Università degli Studi di Ferrara

Corso di Laurea in Matematica - A.A. 2019 – 2020

Programmazione
Lezione 19 – Controllo di Flusso in MATLAB

Docente: Michele Ferrari - michele.ferrari@unife.it

Nelle lezioni precedenti

- Script: scrivere codice in MATLAB
- Funzioni in MATLAB
- Debug: concetti e comandi utili per risolvere problemi nel codice

In questa lezione

- Operatori relazionali e logici
- Costrutto if
- Ciclo for
- Ciclo while

Operatori relazionali

Espressione	Significato
a < b	minore
a > b	maggiore
a <= b	minore od uguale
a >= b	maggiore od uguale
a == b	uguale
a ~= b	diverso

Operatori relazionali

Vengono utilizzati quando si desidera eseguire un **confronto**

Si osservi che:

- a e b possono essere due variabili oppure una variabile ed una costante
- Se si confrontano uno scalare ed una variabile con indice (array), lo scalare si “ espande” fino alla dimensione della variabile

La risposta è una matrice dello stesso ordine della matrice che si confronta con elementi uguali a

- 1 se il confronto risulta VERO
- 0 se il confronto risulta FALSO

Notiamo come, similmente a quanto visto in C, è importante non confondere l’operazione di confronto “a è uguale a b?” che presenta la sintassi “a == b” con l’**operatore di assegnazione** “a=b” (copia in a il valore contenuto in b).

Operatori realizionali: esempio

Proviamo ad eseguire le seguenti istruzioni nella command window:

```
>> x=5;
>> x>=[2 -1 3;8 0 6]
```

Il risultato che otterremo sarà una matrice così formata:

```
ans =
      1  1  1
      0  1  0
```

Operatori logici

Espressione	Significato
&	AND logico
|	OR logico
~	NOT logico

Operatori logici: esempi

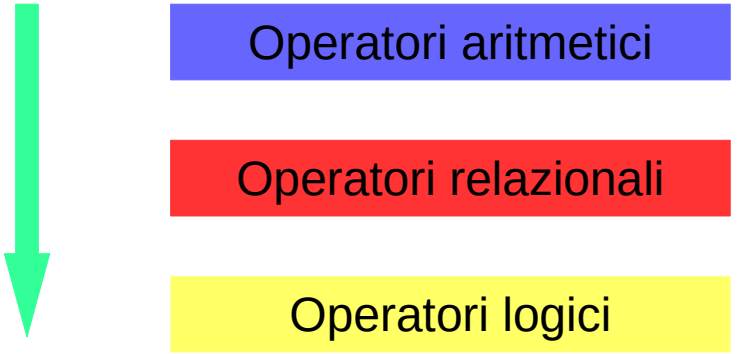
```
>> (3>4) & 1
ans =
   logical
   0

>> (3<4) & 1
ans =
   logical
   1
```

Precedenze tra operatori

Abbiamo dunque visto 3 tipologie di operatori:

- Operatori aritmetici
- Operatori relazionali (di confronto)
- Operatori logici

Nelle istruzioni in cui tali operatori sono presenti,
vengono rispettate le seguenti precedenze:



La programmazione strutturata in Matlab

MATLAB può essere considerato un linguaggio di programmazione alla stregua del Fortran o del C in quanto permette l'utilizzo delle strutture sintattiche tradizionali e l'uso appropriato di funzioni per codificare in modo semplice e flessibile gli algoritmi classici del Calcolo Numerico.

Ricordiamo che MATLAB è un linguaggio interpretato, quindi non utilizza compilatori:

- dal punto di vista dell'efficienza, della velocità e della portabilità può risultare meno competitivo rispetto al C.

La programmazione strutturata in Matlab

Ricordiamo:

La **sequenza**: specifica l'ordine in cui le istruzioni si susseguono

La **selezione**: permette di scegliere fra due alternative la sequenza di esecuzione

L'**iterazione**: permette di ripetere più volte la stessa sequenza fino al verificarsi di una condizione

Le strutture di controllo devono essere pensate come schemi di composizione:

- Una sequenza può contenere iterazioni e selezioni
- Una iterazione può contenere a sua volta sequenze e selezioni
- Una selezione può contenere a sua volta sequenze e iterazioni

Il costrutto if

if

se si verifica la proposizione indicata subito dopo, si eseguono tutte le istruzioni fino a trovare l'istruzione **di end o l'istruzione else**

else

si eseguono le istruzioni fino al primo end successivo se la proposizione indicata nell'if **non** risulta verificata.

Il costrutto if

La struttura descritta si sintetizza nel seguente schema :

```
if <una o più espressioni di confronto>
          ↑
          istruzioni
          ↓
else
          ↑
          istruzioni
          ↓
end
```

Il costrutto if: esempio

Si vuole calcolare il valore assoluto del numero reale x .

Questo si realizza con la sequenza:

```
if x >= 0
  assoluto=x;
else
  assoluto=-x;
end
assoluto
```

Il costrutto for

In MATLAB la sintassi del costrutto **for** è la seguente:

```
for k=n1:n3:n2
```

Dove

- n1: primo valore assunto dalla variabile k
- n2: massimo valore che può assumere k
- n3: incremento che ha k ad ogni esecuzione del ciclo (facoltativo)

La variabile k assume così i valori: n1, n1+n3, n1+2*n3, […] finché non viene superato il valore n2 .

Se il valore n3, se omesso, si assume uguale a 1.

Il costrutto for: esempio

Supponiamo di voler calcolare il valore di:

$$\sum_{k=1}^{n} a_k \, , n=10, a_k=k \, , k=1,2,\ldots,10.$$

In MATLAB possiamo implementare il calcolo richiesto con la seguente sequenza di istruzioni:

```
clear;clc;
n=10;
a=[1 2 3 4 5 6 7 8 9 10];
somma=0;
for k=1:n
   somma=somma+a(k);
end
somma
```

Cicli annidati: esempio

Supponiamo di voler costruire la matrice di Hilbert H , di ordine 10.

Considerato che i suoi elementi sono

$$h_{ij} = \frac{1}{i+j-1}$$

per i, j = 1, 2, . . . , 10

questo si realizza con il seguente codice:

```
clear; clc;
n=10;
for i=1:n
   for j=1:n
       H(i,j)=1/(i+j-1);
   end
end
H
```

Il costrutto while

- Ricordiamo: l'istruzione **while** si utilizza se si prevede di eseguire un insieme di istruzioni finché risultano verificate una o più condizioni.
- Supponiamo di voler ripetere un certo blocco di istruzioni finché risultano verificate contemporaneamente le proposizioni prop1 e prop2.

Per realizzare questo algoritmo si utilizza la seguente sequenza:

```
while prop1 & prop2
    % blocco di istruzioni
end
```

Il costrutto while: esempio

Si vuole determinare la precisione di macchina (eps)

```
clear;clc;
precma=1;
while precma+1>1
      precma=precma/2;
end
precma=precma*2
```

L’ultima istruzione riporta il valore della variabile ad eguagliare eps in quanto l’ultima divisione è già stata eseguita.

L’istruzione break

- Un ciclo FOR o WHILE può essere interrotto in qualunque momento inserendo l’istruzione break
- Quando tale comando è eseguito, MATLAB salta direttamente all’istruzione `end` , con cui termina il ciclo

  - ➔ Non è possibile inserirsi all’interno di un ciclo senza passare necessariamente da una istruzione di for o di while

Esercizio 1

Implementare in uno script l’algoritmo precedentemente visto per il calcolo del valore assoluto di un numero con inserimento del numero da parte dell’utente

Esercizio 1b

Trasformare il precedente script in una function, testare la function richiamandola da command window

Esercizio 2

Implementare in uno script l'algoritmo per il calcolo di:

$$\sum_{k=1}^{n} a_k, n=10, a_k=k, k=1,2,\ldots,10.$$

Esercizio 2b

Trasformare il precedente algoritmo in una function che realizzi la sommatoria generica di qualunque vettore le venga passato in ingresso

Realizzare uno script che inizializzi il vettore a piacere e richiami la funzione, visualizzi poi il risultato

Si confronti il proprio risultato con il risultato della funzione predefinita di MATLAB sum()

Esercizio 3

Realizzare uno script con che trovi il massimo di un vettore:

- Inizializzare un vettore di n elementi (a piacere) con numeri interi
- Implementare l’algoritmo di ricerca del massimo

Soluzione esercizio 3

```
clear;clc;
v=[2 -3 5 7]; n=length(v);
massimo=v(1);
for j=2:n
    if v(j)>massimo
        massimo=v(j);
    end
end
massimo
```

Grazie per l’attenzione

Riferimenti

Il corso di programmazione per il primo anno della Laurea Triennale in Matematica nasce con l'intento di unire ai principi di programmazione una conoscenza basilare di uno degli strumenti software più diffusi nell'ambito matematico: Matlab.

Per la parte introduttiva di MATLAB:

L. Pareschi, G. Dimarco “Introduzione a MATLAB”, corso di Laboratorio di Calcolo Numerico 2006