

MATLAB語法 矩陣取值與運算

盧家鋒 助理教授
alvin4016@ym.edu.tw

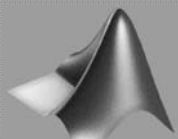


本週內容

- MATLAB基本語法與常用函式
- 矩陣維度與取值、填值
- 矩陣運算



基本語法與常用函式



CF

Define a Variable

- The 1st character must be a letter.
- The variable name is composed of...
 - Letters (a,b,...,y,z,A,B,...,Y,Z)
 - Numbers (0,1,2,...,9)
 - Underscore (_)
- At most 63 characters (or be automatically truncated)
- Use "=" to assign right-hand side value to the left-hand side variable

right: coord_x1
wrong: coord x1

CF

Data type of Variable

- Numbers
 - A=18
- Character/String
 - A='a'
 - A='This is a test.'
- Array/Matrix
 - A=[1 2 3 4 5 6 7 8]
 - A=[1 2 3 4; 5 6 7 8]
 - A=[1:4; 5:8]
 - A=[1:2:7; 2:2:8]

CF

Memory Demands

- Minimal unit: bit (位元)
- Binary format (0 or 1 in a bit)
- Common unit: byte (位元組), 1 byte = 8 bits

- 1 byte = 8 bits → 可儲存 $2^8=256$ 個數字
- 2 bytes = 16 bits → 可儲存 $2^{16}=65536$ 個數字
- 4 bytes = 32 bits → 可儲存 $2^{32}=4.2950e9$ 個數字
- 8 bytes = 64 bits → 可儲存 $2^{64}=1.8447e19$ 個數字

CF

Data precision

- **double** - Convert to 64-bit floating point.
- **single** - Convert to 32-bit floating point.
- **uint8** - Convert to unsigned 8-bit integer.
- **uint16** - Convert to unsigned 16-bit integer.
- **uint32** - Convert to unsigned 32-bit integer.
- **uint64** - Convert to unsigned 64-bit integer.
- **int8** - Convert to signed 8-bit integer.
- **int16** - Convert to signed 16-bit integer.
- **int32** - Convert to signed 32-bit integer.
- **int64** - Convert to signed 64-bit integer.
- **logical** - Convert numeric values to logical (8-bit).
- **char** - Converts nonnegative integer array into a character array (16-bit).

CF

Useful Function – check memory usage

- **whos**
 - List current variables in the workspace, long form.

```
Command Window
>> whos
Name      Size      Bytes Class  Attributes
A         20000x10000  1600000000 double
a          1x1         2 char
ans       1x1         8 uint64
```

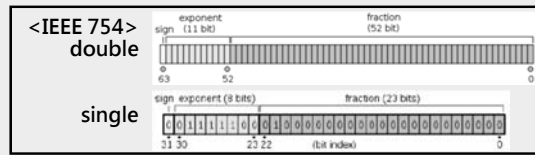
- $20000 \times 10000 \times (64/8) \text{ bytes} = 1.6e+09 \text{ bytes} = 1.6 \text{ Giga bytes (Gb)}$

CF

Check the capacity of data type

< Integer >

- **intmax**
 - Largest positive integer value.
- **intmin**
 - Smallest integer value.



If out of range → reset to maximal or minimal values

< double or single >

- **realmax**
 - Largest finite floating point number.
- **realmin**
 - Smallest positive normalized floating point number.

CF

Numbers vs. Characters

- Character is recognized as numbers by computers
 - ASCII code chart (0~127), UNICODE (128~65535)

ASCII 碼	字元	ASCII 碼	字元	ASCII 碼	字元	ASCII 碼	字元
032	20	056	8	080	P	104	h
033	!	057	9	081	Q	105	i
034	"	058	1A	082	R	106	j
035	#	059	1B	083	S	107	k
036	\$	060	1C	084	T	108	l
037	%	061	1D	085	U	109	m
038	&	062	1E	086	V	110	n
039	'	063	1F	087	W	111	o
040	(064	20	088	X	112	p
041)	065	21	089	Y	113	q
042	*	066	22	090	Z	114	r
043	+	067	23	091	[115	s
044	,	068	24	092	\	116	t
045	-	069	25	093]	117	u
046	.	070	26	094	^	118	v
047	/	071	27	095	_	119	w
048	0	072	28	096	`	120	x
049	1	073	29	097	a	121	y
050	2	074	2A	098	b	122	z
051	3	075	2B	099	c	123	{
052	4	076	2C	100	d	124	
053	5	077	2D	101	e	125	}
054	6	078	2E	102	f	126	~
055	7	079	2F	103	g	127	

(http://www.ym.edu.tw/~cflu/CFLu_course_matlabsig.html
第4週課程)

CF

Numbers vs. Characters

• Try it !

```
char([54 58 48 48 32 112 109 46])
```

- `abs('s')`
- `char(115)`
- %如 input 數值介於 127~65535 則按照 unicode 編碼
- `abs('我很欣賞你')`
- `char([25105 24456 27427 36062 20320])`

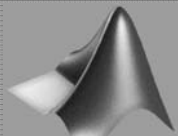
CF

Useful Functions

- **str2num**
 - Convert string matrix to numeric array.
- **num2str**
 - Convert numbers to a string.

Command Window	Command Window
<pre>>> str2num('44')</pre>	<pre>>> num2str(44)</pre>
<pre>ans =</pre>	<pre>ans =</pre>
<pre>44</pre>	<pre>44</pre>
<pre>>> str2num('44 21')</pre>	<pre>>> num2str([44 21])</pre>
<pre>ans =</pre>	<pre>ans =</pre>
<pre>44 21</pre>	<pre>44 21</pre>

CF

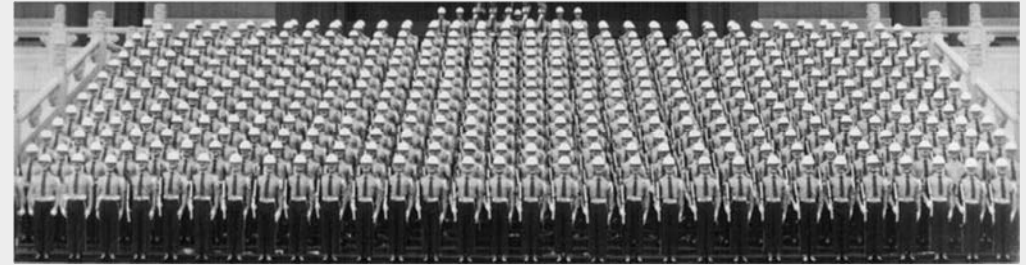


矩陣維度與取值填值

CF

Index of Elements in Matrix

- An efficient way to deal with massive data~



Who forgets his tie?!

CF

Format of Index

- Create matrix A.

12	2	33	4	46
3	7	4	6	5
4	6	64	6	84
6	5	6	7	37
7	4	45	23	78

A

Column-wise index

Row-column index

```
Command Window
>> A(18)
ans =
6
```

(1)	(6)	(11)	(16)	(21)
(2)	(7)	(12)	(17)	(22)
(3)	(8)	(13)	(18)	(23)
(4)	(9)	(14)	(19)	(24)
(5)	(10)	(15)	(20)	(25)

(1,1)	(1,2)	(1,3)	(1,4)	(1,5)
(2,1)	(2,2)	(2,3)	(2,4)	(2,5)
(3,1)	(3,2)	(3,3)	(3,4)	(3,5)
(4,1)	(4,2)	(4,3)	(4,4)	(4,5)
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)

```
Command Window
>> A(3,4)
ans =
6
```

CF

Get Value

- A specific range of index

12	2	33	4	46
3	7	4	6	5
4	6	64	6	84
6	5	6	7	37
7	4	45	23	78

A

Column-wise index

Row-column index

```
Command Window
Command Window
>> A([12:14 17:19])
ans =
4 64 6 6 6 7
```

(1)	(6)	(11)	(16)	(21)
(2)	(7)	(12)	(17)	(22)
(3)	(8)	(13)	(18)	(23)
(4)	(9)	(14)	(19)	(24)
(5)	(10)	(15)	(20)	(25)

(1,1)	(1,2)	(1,3)	(1,4)	(1,5)
(2,1)	(2,2)	(2,3)	(2,4)	(2,5)
(3,1)	(3,2)	(3,3)	(3,4)	(3,5)
(4,1)	(4,2)	(4,3)	(4,4)	(4,5)
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)

```
Command Window
Command Window
>> A(2:4,3:4)
ans =
4 6
64 6
6 7
```

CF

Get Value: Try it

- A specific range of index

12	2	33	4	46
3	7	4	6	5
4	6	64	6	84
6	5	6	7	37
7	4	45	23	78

Column-wise index

Row-column index

(1)	(6)	(11)	(16)	(21)
(2)	(7)	(12)	(17)	(22)
(3)	(8)	(13)	(18)	(23)
(4)	(9)	(14)	(19)	(24)
(5)	(10)	(15)	(20)	(25)

(1,1)	(1,2)	(1,3)	(1,4)	(1,5)
(2,1)	(2,2)	(2,3)	(2,4)	(2,5)
(3,1)	(3,2)	(3,3)	(3,4)	(3,5)
(4,1)	(4,2)	(4,3)	(4,4)	(4,5)
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)

CF

Delete Rows or Columns

12	2	33	4	46
3	7	4	6	5
4	6	64	6	84
6	5	6	7	37
7	4	45	23	78

Column-wise index

Row-column index

(1)	(6)	(11)	(16)	(21)
(2)	(7)	(12)	(17)	(22)
(3)	(8)	(13)	(18)	(23)
(4)	(9)	(14)	(19)	(24)
(5)	(10)	(15)	(20)	(25)

(1,1)	(1,2)	(1,3)	(1,4)	(1,5)
(2,1)	(2,2)	(2,3)	(2,4)	(2,5)
(3,1)	(3,2)	(3,3)	(3,4)	(3,5)
(4,1)	(4,2)	(4,3)	(4,4)	(4,5)
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)

```
Command Window
>> A(1:3,3:4)=[]
Subscripted assignment dimension mismatch.
```

```
Command Window
>> A(4,:)=[]
A =
    12     2    33     4    46
     3     7     4     6     5
     4     6    64     6    84
     7     4    45    23    78
```

CF

Useful Function - Find

- Find
 - Find indices of matched elements.
- ind = find(A==8)
- % get column-wise index
- [row, col] = find(A==8)
- % get row-column index

12	2	33	4	46
3	7	4	6	5
4	6	64	6	84
6	5	6	7	37
7	4	45	23	78

Try to find the indices of value 6~

CF

Relation and Logical Operators

% Relation operators

Equal	==
Not equal	~=
Less than	<
Greater than	>
Less than or equal	<=
Greater than or equal	>=

% logical operators

And	&
Or	
Not	~

CF

Set Value

- Rewrite values at specific locations

12	2	33	4	46
3	7	4	6	5
4	6	64	6	84
6	5	6	7	37
7	4	45	23	78

Column-wise index

Row-column index

```
Command Window
>> A(7)=100
A =
12 2 33 4 46
3 100 4 6 5
4 6 64 6 84
6 5 6 7 37
7 4 45 23 78
```

(1)	(6)	(11)	(16)	(21)
(2)	(7)	(12)	(17)	(22)
(3)	(8)	(13)	(18)	(23)
(4)	(9)	(14)	(19)	(24)
(5)	(10)	(15)	(20)	(25)

(1,1)	(1,2)	(1,3)	(1,4)	(1,5)
(2,1)	(2,2)	(2,3)	(2,4)	(2,5)
(3,1)	(3,2)	(3,3)	(3,4)	(3,5)
(4,1)	(4,2)	(4,3)	(4,4)	(4,5)
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)

```
Command Window
>> A(2,2)=100
A =
12 2 33 4 46
3 100 4 6 5
4 6 64 6 84
6 5 6 7 37
7 4 45 23 78
```

Set Value

- Replace value of 6 by 100

12	2	33	4	46
3	7	4	6	5
4	6	64	6	84
6	5	6	7	37
7	4	45	23	78

- `ind = find(A==6);`
- `A(ind) = 100;`

- `[rows,cols] = find(A==6);`

Set Value

- Rewrite values at specific locations

12	2	33	4	46
3	7	4	6	5
4	6	64	6	84
6	5	6	7	37
7	4	45	23	78

Column-wise index

Row-column index

```
Command Window
>> A([12:14 17:19])=100
A =
12 2 33 4 46
3 7 100 100 5
4 6 100 100 84
6 5 100 100 37
7 4 45 23 78
```

(1)	(6)	(11)	(16)	(21)
(2)	(7)	(12)	(17)	(22)
(3)	(8)	(13)	(18)	(23)
(4)	(9)	(14)	(19)	(24)
(5)	(10)	(15)	(20)	(25)

(1,1)	(1,2)	(1,3)	(1,4)	(1,5)
(2,1)	(2,2)	(2,3)	(2,4)	(2,5)
(3,1)	(3,2)	(3,3)	(3,4)	(3,5)
(4,1)	(4,2)	(4,3)	(4,4)	(4,5)
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)

```
Command Window
>> A(2:4,3:4)=100
A =
12 2 33 4 46
3 7 100 100 5
4 6 100 100 84
6 5 100 100 37
7 4 45 23 78
```

Set Value

- Rewrite values at specific locations

```
Command Window
>> A([12:14 17:19])=[100 102 104 101 103 105]
A =
12 2 33 4 46
3 7 100 101 5
4 6 102 103 84
6 5 104 105 37
7 4 45 23 78
```

12	2	33	4	46
3	7	4	6	5
4	6	64	6	84
6	5	6	7	37
7	4	45	23	78

Column-wise index

Row-column index

```
Command Window
>> A(2:4,3:4)=[100 101;102 103;104 105]
A =
12 2 33 4 46
3 7 100 101 5
4 6 102 103 84
6 5 104 105 37
7 4 45 23 78
```

(1)	(6)	(11)	(16)	(21)
(2)	(7)	(12)	(17)	(22)
(3)	(8)	(13)	(18)	(23)
(4)	(9)	(14)	(19)	(24)
(5)	(10)	(15)	(20)	(25)

(1,1)	(1,2)	(1,3)	(1,4)	(1,5)
(2,1)	(2,2)	(2,3)	(2,4)	(2,5)
(3,1)	(3,2)	(3,3)	(3,4)	(3,5)
(4,1)	(4,2)	(4,3)	(4,4)	(4,5)
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)

Matrix generator

- **Ones**
 - Create a Ones array
- **Zeros**
 - Create a Zeros array
- **rand**
 - Create an array with uniformly distributed pseudorandom numbers.
- **randn**
 - Create an array with normally distributed pseudorandom numbers

CF

More Elegant way to Define Matrix

- Define a 5x7x3 matrix with values from 1 to 105.

• `A=zeros(5,7,3);`

Please save as a Script File (*.m)~

• `A(:)=1:105`

`A(:,:,1) =`

```
1  6  11  16  21  26  31
2  7  12  17  22  27  32
3  8  13  18  23  28  33
4  9  14  19  24  29  34
5 10  15  20  25  30  35
```

`A(:,:,2) =`

```
36 41 46 51 56 61 66
37 42 47 52 57 62 67
38 43 48 53 58 63 68
39 44 49 54 59 64 69
40 45 50 55 60 65 70
```

`A(:,:,3) =`

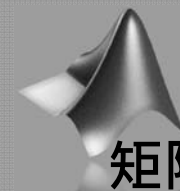
```
71 76 81 86 91 96 101
72 77 82 87 92 97 102
73 78 83 88 93 98 103
74 79 84 89 94 99 104
75 80 85 90 95 100 105
```

CF

Useful Functions

- **size**
 - Size of array.
- **length**
 - Length of vector.

CF



矩陣運算

CF

Top Rule: Matrix Dimension

- Common errors in matrix operation

```
Command Window
>> A(100)
Index exceeds matrix dimensions.
```

```
Command Window
>> A+B
Error using +
Matrix dimensions must agree.
```

```
Command Window
>> A*B
Error using *
Inner matrix dimensions must agree.
```

```
Command Window
>> C=[A B]
Error using horzcat
Dimensions of matrices being
concatenated are not consistent.
```

```
Command Window
>> C=[A; B]
Error using vertcat
Dimensions of matrices being
concatenated are not consistent.
```

CF

Top Rule: Matrix Dimension

- $A+B$ or $A-B$
 - Matrix A and matrix B must have same matrix dimension.
- $A*B$
 - Inner matrix dimension must agree.
 - Ex: A is $M \times N$ and B is $N \times P$

CF

Top Rule: Matrix Dimension

- $A.*B$ or $A./B$
 - Matrix A and matrix B must have same matrix dimension.
- The matrix operation was conducted element-by-element.

```
A =
6 6 6 6 6
6 6 6 6 6
6 6 6 6 6
6 6 6 6 6
6 6 6 6 6

B =
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
```

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

CF

Top Rule: Matrix Dimension

- A^k
 - Matrix A must be a square matrix.
- $A.^k$
 - No specific requirement for matrix dimension.

```
A =
1 4 7
2 5 8
3 6 9
```

```
>> A^2
ans =
30 66 102
36 81 126
42 96 150
```

```
>> A.^2
ans =
1 16 49
4 25 64
9 36 81
```

CF

Top Rule: Matrix Dimension

- A/B
 - $A \cdot \text{inv}(B)$
 - Inner matrix dimensions of A and inverse matrix of B must agree.
- $A \setminus B$
 - $\text{inv}(A) \cdot B$
 - Inner matrix dimensions of inverse matrix of A and B must agree.

CF

Matrix Transpose

- A'

```
>> A
A =
     1     4     7
     2     5     8
     3     6     9

>> A'
ans =
     1     2     3
     4     5     6
     7     8     9
```

CF

Try it~

- Create matrix A and matrix B.
- Set matrix C to be the blue part of A.
- Multiply B by C' element-by-element to get matrix D

12	2	33	4	46	A	8	6	1	B	×	=	48	42	23	D
3	7	4	6	5		4	2	9							
4	6	64	6	84	C'	6	7	23	84	37	78	336	74	702	
6	5	6	7	37		7	4	45							23
7	4	45	23	78											

CF

Useful Functions

- **permute**
 - Permute array dimensions.
- **reshape**
 - Reshape array.
- **repmat**
 - Replicate and tile an array.

```
>> A
A =
     1     5     9    13    17
     2     6    10    14    18
     3     7    11    15    19
     4     8    12    16    20

>> reshape(A,2,10)
ans =
Columns 1 through 6
     1     3     5     7     9    11
     2     4     6     8    10    12
Columns 7 through 10
    13    15    17    19
    14    16    18    20
```

CF

Exercise

• $X=[3\ 1\ 8\ 6\ 5\ 4\ 3\ 5\ 4\ 1\ 6\ 8\ 7\ 9\ 6]$

• 計算sample mean $\mu_x = \frac{\sum_{k=1}^N x_k}{N}$ (where $N=15$) 5.0667

• 計算sample variance $S_x^2 = \frac{\sum_{k=1}^N (x_k - \mu_x)^2}{N - 1}$ 5.9238

• 計算每一個x的z score $z_x = \frac{x_k - \mu_x}{S_x}$

[-0.84 -1.67 1.20 0.38 -0.02 -0.43 -0.84 -0.02 -0.43 -1.67 0.38 1.20 0.79 1.61 0.38]

CF

THE END

CF

alvin4016@ym.edu.tw