

Dezvoltarea de programe MATLAB

Fișiere .M

În primul laborator a fost prezentat modul de creare al unui nou fișier sau de modificare al unui deja existent, în editorul MATLAB. Programele MATLAB trebuie salvate în fișiere cu extensia '.m' pentru a putea fi recunoscute ca noi comenzi. Mai mult, fișierele realizate trebuie salvate, fie în directorul de lucru curent, fie într-un director inclus în calea de căutare. Când sunt respectate aceste condiții, fișierele apelate prin introducerea numelor lor (și a argumentelor necesare, în cazul funcțiilor) vor fi rulate într-un mod similar cu cel al comenzilor MATLAB standard.

Rutine și funcții

Rutinele sunt fișiere ce conțin o serie de comenzi MATLAB la fel cum acestea ar fi introduse direct în linia de comandă. Variabilele utilizate sunt conținute de spațiul de lucru.

Funcțiile posedă propriul lor spațiu de lucru, cu variabile locale, și au argumente de intrare și ieșire. Pe

lângă funcțiile proprii dezvoltate, pot fi editate și funcțiile MATLAB/SIMULINK livrate standard, care sunt dezvoltate în MATLAB (nu și cele implementate de exemplu în C) prin comanda **edit** i.e. **edit factorial**. În operarea cu funcții trebuie avută în vedere următoarea sintaxă:

```
function [argumenteiesire] = numefunctie(argumenteintrare)
% Tot ce este introdus in aceasta sectiune
% apare la apelarea ajutorului functiei prin comanda 'help numefunctie'.

Lista comenzilor specifice functiei
```

Numele funcției trebuie să fie identic cu cel al fișierului 'numefunctie.m'. Primul rând nu trebuie să fie închis printr-un operator punct și virgulă. Cel de-al doilea rând și rândurile care îl urmează și încep cu un %, vor fi afișate la apelarea ajutorului (*help*) acestei funcții.

Pentru ilustrare, urmează un exemplu simplu. Introduceți următoarele comenzi într-un fișier cu numele 'sisteclin.m':

```
function [x, detA] = sisteclin(A,b)
% SISTECLIN rezolva sistemul de ecuatii liniare A*x=b si
% returneaza vectorul solutie x, precum si determinantul matricei A.

detA=det(A);
x=A\b;
```

Rulați suplimentar următoarele comenzi în linia de comandă MATLAB:

```
>> help sisteclin
>> matA=[1,2,3;4,5,6;7,8,0];
>> vec=[2;5;9]
>> sisteclin(matA,vec)
>> [r1,r2] = sisteclin(matA, vec)
```

În cazul în care se dorește scrierea unei funcții ce poate fi apelată cu un număr diferit de argumente, pot fi utilizate funcțiile MATLAB **nargin** și **nargout**. Acestea returnează numărul de argumente de intrare și de ieșire.

Dacă se dorește ca variabilele utilizate de către o funcție, de exemplu t și x , să fie puse la dispoziția și a altor funcții sau să fie vizibile în spațiul de lucru, acestea trebuie să fie declarate ca **global** în fiecare situație în care sunt utilizate (cum ar fi în funcția alternativă care le utilizează, dar și în spațiul de lucru).

» global t, x

Aceasta însă nu este, din punctul de vedere al programării, o soluție elegantă. Cea mai bună practică este preluarea variabilei ca argument de ieșire.

La fel ca în alte limbaje de programare, MATLAB oferă structuri de control al execuției, cum sunt:

- **if ... elseif ... else ... end**
- **switch ... case ... otherwise ... end**
- **for ... end**
- **while ... end**
- **try ... catch ... end**

Pentru detaliile de sintaxă folosiți ajutorul fiecărei structuri în parte.

În continuare este prezentată implementarea unei funcții MATLAB care verifică dacă un număr, introdus ca argument de intrare, este prim sau nu:

```
function prime(n)
% Functia verifica daca numarul n este prim sau nu
% si returneaza un mesaj corespunzator.
% Folositi help rem si help floor
% pentru a studia descrierea si sintaxa celor doua functii.

half_n=floor(n/2);
for k=2:half_n
    if rem(n,k)==0
        'Nu este prim. Se divide cu ', k
        break
    end
end

if k>=half_n
    'Numarul ', n, ' este prim'
end

return
```

Exerciții

1. Aproximați numărul π , folosind relația de calcul de mai jos. Realizați implementarea funcției în două variante, cu structurile *for* și *while*.

$$\pi \approx 4 \sum_{k=0}^m \frac{(-1)^k}{2k+1} \quad (1)$$

2. Scrieți o funcție care:

- primește ca argumente de intrare doi vectori,

- verifică dacă este vorba într-adevăr de vectori și nu de matrici,
- în cazul în care a fost utilizată o matrice ca argument de intrare, afișează un mesaj de eroare în fereastra de comenzi și returnează o valoare nulă ca argument de ieșire,
- altfel, determină valorile maxime ale ambilor vectori și le returnează ca argument de ieșire,
- adăugați funcției un text de ajutor corespunzător.

3. Scrieți o funcție MATLAB care calculează valorile funcției $f(t)$, definită pe intervale, după cum urmează:

$$f(t) = \left\{ \begin{array}{ll} 0 & \text{dacă } t < 0 \\ t^2 \cdot (3 - 2 \cdot t) & \text{dacă } 0 \leq t \leq 1 \\ 1 & \text{dacă } t > 1 \end{array} \right. \quad (2)$$

Reprezentați grafic funcția pe intervalul $[-1, 3]$.

Referințe

[1] B. Hahn, D. Valentine, *Essential MATLAB for Engineers and Scientists*, Third Edition, Elsevier, 2007.