CURS 5: GRAFICĂ ÎN MATLAB

1. REPREZENTĂRI GRAFICE 2D

1.1. Reprezentări grafice elementare

Funcţiile MATLAB pentru reprezentări grafice elementare sunt:

plot Reprezintă grafice în coordonate X-Y liniare;

loglog Reprezintă grafice în coordonate X-Y logaritmice;

semilogx Reprezintă grafice în coordonate X-Y semilogaritmice (axa X este logaritmică);

semilogy Reprezintă graficele în coordonate X-Y semilogaritmice (axa Y este logaritmică);

fill Reprezintă grafic poligoane.

1.1.1. Reprezentarea grafică în coordonate liniare

Pentru reprezentarea datelor în coordonate liniare se utilizează funcţia ***plot***, se apelează cu una dintre sintaxele:

plot(y) plot(x,y)

plot(x,y,'linie-tip') plot(x1,y1,x2,y2,...)

plot(y) - reprezintă grafic argumentul y funcţie de indici, cu următoarele precizări:

a) - dacă argumentul y este complex, plot(y) este echivalent cu plot(real(y),imag(y)),

b) - dacă y este vector (linie sau coloană), funcţia ***plot*** trasează graficul y=y(i), unde

i =1,2,...n este numărul de ordine al elementului y;

c) - dacă y este o matrice m x n, funcţia ***plot*** trasează graficele $y_j = y_j(i)$, unde

i=1,2,...n este numărul de ordine al elementului de pe coloana j;

plot(x,y) - reprezintă grafic vectorul y funcţie de vectorul x, cu următoarele precizări:

a) - dacă x este vector, iar y este matrice, atunci coloanele lui y sunt trasate funcţie de vectorul x;

b) - dacă x şi y sunt matrice de aceeaşi dimensiune, se reprezintă coloanele lui y funcţie de coloanele Iui x.

plot(x1,y1,x2,y2) - reprezintă simultan mai multe grafice în acelaşi sistem de coordonate.

Graficele se pot reprezenta utilizând linii, markere şi culori după codul din tabelul 1.

Pentru reprezentările grafice, se asociază fiecărei caracteristici un şir de 1-3 caractere, dintre cele menţionate în tabelul de mai sus. Aceste şiruri de caractere trebuie cuprinse între apostrofuri şi menţionate în combinaţia culoare-marker sau culoare-linie-tip. Dacă se precizează o singură caracteristică (marker, linie sau culoare), cea de-a doua este selectată de calculator. Astfel, instrucţiunea plot(x1,y1,':',x2,y2,'+r') reprezintă cu „linie din puncte" pentru prima caracteristică (x1,y1), şi cu markere „plus" de culoare roşie, a doua caracteristică.

Tabelul 1.

LINII TIP		MARKERE TIP		CULORI	
continuă	-	plus	+	albastru	b
întreruptă	_	steluţă	*	mov	m
puncte	:	cerc	o	albastru-deschis	c
linie-punct	-.	x	x	roşu	r
		punct	.	verde	i
				galben	y
				alb	w
				negru	k

Dacă nu se specifică culoarea, MATLAB-ul foloseşte implicit albastru. Pentru grafice multiple se utilizează succesiv primele şase culori din tabel.

Pentru o citire mai precisă a datelor pe grafic reprezentarea se poate asocia cu funcţia de trasare a caroiajului ***grid***.

Funcţia ***plot*** returnează un vector coloană al identificatorilor de control al caracteristicilor obiectelor linie. Obiectele linie create cu ***plot*** sunt copii ai axelor curente.

Perechile (x,y) pot fi urmate de perechile parametru/valoare, pentru a specifica proprietăţi suplimentare ale liniilor.

Exemplul 1. Să se reprezinte grafic funcţiile: f(x)=sin(2*pi*50*t) cu linie-punct de culoare verde si g(x)=f(x)+.2 cu markere-stea de culoare roşie. Cu secvenţa MATLAB: t=0:.001:.02; f=sin(2*pi*50*t); g=f+.2; plot(t,f,'-.g',t,g,'*r')

1.2. Reprezentarea grafică în coordonate logaritmice şi semilogaritmice

Pentru reprezentările grafice în coordonate logaritmice sau semilogarirmice se utilizează funcţiile

loglog, semilogx, semilogy; se apelează cu sintaxele:

loglog(x,y) semilogx(x,y) semilogy(x,y)

Funcţia ***loglog*** scalează ambele axe utilizând logaritmul în baza 10, în timp ce funcţiile semilogx sau semilogy scalează logaritmic numai axa x, respectiv axa y, cealaltă axă fiind scalată liniar. Modul de utilizare al acestor funcţii este la fel ca acela al funcţiei ***plot***.

Exemplul 2. Să se reprezinte în coordonate semilogaritmice (axa y) funcţia: f(x)=10". Cu secvenţa MATLAB: x=0:.l:10; f=10.^x; semilogy(x, f)

1.3. Reprezentarea grafică a poligoanelor

Reprezentarea grafică a poligoanelor utilizează funcţia ***fill*** se apelează cu una dintre sintaxele:

Fill(x,y,c) . fill(x,y,'c') fill(x1,y1,c1,x2,y2,c2,...)

Funcţia fill(x,y,c) reprezintă un poligon definit de vectorii x şi y, cu nuanţele de culoare precizate de c. Coordonatele vârfurilor poligonului sunt specificate prin perechile (x,y).

Dacă argumentul c este un singur caracter dintre cele prezentate în lista de culori din tabelul 1 sau un vector cu trei componente [r g b], poligonul va fi colorat într-o singură culoare, în cazul în care c este un vector cu aceeaşi dimensiune ca x şi y, elementele acestuia sunt scalate cu funcţia ***caxis*** şi apoi utilizate ca indici într-o matrice care specifică culorile vârfurilor. Culorile dintre vârfuri sunt obţinute prin interpolare biliniară a culorilor vârfurilor.

Dacă x şi y sunt matrice de aceeaşi dimensiune, fill(x,y,c) reprezintă câte un poligon pentru fiecare coloană, în acest caz, c este un vector linie pentru poligoane cu o singură culoare si respectiv o matrice pentru poligoane cu culori interpolate.

Dacă numai unul dintre argumentele x sau y este matrice, celălalt fiind vector coloană cu acelaşi număr de linii, vectorul coloană se va extinde la o matrice cu aceleaşi dimensiuni, prin adăugarea unor coloane identice.

Pentru specificarea poligoanelor multiple se poate utiliza şi forma fill(x1,y1,c1,x2,y2,c2,...), mult mai uşor de controlat.

Exemplul 3. Să se reprezinte grafic poligoanele cu secvenţa MATLAB:

xl=[0 3 2 -1]; yl=[-l -l 2 2]; x2=[3 5 2]; y2=[-l 3 2]; fill(xl,yl,'b',x2,y2,'g').

2. REPREZENTĂRI GRAFICE SPECIALE

Funcţiile MATLAB pentru reprezentări grafice speciale sunt:

polar Reprezintă grafice în coordonate polare;

bar Reprezintă grafice cu bare;

stem Reprezintă grafice sub formă discretă (util pentru semnale discrete);

stairs Reprezintă grafice în trepte (util pentru semnale cuantizate);

errorbar Evidenţiază eroarea datelor reprezentate grafic;

hist Reprezintă grafic histograma datelor;

rose Reprezintă grafic histograma unghiulară a datelor (coordonate polare);

compass Reprezintă grafic vectorii argument, precizaţi prin proiecţiile pe axe, cu săgeţile orientate dinspre origine;

feather Reprezintă grafic vectorii argument, precizaţi prin proiecţiile pe axe, ordonaţi echidistant pe axa orizontală;

fplot reprezintă controlul reprezentărilor grafice cu parametrii impuşi

cornet Realizează controlul reprezentărilor grafice cu parametri impuşi; Reprezintă dinamic (în mişcare) traiectoria unui punct într-o reprezentare grafică 2D.

2.1 . Reprezentarea grafică în coordonate polare

Reprezentarea în coordonate polare se face cu funcţia ***polar***, se apelează cu una dintre sintaxele:

polar(theta,r) polar(theta, r, 'linie-tip')

Modul de folosire al opţiunii linie-tip este identic cu cel al funcţiei ***plot***.

Exemplul 2.1. Să se reprezinte în coordonate polare funcţia:

f(x)=sin(2t)cos(2t). Cu secvenţa MATLAB: t=0:.01:2*pi; f=sin(2*t)*cos(2*t); polar (t, f)

2.2. Reprezentarea graficelor cu bare

Reprezentarea grafică cu bare se face cu funcţia bar, se apelează cu una dintre sintaxele:

bar(y) - trasează un grafic de bare cu elementele vectorului y, adică $y=y_i$.

bar(x.,y) - trasează un grafic de bare cu elementele vectorului y la locaţiile specificate de vectorul x, adică y=y(x). Valorile lui x trebuie să fie egal depărtate şi crescătoare.

[xb.yb] = bar(y), şi

[xb.yb] = bar{x,y) - nu reprezintă graficele, dar calculează vectorii xb si yb astfel încât plot(xb.yb) să poată trasa graficul de bare. Aceasta e utilă în situaţiile în care se doreşte un control mai mare asupra graficului; spre exemplu, combinarea mai multor grafice de bare elaborate cu instrucţiunea ***plot***.

Exemplul 2.2. Să se reprezinte graficul cu bare al datelor conţinute în vectorul y=[1 3 7 6 5 2 3]. Cu secvenţa MATLAB: y=[l 3 7 6 5 2 3]; bar(y).

2.3. Reprezentarea discretă a datelor

Reprezentarea grafică a semnalelor discrete se face cu funcţia ***stem***, sub forma unor linii terminate cu cerculeţ la extremitatea opusă axei; se apelează cu una dintre sintaxele:

stem(y) - trasează un grafic din linii cu cerculeţ, cu elementele vectorului y; stem(x,y) - trasează un grafic din linii terminate cu cerculeţ, cu locaţiile specificate de vectorul x, adică y=y(x). Valorile lui x trebuie să fie egal depărtate şi crescătoare.

stem(x,y, linie_tip) - trasează un grafic din linii de tipul şi culoarea precizată în şirul de caractere linie_tip, aşa cum a fost precizat la funcţia ***plot***. Spre exemplu:

stem(x,y,':r').

Exemplul 2.3. Să se reprezinte grafic funcţia discretă sinus: f[n] = sin(2*pi/10*n), n∈ [0,20]

2.4. Reprezentarea graficelor în trepte

Graficele în trepte sunt utilizate la reprezentarea diagramelor sistemelor numerice de eşantionare si prelucrare a datelor.

Reprezentarea grafică în trepte se face cu funcţia ***stairs***, care se apelează cu una dintre sintaxele:

stairs(y) - trasează graficul în trepte al elementelor vectorului y.

stars(x,y) - trasează graficul în trepte al elementelor vectorului y la locaţiile specificate în x. Valorile lui x trebuie să fie egal depărtate si în ordine crescătoare,

[xb.yb] = stars(y), şi [xb.yb] = stars(x,y) - calculează vectorii xb şi yb, asfel încât plot(xb.yb) să poată trasa graficul în trepte.

Exemplul 2.4. Să se reprezinte graficul în trepte ai funcţiei y=sin(x). Cu secvenţa MATLAB: x=0:.2:6; y=sin(x); stairs.(x,y).

2.5. Reprezentarea grafică a erorilor

Reprezentarea grafică a datelor cu bare de eroare ataşează fiecărei perechi (x,y) eroarea precizată într-un vector cu aceleaşi dimensiuni; se apelează cu sintaxa:

errorbar(x,y,e)

Vectorul ***e*** conţine lungimea barelor ce reprezintă eroarea. „Barele de erori" se reprezintă simetric în raport cu ordonata y, ceea ce presupune o asociere de erori pozitive sau negative, cu aceeaşi probabilitate. Segmentele de eroare sunt de înălţime 2*e şi se trasează pe curba y=y(x). Figura obţinută reprezintă plaja de valori pe care o poate lua funcţia y cu eroarea „e".

Dacă x şi y sunt matrice de aceeaşi dimensiune,1 funcţia ***errorbar*** va reprezenta graficul cu bare de eroare pentru fiecare coloană în parte.

Exemplul 2.5. Să se reprezinte un grafic cu bare de erori pentru:

y=sin(x). Cu secvenţa MATLAB: x = 0:.2:6; y = sin(x); e=rand(size(x))/5; errorbar(x,y,e) se obţine

2.6. Reprezentarea grafică a histogramelor

Calculul şi reprezentarea grafică a histogramelor se face cu funcţia ***hist***, se apelează cu una dintre sintaxele:

hist(y) - trasează histograma cu 10 segmente a datelor vectorului y;

hist(y,nb) - trasează histograma cu nb segmente a datelor vectorului y;

hist(y,x) - trasează histograma datelor vectorului y la abscisele specificate în x;

[n,x]=hist(y),

[n,x]=hist(y,nb)

[n,x]=hist(y,x) - returnează vectorii n şi x conţinând frecvenţa de apariţie şi localizarea segmentelor; cu bar(x,n) se poate apoi trasa histograma. Aceste proceduri se utilizează în situaţiile în care este necesară o mai mare flexibilitate în reprezentările grafice. Combinarea histogramei cu instrucţiuni mai elaborate de trasare şi reprezentare grafică poate conduce la obţinerea unor grafice deosebit de sugestive.

Exemplul 2.6. Să se genereze histograma unui vector cu elementele distribuite normal (Gaussian). Cu secvenţa MATLAB:

x=-3:.3:3; y=randn(10000,1);.hist(y,x)

Reprezentarea unei histograme în coordonate polare se face cu funcţia ***rose***; se apelează cu una dintre sintaxele:

rose(x) ' rose(x,N) [t,r]=rose(...)

unde x trebuie să fie în intervalul $[0, 2\pi]$, iar N este numărul de subintervale în care se împarte intervalul $[0, 2\pi]$. Valoarea implicită pentru N este 20.

Dacă funcţia ***rose*** se apelează cu argumente, atunci returnează cei doi vectori t şi r, care se vor folosi pentru reprezentarea histogramei polare, cu funcţia polar(i,r).

Exemplul 2.7. Să se reprezinte grafic histograma a 100 numere aleatoare în coordonate polare. Cu secvenţa MATLAB: x=2*pi*rand(100,1); rose(x,10)

2.7. Reprezentarea grafică a vectorilor

Funcţia ***compass*** reprezintă grafic vectori cu originea în originea sistemului de coordonate; se apelează cu una dintre sintaxele:

compass(z) compass(x,y)

unde z este numărul complex x+iy, iar x şi y sunt numere reale - proiecţia vectorului pe abscisă şi ordonată.

Exemplu 2.8. Să se reprezinte grafic vectorii:

zl = 2-5i z2 = -2 + i z3 = 3 + 2i

Cu secvenţa MATLAB:

z=[2-5*i, -2+i, 3+2*i]; compass(z); grid

Funcţia ***feather*** reprezintă grafic vectori cu originea plasată echidistant pe axa Ox; se apelează cu una dintre sintaxele:

feather(z) feather(x,y)

unde z este numărul complex x+iy, iar x şi y sunt numere reale - proiecţia vectorului pe abscisă şi ordonată.

Exemplul 2.9. Să se reprezinte grafic cu originile pe axa Ox vectorii:

zl = 2-5i z2 = -2 + i z3 = 3 + 2i

Cu secvenţa MATLAB: z=[2-5*i, -2+i, 3+2*1]; feather(z); grid

2.8. Reprezentări grafice cu parametri impuşi

Funcţia ***fplot*** realizează o reprezentare grafică cu anumite restricţii; se apelează cu una dintre sintaxele:

fplot('fun', limite) fplot(‘fun', limite,n,unghi)

fplot('fun' limite,n) fplot('fun', limite,n,unghi.subdiv)

[x,y]= fplot('fun', limite,...)

Funcţia ***fplot*** reprezintă grafic funcţia „fun" între limitele specificate de argumentul „limite". Argumentele funcţiei ***fplot*** au următoarele semnificaţii:

fun - numele fişierului funcţie (şir de caractere);

limite=[xmin xmax] - limitele axei x pentru care se doreşte reprezentarea grafică;

n - numărul de eşantioane cu care este reprezentată funcţia (implicit n=25);

unghi - cea mai mare schimbare de unghi dintre două segmente adiacente ale graficului (implicit 10°);

subdiv - numărul maxim de subdiviziuni dintre două diviziuni ale scalei pentru care unghiul dintre două segmente adiacente nu este mai mare decât valoarea impusă (implicit 20).

Dacă funcţia ***fplot*** se apelează cu argumentele de ieşire [x,y], în aceşti vectori coloană sunt returnate valorile abscisei şi ordonatei funcţiei. Reprezentarea grafică se poate face ulterior cu funcţia plof(x,y).

Exemplul 2.9. Fie fişierul funcţie:

function y=test{x); y=sin(x)./x;

înregistrat cu numele testm. Să se reprezinte grafic funcţia din fişierul test.m, între limitele [-20, 20] cu n=50 eşantioane. Cu secvenţa MATLAB:

fplot ('test', [-20 20], 50); grid

2.9. Reprezentări grafice dinamice 2D

Pentru a reprezenta grafic un punct care urmăreşte realizarea grafică („cometă") se foloseşte funcţia ***comet***, se apelează cu una dintre sintaxele:

comet(y) comet(x,y) comet(x,y,p)

Apelată fără argumente de intrare, funcţia cornet lansează demonstraţia cu acelaşi nume. Dacă se apelează cu unul sau cu două argumente de intrare, se realizează o reprezentare grafică a lui y sau y(x), urmărite de un marker („cometă"), la o distantă de 0.1 din lungimea vectorului y. Apelarea cu trei argumente realizează acelaşi lucru, însă distanţa de urmărire este setată la valoarea p din lungimea totală a vectorului y.

Exemplul 20.2.9. Urmăriţi modul de lucru al funcţiei ***comet***, cu secvenţa MATLAB:

t = 0:1/10000:.1; y=sin(2*pi*50*t); cornet (t, y, .2)

3. REPREZENTĂRI GRAFICE 3D

În acest capitol se prezintă funcţiile MATLAB pentru reprezentări grafice în plan sau în spaţiu asociate câmpurilor bidimensionale:

clabel Plasează marcaje pe liniile de contur, referitoare la cota Z;

comet3 Reprezintă dinamic (în mişcare) traiectoria unui punct, într-o reprezentare grafică 3D;

contour Reprezintă grafic în 2D liniile de contur (liniile de nivel constant);

contourc Returnează o matrice care conţine perechile (cota Z - număr puncte şi coordonata X - coordonata Y);

contour3 Reprezintă grafic în 3D liniile de contur (liniile de nivel constant);

fill3 Reprezintă grafic poliedre în 3D;

plot3 Reprezintă grafice în 3D;

quiver Reprezintă grafic orientarea unui câmp de vectori.

3.1. Reprezentarea liniilor de contur

Prin linii de contur se înţeleg liniile situate la acelaşi nivel pe axa Z (linii de nivel constant).

3.1.1. Calculul matricei liniilor de contur

Matricea care conţine perechile de coordonate (X,Y) ale fiecărei linii de contur, se determină cu funcţia ***contourc***, se apelează cu una dintre sintaxele:

C=contourc(Z) - calculează matricea liniilor de contur ale matricei Z. Numărul liniilor de nivel şi valorile acestora sunt alese automat;

C= contourc(Z,n) - calculează matricea liniilor de contur ale matricei Z pentru n linii de contur (n este un scalar);

C= contourc(Z,v) - calculează matricea liniilor de contur ale matricei Z la nivelele specificate de vectorul v;

C= contourc(x,y,Z), C= contourc(x,y,Z,n) şi

C= contourc(x!y,Z,v) - calculează matricea liniilor de contur ale matricei Z şi utilizează date din vectorii x şi y pentru a controla scalarea axelor Ox şi Oy. Elementele x şi y sunt cu pas constant.

Matricea liniilor de contur, C, este o matrice cu două linii, ca în exemplul:

C=[nivel1 x1 x2 x3 ... nivel2 x1 x2 x3 ...

perechi1 y1 y2 y3 ... perechi2 y1 y2 y3 ...]

Vectorii x şi y trebuie să fie monoton crescători şi cu pas constant.

3.1.2. Reprezentarea grafică în plan a liniilor de contur

Reprezentarea grafică în plan a liniilor de contur face apel la funcţia ***contour***; se apelează cu una dintre sintaxele:

contour(Z) - reprezintă conturul liniilor de acelaşi nivel ale matricei Z. Colţul din stânga sus al desenului corespunde elementului Z(1.1). Numărul liniilor conturului şi valorile cotelor Z sunt alese automat;

contour(Z,N) - reprezintă conturul a n linii de nivel ale matricei Z. Dacă N nu este specificat, valoarea implicită este 10;

contour(Z,N) - reprezintă conturul liniilor de nivel ale matricei Z, caracterizate de valorile specificate în vectorul V;

contour(X,Y,Z,N), şi

contour(X,Y,Z,V) - reprezintă conturul liniilor de nivel ale matricei Z, utilizând; N - pentru numărul liniilor de nivel, V - pentru valorile liniilor de nivel, X - pentru controlul scalarii pe axa Ox, Y - pentru controlul scalarii pe axa Oy. Dacă scalele nu se precizează (vectorii x şi y lipsesc), reprezentarea se face după numărul de elemente în care au fost divizate axele x şi y,

contour(...,'tip-linie') - reprezintă liniile de contur cu liniile şi culorile specificate;

C=contour(....) - returnează matricea liniilor de contur, C, descrisă la funcţia contoare si este utilizată de funcţia ***clabel***;

[C,H]=contour(...) - returnează matricea liniilor de contour, C, şi un vector coloană H al identificatorilor obiectelor linie.

Exemplul 3.1. Să se reprezinte grafic conturul liniilor de nivel ale funcţiei $z=x*exp(-x^2-y^2)$

în domeniul: -2 < x < 2, -2 < y <, 3 Cu secvenţa MATLAB:

[x,y] = meshdom(-2:.2:2, -2:.2:3); z = x.*exp(-x.^2-y.^2) ; contour(-2:.2:2,-2:.2:3,z, 8)

3.1.3. Etichetarea cotelor liniilor de contur

Pentru a preciza cotele (nivelul) liniilor de contur într-o reprezentare grafică se foloseşte funcţia ***clabel***; se apelează cu una dintre sintaxele:

clabel(C) • etichetează liniile de nivel. Poziţia acestora este aleasă aleator;

clabel(C,V) - etichetează liniile de nivel precizate de vectorul V;

clabel(C,'manual') - etichetează liniile de nivel selectate cu mouse-ul. Se apasă tasta „ENTER" pentru a termina acţiunea şi „Space Bar" pentru a introduce următoarea linie de nivel.

Exemplul 3.2. Să se reprezinte grafic şi să se eticheteze liniile de nivel ale funcţiei z=x *exp(x^2-y^2) în domeniul: $-2 \le x \le 2$, $-2 \le y \le 3$ Cu secvenţa MATLAB: [x, y] = meshdom(-2:.2:2, -2:.2:3); z= x.*exp(-x.^2-y.^2); C=contour (-2: .2 :2,-2: .2:3, z, 8) clabel(C);

3.1.4. Reprezentarea grafică în spaţiu a liniilor de contur

Reprezentarea grafică în spaţiu a liniilor de nivel constant se realizează cu funcţia ***contour3***;

se apelează cu una dintre sintaxele:

contour3(Z) - realizează reprezentarea 3D a liniilor de contur ale matricei Z;

contour3(Z,N) - realizează reprezentarea 3D a N linii de contur ale matricei Z;

contour3(X,Y,Z) şi

contour3(X,Y,Z,N) - utilizează matricele X si Y pentru a defini limitele axelor.

Vectorii X şi Y trebuie să fie monoton crescători si cu pas constant.

Exemplul 3.3. Să se reprezinte grafic în 3D liniile de contur ale funcţiei predefinite ***peaks***. Cu secvenţa MATLAB;

x=-3:.125:3; y=x; [X, Y]=meshgrid (x, y) ; Z=peaks(X,Y); contour3(X,Y,Z,20)

3.1.5. Reprezentarea grafică a câmpurilor de vectori orientaţi

Reprezentarea grafică a unui câmp de vectori orientaţi foloseşte funcţia ***quiver***, se apelează cu una dintre sintaxele :

quiver(X,Y,DX,DY) - reprezintă mici segmente de dreaptă cu săgeţi (vectori) având originea la perechile de elemente (X,Y). Fiecare pereche deelemente din matricele DX si DY sunt proiecţiile vectorului pe axele Ox şi Oy;

quiver(DX,DY) - presupune implicit X=1:n şi Y=1:m. în acest caz DX şi DY sunt pe o reţea rectangulară;

quiver(x.,y,dx,dy,s) si

quiver(dx,dy,s) - controlează lungimea săgeţilor prin factorul de scală s.

Culorile şi tipurile de linii care se pot folosi sunt cele precizate la funcţia ***plot***.

Exemplul 3.4. Să se reprezinte grafic vectorii unei mişcări browniene. Cu secvenţa MATLAB: xg = -2:.5:2; yg = -2:.5:2; dx=rand(length(xg),length(xg))-0.5; dy=rand (length (yg) , length (yg)) -0.5; quiver(xg,yg,dx,dy)

4. REPREZENTĂRI SPAŢIALE CU LINII

4.1. Reprezentarea liniilor în spaţiu

Reprezentarea liniilor în spaţiu se face cu funcţia ***plot3***, care se apelează cu una dintre sintaxele: plot3(x,y,z) - unde x, y şi z sunt vectori de aceeaşi dimensiune, reprezintă grafic o linie în spaţiul 3D, linie care trece prin punctele ale căror coordonate sunt tripletele (x,y,z);

plot3(X,Y,Z) - unde X, Y şi Z sunt matrice de aceleaşi dimensiuni, reprezintă grafic câte o linie în spaţiul 3D, pentru fiecare triplet al coloanelor matricelor [X(:,i),Y(:,i),Z(:,i)];

plot3(x,y1z,'linie-tip'), sau

plot5(x1,y1,z1,'linie-tip',x2,y2,z2,'linie-tip2',...) - realizează reprezentări grafice 3D multiple, în care se precizează tipurile de linii şi culorile acestora, ca în cazul funcţiei ***plot***.

Pentru o citire mai precisă pe grafic, reprezentarea se poate asocia cu funcţia grid.

Exemplul 4.1. Să se reprezinte grafic o spirală în 3D. Cu secvenţa MATLAB: t=0:pi/50:10*pi; plot3(sin(t), cos(t),t)

4.2. Reprezentarea grafică spaţială a poliedrelor

Reprezentarea grafică în spaţiu a poliedrelor se face cu funcţia ***fill3***, care se apelează cu una dintre sintaxele:fill3(x.y.z,c) fill3(x,y.z,'c') fill(x1,y1,z1,c1,x2,y2,z2,c2..)

Funcţia fill3(x,y,z,c) reprezintă un poliedru 3D, având vârfurile definite de vectorii x, y si z, cu nuanţele de culoare precizate de c. Coordonatele poliedrului sunt specificate prin tripletele (x,y,z).

Dacă c este un singur caracter dintre cele prezentate în lista de culori de la funcţia plot: r, g, b, c, m, y, w, m, k sau un vector cu trei componente [r g b], poliedrul va fi colorat cu o singură culoare. Dacă c este un vector cu aceeaşi lungime ca x şi y, elementele acestuia sunt scalate cu funcţia caxis şi apoi utilizate ca indici în matricea de culoare pentru a specifica culorile vârfurilor. Culorile dintre vârfuri sunt obţinute prin interpolare biliniară a culorilor vârfurilor.

Dacă x, y si z sunt matrice de aceeaşi dimensiune, fill3(x,y,z,c) reprezintă câte un poliedru pentru fiecare coloană, în acest caz, c este un vector linie pentru poliedre reprezentate cu o singură culoare şi este o matrice pentru poliedre realizate cu culori interpolate (efecte speciale).

Dacă numai unul din argumentele x, y sau z sunt matrice, celelalte fiind vectori coloană cu acelaşi număr de linii, vectorii coloană se extind la matrice cu aceleaşi dimensiuni, prin adăugarea unor coloane identice.

4.3. Reprezentări grafice spaţiale dinamice

Reprezentarea grafică a unui punct care urmăreşte realizarea grafică spaţială („cometă") se face cu funcţia comet3; se.apelează cu una dintre sintaxele:

comet3(z) comet3(x,y,z) comet3(x,y,z,p)

Apelată fără argumente, funcţia comet3 lansează demonstraţia cu acelaşi nume. Dacă se apelează cu unul sau cu trei argumente, se obţine o reprezentare grafică a lui z sau a tripletelor (x,y,z), urmărite de un marker („cometă"), la o distanţă de 0.1 din lungimea vectorului z. Apelarea cu patru argumente realizează acelaşi lucru, însă distanţa de urmărire este setată la valoarea p din lungimea totală a vectorului z,

Exemplul 4.2. Urmăriţi modul de lucru al funcţiei comet3, cu secvenţa MATLAB:

t = 0:1/l00:20;comet3{sin(t),cos(t),t)

4.4. Reprezentarea 3D a suprafeţelor şi liniilor de contur

Funcţiile MATLAB folosite pentru reprezentări 3D ale suprafeţelor si liniilor de contur sunt:

mesh Reprezintă grafic suprafeţe 3D sub forma unei „reţele" („mesh");

meshc Reprezintă grafic combinaţia suprafaţă 3D („mesh")/linii de contur (reprezentate sub suprafaţă);

meshz Reprezintă grafic suprafeţe 3D („mesh"), cu plan de referinţă la cota zero (piedestal);

surf Reprezintă grafic suprafeţe pline 3D;

surfc Reprezintă grafic combinaţia suprafaţă 3D (continuă)/linii de contur (reprezentate sub suprafaţă).

O suprafaţă este parametrizată prin două variabile independente, i şi j, care variază continuu în interiorul unui dreptunghi; spre exemplu 1< i < m şi 1< j < n. În aceste condiţii, fiecare punct este specificat prin trei funcţii, X(i,j), Y(i,j) şi Z(i,j). Dacă i şi j sunt numere întregi, atunci suprafaţa este definită în nodurile unei reţele rectangulare (grid), matricele având dimensiunea mxn. Dacă se doreşte

şi precizarea culorii suprafeţei, atunci mai este necesară încă o matrice, C(i,j), de aceeaşi dimensiune, mxn. Fiecare punct al suprafeţei rectangulare este conectat cu alte patru puncte vecine.

$$\begin{array}{ccccc} & & i-1,j & & \\ & & | & & \\ i,j-1 & - & i,j & - & i,j+1 \\ & & | & & \\ & & i+1,j & & \end{array}$$

Punctele de pe margini sunt conectate cu trei noduri, iar colţurile au numai două noduri vecine.

Culoarea suprafeţei poate fi specificată prin două metode diferite: specificând colţurile reţelei sau centrele acesteia. Dacă funcţia ***shading***, care configurează nuanţele, este iniţializată la „interp", C trebuie să aibă aceaşi dimensiune ca matriceie X,Y si Z; ea specifică culorile colţurilor, în interiorul zonei realizându-se o interpolare liniară, Dacă ***shading*** este iniţializată „faceted" (implicit) sau „flat", atunci C(i,j) specifică o culoare constantă în dreptunghiul respectiv:

$$\begin{array}{ccc} i,j & - & i,j+1 \\ | & C(i,j) & | \\ i+1,j & - & i+1,j+1 \end{array}$$

în acest caz C poate avea aceeaşi dimensiune cu X,Y şi Z, însă ultima linie şi coloană sunt ignorate, sau poate fi dimensionată cu o linie şi o coloană mai puţin decât acestea. Reprezentarea grafică 3D a suprafeţelor se poate face fie sub forma unei „reţele" („mesh"), fie sub forma suprafeţelor netede.

4.5. Reprezentarea suprafeţelor cu „mesh"

Reprezentarea suprafeţelor cu „mesh" se face folosind funcţiile: mesh, meshc şi meshz care se apelează cu sintaxele:

mesh(X, Y, Z, C) mesh(X, Y, Z) ' .

mesh(Z) mesh(Z,C)

meshc(...) meshz(...)

în cazul cel mai general funcţia ***mesh*** se apelează cu patru matrice ca parametri de intrare; reprezintă grafic suprafaţa Z(X,Y), cu culorile din matricea C. .

În cele mai multe aplicaţii X si Y sunt vectori. Aceştia trebuie să fie ordonaţi crescător şi cu pas constant, pentru a rezulta o figură corectă. Dacă argumentele X şi Y sunt omise, reprezentarea este făcută pe baza indicilor matricei Z.

În cazul în care matricea C este omisă, se consideră C=Z, astfel încât culoarea este proporţională cu înălţimea suprafeţei.

Poziţia din care este observată suprafaţa reprezentată grafic poate fi precizată cu funcţia ***view***. Gradarea

axelor este dată de intervalele X, Y şi Z, sau de setarea curentă a axelor, prin funcţiile axis sau axes. Culorile utilizate sunt cele determinate de C sau precizate prin funcţia caxis. Valorile scalei de culori sunt utilizate ca indici ai unui tabel de culori.

Funcţia meshc permite reprezentarea 3D a suprafeţelor, cu „mesh", la care se asociază liniile de contur, trasate ca proiecţii în planul bazei; se apelează cu aceleaşi argumente ca funcţia mesh.

Funcţia meshz permite reprezentarea 3D a suprafeţelor, cu „mesh", trasând în plus un plan de referinţă la valoarea minimă (piedestal); se apelează cu aceleaşi argumente ca funcţia mesh.

Funcţia mesh returnează un identificator spre obiectul suprafaţă.

Funcţiile axis, caxis, colormap, hold, shading şi view setează proprietăţile obiectelor figură, axe şi suprafaţă, care afectează suprafeţele „mesh" afişate.

Exemplul 4.3. Reprezentaţi grafic funcţia $Z=X \exp(-X^2-Y^2)$:

1. ca suprafaţă „mesh",

2. ca suprafaţă „mesh" asociată cu linii de contur si

3. ca suprafaţă „mesh" cu plan de referinţă. Cu secvenţa MATLAB: [X,Y]=meshgrid (-2:.2:2, 2:.2:2); Z=X.*exp(-X.^2-Y.^2) , subplot(221); mesh(X,Y);subplot(222); meshc (X, Y, Z);subplot(223); meshz(X,Y,Z)

4.6. Reprezentarea grafică a suprafeţelor netede

Funcţiile ***surf*** şi ***surfc*** reprezintă suprafeţe 3D, sau suprafeţe 3D asociate cu liniile de nivel proiectate pe planul bazei; se apelează cu una dintre sintaxele : surf(X.Y,Z,C) - reprezintă o suprafaţă descrisă de matricele X,Y şi Z, şi colorată cu elementele precizate în matricea C. În utilizări simple argumentele X şi Y pot fi vectori sau pot fi omise; argumentul C poate fi şi el omis;

Punctul din care este văzută o reprezentare 3D poate fi precizat prin funcţia ***view***. Axele sunt date de matricile X,Y şi Z sau setate cu funcţia axis. Culoarea este dată de matricea C sau este setată prin funcţia caxis. Valorile scalate ale culorilor sunt indici la matricea de culoare curentă.

surf(X,Y,Z) -consideră C=Z, astfel încât culoarea este proporţională cu înălţimea suprafeţei;

surf(y.,y,Z) si surf(x,y,Z,C) - realizează reprezentarea suprafeţei descrise de matricea Z (mxn), unde vectorul x are dimensiunea n, vectorul y are dimensiunea m, iar matricea C are dimensiunea m x n. în acest caz colţurile dreptunghiurilor care compun suprafaţa sunt tripletele (x(j), y(i), Z(i,j)). Prin urmare x este asociat numărului de coloane, iar y numărului de linii;

surf(Z) si surf(Z,C) - presupune x=1:n si y=1:m pentru reprezentarea grafică 3D a matricei Z, folosind eventual şi matricea de culoare C;

surfc(...) - este identică cu surf(...), exceptând liniile de nivel, care sunt reprezentate sub suprafaţă, pe acelaşi grafic.

Exemplul 4.4. Reprezentaţi o suprafaţă 3D asociată cu liniile de nivel. Cu secvenţa MATLAB:

[X,Y]=meshgrid(-3:.25:3); Z=peaks(X,Y); surfc(X,Y,Z)

4.7. Reprezentarea obiectelor spaţiale

Obiectele spaţiale predefinite în MATLAB sunt:

cylinder - generează un obiect cilindru;

sphere - generează un obiect sferă.

4.7.1. Reprezentarea grafică a obiectului cilindru

Funcţia cylinder generează un cilindru de rază R, cu cercul bazei aproximat din N puncte echidistante; se apelează cu sintaxa: [x,y ,z]=cylinder(R, N)

Funcţia returnează matricele cu 2x(N+1) elemente, care specifică vârfurile fiecărei suprafeţe rezultate din aproximarea cercurilor bazelor cu poligoane cu N laturi. Vectorul R are două elemente [R1 R2], care precizează raza obiectului la partea inferioară (R1) şi superioară (R2), prin aceasta fiind-posibilă construirea de conuri, trunchiuri de con, piramide, trunchiuri de piramide etc. Valoarea implicită este pentru R=[1 1], iar pentru N=20. Reprezentarea grafică se face cu funcţia. surface(x,y,z). Omiterea argumentelor de ieşire determină reprezentarea grafică a obiectului cilindru.

Exemplul 4.5. Să se reprezinte grafic un con de rază Rc1=0.5 si înălţime Hc=7.5 şi un trunchi de piramidă cu baza hexagon, cu cercurile circumscrise bazelor de raze Rp1=1 şi Rp2=0.5 şi înălţimea Hp=7.5. Cu secvenţa MATLAB: Rcl=0.5; Rc2=0; Hc=7.5; N=30; [xc,yc,zc]=cylinder([Rcl Rc2],N);zc=zc*Hc; view([-37.5 30]);surface(xc,yc,zc); grid

figure

Rpl=l; Rp2=0.5; Hp=7.5; N=6;

[xp,yp,zp]=cylinder([Rpl Rp2],N);zp=zp*Hp;

view{[-37.5 30]);surface(xp,yp,zp); grid

4.7.2. Reprezentarea grafică a obiectului sferă

Funcţia ***sphere*** generează coordonatele (x,y,z) ale sferei unitate, care pot fi utilizate cu funcţiile surf şi mesh; se apelează cu una dintre sintaxele:

[x,y,z]=sphere(n) - generează coordonatele sferei în trei matrice (n+1)x(n+1), care pot fi reprezentate grafic cu funcţia surf(x,y,z) sau mes/i(x,y,z).

sphere(n) - reprezintă grafic suprafaţa unei sfere. Implicit n=20.

Exemplul 4.6. Să se genereze şi să se reprezinte grafic o sferă. Cu secvenţa MATLAB:

[X,Y,Z]=sphere(20); mesh(X,Y,Z);grid

4.8. Poziţionarea observatorului faţă de obiect

Poziţionarea observatorului poate fi precizată cu funcţiile:

View specifică poziţia de vizualizare a unui grafic 3D;

hidden permite vizualizarea sau ascunderea suprafeţelor suprapuse.

4.8.1. Definirea poziţiei observatorului faţă de obiect

Poziţia observatorului faţă de un obiect 3D se precizează prin unghiul pe orizontală (numit azimut) şi unghiul pe verticală (numit elevaţie). Funcţia ***view*** este utilizată pentru vizualizarea unei reprezentări grafice spaţiale din diverse poziţii; se apelează cu una dintre sintaxele :

view(az,elv), şi

view([az,elv]) - stabileşte unghiul din care se vede reprezentarea grafică 3D. az este azimutul (sau unghiul în plan orizontal), iar elv este unghiul pe verticală (ambele în grade). Azimutul poate lua valori pozitive sau negative, cea pozitivă fiind o rotire în jurul axei Oz în sens orar. Valorile pozitive ale unghiului pe verticală corespund unui punct de observare plasat deasupra planului (x,y), iar valorile negative corespund punctelor de sub plan;

view([x,y,z]) - stabileşte unghiul punctului de observare în coordonate carteziene;

[az,elv]=view - returnează unghiul pe orizontală si unghiul pe verticală din care este observată o reprezentare grafică 3D;

view(2) - stabileşte reprezentarea grafică la 2D cu valorile simplificate: az=0, elv=90;

view(3) - stabileşte reprezentarea grafică 3D cu valorile implicite: az=-37.5, elv=30.

Pentru orientare:

elv=90 - vederea de deasupra obiectului;

az=elv=0 - vedere directă din pianul zero;

az=180 - vedere din spate a obiectului.

4.8.2. Vizualizarea şi ascunderea suprafeţelor suprapuse

Funcţia ***hidden*** elimină liniile ascunse ale unei reţele de tip „mesh"; se apelează cu una dintre sintaxele:
hidden on - setează modul de reprezentare astfel încât liniile acoperite ale reţelei de tip „mesh" să fie ascunse. Acesta este modul implicit;

hidden off - setează modul de reprezentare astfel încât liniile acoperite ale reţelei de tip „mesh" să fie vizibile.

5. CREAREA ŞI CONTROLUL AXELOR

în acest capitol se explică următoarele funcţii MATLAB:

subplot Creează reprezentări grafice la poziţii fixe (împarte ecranul în mai multe ferestre grafice folosite pentru reperezentări multiple);

axes Creează axe la poziţii impuse;

gca Returnează caracteristicile axelor figurii curente;

cla Şterge axele curente;

axis Stabileşte sau returnează caracteristicile axelor;

hold Comandă menţinerea (îngheţarea) graficului curent pe ecran;

ishold Returnează starea funcţiei „hold".

5.1. Divizarea ferestrei grafice

Funcţia ***subplot*** creează si controlează acoperirea ecranului cu ferestre grafice; se apelează cu una dintre sintaxele:

subplot(m,n,p) • subplot(h)

Funcţia subplot(m,n,p) împarte ecranul într-o matrice m x n, creează axele în subfereastra p şi returnează identificatorul de control al noilor axe. Argumentele m, n şi p trebuie să fie numere întregi în intervalul [1, 9]. Scalarul m precizează în câte subferestre se împarte fereastra pe verticală, iar n specifică împărţirea pe orizontală. Dacă fereastra grafică a fost împărţită anterior, se face numai specificarea ca subfereastră grafică curentă.

Cu sintaxa subplot(t}), unde h este un identificatorul de control al caracteristicilor axelor, se apelează o subfereastra în care urmează a se realiza o reprezentare grafică.

Comanda clf sau subplot(1,1,1) restaurează fereastra grafică iniţială, ştergând toate celelalte axe şi reprezentări.

Exemplul 5.1. Realizaţi reprezentarea funcţiei y=sin(x), atât în prima jumătate a ferestrei grafice, cât şi în ultima şesime a acesteia. Cu secvenţa MATLAB: y=sin(2*pi*(0:49)/20); subplot (2,2,1); plot (y); grid subplot(2,3,6); plot(y); grid

5.2. Suprapunerea succesiva a graficelor

Funţia ***hold*** „memorează" graficul curent şi adaugă următoarele reprezentări grafice peste acesta; se apelează cu una dintre sintaxele:

hold on - „memorează" graficul curent şi toate proprietăţile axelor, adăugând următoarele grafice peste cel curent;

hold off - returnează starea iniţială prin care fiecare comandă plot şterge graficul anterior şi proprietăţile axelor înainte de a desena unul nou.

Dacă în fereastra grafică există mai multe subferestre grafice, fiecare dintre acestea are o stare a funcţiei hold, stabilită ca proprietate a obiectelor figură sau axe.

Pentru a testa starea funcţiei hold se foloseşte funcţia ishold, care returnează 1 dacă hold este activă (suprapune reprezentările grafice) şi 0 dacă este dezactivată (fiecare nouă reprezentare o şterge pe cea anterioară); se apelează cu sintaxa: a=ishold

Dacă funcţia hold este activă, toate reprezentările grafice ulterioare se fac cu aceleaşi proprietăţi ale axelor.

5.3. Ştergerea axelor curente

Funcţia cla şterge axele curente; se apelează cu una dintre sintaxele:

cla - şterge toate obiectele (linii, texte, module obiect, suprafeţe şi imagini) din axele curente;

cla reset - iniţializează ia valorile implicite toate obiectele din axele curente, precum şi toate proprietăţile axelor, cu excepţia poziţiei.