Ch.04 矩陣的建立與運算

矩陣是一組型態相同的純量排列成的二維資料結構。

IDL建立矩陣內容為零的函數(n,m為矩陣的維度)

函數(n為整數)	功能
BYTARR(n,m)	建立內容為零的短整數矩陣
FIXARR(n,m)	建立內容為零的整數矩陣
UINTARR(n,m)	建立內容為零的無號整數矩陣
LONGARR(n,m)	建立內容為零的長整數矩陣
ULONGARR(n,m)	建立內容為零的無號長整數矩陣
LONG64ARR(n,m)	建立內容為零的64位元長整數矩陣
ULONG64ARR(n,m)	建立內容為零的64位元無號長整數矩陣
FLTARR(n,m)	建立內容為零的浮點數矩陣
DBLARR(n,m)	建立內容為零的雙精度浮點數矩陣
COMPLEXARR(n,m)	建立內容為零的複數矩陣
DCOMPLEXARR(n,m)	建立內容為零的雙精度複數矩陣
STRARR(n,m)	建立內容為零的字元矩陣

```
x=bytarr(5,2)
help,x & print,x
x1=Lonarr(4,3)
help,x1 & print,x1
x2=fltarr(3,4)
help,x2 & print,x2
x3=dblarr(2,3)
help,x3 & print,x3
end
```

```
% Compiled module: $MAIN$.
X               BYTE      = Array[5, 2]
  0   0   0   0   0
  0   0   0   0   0
X1              LONG      = Array[4, 3]
        0       0       0       0
        0       0       0       0
        0       0       0       0
X2              FLOAT     = Array[3, 4]
   0.000000    0.000000    0.000000
   0.000000    0.000000    0.000000
   0.000000    0.000000    0.000000
   0.000000    0.000000    0.000000
X3              DOUBLE    = Array[2, 3]
    0.00000000    0.00000000
    0.00000000    0.00000000
    0.00000000    0.00000000
IDL>
```

建立矩陣內容均為特定值的函數(n為矩陣長度)

函數	功能
REPLICATE(value, n,m)	建立內容為value的nxm矩陣，矩陣長度為n，value為任意資料型態
MAKE_ARRAY(n,m, VALUE=value)	建立內容為value的nxm矩陣，矩陣長度為n，value為任意資料型態

```
x=replicate(1B,5,2)
help,x & print,x
x1=replicate(1.2,3,2)
help,x1 & print,x1
x2=replicate('abc',2,3)
help,x2 & print,x2
x3=make_array(4,3,VALUE=2D)
help,x3 & print,x3
x4=replicate(1.3,5)
help,x4 & print,x4
x5=replicate(3.1D,5,1)
help,x5 & print,x5
end
```

```
% Compiled module: $MAIN$.
X               BYTE      = Array[5, 2]
  1   1   1   1   1
  1   1   1   1   1
X1              FLOAT     = Array[3, 2]
    1.20000     1.20000     1.20000
    1.20000     1.20000     1.20000
X2              STRING    = Array[2, 3]
abc abc
abc abc
abc abc
X3              DOUBLE    = Array[4, 3]
     2.0000000      2.0000000      2.0000000
2.0000000
     2.0000000      2.0000000      2.0000000
2.0000000
     2.0000000      2.0000000      2.0000000
2.0000000
X4              FLOAT     = Array[5]
    1.30000     1.30000     1.30000
1.30000     1.30000
X5              DOUBLE    = Array[5]
     3.1000000      3.1000000      3.1000000
3.1000000      3.1000000
IDL>
```

產生亂數矩陣的函數

函數	功能
Result = RANDOMU(seed, n,m)	產生亂數矩陣Result，包含nxm個均勻分布的元素。Seed是種子參數，可以不給定特定值，元素數值落在0和1間。
Result = RANDOMN(seed, n,m)	產生亂數矩陣Result，包含nxm個均勻分布的元素。Seed是種子參數，可以不給定特定值，元素數值呈現常態分布的浮點數。

```
x=randomu(seed,3,4)
help,x & print,x
y=randomn(seed,4,3)
help,y & print,y
y1=randomn(1L,2,3)
help,y1 & print,y1
y2=randomn(1L,2,2)
help,y2 & print,y2
end
```

```
% Compiled module: $MAIN$.
X               FLOAT     = Array[3, 4]
    0.289934     0.435734     0.742830
    0.528935     0.746580     0.970582
    0.744869     0.845004     0.358634
    0.488555     0.367527     0.163493
Y               FLOAT     = Array[4, 3]
   -0.398051      1.06713     0.249249    -
0.349727
    -1.09287     0.663751    -0.894983
1.51833
   -0.135784     0.424395     -1.80682
0.986354
Y1              FLOAT     = Array[2, 3]
    0.306400     0.156066
   -0.424386    -0.568040
   -0.204547    -0.806289
Y2              FLOAT     = Array[2, 2]
    0.306400     0.156066
   -0.424386    -0.568040
IDL>
```

建立矩陣內容為下標的函數(nxm為矩陣的維度)

函數	功能
BINDGEN(n,m)	建立內容為下標的短整數矩陣
INDGEN(n,m)	建立內容為下標的整數矩陣
UINDGEN(n,m)	建立內容為下標的無號整數矩陣
LINDGEN(n,m)	建立內容為下標的長整數矩陣
ULINDGEN(n,m)	建立內容為下標的無號長整數矩陣
L64INDGEN(n,m)	建立內容為下標的64位元長整數矩陣
UL64INDGEN(n,m)	建立內容為下標的64位元無號長整數矩陣
FINDGEN(n,m)	建立內容為下標的浮點數矩陣
DINDGEN(n,m)	建立內容為下標的雙精度浮點數矩陣
CINDGEN(n,m)	建立內容為下標的複數矩陣
DCINDGEN(n,m)	建立內容為下標的雙精度複數矩陣
SINDGEN(n,m)	建立內容為下標的字元矩陣

A(I,j) 為矩陣A 的第i列(column), 第j行(row)(column)。

```
x0=bindgen(2,3)
help,x0 & print,x0
print,'x0[3]=',x0[3] & print,'x0[1,1]=',x0[1,1]
x1=indgen(3,2)
help,x1 & print,x1 & print,'x1[2,1]=',x1[2,1]
print,'x1[2:3,*]=',x1[1:2,*]
x2=findgen(4,2)
help,x2 & print,x2
x3=dcindgen(3,2)
help,x3 & print,x3
x4=sindgen(3,2)
help,x4 & print,x4
end
```

```
% Compiled module: $MAIN$.
X0              BYTE      = Array[2, 3]
  0   1
  2   3
  4   5
x0[3]=   3
x0[1,1]=   3
X1              INT       = Array[3, 2]
      0      1      2
      3      4      5
x1[2,1]=      5
x1[2:3,*]=      1      2
      4      5
X2              FLOAT     = Array[4, 2]
    0.000000     1.00000     2.00000     3.00000
    4.00000     5.00000     6.00000     7.00000
X3              DCOMPLEX  = Array[3, 2]
(     0.00000000,     0.00000000)(     1.0000000,
0.00000000)(     2.0000000,     0.00000000)
(     3.0000000,     0.00000000)(     4.0000000,
0.00000000)(     5.0000000,     0.00000000)
X4              STRING    = Array[3, 2]
        0         1         2
        3         4         5
IDL>
```

建立內容為不規則數值或字串的矩陣

函數	功能
Var = [[a11,a21,…,an1], [a12,a22,…,an2], … [a1n,a2n,…,ann]]	建立內容為不規則數值或字串的矩陣，anxm為不規則數值或字串，n為矩陣行(row)的個數，m為矩陣列(column)的個數

```
x0=[[5,3,8],[4,9,6]]
help,x0 & print,x0
x1=[[5.1,3.8],[9.3,4.2], $
  [8.4,7.9]]
help,x1 & print,x1
print,'x1[4]=',x1[4]
print,'x1[0,2]=',x1[0,2]
end
```

```
% Compiled module: $MAIN$.
X0              INT       = Array[3, 2]
     5       3       8
     4       9       6
X1              FLOAT     = Array[2, 3]
     5.10000      3.80000
     9.30000      4.20000
     8.40000      7.90000
x1[4]=      8.40000
x1[0,2]=      8.40000
IDL>
```

轉換矩陣資料型態的函數(A為矩陣)

函數	功能
BYTE(A)	轉換A中所有元素為短整數
FIX(A)	轉換A中所有元素為整數
UINT(A)	轉換A中所有元素為無號整數
LONG(A)	轉換A中所有元素為長整數
ULONG(A)	轉換A中所有元素為無號長整數
LONG64(A)	轉換A中所有元素為64位元長整數
ULONG64(A)	轉換A中所有元素為64位元無號長整數
FLOAT(A)	轉換A中所有元素為浮點數
DOUBLE(A)	轉換A中所有元素為雙精度浮點數
COMPLEX(A)	轉換A中所有元素為複數的實部
DCOMPLEX(A)	轉換A中所有元素為雙精度複數的實部
STRING(A)	轉換A中所有元素為字元

```
x0=findgen(4,2)
Y0=fix(x0)
help,x0,y0 & print,x0,y0
x1=[[5.1,3.2],[6.7,2.8]]
y1=complex(x1)
help,x1,y1 & print,x1,y1
end
```

```
% Compiled module: $MAIN$.
X0              FLOAT     = Array[4, 2]
Y0              INT       = Array[4, 2]
    0.000000      1.00000      2.00000      3.0000
     4.00000      5.00000      6.00000      7.0000
      0      1      2      3
      4      5      6      7
X1              FLOAT     = Array[2, 2]
Y1              COMPLEX   = Array[2, 2]
     5.10000      3.20000
     6.70000      2.80000
(     5.10000,     0.000000)(      3.20000,
0.000000)
(     6.70000,     0.000000)(      2.80000,
0.000000)
IDL>
```

查詢矩陣相關資訊的函數

函數	功能
N_ELEMENTS(A)	求出矩陣A中元素的個數
FINITE(A)	判斷矩陣A中各個元素是否為有限

```
x0=[[2,1.3], $
 [!values.f_nan,5.2], $
[4.1,!values.f_infinity]]
help,x0 & print,x0
print,finite(x0)
end
```

```
% Compiled module: $MAIN$.
X0              FLOAT     = Array[2, 3]
     2.00000      1.30000
       NaN        5.20000
     4.10000          Inf
  1   1
  0   1
  1   0
IDL>
```

矩陣下標操作符號

符號	說明
0	代表下標的開始
*	代表全部下標
:	宣告下標範圍或下標增加量, n1 : n2, 起始下標n1, 結束下標n2。 n1:n2:n3 起始下標n1，結束下標n2，下標增加量n3。
,	區隔矩陣的維度

```
x0=[[1.0,2.1,3.2,4.3],[5.4,6.5,7.6,8.7],[9.8,10.9,11.1,12.2]]
print,x0
print,'x0[0]=',x0[0]
print,'x0[2:*,1]=',x0[2:*,1]
print,'x0[1:2,*]=',x0[1:2,*]
print,'x0[0:3:2,1]=',x0[0:3:2,1]
y2=fltarr(5,3)
print,'y2=',y2
y1=x0[1:3]
help,y1 & print,y1
y2[2:4]=y1
help,y1,y2 & print,y1,y2
y2[0,*]=y1
print,'y2[0,*]=',y2
y4=fltarr(5,3)
print,'y4=',y4
y4[1]=y1  ;從下標為1處開始存放資料
help,y4 & print,y4
end
```

```
% Compiled module: $MAIN$.
     1.00000      2.10000      3.20000      4.30000
     5.40000      6.50000      7.60000      8.70000
     9.80000      10.9000      11.1000      12.2000
x0[0]=     1.00000
x0[2:*,1]=      7.60000      8.70000
x0[1:2,*]=      2.10000      3.20000
     6.50000      7.60000
     10.9000      11.1000
x0[0:3:2,1]=      5.40000      7.60000
y2=     0.000000     0.000000     0.000000     0.000000     0.000000
    0.000000     0.000000     0.000000     0.000000     0.000000
    0.000000     0.000000     0.000000     0.000000     0.000000
Y1              FLOAT     = Array[3]
     2.10000      3.20000      4.30000
Y1              FLOAT     = Array[3]
Y2              FLOAT     = Array[5, 3]
     2.10000      3.20000      4.30000
    0.000000     0.000000      2.10000      3.20000      4.30000
    0.000000     0.000000     0.000000     0.000000     0.000000
    0.000000     0.000000     0.000000     0.000000     0.000000
y2[0,*]=      2.10000     0.000000      2.10000      3.20000      4.30000
     3.20000     0.000000     0.000000     0.000000     0.000000
     4.30000     0.000000     0.000000     0.000000     0.000000
y4=     0.000000     0.000000     0.000000     0.000000     0.000000
    0.000000     0.000000     0.000000     0.000000     0.000000
    0.000000     0.000000     0.000000     0.000000     0.000000
Y4              FLOAT     = Array[5, 3]
    0.000000      2.10000      3.20000      4.30000     0.000000
    0.000000     0.000000     0.000000     0.000000     0.000000
    0.000000     0.000000     0.000000     0.000000     0.000000
IDL>
```

回傳下標的數學函數(A為矩陣)

函數	功能
Result = MAX(A, Subscript)	計算矩陣A的最大值，Result記錄著最大值，Subscript記錄著最大值的下標位置
Result = MIN(A, Subscript)	計算矩陣A的最小值，Result記錄著最大值，Subscript記錄著最小值的下標位置\

```
x0=[[1.0,2.1,3.2,4.3],[5.4,6.5,7.6,8.7],[9.8,10.9,11.1,12.2]]
print,x0
result_1=max(x0,sub1)
print,'Max_=',result_1,' sub1=',sub1
result_2 = min(x0,sub2)
print,'Min_=',result_2,' sub2=',sub2
n1=4 & m1=3
i=sub1 mod n1  ;計算最大值第1維下標i
j = (sub1 - i)/n1 ;計算最大值第2維下標j
print,'i=',i,' j=',j,' x0(i,j)=',x0(i,j)
end
```

```
% Compiled module: $MAIN$.
    1.00000     2.10000     3.20000     4.300
    5.40000     6.50000     7.60000     8.700
    9.80000     10.9000     11.1000     12.20
Max_=     12.2000 sub1=        11
Min_=     1.00000 sub2=        0
i=        3 j=        2 x0(i,j)=     12.2000
IDL>
```

矩陣與函數所使用的區隔符號

符號	說明
() 小括號	函數用小括號隔參數
[] 中括號	變數用中括號區隔下標

IDL的優先將名稱當作函數來處理，最好的方式是變數的名稱避免使用到函數相同的名稱。

```
complex= bindgen(5,4)
print,complex
print,complex[3:4,2:3]
print,'complex[4,3]=',co
mplex[4,3]
print,'complex(4,3)=',co
mplex(4,3)
end
```

```
% Compiled module: $MAIN$.
  0   1   2   3   4
  5   6   7   8   9
 10  11  12  13  14
 15  16  17  18  19
 13  14
 18  19
complex[4,3]=  19
complex(4,3)=(     4.00000,     3.00000)
IDL>
```

矩陣轉換的函數

函數	功能
[A, B]	將矩陣A和B橫向併排，亦即擴充行(column)
[[A],[B]]	將矩陣A和B縱向併排，亦即擴充列(row)
REVERSE(A, k)	倒轉矩陣A中元素的順序，k是倒轉的維度
SHIFT(A, c, d)	平移矩陣A中元素的順序，c和d代表兩個維度的各個平移量
REFORM(A, c, d)	重新排列矩陣A中至維度為c x d的矩陣，但元素總數目不變
TRANSPOSE(A)	轉置矩陣A

```
A=[1,2,3] & B=[11,12,13]
AB = [A,B]
print,'[A,B]=',AB
A_B=[[A],[B]]
print,'[[A],[B]]='
print,A_B
A_R1=reverse(A_B,1)
print,'A_R=',A_R1
A_R2=reverse(A_B,2)
print,'A_R2=',A_R2
A_BB=[[A_B],[B]]
print,'[[A_B],[B]]',A_BB
A_S21=shift(A_BB,2,1)
print,' shift(A_BB,2,1)',A_S21
print,'A_B=',A_B
A_B_R = reform(A_B,2,3)
print,'reform(A_B,2,3)',A_B_R
A_B_T = transpose(A_B)
print,'transpose(A_B)',A_B_T
end
```

```
% Compiled module: $MAIN$.
[A,B]=      1      2      3     11     12     13
[[A],[B]]=
      1      2      3
     11     12     13
A_R=      3      2      1
     13     12     11
A_R2=     11     12     13
      1      2      3
[[A_B],[B]]
      1      2      3
     11     12     13
     11     12     13
 shift(A_BB,2,1)
     12     13     11
      2      3      1
     12     13     11
A_B=      1      2      3
     11     12     13
reform(A_B,2,3)
      1      2
      3     11
     12     13
transpose(A_B)
      1     11
      2     12
      3     13
IDL>
```

矩陣的數學運算(矩陣間的維度要相同)

指令	說明
A – B	將矩陣A中的元素減去矩陣B中相同位置的元素
A + B	將矩陣A中的元素加上矩陣B中相同位置的元素
A * B	將矩陣A中的元素乘以矩陣B中相同位置的元素
A + b	將矩陣A中的每一個元素加純量b
A / b	將矩陣A中的每一個元素除以純量b
A ^ b	將矩陣A中的每一個元素取純量b次方
A MOD b	將矩陣A中的每一個元素除以純量b後的餘數
SIN(A)、COS(A)、TAN(A)	將矩陣A中的每一個元素聯三角函數
EXP(A)	將矩陣A中的每一個元素取自然指數
ALOG(A)、ALOG10(A)	將矩陣A中的每一個元素取自然對數值、基底為10的對數值
ABS(A)	將矩陣A中的每一個元素取絕對值
SQRT(A)	將矩陣A中的每一個元素開根號

```
A=[[11,12,13],[21,22,23]]
B=indgen(3,2)
help,A,B & print,A,B
c1=A+B
help,c1 & print,c1
c2=A-B
help,c2 & print,c2
c3=A*B
help,c3 & print,c3
c4=float(B)/float(A)
help,c4 & print,c4
c5=A*3
help,c5 & print,c5
c6=A/3.
help,c6 & print,c6
c7=B^2
help,c7 & print,c7
c8= A mod 3
help,c8 & print,c8
end
```

```
% Compiled module: $MAIN$.
A               INT       = Array[3, 2]
B               INT       = Array[3, 2]
     11      12      13
     21      22      23
      0       1       2
      3       4       5
C1              INT       = Array[3, 2]
     11      13      15
     24      26      28
C2              INT       = Array[3, 2]
     11      11      11
     18      18      18
C3              INT       = Array[3, 2]
      0      12      26
     63      88     115
C4              FLOAT     = Array[3, 2]
   0.000000   0.0833333    0.153846
   0.142857    0.181818    0.217391
C5              INT       = Array[3, 2]
     33      36      39
     63      66      69
C6              FLOAT     = Array[3, 2]
    3.66667     4.00000     4.33333
    7.00000     7.33333     7.66667
C7              INT       = Array[3, 2]
      0       1       4
      9      16      25
C8              INT       = Array[3, 2]
      2       0       1
      0       1       2
IDL>
```

```
x=[[!pi/6,!pi/3],[!pi/2,!pi]]
y1=sin(x)
print,x,' sin(x)=',y1
y=[[30,60],[90,180]]
print,y,' cos(y)=',cos(y*!dtor)
z1=asin(y1)
print,z1,'asin(y1)=',z1*!radeg
A=indgen(3,2)+1
y2=exp(A) & z2=alog(y2)
print,'exp(A)=',y2 & print,'alog(y2)=',z2
print,'alog10(A)=',alog10(A)
c1=A - y
print,c1,' abs(c1)=',abs(c1)
print,y,' sqrt(y)=',sqrt(y)
end
```

```
% Compiled module: $MAIN$.
    0.523599      1.04720
     1.57080      3.14159
 sin(x)=
    0.500000     0.866025
     1.00000-8.74228e-008
     30      60
     90     180
 cos(y)=
    0.866025     0.500000
-4.37114e-008     -1.00000
    0.523599      1.04720
     1.57080-8.74228e-008
asin(y1)=
     30.0000      60.0000
     90.0000-5.00896e-006
exp(A)=      2.71828      7.38906      20.0855
     54.5981      148.413      403.429
alog(y2)=      1.00000      2.00000      3.00000
     4.00000      5.00000      6.00000
alog10(A)=     0.000000     0.301030     0.477121
    0.602060     0.698970     0.778151
   -29    -58
   -87   -176
 abs(c1)=
     29      58
     87     176
     30      60
     90     180
 sqrt(y)=
    5.47723      7.74597
    9.48683      13.4164
IDL>
```

矩陣相乘

運算子	數學運算	說明
#	array1#array2	以array1的columns(行)乘array2的rows(列) 結果矩陣 columns為array1的columns(行)，rows為array2的rows(列)
##	Array1##array2	以array1的rows(列)乘array2的columns(行) 結果矩陣 columns為array2的columns(行)，rows為array1的rows(列)

```
array1=[[1,2,1],[2,-1,2]]
array2=[[1,3],[0,1],[1,1]]
ar3=array1#array2
print,'ar3=',ar3
ar4=array1##array2
print,'ar4=',ar4
end
```

$$\text{Array1}=\begin{bmatrix}1 & 2 & 1\\ 2 & -1 & 2\end{bmatrix}$$

$$\text{Array2}=\begin{bmatrix}1 & 3\\ 0 & 1\\ 1 & 1\end{bmatrix}$$

$$\text{Ar3}=\begin{bmatrix}7 & -1 & 7\\ 2 & -1 & 2\\ 3 & 1 & 3\end{bmatrix}$$

$$\text{Ar4}=\begin{bmatrix}2 & 6\\ 4 & 7\end{bmatrix}$$