

Metodi numerici con elementi di Programmazione

A.A. 2013-2014

Introduzione al MatLab
(parte 1)

Docente: Vittoria Bruni

Email: vittoria.bruni@sbai.uniroma1.it

Ufficio: Via A. Scarpa,
Pal. B, I piano, Stanza n. 16
Tel. 06 49766648

Ricevimento: Giovedì 14.00-15.00

Il **materiale didattico** è disponibile sul sito <http://ingaero.uniroma1.it/> nella pagina dedicata al corso Metodi Numerici con elementi di Programmazione

Per consultazione: Getting Started with MatLab – The mathworks
www.mathworks.com

Matlab

Matlab (MATrix LABoratory) è un sistema software integrato per il calcolo tecnico e scientifico

- Linguaggio di programmazione ad alto livello interpretato, con particolari facilitazioni nelle elaborazioni di matrici
- Contiene i costrutti tipici dei linguaggi di programmazione
- Possiede un ampio insieme di tipi di dato predefiniti
- Supporta la programmazione orientata agli oggetti
- Usato sia per creare rapidamente piccoli programmi test (programmazione in the small) che per applicazioni più complesse (programmazione in the large)

Matlab

Matlab (MATrix LABoratory) è un sistema software integrato per il calcolo tecnico e scientifico

- Grafici in 2 e 3 dimensioni
 - Funzioni per la visualizzazione di vettori e matrici
 - Funzioni per l'impostazione dell'aspetto della visualizzazione (annotazioni, colori, linee,...)
 - Funzioni per elaborare immagini e creare animazioni
 - Funzioni per creare interfacce

Matlab

Matlab (MATrix LABoratory) è un sistema software integrato per il calcolo tecnico e scientifico

- Programmi interni per la **risoluzione dei problemi dell'Analisi Numerica**

- contiene funzioni elementari, algoritmi di calcolo, algebra lineare, ...

- Pacchetti per svariati tipi di applicazioni (**Toolbox**) --- l'elaborazione numerica dei segnali e delle immagini, la simulazione di sistemi dinamici, il calcolo simbolico, wavelet, ecc.

- Interazione con altri linguaggi di programmazione (per es. C e Fortran)

Matlab

- Creato da **Cleve Moler** (Univ. Del New Mexico) alla fine degli anni '70 per fornire agli studenti un facile accesso al software per l'elaborazione di matrici sviluppato in LINPACK e EISPACK
- Proprietà della **MathWorks** dal 1984, è diventato uno standard nella ricerca, nella didattica e anche nell'industria
- **Ambiti applicativi:**
matematica e calcolo numerico; sviluppo di modelli, simulazioni e prototipi; analisi dati; visualizzazione scientifica; applicazioni con interfaccia utente grafica

Octave

- **Ambiente integrato** per il calcolo scientifico e la visualizzazione grafica
- **Distribuito gratuitamente** dalla GNU www.octave.org

http://sourceforge.net/projects/octave/files/Octave%20Windows%20binaries/Octave%203.2.4%20for%20Windows%20MinGW32%20Installer/Octave-3.2.4_i686-pc-mingw32_gcc-4.4.0_setup.exe/download

oppure

<http://octave.softonic.it/>

- E' **compatibile con Matlab**: la maggior parte dei programmi Matlab possono essere eseguiti in ambiente Octave senza necessità di modifiche (e viceversa)
- Ha un' **interfaccia grafica diversa** da Matlab

è possibile scaricare GUIOctave per averne una simile per esempio, GUIOctave v1.0.16 per la versione Octave-3.2.4 Istruzioni dettagliate si possono trovare al seguente link

<http://www.math.uh.edu/~jmorgan/Math2331/install.octave.windows.2011.03.31.pdf>

Matlab

Interfaccia grafica: Finestre

The screenshot displays the MATLAB 7.11.0 (R2010b) interface. The main window is titled "MATLAB 7.11.0 (R2010b)" and contains several panes:

- Command Window:** The central pane where commands are entered. It shows a message about customizable keyboard shortcuts and the execution of the command `>> 3-2`, resulting in `ans = 5`. A red circle highlights the prompt `>>`, with an arrow pointing to the text "Linea di comando o prompt".
- Workspace:** The right-hand pane showing the current workspace. It contains a single variable `ans` with a value of `5`.
- Command History:** The bottom-right pane showing a list of previously executed commands and their timestamps. The last entry is `>> 3+2` at 20:06 on 06/10/2012.

Red annotations include a circle around the "Command Window" title bar and an arrow pointing from it to the text "finestra principale DI LAVORO INTERATTIVA: dare comandi, eseguire funzioni, etc".

finestra principale DI LAVORO INTERATTIVA: dare comandi, eseguire funzioni, etc

Linea di comando o prompt

Matlab

Interfaccia grafica: Command window

Matlab lavora in **modo interattivo**, cioè l'utente digita una istruzione ed ha immediatamente la risposta. Il **prompt** su cui si digita l'istruzione è la coppia di caratteri `>>`

`>>` comando (Per eseguire, digitare Enter)

Esempio

```
>> 3+2 (Enter)
```

```
ans =
```

```
5
```

```
>>
```

Matlab

Interfaccia grafica: Command window

- Per uscire dalla sessione di lavoro interattiva usare il comando:

```
>> quit
```

- Per cancellare il contenuto della finestra usare il comando:

```
>> clc
```

- Per ripetere le ultime operazioni effettuate usare i tasti: ↑ e ↓
- Più comandi sulla stessa riga devono essere separati da una **virgola**

```
>> 3+2, 5*10-4, 55-22
```

- Un'istruzione molto lunga si può scrivere su più righe consecutive usando

...

```
>> 3+2+4+(5*3)-5 ...
```

```
-5+10*3 (Enter)
```

Matlab

Interfaccia grafica: Finestre

MATLAB 7.11.0 (R2010b)

File Edit Debug Parallel Desktop Window Help

Current Folder: C:\Program Files\MATLAB\R2010b\bin

Shortcuts How to Add What's New

Current Folder bin

Command Window

New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#).

MATLAB desktop keyboard shortcuts, such as Ctrl+S, are now customizable. In addition, many keyboard shortcuts have changed for improved consistency across the desktop.

To customize keyboard shortcuts, use [Preferences](#). From there, you can also restore previous default settings by following the steps outlined in [Help](#).

[Click here](#) if you do not want to see this message again.

```
>> 3+2  
ans =  
5  
fx >>
```

Workspace

Name	Value
ans	5

Command History

```
-- 04/10/2012 12:28 --  
0.75+0.5+0.25+0.15+0  
1.5+0.0833+0.3333+0.  
0.05*18  
37.575+7.5  
6.5+1.5+1.5+1.125  
6.5+1.5+1.5+1.125+2.  
6.5+1.5+1.5+1.125+2.  
0.8+0.05+.4+0.075+0..  
29.6+4.025  
-- 05/10/2012 13:25 --  
-- 06/10/2012 20:06 --  
3+2
```

contiene tutti i comandi digitati nel prompt

Matlab

Interfaccia grafica: Command history:

contiene tutti comandi digitati da prompt

per ripeterli, basta selezionare il comando di interesse con un click

Matlab

Interfaccia grafica: Finestre

MATLAB 7.11.0 (R2010b)

File Edit Debug Parallel Desktop Window Help

Current Folder: C:\Program Files\MATLAB\R2010b\bin

Shortcuts How to Add What's New

Current Folder bin

Command Window

New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#).

MATLAB desktop keyboard shortcuts, such as Ctrl+S, are now customizable. In addition, many keyboard shortcuts have changed for improved consistency across the desktop.

To customize keyboard shortcuts, use [Preferences](#). From there, you can also restore previous default settings by following the steps outlined in [Help](#).

[Click here](#) if you do not want to see this message again.

```
>> 3+2  
ans =  
     5  
fx >>
```

Workspace

Name	Value
ans	5

Command History

```
04/10/2012 12:28  
0.75+0.5+0.25+0.15+0  
1.5+0.0833+0.3333+0.1  
0.05*18  
7.575+7.5  
6.5+1.5+1.5+1.125  
6.5+1.5+1.5+1.125+2.  
6.5+1.5+1.5+1.125+2.  
0.8+0.05+.4+0.075+0.  
29.6+4.025  
05/10/2012 13:25  
06/10/2012 20:06  
3+2
```

contiene tutte le variabili in memoria

Matlab

Interfaccia grafica: Workspace

contiene tutte le variabili usate e memorizzate automaticamente

Cliccando sulla variabile di interesse si apre la finestra **Array Editor** in cui è possibile visualizzare il contenuto della variabile

Matlab

Interfaccia grafica: Finestre

Current Folder: C:\Program Files\MATLAB\R2010b\bin

Workspace

Name	Value
ans	5

Command History

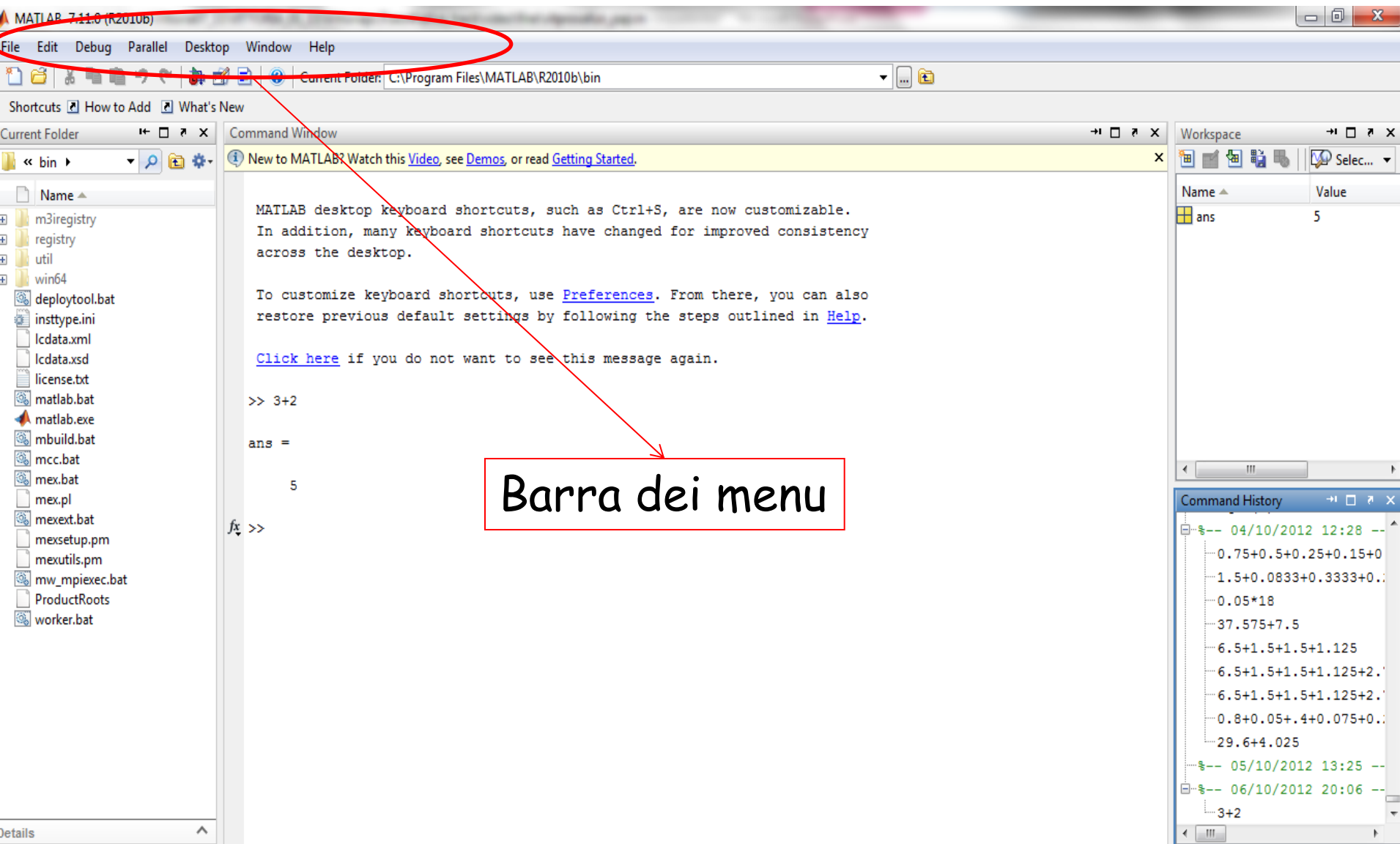
```
04/10/2012 12:28 --  
... 0.75+0.5+0.25+0.15+0  
... 1.5+0.0833+0.3333+0..  
... 0.05*18  
... 37.575+7.5  
... 6.5+1.5+1.5+1.125  
... 6.5+1.5+1.5+1.125+2..  
... 6.5+1.5+1.5+1.125+2..  
... 0.8+0.05+.4+0.075+0..  
... 29.6+4.025  
05/10/2012 13:25 --  
06/10/2012 20:06 --  
... 3+2
```

E' la directory di lavoro in cui Matlab cerca e salva file e dati

Elenco di tutti i file contenuti nella cartella di lavoro. Permette un accesso rapido e diretto ai file

Matlab

Interfaccia grafica: Finestre



Barra dei menu

Octave

Prompt (senza interfaccia grafica)



Si apre cliccando due volte sull'icona

che compare sul Desktop

```
Octave-3.2.4
GNU Octave, version 3.2.4
Copyright (C) 2009 John W. Eaton and others.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE.  For details, type `warranty'.

Octave was configured for "i686-pc-mingw32".

Additional information about Octave is available at http://www.octave.org.

Please contribute if you find this software useful.
For more information, visit http://www.octave.org/help-wanted.html

Report bugs to <bug@octave.org> (but first, please read
http://www.octave.org/bugs.html to learn how to write a helpful report).

For information about changes from previous versions, type `news'.

warning: mark_as_command is obsolete and will be removed from a future version o
f Octave
octave-3.2.4.exe:1> _
```

Octave

Icona sul desktop



Interfaccia grafica: Finestre

The screenshot displays the GUI Octave application window. The 'Window' menu is open, listing various panels and their keyboard shortcuts:

- Command Window Ctrl+0
- Command History Ctrl+1
- Current Directory Ctrl+2
- Workspace Ctrl+3
- Documentation Ctrl+4
- Profiler Ctrl+5
- Comparison Tool Ctrl+6
- Variable Editor Ctrl+7
- Editor / Debugger Ctrl+8

The Command Window shows the following text:

```
Octave 3.2.4
Copyright (C) 2009 John W. Eaton and others.
This software is distributed under the GNU GPL.
For more information, see the source code for copying conditions.
NO WARRANTY; not even for MERCHANTABILITY or
PARTICULAR PURPOSE. For details, type `warranty'.
Configured for "i686-pc-mingw32".
For more information about Octave is available at http://www.octave.org.
Please let us know if you find this software useful.
For more information, visit http://www.octave.org/help-wanted.html
Report bugs to <bug@octave.org> (but first, please read
http://www.octave.org/bugs.html to learn how to write a helpful report).
For information about changes from previous versions, type `news'.
warning: mark_as_command is obsolete and will be removed from a future version of Octave
>> |
```

The Editor / Debugger window is also visible, showing a toolbar and a text editor area with the text 'Senza titolo'.

Matlab

Per salvare la sessione di lavoro interattiva, o parte di essa, si usa il comando **diary**

```
>> diary on
>> diary nome.m
    ( sessione di lavoro che si vuole conservare)
>> diary off
    (parte che non si vuole conservare)
>> diary on
    (si vuole conservare di nuovo)
>> diary off
```

Conserva sia gli input che gli output ma anche tutti i messaggi di eventuali errori --- file di testo

E' paragonabile all'azione di prendere appunti durante una lezione di laboratorio

Matlab come calcolatrice

Operatori

- operazioni elementari
 - somma +
 - differenza -
 - prodotto *
 - divisione /
- operatori logici
 - and &
 - or |
 - not ~
- operatori relazionali
 - maggiore >
 - maggiore o uguale >=
 - minore <
 - minore o uguale <=
 - uguale ==
 - diverso ~=
- elevamento a potenza ^

Matlab come calcolatrice

Costanti predefinite

Costante	Costante in Matlab
Infinito	<code>inf</code>
π	<code>pi</code>
Unità immaginaria	<code>i</code>
Numero massimo rappresentabile (2^{1023})	<code>realmax</code>
Numero minimo rappresentabile (2^{-1022})	<code>realmin</code>
Precisione di macchina ($2.220446049250313 \cdot 10^{-16}$)	<code>eps</code>
Forma indeterminata	<code>nan</code>

Funzioni predefinite

Funzione	Funzione Matlab
Seno	<code>sin(x)</code>
Coseno	<code>cos(x)</code>
Tangente	<code>tan(x)</code>
Arcsin	<code>asin(x)</code>
Arccos	<code>acos(x)</code>
Arctan	<code>atan(x)</code>
Logaritmo naturale	<code>log(x)</code>
Esponenziale	<code>exp(x)</code>
Valore assoluto	<code>abs(x)</code>
Radice quadrata	<code>sqrt(x)</code>
segno	<code>sign(x)</code>

Matlab come calcolatrice

Costanti predefinite

Costante	Costante in Matlab
Infinito	inf
π	pi
Unità immaginaria	i
Numero massimo rappresentabile (2^{1023})	realmax
Numero minimo rappresentabile (2^{-1022})	realmin
Precisione di macchina ($2.220446049250313 \cdot 10^{-16}$)	eps
Forma indeterminata	nan

Rappresentazione dei numeri in Matlab:

forma a virgola mobile (floating point)

su parole di 64 bit (doppia precisione)

Rappresentazione dei numeri

$$x = \pm (1 + a_{-1}2^{-1} + a_{-2}2^{-2} + \dots + a_{-t}2^{-t}) 2^e$$

	s	n	t	Numero totale di bits
Doppia precisione	1	11	52	64

$$L \leq e \leq U$$

	Massimo ($2^{U+1} (1-2^{-t-1})$)	Minimo (2^L)
Doppia precisione ($U=1023, t=52, L=-1022$)	$1.79 \cdot 10^{308}$	$2.2 \cdot 10^{-308}$

Errore di arrotondamento: $\frac{|x - fl(x)|}{|x|} \leq \frac{1}{2}\varepsilon$

$\varepsilon = \frac{1}{2}\beta^{1-t} = 2^{-52} \gg \text{eps } 2.220446049250313\text{e-}016$

Nota. 53 cifre significative in base 2 corrispondono a **15 cifre significative** in base 10.

Matlab

Help: per informazioni sulle funzioni di Matlab (vedere anche lo help da menù)

```
>> help nome_funzione
```

informazioni su una specifica funzione

Esempio: come si usa la funzione log?

```
>> help log
```

LOG Natural logarithm.

LOG(X) is the natural logarithm of the elements of X.
Complex results are produced if X is not positive.

See also **LOG2**, **LOG10**, **EXP**, **LOGM**.

Matlab

Se non si conosce il nome delle funzione, si usa il comando **lookfor**

```
>> lookfor parola chiave
```

Produce l'elenco e la descrizione di tutte le funzioni che sono in qualche modo legate alla **parola chiave**

Esempio: esiste una funzione che produca la matrice identità di una certa dimensione?

```
>> lookfor 'identity matrix'
```

```
EYE Identity matrix.
```

```
SPEYE Sparse identity matrix.
```

A questo punto si chiede lo help della funzione che interessa, per es. **eye**

Matlab

Esempio: esiste una funzione che calcola la radice quadrata di un numero?

```
>> lookfor square
```

```
cir          - Cox-Ingersoll-Ross (CIR) mean-reverting square root diffusion
              class file
magic        - Magic square.
hypot        - Robust computation of the square root of the sum of squares
realsqrt     - Real square root.
sqrt         - Square root.
lscov        - Least squares with known covariance.
lsqnonneg    - Linear least squares with nonnegativity constraints.
sqrtm        - Matrix square root.
cgs          - Conjugate Gradients Squared Method.
.
.
```

```
>> help sqrt
```

```
SQRT Square root.
```

```
SQRT(X) is the square root of the elements of X. Complex
results are produced if X is not positive.
```

```
See also sqrtm, realsqrt, hypot.
```

```
Overloaded methods:
```

```
codistributed/sqrt
```

```
Reference page in Help browser
```

```
doc sqrt
```

Matlab

Digitando solo il comando **help** si ha l'elenco degli argomenti (pacchetti disponibili)

```
>> help
```

```
HELP topics:
```

```
matlab\general      - General purpose commands.
matlab\ops          - Operators and special characters.
matlab\lang         - Programming language constructs.
matlab\elmat        - Elementary matrices and matrix manipulation.
matlab\randfun      - Random matrices and random streams.
matlab\elfun        - Elementary math functions.
matlab\specfun      - Specialized math functions.
matlab\matfun       - Matrix functions - numerical linear algebra.
matlab\datafun      - Data analysis and Fourier transforms.
matlab\polyfun      - Interpolation and polynomials.
matlab\funfun       - Function functions and ODE solvers.
matlab\sparfun      - Sparse matrices.
```

```
>> help nome_argomento
```

Produce l'elenco e la descrizione delle funzioni relative all'argomento selezionato

Matlab

In Matlab non è necessario dichiarare le variabili. Esse vengono automaticamente definite in seguito ad una assegnazione

La assegnazione è data dal comando =

Esempio

```
>> d = 2;
```

attribuisce alla variabile **d** il valore **2** (verificare nel workspace)

```
>> c = 4;
```

attribuisce alla variabile **c** il valore **4**

```
>> b = c * d;
```

attribuisce alla variabile **b** il prodotto delle variabili **c** e **d**

Nota:

1. il nome di una variabile è composto da **caratteri alfanumerici**
2. il primo deve essere alfabetico
3. c'è differenza tra lettere maiuscole e minuscole

Matlab

Per conoscere tali variabili si può anche digitare il comando **whos**

```
>> whos      (Enter)
```

nome della variabile dimensione memoria occupata tipo

Esempio: se d è un numero intero

```
>> whos
```

Name	Size	Bytes	Class
d	1x1	1	int8 array

```
Grand total is 1 elements using 1 bytes
```

Matlab

Oppure si può anche digitare il comando **who**

```
>> who      (Enter)
```

lista delle variabili contenute nel workspace

Esempio: se **d** e **c** sono le sole due variabili in memoria

```
>> who
```

```
Your variables are:
```

```
c  d
```

Matlab

Per visualizzare il contenuto di una variabile, basta digitare il suo nome

Esempio: per visualizzare il contenuto di `b`

```
>> b
```

```
b =
```

```
8
```

Oppure usare il comando `disp`

```
>> disp(b)
```

```
8
```

Se si omette il punto e virgola; alla fine delle istruzioni di comando, viene visualizzato l'output di ogni istruzione

```
>> b=8*2
```

```
b =
```

```
16
```


Matlab

Se il risultato di un'espressione non viene assegnato ad una variabile definita dall'utente, allora viene automaticamente assegnato alla variabile **ans** (answer)

Esempio:

```
>> 3+2 (Enter)
```

```
ans =
```

```
5
```

Matlab

Per cancellare tutte le variabili contenute nel Workspace si usa il comando `clear`

Esempio:

```
>> b=5;, c=b*2-1;, d= c-b;
```

```
>> who
```

Your variables are:

```
b c d
```

```
>> clear
```

```
>> who
```

```
>>
```

Matlab

Per cancellare solo alcune variabili contenute nel Workspace, il comando `clear` deve essere seguito dall'elenco dei nomi delle variabili separati da uno spazio

```
clear b c      (cancella solo le variabili b e c )
```

Esempio:

```
>> b=5;, c=b*2-1;, d= c-b;
```

```
>> who
```

```
Your variables are:
```

```
b  c  d
```

```
>> clear b c
```

```
>> who
```

```
Your variables are:
```

```
d
```

Matlab

E' possibile salvare una o più variabili e riusarle in sessioni successive senza dover rieseguire i comandi con cui sono state create

`save nomefile`

`Salva` tutte le variabili contenute nel Workspace nel file `nomefile.mat`
Il `nomefile` è scelto dall'utente

`load nomefile`

`Carica` tutte le variabili salvate nel file `nomefile.mat` nel Workspace

Matlab

Esempio:

```
>> b=5;, c=b*2-1;, d= c-b;
```

```
>> who
```

Your variables are:

```
b c d
```

```
>> save datilezione
```

```
>> clear
```

```
>> who
```

```
>>
```

```
>> load datilezione
```

```
>> who
```

Your variables are:

```
b c d
```

Matlab

OSS: *Ad ogni variabile corrisponde il nome con cui è stata salvata!!!*
Se nel Workspace è presente una variabile con nome uguale a quello di una variabile contenuta nel file `nomefile.mat`, dopo l'istruzione `load` viene sostituita dalla variabile omonima in `nomefile.mat`

Esempio:

```
>> b=5;, c=b*2-1;, d= c-b;  
>> save datilezione  
>> clear  
>> b=25;  
>> disp(b)  
    25  
>> load datilezione  
>> disp(b)  
    5
```

Matlab

Per salvare nel file **nomefile.mat** solo alcune variabili, è necessario elencare tali variabili, separate da uno spazio, dopo il nomefile

```
save nomefile var1 var2 var3
```

Salva le variabili **var1**, **var2** e **var3** nel file **nomefile.mat**

Esempio:

```
>> b=5;, c=b*2-1;, d= c-b;
```

```
>> save datilezione b c
```

```
>> clear b c
```

```
>> who
```

```
Your variables are:
```

```
d
```

```
>> load datilezione
```

```
>> who
```

```
Your variables are:
```

```
b c d
```

Matlab

Nell'episodio *Treehouse of Horror* dei *Simpsons*, Homer ha un incubo un cui vede la seguente uguaglianza

$$1782^{12} + 1841^{12} = 1922^{12}$$

L'identità contraddice l'ultimo teorema di Fermat per il quale non esistono numeri naturali x, y, z per i quali sia vera l'identità

$$x^n + y^n = z^n \quad \forall n \geq 3$$

Homer ha sognato un controesempio al teorema di Fermat???

Digitiamo i seguenti comandi in Matlab

```
>> format short
```

```
>> (1782^12+1841^12)^(1/12)
```

ans =

1.9220e+003



Matlab

Digitiamo ora i seguenti comandi in Matlab

```
>> format long
```

```
>> (1782^12+1841^12)^(1/12)
```

```
ans =
```

```
1.921999999955867e+003
```

```
«Fermat near miss!!!»
```

Potete dire perché, senza svolgere calcoli, l'uguaglianza sognata da Homer non può essere vera?

Matlab

Indipendentemente dal sistema di rappresentazione dei numeri in Matlab, l'utente può scegliere il formato di visualizzazione usando il comando `format`

`format nomeformato`

Visualizza i numeri secondo il formato `nomeformato`

Digitare `help format` per conoscere tutti i formati di visualizzazione disponibili

Attenzione: il comando `format` non cambia la precisione con cui vengono eseguiti i calcoli !!!

Matlab

Esempi:

```
>> format short          % 4 cifre dopo la virgola (opzione di default)
>> sqrt(2)
ans =
    1.4142
```

```
>> format short e       % forma esponenziale (potenze di 10)
>> sqrt(2)
ans =
    1.4142e+000
```

```
>> format long         %      14 cifre dopo la virgola
>> sqrt(2)
ans =
    1.41421356237310
```

```
>> format long e       % forma esponenziale
>> sqrt(2)
ans =
    1.414213562373095e+000
```

Matlab

In un episodio successivo, Homer ci riprova con la seguente uguaglianza

$$3987^{12} + 4365^{12} = 4472^{12}$$

Homer ha finalmente trovato un controesempio al teorema di Fermat???

Potete dire perché, senza svolgere calcoli, anche questa uguaglianza non può essere vera?

Digitiamo i seguenti comandi in Matlab

```
>> format short
```

```
>> (3987^12+4365^12)^(1/12)
```

```
ans =
```

```
4.4720e+003
```

```
>> format long
```

```
>> (3987^12+4365^12)^(1/12)
```

```
ans =
```

```
4.4720000000007058e+003
```

Esercizi

- Assegnare alla variabile **a** il valore $4+2\log(\pi/2)/5$
- Calcolare in **b** il valore $e^{\cos(2.4)}$
- Calcolare in **c** il valore $b/4$
- Visualizzare **a**, **b**, **c** in formato corto esponenziale
- Visualizzare gli stessi valori in formato long. Poi tornare al formato di default
- Salvare le variabili **a** e **c** nel file **datilez1.mat**
- Assegnare alla variabile **d** l'espressione $\sin(\pi/3)^2+\cos(\pi/3)^2$
- Calcolare le seguenti espressioni

$$y = 6x^3 + \frac{4}{x} \quad x = 2, \quad y = 2 \frac{\sin(x)}{5} \quad x = 2, \quad y = 7x^{\frac{1}{3}} + 4x^{0.58} \quad x = 20$$

- Salvare il contenuto del Workspace in **work1.mat**; cancellare tutte le variabili nel Workspace; caricare il file **work1.mat**. Quali variabili sono contenute nel Workspace? E quale è il loro contenuto?

Matlab

Matlab ha classi di dati predefinite

- **double**: numeri in doppia precisione compresi tra -10^{308} e 10^{308} (8 bytes per elemento)
- **uint8**: interi a 8 bits per elemento senza segno compresi tra 0 e 255 (usato per le immagini)
- **uint16**: interi a 16 bits per elemento senza segno compresi tra 0 e 65535
- **uint32**: interi a 32 bits per elemento senza segno compresi tra 0 e 4294967295
- **int8**: interi a 8 bits per elemento con segno compresi tra -128 e 127
- **int16**: interi a 16 bits per elemento con segno compresi tra -32768 e 32767
- **int32**: interi a 32 bits per elemento con segno compresi tra -2147483648 e 2147483647
- **single**: numeri in singola precisione compresi tra -10^{38} e 10^{38} (4 bytes per elemento)
- **char**: caratteri (2 bytes per elemento)
- **logical**: 0 o 1 (1 byte per elemento)

Matlab

I nomi delle classi sono anche **funzioni** che permettono la conversione da una classe ad un'altra

Esempio: se x è una variabile **double**, il comando **int8(x)** converte x in una variabile intera a 8 bits

```
>> x=sqrt(2)
```

```
x =
```

```
1.414213562373095e+000
```

```
>> int8(x)
```

```
ans =
```

```
1
```

```
>> int8(x*10)
```

```
ans =
```

```
14
```

Matlab

Esempio:

```
>> x =  
      349021
```

```
>> int8(x)  
ans =  
     127
```

```
>> int16(x)  
ans =  
    32767
```

```
>> int32(x)  
ans =  
    349021
```


Matlab

Esempio:

```
>> x=-cos(3/4*pi)
```

```
x =
```

```
7.071067811865475e-001
```

```
>> y=sin(3/4*pi)
```

```
y =
```

```
7.071067811865476e-001
```

```
>> z=x^2+y^2
```

```
z =
```

```
1
```

```
>> xi=int8(x)
```

```
xi =
```

```
1
```

```
>> yi=int8(y)
```

```
yi =
```

```
1
```

```
>> zi=xi^2+yi^2
```

```
z =
```

```
2
```

La rappresentazione dei numeri può essere molto costosa!!!

Ariane 5

Tempi di realizzazione: 10 anni

Costo: 7 bilioni di dollari

Primo volo dell' Ariane 5: giugno 1996

Durata della missione: 40 secondi

Motivo: autodistruzione del razzo per un malfunzionamento del software di controllo

Perché? Un dato a 64 bit in virgola mobile venne convertito in un intero a 16 bit con segno  operazione errata del processore: il numero in virgola mobile era troppo grande per poter essere rappresentato con un intero a 16 bit.

Questo errore scatenò una reazione a catena che causò danni meccanici ai quali seguì l'azionamento del comando di autodistruzione.

Fu necessario quasi un anno e mezzo per capire quale fosse stato il malfunzionamento che aveva portato alla distruzione del razzo!

La rappresentazione dei numeri può essere molto costosa!!!

Come se la massa della terra

```
>> M = 597420000000000000000000000000
```

M =

5.9742000000000000e+024

diventasse

```
>> int16(M)
```

ans =

32767

Matlab

I caratteri **char** si indicano tra 2 apici

Esempio: attribuire alla variabile **A** il carattere **f**

```
>> A = 'f' ;
```

```
>> disp(A)
```

```
f
```

Vettori

Un **array** è un insieme di valori ordinati, secondo uno o più indici, a cui ci si riferisce con un **singolo nome di variabile**

Un **array ad un indice** è detto **vettore**

Un **array a due indici** è detto **matrice**

In Matlab si possono definire facilmente **vettori** e **matrici**

Le **variabili** in Matlab hanno una **struttura vettoriale**, per esempio gli **scalari** sono **matrici** di dimensione **1x1**

Vettori

Un vettore si definisce elencando le sue componenti separate da uno spazio e racchiudendole tra parentesi quadre []

Vettore riga

```
>> x = [10 20 30 40]
```

```
x =
```

```
    10    20    30    40
```

è equivalente a

```
>> x = [10,20,30,40]
```

```
x =
```

```
    10    20    30    40
```

In questo caso le componenti sono separate da una virgola

Vettori

Vettore colonna

```
>> x=[10; 20; 30; 40]
```

```
x =
```

```
10
```

```
20
```

```
30
```

```
40
```

In questo caso le componenti sono separate da un **punto e virgola**

Anche per visualizzare il contenuto di variabili che sono vettori si può usare il comando **disp**

```
>> disp(x)
```

```
10
```

```
20
```

```
30
```

```
40
```

Vettori

Per convertire un vettore riga in una colonna (e viceversa) si usa il comando ' (apice) che produce il trasposto della variabile a cui è applicato

```
>> v=x'
```

```
v =
```

```
    10    20    30    40
```

Per estrarre un elemento di un vettore:

nome_vettore(posizione elemento)

Esempio: estrarre il secondo elemento di **v**

```
>> v(2)
```

```
ans =
```

```
    20
```

Nota: Gli **indici** di un vettore sono sempre **numeri interi e strettamente positivi**

Vettori - la notazione :

Per estrarre contemporaneamente più di un elemento di un vettore si usa il comando `:` (colon)

`nome_vettore(inizio:fine)`

Esempio: estrarre dal primo al terzo elemento di `v`

```
>> v(1:3)
```

```
ans =
```

```
    10    20    30
```

Esempio: estrarre dal terzo al quarto elemento di `v`

```
>> v(3:4)
```

```
ans =
```

```
    30    40
```

Vettori - la notazione :

Esempio: estrarre dal secondo all'ultimo elemento di v

```
>> v(2:end)
```

```
ans =
```

```
    20    30    40
```

Esempio: estrarre tutti gli elementi di v

```
>> v(1:end)
```

```
ans =
```

```
    10    20    30    40
```

Oppure, se lo si vuole come vettore colonna,

```
>> v(:)
```

```
ans =
```

```
    10
```

```
    20
```

```
    30
```

```
    40
```

Vettori - la notazione :

Per estrarre contemporaneamente più di un elemento di un vettore non consecutivi ed equispaziati

`nome_vettore(inizio:passo:fine)`

Esempio: estrarre gli elementi di `v` di indice pari (`passo = 2`)

```
>> v(2:2:end)
```

```
ans =
```

```
    20    40
```

Esempio: estrarre tutti gli elementi di `v` di indice pari ma da destra verso sinistra (`passo = -2`)

```
>> v(end:-2:1)
```

```
ans =
```

```
    40    20
```

Vettori

Un vettore può essere usato per estrarre elementi non consecutivi e non equispaziati di un altro vettore

```
nome_vettore([pos1 pos2 pos3 ...])
```

Esempio: Sia $v = [7 \ 1 \ 3 \ 7 \ 0 \ 8 \ 3]$, estrarre gli elementi di v di indici 1 3 e 6

```
>> v([1 3 6])
```

```
ans =
```

```
    7     3     8
```

Vettori - la notazione :

Il comando `:` può essere usato anche per generare vettori

Nome_vettore = (minimo:incremento:massimo)

Nome_vettore = vettore di elementi equispaziati (di una quantità=**incremento**) nell'intervallo [**minimo, massimo**]

Esempio: Generare un vettore costituito da elementi compresi tra 1.5 e 2.5 con incremento 0.1

```
>> x=[1.5:0.1:2.5]
```

```
x =
```

```
1.5    1.6    1.7    1.8    1.9    2.0    2.1    2.2    2.3    2.4    2.5
```

Esempio: Generare un vettore costituito da elementi compresi tra 100 e 80 con incremento -5

```
>> x=[100:-5:80]
```

```
x =
```

```
100    95    90    85    80
```

Se non specificato, l'incremento è da intendersi pari a 1

Vettori

Per generare vettori equispaziati contenuti in un certo intervallo si può usare anche il comando `linspace`

`Nome_vettore = linspace(minimo, massimo, numero di elementi)`

`Nome_vettore` è un vettore di lunghezza pari a `numero di elementi`, i cui elementi sono numeri equispaziati nell'intervallo `[minimo, massimo]`

Gli elementi del vettore distano della quantità

$$h = (\text{massimo} - \text{minimo}) / (\text{numero di elementi} - 1)$$

e la i -esima componente è `Nome_vettore(i) = minimo + (i-1)*h`

Esempio: Generare un vettore di 10 elementi compresi tra 1.5 e 2.4

```
>> x=linspace(1.5,2.4,10)
```

```
x =
```

```
1.5    1.6    1.7    1.8    1.9    2.0    2.1    2.2    2.3    2.4
```

In questo caso l'incremento `h=0.1`, infatti

$$h = (\text{massimo} - \text{minimo}) / (\text{numero di elementi} - 1) = (2.4 - 1.5) / (10 - 1) = 0.9 / 9 = 0.1$$

Vettori

Se un vettore (o una qualsiasi istruzione) è troppo lungo, prima di andare a capo vanno aggiunti 3 punti ...

```
>> x = [3 1 6 7 9 10 4 29 6 0 ...  
        4 5 8 2 4 ]
```

```
x =  
    3    1    6    7    9   10    4   29    6    0    4    5    8    2    4
```

Se un elemento di un vettore è una espressione, **non bisogna lasciare spazi all'interno dell'elemento**, oppure l'espressione va racchiusa tra parentesi tonde

```
>> x = [1  6  3*2+1  4]
```

```
x =  
    1    6    7    4
```

Oppure

```
>> x = [1  6  (3*2+1)  4]
```

```
x =  
    1    6    7    4
```

Vettori

I vettori non vengono dimensionati. **La loro dimensione può essere modificata in corso di lavoro**

Esempio: Sia $x = [3 \ 1 \ 4 \ 5]$ e si assegni il valore **10** all'ottavo elemento di x

```
>> x = [3 1 4 5]
```

```
x =
```

```
    3    1    4    5
```

```
>> x(8) = 10
```

```
x =
```

```
    3    1    4    5    0    0    0    10
```

alle posizioni non definite viene assegnato il valore zero

Vettori

Esempio: Sia $x = [3 \ 1 \ 4 \ 5]$ e si elimini l'elemento in posizione 3

```
>> x(3) = []
```

```
x =
```

```
     3     1     5
```

[] indica il vettore vuoto

Per conoscere la lunghezza di un vettore si usa il comando `length(x)`

Esempio: determinare la lunghezza del vettore x sopra definito

```
>> length(x)
```

```
ans =
```

```
     3
```

Esercizi

- 1) Assegnare alla variabile x il vettore costituito dai primi 20 numeri naturali. Estrarne il quarto elemento e moltiplicarlo per il quindicesimo.
- 2) Creare un vettore x che ha 6 componenti con valori compresi tra 0 e 10, estremi inclusi, con incremento costante.
- 3) Creare un vettore x che ha componenti con valori compresi tra 0 e 10, estremi inclusi, con incremento costante pari a 0.2.
 - Determinarne la lunghezza.
 - Estrarne gli elementi di indice pari e assegnarli alla variabile y .
 - Estrarre gli elementi di indice dispari di y procedendo da destra verso sinistra e assegnarli alla variabile z .
 - Eliminare il primo elemento di z .
 - Assegnare il valore -9 al ventesimo elemento di z .
 - Sostituire il secondo elemento di z con $2 \cdot \cos(\pi/4)$
 - **Creare il vettore w costituito da tutti gli elementi di y seguiti dai primi 2 elementi di z e gli ultimi 3 di x .**
 - Visualizzare il vettore w .

Matrici

Una **matrice** si può definire come un insieme di **vettori riga** separati da un punto e virgola oppure di **vettori colonna** separati da uno spazio o una virgola

```
>> A = [2 5 7; 1 7 9; 8 1 -4];
```

```
A =
```

```
    2     5     7
    1     7     9
    8     1    -4
```

```
>> A = [[2;1;8] [5;7;1] [7;9;-4]]
```

```
A =
```

```
    2     5     7
    1     7     9
    8     1    -4
```

Matrici

Per estrarre un elemento della matrice si usa

```
nome_matrice(indice_riga, indice_colonna)
```

indice_riga e indice_colonna devono essere interi strettamente positivi!

Esempio: Estrarre l'elemento di A di indici (2,3)

```
>> A(2,3)
```

```
ans =
```

```
9
```

Esempio: Estrarre la terza colonna di A

```
>> A(:,3)
```

```
ans =
```

```
7
```

```
9
```

```
-4
```

Matrici

Esempio: Estrarre la prima riga di A

```
>> A(1, :)
```

```
ans =
```

```
     2     5     7
```

$A(i, j)$	restituisce l'elemento con indice di riga i e indice di colonna j della matrice A
$A(i, :)$	restituisce il vettore riga corrispondente alla riga i della matrice A
$A(:, j)$	restituisce il vettore colonna corrispondente alla colonna j della matrice A
$A(i, m:p:n)$	restituisce un vettore riga contenente gli elementi nelle colonne da m a n con passo p della i -esima riga della matrice A

Matrici

Esempio: estrarre il vettore riga contenente gli elementi nelle colonne 1,3 della terza riga della matrice A

```
>> A(3,1:2:3)
```

```
ans =
```

```
8      -4
```

Esempio: estrarre il vettore colonna contenente gli elementi nelle righe 2,4,6 della quarta colonna di A

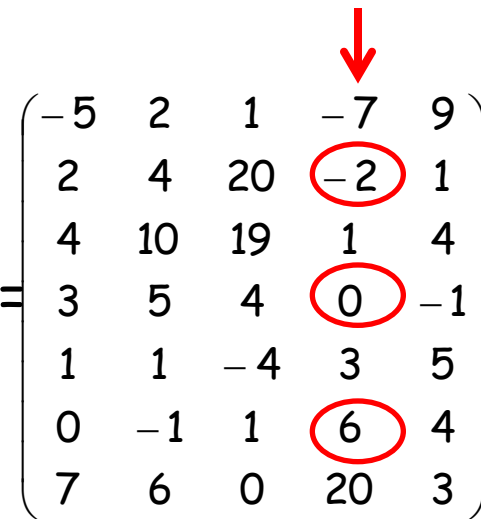
```
>> A(2:2:end,4)
```

```
ans =
```

```
-2
```

```
0
```

```
6
```


$$A = \begin{pmatrix} -5 & 2 & 1 & -7 & 9 \\ 2 & 4 & 20 & -2 & 1 \\ 4 & 10 & 19 & 1 & 4 \\ 3 & 5 & 4 & 0 & -1 \\ 1 & 1 & -4 & 3 & 5 \\ 0 & -1 & 1 & 6 & 4 \\ 7 & 6 & 0 & 20 & 3 \end{pmatrix}$$

Matrici

Estrazione di sottomatrici : $A(m:k, i:j)$ indica la sottomatrice di A con le righe da m a k , e le colonne da i a j .

$A(:, i:j)$ sottomatrice delle colonne da i a j .

$A(m:k, :)$ sottomatrice delle righe da m a k .

Esempio: Sia $A=[1\ 2\ 0; 3\ 4\ 1; 2\ 6\ 3]$, estrarre la sottomatrice con indici di riga da 2 a 3 e di colonna da 1 a 3

```
>>b=A(2:3,1:3)
```

```
b= 3  4  1
    2  6  3
```

Nota: In questo caso, questa istruzione è equivalente a

```
>>b=A(2:3, :)
```

ovvero

```
>>b=A(2:3, 1:end)
```

Matrici

Per conoscere la **dimensione di una matrice** di usa il comando **size()**

size(A) restituisce un vettore di 2 componenti di cui la prima indica il numero di righe mentre la seconda il numero di colonne

Esempio: Se $A = [3 \ 5 \ 8; -1 \ 2 \ 4]$

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

```
ans =
```

2

3



num.righe

num.colonne

Esercizi

- Costruire una matrice 5×5 , ed assegnarla alla variabile A
- Estrarre da A la sottomatrice 2×2 costituita dalla seconda e terza riga e dalla seconda e terza colonna di A ed indicare con D la matrice ottenuta
- Estrarre gli elementi sulla diagonale principale di D e assegnarli al vettore d
- Estrarre gli elementi che non appartengono alla diagonale principale di D e assegnarli al vettore f
- Estrarre gli elementi della seconda e della terza colonna di A e indicare con B la matrice ottenuta
- Determinare la dimensione di B
- Estrarre gli elementi sulla diagonale principale di B e assegnarli al vettore b
- Estrarre gli elementi che non appartengono alla diagonale principale di B e assegnarli al vettore c
- Determinare la dimensione di b e c