

INTRODUZIONE A MATLAB

GEOMETRIA PER ING. BIOMEDICA A.A. 2015-2016

ANNA CODISPOTI Codispoti@dima.unige.it

OUTLINE

Programmare in Matlab

- Funzioni
- Cicli
- Operatori relazionali

Esercizi vari



LE FUNZIONI I

```
function [y1,...,yn] = nome_funzione(x1,...,xn)
```

Script

- parametri in ingresso non modificabili
- le variabili usate sono messe nella memoria di lavoro di MATLAB

Funzioni

- script al quale si possono passare parametri in ingresso ed ottenerne in uscita
 - y_1, \dots, y_n -> parametri in uscita
 - x_1, \dots, x_n -> parametri in entrata
- le variabili usate all'interno sono locali

LE FUNZIONI II

L'm file va salvato col nome `nome_funzione.m`

- il nome del file deve essere identico a quello della funzione

La funzione puo' essere richiamata

- dalla finestra di comando
- all'interno di uno script
- da altre funzioni

digitando `[y1,...,yn] = nome_funzione(x1,...,xn)`

Per poter richiamare la funzione ci dobbiamo mettere nella directory nella quale la salviamo

ESEMPIO

```
Editor - E:\Dottorato\corso\file_matlab\traccia.m
File Edit Text Cell Debug Desktop Window Help
Stack: []
1 function tr = traccia(A)
2
3 d = diag(A);
4 tr = sum(d);
```

- creiamo un m-file traccia.m nel quale implementiamo la funzione che calcola la traccia di una matrice
- A è una variabile temporanea

```
MATLAB
File Edit Debug Desktop Window Help
Current Directory: E:\Dottorato\corso\file_matlab
Shortcuts [?] How to Add [?] What's New
>> .
>>
>>
>>
>>
>> B = [1 2; 3 4];
>> t = traccia(B)
t =
    5
>>
```

- richiamiamo la funzione nella finestra di comando
- B è il parametro in input

CICLO FOR...END

Ciclo incondizionato


```
Editor - E:\Dottorato\corso\file_matlab\media.m
File Edit Text Cell Tools Debug Desktop Window Help
Stack: Ba...
- 1.0 + 1.1 x
1 function m = media(v)
2 % calcola la media del
3 % vettore v
4
5 n = length(v);
6 temp = 0;
7 for i=1:n
8     temp = temp+v(i);
9 end
10 m = temp/n;
```

for i = n1:passo:n2
 blocco di istruzioni
end

ESERCIZIO 1

Scrivere una funzione che sommi gli elementi di un vettore e moltiplichi il risultato per un intero n passato come parametro di input

function s = somma(v,n)



ESERCIZIO 2

Scrivere una funzione che faccia la media degli elementi di una matrice A

- $[r \ c] = \text{size}(A)$

```
function m = media_matrix(A)
```



OPERATORI

Operatori relazionali:

- < <= > >= == = ~=

(il simbolo ~ si ottiene come alt+1 2 6 nella tastiera numerica)

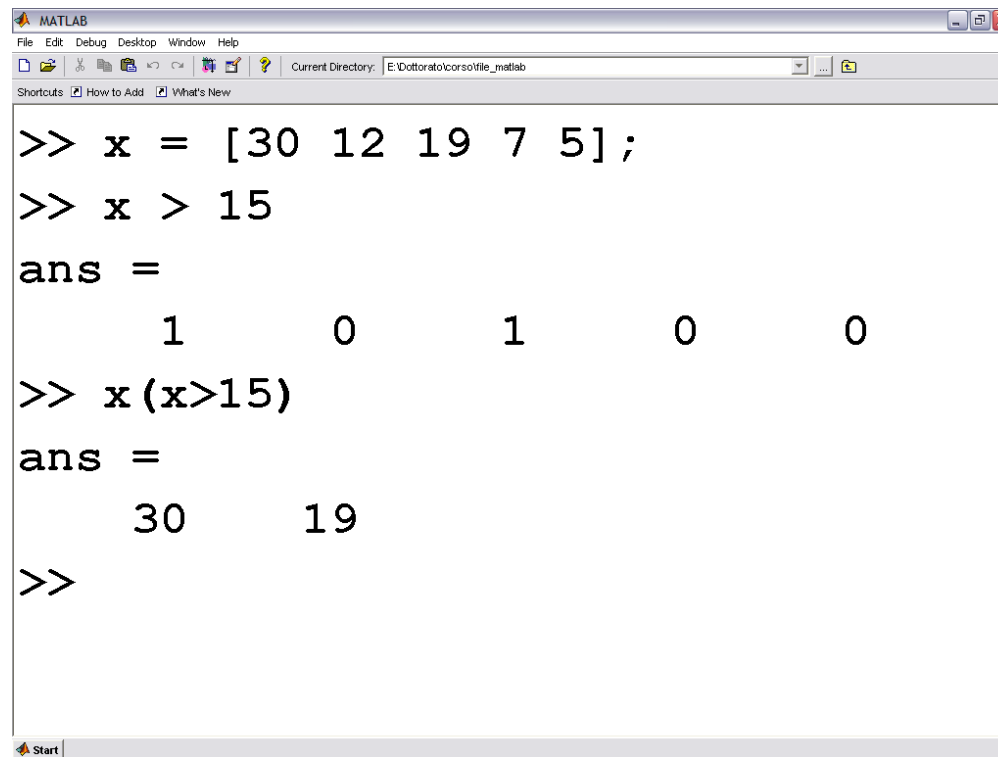
- si usano per confrontare tra di loro gli elementi di 2 matrici; il risultato dell'operazione sarà
 - 0 se la relazione è falsa
 - 1 se la relazione è vera

Operatori logici:

- && ||

- si usano per combinare tra loro gli operatori relazionali

ESEMPIO



```
MATLAB
File Edit Debug Desktop Window Help
Current Directory: E:\Dottorato\corso\file_matlab
Shortcuts How to Add What's New

>> x = [30 12 19 7 5];
>> x > 15
ans =
     1     0     1     0     0
>> x(x>15)
ans =
     30     19
>>
```

The image shows a MATLAB command window with the following content:

```
MATLAB
File Edit Debug Desktop Window Help
Current Directory: E:\Dottorato\corso\file_matlab
Shortcuts How to Add What's New

>> x = [30 12 19 7 5];
>> x > 15
ans =
     1     0     1     0     0
>> x(x>15)
ans =
     30     19
>>
```

CICLO WHILE...END

Ciclo condizionato

```
Editor - E:\Dottorato\corso\file_matlab\media_w.m*
File Edit Text Cell Tools Debug Desktop Window Help
Stack: Ba...
Evaluate cell and advance
1 function m = media_w(v)
2 % calcola la media del
3 % vettore v
4 n = length(v);
5 temp = 0;
6 i=1;
7 while i <= n
8     temp = temp+v(i);
9     i = i+1;
10 end
11 m = temp/n;
```

while *condizione*
blocco di istruzioni
end

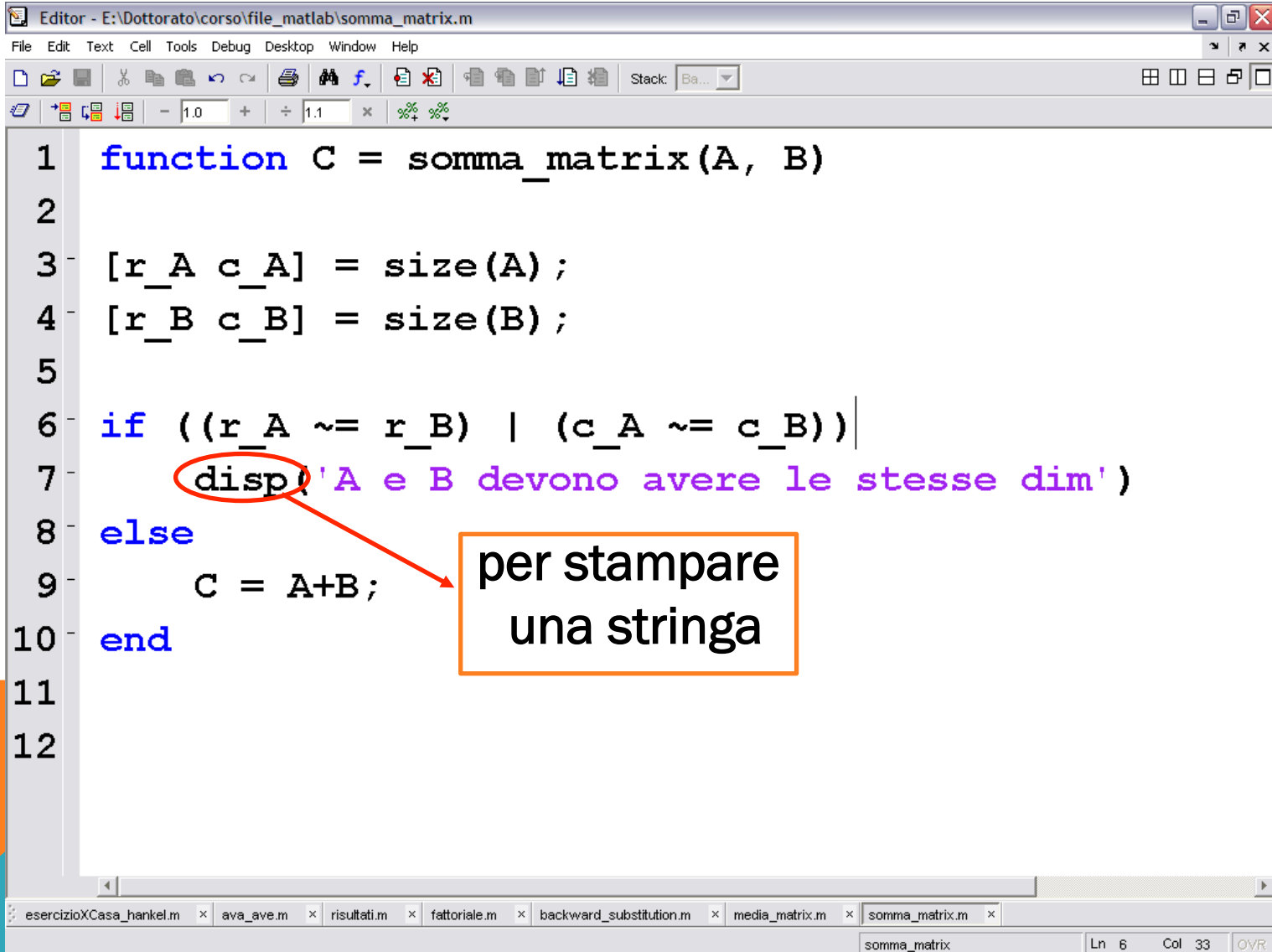
IF...ELSE...END

Test condizionale

```
if condizione1
    blocco di istruzioni
elseif condizione2
    blocco di istruzioni
else
    blocco di istruzioni
end
```

- Se *condizione1* è vera si eseguono i comandi
- I comandi che seguono *elseif* sono eseguiti se *condizione2* è vera
- I comandi che seguono *else* sono eseguiti se le precedenti condizioni sono false

ESEMPIO



```
1 function C = somma_matrix(A, B)
2
3 [r_A c_A] = size(A);
4 [r_B c_B] = size(B);
5
6 if ((r_A ~= r_B) | (c_A ~= c_B))
7     disp('A e B devono avere le stesse dim')
8 else
9     C = A+B;
10 end
11
12
```

per stampare una stringa

esercizioXCasa_hankel.m x ava_ave.m x risultati.m x fattoriale.m x backward_substitution.m x media_matrix.m x somma_matrix.m x

somma_matrix Ln 6 Col 33 OVR

COMANDI “UTILI”

break -> per uscire in maniera forzata da un ciclo

- MATLAB salta all'istruzione che termina il ciclo

return -> interrompe l'esecuzione della funzione

- si ritorna al programma da cui la funzione è stata chiamata

Osservazione:

- Privilegiare operazioni vettoriali ai cicli `for...end` e `while...end`

INPUT\OUTPUT

- Input → `n = input('inserisci un intero');`
- Sprintf → `s = sprintf('n = %d',n);`
- Disp → `disp(s), disp('stringa di caratteri')`

ESERCIZIO 3

Scrivere una funzione che ha

- come output la soluzione del sistema $Ax=b$
- verificare
 - A quadrata
 - A non singolare
- nel caso di A singolare o quadrata ritornare una scritta di errore

function x = risolvi_sistema_quadrato(A,b)



ESERCIZIO 4

Scrivere una funzione che dato in input un intero positivo n restituisca il suo fattoriale

- fare i test su n (n intero, positivo)

```
function r = fattoriale(n)
```