

Introduzione al linguaggio Matlab - 1

L'interfaccia Matlab é la seguente:

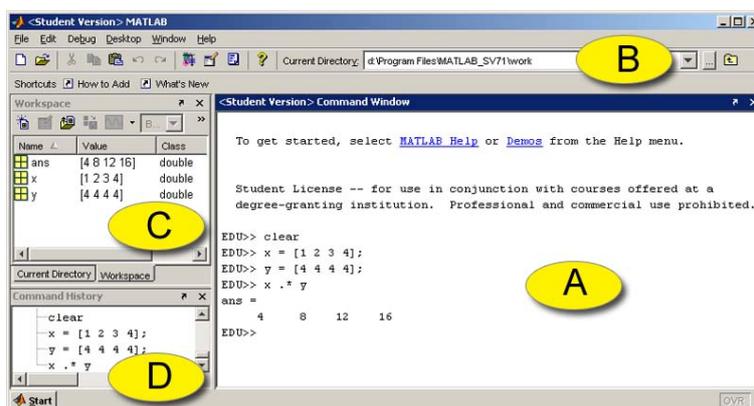


Figure 1: Interfaccia utente Matlab

- A) **Command Window**: é la finestra in cui si eseguono in comandi.
- B) **Current Directory**: indica l'indirizzo di memoria del sistema sul quale stiamo lavorando.
- C) **Workspace**: sono visualizzati i nomi dei variabili, i loro tipi e i loro valori.
- D) **Command History**: le istruzioni eseguite in precedenza ed eventualmente richiamabili.

Per poter eseguire un comando, esso va digitato nella *Command Window*, dopo il simbolo ">>", se il doppio maggiore é visualizzato significa che Matlab é pronto ad eseguire l'operazione.

Andiamo ora a vedere le prime semplici operazioni, se vogliamo eseguire delle operazioni algebriche é sufficiente digitarle con la classica sintassi aritmetica e premere *Invio*:

```
>> 7*3.5-2.1
ans =
    224
```

se l'operazione é stata eseguita correttamente, verrà visualizzato "ans =" seguito dal risultato dell'operazione. Se al comando viene preceduto un nome, quello sarà assegnato alla variabile:

```
>> b=7*3.5-2.1
b =
    224
```

Osservazione: Se si vuole sopprimere l'output a schermo bisogna mettere il ";" alla fine dell'espressione, in questa maniera Matlab salverá il risultato dell'operazione nel *Workspace* senza mostrarlo sulla *Command Window*.

Osservazione: Matlab é *case sensitive*, fa quindi differenza tra le maiuscole e le minuscole.

Nomi riservati e formati: in Matlab alcuni nomi sono riservati, ovvero associati a valori o operazioni predefinite, il che significa che non possono essere utilizzati nelle operazioni a piacimento dell'utente, alcuni esempi

```
>> pi
ans =
    3.1415
```

```
>> eps
ans =
    2.22..... (precisione di macchina)
```

É possibile poi scegliere i formati numerici (precisione che in default é a 5 cifre decimali) con i quali effettuare i calcoli, per scegliere il formato desiderato basta scrivere il formato preceduto dalla parola riservata "*format*", alcuni esempi

```
>> format short
>> format long
```

In Matlab l'elevamento a potenza si effettua con il comando "cappuccio", ad esempio se si vuole calcolare 2^3 :

```
>> 2^3
ans =
      8    (elevamento a potenza)
```

Il comando *clear* elimina tutte le variabili dal *Workspace*, mentre é possibile eliminare solo alcune variabili facendo seguire *clear* dai nomi delle variabili stesse.

Calcoli con variabili: di seguito alcuni esempi di calcolo con variabili che utilizzano funzioni matematiche predefinite:

```
>> y=cos(pi/3);
>> z=sin(pi/3);
>> a=log(y);
>> b=log(z);
```

Osservazione: $\log = \ln$.

Di uso fondamentale in Matlab é il comando di HELP in linea, digitando *help* seguito da una parola riservata si avrá visualizzato a schermo il funzionamento di quel comando:

```
>> help log

LOG natural .....
.....
```

Matrici e vettori: Ricordiamo che é buona norma digitare prima di qualunque sessione di lavoro il comando *clear*, per "ripulire" il *Workspace* da tutte le variabili già in uso.

Vediamo come si definiscono e si stampano matrici:

```
>> A = [ 1 2 3 ; 2 3 4 ; 3 4 5];
```

con lo "spazio" si dividono le varie entrate di ogni riga della matrice, mentre con il ";" si dividono le righe stesse:

```
>> A = [ 1 2 3 ; 2 3 4 ; 3 4 5]
```

```
A =
     1 2 3
     2 3 4
     3 4 5
```

Il comando "' ' " crea la trasposta di una matrice: se

$$B = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 2 & 1 \end{pmatrix}$$

allora

```
>> D=B'
```

```
D =
     1 4
     2 2
     3 1
```

Osservazione: Se durante la sessione di lavoro viene utilizzato un nome già in uso per definire una nuova variabile, allora Matlab sovrascriverá il nuovo valore sulla vecchia variabile.

Calcoli con matrici e vettori: Somma e sottrazione tra matrici

Il comandi per l'algebra tra matrici sono:

```
>> C = A+B      (somma tra matrici)
>> D = 3*A      (prodotto scalare-matrice)
>> D = 3.12*A + 4.5*B (combinazione lineare tra matrici)
```

Osservazione: I calcoli devono essere CONSISTENTI con l'algebra lineare delle matrici.

```
>> A*B          (prodotto righe per colonne)
>> A*b          (prodotto matrice-vettore)
>> D = 3.12*A + 4.5*B (combinazione lineare tra matrici)
```

Andiamo a vedere come si calcola il prodotto scalare tra vettori u e v :

```
>> u=.... ;
    v=.....;

>> u*v'        (prodotto scalare)
```

Osservazione: $u' * v$ da come risultato una matrice.

Calcoliamo ora la *norma* di un vettore u

```
>> u*u';
>> sqrt(u*u');
ans =
    .....norma di u
```

Il comando *norm* calcola questo risultato in una sola operazione

```
>> norm(u) ;
>> sqrt(u*u');
ans =
    norma euclidea
```

In Matlab é possibile effettuare il cosiddetto *prodotto di Hadamard* che non é altro che un operatore puntuale elemento per elemento di una matrice, l'operatore é il punto ".".

```
>> C=A.*B ;
>> D=A./B;
>> E=A.^B; (elevamento a potenza)
```

Osservazione: $A \pm B$ non esiste.

Matrici particolari: In generale una matrice A quadrata di ordine n in Matlab é definita con il comando relativo al tipo di matrice seguito da (n) nel caso di una matrice $n \times m$ il comando é come il precedente ma seguito da (n, m)

```
>> A=eye(3)
A =
    1 0 0
    0 1 0
    0 0 1 (la matrice unitaria)
```

```
>> A=eye(3,5)
A =
    1 0 0 0 0
    0 1 0 0 0
    0 0 1 0 0 (la matrice unitaria completata)
```

Altri esempi di matrici:

```
>> A = ones(5) (matrice 5x5 formata da soli zeri)
>> A = zeros(5) (matrice 5x5 formata da soli uno)
```

Matrici rettangolari:

```
>> A = ones(3,4) (matrice 3x4 formata da soli zeri)
>> A = zeros(3,2) (matrice 3x2 formata da soli uno)
>> A = rand(3)
A =
    ..... (matrice 3x3 con valori nell'intervallo [0,1])
```

Esercizio: studiare il comando

```
>> A = magic(3)
```

Esercizio: costruire la matrice C di dimensione (9×9) così fatta:

$$C = \begin{pmatrix} 12 & 1 & \dots & 1 \\ 1 & 12 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 1 & \dots & 1 & 12 \end{pmatrix}$$

```
>> A = ones(9);
>> B = eye(8);
>> C = A+11*B
C =
.....
```

Se voglio estrarre da una matrice l'elemento di posto (i, j) basta utilizzare il comando $A(i,j)$

```
>> A = [ ..... ];
>> A(2,3)
A =
    elemento di posto (2,3)
```

Se voglio estrarre da una matrice una sua sottomatrice il comando è il seguente:

```
>> B = A( 2:3 , 1:3)   estrae da A la sottomatrice che ha come righe, le righe dalla 2 alla 3
                        di A e come colonne, le colonne da 1 a 3 di A
```

A questo punto possiamo introdurre il comando " $a : \Delta : b$ " che genera il vettore composto da tutti i valori tra a e b con passo Δ .

Osservazione se utilizziamo il comando " $a : b$ ", il passo è di default unitario.

```
>> a = 1:9      (genera il vettore a=[1,2,...,8,9])
a =
    1 2 3 4 5 6 7 8 9
>> a = 1:2:9    (da 1 a 9 con passo 2)
a =
    1 3 5 7 9
>> x = 1:0.2:2;
```

Esempi di grafica: andiamo ora a vedere come si genera il grafico di una funzione reale di una variabile. Il comando è $plot(x,y)$ dove x e y sono due vettori di stessa dimensione che rappresentano i valori della variabile indipendente con i rispettivi valori della variabile dipendente:

```
>> x = 0:0.01:2;
>> y = sin(3*pi*x);
>> plot(x,y)      (produce una finestra contenente il grafico della funzione
                  y=sin(3 pi x) nell'intervallo [0,2])
```

Esercizio: disegnare il grafico della funzione

$$y = \frac{4x^3 - 1}{1 + 2x^2}, \quad -4 \leq x \leq 4.$$

con passo a piacere.

```
>> x = -4:0.02:4;
>> y = (4*x.^3-1)./(1+2*x.^2);
>> plot(x,y)
```

Osservazione: Se dopo eseguito il comando $plot$ non si vedesse la finestra grafica, essa è spesso rimasta come icona, in quanto Matlab sovrascrive il grafico a quello vecchio senza chiuderlo.

Comandi di grafica:

Andiamo a vedere alcuni comandi di grafica

```
>> xx = linspace(0,1,0.11) (11 nodi equidistanti tra 0 ed 1)
ans =
    0
   0.1
   ...
   ...
   0.9
    1
```

```
>> yy=sin(xx);
>> plot(xx,yy)
```

Se voglio disegnare piú grafici contemporaneamente di due colori diversi

```
>> plot(x,y,'r',xx,yy,'k') (r = rosso, k = nero)
```

Osservazione: Usare i tasti direzionali della tastiera per tornare alle vecchie istruzioni.

```
>> plot(x,y,'or',...) (disegna dei cerchietti)
>> plot(x,y,'o-r',...) (cerchietti uniti da linee)
>> plot(x,y,'+k',...) (crocette)
```

Vediamo adesso come si scrive sulle interfacce grafiche

```
>> plot(x,y,xx,yy)
>> title('grafici di seno e coseno') (titolo del grafico)
>> xlabel('asse x') (scrive sull'asse delle x)
```

Ancora un esempio da linea di comando

```
>> clear
>> x=linspace(0,2,201);
>> y4=exp(-x).*sin(4*x);
>> plot(x,y4);
```

Vediamo adesso come si uniscono piú finestre grafiche contemporaneamente, utilizzeremo il comando "subplot":

```
>> subplot(2,2,1) %produce una finestra 2x2 in cui al primo
                  %posto posizionera' il seguente grafico
>> plot(x,y4); title('caso n = 4');
>> subplot(2,2,2); % questo grafico nel posto 1,2
>> plot(x,y5); title('caso n = 5');
```