

ESERCITAZIONI

MATLAB – Introduzione



MATLAB è una piattaforma di programmazione e calcolo numerico utilizzata da milioni di ingegneri e scienziati per l'analisi di dati, lo sviluppo di algoritmi e la creazione di modelli.

MATLAB combina un ambiente desktop ottimizzato per l'analisi iterativa e i processi di progettazione con un linguaggio di programmazione che esprime le operazioni matematiche con matrici e array in modo diretto.

MATLAB dispone di toolbox sviluppati professionalmente, rigorosamente testati e interamente documentati.

Le applicazioni MATLAB consentono di vedere come algoritmi differenti lavorano con i dati. Possono essere eseguite iterazioni finché non si ottengono i risultati desiderati, quindi può essere generato automaticamente un programma MATLAB per riprodurre o automatizzare il lavoro da svolgere.

Le analisi possono essere dimensionate per eseguirle su cluster, GPU e cloud solo con lievi modifiche al codice. Non è necessario riscrivere il codice o apprendere la programmazione di big data e tecniche out-of-memory.

<https://it.mathworks.com/videos/matlab-overview-61923.html>

ESERCITAZIONI

MATLAB – Introduzione



Funzionalità MATLAB



Analisi dei dati

Esplora, modella e visualizza dati



Grafica

Visualizzazione ed esplorazione dei dati



Programmazione

Creazione di script, funzioni e classi



Creazione di app

Creazione di app web e desktop



Utilizzo di MATLAB con altri linguaggi

Utilizzo di MATLAB con Python, C/C++, Fortran, Java e altri linguaggi



Hardware

Connetti MATLAB all'hardware



Calcolo parallelo

Esegui calcoli su larga scala utilizzando computer desktop multicore, GPU, cluster, grid e cloud



Distribuzione su desktop e via web

Condividi i tuoi programmi MATLAB



Cloud computing

Lavora in ambienti cloud da MathWorks Cloud a cloud pubblici, inclusi AWS e Azure

ESERCITAZIONI

MATLAB – Introduzione



Dalla ricerca alla produzione



Distribuzione ai sistemi IT aziendali

Il codice MATLAB è pronto per la produzione, così da poter accedere direttamente ai sistemi cloud e aziendali, integrandosi con sistemi business e fonti di dati.



Implementazione in dispositivi embedded

Converti automaticamente algoritmi MATLAB in codice C/C++ e HDL per essere implementati in dispositivi embedded.



Integrazione con la progettazione Model-Based

MATLAB lavora con Simulink per supportare la progettazione Model-Based, che è utilizzata per la simulazione multidominio, la generazione automatica del codice e i test e le verifiche dei sistemi embedded.

ESERCITAZIONI

MATLAB – Introduzione



Esempi di aree di applicazione



Sistemi di controllo

Progetta, testa e implementa sistemi di controllo



Machine Learning

Addestra modelli, regola i parametri e distribuisce applicazioni



Elaborazione di segnali

Analizza segnali e dati di serie storiche. Modella, progetta e simula sistemi di elaborazione dei segnali.



Deep Learning

Prepara i dati, la progettazione, simulazione e distribuzione per reti neurali profonde



Manutenzione predittiva

Sviluppa e distribuisce software di monitoraggio delle condizioni e di manutenzione predittiva



Verifica e misurazione

Acquisisci, analizza ed esplora i dati e automatizza i test



Elaborazione di immagini e Computer Vision

Acquisisci, elabora e analizza immagini e video per lo sviluppo di algoritmi e la progettazione di sistemi



Robotica

Converti le tue idee e i tuoi concetti di robotica in sistemi autonomi che operano senza problemi in ambienti reali.



Comunicazioni wireless

Crea, progetta, testa e verifica sistemi di comunicazione wireless

https://it.mathworks.com/help/index.html?s_tid=CRUX_lftnav

https://matlabacademy.mathworks.com/?s_tid=getstart_mlacad

ESERCITAZIONI

MATLAB – Installazione



<https://www.ict.univpm.it/node/237>

<https://it.mathworks.com/help/install/install-products.html>

https://it.mathworks.com/help/matlab/matlab_env/check-for-software-updates.html

ESERCITAZIONI

MATLAB – Installazione



Aggiungere (se non sono stati già aggiunti) i seguenti Toolbox:

-MATLAB

-Image Processing Toolbox

-Statistics and Machine Learning Toolbox

-Simulink

-Control System Toolbox

-Symbolic Math Toolbox

ESERCITAZIONI

MATLAB – Installazione



Al fine di effettuare un check sui Toolbox installati, utilizzare il comando `ver` .

Per aggiungere Toolbox, seguire i seguenti passi:

- 1) Aprire Matlab.
- 2) Click su Add-Ons sulla barra in alto.
- 3) Click su Get Add-Ons.
- 4) Digitare il nome del Toolbox da installare (ad esempio Simulink) nello spazio bianco Search for add-ons.
- 5) Click sul Toolbox da installare.
- 6) Click su Install ed eseguire la procedura di installazione.

Una volta terminata la procedura di installazione dei Toolbox indicati, utilizzare nuovamente il comando `ver` al fine di verificare che essi siano stati aggiunti.

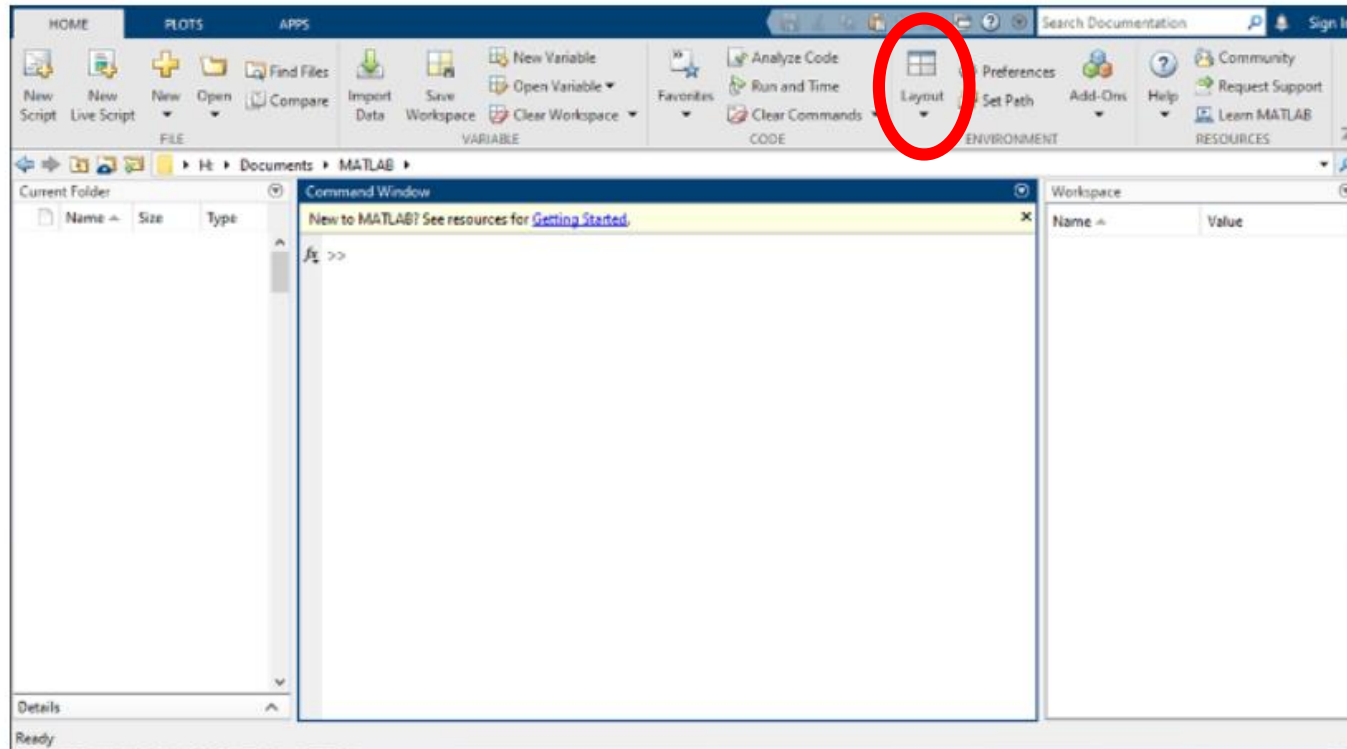
ESERCITAZIONI

MATLAB – Come iniziare



Nozioni di base sul desktop

Quando si avvia MATLAB®, il desktop viene visualizzato nel layout predefinito.



Il desktop comprende i seguenti pannelli:

- **Current Folder:** consente di accedere ai file dell'utente.
- **Command Window:** consente di inserire i comandi dalla riga di comando, indicata dal prompt (>>).
- **Workspace:** consente di visualizzare i dati creati o importati dai file.

ESERCITAZIONI

MATLAB – Come iniziare



Nozioni di base sul desktop

Mentre si lavora in MATLAB, si utilizzano comandi che creano variabili e richiamano funzioni. Ad esempio, creare una variabile denominata `a` digitando questa dichiarazione dalla riga di comando:

```
a = 1
```

MATLAB aggiunge la variabile `a` al workspace e visualizza il risultato nella finestra di comando.

```
a =
```

```
1
```

Creare alcune altre variabili.

```
b = 2
```

```
b =
```

```
2
```

```
c = a + b
```

```
c =
```

```
3
```

```
d = cos(a)
```

```
d =
```

```
0.5403
```

ESERCITAZIONI

MATLAB – Come iniziare



Nozioni di base sul desktop

Quando non si specifica una variabile di output, MATLAB utilizza la variabile `ans`, che sta per *answer* ('risposta' in lingua inglese), per memorizzare i risultati del calcolo effettuato.

```
sin(a)
```

```
ans =
```

```
0.8415
```

Se si termina una dichiarazione con un punto e virgola, MATLAB effettua il calcolo, ma sopprime la visualizzazione dell'output nella finestra di comando.

```
e = a*b;
```

È possibile richiamare i comandi precedenti premendo i tasti Freccia in alto e Freccia in basso, \uparrow e \downarrow . Premere i tasti freccia da una riga di comando vuota o dopo aver digitato i primi caratteri di un comando. Ad esempio, per richiamare il comando `b = 2`, digitare `b` e quindi premere il tasto Freccia in alto.

ESERCITAZIONI

MATLAB – Come iniziare



Matrici e array

MATLAB è un'abbreviazione di "matrix laboratory", "laboratorio di matrici". Mentre altri linguaggi di programmazione lavorano principalmente su un numero per volta, MATLAB® consente di operare soprattutto su intere matrici e array.

Tutte le variabili di MATLAB sono *array* multidimensionali, indipendentemente dal tipo di dati. Una *matrice* è un array bidimensionale frequentemente utilizzato nell'algebra lineare.

Creazione di array

Per creare un array di quattro elementi su un'unica riga, separare gli elementi con una virgola (,) o uno spazio.

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

a = 1×4

1	2	3	4
---	---	---	---

Questo tipo di array è definito *vettore riga*.

Per creare una matrice con righe multiple, separare le righe con punti e virgole.

```
a = [1 3 5; 2 4 6; 7 8 10]
```

a = 3×3

1	3	5
2	4	6
7	8	10

ESERCITAZIONI

MATLAB – Come iniziare



Matrici e array

Creazione di array

Un altro metodo per creare una matrice consiste nell'utilizzare una funzione, come `ones`, `zeros` o `rand`. Ad esempio, creare un vettore colonna 5x1 composto da zeri.

```
z = zeros(5,1)
```

```
z = 5x1
```

```
0  
0  
0  
0  
0
```

Operazioni su matrici e array

MATLAB consente di elaborare tutti i valori di una matrice utilizzando un unico operatore o funzione aritmetica.

```
a + 10
```

```
ans = 3x3
```

```
11    13    15  
12    14    16  
17    18    20
```

ESERCITAZIONI

MATLAB – Come iniziare



Matrici e array

Operazioni su matrici e array

```
sin(a)
```

```
ans = 3×3
```

```
0.8415    0.1411   -0.9589  
0.9093   -0.7568   -0.2794  
0.6570    0.9894   -0.5440
```

Per la trasposizione di una matrice utilizzare l'apice ('):

```
a'
```

```
ans = 3×3
```

```
1     2     7  
3     4     8  
5     6    10
```

ESERCITAZIONI

MATLAB – Come iniziare



Matrici e array

Operazioni su matrici e array

Utilizzando l'operatore `*` è possibile eseguire moltiplicazioni standard di matrici che calcolano i prodotti interni tra righe e colonne. Ad esempio, confermare che moltiplicando una matrice per il suo inverso si ottiene la matrice identità:

```
p = a*inv(a)
```

```
p = 3×3
```

```
1.0000    0.0000   -0.0000
     0    1.0000   -0.0000
     0    0.0000    1.0000
```

Si noti che `p` non è una matrice di valori interi. MATLAB memorizza i numeri come valori a virgola mobile e le operazioni aritmetiche distinguono tra piccole differenze tra il valore reale e la sua rappresentazione con virgola mobile. È possibile visualizzare più cifre decimali con il comando `format`:

```
format long
```

```
p = a*inv(a)
```

```
p = 3×3
```

```
0.9999999999999996    0.0000000000000007   -0.0000000000000002
                     0    1.0000000000000000   -0.0000000000000003
                     0    0.0000000000000014    0.9999999999999995
```

Per ripristinare la visualizzazione al formato più breve utilizzare

```
format short
```

`format` ha effetto solo sulla visualizzazione dei numeri, non sul modo in cui MATLAB li calcola o li salva.

ESERCITAZIONI

MATLAB – Come iniziare



Matrici e array

Operazioni su matrici e array

Per eseguire moltiplicazioni basate sugli elementi, non moltiplicazioni di matrici, utilizzare l'operatore `.*`:

```
p = a.*a
```

```
p = 3x3
```

1	9	25
4	16	36
49	64	100

Tutti gli operatori di matrici per moltiplicazione, divisione ed elevazione a potenza presentano un corrispettivo operatore di array che opera in modo analogo. Ad esempio, elevare al cubo ciascun elemento di `a`:

```
a.^3
```

```
ans = 3x3
```

1	27	125
8	64	216
343	512	1000

ESERCITAZIONI

MATLAB – Come iniziare



Matrici e array

Concatenazione

La *concatenazione* è il processo di congiunzione di array, per crearne di più grandi. In realtà il primo array è stato creato quando se ne sono concatenati i singoli elementi. L'operatore per la concatenazione è la coppia di parentesi quadre `[]`.

```
A = [a, a]
```

A = 3×6

1	3	5	1	3	5
2	4	6	2	4	6
7	8	10	7	8	10

La concatenazione con virgole di array disposti uno accanto all'altro è definita concatenazione *orizzontale*. Ogni array deve presentare lo stesso numero di righe. Analogamente, quando gli array presentano lo stesso numero di colonne, è possibile concatenare queste ultime *verticalmente*, utilizzando dei punti e virgole.

```
A = [a; a]
```

A = 6×3

1	3	5
2	4	6
7	8	10
1	3	5
2	4	6
7	8	10

ESERCITAZIONI

MATLAB – Come iniziare



Matrici e array

Numeri complessi

I numeri complessi comprendono sia parti reali che parti immaginarie, in cui l'unità immaginaria è costituita dalla radice quadrata di -1.

```
sqrt(-1)
```

```
ans = 0.0000 + 1.0000i
```

Per rappresentare la parte immaginaria dei numeri complessi utilizzare i o j.

```
c = [3+4i, 4+3j; -i, 10j]
```

```
c = 2x2 complex
```

```
3.0000 + 4.0000i    4.0000 + 3.0000i  
0.0000 - 1.0000i    0.0000 +10.0000i
```

Riferimenti Bibliografici

[1] <https://it.mathworks.com>