

Introduzione al MATLAB

di Marco Giancola

Il MATLAB è uno dei più diffusi software di calcolo. La prima versione fu creata alla fine degli anni '70 da Cleve Moler, il presidente del dipartimento di scienze informatiche dell'Università del Nuovo Messico, ed era destinata principalmente alla gestione di matrici. Si diffuse rapidamente nelle università e venne apprezzato soprattutto dai matematici applicati. In seguito, Moler, insieme a Jack Little (il quale riconobbe il potenziale commerciale di tale programma) e Steve Bangert, riscrisse il MATLAB nel linguaggio di programmazione C e fondò la società *MathWorks*, nel 1984, per proseguirne lo sviluppo. Il software è ormai giunto alla sua ottava versione, denominata *R2012b*, e possiede capacità che superano di gran lunga quelle della versione originale. Le sue principali funzionalità sono:

- calcolo numerico,
- analisi e visualizzazione di dati,
- programmazione e sviluppo di algoritmi,
- realizzazione di interfacce utenti che consentono anche di utilizzare programmi scritti in altri linguaggi.

MATLAB può essere integrato con altri software, prodotti da MathWorks, (il più famoso è probabilmente *Simulink*, un software per la modellazione, simulazione e analisi di sistemi dinamici) per:

- calcolo parallelo,
- analisi e progettazione di sistemi di controllo,
- elaborazione di segnali,
- elaborazione di immagini,
- test e misurazioni,
- finanza computazionale,
- biologia computazionale,
- generazione e verifica del codice,
- accesso a database e reportistica.

Una delle ragioni del successo di MATLAB va ricercata nel fatto che i suoi comandi sono molto simili al modo in cui esprimiamo i passaggi di un calcolo matematico; per tale motivo, scrivere soluzioni informatiche con MATLAB è più agevole che non scriverle in un altro linguaggio di programmazione.

MATLAB è un acronimo che sta per MATrix LABoratory (laboratorio della matrice), infatti la sua struttura dati principale è la matrice o, più precisamente, l'array, che, a seconda delle sue dimensioni, può corrispondere ad uno scalare, un vettore, una matrice o un tensore.

Finestre

L'interfaccia principale di MATLAB è composta da diverse finestre che è possibile affiancare, spostare, ridurre a icona e ridimensionare. Le più usate sono:

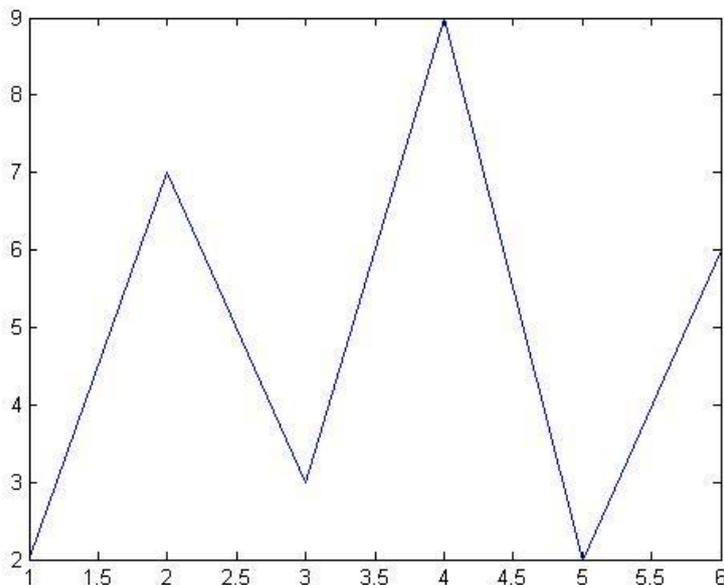
- finestra dei comandi,
- finestra grafica,
- finestra di edit.

Nella finestra dei comandi è possibile digitare i comandi del MATLAB e visualizzare in tempo reale i risultati, praticamente come se si trattasse di una calcolatrice scientifica. Per esempio, per calcolare la radice quadrata di 9, basta digitare il comando `sqrt(9)` e premere il tasto "Invio". Verrà mostrato come output: `ans = 3`, dove `ans` è una variabile standard nella quale viene memorizzato l'ultimo risultato generato.

La finestra grafica viene utilizzata per visualizzare grafici e disegni. Ad esempio, digitando nella finestra dei comandi:

```
plot([1 2 3 4 5 6],[2 7 3 9 2 6])
```

apparirà la finestra grafica raffigurante la seguente poligonale:



La finestra di edit viene usata per creare e modificare i file .m, ovvero quei file che contengono programmi scritti con la sintassi del MATLAB ed eseguibili mediante il MATLAB. Tale finestra appare digitando `edit` nella finestra dei comandi.

Alcuni dei comandi principali

`clc`: “pulisce” la finestra dei comandi.

`clear`: cancella tutte le variabili in memoria o una in particolare se specificata.

`clf`: “pulisce” la finestra grafica.

`clock`: fornisce l'ora e la data corrente.

`date`: fornisce la data corrente.

`edit`: fa apparire la finestra di edit.

`exit`: consente di uscire da MATLAB.

`help`: fornisce aiuto sull'argomento specificato.

`load`: carica in memoria le variabili presenti nel file specificato.

`plot`: genera un grafico cartesiano bidimensionale.

`print`: stampa ciò che appare nella finestra grafica.

`quit`: consente di uscire da MATLAB.

`save`: salva tutte le variabili in memoria nel file specificato.

`who`: elenca le variabili in memoria.

`whos`: elenca le variabili e le loro dimensioni.

Vettori e matrici

L'inserimento di un vettore o di una matrice viene effettuato usando le parentesi quadre e separando:

- gli elementi di ciascuna riga con spazi o virgole,
- le righe con “;” (oppure andando a capo).

Esempi:

vettore riga: `x = [1, 2, 3]` oppure `x = [1 2 3];`

vettore colonna: `y = [1; 4; 7];`

matrice: `A = [1 2 3; 4 5 6; 7 8 9]` oppure `A = [1 2 3`

4 5 6
7 8 9].

MATLAB mette a disposizione varie funzioni per la creazione e manipolazione di matrici e vettori; alcuni esempi sono:

`det`: calcola il determinante di una matrice quadrata;

`diag(X)`: se X è un vettore con n elementi, produce una matrice quadrata diagonale $n \times n$ con gli elementi di X sulla diagonale; se invece X è una matrice quadrata $n \times n$, produce un vettore di n elementi pari a quelli della diagonale di X ;

`eye(n)`: genera la matrice identità $n \times n$;

`inv`: calcola l'inversa di una matrice quadrata;

`length`: determina il numero di elementi di un vettore;

`ones(m,n)`: genera una matrice $m \times n$ di 1;

`rand(m,n)`: genera una matrice $m \times n$ di valori casuali compresi tra 0 e 1;

`size`: determina le dimensioni di una matrice;

`zeros(m,n)`: genera una matrice $m \times n$ di 0.

Il carattere ":" può essere usato per generare vettori, come mostrato nei seguenti esempi:

`t = 1:1:5` (o più semplicemente `t = 1:5`) equivale a `t = [1 2 3 4 5]`;

`t = 0:0.2:1` equivale a `t = [0 0.2 0.4 0.6 0.8 1]`;

`t = 2:-0.2:1` equivale a `t = [2 1.8 1.6 1.4 1.2 1]`.

Polinomi

MATLAB tratta i polinomi come particolari vettori riga i cui elementi sono i coefficienti dei monomi del polinomio in ordine di potenza decrescente. Ad esempio, il polinomio

$$x^4 - 2x^3 - x^2 - 3x + 4$$

corrisponde al vettore `[1, 2, -1, -3, 4]`.

La funzione `roots` calcola le radici del polinomio: fornendogli come argomento il vettore associato al polinomio, restituisce un vettore i cui elementi sono le sue radici.

Funzioni

MATLAB dispone di una ricchissima collezione di funzioni e operatori per il calcolo numerico, alcuni dei quali sono già stati esaminati. Tra le funzioni e gli operatori matematici più utili ci sono:

somma: +

sottrazione: -

moltiplicazione: *

divisione: /

elevamento a potenza: ^

esponenziale: `exp(x)`

logaritmo naturale: `log(x)`

logaritmo decimale: `log10(x)`

radice quadrata: `sqrt(x)`

seno: `sin(x)`

coseno: `cos(x)`

tangente: `tan(x)`

arcoseno: `asin(x)`

arcocoseno: `acos(x)`

arcotangente: `atan(x)`

seno iperbolico: `sinh(x)`

coseno iperbolico: `cosh(x)`

tangente iperbolica: `tanh(x)`
arcoseno iperbolico: `asinh(x)`
arcocoseno iperbolico: `acosh(x)`
segno: `sign(x)`
valore assoluto: `abs(x)`
arrotondamento: `round(x)`
troncamento: `fix(x)`
parte reale: `real(z)`
parte immaginaria: `imag(z)`
complesso coniugato: `conj(z)`

Grafici

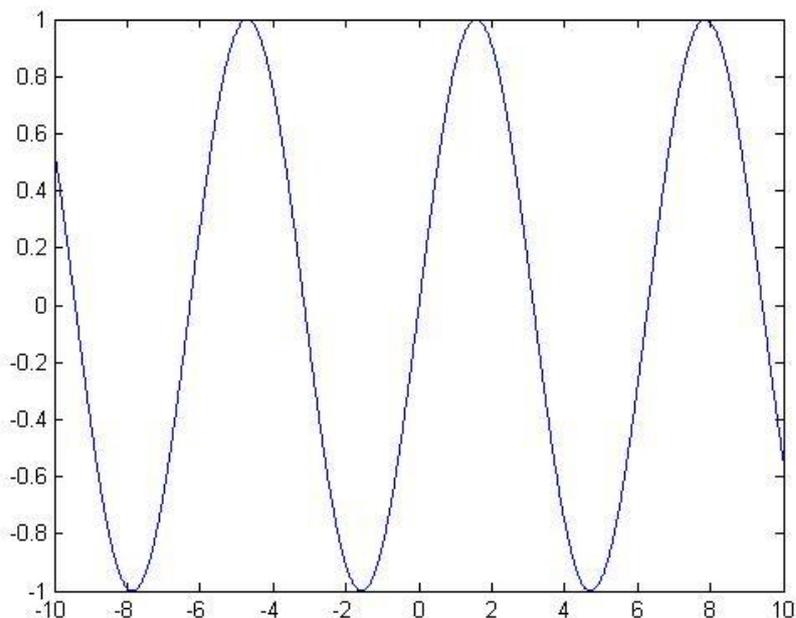
La funzione `plot` consente di creare grafici bidimensionali ricevendo in input due vettori della stessa lunghezza e “stampando” i punti aventi come coordinate gli elementi dei due vettori. Per tracciare il grafico di una qualsiasi funzione, è perciò necessario:

1. creare un opportuno vettore `x` contenente le ascisse,
2. passare `x` alla funzione per ricavare un vettore `y` contenente le ordinate,
3. usare la funzione `plot` sui due vettori `x` e `y` così ottenuti: `plot(x,y)`.

Ad esempio, per tracciare il grafico della funzione $\sin(x)$ tra $x = -10$ e $x = 10$, si possono usare le seguenti istruzioni:

```
x = -10:0.01:10;  
y = sin(x);  
plot(x,y)
```

che, digitate nella finestra dei comandi, dopo aver premuto “Invio”, fanno apparire il grafico sottostante:



Programmazione

Il linguaggio del MATLAB è un linguaggio di programmazione strutturata come, ad esempio, C, Pascal, Fortran, Ada e ALGOL. Proprio come tali linguaggi, esso possiede strutture condizionali

(if-else e switch-case) e cicliche (for e while). Va da se che anch'esso deve possedere gli operatori relazionali e logici:

maggiore: >

minore: <

maggiore o uguale: >=

minore o uguale: <=

uguale: ==

diverso: ~=

and: &

or: |

not: ~

Come già detto, il codice di un programma MATLAB va digitato nella finestra di edit e salvato in un file .m. Per eseguirlo, occorre digitare nella finestra dei comandi `run` seguito dal nome del file privo dell'estensione. Qui di seguito, è mostrato il codice di un semplice programma che:

1. chiede in input l'inclinazione di un cannone e la velocità con cui spara un proiettile;
2. mostra un messaggio di errore se l'angolo fornito non è compreso tra 0° e 180° ;
3. calcola e fornisce in output la massima altezza raggiunta dal proiettile e la gittata;
4. disegna la traiettoria del proiettile.

```
g = 9.81;
v = input('velocità del proiettile: ');
t = input('inclinazione del cannone: ');
if t<=0|t>=180
    error('valore non valido');
end
t = t*pi/180;
h = (v^2)*(sin(t)^2)/(2*g);
G = (v^2)*sin(2*t)/g;
fprintf('altezza: %f\ngittata: %f', h, G);
a = -g/(2*(v^2)*(cos(t)^2));
b = tan(t);
if G>0
    x = 0:0.1:G;
else
    x = G:0.1:0;
end
y = a*(x.^2) + b*x;
plot(x,y), xlabel('x'), ylabel('y'), grid, title('traiettoria')
```

Si può notare che, contrariamente alla maggior parte dei linguaggi di programmazione, per dichiarare una variabile in MATLAB, non occorre specificare a quale tipo di dato (numero intero, non intero, valore booleano, carattere, stringa, eccetera) appartiene, ma è sufficiente scrivere il nome della variabile seguito dall'operatore "=" e dal valore che le si vuole assegnare. Nel caso in cui il valore sia un carattere o una stringa, è necessario specificarlo inserendolo tra apici.

MATLAB è *case-sensitive* e accetta nomi di variabili lunghi fino ad un massimo di 19 caratteri alfanumerici, con il primo obbligatoriamente alfabetico.

Anche nei programmi MATLAB è possibile inserire dei commenti: ogni riga del commento deve essere preceduta dal carattere "%".

Bibliografia

Etter D. M., Kuncicky D. C.: *Introduzione a MATLAB*; Milano, APOGEO srl, 2001.

<http://it.emcelettronica.com/matlab-tutorial-concetti-avanzati>

<http://it.emcelettronica.com/matlab-tutorial-introduzione>

<http://it.wikipedia.org/wiki/MATLAB>
<http://it.wikipedia.org/wiki/Simulink>
<http://www.dsi.unifi.it/~angeli/lezioni/Mat00.pdf>
<http://www.math.unipd.it/~demarchi/matlab.pdf>
<http://www.mathworks.it/products/matlab>