UD3-MATLAB

Funzioni. “Colon notation”

Funzioni su scalari

- Le funzioni di MATLAB operano essenzialmente su scalari;
- Se applicate su matrici operano sui singoli elementi;
- Principali funzioni:

```
sin asin exp abs round cos acos
log       (logaritmo naturale)
sqrt floor tan atan
rem       (resto della divisione intera)
sign ceil
```

Funzioni su Vettori

- Funzioni che operano su vettori (riga o colonna);
- Operano rispetto alle colonne su matrici `m x n` se `m>=2`, e generano un vettore riga dei risultati della funzione applicata ad ogni colonna;
- Es: `A = [1 2 6 -5]`

```
[Y,I] = max(A)  => [6, 3]       Y = valore massimo
                                I = indice del valore
                                    massimo
```

- Es:

```
B = [ 1    -2;
      7    -4]

min(B) => [1 -4]
```

Funzioni su Matrici

- ```
  eig       autovalori e autovettori
  inv       inversa
  sqrtm     radice quadrata di matrice
  poly      polinomio caratteristico
  det       determinante
  size      dimensione
  rank      rango della matrice
  ```

- Es:

```
    A =       2     1
             -2     4

>> inv(A)

    ans =  0.4000   -0.1000
           0.2000    0.2000
```
```

Funzioni su Matrici

```
>> A =     [ 2      1
            -2      4 ]
>> det(A)
   ans = 10
>> eig(A)
   ans =    3.0000 + 1.0000i
            3.0000 - 1.0000i
>> [U,D] = eig(A)
```

U = Matrice le cui colonne sono gli autovettori di A

```
U = [0.4082 - 0.4082i   0.4082 + 0.4082i
     0.8165             0.8165           ]
```

D = Matrice diagonale con gli autovalori di A sulla diagonale

```
D = [3.0000 + 1.0000i        0
            0          3.0000 - 1.0000i ]
```

Modifiche dalla linea comandi e recall

- Uso delle frecce per ← → per spostarsi sulla linea dei comandi;
- Uso delle frecce ↑ ↓ per richiamare comandi precedenti o successivi;
- La linea dei comandi può essere utile per confrontare i risultati di piccole funzioni, per richiamarle e modificare velocemente i parametri;
- Es: Confronto fra i grafici delle funzioni `y=sin(m*x)` e `z=cos(n*x)` nell'intervallo $[0,2\pi]$

```
m=2; n=3; x=0:.01:2*pi; y=sin(m*x); z=cos(n*x);
plot(x,y,x,z)
```

Colon notation (notazione “:”)

- Vettori e sottomatrici sono spesso usati in MATLAB per gestioni complesse di dati;
- “Colon notation” = notazione “:” (usata sia per generare vettori che riferimenti a sottomatrici) sono utili per gestire questi oggetti;
- L’uso della “colon notation” consente di ridurre l’uso di cicli iterativi (che rallentano MATLAB) e di rendere il codice più semplice e leggibile;
- Es:
  – L’espressione 1:5 (già incontrata nel costrutto for) è in effetti un vettore riga [1 2 3 4 5].
- I valori della “colon notation” (estremi dell’intervallo e incrementi) possono essere sia interi che floating point:
- Es:

```
0.2:0.2:1.2  corrisponde a  [0.2, 0.4, 0.6, 0.8, 1.0, 1.2]
5:-1:1       corrisponde a  [5 4 3 2 1].
```

Uso della “Colon notation”

```
>> x = [0.0:0.1:2.0]';
   y = sin(x);
   [x y]

>> ans =
         0         0
    0.1000    0.0998
    0.2000    0.1987
    0.3000    0.2955
      …         …
    1.5000    0.9975
    1.6000    0.9996
    1.7000    0.9917
    1.8000    0.9738
    1.9000    0.9463
    2.0000    0.9093
```

Accesso a sottomatrici mediante “colon notation”

```
A = [ 2       1      3
     -2       4     -6
       1      2      9]
>> A (1:2, 1)
ans =
      2
     -2
>> A (1:2, 2)
ans =
      1
      4
>> A(1:2, 2:3)
ans =
      1     3
      4    -6
```

Accesso a sottomatrici mediante “colon notation”

- Il solo “:” denota tutte le riga o colonne;
- Es:

```
      A = [ 2       1       3
           -2       4      -6
            1       2       9]
>> A (:, 1)
ans =
     2
    -2
     1
>> A (2, :)
ans =
    -2     4    -6
```

Accesso a sottomatrici mediante “colon notation”

- E’ possibile utilizzare la notazione a blocchi per accedere a sottomatrici;
- Es:

```
    A = [ 2      1      3
         -2      4     -6
          1      2      9]
>> A (:, [2 3])
>>ans =
     1     3
     4    -6
     2     9
```

Accesso a sottomatrici mediante “colon notation”

- Le notazioni precedenti possono essere utilizzate anche nelle istruzioni di assegnazione;
- Es:

```
A = [ 2    1     3      B = [  1  2
     -2    4    -6             1  2
      1    2     9]            1  2 ]
```

```
>> A(:,[2 3]) = B(:,1:2)
```

sostituisce le colonne 2,3 di A con le prime due di B.

```
A =
     2     1     2
    -2     1     2
     1     1     2
```

Accesso a sottomatrici mediante "colon notation"

```
A = [ 2     1     3      B = [ 1   2
     -2     4    -6            1   2]
      1     2     9]

A(1:2, [2, 3]) = B (:,1:2)

A =
      2     1     2
     -2     1     2
      1     2     9
```

Operazioni su matrici e “colon notation”

```
A = [ 2   1    3
     -2   4   -6
      1   2    9]
```

- Prodotto scalare fra le colonne 1 e 3 di A e la matrice [1 2; 3 4]

```
A(:,[1,3]) = A(:,[1,3])*[1 2;3 4]

A =
    11      1     16
   -20      4    -28
    28      2     38
```

Operazioni su vettori e “colon notation”

- Dato un vettore `x = [1 2 3 4]`, cos’è `x = x(4:-1:1)` ?

```
>> x = [4 3 2 1]
```