

MATLAB - Introduzione

Enrico Nobile



I^a Parte

1

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

- **MATLAB (MATrix LABoratory);**
- **Versione 5.2;**
- **Basato sul concetto - ampio - di Matrici;**
- **Calcolo numerico, visualizzazione etc.**
- **Facile utilizzo di librerie per il calcolo numerico (LINPACK, EISPACK ...);**
- **Toolboxes: orientati a tipologie definite di problemi;**
- **Estensibilità.**

2

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Documentazione Vers. 5.2 (1):

- *Using MATLAB;*
- *Using MATLAB Graphics;*
- *The MATLAB Application Program Interface Guide;*
- *MATLAB Installation Guide;*
- *Getting Started with MATLAB;*
- *GUIs with MATLAB*
- *MATLAB Notebook User's Guide;*
- *Libri (generali e specifici).*

3

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Documentazione (2):

- **Internet:**
 - **WWW:** <http://www.mathworks.com/>
 - **news:** comp.soft-sys.matlab
 - **Anonymous ftp:** [ftp.mathworks.com](ftp://ftp.mathworks.com)

4

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

MATLAB è costituito da 5 parti principali:

- *MATLAB Language;*
- *MATLAB Working Environment;*
- *Handle Graphics;*
- *MATLAB Mathematical function library;*
- *MATLAB Application Program Interface (API).*

5

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Disponibilità MATLAB:

- PCs (Windows XX, Linux), MACs;
- Workstations (es. SUN);
- Mainframe/servers (es. Digital ALPHA);

Versioni : 5.2 (Differisce in modo significativo dalle versioni della serie 4).

Prompt MATLAB:

>>

6

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Documentazione ed informazioni on-line:

- Comando `help`;
- Comando `lookfor`;
- Help window;
- Help desk;
- Comando `doc`;
- Documentazione in formato pdf;
- Mathworks Web site.

7

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Comandi `help` e `lookfor`:

`»help soggetto`

`»lookfor argomento`

Esempi:

`»help`

`»help help`

`»lookfor singular`

8

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

MATRICI

Oggetto: matrice rettangolare:

- scalare: matrice 1x1;
- vettore: matrice riga o matrice colonna;
- introduzione matrici:
 - lista di elementi;
 - generate da istruzioni *built-in* e funzioni;
 - create via M-files;
 - da files dati.

9

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

I tre statements:

```
» A = [1 2 3; 4 5 6; 7 8 9]
```

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

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

sono equivalenti.

10

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Se il comando non sta su una riga, usare tre punti (...) e poi Return:

```
» a = [1 2 3 4 5 6 7 ...  
      8 9 10]
```

Command line editing.

11

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Le funzioni built-in

`rand`, `magic` e `hilb`

aiutano a creare matrici con le quali sperimentare le funzioni di MATLAB.

Per manipolare (lettura/scrittura) matrici di grandi dimensioni, usare M-files.

12

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

- Gli elementi di una matrice (e di un vettore) sono individuati tramite indici:
 $A(2,3)$: 2^a riga, 3^a colonna di A;
 $x(3)$: 3^o elemento del vettore x;
- MATLAB è *case-sensitive*.

13

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

MATLAB gestisce le matrici in modo *dinamico*:

Esempio:

```
» a = magic(4)
```

```
» a(4,5) = 21
```

La dimensione della matrice viene aggiornata automaticamente.

Eliminazione di righe e colonne:

```
» a = magic(5)
```

```
» a(:,2:4) = []
```

14

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

MATLAB Statements e Variabili:

```
variable = expression
```

oppure

```
expression
```

Esempi:

```
» a = 120/4
```

```
» 120/4
```

15

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Per uscire da MATLAB:

```
»quit oppure »exit
```

Per salvare il Workspace:

```
»save
```

salva tutte le variabili in un file `matlab.mat`;

```
»save pippo
```

salva tutte le variabili in un file di nome `pippo.mat`;

```
»save pippo A
```

salva solo la variabile `A` nel file `pippo.mat`.

16

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Il comando

```
»load pippo
```

legge tutte le variabili precedentemente salvate nel file `pippo.mat`;

I comandi `save` e `load` possono anche esportare ed importare files ASCII.

17

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Esempio di lettura di un file ASCII, e successiva rappresentazione grafica:

```
»load tracce.dat
```

```
»plot(tracce(:,1),tracce(:,2))
```

18

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Operazioni fra matrici:

- + addizione
- sottrazione
- * moltiplicazione
- ^ elevamento a potenza
- \ trasposizione
- \ divisione a sinistra
- / divisione a destra
- () specifica ordine di valutazione

19

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Divisione matriciale:

Se A è una matrice quadrata invertibile e B è un vettore (risp. colonna o riga):

» $X=A \setminus B$ è soluzione di $A * X = B$

» $X=B / A$ è soluzione di $X * A = B$

La *divisione a destra* è definita in termini di *divisione a sinistra*:

$$B/A = (A' \setminus B)'$$

20

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

La matrice A non deve essere necessariamente quadrata.

Se A è una matrice $m \times n$, vi sono 3 casi:

- 4 $m = n$ Ricerca la soluzione esatta;
- 4 $m > n$ Sistema *sovradeterminato*: cerca una soluzione ai *minimi quadrati*;
- 4 $m < n$ Sistema *sottodeterminato*: ricerca una soluzione fondamentale con al più m componenti non nulle.

21

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

L'operatore `\` (backslash) impiega algoritmi diversi nei vari casi, che sono diagnosticati automaticamente da MATLAB, sulla base della matrice dei coefficienti:

- o *Permutazioni di matrici triangolari*;
- o *Matrici simmetriche definite positive*;
- o *Matrici quadrate non-singolari*;
- o *Matrici rettangolari per sistemi sovradeterminati*;
- o *Matrici rettangolari per sistemi sottodeterminati*.

22

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Una matrice quadrata A è detta **singolare** se le sue colonne non sono linearmente indipendenti ($\det(A) = 0$).

L'operatore \ segnala potenziali problemi se A è "quasi singolare", ed un errore se A è singolare.

Le capacità di MATLAB nel risolvere sistemi di equazioni lineari sono basate - per la gran parte - su tre fattorizzazioni:

- Cholesky per matr. simmetriche def. Positive (`chol`);
- Gauss per matr. Quadrate generiche (`lu`);
- Ortogonalizzazione per matrici rettangolari (`qr`).

23

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

AUTOVALORI ED AUTOVETTORI

Un autovalore λ ed un autovettore v di una matrice A soddisfano la:

$$Av = \lambda v$$

Se Λ è la matrice (diagonale) degli autovalori e V la matrice le cui colonne costituiscono tutti gli autovettori di A , allora:

$$AV = V\Lambda$$

Se, inoltre, V non è singolare:

$$A = V\Lambda V^{-1}$$

è detta decomposizione agli autovalori (*eigenvalue decomposition*)

24

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Le operazioni di *addizione e sottrazione* operano *fra gli elementi (element-wise)* delle matrici, ma le altre operazioni NO !

Queste (^,*,\,/) possono venire fatte operare element-wise precedendole con un punto (*period*):

Esempio:

» [1,2,3,4].*[1,2,3,4] = [1,4,9,16]

» [1,2,3,4]*[1,2,3,4]' = 30

25

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Elenca variabili nel Workspace:

» who;whos

Elimina variabile dal Workspace:

» clear nome_variabile

Elimina tutte le variabili dal Workspace (!!!):

» clear

Precisione della macchina:

» eps

Interruzione calcolo (PC): CTRL-BREAK

26

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Funzioni per la generazione di matrici:

```
eye      zeros    ones      diag
triu     tril     rand      hilb
magic    toeplitz
```

Le matrici possono venire costruite da blocchi; se A è una matrice 3×3 :

```
» B=[A,zeros(3,2);zeros(2,3),eye(2)]
```

produce una matrice 5×5 (provare...)

27

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Altre strutture dati

- Matrici (arrays) multidimensionali;
- Cell arrays;
- Caratteri e testo;
- Strutture.

Esempio di matrice 3D:

```
» a = rand(3,3,5)
```

crea una matrice $3 \times 3 \times 5$.

Le matrici 3D - o anche di ordine superiore - sono molto utili (es. campo di temperatura in un dominio 3D).

28

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Cell arrays

Sono arrays multidimensionali che contengono *copie* di altri arrays (no puntatori !). Esempio:

```
» a = magic(3) + 1;  
» c = {a det(a) inv(a)}
```

Attenzione: mentre le matrici multidimensionali possono contenere solo matrici della stessa dimensione, le cell arrays possono venire utilizzate per memorizzare (*ed accedere a*) matrici di dimensioni diverse.

29

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Strutture

Le strutture sono arrays multidimensionali, ai cui elementi si accede attraverso "field designators".

Esempio:

```
» s.nome = 'Marco';  
» s.cognome = 'Rossi';  
» s.corso = 'Mecc';  
» s.anno = 3
```

30

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Strutture

Anche un intero elemento può venire aggiunto con un solo comando:

```
» s(2) = struct('nome', 'Paolo',...  
    'cognome', 'Bianchi', 'corso', 'Mat',...  
    'anno', 4)
```

Vedere il risultato dei comandi:

```
» s.corso  
» [s.corso]  
» char(s.nome)
```

31

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Altre capacità (di base) di MATLAB:

- Interpolazione;
- Polinomi;
- Analisi statistica dei dati;
- Funzioni di funzioni;
- Integrazione numerica;
- Equazioni differenziali ordinarie;
- Classi e oggetti.

32

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

For, while, if, Switch - e relazioni

For

```
» for i=1:5;a(i)=i^2;end;a
```

a =

```
1      4      9     16     25
```

```
» for i=1:5
```

```
    for j=1:5;a(i,j)=i-j;end;end
```

33

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

While. La forma generale è:

```
while relazione
```

```
    istruzioni
```

```
end
```

Esempio:

```
»a=3000;n=0;while 2^n<a;
```

```
    n=n+1;end;n-1
```

34

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

If. La forma generale è:

```
if relazione 1
    istruzioni 1
elseif relazione 2
    istruzioni 2
.....
else
    istruzioni n
end
```

35

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Esempio:

```
»a=rand;b=rand;if a == b; break
elseif a>b; a, else; b,end
```

36

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Switch e Case. Lo statement SWITCH esegue un gruppo di comandi in funzione del valore di una variabile o espressione. Esempio:

```
SWITCH(plotype)
CASE 'l'
    plot(x,y)
CASE 'm'
    plot(x,y,'-+')
OTHERWISE
    error('Opzione inesistente')
END
```

37

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Operatori relazionali:

| | |
|----|----------------------|
| < | minore di |
| > | maggiore di |
| <= | minore di o eguale |
| >= | maggiore di o eguale |
| == | eguale |
| ~= | non eguale |

38

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Operatori logici:

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

Se applicata a scalari, una relazione assume il valore scalare 1 (vera) o 0 (falsa);

Esempi:

```
»3>5,3<5
```

39

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Se applicata a matrici di uguale dimensione, una relazione diventa una matrice composta da 1 (vera) e 0 (falsa).

Esempio:

```
»a=rand(5);b=triu(a);a==b
```

40

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

- Una relazione fra matrici è interpretata da `while` e `if` come vera se tutti gli elementi della matrice-relazione è non-zero.
- Nell'esempio seguente, le *istruzioni* sono eseguite se A e B sono eguali:

```
»if A == B  
    istruzioni  
end
```

41

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Nell'esempio seguente, le istruzioni sono eseguite se A e B non sono eguali:

```
if any(any(A~=B))  
    istruzioni  
end
```

o, in modo più semplice:

```
if A == B else  
    istruzioni  
end
```

42

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Si noti che operando nel seguente modo:

```
»if A~=B, istruzioni, end
```

non si ottiene il risultato voluto, poiché le *istruzioni* verranno eseguite solo se *tutti* i corrispondenti elementi di A e B non sono eguali.

Le funzioni `any` e `all` possono venire utilizzate per ridurre le relazioni matriciali a vettori o scalari.

43

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Funzioni scalari

Alcune funzioni MATLAB operano su scalari e, applicate a matrici, operano element-wise:

| | | | | |
|------------------|-------------------|-----------------------------|-------------------|--------------------|
| <code>sin</code> | <code>asin</code> | <code>exp</code> | <code>abs</code> | <code>round</code> |
| <code>cos</code> | <code>acos</code> | <code>log</code> (naturale) | <code>sqrt</code> | <code>floor</code> |
| <code>tan</code> | <code>atan</code> | <code>rem</code> (resto) | <code>sign</code> | <code>ceil</code> |

44

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Funzioni vettoriali

- Altre funzioni operano su vettori (riga o colonna), ma agiscono su matrici $m \times n$ ($m \geq 2$) colonna per colonna, ottenendo un vettore riga;
- Per operare riga-per-riga, è necessario usare l'operatore ' :

»`mean(A')'`

Per esempio, il valore massimo di una matrice A è dato da `max(max(A))` e non da `max(A)`.

45

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Funzioni vettoriali:

| | | | |
|-------------------|-------------------|---------------------|------------------|
| <code>max</code> | <code>sum</code> | <code>median</code> | <code>any</code> |
| <code>min</code> | <code>prod</code> | <code>mean</code> | <code>all</code> |
| <code>sort</code> | <code>std</code> | | |

46

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Funzioni matriciali

La flessibilità - e la potenza - di MATLAB deriva dalle funzioni matriciali; vediamo alcune:

| | |
|--------------------|------------------------------|
| <code>eig</code> | autovalori e autovettori |
| <code>chol</code> | fattorizzazione di Cholesky |
| <code>svd</code> | singular value decomposition |
| <code>inv</code> | inverse |
| <code>lu</code> | fattorizzazione LU |
| <code>qr</code> | fattorizzazione QR |
| <code>schur</code> | decomposizione di Schur |

47

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Funzioni matriciali (cont.)

| | |
|-------------------|--------------------------|
| <code>poly</code> | polinomio caratteristico |
| <code>det</code> | determinante |
| <code>size</code> | dimensione |
| <code>cond</code> | condition number |
| <code>rank</code> | rango |

48

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Le funzioni MATLAB possono avere uno o più argomenti in uscita:

» $y = \text{eig}(A)$ oppure » $\text{eig}(A)$

produce un vettore colonna con gli *autovalori* di A ;

» $[U,D] = \text{eig}(A)$

produce una matrice U le cui colonne sono gli *autovettori* di A , ed una *matrice diagonale* D con gli *autovalori* di A sulla diagonale (provare.....)

49

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Sottomatrici e notazione *colon*

- Vettori e matrici sono usati per manipolare, in modo complesso, dati aggregati;
- La notazione colon - usata per generare vettori e indirizzare sottomatrici - e l'indirizzamento tramite vettori sono strumenti molto efficienti;
- L'uso di tali caratteristiche permette di evitare, a volte, l'utilizzo di loops, che rendono il calcolo più lento.

50

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

- L'espressione `1:5` (v. *for*) fornisce il vettore riga `[1 2 3 4 5]`;
- l'espressione `0.2:0.2:1.2` fornisce `[0.2 0.4 0.6 0.8 1.0 1.2]`;
- `5:-1:1` fornisce `[5 4 3 2 1]`;
- Le istruzioni seguenti generano una tabella:

```
»x=[0.0:0.1:2.0]';  
»y = sin(x);  
»[x y]
```

51

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

- La notazione colon può essere usata per accedere a sottomatrici:
»`A(1:4,3)` è il vettore colonna costituito dai primi quattro elementi della terza colonna di `A`;
»`A(:,3)` è la terza colonna di `A`;
»`A(1:4,:)` rappresenta le prime quattro righe;
- Vettori di interi possono venire utilizzati per l'indicizzazione:
»`A(:, [2 4])` contiene le colonne 2 e 4 di `A`.

52

MATLAB - Introduzione

E. Nobile - DINMA - Sezione di Fisica Tecnica, Università di Trieste

Tale indicizzazione può usarsi anche:

```
»A(:, [2 4 5]) = B(:, 1:3)
```

Si noti che l'*intera* matrice A viene alterata e visualizzata (provare...);

Altro esempio:

```
»A(:, [2 4]) = A(:, [2 4])*[1 2; 3 4]
```

Anche in tal caso l'*intera* matrice è modificata;

Se x è un vettore di di lunghezza n, qual'è l'effetto dell'istruzione:

```
»x = x(n:-1:1)
```