

# Matematica Applicata

## - MATLAB- Funzioni Utili-

Laboratorio didattico

A.A 2011/2012

Stefano Vena

# Materiale

- Matlab :  
<http://www.mathworks.com/products/matlab>
- Octave :  
<http://octave.sourceforge.net/>
- Materiale del Corso :  
<https://github.com/wetfire2k/Matematica-Applicata>
- Libro:  
Calcolo Scientifico, Esercizi e problemi risolti con  
MATLAB e Octave, Alfio Quarteroni, Fausto Saleri.

# Funzioni matematiche comuni

Funzione	comando
Valore Intero vicino ad <b><i>a</i></b> più grande	ceil(a)
Valore Intero vicino ad <b><i>a</i></b> più piccolo	floor(a)
Arrotondamento di <b><i>a</i></b>	round(a)
Esponenziale di <b><i>a</i></b>	exp(a)
Segno di <b><i>a</i></b>	sign(a)
Logaritmo di <b><i>a</i></b>	log(a),log2(a) log10(a)
Radice quadrata di <b><i>a</i></b>	sqrt(a)

# Funzioni matematiche comuni

Funzione	comando
Resto divisione intera $a/b$	<code>mod(a,b)</code>
Max. elem. di ogni colonna di $a$	<code>max(a)</code>
Min. elemento di un vettore $b$	<code>min(b)</code>
Min. elem. di ogni colonna di $a$	<code>min(a)</code>
Max. elemento di un vettore $b$	<code>max(b)</code>
Somma elementi del vettore $b$	<code>sum(b)</code>
Prodotto elementi del vettore $b$	<code>prod(b)</code>
Prodotto elementi $k$ -ma col. di $a$	<code>prod(a,k)</code>
Somma elementi $k$ -ma colonna di $a$	<code>sum(a,k)</code>

# Input - Elementare

- Quando si richiede un dato da tastiera è bene aggiungere un messaggio che specifica cosa il programma “si aspetta”

```
a = input('Inserisci un numero ')
```

- Viene assegnata alla variabile `a` il valore che viene inserito da tastiera

```
Nome = input('come ti chiami? >', 's');
```

- Il parametro `'s'` serve a trattare l'input come stringa.
- E' buona norma terminare la stringa della richiesta con uno spazio o con un altro separatore poiché il dato inserito da tastiera viene visualizzato immediatamente dopo di essa.

# Output Elementare

- Anche quando si effettua un'operazione di output può essere necessario aggiungere un messaggio che specifichi il significato del dato da stampare
  - `disp('Il numero inserito è: '),`  
`disp(a)`
- mostra in una riga il messaggio e nella riga sottostante il valore della variabile `a` poiché alla fine di ogni comando `disp(x)` viene aggiunta una «andata a capo»
- In generale il comando `disp(x)` stampa a video il contenuto di una matrice.

# Input/Output - Esempio

```
>> a=input('Inserisci i valori della matrice :');  
Inserisci i valori della matrice : [11, 12, 13;  
21,22,23; 31,32,33]  
>> disp( a(1,3) )  
    13  
  
>> disp( a(3,1) )  
    31
```

# Esercizi

- Scrivere un programma che chiede l'inserimento di due numeri da tastiera e stampa il più grande fra i due
- Scrivere un programma che chiede l'inserimento di un numero da tastiera e stampa se è pari o dispari
- Scrivere un programma che prenda in ingresso un intero N e calcoli il doppio della somma dei primi N numeri



# Funzione rand

```
r = rand()
```

restituisce un numero casuale nell'intervallo aperto (0,1)

```
r = rand (n)
```

restituisce una matrice quadrata di dimensione nxn i cui valori sono numeri casuali appartenenti all'intervallo (0,1)

```
r = rand (m,n)
```

Restituisce una matrice di dimensione mxn i cui valori sono numeri casuali appartenenti all'intervallo (0,1)

## **Esempio:**

```
>> rand(2)
```

```
ans =
```

```
    0.8147    0.1270
```

```
    0.9058    0.9134
```

# Funzione rand

- $r = b \cdot \text{rand}(n)$

restituisce una matrice quadrata di dimensione  $n \times n$  i cui valori sono numeri casuali appartenenti all'intervallo  $(0,b)$

- $r = a + (b-a) \cdot \text{rand}(n)$

Restituisce una matrice di dimensione  $n \times n$  i cui valori sono numeri casuali appartenenti all'intervallo  $(a,b)$

## Esempio

```
>> 2 + (20-2)*rand(1)
ans =
    7.2345
```

# Funzione rand: Esercizio

- Indovinare un numero intero fra 1 e 100

```
indovinato=0;
numero = ceil(1 + 99*rand(1));
while (indovinato==0)
    tentativo = input('Inserisci un numero fra 1 e 100');
    if (tentativo==numero)
        disp('Complimenti hai indovinato!');
        indovinato=1;
    end
end
```

# Output avanzato - Files

- Le variabili possono essere salvate in files binari o di testo
- Salvataggio a Caricamento dell'intero workspace.  
`save filename, load filename`
- Salvataggio a Caricamento di variabili in/da 'test.mat'  
`save 'test' a b c; load 'test'`
- Lettura di un file di testo formattato  
`[A,B,C,...] = textread('filename','format')`

```
esempio: [A,B,C]=textread('test.dat', '%f %f %f')
          A=
          1.2000
          5.5000
          1.0000
```

- **Data input/output generico**  
f=fopen('filename','w')                      apri file  
fprintf(f,format,v1,...)                      scrittura output formattato  
fscanf(f,format,v1,...)                      lettura dati formatattati da file  
fclose(f)

# Funzione fprintf

- `fprintf(fileID, format, A, ...)`
- `fprintf(format, A, ...)`
- `count = fprintf(...)`
- `fprintf(fileID, format, A, ...)`
  - stampa il contenuto del vettore **A** per colonne secondo il formato **format** sul file specificato da **fileID**
- `fprintf(format, A, ...)`
  - come sopra, ma visualizza il risultato sullo schermo
- `count = fprintf(...)`
  - assegna alla variabile `count` il numero di byte(caratteri) che sono stati stampati

# Formati per fprintf

- Il formato `format` è una stringa racchiusa fra apici che descrive come deve apparire il testo da stampare.
- Le parti costanti del testo devono essere riportate così come devono apparire.
- Se si deve stampare il contenuto di una variabile, si deve inserire una lettera corrispondente al tipo di dato preceduta da %.
  - %d = intero,
  - %f = reale,
  - %e = reale in formato esponenziale
  - %s = stringa

# Formati per fprintf

- Possono essere presenti anche caratteri speciali (a capo, tabulazione, apici, ecc.)
  - ' ' apice singolo
  - %% simbolo di percentuale
  - \\ backslash (barra inversa singola)
  - \b backspace (cancella il carattere sulla sinistra)
  - \n a capo
  - \t tabulazione orizzontale