

INSTRUCȚIUNI ȘI FUNCȚII DE CONTROL ÎN MATLAB



1. Obiective

Înșușirea modului de utilizare a mediului de programare Matlab pentru realizarea aplicațiilor care necesită execuția condiționată sau repetarea unui grup de instrucțiuni în funcție de valoarea de adevăr a unei expresii logice.



2. Noțiuni teoretice

2.1. Instrucțiuni de control logic

Programele Matlab realizează pași secvențiali, iar operațiile sunt executate succesiv. Uneori, este nevoie să se realizeze o selecție a unui grup de instrucțiuni în funcție de valoarea de adevăr a unei expresii. În acest caz, instrucțiunile utilizează operatori relaționali și operatori logici.

2.1.1. Operatori relaționali

Matlab dispune de 6 operatori relaționali, utilizați pentru a compara două matrice de dimensiuni egale.

Operatorii relaționali sunt:

Simbol Matlab	Semnificația
<	mai mic
>	mai mare
<=	mai mic sau egal
>=	mai mare sau egal
= =	identic egal
~ =	diferit

Operatorii relaționali compară două matrice, element cu element și returnează o matrice de aceeași dimensiune cu cea a matricelor care se compară, având elemente de 1 (unu) dacă relația este adevărată și elemente de 0 (zero) dacă relația este falsă.

Primii 4 operatori compară numai partea reală a operanzilor (partea imaginară este ignorată), iar ultimii 2 operatori tratează atât partea reală cât și partea imaginară.

Forma generală de utilizare a operatorilor relaționali este:

rezultat = expresie_1 operator_relațional expresie_2,

prin care se compară *expresie_1* cu *expresie_2*, prin *operator_relațional*, iar rezultatul este depus în *rezultat*.

Observație: Dacă unul dintre operanzi este un scalar iar celălalt o matrice, scalarul se transformă la dimensiunea matricei.

2.1.2. Operatorii logici

Operatorii logici din Matlab sunt:

Simbol Matlab	Prioritatea	Semnificația
~	1	Nu
&	2	Și
	3	Sau

Observații:

- Operatorii logici returnează 1 (Adevărat) pentru orice valoare diferită de zero.
- În Matlab, orice valoare diferită de zero este considerată adevărată logic, adică 1.
- Operatorii logici au prioritate mai mică decât operatorii relaționali și aritmetici.

Instrucțiunile de control logic în Matlab sunt:

<i>if</i>	Instrucțiune pentru execuția condiționată
<i>else</i>	Instrucțiune asociată cu "if"
<i>elseif</i>	Instrucțiune asociată cu "if"
<i>for</i>	Instrucțiune pentru crearea ciclurilor cu număr specificat de pași
<i>while</i>	Instrucțiune pentru crearea ciclurilor cu condiție logică
<i>break</i>	Instrucțiune pentru terminarea forțată într-un ciclu
<i>return</i>	Instrucțiune care determină execuția la funcția precizată
<i>error</i>	Instrucțiune pentru afișarea unui mesaj de eroare
<i>end</i>	Instrucțiune pentru încheierea ciclurilor "for", "while", "if"

Detalierea acestor instrucțiuni este prezentată în continuare.

1. Instrucțiunea *if* simplă

Instrucțiunea *if* poate fi simplă sau poate include clauzele *else* sau *elseif*.

Forma generală a unei instrucțiuni *if* simplă este:

```
if  expresie logică
    grup de instrucțiuni
end
```

Dacă valoarea logică a expresie logică este adevărată, se execută grupul de instrucțiuni dintre instrucțiunea *if* și instrucțiunea *end*.

2. Instrucțiunea *if* cu clauza *else*

Forma generală a unei instrucțiuni *if* cu clauza *else* este:

```
if  expresie logică
    grup de instrucțiuni A
else
    grup de instrucțiuni B
end
```

Se folosește pentru execuția *grup de instrucțiuni A*, dacă expresie logică este adevărată sau execuția *grup de instrucțiuni B*, dacă expresie logică este falsă.

3. Instrucțiunea *if* cu clauza *elseif*

Forma generală a unei instrucțiuni *if* cu clauza *elseif* este:

```
if  expresie logică 1
    grup de instrucțiuni A
elseif
    expresie logică 2
    grup de instrucțiuni B
end
```

Dacă este nevoie de o instrucțiune *if* cu mai multe condiții logice de testat, atunci se folosește instrucțiunea *if* cu clauza *elseif*.

4. Instrucțiunea repetitivă *for*

Această instrucțiune se folosește pentru repetarea unui grup de instrucțiuni de un anumit număr de ori, cunoscut inițial. Numărul de repetări ale instrucțiunilor din ciclu este evidențiat printr-un contor sau index.

Forma generală a instrucțiunii este:

```
for    index = expresie
        grup de instrucțiuni
end
```

unde:

- *index* este numele contorului
- *expresie* este o matrice, un vector sau un scalar

În aplicații,

index = expresie este de forma: $k = \text{inițial}:\text{pas}:\text{final}$

unde:

k este contor sau index

inițial este prima valoare a lui *k*

pas este incrementul contorului (dacă lipsește este implicit 1)

final este cea mai mare valoare a lui *k*.

La fiecare pas al ciclului *for* indexul are valoarea uneia dintre valorile expresiei. Dacă *expresie* este o matrice, ciclarea se face pe coloane.

Regulile impuse pentru folosirea unui ciclu *for* sunt:

- *index* trebuie să fie o variabilă;
- dacă *expresie* este o matrice goală, bucla nu se execută și se trece la următoarea instrucțiune, după instrucțiunea *end*;
- dacă *expresie* este un scalar, bucla se execută o singură dată, iar indexul ia valoarea scalarului;
- dacă *expresie* este un vector, bucla se execută de atâtea ori câte elemente are vectorul, de fiecare dată indexul având valoarea egală cu cea a elementelor vectorului;
- dacă *expresie* este o matrice, bucla se execută de atâtea ori câte coloane are matricea, iar indexul va avea la fiecare iterație valorile din coloanele matricei;
- la terminarea ciclului, indexul are ultima valoare utilizată (*final*);
- dacă se utilizează forma:


```
for k = inițial:pas:final
        grup de instrucțiuni
end
```

ciclul se execută de n ori, unde $n = \left\lceil \frac{\text{final} - \text{initial}}{\text{pas}} \right\rceil + 1$ (parantezele drepte semnifică parte întreagă). Dacă n este negativ ciclul *for* nu se execută.

5. Instrucțiunea repetitivă *while*

Această instrucțiune se utilizează pentru repetarea unui grup de instrucțiuni în funcție de valoarea adevărată a unei expresii. Numărul de iterații nu este cunoscut inițial.

Forma generală este:

```
while    expresie
        grup de instrucțiuni
end
```

Grup de instrucțiuni se execută cât timp *expresie* are toate elementele nenule.

De cele mai multe ori, *expresie* este de forma:
expresie_1 operator relațional *expresie_2*

Dacă condiția logică din *expresie* este adevărată atunci se execută grup de instrucțiuni, apoi condiția logică este verificată din nou. Dacă este adevărată, atunci se execută grup de instrucțiuni, dacă este falsă se execută următoarea instrucțiune după instrucțiunea *end*.

Dacă *expresie* este tot timpul adevărată, atunci bucla *while* se execută de un număr infinit de ori. Ieșirea dintr-o astfel de buclă se face prin comanda [Ctrl]+[c].

6. Instrucțiunea *break*

Această instrucțiune se utilizează pentru ieșirea dintr-o buclă înainte de terminarea acesteia. Se recomandă instrucțiunea *break* dacă este detectată o condiție de eroare în interiorul unei bucle. Rezultatul acestei instrucțiuni este oprirea execuției ciclurilor *for* și *while*. În cazul unor cicluri incluse, instrucțiunea *break* comandă ieșirea din ciclul cel mai interior.

7. Instrucțiunea *return*

Instrucțiunea comandă ieșirea din fișierul Matlab (*.m) către funcția care l-a apelat sau către tastatură.

8. Instrucțiunea *error* permite afișarea unui mesaj text la întâlnirea unei erori. Sintaxa acestei instrucțiuni este: *error('mesaj')*.

După afișarea mesajului controlul este redat tastaturii.

2.2. Funcții de control logic

Funcțiile de control logic în Matlab sunt:

1. *exist* Verifică dacă variabilele sau funcțiile argument sunt definite;
2. *any* Testează dacă cel puțin un element al unei matrice verifică o condiție logică;
3. *all* Testează dacă toate elementele unei matrice verifică o condiție logică;
4. *find* Returnează indicii elementelor diferite de 0 (zero);
5. *isnan* Testează dacă elementele unei matrice sunt NaN;
6. *isinf* Testează dacă elementele unei matrice sunt infinite;
7. *finite* Testează dacă elementele unei matrice sunt finite

2.3. Funcții de control general

Câteva dintre funcțiile de control general în Matlab sunt:

1. *Doc* Documentație on-line, pe display, help-ul din Matlab
2. *Demo* Acces la producțiile demo prin help-ul din Matlab
3. *Help* În fereastra de comandă prezentarea help-ului pentru funcțiile Matlab
4. *web* Deschidere site web sau fișiere web sau Help
5. *Exit* Închiderea MATLAB (la fel ca și comanda *quit*)
6. *Clc* Ștergere fereastra comenzi
7. *Clear* Ștergere valori variabile din spațiul de lucru
8. *Home* Mutarea cursorului în partea superioară, colțul din stânga, al ferestrei de comenzi.



3. Probleme de rezolvat

3.1. Să se calculeze în Matlab valorile funcției *modul*, pe intervalul $[-4,4]$ (cu pasul 0.5), prin relația cunoscută:

$$f(t) = \begin{cases} -t & \text{daca } -4 \leq t < 0 \\ t & \text{daca } 0 \leq t \leq 4 \end{cases}.$$

3.2. Să se determine în Matlab valoarea funcției f , unde $f = \frac{x^3 - 2x^2 + x - 6.3}{x^2 + 0.5x - 1}$, pentru $x=4$.

3.3. Să se calculeze valorile funcției f pentru toate valorile întregi pentru care este definită prin relația:

$$f(x) = \begin{cases} 2x+3 & \text{daca } x \in [-10,2] \\ 2x^2-1 & \text{daca } x \in (2,20] \end{cases}$$

3.4. Să se calculeze suma elementelor vectorului x , până când întâlnește un număr mai mare ca 8.

Vectorul $x = [5, 2, -9, 10, -1, 9, -1]$.

3.5. Să se calculeze media aritmetică a primelor 10 numere naturale.

3.6. Să se genereze o matrice A , cu n linii și n coloane, ale cărei elemente sunt definite de relația:

$$A = \begin{cases} 2 & \text{daca } i=j \\ -1 & \text{daca } |i-j|=1, \text{ pentru } n=5. \\ 0 & \text{in rest} \end{cases}$$



4. Probleme rezolvate

4.1. O variantă de program în Matlab pentru §3.1 este următoarea:

```
t=-4:0.5:4;
k=max(size(t));
for i=1:k
if (t(i)>=-4)&(t(i)<0)
f(i)=abs(t(i));
end
if (t(i)>=0)&(t(i)<=4)
f(i)=t(i);
end
end
disp('Funcția modul')
f
```

Rezultatul obținut în Matlab va fi:

Funcția modul

f = Columns 1 through 7

4 3.5 3 2.5 2 1.5 1

Columns 8 through 14

0.5 0 0.5 1 1.5 2 2.5

Columns 15 through 17

3 3.5 4



5. Probleme propuse

5.1 Să se calculeze suma elementelor vectorului x , până când întâlnește un număr mai mare ca 10.

Vectorul $x = [8, 3, -9, -10, 11, 9, -1]$.

5.2 Să se calculeze media aritmetică a primelor 15 numere naturale.

5.3 Să se genereze o matrice A , cu n linii și $n+1$ coloane, ale cărei elemente sunt definite de relația:

$$A = \begin{cases} -1 & \text{daca } i=j \\ 2 & \text{daca } |i-j|=1, \text{ pentru } n=6. \\ 0 & \text{in rest} \end{cases}$$

5.4 Să se calculeze în Matlab valorile funcției y definită de relația:

$$y(t) = \begin{cases} \sin(t) & \text{daca } \{a \leq t < b\} \cup \{c \leq t \leq d\} \\ \cos(5t) & \text{daca } b \leq t < c \end{cases},$$

pentru $a = -10; b = -4; c = 4; d = 10$.