

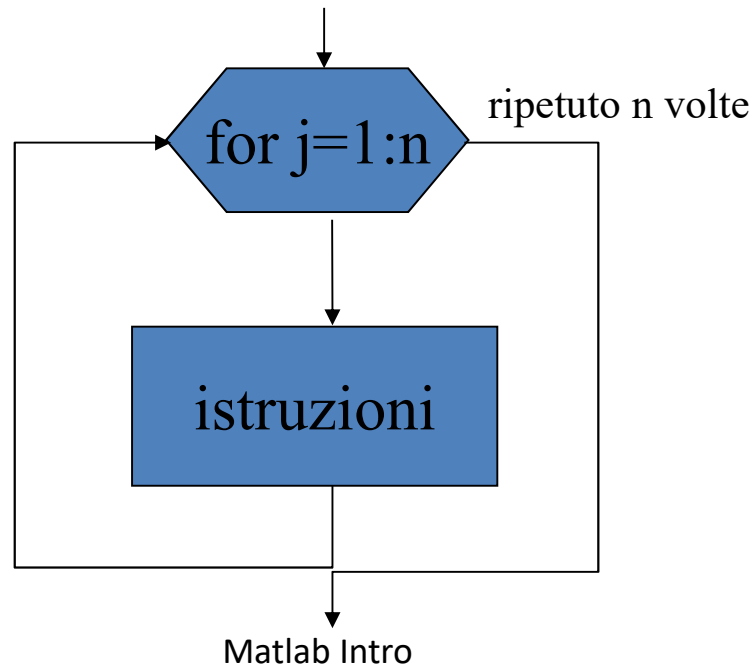
Introduzione a Matlab 4

Il ciclo FOR

- Il costrutto FOR viene utilizzato per ripetere un numero prefissato di volte una o più istruzioni
- Per questo motivo viene anche chiamato **ciclo** FOR
- Tuttavia, se necessario, il ciclo può essere interrotto prima che sia stato ripetuto il numero di volte stabilito

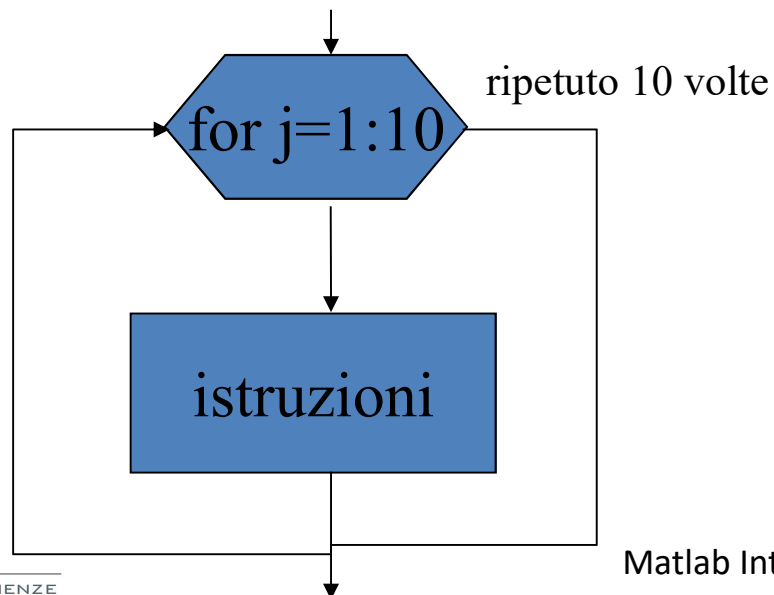
Il ciclo FOR

- La sintassi per realizzare un ciclo FOR prevede l'utilizzo della parola riservata **for** seguita da una variabile, chiamata indice del for, contenente i valori compresi in un intervallo ed infine la parola riservata **end** che chiude il ciclo
- Lo schema seguente illustra il funzionamento del costrutto for



Il ciclo FOR

- Ad esempio se volessimo ripetere 10 volte un certo numero di istruzioni, potremmo utilizzare il seguente ciclo in cui la variabile j assume, ad ogni ripetizione (detta iterazione), i valori da 1 a 10.
- Prima iterazione: $j=1$
- Seconda iterazione: $j=2$
- e così via fino a $j=10$ rappresenta l'ultima esecuzione del ciclo



```
for j=1:10
    %istruzioni;
end
```

Il ciclo FOR

- Per capire il funzionamento del costrutto for, proviamo a calcolare la somma dei primi 10 numeri interi
- Creiamo una variabile s che conterrà la suddetta somma ed eseguiamo il seguente codice

```
>> s=0;  
>> for i=1:10  
s=s+i;  
end  
>> s  
s =  
    55
```

- Prima iterazione: $i=1$, $s=0$ per cui l'istruzione eseguita è $s=0+1=1$
- Seconda iterazione: $i=2$ $s=1$ e l'istruzione eseguita è $s=1+2=3$
- e così via fino alla decima ed ultima iterazione

Il ciclo FOR

- Il ciclo for viene spesso utilizzato per accedere in maniera programmatica agli elementi di un array
- Creiamo un array a di 10 elementi contenente i numeri interi da 10 a 20

```
>> a=10:20;
```

- Per accedere al primo valore di a, usiamo la seguente sintassi

```
>> a(1)  
ans =  
    10
```

- Per accedere al quinto valore di a, usiamo la seguente sintassi

```
>> a(5)  
ans =  
    14
```

- ecc.

Il ciclo FOR

- Immaginiamo ora di voler sommare tutti i valori contenuti nell'array a
- Per eseguire questa somma dobbiamo accedere ai valori contenuti nelle 10 posizioni
- Riprendiamo l'esempio della somma dei primi 10 numeri interi
- Per ogni $i=1\dots 10$ (la dimensione dell'array è sempre 10 a prescindere dai valori in esso contenuti), dobbiamo eseguire la somma di s con il valore contenuto nella posizione i -esima
- $s=s+a(i)$

Il ciclo FOR

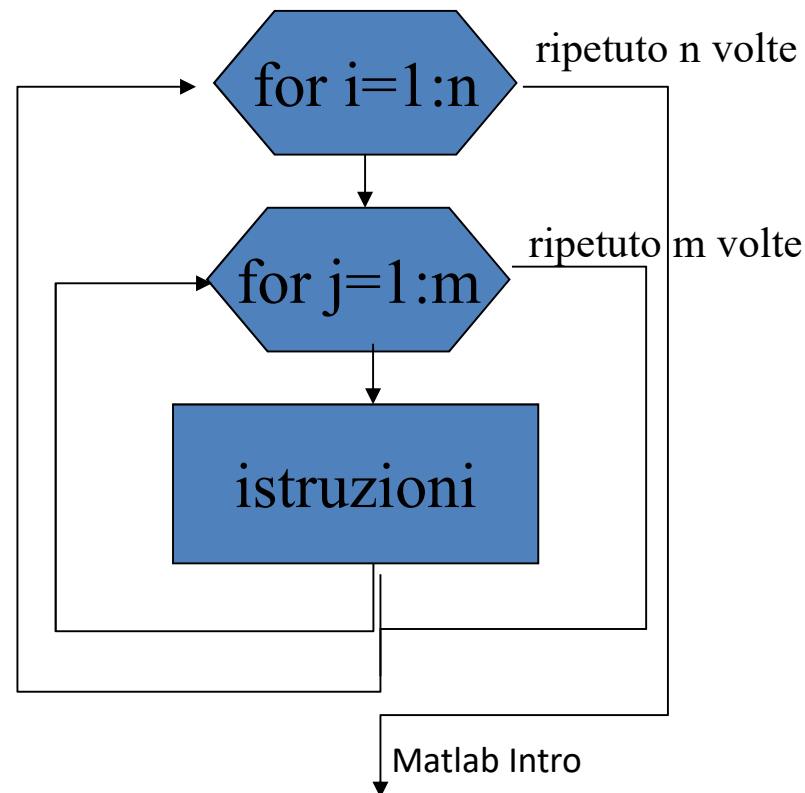
- Prima iterazione: $i=1$ $a(i)=a(1)=10$ $s=0$ per cui l'istruzione eseguita è $s=0+10=10$
- Seconda iterazione: $i=2$ $a(i)=a(2)=11$ $s=10$ e l'istruzione eseguita è $s=10+11=21$
- e così via fino alla decima ed ultima iterazione

```
>> s=0;
>> a=10:20;
>> for i=1:10
s=s+a(i);
end
>> s
s =
    145
```

- In questo esempio abbiamo utilizzato l'indice del ciclo for come indice di posizione dell'array a

I cicli FOR innestati

- La sintassi per realizzare un ciclo FOR innestato prevede l'utilizzo del **for** seguito da un indice, seguito da un secondo **for**, seguito da un altro indice ed infine due **end** che chiudono il primo ed il secondo ciclo
- Lo schema seguente illustra il funzionamento del costrutto **for**



I cicli FOR innestati

- Un esempio di ciclo innestato è riportato di seguito

```
for i=1:10
    for j=1:10
        %istruzioni;
    end
end
```

- Notiamo che in questo caso, per ogni iterazione del primo ciclo vengono eseguite 10 iterazioni del secondo ciclo, per cui il blocco di istruzioni all'interno del secondo ciclo viene eseguito $10*10=100$ volte.
- I cicli innestati rivestono un'importanza fondamentale in Matlab perché consentono di generare gli indici per accedere agli elementi di una matrice ed operare in maniera efficiente con le variabili di questo tipo.

I cicli FOR innestati

- Riprendiamo l'esempio precedente e sommiamo tutti gli elementi di una matrice
- Utilizziamo gli indici dei due cicli (i e j) come indici di riga e di colonna della matrice a
- Per ogni ciclo esterno dobbiamo ripetere tutto il ciclo interno

j=	1	2	3
i =1	1	1	1
2	1	1	1
3	1	1	1

```
>> a=ones(3,3);  
>> s=0;  
>> for i=1:3  
for j=1:3  
s=s+a(i,j);  
end  
end  
>> s  
s =  
9
```

I cicli FOR innestati

- Un altro esempio di due cicli for innestati è quello per la creazione di una matrice triangolare inferiore
- Questo esempio è interessante poiché mostra come il numero di volte che viene eseguito il secondo ciclo non è fisso ma varia in base all'iterazione del primo ciclo
- Ad esempio alla prima iterazione del primo ciclo, il secondo ciclo viene eseguito una volta
- Alla seconda iterazione del primo ciclo, il secondo ciclo viene eseguito due volte
- e così via

```
>> n=4;  
>> for i=1:n  
    for j=1:i  
        a(i,j) = 1;  
    end  
end
```

```
>> a
```

```
a =
```

```
1  0  0  0  
1  1  0  0  
1  1  1  0  
1  1  1  1
```

Altri cicli

- Non sempre i cicli hanno indici interi
- Nell'esempio che segue la variabile x assumerà ad ogni iterazione i valori contenuti nel vettore, nell'ordine in cui compaiono

```
>> for x=[pi, 51, -72.1]
display(x)
end
x =
    3.14159265358979
x =
    51
x =
   -72.1
```