
PON 2007 – 2013
Liceo Scientifico *Leonardo da Vinci*
Vallo della Lucania
Nuovi percorsi matematici:
Osservare, descrivere, costruire.

Matlab - 3: Operazioni su matrici

Arturo Stabile
Vallo della Lucania
26 Settembre 2008

Operazioni su matrici

- `sum(A)`
 - Restituisce la somma per colonne degli elementi di A
- `cumsum(A)`
 - Somma cumulativa degli elementi delle colonne di A
- `prod(A)`, `cumprod(A)`
 - Invece della somma si effettua il prodotto

Esempio – 1

```
a =  
    0.8381    0.6813    0.8318  
    0.0196    0.3795    0.5028  
  
>> sum(a)  
  
ans =  
    0.8578    1.0608    1.3346
```

```
>> cumsum(a)  
  
ans =  
    0.8381    0.6813    0.8318  
    0.8578    1.0608    1.3346
```

Se A è un vettore viene calcolata la somma o prodotto di tutti gli elementi

Esempio – 2

```
>> a=ones(1,5)
```

```
a =
```

```
1  1  1  1  1
```

```
>> sum(a)
```

```
ans =
```

```
5
```

```
>> prod(a)
```

```
ans =
```

```
1
```

```
>> cumsum(a)
```

```
ans =
```

```
1  2  3  4  5
```

```
>> cumprod(a)
```

```
ans =
```

```
1  1  1  1  1
```

Come sommiamo per righe?

- `sum(A)'`
 - Somma degli elementi delle colonne di A

```
>> a=rand(2,3)
a =
    0.7095    0.3046    0.1934
    0.4289    0.1897    0.6822
```

```
>> sum(a)
ans =
    1.2075
    1.3008
```

Funzioni specifiche per matrici

- **triu(A)**
 - Estrae la parte triangolare superiore di una matrice
- **tril(A)**
 - Estrae la parte triangolare inferiore di una matrice

```
> a=rand(3,3)
a =
    0.3028    0.6979    0.8537
    0.5417    0.3784    0.5936
    0.1509    0.8600    0.4966

>> triu(a)
ans =
    0.3028    0.6979    0.8537
         0    0.3784    0.5936
         0         0    0.4966
```

Altre funzioni – 1

- `linspace(a,b,n)`
 - Restituisce un vettore di `n` elementi linearmente intervallati tra `a` e `b`
- `logspace(a,b,n)`
 - Restituisce un vettore di `n` elementi logaritmicamente intervallati tra 10^a e 10^b

```
>> linspace(3,4,4)
ans =
    3.0000    3.3333    3.6667    4.0000
```

Altre funzioni – 2

- `max(A)`
 - Vettore: elemento massimo
 - Matrice: riga di elementi massimi per colonna
- `min(A)`
 - Duale di `max(A)`
- `sort(A)`
 - Dispone in ordine crescente gli elementi delle colonne di A

Funzioni “statistiche”

- `mean(A)`

- Restituisce un vettore riga contenente il valor medio di ogni colonna

- `std(A)`

- Restituisce un vettore riga contenente la deviazione standard di ogni colonna

Esercizi – 1

- Utilizzare due metodi per creare un vettore x che ha 100 elementi linearmente intervallati tra 5 e 28
- Utilizzare due metodi per creare un vettore x che ha gli elementi linearmente intervallati di 0.2 tra 2 e 14
- Creare un vettore x che ha 50 elementi logaritmicamente intervallati tra -2 e 5

Esercizi – 2

- Utilizzare Matlab per creare un vettore X che ha 6 valori compresi tra 0 e 10 (estremi inclusi). Creare una matrice A in modo che la prima riga contenga i valori $3x$ e la seconda riga contenga i valori $5x-20$
- Ripetere l'esercizio precedente, dove al posto delle righe si considerano le colonne

Esercizi – 3

- Creare una matrice casuale A 4x4
 - Ordinare gli elementi di ogni colonna e memorizzare il risultato in B
 - Ordinare gli elementi di ogni riga e memorizzare il risultato in C
 - Sommare gli elementi di ogni colonna e memorizzare il risultato in D
 - Sommare gli elementi di ogni riga e memorizzare il risultato in E

Esercizi – 4

- Creare una matrice casuale A 5×5 con elementi compresi tra 0 e 20
 - Calcolare i valori massimi e minimi di ogni riga
 - Calcolare i valori massimi e minimi di ogni colonna
 - Calcolare il massimo ed il minimo di A

Matrici e file

- In Matlab è possibile leggere o scrivere delle variabili in un file di testo oppure in un file binario
- load
 - Carica da un file una variabile nel workspace
- save
 - Salva una variabile del workspace in un file
- Maggiori dettagli in seguito

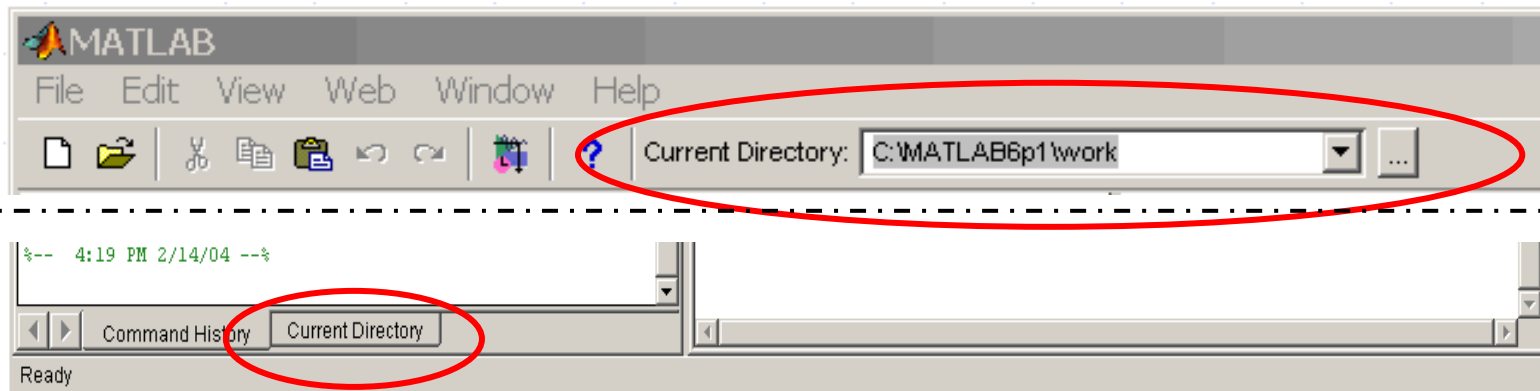
Tipi di file in Matlab

- M-File (estensione .m)
 - Funzione/programma con variabili locali o globali memorizzata in un file
- MAT-File (estensione .mat)
 - Precedenti spazi di lavoro (workspace) memorizzati in un file
- M-File e MAT-File sono memorizzati sul disco fisso o su altra periferica

Cartella di lavoro

■ Current-Directory

- Cartella corrente su cui si sta lavorando
- Se si digita il nome di un M-file nella Command Window, quel nome viene cercato nella Current-Directory; se presente viene eseguito

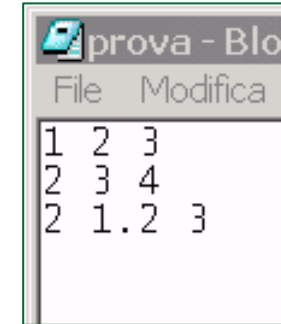


Dove posizionare M-file e MAT-file

■ Search-Path

- ❑ Determina le cartelle del filesystem sono in linea oltre alla directory corrente
- ❑ M-File e MAT-file sono cercati automaticamente nella Search-Path

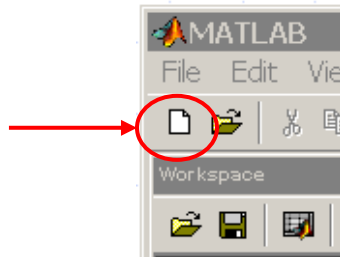
Il comando `load`



- Il file di testo deve essere organizzato come una tabella rettangolare di numeri separati da spazio
 - una riga per linea (righe separate da invio)
 - uno stesso numero di elementi per riga
- Se il file si chiama `prova.dat` allora
`load prova.dat`
carica nel workspace la variabile `prova`

Note

- Si può usare l'estensione che si vuole basta che sia diversa da .m e .mat
- Il file contenente la matrice deve essere posizionato nella cartella di lavoro o in qualsiasi cartella presente nel search-path
- Per creare la matrice si può utilizzare l'editor di Matlab
 - Cliccare su



Il comando `save`

- Le variabili create nella finestra comandi possono essere salvate in un file `.mat`
 - Si può salvare tutto il workspace
- La sintassi da utilizzare è
`save NomeFile NomeVar1 ... NomeVark`
- Per salvare il file in formato testuale
`save NFile NV1 ... NVk -ascii`

Attenzione

- Se prima eseguiamo il comando

`save NomeFile Var1`

e dopo il comando

`save NomeFile Var2`

il contenuto di **NomeFile** viene perso ed aggiornato con **Var₂**

- Per preservare il contenuto di **NomeFile** usare il comando

`save NomeFile Var2 -append`

Ancora sul comando `load`

- Per caricare nel workspace una variabile (gruppo di variabili) salvata con il comando `save` si usa la sintassi

`load` NomeFile NomeVar₁ ... NomeVar_k

Note

- Si usa la sintassi `save('NomeFile','var1','var2')`, quando il nome del file o i nomi delle variabili sono memorizzate come stringhe
- Si usa la sintassi `load('NomeFile')`, quando il nome del file è memorizzato in una stringa oppure quando NomeFile contiene spazi

Esercizio

- Creare tre matrici A, B e C con dimensioni e contenuto a vostra scelta
- Salvare le tre matrici in nel file `pon`
- Cancellare dal workspace le tre matrici
- Caricare dal file `pon` la matrice B
- Modificare il contenuto in riga 1 colonna 1
- Salvare la matrice B nel file `pon`
- Cancellare la matrice B dal workspace
- Caricare tutto il contenuto del file `pon` nel workspace
- Esaminare il contenuto delle matrici

Operatori relazionali

- Gli operatori relazionali sono:

- == uguale
- ~= diverso
- < minore
- <= minore o uguale
- > maggiore
- >= maggiore o uguale

- 0 indica falso

- 1 indica vero

```
>> x=2
x =
    2
>> x==0
ans =
    0
>> x==2
ans =
    1
```

Esempi

```
>> a=[1,2,2,1];
>> b=ones(1,4);
>> a==b
ans =
     1     0     0     1
>> a>b
ans =
     0     1     1     0
```

```
>> a=rand(2,3)
a =
     0.9501     0.6068     0.8913
     0.2311     0.4860     0.7621
>> a>0.6
ans =
     1     1     1
     0     0     1
```

Operatori logici

■ Gli operatori logici sono:

- & and logico
- | or logico
- ~ not logico

```
>> x=1;y=-1;
>> x>0 & y >2
ans =
    0
>> x>0 | y>2
ans =
    1
```

```
>> z=rand(1,5)
z =
    0.6038    0.2722    0.1988    0.0153    0.7468
>> z>0.2 & z<0.7
ans =
    1     1     0     0     0
```

Esempio

```
>> z=rand(3,5)
z =
    0.4451    0.4186    0.2026    0.0196    0.8318
    0.9318    0.8462    0.6721    0.6813    0.5028
    0.4660    0.5252    0.8381    0.3795    0.7095
>> z>0.2 & z<0.7
ans =
     1     1     1     0     0
     0     0     1     1     1
     1     1     0     1     0
```

Logical subscripting

- Abbiamo visto come creare con operatori relazionali o logici vettori/matrici “logici”
- Se L è un vettore/matrice “logico” ed X è un vettore/matrice della stessa dimensione di L , allora $X(L)$ specifica gli elementi di X dove gli elementi di L sono diversi da 0
 - $X(L)$ è un sottovettore/sottomatrice di X
 - L specifica quali elementi di X devono essere presi

Esempi

```
>> x=fix(3*rand(1,7)+1)
```

```
x =
```

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

```
>> L = x==3
```

```
L =
```

```
    1    1    0    1    0    1    0
```

```
>> x(L)
```

```
ans =
```

```
    3    3    3    3
```

```
>> L=(x>0 & x<3)
```

```
L =
```

```
    0    0    1    0    1    0    1
```

```
>> x(L)
```

```
ans =
```

```
    1    2    2
```

Esempio

```
>> x=fix(3*rand(1,7))  
x =  
    1    0    2    1    2    2    1  
  
>> y=fix(3*rand(1,7))  
y =  
    0    0    1    1    0    1    1  
  
>> L = x&y  
L =  
    0    0    1    1    0    1    1  
  
>> x(L)  
ans =  
    2    1    2    1
```

La funzione `find`

- `find(A)`
 - Crea un vettore/matrice contenente gli indici degli elementi non nulli di A

```
>> y=fix(2*rand(1,7))  
y =  
    0    0    1    1    0    1    1  
  
>> find(y)  
ans =  
     3     4     6     7
```

```
>> y(ans)  
ans =  
     1     1     1     1
```


Esempio con matrici

```
>> y=fix(2*rand(2,7)).*rand(2,7)
```

```
y =
```

```
    0    0  0.5155    0  0.5798    0    0
0.9883    0  0.3340    0    0  0.6405    0
```

```
>> find(y)
```

```
ans =
```

```
2
5
6
9
12
```

```
>> y(ans)
```

```
ans =
```

```
0.9883
0.5155
0.3340
0.5798
0.6405
```

Esercizio – 1

- Dati $x=[5 \ -3 \ 18 \ 4]$ ed $y=[-9 \ 13 \ 7 \ 4]$, quale è il risultato delle seguenti operazioni?
 - $z = x < y$
 - $z = (x == y)$
 - $z = \sim y > x$
 - $z = x \& y$
 - $z = x | y$
 - $z = \text{xor}(x,y)$

Esercizi – 2

- Dati $x=[-4 \ -1 \ 0 \ 2 \ 10]$ ed $y=[-5 \ -2 \ 2 \ 5 \ 9]$, trovare i valori e gli indici di x che sono maggiori dei corrispondenti elementi di y
- Generare un vettore casuale R con elementi tra 1 e 10; trovare i valori di R compresi tra 4 e 8
- Generare due vettori casuali A e B di 20 interi compresi tra 23 e 48, selezionare tutti gli elementi di A che sono maggiori dei corrispondenti di B e minori di 39

Esercizi – 3

- Dato i seguenti due vettori

`p1`=[12.3 3.4 -5.4 4 7.8 22.1 44.2 43.1 12.6 24.5]

`p2` = sort(`p1`)

- Trovare il numero degli elementi di `p1` maggiori di 21
- Il numero degli elementi di `p1` maggiori dei corrispondenti di `p2`

Esercizi – 4

- Generare 3 vettori casuali, A B e C, di 12 elementi interi compresi tra 100 e 200
 - Determinare gli elementi di A maggiori dei corrispondenti di B e di C
 - Determinare gli elementi di A maggiori dei corrispondenti di B o di C
 - Determinare gli elementi di A maggiori dei corrispondenti di B o di C ma non di entrambi

Esercizi di riepilogo – 1

- Dato il vettore $z=50:-5:10$ cosa restituiscono i seguenti comandi MATLAB ?
 - `length(z)`
 - `z'`
 - `z.*z(9:-1:1)`
 - `z(1:3:9)=ones(1,3)`
 - `z([3 1 7 5])=zeros(1,4)`

Esercizi di riepilogo – 2

- Data la matrice $A=[1 \ 2 \ 3; 2 \ 3 \ 4; 3 \ 4 \ 5]$, cosa restituiscono i seguenti comandi MATLAB ?
 - $[p,q]=\text{size}(A)$
 - $A.*A'$
 - $A(1,2)=A(2,1)$
 - $A([3 \ 1 \ 2],2)=[1 \ 2 \ 3]$
 - $A(1,:)=A(2,:).*A(2,3:-1:1)$