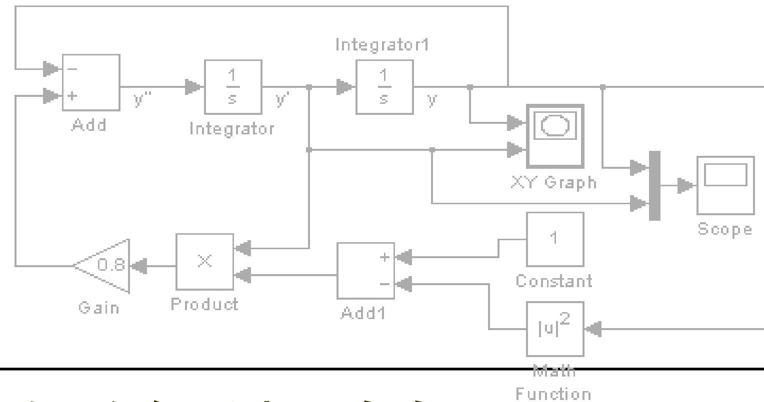
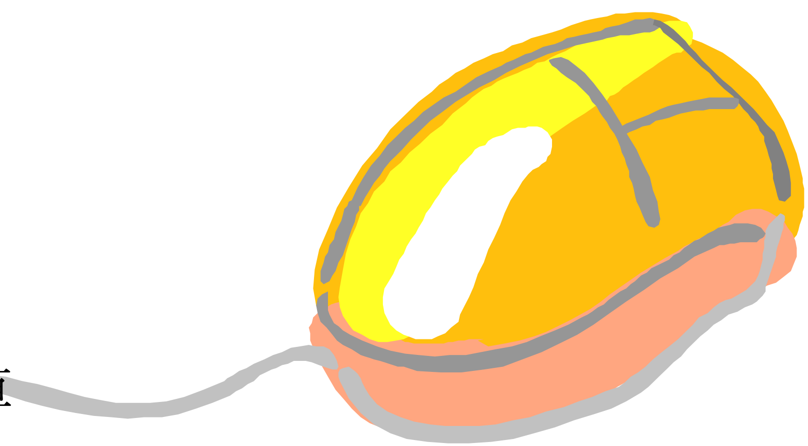


第三章

向量與矩陣的運算

本章學習目標

- 認識陣列裡元素的結構
- 學習多維陣列的建立
- 學習編修矩陣的內容
- 學習基本的矩陣數學運算



3.1 陣列元素的處理

3.1.1 向量元素的操作

- Matlab的陣列索引值是從1開始
- C語言的陣列索引值是從0開始

- 下面是向量元素操作的範例：

```
>> v1=[6 7 8 9];

>> 2*v1+1
ans =
    13    15    17    19

>> v1(2)
ans =
     7
```

3.1.2 矩陣元素的操作

- 矩陣必須有列與行兩個索引值才能取得陣列裡的特定元素：

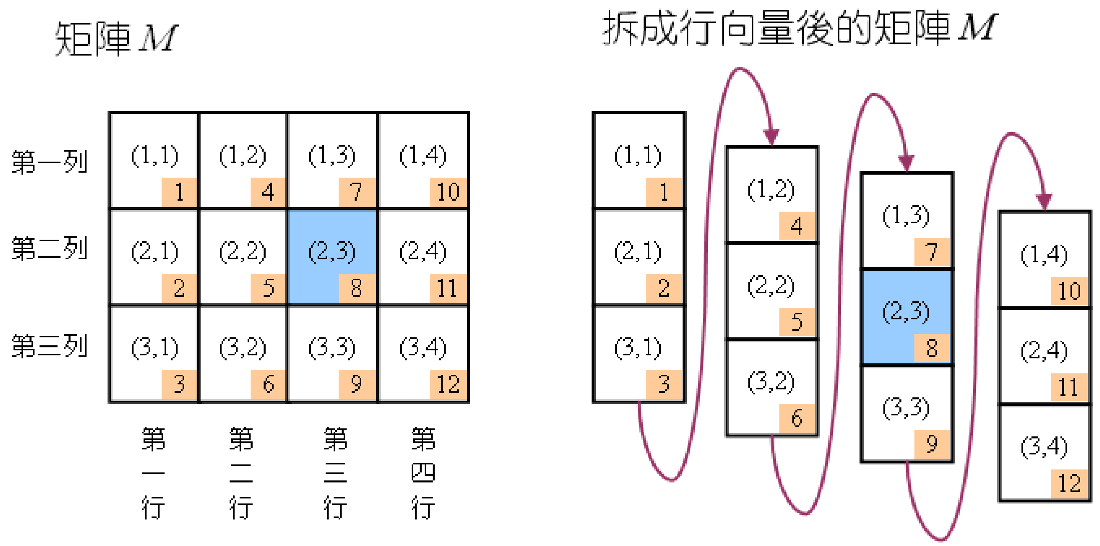
```
>> M=[1 2 3 4;5 6 7 8;9 10 11 12]
M =
     1     2     3     4
     5     6     7     8
     9    10    11    12

>> M(2,3)
ans =
     7

>> M(3,[1 2 3])
ans =
     9    10   11
```

3.1.3 矩陣的索引值之結構

- Matlab的矩陣是利用「以行為主」的結構來儲存，如下圖所示



對於一個 $m \times n$ 的矩陣 M 而言，二維索引值 (i, j) 和一維索引值 k 之間的關係可用下面的公式來表示：

$$M(k) = M(i + (j - 1) \times m)$$

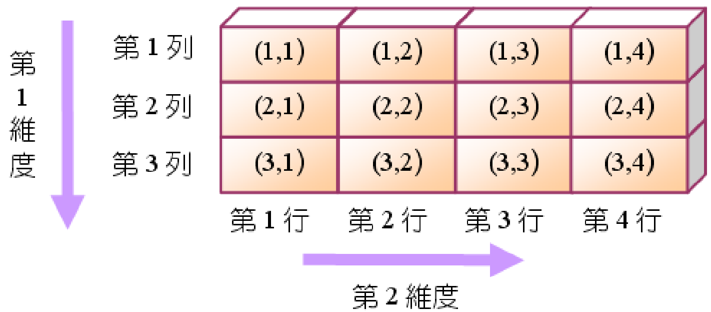
例如，若陣列 M 的維度為 3×4 ，則第二列第三行的元素 M(2,3)，換成以一維的索引值來表示時，可換算得

$$M(k) = M(2 + (3 - 1) \times 3) = M(8)$$

3.2 多維陣列

- 只要陣列的維度多於二維，我們就稱之為多維陣列
- 對於三維陣來說，需要列、行與頁（page）三個維度來描述

二維陣列



這是 3×4 的矩陣。矩陣裡每一個元素的位置可用（列,行）來表示

三維陣列

這是 3×4×2（3 列 4 行 2 頁）的三維陣列。陣列裡每一個元素的位置可用（列,行,頁）來表示

- 要建立一個三維陣列，可針對每一頁分別建立二維陣列：

```
>> A(:,:,1)=[1 2 3 4;5 6 7 8;9 10 11 12]

A =
     1     2     3     4
     5     6     7     8
     9    10    11    12

>> A(:,:,2)=[7 4 2 1;6 1 5 2;3 1 4 5]

A(:,:,1) =
     1     2     3     4
     5     6     7     8
     9    10    11    12

A(:,:,2) =
     7     4     2     1
     6     1     5     2
     3     1     4     5
```

3.3 常用的陣列建立函數

- Matlab常用的陣列建立函數如下表所列：

表 3.3.1　常用的矩陣建立函數

函 數	說 明
`zeros(`n`)`	建立一個 $n \times n$ 的全零矩陣
`zeros(`m`,`n`,...,`p`)`	建立一個 $m \times n \times \cdots \times p$ 的全零矩陣
`ones(`n`)`	建立一個 $n \times n$ 的全 1 矩陣
`ones(`m`,`n`,...,`p`)`	建立一個 $m \times n \times \cdots \times p$ 的全 1 矩陣
`eye(`n`)`	建立一個 $n \times n$ 的單位矩陣（
`eye(`m`,`n`)`	建立一個 $m \times n$ ，且對角線為 1，其它元素為 0 的矩陣
`diag(`v`)`	以向量 v 為對角元素，建立一個矩陣
`magic(`n`)`	建立一個 $n \times n$ 的魔術方陣（magic square）

- 亂數陣列

  Matlab也提供了產生均勻分佈，或者是常態分佈的亂數陣列函數：

表 3.3.2　以亂數來建立陣列之函數

函 數	說 明
`rand`	建立一個 0~1 之間均勻分佈的亂數
`rand(`n`)`	建立一個 0~1 之間，維度為 $n \times n$ 之均勻分佈的亂數矩陣
`rand(`m`,`n`,...,`p`)`	建立一個 0~1 之間，維度為 $m \times n \times \cdots \times p$ 之均勻分佈的亂數矩陣
`randn`	建立一個平均值為 0，標準差為 1 的常態分佈亂數
`randn(`n`)`	同 rand，但是建立常態分佈的亂數矩陣
`randn(`m`,`n`,...,`p`)`	同 rand，但是建立常態分佈的亂數矩陣

3.4 陣列元素的其它操作

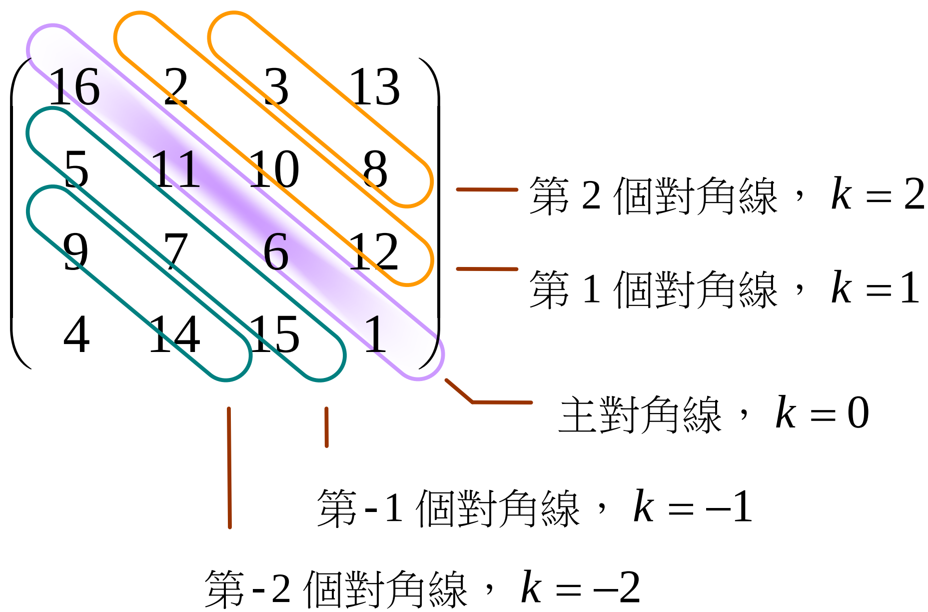
3.4.1 陣列元素的提取

- 要提取陣列的對角線元素，或是上三角形或下三角形矩陣，可利用如下表的函數：

表 3.4.1　陣列元素的提取函數

函 數	說 明
`diag(A)`	取出陣列 A 的主對角線（main diagonal）元素
`diag(A,k)`	取出陣列 A 的第 k 個對角線元素
`triu(A)`	取出陣列 A 之主對角線以上之元素，其它元素則設為 0
`triu(A,k)`	取出陣列 A 之第 k 個對角線以上之元素，其它元素則設為 0
`tril(A)`	取出陣列 A 之主對角線以下之元素，其它元素則設為 0
`tril(A,k)`	取出陣列 A 之第 k 個對角線以下之元素，其它元素則設為 0

- 主對角線與第 k 個對角線的示意圖：



3.4.2 陣列元素的重排

- 陣列轉換函數，可將陣列拆解成另一種形式：

表 3.4.2　陣列轉換函數

函數	說明
`fliplr(`A`)`	將陣列 A 的元素左右翻轉（flip left/right）
`flipud(`A`)`	將陣列 A 的元素上下翻轉（flip up/down）
`flipdim(`A,n`)`	將陣列 A 的元素依第 n 個維度翻轉
`reshape(`$A,m,n,\ldots,p$`)`	將陣列 A 的元素依由上到下，由左到右的次序重新排列成一個 $m\times n\times\cdots\times p$ 的矩陣
`repmat(`$A,m,n,\ldots,p$`)`	以陣列 A 為單位，將陣列 A 以類似排列磁磚的方式排成 $m\times n\times\cdots\times p$ 個陣列 A
`rot90(`A`)`	將陣列 A 逆時針旋轉 90°
`rot90(`A,k`)`	將陣列 A 逆時針旋轉 $k\times 90^\circ$，k 為整數

3.4.3 陣列的合併

- 合併陣列的指令如下表所示：

表 3.4.3　陣列轉換函數

函 數	說 明
`[`A`,`B`]`	將陣列 A, B 橫向併排，組合成一個新的陣列
`[`A`;`B`]`	將陣列 A, B 垂直併排，組合成一個新的陣列
`cat(`dim`,`A`,`B`,`...`)`	依 dim 所指定的方向合併陣列 A, B,...

3.5 矩陣的數學運算

3.5.1 基本的矩陣運算函數

- 下表列出了矩陣的基本運算函數：

矩陣的運算	說 明
A+B	矩陣 A 加上矩陣 B
A-B	矩陣 A 減去矩陣 B
A*B	矩陣 A 乘上矩陣 B
A^n	矩陣 A 的 n 次方，即矩陣 A 連乘 n 次，A 必須為方陣
A'	計算矩陣 A 的共軛轉置（conjugate transpose）
inv(A)	計算矩陣 A 的反矩陣（inverse）
det(A)	計算矩陣 A 的行列式（determinate）
expm(A)	計算矩陣 A 的指數（matrix exponential）
logm(A)	計算矩陣 A 的對數（matrix logarithm）
sqrtm(A)	計算矩陣 A 的開平方根

3.5.2 矩陣的左除與右除

- 左除「\」與右除「/」運算子，可分別用$AX=B$與$XA=B$來計算：

表 3.5.2　矩陣的數學運算

指令	說 明
$A\backslash B$	A 左除 B，此運算相當於把 A 的反矩陣乘以 B，也就是 $A^{-1}B$
B/A	B 右除 A，此運算相當於把 B 乘上 A 的反矩陣，也就是 BA^{-1}

例如，設

$$AX = B,\quad A = \begin{pmatrix} 3 & 4 \\ 1 & 2 \end{pmatrix},\quad B = \begin{pmatrix} 10 \\ 4 \end{pmatrix}$$

若要求解向量 X，則

$$X = A^{-1}B = \begin{pmatrix} 1 & -2 \\ -0.5 & 1.5 \end{pmatrix}\begin{pmatrix} 10 \\ 4 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

上式可以寫上 X=inv(A)*B，或用 A 來左除 B，即

```
>> A\B
```

右除是用在另外的一種情況，例如，設

$$XA = B \qquad A = \begin{pmatrix} 3 & 4 \\ 1 & 2 \end{pmatrix} \qquad B = \begin{pmatrix} 10 & 14 \end{pmatrix}$$

則

$$X = BA^{-1} = \begin{pmatrix} 10 & 14 \end{pmatrix} \begin{pmatrix} 1 & -2 \\ -0.5 & 1.5 \end{pmatrix} = \begin{pmatrix} 3 & 1 \end{pmatrix}$$

上式可以寫上 X=B*inv(A)，或用 B 右除 A ，即

```
>> B/A
```

3.5.3 陣列元素對元素的運算

在 Matlab 裡，矩陣 A 乘上 B 可以寫成 $A*B$。例如，設

$$A=\begin{pmatrix}2 & 4\\3 & 1\end{pmatrix},\quad B=\begin{pmatrix}3 & 2\\4 & 6\end{pmatrix};$$

則

$$A*B=\begin{pmatrix}2 & 4\\3 & 1\end{pmatrix}\begin{pmatrix}3 & 2\\4 & 6\end{pmatrix}=\begin{pmatrix}2\times3+4\times4 & 2\times2+4\times6\\3\times3+1\times4 & 3\times2+1\times6\end{pmatrix}=\begin{pmatrix}22 & 28\\13 & 12\end{pmatrix}$$

如果是希望陣列 A 內的元素乘上陣列 B 內相對應的位置，也就是計算

$$A.*B=\begin{pmatrix}2 & 4\\3 & 1\end{pmatrix}\begin{pmatrix}3 & 2\\4 & 6\end{pmatrix}=\begin{pmatrix}2\times3 & 4\times2\\3\times4 & 1\times6\end{pmatrix}=\begin{pmatrix}6 & 8\\12 & 6\end{pmatrix}$$

則可以利用 Matlab 所提供的 "元素對元素"的計算指令

- 元素對元素的計算符號是在矩陣的計算符號之前加上一個「.」：

表 3.5.3　陣列的數學運算

指 令	說 明
A.*B	將矩陣 *A* 內的元素乘上矩陣 *B* 內相同位置的元素
A.^n	計算矩陣 *A* 的 *n* 次方，即矩陣 *A* 連乘 *n* 次，*A* 必須為方陣
A.'	計算矩陣 *A* 的轉置（transpose）矩陣。
A./B	將 *A* 裡面的每一個元素除以 *B* 裡面每一個相對應的元素
A.\B	將 *B* 裡面的每一個元素除以 *A* 裡面每一個相對應的元素

```
>> A=[2 4;3 1]
A =
     2     4
     3     1

>> B=[3 2;4 6]
B =
     3     2
     4     6

>> A*B
ans =
    22    28
    13    12

>> A.*B
ans =
     6     8
    12     6
```

-The End-