

CICLI IN MATLAB

FOR

In Matlab la ripetizione di blocchi di istruzioni per un numero di volte specificato e in modo incondizionato viene eseguita tramite l'istruzione di ciclo **FOR ...END** la cui sintassi è:

```
for indice = espressione  
  
    blocco di istruzioni  
  
end
```

Dove **indice** è una quantità che assume diversi valori a seconda di **espressione** e **end** segna la fine del blocco di istruzioni da ripetere.

Spesso **espressione** ha la forma **x1:x2** oppure **x1:step:x2**. **indice** viene utilizzato come contatore, **x1** è il valore iniziale assunto dal contatore, **step** l'incremento dato al contatore ad ogni ciclo (in mancanza di **step** l'incremento è 1), **x2** il valore finale che controlla il ciclo. Il blocco di istruzioni verrà ripetuto tante volte quanti sono i valori che assume il contatore.

Esempio:

```
function f = fatt(n)  
f=1;  
for k=n:-1:2;    %indice=valore iniziale:passo:valore finale  
    f=f*k;  
end              %ogni ciclo for termina con un end
```

n è il parametro in ingresso della function **fatt** che calcola il prodotto di tutti i primi numeri interi a decrescere da **n** a 2 e produce, in uscita, il risultato (il fattoriale di **n**).

Più cicli **for** possono essere annidati l'uno all'interno dell'altro.

IF

Se una o più istruzioni devono essere eseguite solo sotto condizione si usa l'istruzione **IF** la cui sintassi nella forma più semplice è:

```
if condizione
```

```
    blocco di istruzioni
```

```
end
```

dove **condizione** può essere un test di confronto tra due espressioni numeriche che utilizza i seguenti caratteri:

<	minore di
<=	minore o uguale di
>	maggiore di
>=	maggiore o uguale di
=	uguale a
~	diverso da

oppure una combinazione di singoli test combinati per mezzo dei seguenti operatori logici:

&	and
	or
~	not

Il *blocco di istruzioni* sarà eseguito solo se è verificata la **condizione**.

La sintassi più generale dell'istruzione **if** è:

```
if condizione1
```

```
    blocco di istruzioni
```

```
elseif condizione2
```

```
    blocco di istruzioni
```

```
    ...
```

```
else
```

```
    blocco di istruzioni
```

```
end
```

Dove il primo *blocco di istruzioni* sarà eseguito solo se la **condizione1** risulta essere verificata, il secondo solo se la **condizione1** risulta essere falsa e la **condizione2** vera ecc. Il *blocco di istruzioni* dopo **else** sarà eseguito soltanto se nessuna delle precedenti condizioni risulterà vera.

Esempio:

```
%esif.m
m = 4; %numero righe
n = 6; %numero colonne
A = ones(m,n); %matrice mxn con elementi tutti pari a 1
for c = 1:n
    for r = 1:m

        if r == c
            A(r,c) = 2;
        elseif abs(r-c) == 1
            A(r,c) = -1;
        else
            A(r,c) = 0;
        end
    end
end
disp (A)
```

WHILE

Se si ha la necessità di ripetere una o più istruzioni fintanto che una condizione sarà verificata non sapendo a priori il numero di ripetizioni, è necessario usare l'istruzione **WHILE ...END** la cui sintassi è:

```
while condizione

    blocco di istruzioni

end
```

Dove *blocco di istruzioni* verra` eseguito fintanto che **condizione** risulta vera.

Esempio:

Dovessimo sommare tutti i voti di uno studente (ad esempio per farne la media), senza doverne contare a priori il numero, potremmo costruire questo script:

```
nvoti=0; somma=0;
voto=input('voto? (0 per finire) ');
while voto~=0
    somma=somma+voto;
    nvoti=nvoti+1;
    voto=input('voto? (0 per finire) ');
end
media=somma/nvoti;
disp (media)
```

Break

break permette di terminare immediatamente l'esecuzione di un ciclo **for** o **while** (ad esempio in caso di errore). Quando è eseguita l'istruzione **break** Matlab salta automaticamente all'istruzione **end** che termina il ciclo.

Se vi sono più cicli annidati, **break** termina solo quello più interno in cui si trova