

Introduzione al MATLAB[©]

Parte 1

Variabili ed array

Lucia Gastaldi

DICATAM - Sezione di Matematica,
<http://lucia-gastaldi.unibs.it>

Indice

- 1 Cos'è il MATLAB
 - Componenti principali di MATLAB
 - Avvio e Strumenti del Desktop
- 2 Variabili
 - Assegnazione di variabili scalari
 - Formati di rappresentazione dei numeri
 - Operazioni con variabili scalari
- 3 Array
 - Vettori e matrici
 - Operazioni con array
 - Trasposizione
 - Operazioni punto

MATLAB = MATrix LABoratory

MATLAB è un linguaggio ad alto livello unito a un ambiente interattivo che consente di risolvere problemi di **calcolo scientifico**.

Calcolo scientifico:

si occupa dello sviluppo, della implementazione e dell'analisi degli algoritmi numerici utili per l'impiego di **modelli matematici**.

MATLAB originariamente forniva un accesso facilitato alle routine per il calcolo con le matrici sviluppate nei pacchetti LINPACK e EISPACK, scritti in FORTRAN. Al giorno d'oggi, include le librerie LAPACK e BLAS, che servono nel calcolo matriciale.

Cinque parti principali

- Strumenti del Desktop e ambiente di sviluppo
- Libreria delle funzioni matematiche
- Linguaggio MATLAB
- Grafica
- Interfacce esterne

Funzionalità di Matlab

- Calcolo numerico
- Analisi e visualizzazione dei dati
- Programmazione e sviluppo di algoritmi
- Sviluppo e distribuzione di applicazioni

MATLAB è un sistema interattivo il cui elemento di base è l'**array** che non richiede dimensionamento.

TOOLBOX

TOOLBOX = Librerie per applicazioni specifiche

- Calcolo Parallelo
- Matematica, Statistica e Ottimizzazione
- Analisi di dati, e visualizzazione.
- Grafica scientifica ed ingegneristica.
- Analisi e Progettazione di Sistemi di Controllo
- Elaborazione di Segnali e Comunicazioni
- Image Processing e Computer Vision
- Test & Misurazioni
- Computational Finance
- Computational Biology

Per maggiori dettagli: www.mathworks.com

Aree di applicazione

- Calcolo tecnico
- Sistemi di controllo
- Elaborazione segnali digitali
- Sistemi di comunicazione
- Elaborazione immagini e video
- Meccatronica
- Misura e collaudo
- Computational Biology
- Computational Finance

Avviare Matlab

Selezionare l'icona di **MATLAB** nel desktop.

Si apre il **Desktop** diviso in cinque parti

- Command Window
- Current Folder
- Workspace
- Command History
- Details

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.

Nomi delle variabili

I nomi delle variabili **non** devono contenere **spazi** e caratteri speciali come:

- **simboli di operazione**: -, =, +, *;
- **apostrofi**
- **punteggiatura**
- **slash e backslash**

Il comando

```
>> a=1.54;
```

non produce risposta

Variabile di default

```
>> 1.67
```

```
produce
```

```
>> ans =
```

```
1.6700
```

ans è il nome della variabile di default.

Per vedere il contenuto di una variabile

Visualizzo il contenuto della variabile a

```
>> a      produce
```

```
a =  
    1.5400
```

Visualizzo il contenuto della variabile ans

```
>> ans    produce
```

```
ans =  
    1.6700
```

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 =  
    0.45672300000000
```

Il numero è rappresentato con 15 cifre

Formati disponibili

Variabile	Significato
FORMAT	Default.
FORMAT SHORT	Virgola fissa "scalata" con 5 cifre.
FORMAT LONG	Virgola fissa "scalata" con 15 cifre.
FORMAT SHORT E	Forma esponenziale con 5 cifre di "mantissa".
FORMAT LONG E	Forma esponenziale con 15 cifre di "mantissa".
FORMAT SHORT G	Rappresentazione migliore con 5 cifre.
FORMAT LONG G	Rappresentazione migliore con 15 cifre.

Variabili predefinite

Variabile	Significato
ans	valore ultima operazione eseguita e non assegnata ad una variabile
i, j	unità immaginaria, $\sqrt{-1}$
pi	approssimazione di π
eps	precisione macchina
realmax	massimo numero macchina positivo rappresentabile
realmin	minimo numero macchina positivo rappresentabile
Inf	∞ , ossia un numero maggiore di <code>realmax</code>
NaN	Not a Number (0/0, Inf/Inf, ...)
computer	tipo di computer
version	versione di MATLAB

Il comando `clear`

`clear`

Per cancellare il contenuto della variabile `a`: `>> clear a`

Per cancellare il contenuto di tutte le variabili: `>> clear`

Il contenuto delle variabili predefinite 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
```

Area di lavoro **WORKSPACE**

Le variabili vengono memorizzate nell'area di lavoro **Workspace**. La finestra **Workspace** contiene la lista della variabili e le seguenti informazioni:

- **Name**: nome della variabile.
- **Value**: valore assegnato alla variabile.
- **Size**: dimensione come array (righe per colonne).
- **Bytes**: occupazione di memoria in termini di **bytes**.
- **Class**: il tipo di variabile `char`, `double`, `sparse`, `cell`, `struct`, `uint8`.

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).

Lettere maiuscole e minuscole sono considerate diverse sia nei comandi che nei nomi delle variabili.

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**

Per spezzare le righe

Il comando

```
>> b=1+1/2+5/3+1/4+23/6+...  
2/9+1/10;
```

permette di spezzare un'istruzione troppo lunga

Array

Il linguaggio MATLAB lavora con un solo tipo di oggetti: **l'array di MATLAB**.

Tutte le variabili di Matlab, inclusi **scalari, vettori, matrici, stringhe, celle (cell arrays), strutture e oggetti** sono memorizzati in Matlab come **array**.

Ogni array contiene le seguenti informazioni:

- Il tipo
- La dimensione
- I dati associati all'array
- Se la variabile è reale o complessa, nel caso di array numerico
- Gli indici e gli elementi diversi da zero, nel caso di array sparse
- Il numero di campi e il nome dei campi, nel caso di una struttura o oggetto.

Problema

Trovare il polinomio $P(x)$ di grado 5 che soddisfa le seguenti condizioni:

- Il polinomio si annulla per $x = -2$ e $x = 4$ e assume in questi punti il valore minimo;
- Il polinomio ha un massimo relativo per $x = 1$ dove assume il valore 10.

Il polinomio ha la seguente forma

$$P(x) = a_1x^5 + a_2x^4 + a_3x^3 + a_4x^2 + a_5x + a_6.$$

Le incognite del nostro problema sono i coefficienti che possono essere considerati come le componenti di un vettore.

Imponiamo le condizioni:

$$P(-2) = P(4) = 0; P(1) = 10; P'(-2) = P'(1) = P'(4) = 0.$$

Dobbiamo calcolare anche la derivata:

$$P'(x) = 5a_1x^4 + 4a_2x^3 + 3a_3x^2 + 2a_4x + a_5.$$

La soluzione si trova risolvendo il seguente sistema di equazioni lineari

$$-32a_1 + 16a_2 - 8a_3 + 4a_4 - 2a_5 + a_6 = 0$$

$$1024a_1 + 256a_2 + 64a_3 + 16a_4 + 4a_5 + a_6 = 0$$

$$a_1 + a_2 + a_3 + a_4 + a_5 + a_6 = 10$$

$$80a_1 - 32a_2 + 12a_3 - 4a_4 + a_5 = 0$$

$$5a_1 + 4a_2 + 3a_3 + 2a_4 + a_5 = 0$$

$$1280a_1 + 256a_2 + 48a_3 + 8a_4 + a_5 = 0.$$

Oppure scritto in forma matriciale

$$\begin{pmatrix} -32 & 16 & -8 & 4 & -2 & 1 \\ 1024 & 256 & 64 & 16 & 4 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 80 & -32 & 12 & -4 & 1 & 0 \\ 5 & 4 & 3 & 2 & 1 & 0 \\ 1280 & 256 & 48 & 8 & 1 & 0 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 10 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Assegnazione di vettori e matrici

L'assegnazione di matrici in MATLAB può essere fatta in diversi modi:

- introducendo una lista di elementi;
- caricando le matrici da file di dati esterni;
- generando le matrici usando una funzione interna matlab;
- creando le matrici mediante un proprio programma.

Vettori riga e colonna

Vettore riga

Array 1×4 , 1 riga e 4 colonne contenente i numeri da 1 a 4

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

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

```
>> a=(1:4);
```

```
>> a
```

```
a =
```

```
     1     2     3     4
```

Vettore colonna

Array 4×1 , 4 righe e 1 colonna

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

```
b =
```

```
     1
```

```
     2
```

```
     3
```

```
     4
```

Assegnazione di array - matrici

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

```
c =
```

```
    5    3    4  
    2    4   -2
```

Genera un array 2x3, matrice 2 righe e 3 colonne

```
>> d(3,4)=3
```

```
d =
```

```
    0    0    0    0  
    0    0    0    0  
    0    0    0    3
```

Genera una matrice 3x4, che ha tutti elementi nulli tranne quello di posto 3,4

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

Dimensioni di un array

Il comando **size** fornisce le dimensioni di una matrice.

```
>> size(c)
```

produce il vettore riga di due elementi contenenti il numero di righe e di colonne di c.

```
ans =  
     2     3
```

Il comando **length** fornisce la lunghezza di un vettore.

```
>> length(a)
```

produce un numero pari alla lunghezza del vettore a.

```
ans =  
     4
```

```
length(c)=max(size(c))
```


Come accedere agli elementi di array

```
>> 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

Come modificare un elemento di un array

```
>> 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.

Soluzione di un sistema lineare

Sia A matrice quadrata $n \times n$, b vettore colonna $n \times 1$, la soluzione del sistema lineare

$$Ax = b,$$

è il vettore colonna $x \in \mathbb{R}^n$.

In Matlab la soluzione del sistema lineare si ottiene con il comando

$$x=A \setminus b.$$

Da usare con attenzione

L'operatore \setminus usa algoritmi differenti per trattare diversi tipi di matrici:

- Permutazioni di matrici triangolari.
- Matrici simmetriche e definite positive.
- Matrici quadrate, non singolari e piene.
- Matrici quadrate, non singolari e sparse.
- Sistemi rettangolari sovradeterminati.
- Sistemi rettangolari sottodeterminati.

Operazioni standard dell'algebra lineare

- + 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:** il numero delle colonne della prima matrice deve essere uguale al numero delle colonne della seconda matrice.

Operazioni su array

```
>> a1+b
```

entrambi vettori colonna 4x1

```
ans =
```

```
2
```

```
4
```

```
8
```

```
8
```

```
>> a-b
```

```
??? Error using ==> -
```

```
Matrix dimensions must agree.
```

a =vettore riga 1x4

b =vettore colonna 4x1

```
>> a*b  
ans =  
    36
```

$(1 \times 4)(4 \times 1)$ *prodotto scalare*

```
>> c*d'  
ans =  
    358  
   -14
```

$(2 \times 3)(3 \times 1)$ *prodotto matrice vettore*

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

$(3 \times 1)(2 \times 3)$ *prodotto non possibile*

Trasposizione di vettore

```
>> 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***

... trasposizione di matrici:

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


Operazioni “punto”

Le operazioni **punto** agiscono su array che abbiano le stesse dimensioni:

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

```
>> a1b=a1.*b
```

```
a1b =
```

```
1  
4  
15  
16
```

$$(a1b)_i = (a1)_i * b_i$$

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