

INTRODUZIONE AL MATLAB

Luisa Fermo

Dipartimento di Matematica e Informatica, Università di Cagliari
fermo@unica.it

Il Matlab

- ▶ **MATLAB significa Matrix Laboratory**



Ogni quantità viene trattata come una matrice.

Uno scalare è una matrice 1×1

- ▶ **Il Matlab è uno strumento scientifico che può essere usato:**

1. come una calcolatrice scientifica;
2. per implementare simulazioni di sistemi complessi;
3. come linguaggio di programmazione.

- ▶ **Manuali Matlab**

1. Consultate il sito www.mathworks.com;
2. Guardate l'Help del Matlab.

E' stata attivata la licenza Campus Matlab

Istruzioni per l'installazione sono disponibili su questa pagina web.
https://www.unica.it/unica/it/studenti_s08_ss09.page

Vi è la possibilità di seguire i corsi online gratuiti associati alla licenza.

[https://trainingenrollment.mathworks.com/
selfEnrollment?code=HGSMWNFEXNCT.](https://trainingenrollment.mathworks.com/selfEnrollment?code=HGSMWNFEXNCT)

Scalari in Matlab

Iniziamo ad usare Matlab come una normale calcolatrice.

- **Dal prompt del Matlab scriviamo, ad esempio $2+2$ e premiamo il tasto enter/invio:**

```
>> 2+2
```

```
ans =
```

```
4
```

← *Il risultato viene memorizzato
in una variabile di default "ans"
abbreviazione di answer*

```
>> ans
```

```
ans =
```

```
4
```

Scalari in Matlab

- **Assegniamo invece ora il valore dell'espressione precedente ad una variabile che chiameremo x**

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

```
x =
```

```
    4
```

oppure digitiamo

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

← *Il ; alla fine dell'espressione
sopprime la visualizzazione a schermo
del risultato*

```
>> x
```

```
x =
```

```
    4
```

Scalari in Matlab

- Una variabile puo' essere comunque sovrascritta

```
>> x=3*2;
```

```
>> x
```

```
x =
```

```
6
```

Alcuni trucchi per risparmiare tempo

- ▶ Durante la sessione di lavoro è possibile richiamare i comandi precedentemente digitati utilizzando i tasti \uparrow , \downarrow , \rightarrow , \leftarrow ;
- ▶ I tasti \uparrow , \downarrow permettono di recuperare delle istruzioni già digitate;
- ▶ I tasti \rightarrow , \leftarrow permettono di riposizionare sulla linea di comando il cursore e di modificare il testo scritto;
- ▶ Con il tasto sinistro del mouse sulla finestra di calcolo si possono selezionare parti di testo che è poi possibile copiare, tagliare ed incollare sulla linea di comando.

Un comando fondamentale da conoscere: la function **diary**

Una function MATLAB molto utile è la function **diary** che permette di salvare una intera sessione di lavoro.

Per creare un file diary che contenga tutti i comandi basterà digitare nella finestra di comando:

```
>> diary Laboratorio1
```

dove in questo caso “Laboratorio” 1 indica il nome del file che conterrà tutte le istruzioni digitate in questa sessione di lavoro.

Il comando

```
>> diary off
```

chiude la sessione di lavoro mentre il comando

```
>> type Laboratorio1
```

permette di visualizzare tutti i comandi memorizzati nel file appena creato.

Esempio

```
>> diary esempiolimit
```

```
>> x=1.2e-08;
```

```
>> f = (1 - cos(x))/x^2
```

```
f = 0.7710
```

```
>> f1 = 1/2 * (sin(x/2)/(x/2))^2
```

```
f1 = 0.5000
```

```
>> diary off
```

Scalari in Matlab

► **Il Matlab contiene delle variabili predefinite**

```
>> pi      ← pi greco
```

```
ans =
```

```
3.1416
```

```
>> eps      ← epsilon di macchina
```

```
ans =
```

```
2.2204e-16
```

Ovviamente anche a queste variabili possiamo assegnare un nome. Ad esempio:

```
>> y=pi;
```

```
>> z=eps;
```

Scalari in Matlab

- ▶ Ora tutte le variabili che avete definito sul vostro computer sono state memorizzate e potete usarle per le normali operazioni:

1. $a = x + y;$

2. $b = z * x;$

3. $c = x - z;$

4. $d = y / x;$

- ▶ La moltiplicazione (e divisione) ha precedenza sulla addizione (e sottrazione) e l'elevamento a potenza ha precedenza su addizione, sottrazione, moltiplicazione e divisione.

» $2 + 3 * 4$

ans =

14

» $(2 + 3) * 4;$

← Si usano le parentesi tonde per alterare l'ordine di precedenza

Esercizi

- ▶ Posto $x = 7$ e $y = 4$ calcolare $\left(\frac{2}{x+y}\right)^5$
- ▶ Se $z = 3$ e $w = 8$ calcolare $\frac{3z-w}{2+\sqrt{8w}}$
- ▶ Se $a = \frac{1}{3}$ e $b = \frac{1}{5}$ calcolare $\sqrt{\frac{a^{-3}}{(1-b+3a)}}$
- ▶ Se $c = -4$ calcolare \sqrt{c}

NOTA **i** e **j** (unità immaginarie)

>> 3+2i

>> 7+5j

Scalari in Matlab

- Le variabili memorizzate in questo ambiente di lavoro sono visibili nel workspace o spazio di lavoro. Tuttavia il comando *whos* or *who* elenca tutte le variabili che sono state memorizzate:

```
>> who
```

Your variables are:

```
ans x y z
```

```
>> whos
```

name	Size	Bytes	Class	Attributes
ans	1x1	8	double	
x	1x1	8	double	
y	1x1	8	double	
z	1x1	8	double	

Scalari in Matlab

- ▶ **Le variabili possono essere anche cancellate. Ad esempio se vogliamo cancellare la variabile x basta scrivere**

» clear x ← Osservate adesso il workspace
usate il comando **whos** oppure **who**
per vedere le variabili disponibili

- ▶ **Se invece volete cancellare tutte le variabili digitate il comando**

» clear all ← A questo punto il workspace sarà vuoto
e se provate ad esempio a scrivere sul
prompt x vi apparirà il messaggio di errore
??? Undefined function or variable 'x'.

- ▶ **Per salvare tutte e solo alcune variabili x , y e z della sessione di lavoro in un file (con estensione.mat) di nome *nomefile* si utilizza il comando `save`**
 `>> save nomefile`
 oppure
 `>> save nomefile x y z`
- ▶ **Se si digita solo il comando `save` tutte le variabili poste in memoria vengono salvate nel file `matlab.mat`.**
- ▶ **Per ricaricare le variabili salvate nel file e renderle attive in memoria si usa il comando `load`**
 `>> load nomefile`

Scalari in Matlab

- ▶ **Tutti i calcoli vengono effettuati in doppia precisione, mentre diversa è la visualizzazione delle variabili che viene determinata con il comando format:**
 - ▶ **format short:** virgola fissa con 5 cifre (è il formato di default);
>>> pi
ans = 3.1416
 - ▶ **format long:** virgola fissa con 15 cifre:
>>> pi
ans = 3.14159265358979
 - ▶ **format short e:** virgola mobile con 5 cifre:
>>> pi
ans = 3.1416e+00
 - ▶ **format long e:** virgola mobile con 15 cifre:
>>> pi
ans = 3.141592653589793e+00

Il comando tic e toc

Per valutare l'efficienza di un programma in termini di tempo d'esecuzione espresso in secondi, si possono utilizzare i comandi **tic** e **toc**.

Essi consentono di conoscere il numero dei secondi richiesti da un determinato calcolo e si utilizzano secondo la seguente sintassi

```
>> tic  
calcolo;  
>> toc
```

tic attiva il timer, toc lo arresta e restituisce l' "elapsed time", ovvero il tempo (in secondi) trascorso dal momento in cui tic è stato attivato.

Vettori in Matlab: sintassi generale

- ▶ **Per scrivere un vettore riga è sufficiente scrivere le componenti fra parentesi quadre separati da spazi o da virgole:**

```
>> w=[3 4 5];
```

oppure

```
>> w=[3,4,5];
```

- ▶ **Per scrivere un vettore colonna è sufficiente scrivere le componenti fra parentesi quadre separati da punto e virgola:**

```
>> w=[3;4;5];
```

Vettori in Matlab: componenti

- ▶ Per identificare le componenti di un vettore, ad esempio la seconda o la terza, si usa la seguente sintassi :
 >> w(2)
 oppure
 >> w(3)
- ▶ Per identificare l'ultima componente del vettore si può anche usare il comando
 >> w(end)
- ▶ L'indicizzazione inizia da 1. La componente w(0) non esiste!!!!

Vettori in Matlab: dimensioni

- ▶ **Per conoscere la lunghezza del vettore usate il comando**
length:
 >> length(w)
 ans =
 3
- ▶ **Per conoscere le dimensioni della variabile usate il comando** size
 >> size(w)
 ans =
 1 3

Vettori in Matlab: esempi di vettori speciali

- ▶ **Per costruire un vettore di componenti tutte nulle si può usare il comando `zeros`:**
 - » `u=zeros(1,7);`
 - » `u=zeros(7,1);`
- ▶ **Per costruire un vettore di componenti tutte uguali a 1 si può usare il comando `ones`**
 - » `z=ones(1,8);`
 - » `z=ones(8,1);`

Vettori in Matlab: esempi di vettori speciali

► Il comando

```
>> v=[1:10]
```

1 2 3 4 5 6 7 8 9 10

genera un vettore riga di lunghezza 10 di componenti 1,2,3,...,10.

► Il comando $v=[c:h:d]$ genera un vettore riga la cui componente iniziale ha valore c e le successive si incrementano di h fino ad arrivare a d . Ad esempio:

```
>> v=[0:.5:3]
```

0 0.5000 1.0000 1.5000 2.0000 2.5000 3.0000

Vettori in Matlab: esempi di vettori speciali

- ▶ Il comando `linspace(a,b,n)` genera un vettore riga di lunghezza `n` le cui componenti sono `n` punti equispaziati tra `a` e `b` (estremi compresi). Ad esempio:

```
>> v=linspace(0,5,4)
```

```
v =
```

```
    0    1.6667    3.3333    5.0000
```

- ▶ Il comando `rand(1,n)` (`rand(n,1)`) genera un vettore di riga (colonna) di lunghezza `n` aventi componenti casuali.

Vettori in Matlab: operazioni

- Supponiamo di avere due vettori v e w di lunghezza n .

Ad esempio:

```
>> v=rand(1,5);
```

```
>> w=rand(5,1);
```

Possiamo calcolare

1. i loro trasposti ad esempio

```
>> u=v';
```

2. i loro moduli

```
>> nu=norm(u);
```

3. il prodotto scalare

```
>> p=v*w;
```

oppure

```
>> p=dot(v,w);
```

4. la loro somma

```
>> s=v+w';
```


Vettori in Matlab: operazioni

- ▶ **Dati due vettori della stessa lunghezza possiamo anche effettuare delle operazioni "componente per componente", usando la sintassi con il punto.**

- 1. Prodotto componente per componente**

```
» v=rand(1,5);  
» w=rand(1,5);  
p1=v.*w;
```

- 2. elevamento a potenza componente per componente**

```
» pot=v.^2;
```

Vettori in Matlab: operazione

- Possiamo, inoltre, concatenare due vettori. Ad esempio:

```
>> u=[1 2 3];
```

```
>> v=[3 4];
```

```
>> w=[u,v]
```

```
w
```

```
1 2 3 3 4
```

- Possiamo sostituire delle componenti. Ad esempio supponiamo di voler cambiare le ultime due componenti di $u=[1\ 2\ 3]$ con 4 5. Allora

```
>> r=[4 5];
```

```
>> u(2:3)=r;
```

```
>> r
```

```
r=
```

```
1 4 5
```

Matrici in Matlab: sintassi generale

Supponiamo di voler scrivere le seguenti matrici

$$A = \begin{pmatrix} 1 & 2 & 4 \\ 5 & 7 & 7 \\ 4 & 1 & 2 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 4 \\ 5 & 2 \\ 4 & 0 \end{pmatrix}.$$

La sintassi da usare in Matlab é la seguente

```
>> A=[1 2 4;5 7 7;4 1 2];
```

```
>> B=[1 4;5 2;4 0];
```

Matrici in Matlab: componenti

- ▶ **Per identificare un elemento della matrice, ad esempio quello che si trova nella prima riga e seconda colonna**
 $\gg A(1,2)$
- ▶ **Per identificare una intera riga, ad esempio la seconda**
 $\gg A(2,:)$
- ▶ **Per identificare una intera colonna, ad esempio la prima**
 $\gg A(:,1)$
- ▶ **Per estrarre un vettore ad esempio quello costituito dalle ultime due componenti**
 $\gg A(3,2:3)$
- ▶ **Per estrarre un blocco della matrice, ad esempio il primo blocco quadrato**
 $\gg A(1:2,1:2)$

Matrici in Matlab: dimensioni

- ▶ Il comando `size` ci permette di vedere le dimensioni della matrice

```
>> size(A)
```

```
ans=
```

```
    3    3
```

```
>> size(B)
```

```
ans=
```

```
    3    2
```

Matrici in Matlab: esempi di matrici speciali

- ▶ **matrici nulle di dimensioni $n \times m$**
 >> zeros(n,m)
- ▶ **matrici di n righe ed m colonne di componenti pari a uno**
 >> ones(n,m)
- ▶ **matrice identità di dimensione n**
 >> eye(n)
- ▶ **matrice di n righe ed m colonne di componenti casuali**
 >> rand(n,m)

Matrici in Matlab: operazioni

Supponiamo di avere

$$A = \begin{pmatrix} 1 & 2 & 4 \\ 5 & 7 & 7 \\ 4 & 1 & 2 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 4 \\ 5 & 2 \\ 4 & 0 \end{pmatrix}.$$

- ▶ **Possiamo sommarle**
 $\gg S=A+B;$ ATTENZIONE ALLE DIMENSIONI!!!!
- ▶ **Possiamo farne il prodotto**
 $\gg P=A*B;$
- ▶ **Possiamo farne le sue potenze, ad esempio**
 $\gg C = B^2;$
- ▶ **Possiamo calcolarne la sua inversa**
 $\gg I=\text{inv}(A);$

Matrici in Matlab: operazioni

- ▶ **Date due matrici delle stesse dimensioni possiamo anche effettuare delle operazioni "componente per componente", usando la sintassi con il punto, proprio come i vettori**
 - 1. Prodotto componente per componente**
 $\gg P1=A.*B;$ ATTENTI ALLE DIMENSIONI !!
 - 2. Elevamento a potenza componente per componente**
 $\gg P2 = A.^2;$

Programmare in Matlab: generalità

- ▶ All'interno di un file Matlab si possono inserire delle righe di commento utilizzando il comando % prima del commento stesso
- ▶ Gli m-files possono essere di due tipi:
 1. Script: sono definiti semplicemente da una serie di comandi. Per eseguire un file di questo tipo è necessario digitare il nome del file nel prompt del Matlab;
 2. Function: Sono degli m-file la cui prima riga deve essere necessariamente
function [x,y]=nomefunction(a,b)
Per poter eseguire un file di questo tipo è necessario copiare questa prima riga all'interno del prompt, ad eccezione della parola function.

Programmare in Matlab: cicli for e while

Un ciclo o loop è una struttura che consente di ripetere una istruzione più volte. Ogni ripetizione del ciclo si chiama passaggio. Il Matlab dispone di due cicli:

1. Ciclo **for** utilizzato quando il numero dei passaggi è noto a priori.
2. Ciclo **while** utilizzato quando il numero dei passaggi non è noto in anticipo e termina quando una certa condizione è soddisfatta.

Programmare in Matlab: istruzione di controllo

Molte volte succede che determinate operazioni si devono fare solo se una determinata istruzione è verificata. Per fare ciò in Matlab si adopera la seguente istruzione di controllo:

```
if   istruzione
    comandi
elseif istruzione
    comandi
else  comandi
end
```

Operatori relazionali

Il Matlab dispone di sei operatori relazionali che consentono di confrontare variabili ed array. Questi operatori sono i seguenti:

< minore

> maggiore

<= minore o uguale

>= maggiore o uguale

== uguale

~= diverso

Operatori relazionali

**Supponiamo di voler effettuare un confronto tra due dati.
Come si procede?**

1. Si scrivono i dati che si vogliono confrontare. Ad esempio:
 $\gg x=2; y=5;$
2. Si effettua il confronto. Ad esempio, se vogliamo sapere se $x < y$ scriviamo:
 $\gg x < y$
3. A questo punto il Matlab restituirà zero se il confronto è falso oppure uno se il confronto è vero.

N.B. Gli operatori aritmetici $+, *, -, /$ hanno precedenza rispetto agli operatori relazionali. Quindi se ad esempio scriviamo $5 < 2 + 7$ il Matlab esegue prima l'addizione e poi il confronto.

Grafica in Matlab

Supponiamo di voler rappresentare graficamente la funzione

$$y(x) = \sin(x), \quad x \in [0, 2\pi].$$

Possiamo procedere in due modi:

- ▶ **Usare il comando `fplot`**
 \gg `fplot('sin(x)', [0 2*pi])`
- ▶ **Usare il comando `plot`**
 \gg `x=linspace(0,2*pi,100);`
 \gg `y=sin(x);`
 \gg `plot(x,y)`

Grafica in Matlab

Il precedente grafico, indipendentemente dal comando usato, puo' essere arricchito. Ad esempio:

- ▶ **etichettando gli assi attraverso i comandi `xlabel` oppure `ylabel`**
 >> `xlabel('x')`
 >> `ylabel('y')`
- ▶ **dando un titolo al grafico con il comando `title`**
 >> `title('Grafico di sin(x)')`
- ▶ **scrivendo all'interno del grafico stesso mediante il comando `gtext`**
 >> `gtext('sin(x)')`