Introducere în MATLAB

Grigore Stamatescu
grigore.stamatescu@upb.ro

MATLAB este unul dintre cele mai răspândite programe, în special în teoria reglării automate, pentru calculul ştiinţific şi numeric. Pe lângă calculul efectiv, MATLAB oferă şi posibilităţi de reprezentare grafică şi programarea propriilor rutine. Este vorba de un sistem interactiv cu matricea ca element fundamental. Denumirea MATLAB provine de la *Mat*rix *Lab*oratory. SIMULINK este un mediu de operare grafic, bazat pe MATLAB prin intermediul căruia pot fi modelate şi simulate sisteme complexe. Calculele matriceale necesare sunt realizate automat de MATLAB. Dezvoltatorul ambelor produse este firma The Math Works Inc., SUA. Scopul acestei lucrări este o familiarizare cu mediul de lucru MATLAB. Instrucţiunile de lucru sunt redactate pe baza versiunii MATLAB 7.11.0 (R2010b). În cazul în care lucraţi cu o versiune diferită pot apărea modificări în structura meniurilor şi a căsuţelor de dialog.

1 Primii paşi

Fereastra de comandă

După lansarea MATLAB apare spaţiul de lucru MATLAB (vezi 1). Acesta este compus din mai multe ferestre, dintre care fereastra de comenzi (**Matlab Command Window**) cu linia de comandă MATLAB $\gg$, ocupă cea mai mare suprafaţă. Fereastra de comenzi este utilizată la introducerea de variabile şi rularea comenzilor, funcţiilor şi rutinelor. Fiecare rând de comandă este terminat cu un Return $\hookleftarrow$ iar expresia este evaluată imediat. Exemplele prezentate în acest document încep după simbolul liniei de comandă. Comentariile, cele care au în faţă simbolul procent %, nu trebuie introduse. Încercaţi să rulaţi următoarele exemple:

$\gg 5 * 7$ % Atribuirea variabilei standard **ans**
$\gg a = 3 * pi$ % Atribuirea variabilei **a**
$\gg b = a/5$
$\gg c = a * b$
$\gg b = 1 : 3 : 20$
$\gg$ plot(b)
$\gg$ help cos
$\gg b = 2\cos(b)$

Ultimul rând produce unul dintre cele mai des întâlnite mesaje de eroare: **missing operator, comma, or semi-colon**. În acest caz lipseşte operatorul $*$ pentru înmulţirea dintre 2 şi cosinus. Comanda corectă are forma: $\gg b = 2*\cos(b)$
Afişarea rezultatului pe ecran poate fi suprimată print adăugarea unui ; la sfârşitul rândului. Testaţi această funcţionalitate prin repetarea comenzii cu ;.

În locul reintroducerii comenzii de la tastatură, poate fi folosită şi săgeata sus $\uparrow$, prin care se pot accesa rapid comenzile anterioare. Aceste comenzi pot fi editate înainte de apăsarea tastei Return $\hookleftarrow$. Se poate opera şi mai rapid atunci când este cunoscută prima literă a comenzii căutate. La introducerea acesteia în linia de comandă, urmată de apăsarea tastei $\hookleftarrow$, vor fi afişate numai comenzile care încep cu această literă.

Ce comandă este afişată la introducerea **p** $\uparrow$?

Paleta de simboluri a ferestrei de comenzi

Paleta de simboluri (vezi fig. 1) permite accesul rapid la funcţionalităţi des utilizate ale mediului de dezvoltare. Semnificaţia simbolurilor individuale este descrisă de la stânga la dreapta:

- Pornirea editorului MATLAB într-o fereastră nouă

  Această operaţie este similară cu selectarea din meniu a **File:New:M.File**. Editorul precum şi realizarea respectiv modificarea unor rutine proprii, vor fi prezentate ulterior.

- Deschiderea unui document MATLAB existen în editorul MATLAB

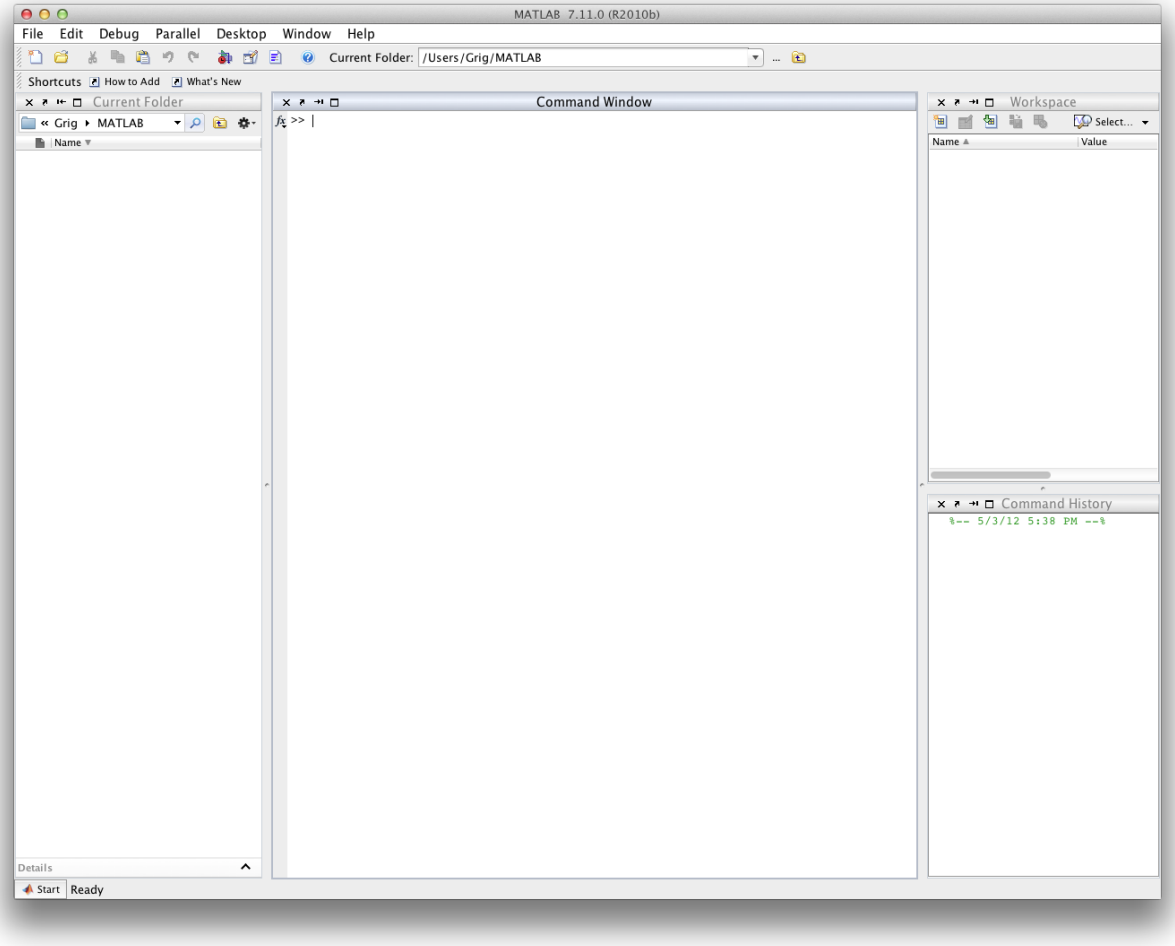

Figure 1: Spaţiul de lucru MATLAB

Este afişată o selecţie de fişiere cu extensiile: **.m** pentru rutine MATLAB, **.mdl** pentru modele SIMULINK sau **.fig** pentru figuri MATLAB. Editorul permite însă şi modificarea oricăror fişiere ASCII.

- Decupare, copiere şi lipire
- *Undo* şi *Redo*
- Lansare SIMULINK
- Lansare GUIDE

  Un editor de interfeţe grafice cu utilizatorul
- Lansare Profiler

  pentru optimizarea fişierelor M
- Deschiderea ferestrei de ajutor

  Fereastra de ajutor conţine acelaşi text referitor la o funcţie arbitrară *fname* precum cel care apare la introducerea **help *fname*** în linia de comandă. Textul nu apare însă în linia de comandă ci într-o fereastră dedicată, ceea ce permite navigarea uşoară între diferite comenzi.

- Calea actuală

  Aici sunt căutate fişierele necesare şi sunt salvate rezultatele operării. Dacă o căutare nu are succes în calea actuală este continutaă la celelalte adrese din calea de căutare. Lista de căi poate fi afişată şi modificată din meniul **File - Set Path**. În acest context, sunt date o serie de comenzi utile pentru operarea cu directoare:

  ≫ cd %Afişează calea actuală
  ≫ cd .. %Trecere în directorul superior
  ≫ cd work %Trecere în directorul inferior work
  ≫ dir %Listează toate fişierele din directorul actual
  ≫ what %Listează toate fişierele MATLAB/SIMULINK din directorul actual

Ferestre suplimentare

Toate variabilele utilizate sunt salvate automat în fereastra **Workspace**. Aici sunt afişate şi dimensiunile şi tipul acestor variabile. Printr-un dublu clic pe numele variabilei este afişat şi poate fi modificat conţinutul acesteia.

Fereastra **Current Directory** prezintă conţinutul directorului de lucru actual într-o structură arborescentă. Sunt posibile operaţiile uzuale de: deschidere, ştergere, redenumire şi copiere.

În fereastra **Command History** este afişată o listă a ultimelor comenzi introduse. Printr-un dublu clic pe o comandă aceasta este rulată din nou.

Ajutor

Se întâlnesc adesea situaţii în care este necesară descrierea şi, în special, sintaxa unei funcţii. Pentru aceasta este folosită comanda **help** (sau **doc** pentru detalii suplimentare) urmată de numele funcţiei:

≫ help cos
≫ help linspace
≫ help plot
≫ doc cos
≫ lookfor plot

Comanda **lookfor** realizează o căutare după cuvântul introdus în primele rânduri ale tuturor fişierelor din calea de căutare.

Există mai multe modalităţi care pot fi utilizate pentru a obţine ajutor referitor la introducerea comenzilor:

- Help Window

  O fereastră care prezintă textul de ajutor aferent unei funcţii (vezi paleta de simboluri). Prin dublu clic pe unul din elementele listei **Help Topics** poate fi parcurs un domeniu de interes. Fereastra de ajutor apare şi la introducerea comenzii **helpwin** în linia de comandă.

- Help Desk

  Fereastra Help Desk conţine fişiere în format .html sau .pdf cu informaţii complete despre fiecare funcţie. Aceasta apare şi la introducerea comenzii **helpdesk** în linia de comandă.

- Exemple şi demonstraţii

  Fereastra de demonstraţii oferă o listă de exemple detaliate incluse cu mediul de dezvoltare MATLAB. Aceasta apare şi la introducerea comenzii **demo** în linia de comandă.

Exerciţii

1.1. Modificaţi calea de căutare prin adăugarea unui director de lucru personalizat de forma **Nume_Grupa**.

1.2. Descrieţi pe scurt funcţionalitatea şi sintaxa comenzilor: **exp**, **grid**, **figure** şi **hold**.

1.3. Prezentaţi funcţionalitatea comenzilor: **which**, **pause** şi **disp**.

1.4. Afişaţi cu ajutorul comenzii **linspace** o serie de 22 de numere de la 1.5 la 9.

1.5. Deschideţi un fişier nou în editor şi introduceţi comenzile:

```
% Aceasta este prima incercare!
cd
disp('Salut!')
a=5*7
b=5*7;
```

Salvaţi fişierul în directorul de lucru cu numele **exemplu1.m**. Închideţi editorul. Introduceţi în linia de comandă comanda **exemplu1** şi **help exemplu1.m**. Care este funcţionalitatea noii comenzi?

Accesaţi **Current Directory**. Deschideţi fişierul exemplu1.m prin dublu clic şi apăsaţi tasta **F5**. (Astfel fişierul este salvat şi rulat imediat.)

2 Constante şi Variabile

Constante

MATLAB a fost dezvoltat special pentru calculul matricial. O matrice reprezintă pentru MATLAB un câmp cu valori numerice. Cuvintele (*strings*) sunt câmpuri numerice, pentru care elementele conţin codurile ASCII ale caracterelor (valori întregi, pozitive). MATLAB cunoaşte şi alte metode pentru salvarea datelor alfanumerice, în această etapă vom rămâne însă la reprezentarea matricilor numerice.

Constantele (scalarii) sunt reprezentate intern ca matrice 1x1. Vectorii sunt matrice 1xn (vectori linie) sau nx1 (vectori coloană). În cazul calculelor matriciale (e.g. înmulţirea) trebuie ţinut întotdeauna cont de ce se doreşte a fi calculat şi ce tip de vectori sunt folosiţi. Altfel vor fi obţinute mesaje de eroare sau rezultate care nu corespund cu intenţia iniţială.

Matricile cu mai mult de un element sunt introduse pe linii, încadrate în paranteze pătrate []. În interiorul unei matrici se folosesc virgule sau spaţii goale pentru separarea coloanelor şi ; pentru separarea liniilor:

$\gg$ 5
$\gg$ 3.14159
$\gg$ 2.0e-10
$\gg$ [1,2,3] % [1 2 3] Vector linie

$\gg$ [4;5;6] % $\begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$ Vector coloană

$\gg$ A=[1,2,3;4,5,6;7,8,9] % $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ Matrice

$\gg$ b=[1,2,3]

$\gg$ c='test'

Vectorii pot fi introduşi şi, mai rapid, în următoarea formă:

vector = [primul element:increment:ultimul element];

Vectorul astfel generat va conţine elemente echidistante. Incrementul poate fi şi negativ sau, în cazul în care este 1, poate fi omis.

≫ [1:10]

≫ [1:-1.5:-8]

Pentru a genera vectori, pot fi folosite şi funcţiile **linspace** şi **logspace**. linspace este cunoscută din secţiunea precedentă. Cum se diferenţiază logspace de linspace?

Variabile

Precum în orice alt limbaj de programare, MATLAB permite definirea de variabile. Atribuirea se face prin semnul egal =. Fiecare şir de caractere, ce începe cu o literă şi nu include simboluri speciale, poate fi folosit ca nume de variabilă. Este făcută diferenţa între litere mari şi litere mici. MATLAB nu necesită declarări iniţiale; o variabilă este creată automat atunci când este plasată în partea din dreapta a unei alocări. În partea din dreapta pot fi utilizate numai variabile sau constante create anterior. Există posibilitatea alocărilor recursive.

Variabilele MATLAB se bazează pe matrici. Acestea pot conţine atât cifre cât şi litere. Pe lângă cifre şi litere, mai există **celule** şi **structuri**.

Analog ferestrei **Workspace**, prin comanda **who** + numele variabilei utilizate şi **whos** + numele şi dimensiunea variabilei utilizate, este afişat conţinutul acesteia în fereastra de comenzi.

≫ b∗A

≫ d=2∗A

≫ d

≫ who

≫ whos

Suplimentar, există un număr de variabile de sistem rezervate, a căror modificare nu este permisă:

- **ans** conţine rezultatul ultimei comenzi dacă nu a fost făcută nicio alocare către o altă variabilă.
- **pi** conţine numărul π.
- **eps** conţine numărul $2^{-52} = 2.22e - 16$.
- **realmin, realmax** sunt cele mai mici, respectiv cele mai mari valori reale disponibile în MATLAB.
- **Inf** infinit, rezultatul operaţiei $1/0$
- **NaN** (Not a Number) rezultatul operaţiei $0/0$
- **i,j** conţin ambele unitatea imaginară $\sqrt{-1}$ pentru introducerea de numere complexe.

În cazul în care o variabilă de sistem *vname* este suprascrisă cu o altă valoare, aceasta poate fi resetată la valoarea originală prin comanda **clear *vname***, care şterge variabila redefinită din spaţiul de lucru.

≫ pi=2.3

≫ clear pi

≫ pi

≫ complex=$3 + 2 * i$

Dacă se doreşte accesul la valori individuale ale elementelor unei matrici, numărul liniei şi al coloanei corespunzătoare trebuie să fie puse între paranteze rotunde. Numărul liniei şi al coloanei pot fi şi vectori, pentru a accesa simultan mai multe linii sau coloane.

≫ A(1,3)

≫ A(1,:) % ':' reprezintă linii complete

≫ A(:,2) % ':' reprezintă coloane complete

≫ A([1 3],1)

Matricile logice ajută la extragerea de valori care îndeplinesc o serie de condiţii, dintr-o matrice A. Suplimentar, matricea logică trebuie să aibă aceleaşi dimensiuni şi constrângeri, e.g. valori între 2 şi 5. Ulterior, aceste valori pot fi extrase. Exemplul următor ilustrează modul de lucru:

≫ X=(A>=2&A<=5)

≫ A(X)

Cu ajutorul unei matrici vide [] se pot şterge linii sau coloane dintr-o altă matrice:

≫ A(:,2)=[] Ştergerea celei de-a doua coloane a matricii A.

Încărcare, salvare şi ştergere

Următoarele comenzi servesc la administrarea variabilelor din spaţiu de lucru şi punerea lor la dispoziţia utilizatorului pentru sesiuni ulterioare. Acestea sunt:

- **clear**: şterge toate variabilele din spaţiul de lucru
- **clear *variable***: şterge *variable* din spaţiul de lucru
- **save**: salvează toate variabilele în fişierul 'matlab.mat'
- **save *fname***: salvează toate variabilele în fişierul 'fname.mat'
- **save *fname* x y z**: salvează variabilele x, y şi z în fişierul 'fname.mat'
- **load**: încarcă toate variabilele din fişierul 'matlab.mat' în spaţiul de lucru
- **load *fname***: încarcă toate variabilele din fişierul 'fname.mat' în spaţiul de lucru
- **load *fname* x y z**: încarcă variabilele x, y şi z din fişierul 'fname.mat' în spaţiul de lucru

Exerciţii

2.1. Creaţi două variabile **t**=[2 4 6 ... 14], **y**= $\begin{bmatrix} 1 & 5 & 7 \\ 2 & 5 & \pi \end{bmatrix}$ şi modificaţi-le conţinutul printr-un dublu clic pe numele variabilei din fereastra corespunzătoare din spaţiul de lucru. Verificaţi modificarea prin introducerea numelui variabilei în linia de comandă.

2.2. Modificaţi numele lui **y** în **ymat**.

2.3. Creaţi un vector linie **t1**, cu valori de la 0 la 1 şi incrementul 0.01.

2.4. Creaţi un vector coloană **t2** cu o distribuţie logaritmică şi valori de la 10^{-3} la 10^{0}.

2.5. Extrageţi prima linie **L1** şi cea de-a treia coloană **C3** din **ymat**.

2.6. Extrageţi matricea **ymic**, care conţine prima şi cea de-a treia coloană din **ymat**.

2.7. Extrageţi vectorul **interm** cu valorile mai mari de 3 din matricea **ymat**.

2.8. Ştergeţi cea de-a doua linie din **ymat**.

2.9. Salvaţi toate variabilele într-un fişier şi ştergeţi spaţiul de lucru prin comanda **clear all**. Verificaţi conţinutul spaţiului de lucru (e.g. **who**). Încărcaţi variabilele din fişier înapoi în spaţiul de lucru.

3 Operatori şi funcţii

Operatori

Partea din dreapta a unei expresii MATLAB poate conţine o formulă algebrică complicată. Constantele şi variabilele cunoscute trebuie conectate logic prin intermediul operatorilor. În cazul operatorilor aritmetici, pe lângă cei comuni (+,-,*,/), mai sunt folosiţi şi ridicarea la putere ($\wedge$) şi transpunerea (’). Suplimentar, există şi operatorii logici: negare ~, şi &, sau |, sau exclusiv xor() şi operatorii relativi: egalitate ==, inegalitate $\tilde{=}$, $<$, $\leq$, $>$, $\geq$. Prin următoarea comandă este afişată o imagine de ansamblu a operatorilor:

$\gg$ help ops

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

$\gg$ A=[1,2,3;4,5,6;7,8,0]

$\gg$ b % Vector linie

$\gg$ b=b’ % Vector coloană

$\gg$ b+b % Sumă de vectori

$\gg$ b’$*$b % Produs scalar

$\gg$ b$*$b % Eroare datorată dimensiunilor incompatibile

$\gg$ b.$*$b % Multiplicare la nivel de element

$\gg$ 2$*$A % Înmulţire cu un scalar

$\gg$ A$*$A % A ori A şi

$\gg$ A$\wedge$2 %A la pătrat, sunt identice

$\gg$ A.$\wedge$2 % dar nu şi în acest caz!

Funcţii

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

$\gg$ help elfun

$\gg$ cos(0)

$\gg$ 4$*$ atan(1) % pi, $atan(x) = tan^{-1}(x)$

$\gg$ bexp=exp(b) % Funcţia exponenţială , aplicată fiecărui element din b

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

$\gg$ A=[1,2,3;4,5,6]

$\gg$ size(A)

$\gg$ length(A)

$\gg$ size(b)

$\gg$ length(b)

Comanda **help elmat** afişează o imagine de ansamblu asupra matricilor standard şi a funcţiilor de manipulare ale matricilor precum şi asupra funcţiilor din domeniul algebrei liniare **help matfun**. Funcţiile **ones** şi **zeros**, de exemplu, generează matrici numai cu elemente 1 resp. 0 şi funcţia **eye** generează matricea unitate.

≫ help elmat

≫ D=ones(3,4)

≫ E=zeros(5)

≫ F=eye(4)

≫ G=eye(4,3)

≫ help matfun

Exerciţii

3.1. Determinaţi numărul de linii şi de coloane ale matricilor D, E, F şi G.

4 Reprezentări grafice

MATLAB permite realizarea unei game variante de reprezentări grafice ale datelor. Cele mai importante funcţii 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 foloseşte comanda **plot(y)**. În acest caz, elementele lui **y** (y_1, y_2, y_3,...) sunt reprezentate bidimensional împreună cu indicii lor (1,2,3,...). Dacă se doreşte utilizarea unei alte axe x, de exemplu timpul t, acest vector t trebuie adăgat ca parametru în apelarea funcţiei, **plot(t,y)**. Comanda corespunzătoare pentru reprezentarea tridimensională conţine toţi 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 comparaţie

≫ plot(y1,t) % Pentru comparaţie

≫ plot3(t,y1,y2)

Graficele sunt reprezentate în figuri/**figures**. Acestea reprezintă ferestre individuale care, pe lângă curbe, includ şi axele şi diferite meniuri pentru modificarea reprezentării grafice. Dacă există o fereastră de figură deja deschisă, curba va fi reprezentată în aceasta şi vechea curbă va fi ştearsă. Cu **hold** sunt păstrate vechile curbe şi 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 şi 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 şir de caractere, e.g. **plot(t,y1,'g')**. În acest şir de caractere sunt incluse informaţii despre culoare, reprezentarea liniilor şi a punctelor de date. Pentru o descriere completă a opţiunilor disponibile, apelaţi descrierea funcţiilor prin **help plot**, **help graph2d**, **help graph3d**.

O fereastră de figură poate fi, similar cu o matrice, împărţită 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 şi o coloană şi alege primul element (numărarea este făcută de la stânga la dreapta şi de sus în jos) pentru reprezentarea primei curbe

≫ subplot(2,1,2); plot(y1,t) % subplot foloseşte aceeaşi partajare, alege însă a doua fereastră

Pentru inscripţionare, există următoarele funcţii, ce au efect asupra graficului curent:

≫ title('Grafic sugestiv') % inscripţionează titlul graficului

≫ xlabel('Timp [s]') % inscripţionează axa x

≫ ylabel('Distanţă [m]') % inscripţionează axa y

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

Alte tipuri de grafice

Funcţiile **stem** respectiv **stem3** şi **stairs** sunt adecvate pentru reprezentarea semnalelor discrete. Fiecare punct este reprezentat printr-o tijă verticală cu un cerc la capăt. Stairs este funcţia treaptă şi uneşte punctele individuale precum o scară, valoarea punctul anterior fiind păstrată până la următoarul.

fplot calculează într-un interval dat valorile şi evoluţia unei funcţii, fără a cunoaşte numărul exact de puncte.

≫ stem(y1)

≫ stairs(t,y1)

≫ fplot('3∗sin(r).∗exp(-r/(pi/4))',[0,2∗pi]);

Reprezentarea suprafeţelor

Pentru reprezentarea suprafeţelor sunt folosite funcţiile *mesh* şi $surface$:

- **mesh** desenează o reţea colorată, ale cărei noduri sunt punctele funcţiei reprezentate.
- **surf** colorează suplimentar şi textura plasei

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

Prin intermediul unui exemplu este clarificat cum pot fi generaţi vectorii necesari. Dorim să reprezentăm suprafaţa funcţiei $z = -2x + 3y$, unde x şi y iau următoarele valori:

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

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

Deoarece x conţine 4 valori iar y 6 valori, z va lua 6X4=24 de valori. Suplimentar, deoarece fiecare punct este reprezentat printr-o tripletă (x_i, y_i, z_i), trebuie generate matricile **X** şi **Y** care conţin în fiecare linie, respectiv fiecare coloană, valorile lui x şi y. Aceasta se realizează cel mai simplu prin comanda **meshgrid**. Ulterior este calculat z şi este desenată suprafaţa:

≫ x=[-1:2]; y=[0:0.1:0.5] % Definirea vectorilor x şi y

≫ [X,Y]=meshgrid(x,y); % Generarea matricilor **X** şi **Y**

$\gg$ Z=-2.∗X+3.∗Y; % Calculul matricii **Z**

$\gg$ mesh(X,Y,Z) % sau surf(X,Y,Z)

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

Exerciţii

4.1. Generaţi matricea **graphA**= $\begin{bmatrix} 1 & 2 & 3 & ... & 6 \\ 1 & 4 & 9 & ... & 36 \\ 1 & 8 & 27 & ... & 216 \end{bmatrix}^T$ şi reprezentaţi-o grafic. Introduceţi în grafic o legendă cu textele 'liniar', 'pătratic' şi 'cubic'. Activaţi rasterul de fundal. Setaţi prin meniul **Edit: Axes Properties** scalarea axei y ca logaritmică. Daţi graficului titlul 'graphA'.

4.2. Reprezentaţi grafic funcţia $c = y * sin(x)$ pentru x de la -10 la 10 (increment 1) şi y de la 0 la 30 (increment 3) şi setaţi suprafaţa de tip plasă.

Referinţe

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