

# Dezvoltarea de programe MATLAB

## 1 Operatori și funcții

### Operatori

Partea din dreapta a unei expresii MATLAB poate conține o formulă algebrică complicată. Constantele și variabilele cunoscute trebuie conectate logic prin intermediul operatorilor. În cazul operatorilor aritmetici, pe lângă cei comuni (+, -, \*, /), mai sunt folosiți și ridicarea la putere (^) și transpunerea ('). Suplimentar, există și operatorii logici: negare ~ și &, sau |, sau exclusiv xor() și operatorii relativi: egalitate ==, inegalitate ~=, <, ≤, >, ≥. Prin următoarea comandă este afișată o imagine de ansamblu a operatorilor:

```
>> help ops
```

**Atenție:** Operatorii fac referire la calcule matriciale. Pentru a utiliza operatorii pe elementele individuale ale unei matrici, trebuie plasat un punct (.) în fața operatorului!

```
>> A=[1,2,3;4,5,6;7,8,0]
>> b      % Vector linie
>> b=b'   % Vector coloană
>> b+b    % Sumă de vectori
>> b'*b   % Produs scalar
>> b*b    % Eroare datorată dimensiunilor incompatibile
>> b.*b   % Multiplicare la nivel de element
>> 2*A    % Înmulțire cu un scalar
>> A*A    % A ori A și
>> A^2    %A la pătrat, sunt identice
>> A.^2   % dar nu și în acest caz!
```

### Funcții

După cum s-a putut vedea din secțiunile anterioare, MATLAB include un număr mare de funcții din domenii variate. Apelarea unei funcții utilizează numele acesteia și un număr de parametri. Rezultatul poate fi stocat într-o variabilă. O imagine de ansamblu asupra funcțiilor matematice standard, oferă comanda **help elfun**.

```
>> help elfun
>> cos(0)
>> 4* atan(1)    % pi, atan(x) = tan-1(x)
>> bexp=exp(b)   % Funcția exponențială , aplicată fiecărui element din b
```

Doă funcții foarte utile sunt **size** și **length**. **size** returnează numărul de linii și coloane al unei matrici și permite astfel diferențierea vectorilor linie de vectorii coloană. **length** returnează lungimea unui vector și, în cazul matricilor, valoarea mai mare dintre numărul de linii și cel al coloanelor.

```
>> A=[1,2,3;4,5,6]
>> size(A)
>> length(A)
>> size(b)
>> length(b)
```

Comanda **help elmat** afișează o imagine de ansamblu asupra matricilor standard și a funcțiilor de manipulare

ale matricilor precum și asupra funcțiilor din domeniul algebrei liniare **help matfun**. Funcțiile **ones** și **zeros**, de exemplu, generează matrici numai cu elemente 1 resp. 0 și funcția **eye** generează matricea unitate.

```
>> help elmat
>> D=ones(3,4)
>> E=zeros(5)
>> F=eye(4)
>> G=eye(4,3)
>> help matfun
```

### Exerciții

1.1. Determinați numărul de linii și de coloane ale matricilor D, E, F și G.

## 2 Calcul simbolic

În Matlab există o bibliotecă numită *Symbolic Math* care permite efectuarea de calcule simbolice. Folosirea acestei biblioteci se face în felul următor:

```
>> syms x
```

Un exemplu de folosire al acestei biblioteci este:

```
>> expand ((x-1)3)
>> expand ((x-1)3 - 2 * x2)
```

### Exerciții

2.1. Să se determine rezultatul următoarelor operații:

- `expand((x+1)*(x+2)*(x+3)-11*x)`
- `expand(((x-1)3 * (y - 1))/(y - 1)3 - (x - 1)/(y - 1)2)`

2.2. Care este utilitatea instrucțiunii *pretty*?

## 3 Reprezentări grafice

MATLAB permite realizarea unei game variate de reprezentări grafice ale datelor. Cele mai importante funcții grafice sunt:

- curbe 2D: **plot**, **fplot**, **ezplot**, **subplot**, **stem**, **stairs**
- curbe 3D: **plot3**, **stem3**, **surf**, **mesh**, **contour**

### Comanda “plot”

Pentru reprezentarea grafică a unui vector **y** se folosește comanda **plot(y)**. În acest caz, elementele lui **y** ( $y_1, y_2, y_3, \dots$ ) sunt reprezentate bidimensional împreună cu indicii lor (1,2,3,...). Dacă se dorește utilizarea unei alte axe x, de exemplu timpul t, acest vector t trebuie adăgat ca parametru în apelarea funcției, **plot(t,y)**. Comanda corespunzătoare pentru reprezentarea tridimensională conține toți cei trei vectori, **plot3(t,x,y)**.

```
>> t=linspace(-pi,pi,30);
>> y1=sin(2*t);
>> y2=cos(5*t);
```

```

>> plot(y1)
>> plot(t,y1)    % Pentru comparație
>> plot(y1,t)    % Pentru comparație
>> plot3(t,y1,y2)

```

Graficele sunt reprezentate în figuri/**figures**. Acestea reprezintă ferestre individuale care, pe lângă curbe, includ și axele și diferite meniuri pentru modificarea reprezentării grafice. Dacă există o fereastră de figură deja deschisă, curba va fi reprezentată în aceasta și vechea curbă va fi ștearsă. Cu **hold** sunt păstrate vechile curbe și cele noi sunt desenate peste acestea. Dacă nu există o fereastră de figură deja deschisă sau este rulată comanda **figure** înainte de **plot**, va fi deschisă o nouă fereastră.

Majoritatea setărilor de grafic nu se regăsesc numai în meniul figurii ci pot fi modificate și prin fereastra de comenzi. Comenzi utile sunt:

```

>> grid on/off    % activează dezactivează rasterul de fundal
>> box on/off     % activează dezactivează un container de încadrare
>> axis on/off    % activează dezactivează reprezentarea axelor

```

Dacă nu este specificat parametrul on/off în cazul primelor două comenzi, este realizată o comutare între cele două stări.

Comanda **plot** poate primi un argument suplimentar, un șir de caractere, e.g. **plot(t,y1,'g')**. În acest șir de caractere sunt incluse informații despre culoare, reprezentarea liniilor și a punctelor de date. Pentru o descriere completă a opțiunilor disponibile, apelați descrierea funcțiilor prin **help plot**, **help graph2d**, **help graph3d**.

O fereastră de figură poate fi, similar cu o matrice, împărțită în mai multe “sub-figuri”, cu comanda **subplot**:

```

>> figure        % deschide o nouă fereastră de figură
>> subplot(2,1,1); plot(t,y1)    % subplot partajează fereastra în două linii și o coloană și alege primul
element (numărarea este făcută de la stânga la dreapta și de sus în jos) pentru reprezentarea primei curbe
>> subplot(2,1,2); plot(y1,t)    % subplot folosește aceeași partajare, alege însă a doua fereastră

```

Pentru inscripționare, există următoarele funcții, ce au efect asupra graficului curent:

```

>> title('Grafic sugestiv')    % inscripționează titlul graficului
>> xlabel('Timp [s]')          % inscripționează axa x
>> ylabel('Distanță [m]')      % inscripționează axa y

```

În cazul în care graficul conține mai multe curbe, legenda poate fi generată cu ajutorul comenzii **'Curba 1'**, **'Curba 2'**, **'Curba 3'**.

### Alte tipuri de grafice

Funcțiile **stem** respectiv **stem3** și **stairs** sunt adecvate pentru reprezentarea semnalelor discrete. Fiecare punct este reprezentat printr-o tijă verticală cu un cerc la capăt. **Stairs** este funcția treaptă și unește punctele individuale precum o scară, valoarea punctul anterior fiind păstrată până la următorul.

**fplot** calculează într-un interval dat valorile și evoluția unei funcții, fără a cunoaște numărul exact de puncte.

```

>> stem(y1)
>> stairs(t,y1)
>> fplot('3*sin(r).*exp(-r/(pi/4))',[0,2*pi]);

```

### Reprezentarea suprafețelor

Pentru reprezentarea suprafețelor sunt folosite funcțiile *mesh* și *surf*:

- **mesh** desenează o rețea colorată, ale cărei noduri sunt punctele funcției reprezentate.
- **surf** colorează suplimentar și textura plasei

Paleta de culori este stabilită prin comanda **colormap** și poate fi distinsă cu ajutorul **colorbar**.

Prin intermediul unui exemplu este clarificat cum pot fi generați vectorii necesari. Dorim să reprezentăm suprafața funcției  $z = -2x + 3y$ , unde  $x$  și  $y$  iau următoarele valori:

```
x=[-1,0,1,2]
```

```
y=[0,0.1,0.2,0.3,0.4,0.5]
```

Deoarece  $x$  conține 4 valori iar  $y$  6 valori,  $z$  va lua  $6 \times 4 = 24$  de valori. Suplimentar, deoarece fiecare punct este reprezentat printr-o tripletă  $(x_i, y_i, z_i)$ , trebuie generate matricile **X** și **Y** care conțin în fiecare linie, respectiv fiecare coloană, valorile lui  $x$  și  $y$ . Aceasta se realizează cel mai simplu prin comanda **meshgrid**. Ulterior este calculat  $z$  și este desenată suprafața:

```
>> x=[-1:2]; y=[0:0.1:0.5]    % Definirea vectorilor x și y
>> [X,Y]=meshgrid(x,y);      % Generarea matricilor X și Y
>> Z=-2.*X+3.*Y;            % Calculul matricii Z
>> mesh(X,Y,Z)              % sau surf(X,Y,Z)
```

Alte funcții relevante pentru desenarea suprafețelor sunt: **compass**, **contour**, **contour3**, **surfc**, **waterfall**, **pcolor**, **view**.

### Exerciții

*Observație: Se vor păstra graficele cerute (generate) în continuare în ferestre separate.*

3.1. Generați matricea **graphA** =  $\begin{bmatrix} 1 & 2 & 3 & \dots & 6 \\ 1 & 4 & 9 & \dots & 36 \\ 1 & 8 & 27 & \dots & 216 \end{bmatrix}^T$  și reprezentați-o grafic. Introduceți în grafic

o legendă cu textele 'liniar', 'pătratic' și 'cubic'. Activați rasterul de fundal. Setați prin meniul **Edit: Axes Properties** scalarea axei  $y$  ca logaritmică. Dați graficului titlul 'graphA'.

3.2. Reprezentați grafic funcția  $c = y * \sin(x)$  pentru  $x$  de la -10 la 10 (increment 1) și  $y$  de la 0 la 30 (increment 3) și setați suprafața de tip plasă.

3.3. Genrați graficul următoarei funcții:

$$y(x) = \begin{cases} \sin(x) & : \sin(x) \geq 0 \\ 0 & : \sin(x) < 0 \end{cases}$$

pentru intervalul  $[0, 3\pi]$ .

3.4. Faceți graficul  $y = \cos(2 * t)$  pentru 40 valori ale lui  $t$  între 0 și  $\pi$ . Curba  $y$  rezultată va fi de culoare roșie, se va afișa legenda cu font de 14, titlul  $y = \cos(2t)$  cu font de 16, Times New Roman.

### Referințe

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