MATRICI (ARRAY) IN MATLAB/OCTAVE

MANOLO VENTURIN

UNIVERSITÀ DEGLI STUDI DI PADOVA
DIP. DI MATEMATICA PURA ED APPLICATA

A. A. 2007/2008

INDICE

- Cosa sono
- Esempio
- Operazioni sulle matrici
- Importare matrici da file
- Vettorizzazione (operazioni $.*$, $./$ e $.^$)

Cosa sono

- Una matrice è un insieme di **variabili**, tutte dello **stesso tipo**, identificate da un nome unico.
- Ogni valore all'interno di una matrice si chiama **elemento** della matrice e viene **identificato** dal **nome** della matrice e da un **indice/coppia di indici** che punta ad una particolare posizione all'interno della matrice.

Esempio: Battaglia navale

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1				2							3	3	3		
2				2											
3	1														
4						2	2								1
5											3				
6											3				
7			2				1				3				
8			2												
9															
10								4	4	4	4				

Informazioni sulla matrice

- Numero di righe
- Numero di colonne
- Valori degli elementi della matrice

Casi particolari

- **Vettore orizzontale:**
matrice con una riga e più colonne
- **Vettore verticale:**
matrice con una colonna e più righe
- **Scalare (numero):**
matrice con una riga e una colonna

Operazioni sulle matrici

- Creazione di una matrice
- Assegnazione di elementi di una matrice
- Estrarre elementi da una matrice

Per elementi intendiamo:

- Singolo o singoli elementi della matrice
- Una o più colonne della matrice
- Una o più righe della matrice

Creazioni di matrici: elemento per elemento

```
>> % Creazione per assegnamento diretto  
>> % Matrice 3 x 3  
>> % Ogni riga è separata dal carattere ;  
>> x = [1 2 3; 4 5 6; 7 8 9]
```

```
x =
```

1	2	3
4	5	6
7	8	9

Creazioni di matrici: elemento per elemento

```
>> % VETTORE COLONNA  
>> % Matrice 3 x 1  
>> % Ogni riga è separata dal carattere ;  
>> x = [1 ; 2 ; 3]  
x =  
    1  
    2  
    3
```

Creazioni di matrici: elemento per elemento

```
>> % VETTORE RIGA
```

```
>> % Matrice 1 x 3
```

```
>> % Ogni colonna è separata da uno  
spazio o da una virgola
```

```
>> x = [1 2 3]
```

```
x =
```

```
1 2 3
```

```
>> % oppure
```

```
>> x = [1,2,3]
```

```
x =
```

```
1 2 3
```

Trasporsto di una matrice

```
>> % Scambio righe-colonne attraverso il  
comando ' posto alla fine della matrice
```

```
>> x = [1 2 ; 3 4; 5 6]
```

```
x =
```

```
    1    2
```

```
    3    4
```

```
    5    6
```

```
>> y = x'
```

```
y =
```

```
    1    3    5
```

```
    2    4    6
```

Trasposto per i vettori

```
>> % Vettori riga diventano vettori colonna
```

```
>> x = [1 2 3]
```

```
x =
```

```
    1    2    3
```

```
>> % mentre vettori colonna diventano  
vettori riga
```

```
>> y = x'
```

```
y =
```

```
    1
```

```
    2
```

```
    3
```

Sovrascrivere variabili

>> % Una volta creata una variabili
essa può essere sovrascritta

```
>> x = 2;
```

```
>> x = x + 2;
```

```
x =
```

```
4
```

Sovrascrivere variabili

```
>> % vale anche per le matrici
```

```
>> x = [1 2 3 4]
```

```
x =
```

```
    1    2    3    4
```

```
>> x = x';
```

```
x =
```

```
    1
```

```
    2
```

```
    3
```

```
    4
```

Creazioni di matrici: funzioni predefinite

- Funzioni per vettori
(`linspace, :`)
- Funzioni per le matrici
(`ones, zeros`)

Creazioni di vettori: linspace

>> % linspace crea un vettore di
elementi equispaziati

>> x = linspace(0,10,6)

x =

0 2 4 6 8 10

Creazioni di vettori: linspace

```
>> % converto vettore riga in vettore  
colonna
```

```
>> y = x'
```

```
y =
```

```
0
```

```
2
```

```
4
```

```
6
```

```
8
```

```
10
```

Creazioni di vettori: operatore :

>> % Esempio: creazione di un vettore
equispaziato da 1 da 1 a 8

```
>> x = 1:8
```

```
x =
```

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

>> % Esempio: creazione di un vettore
equispaziato da 2 da 1 a 8

```
>> x = 1:2:8
```

```
x =
```

```
1 3 5 7
```

Esempio

tabulazione di una funzione

```
>> x = linspace(0,2*pi,6)';
```

```
>> y = sin(x); z = cos(x);
```

```
>> [x y z]
```

```
ans =
```

0	0	1.0000
1.2566	0.9511	0.3090
2.5133	0.5878	-0.8090
3.7699	-0.5878	-0.8090
5.0265	-0.9511	0.3090
6.2832	-0.0000	1.0000

Creazioni di matrici: zeros

>> % Esempio: creazione di una
matrice di 2 righe e 3 colonne di tutti
zeri

```
>> x = zeros(2,3)
```

```
x =
```

```
    0    0    0
```

```
    0    0    0
```

Creazioni di matrici: ones

```
>> % Esempio: creazione di una  
matrice di 2 righe e 3 colonne di tutti 1
```

```
>> x = ones(2,3)
```

```
x =
```

```
    1    1    1  
    1    1    1
```

Creazioni di matrici: ones

```
>> % Esempio: creazione di una  
matrice di 2 righe e 3 colonne di tutti 2
```

```
>> x = ones(2,3)*2
```

```
x =
```

```
    2    2    2
```

```
    2    2    2
```

Estrarre elementi

>> % Estrarre un elemento

>> % Se A è una matrice allora A(i,j)
estrae l'elemento alla riga i e colonna j

>> A = [1:3 ; 4:6 ; 7:9]

>> b = A(2,3)

b =

6

Estrarre elementi

```
>> % Estrarre una colonna
```

```
>> % Se A è una matrice allora A(:,i)  
estrae la colonna i-esima
```

```
>> A = [1:3 ; 4:6 ; 7:9]
```

```
>> b = A(:,3)
```

```
b =
```

```
3
```

```
6
```

```
9
```

Estrarre elementi

```
>> % Estrarre una colonna
```

```
>> % Se A è una matrice allora A(i,:)
    estraee la riga i-esima
```

```
>> A = [1:3 ; 4:6 ; 7:9]
```

```
>> b = A(3,:)
```

```
b =
```

```
    7    8    9
```

Estrarre elementi da un vettore

>> % Se v è un vettore (riga o colonna)
allora $v(i)$ estrae l' i -esimo elemento

>> $v = [4,3,2,1]$

>> $b = v(3)$

$b =$

2

Impostare un elemento

```
>> % Impostare un elemento della  
matrice
```

```
>> A = [1:3 ; 4:6 ; 7:9];
```

```
>> A(1,1) = 100
```

```
A =
```

```
100    2    3
```

```
4      5    6
```

```
7      8    9
```

Impostare una riga

```
>> % Impostare una riga della matrice
```

```
>> A = [1:3 ; 4:6 ; 7:9];
```

```
>> A(2,:) = [0 0 0]
```

```
A =
```

1	2	3
0	0	0
7	8	9

Impostare una colonna

```
>> % Impostare una colonna della  
matrice
```

```
>> A = [1:3 ; 4:6 ; 7:9];
```

```
>> A(:,2) = [0 ;0 ;0]
```

```
A =
```

1	0	3
4	0	6
7	0	9

Impostare una sottomatrice

```
>> % Impostare una sottomatrice della  
matrice con dei vettori
```

```
>> A = [1:3 ; 4:6 ; 7:9];
```

```
>> A(1:2,1:2) = [0 0;0 0]
```

```
A =
```

```
0    0    3
```

```
0    0    6
```

```
7    8    9
```

Impostare una sottomatrice

```
>> % Impostare una sottomatrice della  
matrice con dei vettori
```

```
>> A = [1:3 ; 4:6 ; 7:9];
```

```
>> A([1,3],[1,3]) = [0 0;0 0]
```

```
A =
```

```
    0     2     0  
    4     5     6  
    0     8     0
```

Impostare un elemento di un vettore

```
>> % raddoppio del 3 elemento
```

```
>> v = [4,3,2,1]
```

```
>> v(3) = v(3)*2
```

```
v =
```

```
    4     3     4     1
```

Ridimensionamento dinamico della matrice

```
>> % Ridimensionamento dinamico
```

```
>> A = [1:3 ; 4:6 ; 7:9];
```

```
>> A(4,4) = 1
```

```
A =
```

1	2	3	0
4	5	6	0
7	8	9	0
0	0	0	1

Ridimensionamento dinamico della matrice

```
>> % Ridimensionamento dinamico di un  
vettore riga (utile esempio)
```

```
>> v = 1:3;
```

```
>> v(4) = 4
```

```
v =
```

```
    1    2    3    4
```

```
>> % Concateno due vettori
```

```
>> v = [v,v]
```

```
v =
```

```
    1    2    3    4    1    2    3    4
```

Ridimensionamento dinamico della matrice

```
>> % Ridimensionamento dinamico di un  
vettore colonna (utile esempio)
```

```
>> v = (1:2)'
```

```
v =
```

```
1
```

```
2
```

```
>> % Concateno due vettori
```

```
>> v = [v;1]
```

```
v =
```

```
1
```

```
2
```

```
1
```

Informazioni su una matrice

```
>> % Numero di righe
```

```
>> A = [1:4 ; 5:8 ; 9:12];
```

```
>> r = size(A,1)
```

```
r =
```

```
3
```

```
>> % Numero di colonne
```

```
>> c = size(A,2)
```

```
c =
```

```
4
```

Informazioni su un vettore

```
>> % Lunghezza di un vettore sia per  
vettore riga sia per vettori colonna
```

```
>> v = [1,2,3,4];
```

```
>> n = length(v)
```

```
n =
```

```
4
```

Esercizio in classe

SCRIVERE LA MATRICE DELL'ESEMPIO
SULLA BATTAGLIA NAVALE
DOVE LE CASELLE GRIGIE SENZA NUMERI
RAPPRESENTANO L'ELEMENTO 0

Esercizio

```
>> % Soluzione
```

```
>> A = zeros(10,15);
```

```
>> % Se si specifica uno scalare (numero)  
    allora tutti gli elementi che assegno  
    avranno lo stesso valore
```

```
>> A(10,8:11) = 4;
```

```
>> A(5:7,11) = 3;
```

```
>> A(4,6:7) = 2;
```

```
>> A(4,15) = 1;
```

```
>> % ed altri
```

Importare matrici da file

Supponiamo di avere un file [risultati.dat] così tabulato:

0.0000000e+000	0.0000000e+000	1.0000000e+000
1.2566371e+000	9.5105652e-001	3.0901699e-001
2.5132741e+000	5.8778525e-001	-8.0901699e-001
3.7699112e+000	-5.8778525e-001	-8.0901699e-001
5.0265482e+000	-9.5105652e-001	3.0901699e-001
6.2831853e+000	-2.4492936e-016	1.0000000e+000

Importare matrici da file

>> % Il file risultati.dat può essere importato direttamente nel programma utilizzando la seguente sintassi

```
>> A = load('risultati.dat')
```

```
A =
```

```
    0    0  1.0000
 1.2566  0.9511  0.3090
 2.5133  0.5878 -0.8090
 3.7699 -0.5878 -0.8090
 5.0265 -0.9511  0.3090
 6.2832 -0.0000  1.0000
```

Importare matrici da file

- Tra " viene indicato il nome del file (stringa di caratteri) risultati.dat
- L'istruzione load legge il file (risultati.dat) e salva i dati nella matrice di uscita A
- Per estrarre le singole colonne della matrice utilizzo le istruzione viste in precedenza
- Per disegnare i dati utilizzo le istruzione che sono spiegate nella lezione di grafica

Salvataggio dati

- E' possibile salvare i dati presenti nel workspace in un formato proprietario mediante il comando `save nomefile`
- E' possibile leggere i dati salvati nel file proprietario mediante il comando `load nomefile`

Salvataggio dati

```
>> x = linspace(0,2*pi,6)';  
>> y = sin(x); z = cos(x);  
>> A = [x y z]  
>> % salva nel file risA tutti i dati  
    presenti nel workspace  
>> save risA
```

Salvataggio dati

```
>> clear all
```

```
>> % importa i dati
```

```
>> load risA
```

```
>> who
```

Your variables are:

```
A x y z
```

Vettorizzazione

vettore * costante

```
>> % vettore
```

```
>> v = [1,2,3,4]
```

```
v =
```

```
    1    2    3    4
```

```
>> % vettore * costante = tutti gli  
    elementi sono moltiplicati per la  
    costante
```

```
>> v = v*3
```

```
v =
```

```
    3    6    9   12
```

Vettorizzazione vettore / costante

```
>> % vettore
```

```
>> v = [1,2,3,4]
```

```
v =
```

```
    1    2    3    4
```

```
>> % vettore / costante = tutti gli  
    elementi sono divisi per la costante
```

```
>> v = v/3
```

```
v =
```

```
    0.3333    0.6667    1.0000    1.3333
```

Vettorizzazione vettore + costante

```
>> % vettore
```

```
>> v = [1,2,3,4]
```

```
v =
```

```
    1    2    3    4
```

```
>> % vettore + costante = tutti gli  
    elementi sono addizionati della  
    costante
```

```
>> v = v+3
```

```
v =
```

```
    4    5    6    7
```

Vettorizzazione vettore - costante

```
>> % vettore
```

```
>> v = [1,2,3,4]
```

```
v =
```

```
    1    2    3    4
```

```
>> % vettore - costante = tutti gli  
    elementi sono sottratti della costante
```

```
>> v = v-3
```

```
v =
```

```
   -2   -1    0    1
```

Vettorizzazione vettore .* vettore

```
>> % vettore
```

```
>> v = [1,2,3,4]; w = [2,3,4,5];
```

```
>> % vettore .* vettore = componente  
per componente  $y(i) = v(i)*w(i)$ 
```

```
>> y = v.*w
```

```
y =  
    2     6    12    20
```

Vettorizzazione vettore ./ vettore

```
>> % vettore
```

```
>> v = [1,2,3,4]; w = [2,3,4,5];
```

```
>> % vettore ./ vettore = componente  
per componente  $y(i) = v(i)/w(i)$ 
```

```
>> y = v./w
```

```
y =
```

```
0.5000    0.6667    0.7500    0.8000
```

Vettorizzazione vettore .^ vettore

```
>> % vettore
```

```
>> v = [1,2,3,4]; w = [2,3,4,5];
```

```
>> % vettore .^ vettore = componente  
per componente  $y(i) = v(i)^w(i)$ 
```

```
>> y = v.^w
```

```
y =
```

1

8

81

1024

Funzioni MatLab/Octave vettorizzate

```
>> % Tutte le funzioni MatLab/Octave  
sono vettorizzate
```

```
>> v = [1,2,3,4]*pi/2;
```

```
>> y = sin(v)
```

```
y =
```

```
1.0000    0.0000   -1.0000   -0.0000
```