatrix}$$

K=2

$$B=\begin{bmatrix} 11 & 2.8687 & 28.2668 & -4.6645 \\ -32.4191 & -8.0780 & -107.1151 & 16.6447 \\ 0 & 0.0437 & 2.3056 & 0.1142 \\ 0 & 2.8754 & -20.5835 & 4.7724 \end{bmatrix}$$

例題 4.10 Hessenberg 形式的相似轉換

當 k = 2時

$$x = [0 \quad 0 \quad 0.0437 \quad 2.8754\ ]^T$$

$$g = 2.8757$$

$$s = 4.0977$$

$$w = [0 \quad 0 \quad 0.7125 \quad 0.7017\ ]^T$$

$$d = -6.7132$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5076 & 0.4999 \\ 0 & 0 & 0.4999 & 0.4924 \end{bmatrix}$$

$$B = \begin{bmatrix} 11 & 2.8687 & 4.2341 & -28.3345 \\ -32.4191 & -8.0780 & -15.0141 & 107.3558 \\ 0 & 2.8687 & 4.4607 & -20.6163 \\ 0 & 0 & 0.0814 & 2.6174 \end{bmatrix}$$

例題 4.10 MATLAB程式

- 主程式：Main_House_sim_to_Hess.m
- 因式分解程式：House_sim_to_Hess.m

```
clear all;
close all;

A=[11 -26 3  -12
    3 -12 3   -6
   31 -99 15 -44
    9 -10 -3  -4];
B = House_sim_to_Hess(A)
```

例題 4.10 MATLAB程式

- 結果：

```
(k=1)
A =
   11.0000    2.8687   28.2668   -4.6645
  -32.4191   -8.0780 -107.1151   16.6447
    0.0000    0.0437    2.3056    0.1142
    0.0000    2.8754  -20.5835    4.7724

(k=2)
A =
   11.0000    2.8687    4.2341  -28.3345
  -32.4191   -8.0780  -15.0141  107.3558
   -0.0000   -2.8757    4.4607  -20.6163
   -0.0000         0    0.0814    2.6174
```


Hessenberg 轉換的分析

- 依據方程式 $\mathbf{H=I\text{-}2ww^T}$ 所組成的任何矩陣都是對稱的

$$H^T = (I - 2ww^T)^T = I^T - 2(ww^T)^T = I - 2ww^T = H$$

w是一個單位行向量 $(\|w\|^2 = w^T w = 1)$，則H為正交

$$H^T H = (I - 2ww^T)(I - 2ww^T)$$

$$= I - 4ww^T + 4ww^T ww^T = I$$

- 既然同時為對稱且正交，H是它本身的逆矩陣。

Hessenberg 轉換的分析

- **對任何非單位長的向量v≠0，定義** $w = \frac{v}{\|v\|}$**，可將其尺度化為單位向量w。**
- **向量v所決定的Householder矩陣，可以使得x乘上H後，除x的第k個分量之外的所有其它分量都化簡為零。**
- **令e_k代表第k個分量為1其它分量為0的向量。則我們希望找到v以使得，對某一純量α_1有**

$$Hx = \alpha_1 e_k$$

Hessenberg 轉換的分析

- **利用純量和向量適用交換律的關係可得**

$$Hx = x - 2\frac{vv'}{\|v\|^2}x = x - 2\frac{v'x}{\|v\|^2}v = x - \alpha_2 v$$

- **我們看到v必定是x和e_k的線性組合，假設$v=x+\alpha e_k$，則**

$$v'x = (x+\alpha e_k)'x = x'x + \alpha x_k$$

$$v'v = (x+\alpha e_k)'(x+\alpha e_k) = x'x + 2\alpha\alpha_k + \alpha^2$$

Hessenberg 轉換的分析

- 利用以上表示式可得

$$\mathbf{Hx} = \mathbf{x} - 2\frac{\mathbf{v'x}}{\|\mathbf{v}\|^2}\mathbf{v} = \mathbf{x} - \frac{\mathbf{x'x} + \alpha\mathbf{x}_k}{\|\mathbf{v}\|^2}2(\mathbf{x} + \alpha\mathbf{e}_k)$$

$$= (1 - 2\frac{\mathbf{x'x} + \alpha\mathbf{x}_k}{\|\mathbf{v}\|^2})\mathbf{x} - 2\alpha\frac{\mathbf{x'x} + \alpha\mathbf{x}_k}{\|\mathbf{v}\|^2}\mathbf{e}_k$$

$$= (\frac{\mathbf{x'x} + 2\alpha\mathbf{x}_k + \alpha^2 - 2\mathbf{x'x} - 2\alpha\mathbf{x}_k}{\mathbf{x'x} + 2\alpha\mathbf{x}_k + \alpha^2})\mathbf{x} - 2\alpha\frac{\mathbf{x'x} + \alpha\mathbf{x}_k}{\|\boldsymbol{\nu}\|^2}\mathbf{e}_k$$

- 因為x的係數必須為0，又α^2=x’x，所以

$$\mathrm{v} = \mathrm{x} \pm \|\mathrm{x}\|\mathrm{e}_\mathrm{k} \text{且} \mathrm{Hx} \pm \|\mathrm{x}\|\mathrm{e}_\mathrm{k} = 0$$

Hessenberg 轉換的分析

- 將$\alpha=\pm\|x\|$的符號選成和x_k的符號一樣可避免v的第k個分量被抵消。因此，在 Householder 轉換中求 $w_k=(x_k+\mathrm{sign}(x_k)g)/s$。
- 要將Householder轉換用於矩陣A的第一行，求k=1時的向量v，以A的第一行當做向量x。對於第一行之外的其它各行k>1，而將第k行的第k個分量以後的元素化為零。因此將v的第k個分量之前的所有分量設為零;亦即，當i=1,···,k-1時$v_i=0$。在步驟2，我們只利用k到n分量來計算x的範數，記為g。忽略前k-1個分量，並將其餘分量視為是一個(n-k+1)維的向量。
- 在Householder轉換的演算法中，v的範數記為s。向量w是對v做正規化的結果，所以w=v/s。

Householder矩陣有用的性質

- Householder矩陣的乘積滿足$HH^T=I$
- 若$H_1H_1=I$且$H_2H_2=I$，則

$$(H_1H_2)(H_1H_2)^T=(H_1H_2)(H_2^TH_1^T)=I$$

- 對一個對稱矩陣A做$B=H^{-1}AH$形式的相似轉換，會得到一個對稱矩陣B(具有和A一樣的特徵值)。

$$B^T=(H^{-1}AH)^T=H^TA^T(H^{-1})^T$$

$$=HAH=H^{-1}AH=B$$

Householder轉換的用途

- 在4.3節用Householder轉換來求矩陣A的QR因式分解。對一個沒有特別性質的矩陣做單次因式分解，Householder轉換是一個不錯的方法。但是，如果要用QR因式分解求特徵值(於下章介紹)，必須進行一系列的因式分解。在那種情形下，使用相似轉換 (利用Householder轉換)將原矩陣變成Hessenberg矩陣 (特徵值不變)的效率比較好。然後再用Givens旋轉對Hessenberg矩陣做因式分解，。
- Givens旋轉所須的乘法次數大約是Householder轉換的兩倍。但如果要轉換的矩陣中，只有少數幾個元素需要化簡為零，則Givens旋轉是一個不錯的方法。

Givens旋轉

- Givens旋轉是一個如下形式的矩陣

$$G=\begin{array}{c}\begin{bmatrix}1 & 0 & 0 & 0 & 0\\0 & c & 0 & -s & 0\\0 & 0 & 1 & 0 & 0\\0 & s & 0 & c & 0\\0 & 0 & 0 & 0 & 1\end{bmatrix}\end{array}\begin{array}{l}\\ \leftarrow row\ \ i\\ \\ \leftarrow row\ \ j\\ \\ \end{array}$$

其中$c^2+s^2=1$, c出現於列i和j的對角線，其中$i<j$。

Givens旋轉

- 藉由選取適當的c和s的值，Givens旋轉可將矩陣中某一特定元素化為零。當

$$c=\frac{x_1}{\sqrt{x_1^2+x_2^2}}\text{及}s=-\frac{x_2}{\sqrt{x_1^2+x_2^2}}$$

則$c^2+s^2=1$且

$$\begin{bmatrix}c & -s\\ s & c\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}=\begin{bmatrix}\sqrt{x_1^2+x_2^2}\\ 0\end{bmatrix}$$

若G是$Givens$矩陣，則將A乘上G只會影響矩陣A的列i和j；

即若$C=GA$，則

$$C(i,1:n)=cA(i,1:n)-sA(j,1:n)$$

$$C(j,1:n)=sA(i,1:n)+cA(j,1:n)$$

$$C(k,1:n)=A(k,1:n)\text{，其中}k\neq i,j$$

Givens旋轉

- 由右側乘上G^T只會影響行i和j；即若$D=CG^T$，則

$$D(1:n,i) = cC(1:n,i) - sC(1:n,j)$$
$$D(1:n,j) = sC(1:n,i) + cC(1:n,j)$$
$$D(1:n,k) = C(1:n,k)\text{，其中} k \neq i,j$$

- 因此，由左側乘上G(或在右側乘G^T)的動作，不需要完整的構建出G或G^T，也不需要真的執行矩陣相乘。
- Givens旋轉矩陣是正交的，因為$G^TG=I$。這代表乘積$B=GAG^T$是矩陣A的一個相似轉換。

Givens旋轉

- Givens旋轉的主要是將一個上Hessenberg矩陣轉換成上三角矩陣。
- 若Givens矩陣中的參數c和s表示成cos(θ)和sin(θ)，則乘上2x2Givens矩陣

$$G = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \text{相當於在} R^2 \text{中旋轉} \theta \text{角。}$$

例題4.11 Hessenberg矩陣的Givens轉換

- 矩陣

$$B = \begin{bmatrix} 11 & 2.8687 & 4.2341 & -28.3345 \\ -32.4191 & -8.0780 & -15.0141 & 107.3558 \\ 0 & -2.8757 & 4.4607 & -20.6163 \\ 0 & 0 & 0.0814 & 2.6174 \end{bmatrix}$$

例題4.11 MATLAB程式

- 程式：Hessenberg_Givens.m
- 結果：

```
x =   11.0000   -32.4191
c =   0.3213    s =  0.9470
B =34.2345     8.5714   15.5784 -110.7673
         0     0.1210   -0.8147    7.6629
         0    -2.8757    4.4607  -20.6163
         0          0    0.0814    2.6174

x =   0.1210    -2.8757
c =   0.0420    s =  0.9991
B =34.2345     8.5714   15.5784 -110.7673
         0     2.8782   -4.4910   20.9202
         0          0   -0.6264    6.7894
         0          0    0.0814    2.6174

x =   -0.6264    0.0814
c =   -0.9917    s = -0.1289
B =34.2345     8.5714   15.5784 -110.7673
         0     2.8782   -4.4910   20.9202
         0          0    0.6317   -6.3954
         0          0    0.0000   -3.4705
```

QR 因式分解

- 一個實數矩陣A可以分解為A=QR，其中Q為正交，R是上(或右)三角矩陣，且兩者均為實數。如果A是Hessenberg，用Givens轉換做因式分解的效率最好；否則就用Householder轉換做因式分解。
- QR因式分解所需的計算量比LU因式分解多(約為兩倍的運算次數)，但有較優的穩定性。在下一章中會介紹一種利用QR因式分解求矩陣所有特徵值的方法。

用 Householder轉換做QR因式分解

- 有一nxn矩陣A=QR，而在最後階段Q是正交且R是上三角矩陣。此一過程具體表現於以下方程式，其中H_k代表Householder矩陣，不論它乘上那個矩陣，可將該矩陣的第k行在對角線以下元素化為零。
- 此一過程是利用適當的轉換來完成，而不是實際上去構建矩陣並執行矩陣相乘。

用 Householder轉換做QR因式分解

$$
\begin{aligned}
A &= (H_1)(H_1A) \qquad = Q_1R_1 \\
&= (Q_1H_2)(H_2R_1) = Q_2R_2 \\
&= (Q_2H_3)(H_3R_2) = Q_3R_3 \\
&= \qquad \vdots \qquad\qquad \vdots \\
&= (Q_{n-2}H_{n-1})(H_{n-1}R_{n-2}) = QR
\end{aligned}
$$

例題4.12 用 Householder轉換QR做因式分解

$$
A = \begin{bmatrix} 1.72 & 1.04 & 0.32 & 0.24 \\ 1.04 & 3.28 & 0.24 & -0.32 \\ 0.32 & 0.24 & 2.92 & -0.56 \\ 0.24 & -0.32 & -0.56 & 2.08 \end{bmatrix}
$$

MATLAB程式

- 主程式：Main_QR_Factor.m
- 因式分解程式：QR_Factor.m

```
clear all;
close all;

A=[1.72 1.04 0.32 0.24
   1.04 3.28 0.24 -0.32
   0.32 0.24 2.92 -0.56
   0.24 -0.32 -0.56 2.08];
[Q, R] = QR_factor(A);
```


MATLAB程式

- 結果：

```
k = 1
Q =-0.8393    -0.5075    -0.1561    -0.1171
   -0.5075     0.8600    -0.0431    -0.0323
   -0.1561    -0.0431     0.9867    -0.0099
   -0.1171    -0.0323    -0.0099     0.9925
R =-2.0494    -2.5373    -0.7807    -0.1952
    0.0000     2.2930    -0.0637    -0.4401
    0.0000    -0.0637     2.8266    -0.5969
    0.0000    -0.5478    -0.6301     2.0523

k = 2
Q =-0.8393     0.4620    -0.1694    -0.2313
   -0.5075    -0.8448    -0.0197     0.1685
   -0.1561     0.0662     0.9852    -0.0228
   -0.1171     0.2617    -0.0140     0.9579
R =-2.0494    -2.5373    -0.7807    -0.1952
   -0.0000    -2.3584    -0.0081     0.8884
    0.0000     0.0000     2.8258    -0.6151
    0.0000     0.0000    -0.6366     1.8958

k =3
Q =-0.8393     0.4620     0.1144    -0.2629
   -0.5075    -0.8448     0.0563     0.1600
   -0.1561     0.0662    -0.9662     0.1943
   -0.1171     0.2617     0.2242     0.9314
R =-2.0494    -2.5373    -0.7807    -0.1952
   -0.0000    -2.3584    -0.0081     0.8884
   -0.0000     0.0000    -2.8966     1.0168
    0.0000     0.0000    -0.0000     1.7143
```

用Givens旋轉做QR因式分解

- 對於一個(nxn)Hessenberg矩陣A的QR因式分解，用Givens轉換的效率比Householder轉換好。基本觀念相同:若組成一序列的矩陣，稱它們Q和R，以使得A=QR，並且在最後階段Q是正交的，且R是上三角矩陣。

用Givens旋轉做QR因式分解

- G_k代表Givens旋轉矩陣，R為上(右)三角矩陣。

$$\begin{aligned}
A &= (G_1^T)(G_1 A) &&= Q_1 R_1 \\
&= (Q_1 G_2^T)(G_2 R_1) &&= Q_2 R_2 \\
&= (Q_2 G_3^T)(G_3 R_2) &&= Q_3 R_3 \\
&= \vdots \\
&= (Q_{n-2} G_{n-1}^T)(G_{n-1} R_{n-2}) &&= QR
\end{aligned}$$

例題 4.13 Hessenberg 矩陣的 QR 因式分解

- 矩陣

$$B=\begin{bmatrix} 11 & 2.8687 & 4.2341 & -28.3345 \\ -32.4191 & -8.0780 & -15.0141 & 107.3558 \\ 0 & -2.8757 & 4.4607 & -20.6163 \\ 0 & 0 & 0.0814 & 2.6174 \end{bmatrix}$$

MATLAB程式

- 主程式：Main_QR_Factor_Hess.m
- 因式分解程式：QR_Factor_Hess.m

```
A  =  [
 11          2.8687      4.2341    -28.3345
-32.4191    -8.0780    -15.0141    107.3558
  0         -2.8757      4.4607    -20.6163
  0          0           0.0814      2.6174];
[Q, R] = QR_Factor_Hess(A);
```

MATLAB程式

- 結果：

```
x1 =11        x2 =-32.4191
c =0.3213     s =0.9470
R =34.2345     8.5714    15.5784  -110.7673
         0     0.1210    -0.8147     7.6629
         0    -2.8757     4.4607   -20.6163
         0          0     0.0814     2.6174
Q = 0.3213     0.9470          0          0
   -0.9470     0.3213          0          0
         0          0     1.0000          0
         0          0          0     1.0000

x1 =0.1210      x2 =-2.8757
c =0.0420       s =0.9991
R =34.2345     8.5714    15.5784  -110.7673
         0     2.8782    -4.4910    20.9202
         0          0    -0.6264     6.7894
         0          0     0.0814     2.6174
Q = 0.3213     0.0398     0.9461          0
   -0.9470     0.0135     0.3210          0
         0    -0.9991     0.0420          0
         0          0          0     1.0000

x1 =-0.6264      x2 =0.0814
c =-0.9917       s =-0.1289
R =34.2345     8.5714    15.5784  -110.7673
         0     2.8782    -4.4910    20.9202
         0          0     0.6317    -6.3954
         0          0     0.0000    -3.4705
Q = 0.3213     0.0398    -0.9382    -0.1219
   -0.9470     0.0135    -0.3184    -0.0414
         0    -0.9991    -0.0417    -0.0054
         0          0     0.1289    -0.9917
```


進階問題

- 在本章最後，介紹一個使用隱式列樞軸變換的LU因式分解演算法、另一種將矩陣轉換成Hessenberg形式的方法、並彙整一些用於LU和QR因式分解的MATLAB內建函數。

使用隱式列樞軸變換的LU因式分解

- 在類似MATLAB的套裝軟體中，用於LU及QR因式分解的函數，通常會比本章所介紹的更為精細。特別是，它們**不會實際執行部份樞軸變換中的列對調**。它們使用一個記錄所須對調的**指標向量v**。為避免實際執行列對調，我們定義一個由列編號所組成的指標向量。一開始v=[1,2,3]。如果樞軸變換時須要將列1和3對調，我們調換v中相對的元素，得到v=[3,2,1]。任何要對第一列執行的動作，現在都用於列v(1)。如果此一指標向量所代表的所有列對調都作用於一個單位矩陣，就得到置換矩陣P。

使用隱式列樞軸變換的LU因式分解

- 演算法：

```
Input      A                                        要分解之矩陣(n乘n)
Initialize
    v = [ 1   2   3   ... n ]
For k = 1 to n-1
    pivot = | A(v(k),k) |                           樞軸元素
    p = k                                           樞軸列
    For j = k+1 to n
        If (|A(v(j),k)| > pivot)
            pivot = | U(v(j),k) |                   更新樞軸元素
            p = j                                   更新樞軸列
        End
    End
    If (p > k)
            t = v(k)                                更新指標向量
            v(k) = v(j)
            v(j) = t
    End

    For j = k+1 to n
        s = -A(v(j), k)/A(v(k),k)
        A(v(j), k+1:n) = A(v(j), k+1:n) + s*A(v(k), k+1:n)
                          更新A之列v(j)中適當各行
        A(v(j), k) = - s                            儲存變號後的乘數
    End
End
Return      A                    U的元素在上三角部份
                                 L的元素在對角線以下部份
```

例題4.14 使用隱式列樞軸變換的LU因式分解

$$A = \begin{bmatrix} 2 & 6 & 10 \\ 1 & 3 & 3 \\ 3 & 14 & 28 \end{bmatrix}$$

$$v = [1 \quad 2 \quad 3]$$

- 為避免實際執行列對調，在此定義指標向量，初始值為v=[1,2,3]。並不需要在每一階段記錄置換矩陣，在最後可以一次獲得，矩陣A被L和U的元素所覆蓋。

例題4.14 使用隱式列樞軸變換的LU因式分解

第一階段 $(k=1)$ ：樞軸變換讓指標向量成爲 $\mathbf{v} = [3 \quad 2 \quad 1]$。

消去法的第一步 $(j=2$、$v_j = v_2 = 2$、所以更新列 2)：

$s = -1/3$；將 $-s$ 存爲 $A(2,1)$，並更新列 2 的第 2 及 3 行。

$$\mathbf{A} = \begin{bmatrix} 2 & 6 & 10 \\ 1/3 & -5/3 & -19/3 \\ 3 & 14 & 28 \end{bmatrix}$$

消去法的第二步 $(j=3$、$v_j = v_3 = 1$，所以更新列 1) ：

$s = -2/3$，將 $-s$ 存爲 $A(1,1)$，並更新列 1 的第 2 及 3 行。

$$\mathbf{A} = \begin{bmatrix} 2/3 & -10/3 & -26/3 \\ 1/3 & -5/3 & -19/3 \\ 3 & 14 & 28 \end{bmatrix}$$

例題4.14 使用隱式列樞軸變換的LU因式分解

第二階段 $(k=2)$ ：樞軸變換讓指標向量成爲 $\mathbf{v}=[3 \quad 1 \quad 2]$。

消去 $(j=3$、$v_j=v_3=2$，所以更新列 2) ：$s=-1/2$

$$\mathbf{A}=\begin{bmatrix} 2/3 & -10/3 & -26/3 \\ 1/3 & 1/2 & -2 \\ 3 & 14 & 28 \end{bmatrix}$$

爲求得 **L** 和 **U**，我們用向量 **v** 中的資訊以組成 **P** 並計算 **PA**。

$$\mathbf{P}=\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{PA}=\begin{bmatrix} 3 & 14 & 28 \\ 2/3 & -10/3 & -26/3 \\ 1/3 & 1/2 & -2 \end{bmatrix}$$

U 的元素是 **PA** 的對角線及上三角部份;**L** 的元素爲 **PA** 的下三角部份 (**L** 的對角線元素均爲 1)。

$$\begin{bmatrix} 1 & 0 & 0 \\ 2/3 & 1 & 0 \\ 1/3 & 1/2 & 1 \end{bmatrix}\begin{bmatrix} 3 & 14 & 28 \\ 0 & -10/3 & -26/3 \\ 0 & 0 & -2 \end{bmatrix}=\begin{bmatrix} 3 & 14 & 28 \\ 2 & 6 & 10 \\ 1 & 3 & 3 \end{bmatrix}$$

Hessenberg形式的快速轉換

- 利用高斯消去法Householder轉換以將矩陣化為Hessenberg形式的演算法。
- 消去法的計算效率比Householder法好[浮點運算次數約為一半。但對某些矩陣會出現Householder法穩定但消去法不穩定的情形，不過實用上很少碰到這種矩陣。在下列演算法中加入了樞軸變換以增進穩定性。

Hessenberg形式的快速轉換

- 高斯消去法必須加以修改使成為相似轉換，對於為轉換成Hessenberg形式所需的列運算，必須執行相對應的行運算。對一個nxn矩陣，此一程序需要n-2個階段。

Hessenberg形式的快速轉換

輸入
  A　　　　　要分解之矩陣(n 乘 n)
第一階段：
  求行 1 在對角線以下絕對值最大的元素
    若該元素爲零，此階段結束
    否則，將此元素標記爲 a(p,1)
          (亦即此元素位於列 p)
    If p>2,
      列 2 與 p 對調;
      同時對調行 2 與 p.
    For rows i=3, 4,…,n
      求乘數
        m (i,2)=a(i,1)/a(2,1)
      列 i 減去列 2 的乘積
      行 i 的乘積加到行 2

第 k 階段：
  求行 k、列 k+1,…n，中絕對值最大的元素
    若該元素爲零，此階段結束
    否則，將此元素標記爲 a(p,k)
          (亦即此元素位於列 p)
    If p>k+1,
      列 k+1 與 p 對調;
      同時對調行 k+1 與 p.
    For rows i=k+2,…,n
      求乘數
        m (i,k+1) = a (i,k)/a(k+1,k)
      列 i 減去列 k+1 的乘積
      行 i 的乘積加到行 k+1

使用 MATLAB的函數

行列式值與逆矩陣

- 方陣的行列式值 `d=det(X)`

  由 LU 因式分解求行列式值。

- 求方陣的逆矩陣 `Y=inv(X)`

LU 因式分解

對一個方形或矩形矩陣 **A** 的 LU 因式分解，MATLAB 的內建函數可產生 「基本爲三角的」 矩陣 **L** 和 **U**。

- 基本形式 `[L,U]=lu(A)`

  L 是下三角矩陣與置換矩陣的乘積，**U** 是上三角矩陣，可使得 **A**=**LU**。

使用 MATLAB的函數

- 外顯置換形式 `[L,U,P]=lu(A)`

  傳回滿足 **PA**=**LU** 的上三角、下三角和置換矩陣。

- 壓縮形式 `Y=lu(A)`

  傳回矩陣 **Y**，它包含單純的下三角矩陣 **L** 和上三角矩陣 **U** 做爲子矩陣。

- 方陣的快速因式分解 `[L,U,P,Q]=lu(X)`

  傳回一個單位下三角矩陣 L、上三角矩陣 U、列置換矩陣 P 及行重排序矩陣 Q，可滿足 P∗X∗Q=P∗X∗Q。如果 X 是空矩陣或不是稀疏矩陣，`lu` 會顯示錯誤訊息。

- 控制的樞軸變換形式 `[L,U,P]=lu(X,t)`

  控制稀疏矩陣的樞軸變換；取 $0 \le t \le 1$，當某一行之對角線元素的絕對值，小於同行其它元素的絕對值乘 t 時，進行樞軸變換。取 $t=0$ 會強制對角線樞軸變換。$t=1$ (內定值) 則爲慣用的部份樞軸變換。

- Cholesky 因式分解 `R=chol(A)`

  假定 **A** 爲對稱；只會使用 **A** 的對角線與上三角部份的元素。下三角部份則被設爲是上三角部份的 (共軛複數) 轉置。如果 **A** 爲正定，則 `R=chol(A)` 會產生上三角矩陣 **R** 滿足 $\mathbf{R}^T\mathbf{R}=\mathbf{A}$。如果 **A** 不爲正定，則會印出錯誤訊息。

使用 MATLAB的函數

QR 因式分解

MATLAB 的內建函數可以對 m 乘 n 矩陣 **A** 做 QR 因式分解，以產生上三角矩陣 **R** 和 unitary matrix **Q**，以使得 **A** = **QR**。

- 基本形式：`[Q,R]=qr(A)`

  Q 爲 m 乘 m 且 **R** 爲 m 乘 n。對於稀疏矩陣，**Q** 經常是將近全滿。

- 經濟大小`[Q,R]=qr(A,0)`

  如果 $[mn] = size(\mathbf{A})$ 且 $m > n$，它只會求 **Q** 的前 n 行，而 **R** 爲 n 乘 n。否則的話，它和基本形式一樣。

- 含置換的分解`[Q,R,E]=qr(A)`

  加入置換矩陣 **E** 使得 **AE** = **QR** 且 abs (diag(**R**)) 是漸減。`[Q,R,E]=qr(A,0)`會產生一個含置換的經濟大小的因式分解。

- 稀疏矩陣 **A** 的 Q-less 分解 **A** `R=qr(A)`

  只傳回 **R**。在此 `R=chol(A'A)`。