

Inițiere în limbajul MATLAB (II)

ing. Ioan Tăbuș

Institutul Politehnic București

2. Descrierea elementelor de bază ale limbajului

În cele ce urmează, vor fi introduse, într-o modalitate sistematică, noțiunile care constituie nucleul operațional matematic al limbajului MATLAB.

2.1 Instrucțiuni și variabile

MATLAB este un limbaj bazat pe expresii. Instrucțiunile sînt de cele mai multe ori de forma unor simple atribuiri

variabila = expresie (1)

sau numai

expresie (2)

fiind interpretate, prin evaluarea expresiei (și eventual tipărirea rezultatului, dacă expresia nu este urmată de caracterul; care anulează tipărirea) imediat ce instrucțiunea a fost introdusă de operator, atribuindu-se valoarea rezultatului variabilei specificate în stînga semnului = al instrucțiunii (pentru forma (2), lipsind numele unei variabile, rezultatul este atribuit variabilei ans (răspuns = answer)).

Alte tipuri de instrucțiuni, cum ar fi cele de control, vor fi descrise în altă secțiune.

Instrucțiunea poate fi scrisă pe o singură linie terminată cu CR sau; și CR. În cazul instrucțiunilor lungi sau care este de preferat să fie scrise într-o formă grafică mai sugestivă, instrucțiunea poate continua pe mai multe linii, marcarea continuării liniei curente fiind semnalizată de către operator prin două sau mai multe puncte.

Expresiile sînt formate din nume de variabile, funcții, operatori și caractere speciale.

Exemple:

```
>> (1/2)^2
ans =
    0.2500
>> pi = 4*atan(1)
pi =
    3.1416
>> format long
>> pi
ans =
    3.14159265358979
```

Formatul de afișare al numerelor poate fi trecut de la forma scurtă (5 digiți scalați în virgulă fixă) la forma lungă (15 digiți scalați în virgulă fixă) prin comanda `format long` sau la forma lungă cu exponent prin `format long e`, revenirea la forma scurtă făcîndu-se prin `format short`.

De notat că `pi` este o variabilă permanentă, inițializată automat înainte de începerea sesiunii de lucru, așa încît nu este necesară secvența de mai sus pentru calcularea ei. Tot variabile permanente setate de sistem la inițializare sînt și:

- `eps` - o toleranță utilizată pentru discriminarea rangului sau singularității unei matrice și inițializată de sistem cu distanța dintre 1 și următorul număr mai mare, reprezentabil în virgulă mobilă, avînd în aritmetica IEEE valoarea $\text{eps}=2^{-52}$;
- `Inf` este un simbol special (cu semnificația infinit) fiind asociat valorilor rezultate în urma operațiilor de forma $1/0$;
- `NaN` este variabila (Not a Number) asociată cu operații de tipul $0/0$ sau Inf/Inf .

Variabilele introduse de către utilizator trebuie să aibă numele format dintr-o literă urmată de orice număr de litere, cifre sau sublinieri (dar numai primele 19 caractere sînt memorate). În mod normal literele mari sînt deosebite de cele mici, dar prin comanda `casesen` se poate trece la ignorarea diferenței dintre cele două seturi de litere.

Pentru a lucra cu numere complexe în MATLAB, este mai întîi necesară introducerea unității imaginare, de exemplu prin:

```
>> i=sqrt(-1)
```

după care numerele complexe sau matricele cu elemente complexe pot fi introduse în una din următoarele forme:

```
>> z = 10 + 5*i;
```

```
>> x = [10+5*i 20-3*i];
```

```
>> x = [10 20] + [5*i -3*i].
```

În MATLAB nu există secțiune de asignare de tipuri sau de declarare a variabilelor, pentru că, implicit, toate variabilele sînt considerate a fi matrice bidimensionale (variabilele scalare fiind tratate ca matrice de dimensiuni 1×1 , vectorii coloană fiind considerați matrice $n \times 1$, iar vectorii linie matrice $1 \times n$), informația memorată pentru fiecare variabilă (dimensiuni, tipul elementelor, real sau complex) este afișată prin comanda `whos`.

2.2 Specificarea matricelor

Există mai multe modalități de a atribui valori variabilelor sistemului, deci pentru construirea de matrice.

a) Atribuire prin listă explicită de elemente:

```
>> A = [1 2 3; 4 5 6; 7 8 9];
```

sau

```
>> A = [1 2 3
```

```
    4 5 6
```

```
    7 8 9];
```

Lista trebuie să fie cuprinsă între simbolurile `[și]`, spațiul sau virgula separînd elementele a două coloane diferite, iar caracterul; sau CR separînd între ele liniile

matricelor. Cele două modalități de introducere a variabilei A de mai sus vor produce același rezultat.

b) Utilizarea unor funcții de generare a unor matrice particulare:

- funcția `eye` creează o matrice unitate de dimensiune specificată în argumentul funcției, ca în

```
>> I3 = eye(3);
```

În care se creează o matrice unitate de dimensiune 3. Același rezultat se poate obține prin

```
>> I3 = eye(A);
```

unde A este o matrice pătrată de dimensiune 3.

- funcțiile `zeros` și `ones` creează matrice cu toate elementele 0 sau 1, iar `rand` creează matrice cu elementele aleatoare uniform distribuite. Pentru specificarea dimensiunii matricelor, la matricele pătrate se specifică un singur argument (de exemplu `zeros(3)`), iar pentru cele dreptunghiulare se specifică două argumente, numărul de linii și numărul de coloane (de exemplu `ones(2,4)`) sau mai simplu, funcția va prelua dimensiunile matricei dată ca argument (de exemplu `rand(A)`).

c) Atribuire prin listă de submatrici:

```
>> C = [A, eye(A); zeros(A) A*A].
```

Rolul caracterelor spațiu , ; CR este același ca și în listele de elemente.

d) Încărcarea din M-file:

Fișierele aflate în directorul curent, având extensia `.m` (M-files), pot avea două roluri: fișiere script și fișiere funcții. În situația fișierelor script, conținutul lor (în format ASCII) trebuie să aibă aceeași sintaxă ca și liniile introduse în mod obișnuit de către utilizator de la tastatură. În momentul în care utilizatorul invocă numele acestui fișier (fără extensie), în linia de comandă MATLAB (de la tastatură sau din interiorul altui fișier script), liniile fișierului script vor fi interpretate una câte una, ca și cum ar fi fost introduse de la tastatură.

e) Încărcarea din fișiere externe de date:

Comenzile `save` și `load` pot fi utilizate pentru memorarea în fișiere a informației într-un format mult mai economic decât forma ASCII. Prin:

```
>> save nume_fișier lista_nume_variabile
```

se pot memora în fișierul cu numele specificat (având implicit extensia `.mat`) variabilele date în lista de variabile. Comanda:

```
>> load nume_fișier
```

citește din fișierul specificat informația prin care restaurează variabilele specificate anterior în comanda `save`.