

Il programma MATLAB

In queste pagine si introduce in maniera molto breve il programma di simulazione MATLAB (una abbreviazione di MATrix LABoratory).

Introduzione

MATLAB è un programma interattivo di calcolo che permette di risolvere problemi numerici senza che sia necessario scrivere esplicitamente una procedura in un linguaggio di programmazione ad alto livello. MATLAB si basa sul calcolo matriciale, e si può pensare ad esso come ad un sistema efficace per accedere ad una libreria di procedure di calcolo numerico molto sofisticate, con in più la possibilità di rappresentare graficamente i risultati. Ha una interfaccia molto semplice, ed un singolo comando permette di risolvere problemi come l'inversione di una matrice o la soluzione di un sistema di equazioni differenziali. Inoltre, una successione di comandi può essere raccolta in un file ed essere eseguita invocando, dal programma MATLAB, il nome del file stesso: è poi possibile definire delle funzioni che forniscono dei valori in uscita, di modo che MATLAB risulta esso stesso programmabile e fornisce la possibilità di effettuare simulazioni e calcoli, anche molto complessi, in modo semplice e diretto.

Una guida completa a MATLAB è al di là degli scopi di queste note: si ricorda comunque che MATLAB fornisce un sistema di aiuto in linea a cui è molto conveniente ricorrere. Dopo essere entrati in MATLAB, il comando `help` mostra l'elenco delle procedure per le quali è disponibile una descrizione; per avere informazioni specifiche su di un comando, occorre digitare `help nome_comando`. Ad esempio, con il comando `help exp` si ottengono informazioni sull'uso della funzione `exp` per il calcolo della funzione esponenziale. Per una dimostrazione delle potenzialità di MATLAB, si dia il comando `demo`.

Per ulteriori informazioni su come ottenere documentazione sull'uso di MATLAB, si rimanda alla fine di questa sezione.

Matrici in MATLAB

MATLAB lavora essenzialmente con un unico tipo di dati: le matrici. Queste sono vettori bidimensionali, con m righe e n colonne in generale, i cui singoli elementi sono variabili reali oppure complesse. Una matrice può essere creata introducendo esplicitamente i suoi valori mediante un comando di assegnazione, oppure come risultato di comandi o funzioni predefiniti. Ad esempio, il comando

```
a=[1 2 3; 4 5 6]
```

crea la matrice `a` di dimensioni 2×3 . Si noti che il punto e virgola separa una riga della matrice dalla riga successiva. Le variabili complesse sono gestite in maniera molto intuitiva: Il comando `a=[3+i*2 4]` crea un vettore con due elementi complessi.

MATLAB fornisce l'eco del risultato di ogni comando, a meno che questo non venga fatto seguire da punto e virgola. Ad esempio, il comando

```
a=[1 2 3; 4 5 6];
```

crea, come prima, la matrice `a`, senza che i suoi valori vengano visualizzati immediatamente dopo sul terminale. La possibilità di sopprimere l'eco è particolarmente utile quando si

scrivono programmi che contengono sequenze di comandi, oppure quando le dimensioni di una matrice siano particolarmente grandi.

Le seguenti operazioni fra matrici sono disponibili in MATLAB:

+	addizione
-	sottrazione
*	moltiplicazione
^	elevazione a potenza
'	trasposto coniugato
\	divisione a sinistra
/	divisione a destra.

Ad esempio, date le matrici A e b , il comando $x=A \backslash b$ fornisce, nel vettore x , la soluzione del sistema di equazioni $Ax = b$. Analogamente, $x=b/A$ è la soluzione di $xA = b$ (nel caso di matrice A rettangolare, MATLAB fornisce la soluzione del sistema ai *minimi quadrati*).

Le operazioni introdotte sono operazioni *matriciali*: qualora si vogliano effettuare operazioni sulle matrici *elemento per elemento*, gli operatori precedenti devono essere preceduti da un punto. Per esempio, il comando $[1 \ 2] .* [1 \ 3]$ fornisce il vettore $[1 \ 6]$, mentre l'ommissione del punto darebbe luogo ad un messaggio di errore, dato che la dimensione delle matrici non è compatibile con l'operazione di moltiplicazione matriciale.

MATLAB permette di creare vettori o matrici con una notazione conveniente per specificare gli intervalli.

Ad esempio, il comando $a=1:0.1:10$; crea un vettore di 91 elementi, spazati uniformemente con passo 0.1, il cui primo elemento è il valore 1 e l'ultimo elemento è il valore 10. Per accedere all'elemento i -esimo, si usa la notazione $a(i)$, mentre per selezionare gli elementi dall' i -esimo al j -esimo si usa la notazione $a(i:j)$. Ad esempio, il comando $b=a(2:30)$ crea un vettore b con gli elementi di a dal 2 al 30. Si noti che gli indici dei vettori in MATLAB iniziano da 1.

Più in generale, se a è una matrice $m \times n$, il comando $b=a(1:2, 2:3)$ crea una matrice b di dimensioni 2×2 che contiene le righe dalla 1 alla 2 e le colonne dalla 2 alla 3 della matrice a .

Strutture di controllo

In MATLAB è possibile utilizzare strutture di controllo analoghe a quelle che sono disponibili nei linguaggi di programmazione (**if**, **for**, **while**).

Struttura if. Il comando:

```
if t>=0
    x=1;
else
    x=0;
end
```

fornisce, nella variabile x , il valore della funzione gradino a tempi continui, calcolata in t .

Struttura for. I comandi

```
[m n]=size(a);
for i=1:m
    for j=1:n
        c(i,j)=2*a(i,j);
    end
end
c
```

creano e visualizzano la matrice c ottenuta moltiplicando per 2 gli elementi della matrice a , come se si fosse scritto $c=2*A$. Si noti che MATLAB permette di moltiplicare una matrice per uno scalare senza ricorrere a notazioni particolari.

Struttura while. La sequenza di comandi

```
sum=1; add=x;
while abs(add)>=1e-3
    sum=sum+add;
    add=x*add;
end
```

calcola la somma della serie geometrica $\sum_{k=0}^{\infty} x^k$, $|x| < 1$, fermandosi quando il valore assoluto del termine corrente è minore di 10^{-3} .

I seguenti operatori di relazione e logici sono definiti in MATLAB:

<	minore
>	maggiore
<=	minore uguale
>=	maggiore uguale
==	uguale
~=	diverso
&	AND
	OR
~	NOT

Si noti la differenza fra il simbolo usato per l'assegnazione (=) ed il simbolo usato per il confronto (==).

Funzioni predefinite

Le funzioni in MATLAB operano su scalari o sui singoli elementi delle matrici. In quest'ultimo caso, il risultato è una matrice della stessa dimensione di quella cui è stata applicata la funzione. Ad esempio, il comando $b=\text{sqrt}(a)$ crea una matrice b i cui elementi sono la radice quadrata degli elementi di a . Fanno eccezione le funzioni tipo max , min , sum , prod , mean , std , etc., che hanno come argomento una matrice e ritornano

un vettore. Ad esempio, `sum(a.^2)` dà come risultato un vettore che contiene la somma dei quadrati degli elementi di ciascuna colonna di `a`.

Visualizzazione grafica e stampa

MATLAB può produrre sia grafici di funzioni monodimensionali che curve di livello e grafici di funzioni a più dimensioni. Per una dimostrazione delle potenzialità grafiche di MATLAB, si può dare il comando `demo`.

Se `x` e `t` sono due vettori della stessa dimensione, il comando `plot(t,x)` apre una finestra con il grafico di `x` in funzione di `t`. Ad esempio, i comandi

```
t=0:0.01:2*pi; x=sin(t); y=cos(t); plot(t,x,t,y);
```

creano una nuova finestra con il grafico delle funzioni $x(t) = \sin(t)$ e $y(t) = \cos(t)$ fra 0 e 2π . È possibile definire il titolo del grafico, le caratteristiche degli assi, posizionare del testo all'interno del grafico: a questo proposito, con il comando `help`, si vedano le descrizioni dei comandi `plot`, `title`, `xlabel`, `ylabel`, `gtext`, `text`.

È inoltre possibile creare una copia su file del grafico, contenente i comandi necessari per la successiva stampa: il comando `print`, la cui documentazione è ottenibile con il comando `help print`, permette di gestire la maggior parte delle stampanti e dei linguaggi di descrizione grafica presenti nel mercato.

M-files

Come accennato nell'introduzione, è possibile raggruppare una sequenza di comandi MATLAB in un file esterno, creato con un editor di testi qualsiasi, che deve avere estensione `.m`. Se il nome del file è `nome_file.m`, dando il comando `nome_file` dall'interno di MATLAB, verrà eseguita la sequenza dei comandi contenuti nel file stesso. È opportuno far precedere i comandi di un M-file dal comando `clear` che annulla lo spazio delle variabili create fino a quel momento in MATLAB: questo affinché i risultati od i nomi di variabili precedentemente utilizzati non interferiscano con quelli usati nel file `.m`. Per vedere quali sono, in ogni momento, le variabili definite in MATLAB e la loro dimensione, si usino i comandi `who` e `whos`. È inoltre conveniente poter aggiungere dei commenti nel file: a questo scopo, si può far precedere una riga dal carattere `%`, che forza MATLAB a non interpretare i caratteri successivi. Se un comando si estende oltre la lunghezza di una riga, esso può essere continuato nella riga seguente interrompendo con tre caratteri di interpunzione:

```
variabile_lunga= ...  
matrice_A.^2;
```

Funzioni esterne

È possibile aggiungere nuove funzioni al vocabolario di MATLAB. I comandi relativi ad una funzione devono essere contenuti in un file esterno, ancora con estensione `.m`. La prima riga del file deve contenere la lista dei parametri della funzione (che possono essere sia scalari che matrici) e la parola chiave `function`. Ad esempio, volendo scrivere una funzione che calcola la media, la potenza e la deviazione standard campionarie dei valori contenuti in una matrice, possiamo creare il file `stats.m`:

```
% STATS: calcola la media, la potenza e deviazione standard campionarie  
% della matrice di ingresso
```

```
function [m,p,s] = stats(x)  
    n = prod(size(x));  
    m = sum(sum(x)) / n;  
    p = sum(sum(x.^2)) / n;  
    s = sqrt(p - m^2);  
end
```

Per calcolare media, potenza e deviazione standard della matrice **a**, e porre il risultato nelle variabili **media**, **pot**, **stdv**, possiamo dunque invocare la funzione con il comando

```
[media,pot,stdv]=stats(a)
```

Si possono avere un qualsiasi numero di argomenti di uscita e di ingresso: questi ultimi vengono passati alla funzione *per valore* e non possono essere modificati al suo interno. Se la funzione ha un solo argomento di uscita, le parentesi quadre possono essere omesse. Ad esempio, la funzione precedente potrebbe essere riscritta come:

```
% STATS: calcola la media, la potenza e deviazione standard campionarie  
% della matrice di ingresso
```

```
function y = stats(x)  
    n = prod(size(x));  
    y(1) = sum(sum(x)) / n;  
    y(2) = sum(sum(x.^2)) / n;  
    y(3) = sqrt(y(2) - y(1)^2);  
end
```

e darebbe come risultato un vettore con i valori desiderati. Le variabili utilizzate all'interno della funzione e che non compaiono nella lista dei parametri sono *locali* alla funzione stessa, e possono pertanto avere lo stesso nome di variabili definite in altre procedure senza che si abbiano pericolose interferenze.

Si noti che le prime due linee del file sono dei commenti: esse vengono visualizzate all'interno di MATLAB, nel caso specifico, se si dà il comando **help stats**. In realtà, molte delle funzioni predefinite di MATLAB sono esse stesse definite all'interno di M-files usando la sintassi appena vista.

Nota sull'uso di MATLAB per le esercitazioni del Laboratorio di Comunicazioni Elettriche

Per agevolare lo sviluppo delle esercitazioni, un certo numero di funzioni di uso comune sono state raccolte in altrettanti M-files, che sono direttamente disponibili all'utilizzatore. In particolare, si è costituita la seguente libreria di funzioni, la cui descrizione è ottenibile dall'interno di MATLAB usando il comando `help`:

Area.m
convoluzione.m
Energia.m
Gradino.m
Phi.m
phi.m
Q.m
Rect.m
Sinc.m
log2.m