

Capitolo 1

Prime nozioni di Matlab

1.1 Primi passi con Matlab

Matlab manipola matrici, dunque vediamo anzitutto come si può creare una matrice

```
A=[1,2,3;4,5,6;7,8,9]
```

```
A =
```

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

Il simbolo ; significa: *nuova riga*; la virgola, o lo spazio, hanno il significato di *nuova colonna*. Come si vede la matrice A è stata introdotta per righe. Si noti che Matlab è *case sensitive* dunque B e b sono oggetti differenti

```
B=23
```

```
b=45
```

```
B =
```

```
    23
```

```
b =
```

```
    45
```

Le matrici possono essere visualizzate e modificate globalmente e anche per righe,

per colonne, o per elementi. Vediamo come.

Visualizziamo tutta la matrice

A

A =

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

Visualizziamo ora la seconda colonna

A(:,2)

ans =

```
2
5
8
```

Visualizziamo ora la terza riga

A(3,:)

ans =

```
7    8    9
```

Visualizziamo ora un singolo elemento

A(2,3)

ans =

```
6
```

Consideriamo ora un sottoblocco

A(2:3,1:2)

ans =

```
4    5
7    8
```

Per la modifica di una matrice, si può procedere in modo analogo

A(:,2)=[11;12;13]

A =

```
1    11    3
```

```
4    12    6
7    13    9
```

Si noti l'uso del punto e virgola, in modo che l'oggetto [11;12;13] sia un vettore colonna. Altri esempi

```
A(1,:)= [21 22 23]
```

```
A =
```

```
21    22    23
4     12     6
7     13     9
```

```
A(1:2, 1:2) = [101 102;103 104]
```

```
A =
```

```
101    102    23
103    104     6
7      13     9
```

1.2 Principali strumenti del calcolo matriciale

Matrice trasposta

```
A=[1 2; 3 4; 5 6]
```

```
B=A'
```

```
A =
```

```
1     2
3     4
5     6
```

```
B =
```

```
1     3     5
2     4     6
```

Determinante

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

```
det(A)
```

```
A =
```

```
1     2
```

```
      3      4
ans =
      -2
```

Matrice inversa

B=inv(A)

```
B =
-2.000000000000000    1.000000000000000
 1.500000000000000   -0.500000000000000
```

Verifica

A*B

B*A

```
ans =
 1.000000000000000    0
 0.000000000000000    1.000000000000000
ans =
 1.000000000000000    0
 0.000000000000000    1.000000000000000
```

Matrice nulla

A=zeros(3,2)

```
A =
 0    0
 0    0
 0    0
```

Matrice identica

A=eye(3)

```
A =
 1    0    0
 0    1    0
 0    0    1
```

Somma fra matrici

A=[1 2; 3 4]

B=[11 12; 13 14]

C=A+B

A =

1	2
3	4

B =

11	12
13	14

C =

12	14
16	18

Prodotto matriciale, il classico prodotto righe per colonne

A=[1 2;3 4; 5 6]

B=[1 2; 2 1]

C=A*B

A =

1	2
3	4
5	6

B =

1	2
2	1

C =

5	4
11	10
17	16

D=B*A

??? Error using ==> *

Inner matrix dimensions must agree.

Prodotto elemento per elemento

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

```
B=[11 12 ; 13 14]
```

```
C=A .* B
```

```
A =
```

```
    1    2  
    3    4
```

```
B =
```

```
   11   12  
   13   14
```

```
C =
```

```
   11   24  
   39   56
```

Prodotto di una matrice per uno scalare

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

```
B=12*A
```

```
A =
```

```
    1    2  
    3    4
```

```
B =
```

```
   12   24  
   36   48
```

Somma a una matrice di uno scalare

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

```
B=12+A
```

```
A =
```

```
    1    2  
    3    4
```

```
B =  
    13    14  
    15    16
```

1.3 Visualizzazione dei risultati

Eco a video. Il comando ; impedisce la visualizzazione dei risultati

```
a=3  
b=5  
c=a+b  
a =  
    3  
b =  
    5  
c =  
    8
```

```
a=3;  
b=5;  
c=a+b;  
c  
c =  
    8
```

1.4 Funzioni informative

Per conoscere la working directory: `cd`.

Per avere l'elenco delle variabili in memoria: `whos`.

Per avere l'elenco dei file contenuti nella working directory: `dir`

```
cd  
e:\vittorio\matvit  
cd prova
```

```
cd
```

```
dir
```

```
e:\vittorio\matvit\prova
```

```
.          imco_image_list.m  imco_strip_list.m
..         imco_image_num.m  imco_strip_num.m
imco_coord.m  imco_list.m      imco_which_stripe.m
imco_filter.m  imco_load_txt.m  imco_write_txt.m
imco_filter_image.m  imco_point_list.m
imco_filter_point.m  imco_point_num.m
```

```
whos
```

Name	Size	Bytes	Class
A	2x2	32	double array
B	2x2	32	double array
C	2x2	32	double array
a	1x1	8	double array
ans	2x2	32	double array
b	1x1	8	double array
c	1x1	8	double array

```
Grand total is 19 elements using 152 bytes
```

1.5 Funzioni

```
function norma=edu_norma(x)
% edu_norma: calcola la norma di un vettore assegnato
% -----
% Prototipo
%   norma=edu_norma(x)
% Input
%   x (double, nx1): vettore passato dall'utente
% Output
%   norma (double, 1x1): norma del vettore passato
% -----
% Vittorio Casella, 6.4.2004
%
%
%
norma=sqrt(x'*x);
```

Figura 1 – Primo esempio di programma