Grafici

MATLAB consente di rappresentare graficamente funzioni e dati memorizzati in vettori e matrici. Di seguito sono presentati i principali comandi per rappresentare graficamente curve bidimensionali. Per rappresentare graficamente una funzione si utilizza il comando `plot`; la sintassi del comando è:

`plot` (*vettorex*, *vettorey*, *'opzioni'*)

dove *vettorex* e *vettorey* sono i vettori di dati (rispettivamente ascisse e ordinate dei punti) e *opzioni* è una stringa opzionale che definisce il tipo di colore, di simbolo e di linea che si vogliono usare nel grafico.
Per esempio per produrre il grafico della funzione $y = \sin t$ da 0 a 2π, si usa

```
>> t = 0:0.05:2*pi; y = sin(t);
>> plot(t,y)
```

Si possono creare grafici multipli con una singola chiamata; MATLAB automaticamente traccia i diversi grafici attraverso un predefinito (ma settabile dall'utente) elenco di colori che permette di distinguere ciascuna funzione. Per esempio, questi comandi definiscono e tracciano tre funzioni di t, ciascuna curva in un colore diverso. Si noti che gli intervalli delle ascisse possono anche essere diversi per le varie curve.

```
>> t1 = 0:0.05:pi; t2 = 0:0.05:2*pi; t3 = 0:0.05:3*pi;
>> y1 = sin(t1 - .4); y2 = sin(t2 - .8); y3 = sin(t3 - 1.2);
>> plot(t1, y1, t2, y2, t3, y3)
```

È possibile specificare colore, stile della linea, e marcatori dei punti con il seguente comando:

`plot` (*vettorex*, *vettorey*, *'colore-stile-marcatore'*)

dove il *colore-stile-marcatore* è costituito da una sequenza di caratteri che rappresentano un colore, uno stile di linea ed un tipo di marcatore; tipi di colore sono:

```
'c', 'm', 'y', 'r', 'g', 'b', 'w', e 'k'
```

Questi corrispondono a cyan, magenta, giallo, rosso, verde, azzurro, bianco, e nero.
Stringhe di stile di linea sono:

```
'-' per la linea continua
'--' per la linea tratteggiata
':'  per una linea a puntini
'-.' per una linea a puntini e tratteggio
'none' senza linea
```

Tipi di marcatore comuni sono: `'+'`, `'o'`, `'*'`, `'x'`

Ad esempio, il comando: `plot(x, y, 'r:+')` traccia in rosso una linea a puntini e pone un marcatore + a ciascun dato. Se specifica un tipo di marcatore ma non uno stile di linea, MATLAB disegna solamente il marcatore.

Per commentare un grafico sono disponibili i seguenti comandi:

Comando	**Significato**
title	inserisce un titolo nel grafico
xlabel	inserisce un nome per l'asse x
ylabel	inserisce un nome per l'asse y
grid on	inserisce una griglia sugli assi x ed y
legend	inserisce una legenda
text	inserisce una stringa di testo in una specificata posizione
gtext	inserisce una stringa di testo nella posizione selezionata tramite mouse

Esempio di script:

```
x=[-pi:0.2:pi];
y=sin(x);
plot(x,y,'-o')
title('Grafico della funzione sin(x)')
xlabel('x')
ylabel('y')
grid on
text(1.1,-0.3,'funzione dispari')
```

La seguente function, memorizzata come file **disegna.m**, permette di produrre una figura che contiene il grafico di una funzione inserita come stringa (si faccia attenzione ad usare sempre gli operatori `.*`, `./` e `.^`, e non gli operatori `*`, `/` e `^` nella definizione della funzione) e l'asse delle ascisse, in un intervallo di estremi a e b suddiviso in n punti. L'argomento n è opzionale e se omesso assume il valore 100:

```
function disegna(exprf,a,b,n)
%DISEGNA Funzione per tracciare il grafico di una
%        funzione in un intervallo [a,b]
%
% disegna(exprf,a,b,n)
%
% Dati di ingresso:
%   exprf:  stringa contenente l'espressione
%           della funzione (attenzione alle operazioni
%           che necessitano del . (punto)
%   a:      primo estremo
%   b:      secondo estremo
%   n:      numero di punti equidistanti dell'intervallo
%           in cui viene valutata la funzione.
%           Argomento facoltativo, default = 100.
```

```
if nargin < 4
    x = linspace(a,b);
else
    x = linspace(a,b,n);
end
f = inline(exprf);
y = feval(f,x);
asx = zeros(2,1);
plot(x,y,'r',[x(1),x(end)],asx,'k')
title(exprf);
```

Esempio di utilizzo:

```
>> a=10; b=20; n=150;
>> expr='3.*x.^2-2./x';
>> disegna(expr,a,b,n)
```