2. Reprezentări grafice

Matlab oferă numeroase facilități pentru reprezentări grafice, oferind utilizatorului posibilitatea de a controla scalarea, modul de reprezentare, etichetarea axelor, existența unui caroiaj etc.

O reprezentare grafică 2D se bazează pe doi vectori, unul pentru axa 0x, celălalt pentru axa 0y, care definesc un set de puncte în plan. Comanda de construire a unui grafic 2D este plot, având forma generală:

plot(vector_x, vector_y,'stil'),

unde 'stil' este un șir de caractere, ce specifică o serie de opțiuni legate de stilul liniei ce unește punctele graficului, culoarea, grosimea, eventualele marcaje folosite. În cazul când nu se specifică stilul, Matlab va trasa graficul cu linie continuă de culoare albastră. In cazul cînd întrun grafic apar mai multe reprezentări, acestea vor fi trasate cu culori diferite

Cele două variabile, vector_x și vector_y, trebuie să conțină același număr de puncte, iar variabila specificată în prima poziție va fi reprezentată pe axa orizontală.

Șirul de caractere ce specific stilul poate fi format prin combinații de caractere conform tabel 2.1

Culoare	Marcaj punct	Tip de linie
b - blue	. punct	- linie continuă
g - green	o cerc	: linie punctată
r - red	x semn x	linie punct
c – cyan	+ semn +	linie întreruptă
m- magenta	* asterisc	
k - black	s pătrat	
w - white	d romb (diamond)	
	v triunghi cu vârful în jos	
	 triunghi cu vârful în sus 	
	< triunghi cu vârf spre dreapta	
	> triunghi cu vârf spre stânga	
	p stea cu 5 colțuri	
	h stea cu 6 colțuri	
	(nimic) fără marcaj	

Tabel 2.1 Caractere pentru specificarea modului de trasare a unui grafic

Ex. Fie doi vectori, v1 și v2: v1=[1, 3, 4, 6, 5, 2]; v2=[1, 2, 2, 3, 4, 3]; plot(v1,v2,'-o')



Fig. 2.1

Secvența de cod va produce graficul din. fig 2.1. Se observă că șirul '-o' va genera o linie continua ce unește cerculețele cu care sunt marcate punctele din grafic. Alte exemple de opțiuni:

'r-d' – linie roșie și marcaj romb;

'k:' – linie punctată neagră;

'pg-' – pentagon cu linie verde.

Comanda poate fi apelată și cu un singur argument, plot(x), rezultnd o reprezentare a vectorului x în raport cu numărul de ordine al elementului.

Comanda plot va genera reprezentarea grafică atribuind în mod automat valori pe cele două axe, astfel încât să fie vizibile toate valorile din grafic. Există numeroase situații când este necesar un control mai strict al scalării. Acesta poate fi realizat cu comanda axis, având sintaxa:

axis([xmin xmax ymin ymax]),

unde xmin este valoarea minima pe axa 0x, xmax – valoarea maximă pe axa 0x, ymin – valoarea minima pe axa 0y și ymax – valoarea maximă pe axa 0y.

Există o serie de alte comenzi cu care se poate controla comportamentul axelor:

axis manual - blochează scalarea la limitele curente;

axis auto – revine la modul de scalare automata;

axis tight - stabilește limitele pe axe egale cu domeniul de variație al datelor;

axis equal - scalarea pe axa 0x este egală cu cea de pe axa 0y;

axis square – egalează domeniile de variație pe cele două axe;

axis normal – anulează efectul comenzilor axis square și axis equal;

axis off – dispar axele din reprezentare;

axis on - sunt readuse axele în reprezentare;

grid on – apare un caroiaj pentru a facilita aprecierea valorilor numerice din grafic, iar cu grid off dispare caroiajul.

Se recomandă să existe etichete pe axele graficului, iar introducerea acestora se face cu comanda xlabel, ylabel. Cu comanda title se atribuie un titlu reprezentării grafice. Toate trei comenzile acceptă argumente de tip șir:

xlabel('timp [s]')

ylabel(['viteza [m/s]')

title('Variatia vitezei')

O altă posibilitate este introducerea unui text într-un grafic cu comanda

text(x,y,'text'). Caracterele aflate între semnul ', se introduc începând cu locația (x, y) din graficul curent. Textul mai poate fi introdus prin plasare cu mouse-ul, cu comanda **gtext('text').** Cursorul se modifică într-o cruciuliță subțire, care se poziționează în locația dorită.

Se mai poate introduce legenda cu comanda **legend ('nume1', 'nume2')**. Numărul de argumente din paranteză trebuie să fie în concordanță cu numărul funcțiilor reprezentate

în grafic. Legenda se poate deplasa cu mouse-ul, iar eliminarea ei se face cu comanda **legend off**.

Obs.: Toate comenzile ce controlează aspectul graficului se introduc în cod după comanda plot, nu înainte.

Etapele ce trebuie parcurse pentru construirea unei reprezentări grafice 2D sunt:

- se specifică vectorul ce reprezintă variabila independentă, care se va reprezenta pe axa 0x;
- se specifică vectorul asociat axei 0y;
- se construieşte graficul cu comanda plot();
- se specifică (dacă este cazul) scalarea, etichetele axelor etc.

În cazul când se dorește reprezentarea pe un domeniu de valori, a unei funcții matematice cunoscută analitic, cel mai frecvent se utilizează comanda linspace() pentru a genera vectorul corespunzător axei orizontale. Numărul de puncte ales pentru discretizare trebuie să fie suficient de mare, pentru a genera un grafic neted. Dacă se aleg puține puncte graficul apare format din linii frânte, deoarece la reprezentarea grafică, Matlab unește prin drepte punctele de coordonate (x_i, y_i).

Ex. Reprezentarea grafică a funcției:

$$x(t) = (at^2 + bt + c)e^{-t/\beta},$$

ce ar putea fi interpretată ca ecuația de mișcare oscilatorie amortizată (coeficientul de amortizare β) pe direcția 0x, t fiind timp. În acest caz variabila independentă este t, iar variabila dependentă este x. Datele de intrare sunt: a, b, c, β și domeniul de valori pentru variabila t.

%% grafic_polinom_amortizare.m % construieste grafic pentru functia % x(t)=(at^2+bt+c)*exp(-t/beta)

%%date intrare tmin=0; tmax=3; n=100;% nr puncte de discretizare a=1; b=-7; c=1; beta=0.4; %% calcule t=linspace(tmin,tmax,n);

 $x=(a^t.^2+b^t+c).^*exp(-t./beta);$

%% afisare rezultate plot(t,x,'k-') xlabel('t'), ylabel('x(t)') grid on



Fig. 2.2

Se observă delimitarea clară între datele de intrare și partea de calcule și afișare. Adaptarea programului pentru alte valori se face prin interventia numai în zona de date de intrare, ceea ce permite modificări care se pot urmări foarte ușor. Evident programul ar fi functionat la fel de correct, dacă vectorul t era definit direct t = linspace(0,3,100), dar se recomandă ca orice aplicație, oricât de simplă este să aibă un anumit grad de generalitate.

În unele situații este util să se marcheze anumite puncte din grafic, de ex. un punct de minim sau de maxim. Acest lucru se poate face simplu în Matlab, prin utilizarea funcției min sau max cu două argumente de iesire:

 $[val_min, pozitie] = min(x);$

O astfel de apelare va returna valoarea minimă (val min) a funcției și poziția acesteia (adică indexul din vectorul x ce corespunde acestei valori). Dacă se dorește marcarea acestui punct în reprezentarea grafică se poate utiliza comanda hold on, care mentine reprezentarea anterioară din fereastra grafică si face următoarea reprezentare în aceeasi fereastră. Punctele care sunt reprezentate grafic sunt în pereche, deci daca valoarea minimă a lui x are indexul pozitie, cealaltă variabilă t corespunzătoare valorii minime va avea acelasi indice, adică punctul de minim este de coordonate (t(pozitie), x(pozitie))

În programul anterior se inserează la sfârsit codul:

[val min, pozitie] = min(x);

hold on,plot(t(pozitie),x(pozitie),'ro') Rezultatul se evidențiază în fig 2.3





Comanda hold off revine la varianta implicită, când fiecare reprezentare se face în fereastră separate sau se poate utiliza și comanda hold, care modifică setarea între on/off.

La fiecare reprezentare în fereastra grafică, conținutul anterior se șterge. Pentru a menține toate reprezentările grafice ce pot apărea într-o aplicație se poate utiliza comanda figure, ce deschide o nouă fereastră grafică. De regulă, comanda figure se plasează înaintea comenzii de reprezentare grafică. Ex. figure, plot(x,y)

Cu comanda subplot se pot reprezenta mai multe grafice în aceeaşi fereastră. Sintaxa este **subplot(m,n,p)** – se împarte fereastra în m x n locații, iar locația p este activă, adică în acest loc se va face următoarea reprezentare grafică. Zonele active se numerotează de la stânga la dreapta, începând cu rândul superior. Pentru revenirea la fereastra grafică cu o singură reprezentare, se utilizează comanda **subplot(1,1,1)**. Exemplu:



Fig. 2.4. Utilizarea comenzii subplot

Mai există și alte tipuri de grafice:

- area(x,y) are acelaşi efect cu comanda plot, dar suprafaţa de sub curbă este umplută;
- pie(a) sau pie(a,b) generează grafice tip pie-chart, unde a este un vector, iar b este un vector de aceeaşi dimensiune cu a. Segmentele de cerc corespunzătoare elementelor lui b care sunt diferite de zero vor fi scoase în afara graficului.

Exemplu: a=[.5 1 1.6 1.2 .8 2.1];

```
subplot(1,2,1)
pie(a,a==max(a))
subplot(1,2,2)
q=sort(a);
```

b=zeros(size(a)); b(end-2:end)=q(end-2:end); pie(a,b)



Fig.2.5. Grafice tip pie-chart

Reprezentarea grafică poate fi urmărită în fig. 2.5

- grafice Pareto ce reprezintă valorile unui vector în ordine descrescătoare, sub formă de segmente, împreună cu valoarea cumulată ce se asociază unui procent de 100%. Pentru exemplul anterior se dă comanda **pareto(a)**, iar graficul generat este prezentat în fig. 2.5
- histograma cu împărţire în 10 clase se face cu comanda hist(y). Dacă se doreşte împărţirea în k clase, comanda va avea sintaxa hist(y,k) sau hist(y,x), unde x este un vector ce specifică clasele.

Ex.: Pentru exemplificare se va utiliza funcția randn(1,N), ce returnează un set de N numere pseudoaleatoare, distribuite normal, având media egală cu 0 și deviația standard egală cu 1. Prin translatare și scalare se va genera un set de date provenind dintr-o repartiție normală cu medie m și deviatie standard sigma.

%%hist_norm.m

%genereaza histograma de numere aleatoare cu medie m si abatere sigma

%% date intrare N=10000;%volum date m=1.5; % media sigma=0.25; %abatere standard val_min=m-4*sigma; %valoare minima axa val_max=m+4*sigma: %valoare maxima axa clase=25; %numar clase %%generare date x=m+sigma*randn(1,N); %% construire histograma hist(x,clase); xlabel('x') ylabel('Frecventa absoluta') axis([val_min, val_max, 0, inf])



Fig. 2.6. Diagrama Pareto



Fig. 2.7. Histogramă



Fig. 2.8. Grafic în coordonate polare

Histograma este prezentată în fig. 2.7.

grafice în coordonate polare se fac cu comanda polar(t,r) – t este unghiul în radiani, iar r este raza vectoare. Evident pot apărea şi caractere opţionale pentru definirea culorii, marcajului sau tipului de linie.

```
Ex.: t=0:.01:2*pi;
```

```
r=sin(2*t).*cos(2*t);
polar(r,t)
title('Reprezentarea in coordonate polare a
sin(2t)cos(2t)')
```

Graficul corespunzător este în fig. 2.8.

- grafice tip bară evidențiază elementele stocate într-un vector. Comanda bar(v), având ca argument un singur vector construiește graficul menționat.

%% grafic_bar1.m % demonstreaza graficul tip bara

%% date intrare

temperatura=[71, 80, 73, 72, 78, 81, 73, 77];



scalare a axei verticale pentru a evidenția diferențele dintre valori.

%% grafic_bar2.m

% demonstreaza graficul tip bara

% sorteaza elementele si decaleaza scala 0y

%% date intrare

temperatura=[71, 80, 73, 72, 78, 81, 73, 77];

temp_sort=sort(temperatura);

Tmin=min(temperatura);

Tmax=max(temperatura);

%% constructie grafic bar(temp_sort); xlabel('Masurari') ylabel('Temperaturi [°C]'); grid on

axis([0, length(temperature)+1, Tmin, Tmax]);

În mod normal, la fiecare comandă plot() fereastra grafică se șterge și se trasează noul grafic. În cazul când se dorește să se reprezinte mai multe funcții în același grafic se poate apela comanda plot cu mai multe perechi de argumente sau se poate utiliza comanda hold. În general hold se utilizează la grafice de tip diferit, ex. o histogramă suprapusă cu un grafic de tip plot.

Reprezentarea unei funcții sinus și a unei funcții cosinus într-un același grafic se poate face cu următorul program (fig.2.11):

%% plot_sin_cos.m

% construieste grafic functia sin si cos in acelasi grafic

%% date intrare N=300; % numar puncte xmax=3*pi; % valoare maxima

%% calcul functii x=linspace(0,xmax,N); y1=sin(x); y2=cos(x);

%%reprezentare grafica plot(x,y1,x,y2); grid on xlabel('x') ylabel('sin(x), cos(x)') legend('sin(x)', 'cos(x)');

Se observă că pentru clarificare s-a introdus și o legenda pentru grafic, sintaxa fiind: legend('sir1', 'sir2', 'sir3',...)

șirurile de caractere incluse în comandă trebuie specificate ordinea corespunzătoare comenzii plot.



Fig. 2.11

Într-o comandă plot se pot introduce oricâte reprezentări se dorește, singura condiție este ca numărul de puncte din perechea de vectori care constituie o reprezentare să fie egal. Evident, numărul de puncte al fiecărei reprezentări poate diferi. De exemplu, în graficul anterior se mai poate introduce o linie verticală construită în dreptul valorii π , cu următorul program:

%% plot_sin_cos1.m

% construieste grafic functia sin si cos in acelasi grafic

%% date intrare	
N=300; % numar puncte	
xmax=3*pi; % valoare maxima	

%% calcul functii x=linspace(0,xmax,N); y1=sin(x); y2=cos(x);

%%reprezentare grafica plot(x,y1,x,y2,[pi, pi], [-1, 1]); grid on xlabel('x') ylabel('sin(x), cos(x)') axis([0, xmax, -1, 1]) legend('sin(x)', 'cos(x)');

Comanda plot întotdeauna construiește reprezentarea grafică în fereastra grafică, reprezentarea anterioară fiind ștearsă. Spre deosebire de comanda plot, comenzile line și text adaugă elemente grafice la reprezentarea existentă în fereastra grafică.



Comanda line are sintaxa:

line(x, y)

unde x și y sunt vectori de aceeași dimensiune. Efectul este conectarea prin linii a punctelor de coordonate (x_i, y_i). Obiectul line admite specificarea anumitor proprietăți: Color, LineWidth etc. Există multe proprietăți ale obiectului line, ce se pot consulta în documentația Matlab, dar cele mai utilizate sunt Color și LineWidth. De exemplu, proprietatea Color poate admite ca valori: 'blue', 'green', 'red' etc. sau se pot utiliza ințialele, conform Tabel 2.1, 'b', 'g', 'r'. Specificarea unei linii de culoare roșie se poate face cu comanda:

line(u,v,'Color','r')

rezultatul fiind o suucesiune de linii ce unesc punctele de coordonate (u₁, v₁), (u₂, v₂), (u₃, v₃)...

Proprietatea LineWidth permite specificarea grosimii liniei, valoarea implicită fiind 0.5 puncte. Un punct este 1/72 inch. Valorile admise pentru această proprietate sunt valori reale pozitive.



Fig.2.13 Repartiție normală cu marcarea domeniului [- σ , + σ]

În afara graficelor bidimensionale, se pot trasa grafice 3D. O primă comandă pentru desenarea graficelor tridimensionale este **plot3(x,y,z)**, unde x, y, şi z sunt vectori de aceeaşi dimensiune.



Fig. 1.11. Grafic 3D

Comanda axis se menține la graficele tridimensionale, cu sintaxa: **axis([xmin xmax ymin ymax zmin zmax])**, respectiv cu indicarea limitelor inferioare și superioare de pe fiecare axă. De asemenea, comanda **text(x,y,z,'text')** poate fi aplicată și în acest caz.

În afara desenării curbelor tridimensionale cu comanda plot3, pot fi desenate suprafeţele folosind comanda **mesh** şi **surf.** Pentru generarea graficului z = f(x, y) trebuie, în primul rând, generată matricea x şi y, în care se evaluează funcţia. Comanda destinată acestui scop este: **[X,Y]=meshgrid(x,y)**, care generează o matrice X, ale cărei linii sunt copii ale vectorului x şi o matrice Y, ale cărei coloane sunt copii ale vectorului y. Sintaxa comenzii de desenare a suprafeţei, reprezentată "cadru-sârmă", este **mesh(X,Y,Z)**, iar pentru umplerea elementelor de suprafaţă **surf(X,Y,Z)**.

Ex.: **x=-7.5:.5:7.5;**

y=x;

```
[X,Y]=meshgrid(x,y);
```

R=sqrt(X.^2+Y.^2)+eps;%calculeaza distanta pana la origine pentru fiecare

punct

```
Z=sin(R)./R;
subplot(2,1,1),mesh(X,Y,Z)
subplot(2,1,2),surf(X,Y,Z)
```



Fig. 1.12. Suprafețe desenate cu comanda mesh și surf

Ex.2: Se cere trasarea graficului funcției: $z = ye^{-(x^2+y^2)}$ pentru $x \in [-2, 2]$ și $y \in [-1, 2]$. x=-2:.1:2; y=-1:.1:2; [X,Y]=meshgrid(x,y); Z=Y.*exp(-(X.^2+Y.^2)); subplot(2,1,1),mesh(X,Y,Z),xlabel('x'),... ylabel('y'),zlabel('z') subplot(2,1,2),surf(X,Y,Z),xlabel('x'),... ylabel('y'),zlabel('z') Graficele de la cele două exemple sunt prezentate ln fig. 1.12, respectiv în fig. 1.13.



Fig. 1.13. Reprezentarea funcției $z = ye^{-(x^2+y^2)}$



Fig. 1. 14. Reprezentarea curbelor de nivel cu comanda contour

O altă posibilitate de reprezentare grafică pentru graficele tridimensionale este reprezentarea curbelor de nivel (curbelor ce au valorile z = ct.). În acest context, x și y reprezintă vectorii ce definesc domeniul de valori pentru variabila x, respectiv y (nu se mai transformă vectorii în matrice). Numărul liniilor de contur și valorile acestora se aleg automat sau se indică de utilizator. Pentru exemplul 2, graficul corespunzător comenzii contour(x,y,z) se prezintă în fig. 1.14.

Se pot genera simultan atât reprezentarea suprafeței, cât și a curbelor de nivel cu comanda meshc(X,Y,Z) (fig. 1.15)



Fig. 1.15. Reprezentare cu comanda meshc

Matlab-ul permite specificarea punctului din care se face vizualizarea graficelor 3D, prin proiecție perspectivă. Funcția este view(azimut, elevație), unde elevația definește unghiul făcut de observator cu planul x0y, iar azimutul definește unghiul din planul x0y. Azimutul se măsoară în grade, în sens orar, față de axa negativă 0y (vezi fig. 1.16).

Implicit valoarea azimutului este -37,5°, iar elevaţia este 30°. În fig. 1.17 sunt prezentate câteva exemple de vizualizări a unui grafic, din mai multe poziţii, comanda unghiului de azimut şi elevaţie fiind specificată în titlul fiecărui grafic. Cu **view(3)** se revine la setarea implicită.



Fig. 1.16. Unghiurile de azimut și elevație



Fig. 1.17. Exemplificări pentru comanda view

Pentru modificare paletei de culori utilizate de grafic, se foloseşte comanda: **colormap(varianta)**. variantele existente sunt: hsv, hot, gray, bone, white, copper, pink, flag, jet, prism, cool, lines, colorcube, summer, autumn, winter, spring. Ex.: **colormap(cool)**.

Probleme

- 1. Să se reprezinte funcția $f(x) = sin(\frac{x}{\lambda})$, pentru x ε [-3π, 3π]. Se vor verifica mai multe valori pentru λ.
- 2. Să se reprezinte grafic șirul definit pe baza relațiilor:

$$f_{1} = 0$$

$$f_{2} = 1$$

$$f_{k} = \frac{f_{k-2} + f_{k-1}}{2}, \quad k > 2$$

true k 25

Pentru k=25

3. Să se reprezinte grafic funcția:

$$f(x) = e^{-\frac{x}{a}} \cos\left(2\pi \frac{x}{\lambda}\right).$$

Se alege un domeniu pentru x și se modifică valorile pentru parametrii a și λ.

4. Să se reprezinte grafic mișcarea balistică:

$$h(t) = h_0 + v_0 t - \frac{1}{2}gt^2.$$

Se consideră g = 9.8 m/s² și h_0 >0. Se vor testa diferite valori pentru v_0 și h_0