

# Cenni introduttivi sul “laboratorio matriciale” MATLAB

http://matlab/math/multidimensional-arrays.html

Search

Documentation Examples Functions Videos Answers

row

column

page

(1,1,1)	(1,2,1)	(1,3,1)	(1,4,1)
(2,1,1)	(2,2,1)	(2,3,1)	(2,4,1)
(3,1,1)	(3,2,1)	(3,3,1)	(3,4,1)
(4,1,1)	(4,2,1)	(4,3,1)	(4,4,1)

(1,1,2)	(1,2,2)	(1,3,2)	(1,4,2)
(2,1,2)	(2,2,2)	(2,3,2)	(2,4,2)
(3,1,2)	(3,2,2)	(3,3,2)	(3,4,2)
(4,1,2)	(4,2,2)	(4,3,2)	(4,4,2)

(1,1,3)	(1,2,3)	(1,3,3)	(1,4,3)
(2,1,3)	(2,2,3)	(2,3,3)	(2,4,3)
(3,1,3)	(3,2,3)	(3,3,3)	(3,4,3)
(4,1,3)	(4,2,3)	(4,3,3)	(4,4,3)

Trieste, 03.10.2022

**Creating Multidimensional Arrays**

You can create a multidimensional array by creating a 2-D matrix first, and then extending it. |

**MATLAB** è un

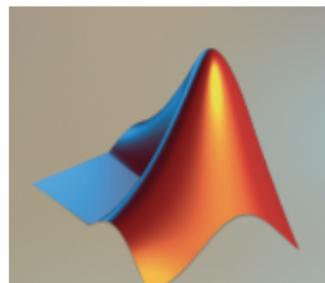
- linguaggio di programmazione incentrato sugli *array* (i.e., vettori e matrici, anche multi-dimensionali o annidati);
- ambiente di sviluppo costituito di finestre, menú a tendina, *dock* (una banchina di 'attracco').

# DI CHE SI TRATTA ?

**MATLAB** è un

- linguaggio di programmazione incentrato sugli *array* (i.e., vettori e matrici, anche multi-dimensionali o annidati);
- ambiente di sviluppo costituito di finestre, menú a tendina, *dock* (una banchina di 'attracco').

L'ambiente integra con il linguaggio di programmazione: *editor*, spazio di lavoro, interfaccia grafica ('*GUI*'), strumenti di aiuto ('*help*', '*doc*', '*lookfor*'),...



**MATLAB** è un

- linguaggio di programmazione incentrato sugli *array* (i.e., vettori e matrici, anche multi-dimensionali o annidati);
- ambiente di sviluppo costituito di finestre, menú a tendina, *dock* (una banchina di 'attracco').

L'ambiente integra con il linguaggio di programmazione: *editor*, spazio di lavoro, interfaccia grafica ('*GUI*'), strumenti di aiuto ('*help*', '*doc*', '*lookfor*'),...

## ESERCIZIO

Impartire i comandi:

- » `help doc`
- » `doc help`
- » `lookfor('square root')`, `lookfor('prime number')`
- » `why`

# COME SI PRESENTA L'AMBIENTE:



The screenshot displays the MATLAB R2018b academic use environment. The Editor window shows a script named 'matrix.m' with the following code:

```
1 - clear all
2 - close all
3 - clc
4 - |
5 - %Massimo intervallo
6 - l=1000;
7 - %Numero righe matrice
8 - m=100;
9 - %Numero colonne matrice
10 - n=100;
11
12
13
14 - %Vettore
15 - vec=rand(n,n);
16
17 - %Inizializzo indici
18 - indexmin=0;
19 - indexmax=0;
20 - nmax=0;
21 - nmin=1;
22
23
24 - for i=1:m % scansione delle righe
25 -     for j=1:n % scansione delle colonne
26 -         if vec(i,j)<nmin
27 -             nmin=vec(i,j);
28 -             indexmin(1)=i;
29 -             indexmin(2)=j;
30 -         end
31 -         if vec(i,j)>nmax
32 -             nmax=vec(i,j);
33 -             indexmax(1)=i;
```

The Command Window shows the execution of the script, displaying the results of the calculations:

```
>> 4 * 25 + 6*52 + 2 * 99
ans =
    610
>> gomme=4; taccuini=6; rotoliNastroAdesivo=2; cose=gomme+taccuini+rotoliNastroAdesivo;
>> costo=gomme*25+taccuini*52+rotoliNastroAdesivo*99; costoMedio=costo/cose
costoMedio =
    50.8333
```

The Workspace window shows the variables created during the execution:

Name	Value
ans	610
cose	12
costo	610
costoMedio	50.8333
gomme	4
rotoliNastroAd...	2
taccuini	6



# IMPIEGO DI MATLAB PER QUALCHE CONTICINO:

Command Window

New to MATLAB? See resources for [Getting Started](#).

```
>> % Esempi di utilizzo di MATLAB per qualche semplice calcolo:
```

```
>> 2+2
```

```
ans =
```

```
4
```

```
>> atan(1)
```

```
ans =
```

```
0.7854
```

```
>> rand(10,10)
```

```
ans =
```

0.9223	0.6237	0.1932	0.5102	0.4857	0.1248	0.5493	0.8378	0.5906	0.1378
0.7710	0.2364	0.8959	0.9064	0.8944	0.7306	0.3304	0.7391	0.6604	0.8367
0.0427	0.1771	0.0991	0.6289	0.1375	0.6465	0.6195	0.9542	0.0476	0.1386
0.3782	0.8296	0.0442	0.1015	0.3900	0.8332	0.3606	0.0319	0.3488	0.5882
0.7043	0.7669	0.5573	0.3909	0.9274	0.3983	0.7565	0.3569	0.4513	0.3662
0.7295	0.9345	0.7725	0.0546	0.9175	0.7498	0.4139	0.6627	0.2409	0.8068
0.2243	0.1079	0.3119	0.5013	0.7136	0.8352	0.4923	0.2815	0.7150	0.5038
0.2691	0.1822	0.1790	0.4317	0.6183	0.3225	0.6947	0.2304	0.8562	0.4896
0.6730	0.0991	0.3390	0.9976	0.3433	0.5523	0.9727	0.7111	0.2815	0.8770
0.4775	0.4898	0.2101	0.8116	0.9360	0.9791	0.3278	0.6246	0.7311	0.3531

- 1 Ricalcolando `rand(10,10)`, riottenremo la stessa matrice ?

- 2 Che significa 'arcotangente' di  $x$  ?

- ③ Calcolare  $\pi$  (pi greco) in tre maniere:
- tramite un'espressione trigonometrica;
  - tramite l'espressione  $355/113$
  - tramite una 'variabile' speciale di **MATLAB**, della quale sta a voi indovinare il nome

- 3 Calcolare  $\pi$  (pi greco) in tre maniere:
- tramite un'espressione trigonometrica;
  - tramite l'espressione `355/113`
  - tramite una 'variabile' speciale di **MATLAB**, della quale sta a voi indovinare il nome

Utilizzando il `==` di **MATLAB**, confrontare poi i tre valori di  $\pi$  trovati e se non sono uguali, spiegare perché; servendosi ancora di **MATLAB**, mostrare piú cifre di  $\pi$ .

- ④ Generare in modo casuale numeri interi compresi fra 1 e 30 (estremi inclusi).

## QUALCHE ESERCIZIO SU QUANTO PRECEDE:

- 1 Ricalcolando `rand(10,10)`, riottenremo la stessa matrice ?
- 2 Che significa 'arcotangente' di  $x$  ?
- 3 Calcolare  $\pi$  (pi greco) in tre maniere:
  - tramite un'espressione trigonometrica;
  - tramite l'espressione `355/113`
  - tramite una 'variabile' speciale di **MATLAB**, della quale sta a voi indovinare il nome

Utilizzando il `==` di **MATLAB**, confrontare poi i tre valori di  $\pi$  trovati e se non sono uguali, spiegare perché; servendosi ancora di **MATLAB**, mostrare più cifre di  $\pi$ .

- 4 Generare in modo casuale numeri interi compresi fra 1 e 30 (estremi inclusi).
- 5 Calcolare l'unità immaginaria  $\sqrt{-1}$ .

- 1 No ( altrimenti il risultato non sarebbe 'random', cioè casuale )

- ② L'arco  $\alpha$  il cui rapporto  $\sin(\alpha)/\cos(\alpha)$  vale  $x$

- ⑧ A differenza di  $\pi$  —calcolabile sia come `4 * atan(1)` che come `pi`—il valore espresso da `355/113` è razionale. . . Questo può spiegare perché il confronto `355/113 == pi` dà risultato logico `0` da intendersi come `'falso'`

- 8 A differenza di  $\pi$  —calcolabile sia come `4 * atan(1)` che come `pi`—il valore espresso da `355/113` è razionale. . . Questo può spiegare perché il confronto `355/113 == pi` dà risultato logico `0` da intendersi come 'falso' . . . Per evidenziare la differenza, si impartisca il comando `format long`<sup>1</sup> e poi si ricalcolino `355/113`, `4 * atan(1)` e `pi`.

---

<sup>1</sup>Farsi illustrare altre possibilità tramite il comando `help format`

- Si può scrivere `floor(1 + 30 * rand(10, 10))`

- 1 No ( altrimenti il risultato non sarebbe 'random', cioè casuale )
- 2 L'arco  $\alpha$  il cui rapporto  $\sin(\alpha)/\cos(\alpha)$  vale  $x$
- 3 A differenza di  $\pi$  —calcolabile sia come  $4 * \text{atan}(1)$  che come  $\text{pi}$ —il valore espresso da  $355/113$  è razionale. . . Questo può spiegare perché il confronto  $355/113 == \text{pi}$  dà risultato logico  $0$  da intendersi come 'falso' . . . Per evidenziare la differenza, si impartisca il comando `format long`<sup>1</sup> e poi si ricalcolino  $355/113$ ,  $4 * \text{atan}(1)$  e  $\text{pi}$ .
- 4 Si può scrivere `floor(1 + 30 * rand(10, 10))`
- 5 Si può scrivere in vari modi: `sqrt(-1)`, `i`, `j`, `1i`

---

<sup>1</sup>Farsi illustrare altre possibilità tramite il comando `help format`

La finestra dei comandi, oltre a comandi interattivi, quali

- `pwd`     % Print current Working Directory
- `format long`
- `help echo`
- `close all` % Chiude tutte le Figure
- `clear`    % Rimuove tutte le variabili dallo spazio di lavoro
- `clc`      % Ripulisce la finestra dei comandi

ospita anche istruzioni di un linguaggio di programmazione Turing-completo. (☞ adeguato a specificare qualsiasi algoritmo)

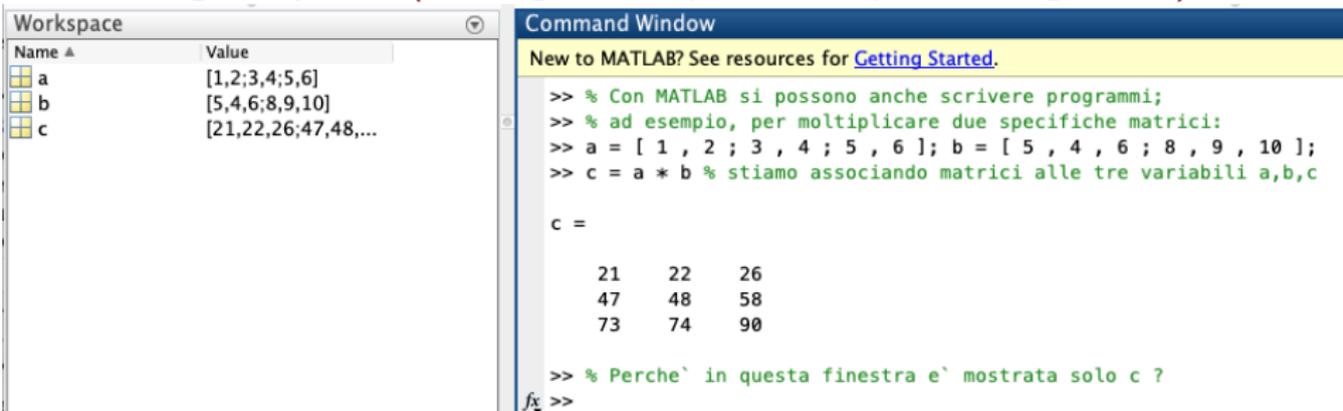
# FARE USO DI VARIABILI, IN AGGIUNTA ALLA *ans*

La finestra dei comandi, oltre a comandi interattivi, quali

- `pwd`     % Print current Working Directory
- `format long`
- `help echo`
- `close all` % Chiude tutte le Figure
- `clear`    % Rimuove tutte le variabili dallo spazio di lavoro
- `clc`      % Ripulisce la finestra dei comandi

ospita anche istruzioni di un linguaggio di programmazione

Turing-completo. (☞ adeguato a specificare qualsiasi algoritmo)



The screenshot shows the MATLAB interface. On the left is the 'Workspace' window, and on the right is the 'Command Window'.

**Workspace:**

Name	Value
a	[1,2;3,4;5,6]
b	[5,4,6;8,9,10]
c	[21,22,26;47,48,...

**Command Window:**

```
New to MATLAB? See resources for Getting Started.  
  
>> % Con MATLAB si possono anche scrivere programmi;  
>> % ad esempio, per moltiplicare due specifiche matrici:  
>> a = [ 1 , 2 ; 3 , 4 ; 5 , 6 ]; b = [ 5 , 4 , 6 ; 8 , 9 , 10 ];  
>> c = a * b % stiamo associando matrici alle tre variabili a,b,c  
  
c =  
  
    21    22    26  
    47    48    58  
    73    74    90  
  
>> % Perché in questa finestra è mostrata solo c ?  
fx >>
```

... bastano pochissime cose:

- istruzione di assegnamento `variabile=espressione`
- istruzione `if condizione ... elseif ... else ... end`
- istruzione `while condizione istruzioni end`

Ma già nella dotazione iniziale di **MATLAB** disporremo di ben altro:

- altre istruzioni (ad es la `for`);
- un potente meccanismo di ricorsione;
- funzioni *built-in*;
- *array*, stringhe di caratteri e altre strutture;
- ecc.

# script PER NON PROGRAMMARE A 'PALLA DI FANGO'

```
Editor - /Users/eugenioomodeo/Documents/MATLAB/esempio0.m
+5  esempio0.m  estrazione_lotto.m  matrix.m  Eserc
1   % script M-file esempio0.m
2
3   % echo on
4 -  gomme = 4; % numero di oggetti di ciascun tipo
5 -  taccuini = 6;
6 -  nastri = 2;
7 -  quanteCose = gomme + taccuini + nastri
8 -  spesa = gomme * 25 + taccuini * 52 + nastri * 99
9 -  costo_medio = spesa / quanteCose
10
```

Command Window

New to MATLAB? See resources for [Getting Started](#).

```
>> esempio0

quanteCose =

    12

spesa =

    610

costo_medio =

    50.8333

fx >> |
```

# echo PER IL *debugging*

```
Editor - /Users/eugenioodeo/Documents/MATLAB/esempio0.m
+5 esempio0.m x estrazione_lotto.m x matrix.m x Eserc
1 % script M-file esempio0.m
2
3 - echo on
4 - gomme = 4; % numero di oggetti di ciascun tipo
5 - taccuini = 6;
6 - nastri = 2;
7 - quanteCose = gomme + taccuini + nastri
8 - spesa = gomme * 25 + taccuini * 52 + nastri * 99
9 - costo_medio = spesa / quanteCose
10
11 % Come si puo` notare in questo caso,
12 % il comando 'echo' rende il risultato
13 % molto piu` difficile da leggere.
14 % Pero`, d'altra parte, il comando 'echo'
15 % puo` risultare molto utile quando si
16 % collaudano e correggono degli script
17 % piu` complicati di questo.
18
19 - echo off
20
```

## Command Window

New to MATLAB? See resources for [Getting Started](#).

```
>> % Qui sotto richiamo uno script
>> esempio0
gomme = 4; % numero di oggetti di ciascun tipo
taccuini = 6;
nastri = 2;
quanteCose = gomme + taccuini + nastri

quanteCose =
```

12

# echo PER IL *debugging*

```
Editor - /Users/eugenioomodeo/Documents/MATLAB/esempio0.m
+5 esempio0.m x estrazione_lotto.m x matrix.m x Eserc
1 % script M-file esempio0.m
2
3 - echo on
4 - gomme = 4; % numero di oggetti di ciascun tipo
5 - taccuini = 6;
6 - nastri = 2;
7 - quanteCose = gomme + taccuini + nastri
8 - spesa = gomme * 25 + taccuini * 52 + nastri * 99
9 - costo_medio = spesa / quanteCose
10
11 % Come si puo` notare in questo caso,
12 % il comando 'echo' rende il risultato
13 % molto piu` difficile da leggere.
14 % Pero`, d'altra parte, il comando 'echo'
15 % puo` risultare molto utile quando si
16 % collaudano e correggono degli script
17 % piu` complicati di questo.
18
19 - echo off
20
```

## Command Window

New to MATLAB? See resources for [Getting Started](#).

```
>> % Qui sotto richiamo uno script
>> esempio0
gomme = 4; % numero di oggetti di ciascun tipo
taccuini = 6;
nastri = 2;
quanteCose = gomme + taccuini + nastri

quanteCose =

    12
```

## Command Window

New to MATLAB? See resources for [Getting Started](#).

```
>> % Qui sotto richiamo uno script
>> esempio0
gomme = 4; % numero di oggetti di ciascun tipo
taccuini = 6;
nastri = 2;
quanteCose = gomme + taccuini + nastri

quanteCose =

    12

spesa = gomme * 25 + taccuini * 52 + nastri * 99

spesa =

    610

costo_medio = spesa / quanteCose

costo_medio =

    50.8333

% Come si puo` notare in questo caso,
% il comando 'echo' rende il risultato
% molto piu` difficile da leggere.
% Pero`, d'altra parte, il comando 'echo'
% puo` risultare molto utile quando si
% collaudano e correggono degli script
% piu` complicati di questo.

echo off
```

# C'È INOLTRE LA POSSIBILITÀ DI USARE DEI *live script*!

The screenshot displays the MATLAB environment. At the top, the Command Window shows the execution of a script with the following commands and output:

```
>> clear, format long, 818/16  
  
ans =  
  
51.125000000000000
```

Below the Command Window is the Live Editor window titled "Live Editor - esempio0.mlx \*". It contains a script file named "esempio0.m" with the following code:

```
1 echo on  
2 gomme = 4; % numero di oggetti di ciascun tipo  
3 taccuini = 10;  
4 nastri = 2;  
5 quanteCose = gomme + taccuini + nastri  
6 spesa = gomme * 25 + taccuini * 52 + nastri * 99  
7 costo_medio = spesa / quanteCose  
8  
9 % Come si puo` notare in questo caso,  
10 % il comando 'echo' rende il risultato  
11 % molto piu` difficile da leggere.  
12 % Pero`, d'altra parte, il comando 'echo'  
13 % puo` risultare molto utile quando si  
14 % collaudano e correggono degli script  
15 % piu` complicati di questo.  
16  
17 echo off
```

On the right side of the Live Editor, the output of the script is displayed in a table:

quanteCose =
16
spesa =
818
costo_medio = 51.1250

# INCAPSULAMENTO ( $\approx$ *information hiding* )

Quando viene lanciato uno *script*, può accadere che lo spazio di lavoro ( *workspace* ) si popoli di nuove variabili o che venga modificato il valore di variabili che già c'erano.

# INCAPSULAMENTO ( $\approx$ *information hiding* )

Quando viene lanciato uno *script*, può accadere che lo spazio di lavoro ( *workspace* ) si popoli di nuove variabili o che venga modificato il valore di variabili che già c'erano.

L'impiego di *M-funzioni* riduce la confusione, assegnando a uno *script* un suo privato spazio di lavoro;

L'impiego di *M-funzioni* riduce la confusione, assegnando a uno *script* un suo privato spazio di lavoro;

lo scambio di informazioni fra

- finestra dei comandi e un' *M-funzione*,
- due *M-funzioni* (una 'invocante', l'altra 'invocata'),
- due incarnazioni(??) della stessa *M-funzione*

avviene tramite un meccanismo chiamato *passaggio di parametri*.

Da [Heslop, pag.33]:

$$g = \sin(2 * \pi * f + \theta)$$

We can use the `inline` function to define this expression and its input, so that it can be called repeatedly. Note that during the definition of the inline function the expression and input variable names are defined as character strings.

```
>> g = inline('sin(2*pi*f + theta)', 'f', 'theta')
```

```
g =
```

```
Inline function:
```

```
g(f,theta) = sin(2*pi*f + theta)
```

```
>> g(0.5,pi./4)
```

```
ans =
```

```
-0.707106781186547
```

Inline functions are useful for simple expressions, but in more complex situations it is necessary to create an M-function.

# ALTRA IMPORTANTE RISORSA: LA GRAFICA

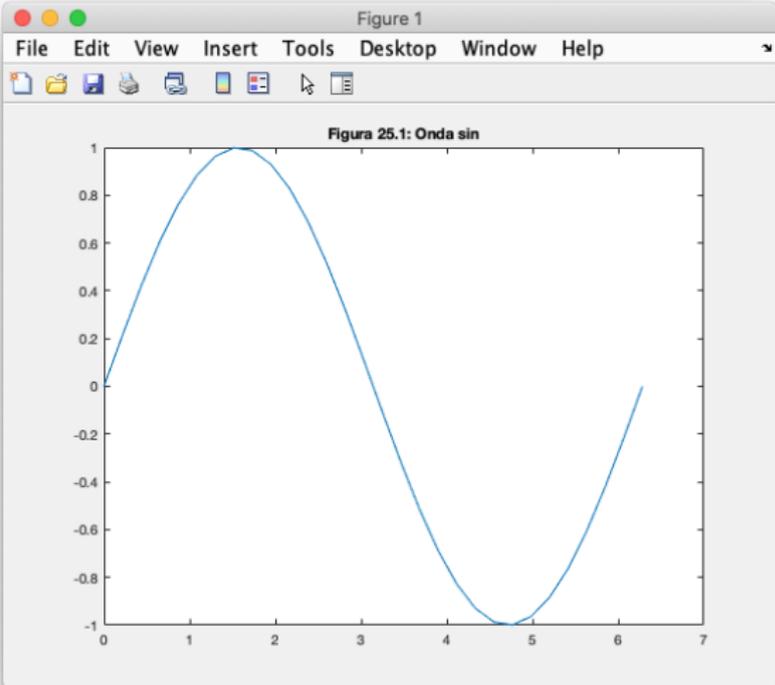
Workspace

Name	Value
x	1x30 double
y	1x30 double

Command Window

New to MATLAB? See resources for [Getting Started](#).

```
>> x = linspace(0,2*pi,30); y = sin(x); plot(x,y); title('Figura 25.1: Onda sin');  
fx >>
```



The figure window displays a plot of a sine wave. The x-axis ranges from 0 to 7, and the y-axis ranges from -1 to 1. The plot shows a single cycle of a sine wave starting at (0,0), peaking at approximately (1.57, 1), crossing the x-axis at approximately (3.14, 0), reaching a trough at approximately (4.71, -1), and ending at approximately (6.28, 0). The window title is 'Figure 1' and the plot title is 'Figura 25.1: Onda sin'.

# ALTRA IMPORTANTE RISORSA: LA GRAFICA

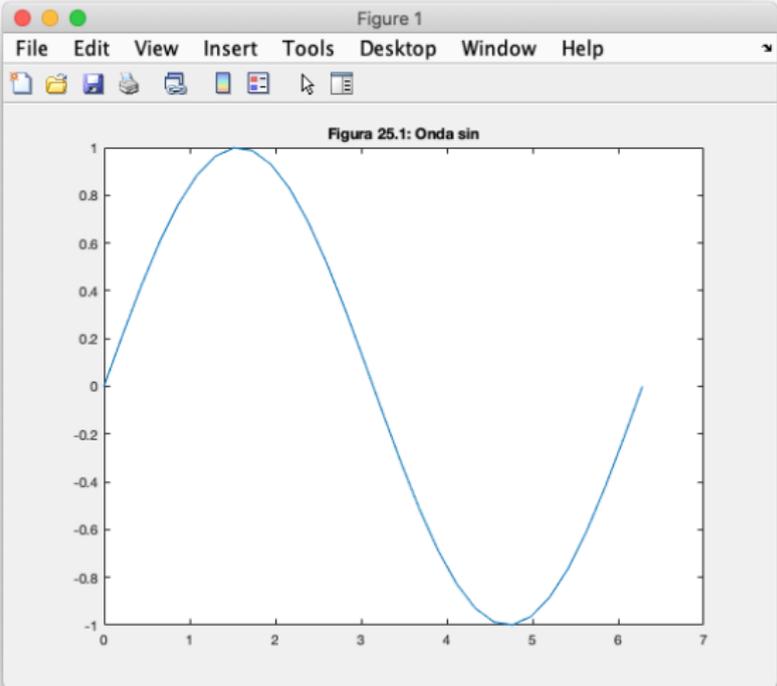
Workspace

Name	Value
x	1x30 double
y	1x30 double

Command Window

New to MATLAB? See resources for [Getting Started](#).

```
>> x = linspace(0,2*pi,30); y = sin(x); plot(x,y); title('Figura 25.1: Onda sin');  
fx >>
```



The figure window displays a plot of a sine wave. The x-axis ranges from 0 to 7, and the y-axis ranges from -1 to 1. The plot shows a smooth curve starting at (0,0), reaching a peak of 1 at approximately x=1.57, crossing the x-axis at approximately x=3.14, reaching a trough of -1 at approximately x=4.71, and ending at approximately x=6.28. The plot is titled 'Figura 25.1: Onda sin'.

Che succede se suddivido per 300 invece che per 30 ?

# ANCHE ARRAY *multi*-DIMENSIONALI ( CON *multi*>2 )

http://matlab/math/multidimensional-arrays.html

Search

Documentation Examples Functions Videos Answers

row

column

page

(1,1,1)	(1,2,1)	(1,3,1)	(1,4,1)
(2,1,1)	(2,2,1)	(2,3,1)	(2,4,1)
(3,1,1)	(3,2,1)	(3,3,1)	(3,4,1)
(4,1,1)	(4,2,1)	(4,3,1)	(4,4,1)

(1,1,2)	(1,2,2)	(1,3,2)	(1,4,2)
(2,1,2)	(2,2,2)	(2,3,2)	(2,4,2)
(3,1,2)	(3,2,2)	(3,3,2)	(3,4,2)
(4,1,2)	(4,2,2)	(4,3,2)	(4,4,2)

(1,1,3)	(1,2,3)	(1,3,3)	(1,4,3)
(2,1,3)	(2,2,3)	(2,3,3)	(2,4,3)
(3,1,3)	(3,2,3)	(3,3,3)	(3,4,3)
(4,1,3)	(4,2,3)	(4,3,3)	(4,4,3)

**Creating Multidimensional Arrays**

You can create a multidimensional array by creating a 2-D matrix first, and then extending it.

# ANCHE ARRAY *multi*-DIMENSIONALI ( CON *multi*>2 )

```
>> A=zeros(10,5); A(:,2)=floor(rand(10,5)*90+1); A(:,3)=ones(10,5);  
>> A % per visualizzare l'array tridimensionale costruito qui sopra
```

A(:,1) =

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

A(:,2) =

3	81	74	61	13
76	11	10	43	20
51	89	17	83	81
77	49	33	10	7
32	64	6	68	22
41	90	47	67	5
5	26	31	51	40
16	38	16	17	2
60	42	19	54	81
30	69	82	27	18

A(:,3) =

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

# ANCHE ARRAY *multi*-DIMENSIONALI ( CON *multi*>2 )

## ...e oltre la terza dimensione ...

```
>> A=zeros(10,5); A(:,:,2)=floor(rand(10,5)*90+1); A(:,:,3)=ones(10,5);  
>>  
>> A(:,:,,2) = rand(10,5,3)
```

A(:,:,1,1) =

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

A(:,:,2,1) =

74	15	60	64	40
82	88	4	3	35
12	87	77	25	69
83	44	85	5	72
57	73	62	9	17
9	13	69	75	45
26	38	67	63	41
50	83	36	29	59
87	72	59	86	64
87	87	16	4	68

A(:,:,1,2) =

0.2760	0.7513	0.8407	0.3517	0.0759
0.6797	0.2551	0.2543	0.8308	0.0540
0.6551	0.5060	0.8143	0.5853	0.5308
0.1626	0.6991	0.2435	0.5497	0.7792
0.1190	0.8909	0.9293	0.9172	0.9340
0.4984	0.9593	0.3500	0.2858	0.1299
0.9597	0.5472	0.1966	0.7572	0.5688
0.3404	0.1386	0.2511	0.7537	0.4694
0.5853	0.1493	0.6160	0.3804	0.0119
0.2238	0.2575	0.4733	0.5678	0.3371

A(:,:,2,2) =

0.1622	0.4505	0.1067	0.4314	0.8530
0.7943	0.0838	0.9619	0.9106	0.6221
0.3112	0.2290	0.0046	0.1818	0.3510
0.5285	0.9133	0.7749	0.2638	0.5132
0.1656	0.1524	0.8173	0.1455	0.4018
0.6020	0.8258	0.8687	0.1361	0.0760
0.2630	0.5383	0.0844	0.8693	0.2399
0.6541	0.9961	0.3998	0.5797	0.1233
0.6892	0.0782	0.2599	0.5499	0.1839
0.7482	0.4427	0.8001	0.1450	0.2400

## Cell Arrays

Arrays that can contain data of varying types and sizes

A cell array is a data type with indexed data containers called *cells*, where each cell can contain any type of data. Cell arrays commonly contain either lists of character vectors of different lengths, or mixes of strings and numbers, or numeric arrays of different sizes. Refer to sets of cells by enclosing indices in smooth parentheses, (). Access the contents of cells by indexing with curly braces, {}. For more information, see [Access Data in Cell Array](#) or watch [Introducing Structures and Cell Arrays](#).

```
>> Nido=[{[1,2,3;7,6,8]},4;{[5,6]},7], Pagella=[{'Italiano'},7;{'Cinese'},6],
```

```
Nido =
```

```
2x2 cell array
```

```
{2x3 double}    {[4]}
{1x2 double}    {[7]}
```

```
Pagella =
```

```
2x2 cell array
```

```
{'Italiano'}    {[7]}
{'Cinese' }     {[6]}
```

```
>> Nido(1), Nido{1}, Nido{2,1}, Nido{2}, Pagella{1}
```

```
ans =
```

```
1x1 cell array
```

```
{2x3 double}
```

```
ans =
```

```
    1    2    3
    7    6    8
```

```
ans =
```

```
    5    6
```

```
ans =
```

```
    5    6
```

```
ans =
```

```
'Italiano'
```

Vedi anche: [strutture](#)

Di qui in avanti: Esercizi del giorno

## PROBABILITÀ DI AZZECCARE LA CINQUINA CHE SARÀ ESTRATTA

① Su una ruota:  $\frac{1}{\binom{90}{5}} = \frac{1}{43\,949\,268} \approx 0,00000228 \%$

② Su  $R$  ruote:  $1 - \left(1 - \frac{1}{43\,949\,268}\right)^R$

## PROBABILITÀ DI AZZECCARE LA CINQUINA CHE SARÀ ESTRATTA

① Su una ruota:  $\frac{1}{\binom{90}{5}} = \frac{1}{43\,949\,268} \approx 0,00000228 \%$

② Su  $R$  ruote:  $1 - \left(1 - \frac{1}{43\,949\,268}\right)^R$

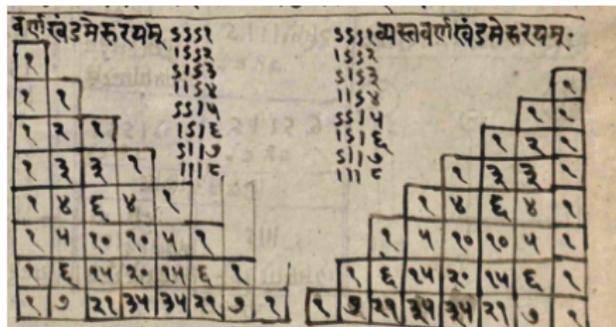
## COS'È E COME SI CALCOLA IL COEFFICIENTE BINOMIALE $\binom{N}{k}$ ?

$$\binom{N}{k} = \begin{cases} 0 & \text{se } N < k, \\ 1 & \text{se } k = 0, \\ \binom{N-1}{k-1} + \binom{N-1}{k} & \text{altrimenti.} \end{cases}$$

$N$	coefficiente binomiale $\binom{N}{k}$						
0	1	0	0	0	0	0	...
1	1	1	0	0	0	0	...
2	1	2	1	0	0	0	...
3	1	3	3	1	0	0	...
4	1	4	6	4	1	0	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

$k=0,1,2,3,4,5,\dots$

FIGURA: Il triangolo di ... Khayyám, Tartaglia, Pascal



Over time people have used a number of different approximations for  $\pi$ , a popular one is  $\frac{22}{7}$ . Probably the strangest is a legal bill which was proposed in 1897 that would set  $\pi = 3.2$  in order that people could remember it more easily. There is a long history of methods with which to approximate  $\pi$ , one of the most elegant of which was given in the 14<sup>th</sup> century by the Indian mathematician Madhava of Sangamagrama (interestingly the technique was “rediscovered” in Europe over 300 years later). Madhava of Sangamagrama showed that  $\frac{\pi}{4}$  could be obtained as the sum of an infinite series:

$$\sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} = \frac{\pi}{4}$$

Therefore if we work with the first 4 terms, for  $n=0$  to  $n=3$  and ultimately working to infinity we would get:

31

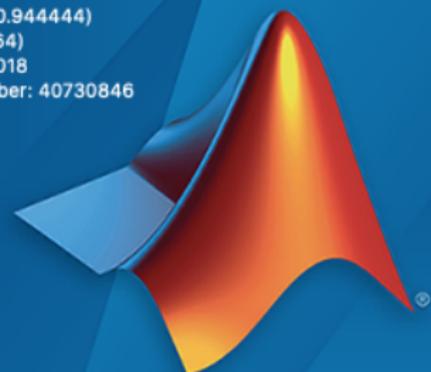
$$\frac{(-1)^0}{2*0+1} + \frac{(-1)^1}{2*1+1} + \frac{(-1)^2}{2*2+1} + \frac{(-1)^3}{2*3+1} + \dots + \frac{(-1)^\infty}{2*\infty+1} = \frac{\pi}{4}$$

the beginning of which simplifies to:

$$1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots = \frac{\pi}{4}$$

We’re going to use this approach to estimate  $\pi$  in MATLAB. Obviously we can’t calculate an infinite number of terms so we’ll only work to a maximum of  $n = 10000$ . One function you might find useful is `sum`, look at its help entry to get more information about its use. For those of you familiar with MATLAB this exercise won’t be too much of a challenge so you should try to perform same task whilst playing MATLAB “golf”. That is, what is the smallest number of characters you can use in your code to make the calculation (mine was 30 characters). I would

R2018b (9.5.0.944444)  
64-bit (mac64)  
August 28, 2018  
License Number: 40730846



# MATLAB®

Academic License - for use in teaching, academic research, and meeting course requirements at degree granting institutions only. Not for government, commercial, or other organizational use.

© 1984-2018 The MathWorks, Inc. Protected by U.S and international patents. See [mathworks.com/patents](http://mathworks.com/patents). MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [mathworks.com/trademarks](http://mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.



R2018b

# ECCO ORA UNO *script* PER LA CONVERS. DI BASE...

```
Editor - /Users/eugenioomodeo/Documents/MATLAB/conversBase.m
+6  Esercizi.mlx  x  estrazRadiceQuadrata.m  x  intro_if.m  x  pari_dispari.m  x  vettore.m
1   % conversione di un numero naturale nelle basi 2, 3, 8, 16
2
3   cifre = ['0','1','2','3','4','5','6','7','8','9','a','b','c','d','e','f'];
4
5   N=input('Dammi un valore intero: ');
6
7   if N < 0
8
9       fprintf('%1.0f e` un numero negativo', N);
10  else
11
12      for base = [ 2 , 3 , 8 , 16 ]
13
14          Ncopia = N ; % Per non rovinare l'originale
15
16          NinBase = '' ; % Diventera` la rappresentazione di N in altra base
17
18          while Ncopia ~= 0 % qui inizia la conversione in base 'base'
19
20              NinBase = strcat( cifre(rem( Ncopia, base ) + 1) , NinBase );
21
22              Ncopia = floor( Ncopia / base );
23
24          end
25
26          if size( NinBase ) == 0
27
28              NinBase = '0';
29          end
30
31          NinBase
32
33      end
34
35  end
```

# ECCO ORA UNO *script* PER LA CONVERS. DI BASE...

```
Editor - /Users/eugenioomodeo/Documents/MATLAB/conversBase.m
+6  Esercizi.mlx  x  estrazRadiceQuadrata.m  x  intro_if.m  x  pari_dispari.m  x  vettore.m
1   % conversione di un numero naturale nelle basi 2, 3, 8, 16
2
3   cifre = ['0','1','2','3','4','5','6','7','8','9','a','b','c','d','e','f'];
4
5   N=input('Dammi un valore intero: ');
6
7   if N < 0
8
9       fprintf('%1.0f e` un numero negativo', N);
10  else
11
12      for base = [ 2 , 3 , 8 , 16 ]
13
14          Ncopia = N ; % Per non rovinare l'originale
15
16          NinBase = '' ; % Diventera` la rappresentazione di N in altra base
17
18          while Ncopia ~= 0 % qui inizia la conversione in base 'base'
19
20              NinBase = strcat( cifre(rem( Ncopia, base ) + 1) , NinBase );
21
22              Ncopia = floor( Ncopia / base );
23
24          end
25
26          if size( NinBase ) == 0
27
28              NinBase = '0';
29          end
30
31          NinBase
32
33      end
34
35  end
```

Command Window

New to MATLAB? See resources for [Getting](#)

Dammi un valore intero: 367

NinBase =

'101101111'

NinBase =

'111121'

NinBase =

'557'

NinBase =

'16f'

f<sub>x</sub> >>

Scrivere uno *script* che consenta di

*Trovare il massimo numero naturale non eccedente la radice quadrata di un dato numero  $N$*

- 1 Utilizzando la funzione predefinita `sqrt`
- 2 Effettuando il calcolo di quadrati crescenti consecutivi, fino a superamento della meta
- 3 Come sopra, ma senza fare moltiplicazioni ( solo somme )