

# Calcolo Numerico

A.A. 2013-2014

## Esercitazione n. 1

07-03-2014

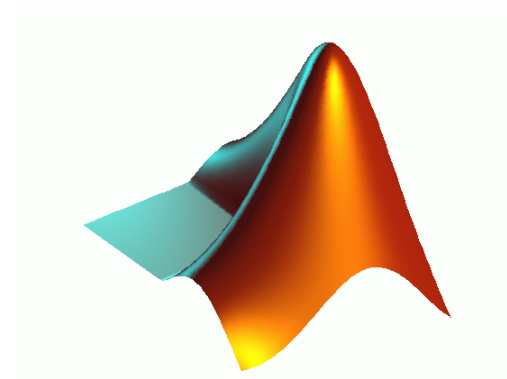
# Info

Annalisa Pascarella

- email
  - [a.pascarella@iac.cnr.it](mailto:a.pascarella@iac.cnr.it)
- webpage
  - <http://www.iac.rm.cnr.it/~pasca>
- materiale del corso
  - slide delle esercitazioni, programmi MATLAB
- laboratorio? quanti siamo?
- ricevimento?

# MATLAB/OCTAVE

- Introduzione a Matlab
- Rappresentazione dei numeri
- Underflow e overflow
- Vettori
- Cenni di programmazione
- Script
- Grafica 2D
- Cancellazione numerica
- Errori di arrotondamento

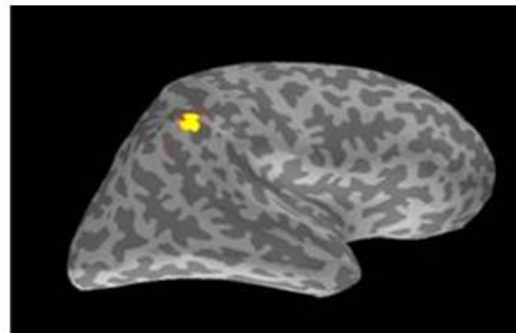
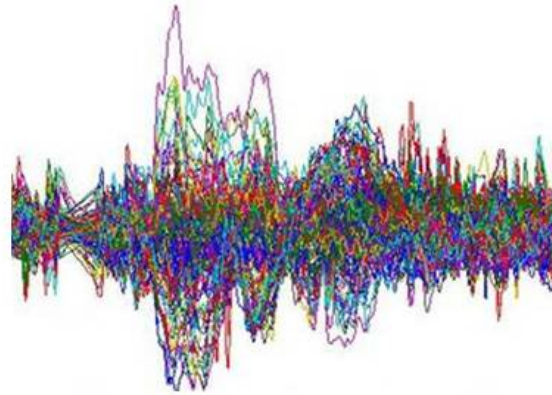
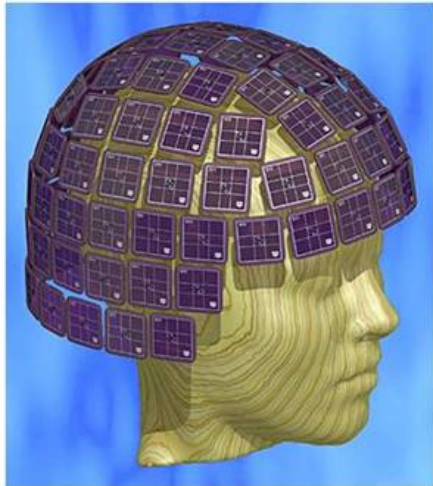


# Calcolo numerico

Che cos'è il calcolo numerico?

- è quella branca della matematica che costruisce e analizza i **metodi numerici** adatti a risolvere, con l'aiuto del calcolatore, differenti **problemi matematici** che nascono in varie discipline: ingegneria, biologia, medicina, ...

# MEG

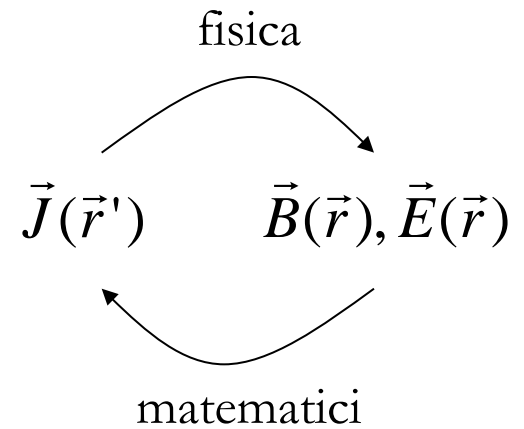


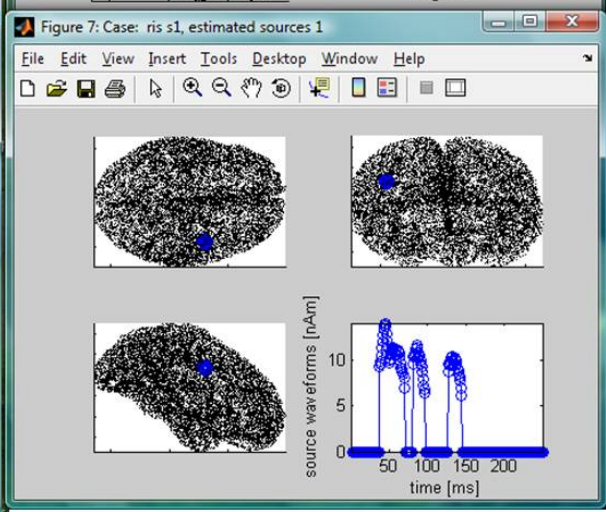
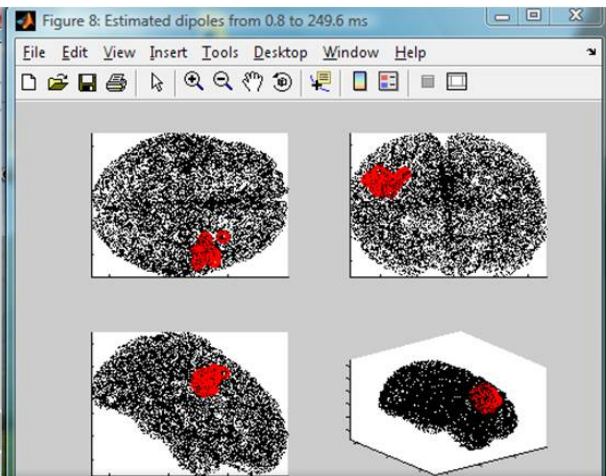
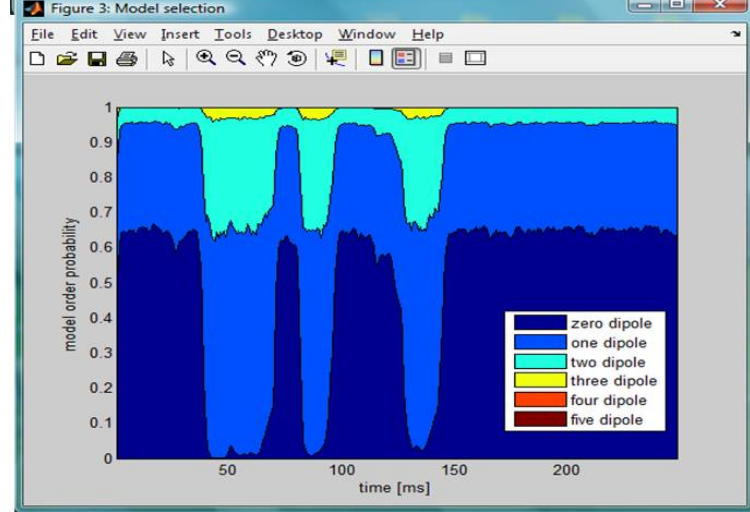
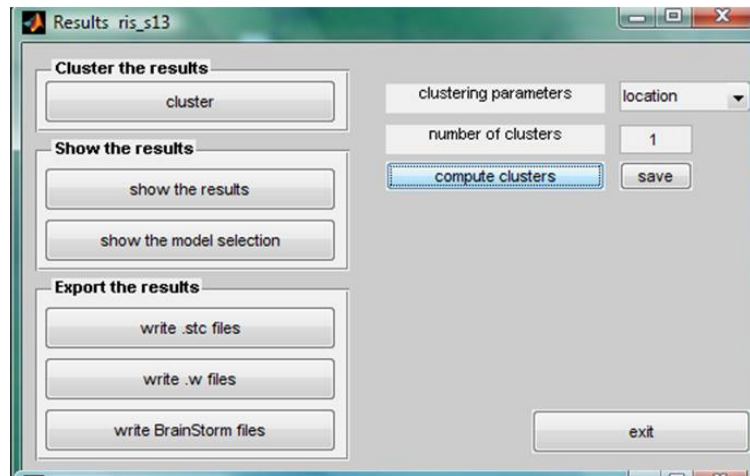
$$\mathbf{g} = \mathbf{A}\mathbf{f}$$

$\mathbf{g}$  -> magnetic field

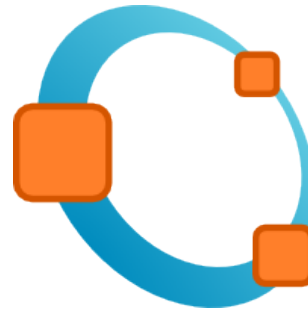
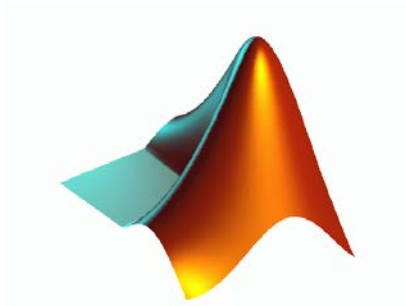


$\mathbf{f}$  -> neural currents





# MATLAB/OCTAVE



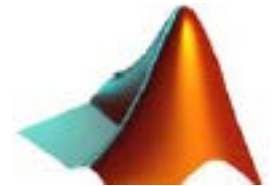
# MATLAB

- **Matlab (MATrix LABoratory)** è un sistema software integrato per il calcolo scientifico sviluppato a partire dagli anni '70 utilizzabile sia in maniera interattiva che come linguaggio di programmazione.
- La struttura dati di base è la **matrice**, per la quale sono già predefinite **numerosi tipi elementari** (matrice identità, matrice nulla, matrice unità...), **funzioni algebriche e di manipolazione** (somma, prodotto, calcolo del determinante).
- Fornisce un ambiente di calcolo, visualizzazione e programmazione scientifica in cui è possibile:
  - calcolare direttamente espressioni matematiche
  - utilizzare il semplice ambiente di programmazione per creare i propri algoritmi
  - creare **grafici** in 2 e 3 dimensioni



# Altre funzionalità di MATLAB

- Esistono vari pacchetti per i diversi tipi di applicazioni (**Toolbox**): l'elaborazione numerica dei segnali e delle immagini, la simulazione di sistemi dinamici, il calcolo simbolico, wavelet, ecc.
- Per informazioni su Matlab: [www.mathworks.com](http://www.mathworks.com)
- Matlab è un software a pagamento. **Octave** è un software gratuito che ne riproduce buona parte delle funzioni fondamentali. Per info vedere [www.gnu.org/software/octave/](http://www.gnu.org/software/octave/)



# MATLAB linguaggio per programmare

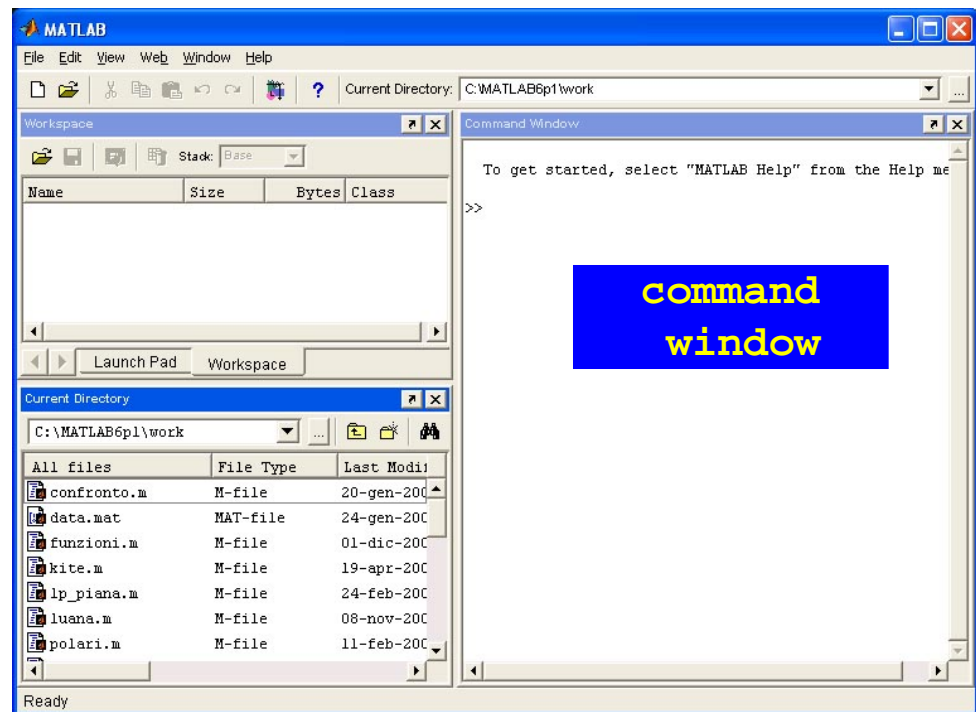
- E' un **linguaggio interpretato e non compilato**: questo significa che le istruzioni vengono tradotte in linguaggio macchina (il linguaggio “capito” dal processore) e subito eseguite una per volta.
  - questo fatto si riflette in una **maggiore lentezza** di esecuzione rispetto ad un linguaggio compilato
  - Matlab possiede istruzioni molto potenti per la manipolazione di vettori e matrici. Se si riesce a fare ricorso il più possibile a queste istruzioni, evitando di utilizzare cicli for che, come vedremo, accedono singolarmente a ciascun elemento di un vettore o di una matrice, i tempi di esecuzione miglioreranno sensibilmente
- Utilizzando C, Fortran, C++, la traduzione da linguaggio ad alto livello a linguaggio macchina avviene invece nel processo di compilazione, in cui tutto il programma viene tradotto in linguaggio macchina e poi eseguito.

# MATLAB

- Per lanciare Matlab da ambiente **Windows** basta cliccare con il mouse sull'icona corrispondente
- In ambiente **Unix**, digitare il comando `matlab` e quindi dare il comando di invio
- All'avvio viene aperta una finestra (spazio di lavoro) nella quale è possibile digitare comandi dal **prompt** `>>`
- Per entrare in confidenza con l'ambiente di lavoro è utile:
  - lanciare il comando **demo** che illustra le potenzialità del software attraverso significativi esempi numerici e casi test;
  - fare costante riferimento all'uso dell'**help**, ad esempio `help sqrt` (calcolo della radice quadrata di un numero).

# MATLAB - Finestre

- **Command window:** finestra principale DI LAVORO INTERATTIVA. Contiene il **prompt >>**
- **Command history:** contiene tutti i comandi digitati da prompt per ripeterli, basta selezionare il comando di interesse con un click
- **Workspace:** contiene tutte le variabili usate e memorizzate automaticamente



# MATLAB

Linguaggio di programmazione interpretato: legge un comando per volta eseguendolo immediatamente. Matlab lavora in **modo interattivo**

- l'utente digita una istruzione sul **prompt >>** ed ha immediatamente la risposta

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

## **Esempio**

**>> 3+2 (Enter)**

**ans =**

**5**

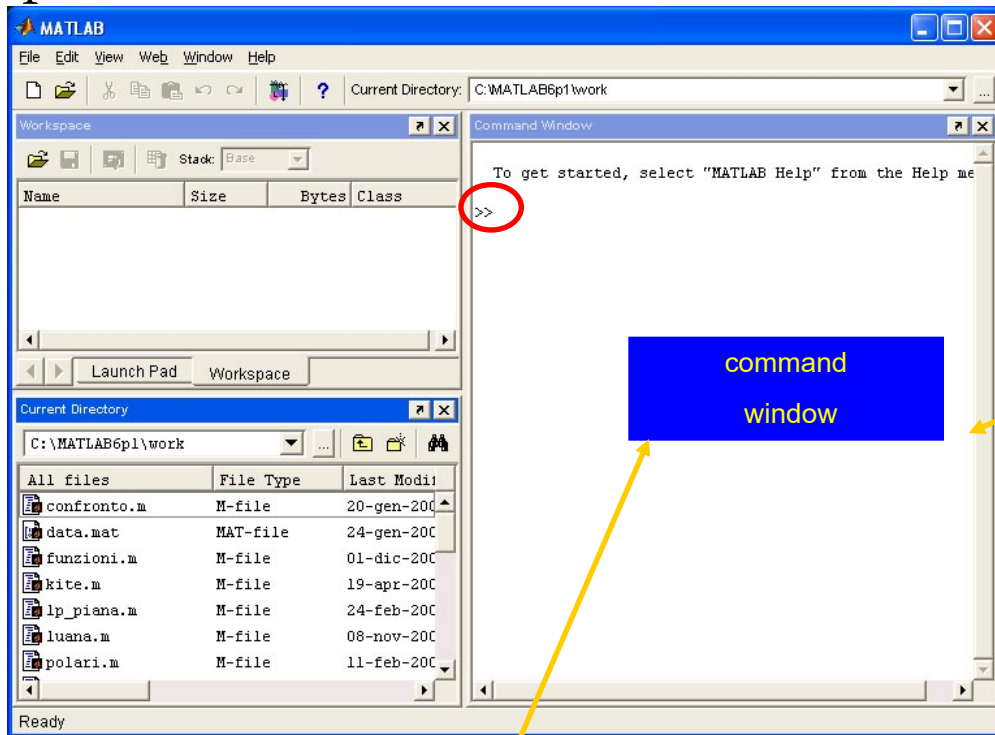
**>>**

Per uscire dalla sessione di lavoro interattiva:

**>> quit**

# MATLAB come calcolatrice

- Può essere utilizzato in modo diretto per calcolare semplici espressioni matematiche:



4 + 7

Enter

- E' possibile definire **variabili** e operare su esse

x = 9 -> Enter

# Variabili

- Una **variabile** è un insieme di dati modificabili
  - situati in una porzione di memoria destinata a contenere dei dati
  - suscettibili di modifica nel corso dell'esecuzione di un programma
- Ogni variabile è caratterizzata da un nome
  - una variabile è un contenitore, identificato da un nome univoco, di un qualsiasi valore, sia esso numero o stringa
  - anche in Matlab è possibile definire variabili
  - tipi di variabili
  - variabili di tipi diversi occupano spazi di memoria differenti
- Le variabili sono quindi definite da un **tipo** e da un **nome**

# Variabile ans

- Se alle espressioni non si assegna una variabile la risposta è del tipo **ans=...**, dove **ans** è una variabile generata automaticamente da MATLAB.

```
>> 3+2
```

```
ans =
```

```
5
```

- In Matlab non è necessario definire 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!)



# Variabili - definizione

Le variabili definite dall'utente possono essere utilizzate in altre espressioni matematiche

```
>> a = 8 + 3 - 1  
a =  
10
```

```
>> b = 1/sqrt(2)  
b =  
1.4142
```

```
>> c = a + b  
c =  
11.4142
```

**Nota:** c'è differenza tra maiuscole e minuscole

# Variabili - visualizzazione

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
```

- le variabili sono sovrascrivibili
- per cancellare una variabile si usa il comando `clear`
- le variabili predefinite possono essere ridefinite (esempio `i`)

# Variabili - whos

Le variabili sono create automaticamente al momento del loro uso. Per conoscere le variabili create si può digitare il comando **whos** che da anche indicazioni utili sullo spazio occupato in memoria dalle variabili

```
>> 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
```

# Variabili

- E' importante sottolineare che, a differenza di altri linguaggi di programmazione, non è necessario specificare il tipo della variabile usata, ma questo risulta implicitamente definito in seguito all'assegnazione dei valori che deve assumere
- Tutte le variabili numeriche sono memorizzate in Matlab in formato **double**, ovvero in doppia precisione
- **clear nome\_variabile**
  - per cancellare una variabile dal workspace
- **clear all**
  - per cancellare tutte le variabili dal workspace
- **clc**
  - pulisce lo schermo

# Variabili

- Più espressioni o più istruzioni MATLAB possono convivere sulla stessa riga, basta che siano separate da un punto e virgola (se vogliamo sopprimere la visualizzazione di qualche risultato) oppure da una virgola (se vogliamo visualizzare il risultato)
  - per evitare la visualizzazione del risultato basta concludere l'espressione con un **punto e virgola**
  - per visualizzare il contenuto di una variabile è sufficiente scriverne i nomi eventualmente separati con una **virgola**

## Esempio

```
>> a=2+3/5; b=sqrt(a), c=a*b
```

```
b =
```

```
1.6125
```

```
c =
```

```
4.1924
```

# Diary

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 sul file di testo

Si può memorizzare come **Script** (vedremo in seguito) escludendo tutti gli output generati e gli eventuali messaggi di errore.

# Save

- Per conservare il contenuto delle variabili si deve invece salvare tutta l'area di memoria (o parte di essa) con il comando **save**. Ad esempio:

```
>> x = 1;  
>> a = 3;  
>> z = sqrt(a)-x;  
>> save areawork  
>> save variables x z
```

I file prodotti (binari) hanno estensione **.mat**. Per ricaricarli nello spazio di lavoro usiamo il comando **load**:

```
>> load areawork  
>> load variables
```





# MATLAB come calcolatrice

## Costanti predefinite

---

Variabile	Significato
<code>ans</code>	valore ultima operazione eseguita non assegnata a variabile
<code>i, j</code>	unità immaginaria
<code>pi</code>	$\pi$ , 3.14159265...
<code>eps</code>	precisione di macchina
<code>realmax</code>	massimo numero macchina positivo
<code>realmin</code>	minimo numero macchina positivo
<code>Inf</code>	$\infty$ , ossia un numero maggiore di <code>realmax</code>
<code>NaN</code>	Not a Number, tipicamente il risultato di un'espressione 0/0

---

numero massimo rappresentabile ( $2^{1023}$ ) `realmax`

numero minimo rappresentabile ( $2^{-1022}$ ) `realmin`

Nonostante sia possibile ridefinire tali variabili è buona regola non farlo, eccezion fatta per le variabili `i, j`

# MATLAB come calcolatrice

**Funzioni predefinite** (Oltre alle operazioni di base, molte delle funzioni presenti in una calcolatrice scientifica sono presenti anche in MATLAB). Per una lista completa digitare **help elfun**

---

Funzione	Significato
<code>sin</code>	seno
<code>cos</code>	coseno
<code>asin</code>	arcoseno
<code>acos</code>	arcocoseno
<code>tan</code>	tangente
<code>atan</code>	arcotangente
<code>exp</code>	esponenziale
<code>log</code>	logaritmo naturale
<code>sqrt</code>	radice quadrata
<code>abs</code>	valore assoluto
<code>sign</code>	la funzione segno

---

Alcune funzioni predefinite in MATLAB

Una **funzione** necessita di alcuni parametri in ingresso, elencati tra parentesi tonde, e solitamente restituisce un risultato che può essere assegnato ad una variabile.

Per esempio l'espressione  
**>> y=cos(pi/4)**

**y =  
0.7071**

utilizza la funzione coseno con argomento  $\pi/4$  e ne assegna il risultato alla variabile `y`.

# Help!

Per informazioni sulle funzioni di Matlab (vedere anche l' 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.
```

# Help!

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\elfun        - Elementary math functions.
matlab\specfun      - Specialized math functions.
matlab\matfun       - Matrix functions - numerical linear algebra.
matlab\datafun      - Data analysis and Fourier transforms.
matlab\audio        - Audio support.
```

```
>> help nome_argomento
```

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

# Lookfor

Se non si conosce il nome della 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 l'help della funzione che interessa

# Format

- Il risultato dell'operazione `cos(pi/4)` è visualizzato con quattro cifre decimali, questa è l'impostazione standard di Matlab
- In Matlab tutte le variabili sono in doppia precisione (forma a virgola mobile, floating point), ovvero sono rappresentate internamente con 64 bit, cui corrispondono 16 cifre significative decimali.
- Tutti i calcoli vengono effettuati in doppia precisione, mentre diversa è la visualizzazione delle variabili che viene determinata con il comando `format`.
- Il formato esterno può essere deciso dall'utente: (`help format`)
- Tale comando serve per modificare il formato di visualizzazione dei risultati ma NON la precisione con cui i calcoli vengono condotti.<sup>30</sup>

# Format

```
>> format short % 4 cifre significative dopo la virgola  
(opzione di default)
```

```
>> sqrt(2)
```

```
ans =  
    1.4142
```

```
>> format short e % forma esponenziale (potenze di 10)
```

```
>> exp(10)
```

```
ans =  
    2.2026e+004
```

```
>> format long % 14 cifre dopo la virgola
```

```
>> sqrt(2)
```

```
ans =  
    1.41421356237310
```

```
>> format long e % forma esponenziale
```

```
>> exp(10)
```

```
ans =  
    2.202646579480672e+004
```

# Precisione macchina

- Eps prende il nome di **precisione macchina** ed è legata all'approssimazione ottenibile con l'insieme dei numeri macchina a disposizione
  - rappresenta quella costante caratteristica di ogni aritmetica floating-point ed è la massima precisione con cui vengono effettuati i calcoli sul calcolatore; è il più piccolo numero sentito dall'aritmetica dei numeri macchina
- **eps** è il minimo valore tale che  $(1+eps) > 1$ 
  - è il più numero che sommato a 1 da un numero maggiore di 1
- Sul calcolatore che stiamo usando

```
>> eps
```

```
?
```

```
>> 1+eps/2 -1
```

```
0
```



# Precisione macchina

- Riassumendo, bisogna prestare molta attenzione quando si effettuano operazioni di calcolo tramite un calcolatore o una semplice calcolatrice tascabile. Infatti sarà inevitabile la presenza di errori nel risultato dovuta all'approssimazione numerica.

# Classi di dati

- **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)

# Classi di dati

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

**Esempio:** se `x` è una variabile double

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

converte `x` in una variabile intera

I caratteri `char` si indicano tra 2 apici

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

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

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

```
f
```

# Trucchetti...

- Durante la sessione di lavoro è possibile richiamare i comandi precedentemente digitati utilizzando il tasto ←
- Immettendo i primi caratteri di un'istruzione già digitata e poi premendo il tasto ⇔ viene completata la riga con l'ultima istruzione che inizia con quegli stessi caratteri;
- Con il tasto sinistro del mouse sulla finestra di calcolo si possono selezionare parti di testo che è poi possibile copiare, tagliare ed incollare sulla linea di comando.

# Numeri complessi

- I numeri in virgola mobile (la rappresentazione che un calcolatore fa dei numeri reali) non sono l'unico tipo di dato numerico ammesso. Un altro tipo di dato utile in varie applicazioni e presente in MATLAB sono i **numeri complessi** e le operazioni con questi.
- Un numero complesso  $z$ , in forma algebrica  $\mathbf{z} = \mathbf{Re}(z) + \mathbf{iIm}(z)$  (con  $\mathbf{Re}(z)$ ,  $\mathbf{Im}(z)$  parte reale e parte immaginaria), può essere scritto in modo simile anche in MATLAB:

```
>> a=3+4i;
```

- L'utilizzo di operazioni su numeri complessi è ammesso

```
>> a=3+2i;
```

```
>> b=3.6+2.4*i;
```

```
>> a+b
```

```
ans =
```

```
6.6000 + 4.4000i
```

# 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
- Se **x=5**, **y=3**, **z=2** calcolare  $(3x-4)^2/(5y-z)$
- Calcolare  $y = \frac{e^{x+1} - \sqrt{(x+1)^3}}{x^3 + \ln x + 1}$  per **x=10**