Introduzione all’uso di MATLAB

Miscione Giuseppe

g.miscione@virgilio.it

Indice

- Descrizione di MATLAB
- Definizione di matrici
- Accesso agli elementi di una matrice
- Operazioni comuni sulle matrici
- Operazioni algebriche tra matrici
- Esempio: calcolo delle radici di un polinomio

Cos’è **MATLAB**

MATLAB è un ambiente di sviluppo corredato da un linguaggio di programmazione che permette la computazione tecnica in un ambiente di facile utilizzo e tramite una notazione matematica familiare.

Campi di utilizzo - 1

I campi di utilizzo del MATLAB sono molti, ad esempio:

- Matematica e calcolo numerico;
- Analisi dei dati e loro visualizzazione;
- Modellistica e simulazione;
- Disegno industriale;
- Sviluppo di applicazioni.

Campi di utilizzo - 2

- E’ possibile usare MATLAB come un linguaggio ad alto livello (es. C++, Java) per costruire applicazioni complesse o con interfaccia grafica;
- Nonostante ciò, è sconsigliabile farlo dato che il MATLAB è orientato principalmente al calcolo numerico.

Unità di elaborazione

- L’unità di elaborazione del MATLAB è la matrice (MATLAB = MATrix LABoratory);
- I numeri scalari sono, infatti, considerati come matrici 1x1 e i vettori come matrici 1xN o Nx1;
- Quindi il MATLAB può eseguire tutte le operazioni tra matrici e tutte quelle riconducibili ad operazioni tra matrici.

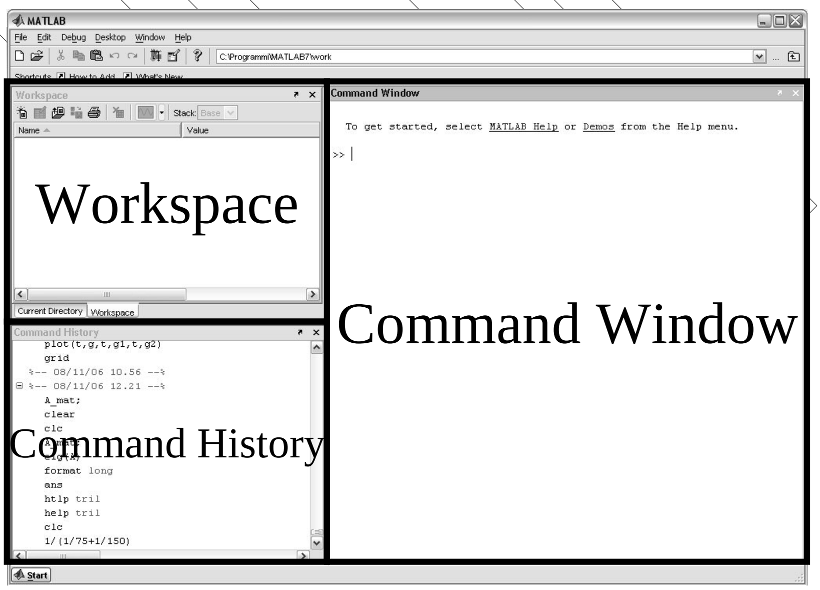
Comandi e variabili - 1

- Le operazioni che vogliamo far eseguire al MATLAB vengono espresse con comandi in un linguaggio di programmazione ad hoc;
- Una variabile è un nome che si può assegnare al risultato di un’operazione, in modo da poterlo richiamare facilmente in operazioni successive.

Comandi e variabili - 2

- Il MATLAB mette a disposizione dell’utente una serie di funzioni predefinite che permettono di eseguire praticamente tutte le operazioni di cui si ha, in genere, bisogno;
- Poiché tutte le operazioni producono matrici, ad ogni variabile sarà associata una matrice.

Interfaccia del MATLAB



Comandi e variabili - 3

- Con il comando “help” si ottiene l’indice del manuale dei comandi;
- Digitando “help <*nome comando*>” si ottiene una descrizione esaustiva della funzione e dell’uso del comando specificato.

Regole per i nomi di variabili

- Possono contenere solo i seguenti caratteri:
  - ABCDEFGHIJKLMNOPQRSTUVWYXZ
  - abcdefghijklmnopqrstuvwyxz
  - 1234567890
  - _ (il carattere “underscore”)
- Non possono contenere spazi;
- Non possono cominciare con una cifra;
- Sono case-sensitive.

Nomi di variabili - esempi

- risultato2
- epsilon_r
- Matrice
- nome ≠ Nome ≠ NoMe ≠ nOmE
- ~~2risultato~~
- ~~epsilon r~~

Variabili predefinite

- **i, j** : unità immaginaria per costruire numeri complessi (es.: c=1+3i);
- **pi** : il valore di π;
- **inf, NaN** : valori di errore derivanti, ad esempio, da divisioni per 0.

Esempio guida - 1

Risolviamo il sistema lineare:

$$\begin{cases} x_1 & -x_2 & & -6\cdot x_4 & = & -10 \\ 2\cdot x_1 & & -3\cdot x_3 & 2\cdot x_4 & = & 3 \\ 5\cdot x_1 & +x_2 & -2\cdot x_3 & & = & 2 \\ & & x_3 & -x_4 & = & -1 \end{cases}$$

Esempio guida - 2

Riscriviamolo come:

$$\begin{bmatrix} 1 & -1 & 0 & -6 \\ 2 & 0 & -3 & 2 \\ 5 & 1 & -2 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -10 \\ 3 \\ 2 \\ -1 \end{bmatrix}$$

Esempio guida - 3

$$A = \begin{bmatrix} 1 & -1 & 0 & -6 \\ 2 & 0 & -3 & 2 \\ 5 & 1 & -2 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \qquad b = \begin{bmatrix} -10 \\ 3 \\ 2 \\ -1 \end{bmatrix}$$

Esempio guida - 4

$\det(A) \neq 0 \Rightarrow$ **Il sistema può avere una ed una sola soluzione**

$rango(A) = rango([A|b]) \Rightarrow$ **Il sistema ha una ed una solo soluzione**

$x = A^{-1} \cdot b \Rightarrow$ **Si trova la soluzione**

Esempio guida - 5

I comandi MATLAB che risolvono il problema:

```
b = [ -10; 3; 2; -1 ]
A11 = [1 -1; 2 0; 5 1];
A21 = zeros(1,2);
A12 = [0 -6; -3 2; -2 0];
A22 = [1 -1];
```

Esempio guida - 6

```
A = [A11 A12; A21 A22]
Ab = [ A b ]
d = det(A)
rA = rank(A)
rAb = rank(Ab)
x = inv(A)*b
```

Definizione di matrici

- Ci sono vari modi per definire le matrici:
  - Facendo un elenco esplicito degli elementi della matrice;
  - Usando funzioni predefinite del MATLAB;
  - Definendo la matrice per blocchi;
  - Per intervalli.

Definizione di matrici con elenco esplicito - 1

- La definizione degli elementi della matrice va racchiusa tra le parentesi quadre [e];
- La matrice va definita per righe, dall'alto in basso.

Definizione di matrici con elenco esplicito - 2

- Gli elementi su una riga vanno separati da uno spazio o da una virgola;
- Le righe vanno separate da un punto e virgola.

Definizione di matrici con elenco esplicito - esempio

$$b = \begin{bmatrix} -10 \\ 3 \\ 2 \\ -1 \end{bmatrix}$$

b = [-10; 3; 2; -1]

Definizione di matrici con elenco esplicito - note

- Gli scalari possono essere definiti senza ricorrere alle parentesi quadre

  (es.: A = 1);
- Facendo seguire la definizione di una variabile da un carattere “;” si evita l’eco del comando nella Command Window

  (es.: b = [-10; 3; 2; -1];)

Definizione di matrici con funzioni predefinite - 1

- **A = zeros(m,n)** : crea una matrice di dimensione mxn i cui elementi sono tutti 0;
- **A = ones(m,n)** : crea una matrice di dimensione mxn i cui elementi sono tutti 1;
- **A = eye(m)** : crea la matrice identità di dimensione mxm;

Definizione di matrici con funzioni predefinite - 2

- **A = diag(B)**
  - se B è una matrice, A sarà un vettore i cui elementi sono uguali a quelli sulla diagonale di B;
  - se B è un vettore, A è una matrice diagonale con gli elementi di B sulla diagonale principale;

Definizione di matrici con funzioni predefinite - 3

$$\begin{array}{c} \begin{array}{cccc} k=0 & k=1 & k=2 & \end{array} \\ A=\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \\ \begin{array}{cc} k=-2 & k=-1 \end{array} \end{array}$$

Definizione di matrici con funzioni predefinite - 4

- **A = diag(B, k)**
  - se B è una matrice, A sarà un vettore i cui elementi sono uguali a quelli sulla k-esima diagonale di B;
  - se B è un vettore, A è una matrice con gli elementi di B sulla k-esima diagonale;

Definizione di matrici per blocchi

- Si ricorre a regole identiche a quelle già descritte, ma gli elementi che compaiono nelle definizione non sono valori scalari ma altre matrici.

Definizione di matrici - esempio

$$A = \left[\begin{array}{cc|cc} 1 & -1 & 0 & -6 \\ 2 & 0 & -3 & 2 \\ 5 & 1 & -2 & 0 \\ \hline 0 & 0 & 1 & -1 \end{array}\right] = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

A11 = [1 -1; 2 0; 5 1];   A21= zeros(1,2);

A12 = [0 -6; -3 2; -2 0]; A22=[1 –1];

A = [A11 A12; A21 A22]

Definizione di matrici - esempio

$$Ab = \left[A \middle| b \right]$$

Ab = [A b]

Definizione di matrici per intervalli

- **t = [a:delta:b]** : crea un vettore i cui elementi sono i seguenti:

t = (a, a+delta, a+2·delta, a+3·delta, …, b)

t=[1:2:5] ➔ t=(1,3,5);

t=[5:-2:1] ➔ t=(5,3,1);

t=[1:5] ➔ t=(1,2,3,4,5);

- **t = linspace(a, b, n)** : crea un vettore di *n* elementi compresi tra *a* e *b* con spaziatura costante;

Eliminazione di variabili dal workspace

- Con il comando "clear" si cancellano tutte le variabili del workspace;
- Con il comando

  "clear <*nome variabile*>"

  si cancella solo la variabile indicata.

Accesso agli elementi di una matrice - 1

- **A(i, j)** : restituisce l'elemento in posizione (i,j) della matrice A. Se A è una matrice mxn, si deve avere che $1 \leq i \leq$ m e $1 \leq j \leq$ n;
- **A(:, j)** : restituisce la j-esima colonna di A;
- **A(i, :)** : restituisce la i-esima riga di A;

Accesso agli elementi di una matrice - 2

- **A(i1:i2, j1:j2)** : estrae la sottomatrice di A così definita:

$$\begin{bmatrix} A_{i1,j1} & A_{i1,j1+1} & \dots & A_{i1,j2} \\ A_{i1+1,j1} & A_{i1+1,j1+1} & \dots & A_{i1+1,j2} \\ \dots & \dots & \dots & \dots \\ A_{i2,j1} & A_{i2,j1+1} & \dots & A_{i2,j2} \end{bmatrix}$$

Operazioni comuni sulle matrici - 1

- **A’** : calcola la trasposta di A;
- **rank(A)** : calcola il rango di A;
- **det(A)** : calcola il determinante di A (che deve essere quadrata);
- **inv(A)** : calcola l’inversa di A;

Operazioni comuni sulle matrici - esempio

$$\det(A) \neq 0$$

$$rango(A) = rango(Ab)$$

$$x = A^{-1} \cdot b$$

```
d = det(A)
rA= rank(A)
rAb= rank(Ab)
x = inv(A)*b
```

Operazioni comuni sulle matrici - 2

- **aval=eig(A)** : calcola gli autovalori (eigenvalues, in inglese) di A e li restituisce nel vettore *aval*;
- **[avet, aval]=eig(A)** : calcola gli autovalori e gli autovettori di A; *aval* è una matrice diagonale e *avet* è una matrice tale che *avet(:, j)* è l'autovettore associato all'autovalore *aval(j, j)*;

Operazioni comuni sulle matrici - 3

- **[righe,colonne]=size(A)** : restituisce le dimensioni della matrice *A*;
- **length(b)** : restuisce la lunghezza del vettore *b*;
- **fliplr(A)** : inverte l'ordine delle colonne di A;
- **flipud(A)** : inverte l'ordine delle righe di A;

Operazioni algebriche sulle matrici - 1

- **C = A+B** : calcola la somma di A e B (che devono avere le stesse dimensioni);
- **C = A-B** : calcola la differenza tra A e B (che devono avere le stesse dimensioni);
- Se A o B sono scalari, la somma (o differenza) è fatta elemento per elemento.

Operazioni algebriche sulle matrici - 2

- **C = A*B** : calcola il prodotto matriciale;

$$C_{i,j} = \sum_{k=1}^{p} A_{i,k} \cdot B_{k,j}$$
$$A \in \Re^{m \times p}, B \in \Re^{p \times n},$$
$$C \in \Re^{m \times n}$$

- **C = A.*B** : calcola il prodotto elemento per elemento;

$$C_{i,j} = A_{i,j} \cdot B_{i,j}$$
$$A \in \Re^{m \times n}, B \in \Re^{m \times n},$$
$$C \in \Re^{m \times n}$$

Operazioni algebriche sulle matrici - 3

- **C = A^n** : calcola AxAxAx…xA ;
- **C = A.^B** : calcola l'elevamento a potenza elemento per elemento;

$$C_{i,j} = (A_{i,j})^{B_{i,j}}$$
$$A \in \Re^{m \times n}, B \in \Re^{m \times n},$$
$$C \in \Re^{m \times n}$$

Operazioni algebriche sulle matrici - 4

- **C = A./B** : calcola il rapporto elemento per elemento;

$$C_{i,j} = A_{i,j} / B_{i,j}$$
$$A \in \Re^{m \times n}, B \in \Re^{m \times n},$$
$$C \in \Re^{m \times n}$$

- **C = A.\B** : calcola il rapporto elemento per elemento;

$$C_{i,j} = B_{i,j} / A_{i,j}$$
$$A \in \Re^{m \times n}, B \in \Re^{m \times n},$$
$$C \in \Re^{m \times n}$$

Operazioni algebriche sulle matrici - 4

- **C = A/B** : equivale ad eseguire A*inv(B), ma usa un algoritmo più veloce e preciso dell'inversione;
- **C = A\B** : equivale ad eseguire inv(A)*B, ma usa un algoritmo più veloce e preciso dell'inversione;

Esempio – radici di polinomio

$$p(\lambda) = \lambda^n + p_1\lambda^{n-1} + p_2\lambda^{n-2} + \ldots + p_{n-1}\lambda + p_n$$

$$A = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 1 & 0 \\ 0 & 0 & 0 & \ldots & 0 & 1 \\ -p_n & -p_{n-1} & -p_{n-2} & \ldots & -p_2 & -p_1 \end{bmatrix}$$

$$\det(\lambda I - A) = p(\lambda)$$

Esempio – radici di polinomio

$$p(\lambda) = \lambda^4 - 1.6\lambda^3 - 3.85\lambda^2 - 1.1\lambda + 0.15$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -0.15 & 1.1 & 3.85 & 1.6 \end{bmatrix}$$

Esempio – radici di polinomio

```
p = [1 -1.6 -3.85 -1.1 0.15]
lunghP = length(p)
p1 = p(2:lunghP)
lunghP1 = length(p1)
A = diag(ones(1,lunghP1-1),1)
A(lunghP1,:) = - fliplr(p1)
radici = roots(p)
radici_prova = eig(A)
```