第二章　矩陣的輸入與輸出

矩陣是二維方形陣列，陣列也是資料結構之一種，矩陣資料結構在 C#、Python ...等等一般目的(General Purpose)程式語言或是在矩陣計算為主要目的 Matlab 程式語言中，矩陣的表示方式並不相同，故矩陣輸入的型態也不同。另矩陣計算結果，Python 提供函式 print()，Matlab 提供 disp()函數，C#提供 Console.Write()方法，SMS-2029 類別庫提供 Print 類別，能在命令列提示環境中，作為矩陣的顯示輸出。

2.1　變數的名稱

C#程式語言的變數，用以指定(Assignment)為數學式或是資料結構，而矩陣、向量的運算，包含加、減、乘、分解、逆算等等，變數除了符合 C#程式語言之規定外，也可作爲線性代數中矩陣的運算式，將兩者整合在一起。

變數的第一個英文字大寫表示矩陣，小寫表示向量，變數最後一個英文字的保留字為 i、t、h、n。即 i 代表逆矩陣(Inverse)，t 代表轉置(Transpose)，h 代表轉置並共軛複數(Hermitian 或 self-adjoint)，n 代表單位向量(Unit Vector)。在線性代數中常常以 AB 代表 A 矩陣乘 B 矩陣，但 SMS-2029 類別庫中，AB 代表一個變數，變數 A 矩陣乘以變數 B 矩陣，應寫成 A*B，就像 A+B 表示變數 A 矩陣加變數 B 矩陣，在 SMS-2029 類別庫也提供矩陣運算和分解的各種不同的類別(class)和矩陣的運算子。譬如 C = A/B 代表 A 矩陣除以 B 矩陣，並指定(Assign)給矩陣變數 C，"/"符號是矩陣除以的運算子，當執行時若有錯誤或無法執行，即產生錯誤(Exception)。另在線性代數 Ay = b，如果寫成程式碼時，則寫成 A * y = b，即 y = Ai * b 的方式表示，向量變數 y = A 的逆矩陣乘以 b。程式碼以 ReMatrix y = ~A * b;上式中~A 表示已知矩陣 A 的逆矩陣，"~"是矩陣的運算子，Ai 變數則表示矩陣 A 的逆矩陣。At 矩陣變數代表矩陣 A 的轉置(Transpose)，un 向量變數代表 u 向量的單位向量(Unit Vector)。

2.2 變數的型態

C#是靜態物件導向程式語言，第一次宣告變數時須要標示型別，Python 是動態程式語言，變數無須標示資料型別，由輸入的數據來判斷資料的型別 ，譬如 C#標示為 double 型別的變數，即使輸入整數的資料，仍然能隱含(Implicit)轉換為 double 型別的實數資料。

數值型陣列資料結構，零維度表示純量(Scalar)，純量是整數、實數、複數。一維度稱作列表(List)，以 douable[]表示實數型態，二維度稱作矩陣，C#程式語言中有方形(長方形或是正方形)，以 double[,]表示，另一種是鋸齒形，以 double[][]表示，SMS-2029 矩陣計算類別庫以方形表示，實數矩陣以 ReMatrix 類別(Class)表示，複數矩陣以 CxMatrix 類別表示，此外尚有三維度矩陣，以 double[,,]表示，三維度可能僅用於資料的儲存，並不適合矩陣的計算，矩陣計算僅使用二維度方形陣列。

在線性代數或是矩陣計算中，對於維度較模糊，譬如陣列[5,7,8]的轉置後，是一維或是二維？但 SMS-2029 類別庫都是二維度，譬如上示陣列的表示方式為 double[,] A = {{5,7,8}}; 或是 double[,] A = new double[1,3] {{5,7,8}}; 在 SMS-2029 類別庫中僅 Arange 和 LinSpace 兩個類別為一維度的陣列，但可隱性(Implicit)轉換為二維方形陣列。Arange 和 LinSpace 兩個類別，分別參考 Python 程式語言 numpy 模組，兩個函式 arange()和 linspace()而建構的。

2.3 數值資料的輸入

何種資料結構的數據，才能在 C#主控台的環境中執行，零維度是純量，如 double 型態的實數值，int 型態的整數值，或是 Complex 型態的複數值，以 double[]的型態表示一個維度的陣列，陣列的儲存內容是實數值，以 double[,]的型態表示二個維度的陣列，陣列的儲存內容是實數值，SMS-2029 類別程式庫是 C#程式語言撰寫，將複數矩陣資料結構轉換為複數矩陣類別 CxMatrix，實數矩陣資料結構轉換為實數矩陣類別 ReMatrix，另 System.Numerics 名稱空間的 Complex 複數純量轉換為複數純量類別 CxScalar，轉換為矩陣類別庫物件的目

的，可以使用矩陣計算的運算子，也可以引用類別的屬性(Property)、方法(Method)、或是索引子(Indexer)，提供矩陣元素的擷取運算等等。

CxScalar 與 Complex 是純量複數，CxMatrix、ReMatrix、和 double[,]等是不同型態的矩陣物件，它們之間可以互相隱性或是明確轉換型態(Implicit/Explicit Convertion)。在 SMS-2029 類別庫中，除了 Arange 和 LinSpace 兩個類別為一維陣列，這兩個類別是參考 Python Numpy 模組的 arange()和 linspace()函式建構，其餘矩陣類別均是二維型態矩陣類別。

舉列來說一個數值為 5，則可寫成 double a = 5;，寫成一維陣列 double[] b = {5};，寫成二維陣列 double[,] A = {{5}};，但變數 a，b，和 A 完全不同的型態。一般線性代數不會去探討變數 a、b、和 C 的型態上的差異，但靜態物件導向程式語言非常重要。

2.4 數值資料的輸出

在 System 名稱空間 Console 類別 Write()或 WriteLine()方法，提供主控台螢幕輸出顯示。譬如實數純量變數 a 等於 25，其螢幕主控台的輸出如下:

```
double a = 25;
Console.Write("\n 變數 a = {0}\n", a);
Console.Write($"\n 變數 a = {a}\n");
```

其他複數、實數和複數矩陣螢幕主控台的輸出，使用名稱空間 Matrix_0(即 Matrix_0.dll 檔案中)的 PR 類別。譬如負數純量變數 b 等於 5+3i 的輸出如下:

```
CxScalar b = new CxScalar(5, 3);
Console.Write("\n 變數 b = {0}\n", new PR(b));
Console.Write($"\n 變數 b = {new PR(b)}\n");
```

實數矩陣類別 ReMatrix，譬如實數矩陣之第 0 列為(8,9)，第 1 列為(-5,7)，其輸入及輸出如下:

```
double[,] Areal = { {8,9}, {-5,7} };
ReMatrix A = new ReMatrix(Areal);
```

Console.Write("\n 矩陣 A :\n{0}\n", new PR(A));

複數矩陣類別 CxMatrix，譬如複數矩陣之第 0 列為（5+3i，7+3i），第一列為(-3+5i，2-3i)，其螢幕主控台輸入和輸出如下:

double[,] Breal = { {5,7}, {-3,2} };

double[,] Bimage = { {3,3}, {5,-3} };

CxMatrix B = new CxMatrix(Breal, Bimage);

Console.Write("\n 矩陣 B :\n{0}\n", new PR(B));

Console.Write($"\n 矩陣 B :\n{new PR(B)\n");

相關之程式碼如下 :

```
using System;
using Matrix_0;

namespace ConsoleApp25
{
    internal class Program
    {
        static void Main(string[] args)
        {

double a = 25;
Console.Write("\n 變數 a = {0}\n", a);
Console.Write($"\n 變數 a = {a}\n");
//
CxScalar b = new CxScalar(5, 3);
Console.Write("\n 變數 b :\n{0}\n", new PR(b));
Console.Write($"\n 變數 b :\n{new PR(b)}");
//
double[,] Areal = { { 8, 9 }, { -5, 7 } };
ReMatrix A = new ReMatrix(Areal);
Console.Write("\n矩陣 A :\n{0}\n", new PR(A));
Console.Write($"\n矩陣 A :\n{new PR(A)}");
//
double[,] Breal = { {5, 7}, {-3, 2} };
double[,] Bimage = { {3, 3}, {5, -3} };
```

```
CxMatrix B = new CxMatrix(Breal, Bimage);
Console.Write("\n矩陣 B :\n{0}\n", new PR(B));
Console.Write($"矩陣 B :\n{new PR(B)}");

        }
    }
}

/*輸出結果如下 :
 變數 a = 25
 變數 a = 25
 變數 b :
        5.00000  +    3.00000i,
 變數 b :
        5.00000  +    3.00000i,
矩陣 A :
        8.00000     9.00000
       -5.00000     7.00000
矩陣 A :
        8.00000     9.00000
       -5.00000     7.00000
矩陣 B :
        5.00000  +   3.00000i,    7.00000  +    3.00000i
       -3.00000  +   5.00000i,    2.00000  -    3.00000i
矩陣 B :
        5.00000  +   3.00000i,    7.00000  +    3.00000i
       -3.00000  +   5.00000i,    2.00000  -    3.00000i
*/
```

2.5 矩陣與座標軸的關係

二維方形(長方形或是正方形)陣列稱作矩陣，在 C#程式語言中以 double[,] A 表示，double[,]表示型態，A 是變數，另一種鋸齒形的二維陣列 double[][] B 表示，double[][]表示型態，B 是變數，在 SMS-2029 類別庫中，除了 Arange 和 LinSpace 類別為一維陣列外，其餘的類別均爲二維方形陣列，稱作矩陣。

m 列乘 n 行的矩陣，我們可以考慮為垂直維度上有 m 個互爲垂直的座標軸，

每一個座標軸的水平維度上有 n 個位置，在某一座標軸某一個位置上，代表電腦記憶體儲存的數值，廣義的數值是複數，複數的範圍大於實屬，實屬的範圍大於整數。原則上將複數矩陣轉換為 CxMatrix 類別，double[,]實數型態的矩陣轉換為 ReMatrix 類別，SMS-2029 類別庫上提供各種不同的運算子，使矩陣的計算就像純量的數值計算一樣方便。可將 mxn 的矩陣類別，考慮為儲存數值的資料單位，也就是 m 個互相垂直的座標軸上，在每個座標軸上有 n 個位置，在位置上存放的數值資料。

複數是人們最感到困惑的數值，Python 程式語言以 j，Matlab 程式語言以 i，代表虛數單位 √-1 輸入電腦中，C#程式語言沒有這種方式輸入電腦，作者也認爲以一個英文字母代表虛數單位，這種方式輸入是一種奇特的事情。但 C#環境中以 System.Numerics 名稱空間 Complex 類別代表純量複數。早期作者甚至不知道在 C#環境中，如何輸入複數矩陣？直到後來求得複數的特徵值和特徵向量，並且更進一步計算求得實數可量測的空間狀態響應值，包含空間不同位置點的變位、速度、和加速度，故作者的結論是，複數是暫時的計算過程中的數值，最後的結果是真實可量測的實數世界，在計算過程中，使用複數是不可避免的，除非使用如 Runge-Kutta 等近似求解法，或是 Matlab 程式語言中的 ode45、ode23 等等常微分方程式的近似求解法。另外作者從相關的文件中，並沒有發現有人直接使用上述方法，求解狀態空間響應，這是新穎的求解法。