

Introduzione all'ambiente

MATLAB[©]

Utilizzo di Matlab ed operazioni fondamentali.

Primi rudimenti di grafica.

Lezione 1

Paola Gervasio

1

Ha **funzioni intrinseche** molto potenti (es: risoluzione di sistemi lineari, calcolo di autovalori e autovettori di una matrice) ed esistono **toolbox** (librerie di software specifico -file scritti in linguaggio matlab-):

- Control System
- Signal Processing
- Statistics
- Neural Networks
- Fuzzy Logic
- Communications
- ...

Prompt di Matlab:

>>

MATLAB = MATrix LABoratory

È un sistema interattivo in cui l'unità base dei dati è un **array** (es: vettore=array a 1 indice, matrice=array a 2 indici), per il quale **non** è chiesto il **dimensionamento**.


- È un interprete di comandi
- È un linguaggio di programmazione
- Ha una buona potenzialità grafica
- Versioni per Unix/Linux, Windows, Mac.
- I files scritti in Matlab sono portabili da una piattaforma all'altra.

2

Assegnazione di variabili scalari

>>a=1.54

- **a** nome della variabile (max 31 caratteri alfanumerici, il primo dei quali non deve essere un numero)
 - 1.54 valore numerico assegnato alla variabile.
- Il comando

>> a=1.54  produce

a = 1.5400

>> a=1.54;  non produce risposta

3

4

Operazioni aritmetiche

`^` potenza
`*` prodotto
`/` divisione
`+` somma
`-` differenza

Es: per calcolare $x = \frac{3 + 5^3 - 2/3}{4(5 + 2^4)}$

>> x=(3+5^3-2/3)/(4*(5+2^4))

- Sono osservate le precedenze classiche dell'aritmetica
- Per alterare le precedenze si utilizzano esclusivamente le parentesi **tonde**

6

Formato di memorizzazione dei numeri

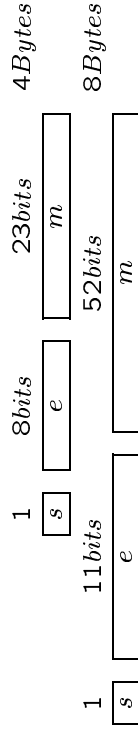
- **Singola** (o semplice) precisione, 4 Bytes
- **Doppia** precisione, 8 Bytes

Come vengono utilizzati questi Bytes?

Si considera la forma esponenziale di un numero reale:

$$x = 123456.789 = (-1)^0 1.23456789 \cdot 10^5 = (-1)^s m \cdot \beta^e$$

$s = 0, 1$; m mantissa; β base (es: 2,10); e esponente



8

>> 1.67 produce

ans =
1.6700

ans è il nome della variabile di default.

>> a produce

a = 1.5400
per visualizzare il contenuto della variabile a

>> b=1+1/2+5/3+1/4+23/6+...
2/9+1/10;
per poter spezzare un'istruzione troppo lunga

5

>> whos

Name	Size	Bytes	Class
a	1x1	8	double array
ans	1x1	8	double array
b	1x1	8	double array
x	1x1	8	double array

Di default, Matlab lavora con variabili in **doppia precisione**.

Ogni numero memorizzato in doppia precisione occupa 8 Bytes.

Le variabili scalari sono viste come **array** di dimensione 1x1 (una riga e una colonna).

OSS. Di default lettere maiuscole e minuscole sono considerate diverse sia nei comandi che nei nomi delle variabili.

7

Formato di rappresentazione dei numeri

```
>> c=0.456723
c =
    0.4567
Il numero è stato rappresentato con 5 cifre

>> format short e
>> c
c =
    4.5672e-01
Forma esponenziale con 5 cifre per la mantissa

>> format long e
>> c
c =
    4.5672300000000000e-01
Forma esponenziale con 16 cifre per la mantissa

>> format long
>> c
c =
    0.4567230000000000
Il numero è rappresentato con 15 cifre
```

9

Il contenuto di queste variabili può essere variato con una semplice operazione di assegnazione:

```
>> pi=18
pi =
    18
```

Per riassegnare alla variabile `pi` il valore π :

```
>> clear pi
>> pi
ans =
    3.1416
```

Per cancellare il contenuto della variabile `a`:

```
>> clear a
```

Per cancellare il contenuto di tutte le variabili:

```
>> clear
```

11

Di default Matlab utilizza il formato `format short`. Per tornare a questo formato di rappresentazione:

```
>> format short
```

N.B. Il formato di rappresentazione può cambiare, ma il formato di memorizzazione dei numeri è sempre lo stesso (8Bytes).

Variabili predefinite

```
pi       $\pi$ 
i, j     $\sqrt{-1}$  unità immaginaria
NaN     not a number
eps     2.2204e-16 precisione di macchina
```

10

Assegnazione di array

```
>> a=[1 2 3 4];
>> a=[1,2,3,4];
>> a=(1:4);
```

Modi equivalenti per generare un array 1x4, 1 riga e 4 colonne, vettore riga

```
>> a
a =
    1    2    3    4

>> b=[1;2;3;4]
b =
    1
    2
    3
    4
```

Per generare un array 4x1, 4 righe e 1 colonna, vettore colonna

12

```
>> c=[5 3 4; 2 4 -2]
c =
     5     3     4
     2     4    -2
```

Per generare un array 2x3, matrice 2 righe e 3 colonne

- Lo spazio o la virgola separano elementi sulla stessa riga. Il punto e virgola separa le righe.

Operazione di trasposizione:

```
>> a'
ans =
     1
     2
     3
     4
```

Il vettore trasposto di a viene memorizzato nella variabile ans

```
>> a1=a'
```

Il vettore trasposto di a viene memorizzato nella variabile a1

13

```
>> a(2)
ans =
     2
```

Per accedere ad un elemento di un vettore

```
>> c(2,1)
ans =
     2
```

Per accedere ad un elemento di una matrice

```
>> d=c(1,:)
d =
     5     3    18
```

Per estrarre la prima riga di una matrice

```
>> e=c(:,1:2)
e =
     5     3
     2     4
```

Per estrarre le prime due colonne di una matrice

15

Analogo discorso vale per la trasposizione di matrici:

```
>> c1=c'
c1 =
     5     2
     3     4
     4    -2
```

```
>> whos
Name      Size      Bytes  Class
a         1x4         32  double array
ans       4x1         32  double array
b         4x1         32  double array
c         2x3         48  double array
c1        3x2         48  double array
```

Grand total is 24 elements using 192 bytes

14

```
>> b(3)=5
```

```
b =
     1
     2
     5
     4
```

Per modificare un elemento di un vettore. Se non si utilizza il ";" viene visualizzato l'array completo

```
>> c(1,3)=18
c =
     5     3    18
     2     4    -2
```

Per modificare un elemento di una matrice.

16

Operazioni su array

- + somma di vettori o matrici (elemento per elemento)
- differenza di vettori o matrici (elemento per elemento)
- * prodotto tra vettori e/o matrici (righe per colonne)

Sono le operazioni dell'algebra lineare; quindi:

- per somma e differenza: gli operandi devono avere le stesse dimensioni
- per il prodotto: la dimensione interna dei due array deve coincidere.

```
>> a1+b          entrambi vettori colonna 4x1
```

```
ans =
```

```
2
4
8
8
```

17

Esistono poi le operazioni "punto" che agiscono su array che abbiano le stesse dimensioni:

- .* prodotto elemento per elemento
- ./ divisione elemento per elemento
- .^ potenza elemento per elemento

```
>> a1b=a1.*b      (a1b)i = (a1)i * bi
```

```
a1b =
1
4
15
16
```

con $a1 = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$ e $b = \begin{bmatrix} 1 \\ 2 \\ 5 \\ 4 \end{bmatrix}$

19

```
>> a-b
??? Error using ==> -
Matrix dimensions must agree.
```

a =vettore riga
(1x4)
b =vettore colonna
(4x1)

```
>> a*b
```

```
ans =
36
```

(1x4)(4x1) -prodotto scalare-

```
>> c*d'
```

```
ans =
358
-14
```

(2x3)(3x1) -prodotto matrice
vettore-

```
>> d*c
```

```
??? Error using ==> *
Inner matrix dimensions must agree.
```

(3x1)(2x3) -
prodotto non
possibile-

18

Funzioni matematiche e grafica

```
>> f='(2*x-sqrt(2))^2*sin(2*x)'
```

```
f =
```

```
(2*x-sqrt(2))^2*sin(2*x)
```

```
>> whos
```

Name	Size	Bytes	Class
f	1x24	48	char array

f è una stringa di 24 caratteri, Matlab la interpreta come un vettore riga di 24 elementi (di tipo carattere). Ogni elemento di un char array occupa 2 Bytes.

Per valutare f in un punto:

```
>> x=1.718; y=eval(f)
```

N.B. Se f è stata definita sulla variabile x, nella fase di valutazione (eval), MATLAB si serve sempre e solo del contenuto della variabile x.

20

Problema 1: valutare $f(x) = x^2 \cos(x)$ sull'intervallo $I = [-1, 2]$ e rappresentarla graficamente.

1) Definire una griglia sull'intervallo $I = [-1, 2]$, ovvero scegliere un insieme discreto di punti rappresentativo per I :

```
>> x=linspace(-1,2,50);
```

Crea un vettore riga di 50 elementi, contenente i valori di 50 punti equispaziati in I

2) Definire la funzione e valutarla:

```
>> f='x.^2.*cos(x)';
```

x è un vettore, si vuole calcolare $y_i = x_i^2 \cos(x_i)$ per ogni i , quindi si devono usare le operazioni "."

```
>> y=eval(f);
```

3) Rappresentare i punti (x_i, y_i) su di un piano cartesiano:

```
>> plot(x,y)
```

21

Per conoscere nel dettaglio tutte le opzioni di un comando, oppure se non ci si ricorda la sintassi del comando:

```
help nome_comando
```

```
>> help plot
```

Se non ci si ricorda il nome del comando, ma si vuole fare una ricerca per *parola_chiave* (in inglese), oppure se si cercano tutti i comandi che facciano riferimento ad una *parola_chiave*:

```
lookfor parola_chiave
```

```
>> lookfor plot
```

23

La sintassi del comando plot è:

```
plot(x,y, 'color_linestyle_marker')
```

```
>> plot(x,y,'m-*)
```

color: c,m,y,r,b,g,w,k

linestyle: -,--,:-.,none

marker: +,o,*,.x,s

Per disegnare 2 o più coppie di vettori sullo stesso grafico:

```
>> g='sin(x).*exp(x)';
```

```
>> yg=eval(g);
```

```
>> plot(x,y,'b:',x,yg,'r-');
```

22

Funzioni matematiche intrinseche

```
sqrt(x)
```

\sqrt{x}

```
round(x)
```

arrotondamento: round(3.6)=4

```
fix(x)
```

parte intera del numero: fix(3.6)=3

```
sign(x)
```

segno di x (vale -1, 0 o 1)

```
sin(x), cos(x), tan(x)
```

$\sin(x)$, $\cos(x)$, $\tan(x)$

```
sinh(x), cosh(x), tanh(x)
```

$\sinh(x)$, $\cosh(x)$, $\tanh(x)$

```
asin(x), acos(x), atan(x)
```

$\arcsin(x)$, $\arccos(x)$, $\arctan(x)$

```
exp(x), log(x), log10(x)
```

e^x , $\log_e(x)$, $\log_{10}(x)$

Per z complesso:

```
>> z=3+i*4
```

```
real(z)
```

parte reale di z

```
imag(z)
```

parte immaginaria di z

```
conj(z)
```

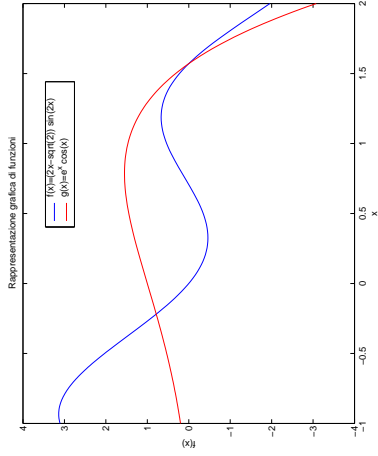
complesso coniugato di z

24

Generazione di un file MATLAB

Problema 2: Creazione di un grafico 2D.

Disegnare $f(x) = (2x - \sqrt{2})\sin(2x)$ e $g(x) = e^x \cos(x)$ sull'intervallo $I = [-1, 2]$.



25

Per salvare il contenuto del file: dal menù dell'Editor selezionare **File**, **Save as**. Specificare il direttorio in cui salvare (es: c:\tmp o a:\) ed il nome per il file (es: dis2d.m).

N.B. L'estensione dei file matlab è sempre **m**.

Dalla finestra dei comandi matlab:

```
>> addpath c:\tmp           oppure  
>> addpath a:\
```

per dire di cercare il file in tale direttorio, quindi richiamare il file generato, dando il nome del file stesso:

```
>> dis2d
```

Dal menù della finestra di Matlab, selezionare **File**, poi **New** e poi **M-file**. Si apre una finestra dell'**Editor/Debug**. Si scrivono i comandi matlab.

```
clf;  
f='(2*x-sqrt(2))*sin(2*x)';  
fplot(f,[-1,2])  
xlabel('x'); ylabel('f(x)')  
title('Rappresentazione grafica di funzioni')  
hold on  
g='exp(x)*cos(x)';  
fplot(g,[-1,2], 'r')  
legend('f(x)=(2x-sqrt(2)) sin(2x)', 'g(x)=e^x cos(x)')  
hold off
```

26

Matlab segnala errori?

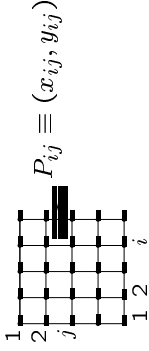
- 1) Leggere il tipo di errore
- 2) Tornare nell'editor, cercare l'errore e modificare il file
- 3) Salvare le modifiche effettuate
- 4) Tornare alla finestra dei comandi Matlab e ridare il comando

```
>> dis2d
```

28

Grafici 3D

Problema: Rappresentare graficamente $f(x, y) = xe^{-(x^2+y^2)}$ sul dominio $\Omega = [-2, 2]^2$.



Anzitutto bisogna definire una griglia su Ω .

```
>> [x,y]=meshgrid(-2: .1:2, -2: .1:2);      x e y sono due ma-
>>                                     trici.
>> clf      Per pulire la figura precedente
>> f='x.*exp(-x.^2-y.^2)';
>> z=eval(f); surf(x,y,z); colorbar
```

29

Per creare più figure, basta anteporre al comando di disegno l'istruzione `figure(k)` dove `k` è un numero intero positivo di una figura non attiva.
Es.:

```
>> mesh(x,y,z);
>> figure(2); surf(x,y,z,gradient(z));
>> figure(3); plot3(x,y,z);
```

Per passare il comando da una finestra all'altra, al fine di modificare il grafico:

```
>> figure(2)
>> colorbar
```

31

Altri comandi di grafica 3D:

```
>> mesh(x,y,z)          Superficie
>> meshc(x,y,z)        Superficie e countour-lines
>> surfc(x,y,z)        Superficie e countour-lines
>> pcolor(x,y,z)       Superficie colorata piatta
>> surf(x,y,z,gradient(z)) Superficie colorata secondo
                           la grandezza di  $\partial z/\partial x$ 
>> contour(x,y,z)      Contour-lines (linee di livello)
>> plot3(x,y,z)        Linee lungo la direzione y
                           serve anche per disegnare linee in 3D
```

30

Se si vuole una sola finestra con più grafici:

```
>> figure(1)
>> subplot(2,2,1); mesh(x,y,z);
>> title('mesh')
>> subplot(2,2,2); surfc(x,y,z);
>> title('surfc')
>> subplot(2,2,3); plot3(x,y,z);
>> title('plot3')
>> subplot(2,2,4); surf(x,y,z,gradient(z));
>> title('surf,gradient')
```

32

Salvataggio e Stampa di una figura

Per salvare la figura realizzata in formato matlab:

Dal Menù della finestra grafica scegliere **File, Save as;** quindi scegliere il direttorio e il nome con estensione **.fig.**
Per riaprire la figura, sempre dalla finestra grafica, scegliere **File, Open** e selezionare il nome del file.

Per salvare la figura in formato jpeg dal Menù della finestra grafica si sceglie **File, Export**, si seleziona l'estensione JPEG **.jpg** si specifica il nome del file.
Altri formati: **.ps, .eps, .tiff, .png,**

33

Generazione di un filmino

Disegnare $f(x, y, t) = \frac{1}{5} \sin(x)y \cos(t)$
per $(x, y) \in [-\pi, \pi]^2$ e $t \in [0, 2\pi]$

```
[x,y]=meshgrid(-pi:.5:pi);  
f='sin(x).*y/5*cos(t)';  
nframes=20;  
tt=linspace(0,2*pi,nframes);  
figure(1); clf  
Mv=moviein(nframes);  
for n=1:nframes  
    t=tt(n); z=eval(f); surf(x,y,z);  
    axis([-pi pi -pi -1 1]);  
    Mv(:,n) = getframe;  
end  
movie(Mv,4);
```

35

Disegno di una superficie attraverso le equazioni parametriche

$$\gamma(r, \theta) = \begin{pmatrix} r \cos(\theta) & r \sin(\theta) & \theta \end{pmatrix} \begin{matrix} x & y & z \end{matrix}$$

per $r \in [0, 2]$ e $\theta \in [0, 6\pi]$.

```
[r,theta]=meshgrid(0:.1:2,0:.1:6*pi);  
x=r.*cos(theta);  
y=r.*sin(theta);  
z=theta;  
surf(x,y,z)
```

34