

Inițiere în limbajul MATLAB

Ing. Ioan Tăbuș

Institutul Politehnic București

1. Introducere

MATLAB este un mediu de programare având ca nucleu proceduri de transformare a matricilor, care sînt elementele de bază ale limbajului. Modul de lucru este interactiv, sistemul interpretînd fiecare linie introdusă de utilizator de la consolă, dar există atît posibilitatea extinderii setului de comenzi admisibile indefinit (limitată practic numai de spațiul disponibil pe disc), cît și posibilitatea de lansare în execuție a unor fișiere de comenzi.

Sintaxa limbajului este extrem de simplă și expresiile sînt foarte apropiate de limbajul matematic comun, MATLAB fiind foarte ușor de asimilat chiar de cei care nu au o pregătire specială în domeniul calculatoarelor. Simplitatea poate fi considerată o caracteristică a limbajului, dar numai la nivelul ușurinței de utilizare (similar BASIC-ului), pentru că la nivelul ofertei de proceduri, acest pachet de programe se caracterizează prin una din cele mai elaborate colecții de funcții, devenind imediat după apariția sa un standard în rîndul creatorilor și utilizatorilor de software.

În momentul de față MATLAB deține supremația în domeniul aplicațiilor în care algebra liniară are un rol important, cum ar fi: prelucrarea semnalelor, identificarea proceselor, conducerea sistemelor. Procedurile fundamentale din aceste domenii sînt grupate în "toolbox"-uri - pachete de instrumente -specifice fiecărui domeniu, în care se găsesc implementate principalele metode "clasice" (dezvoltate pînă acum trei-patru ani) și care se pot completa cu ușurință cu dezvoltările algoritmice de ultimă oră. Aceasta, deoarece în revistele de specialitate descrierile de algoritmi în pseudocod sînt completate din ce în ce mai des și cu implementarea sub forma de program MATLAB (pentru a pune în evidență care din formulările echivalente în MATLAB ale aceluiași algoritm are o mai bună comportare din punct de vedere numeric). Datorită faptului că introducerea noilor funcții este foarte simplă și pentru că mediul de operare este foarte potrivit testării noilor idei, foarte mulți creatori de software preferă să programeze în MATLAB și să folosească limbajele tradiționale FORTRAN, PASCAL, C doar la scrierea procedurilor pentru care se dorește o eficiență deosebită ca timp de calcul (eficiența în timpul execuției, dar obținută cu prețul unui timp de elaborare și testare a programelor foarte mare). Pentru a îmbina posibilitățile oferite atît de limbajele tradiționale cît și de MATLAB, sînt în curs de dezvoltare tehnici de linkare foarte simple ale

rutinelor FORTRAN sau C cu mediul MATLAB [2]. În cele ce urmează vor fi prezentate cîteva exemple din domenii diferite de aplicare a MATLAB-ului, pentru a ilustra simplitatea deosebită a scrierii (nu mai sînt necesare specificări de tip, alocări de memorie, transferul parametrilor la apelarea procedurilor este foarte natural) ca și forța deosebită dată de numărul mare de funcții disponibile (practic în cîteva linii de program putîndu-se rezolva probleme care tradițional ar necesita pagini întregi de text sursă). Pentru a pune mai bine în evidență modul de operare în acest mediu de programare, am tipărit în mod "bold" (întărit) caracterele introduse de operator de la consolă, lăsînd în format normal mesajele sistemului date la display. Liniile introduse de operator apar după "prompt"-ul MATLAB care este simbolul ».

Exemplul 1 (Rezolvarea sistemelor de ecuații)

```
» A = [ 1 2 0  
        3 4 5  
        6 7 8 ];  
» b = [ 1 2 3 ]';  
» x = A*b;  
» y = A\x
```

```
y =  
→ 1  
→ 2  
→ 3
```

În primele trei linii este definită matricea A, în a patra linie este definit vectorul "coloana" b (de notat că transpunerea unei matrici se realizează cu operatorul '), în linia a cincea se efectuează produsul dintre matricea A și vectorul b, iar în ultima linie se rezolvă sistemul de ecuații $Ay = x$, care în mod evident are ca soluție vectorul $y = b$.

Exemplul 2 (Calculul și trasarea caracteristicilor de frecvență ale unui sistem dinamic liniar)

```
» num = [ 0 0 1 ];  
» den = [ 1 0.05 1 ];  
» w = logspace (-1, 1);  
» [amp faza] = bode (num, den, w);  
» loglog (w, amp, '-w'), title ('Caract. amplitudine -  
pulsafie')  
» semilog x(w, faza, '-w'), title ('Caracteristica faza -  
pulsafie')
```

În primele două linii este definit sistemul de ordinul doi, cu funcția de transfer $H(s) = 1 / (s^2 + 0.05s + 1)$, în linia a treia se generează vectorul w al pulsațiilor în care va fi evaluat $H(j\omega)$ (50 de valori echidistante pe o scară logaritmică, pulsația minimă fiind $10^{-1} = 0.1$ și cea maximă $10^1 = 10$). În linia a patra este apelată funcția de calcul a valorilor complexe $H(j\omega)$, care vor fi returnate de către funcția "bode" în doi vectori: vectorul modulelor acestor numere complexe, notat "amp" și vectorul fazelor, notat "faza". Ultimele două linii

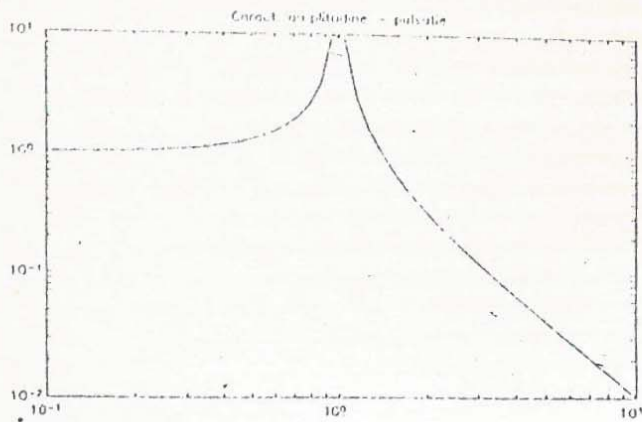


Fig. 1

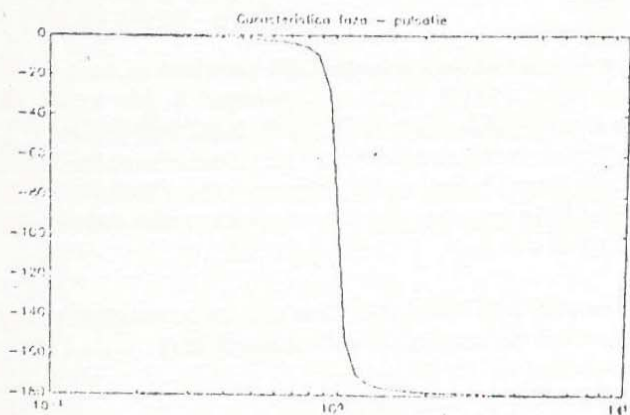


Fig.2

apelează procedurile de reprezentare grafică, prima în coordonate logaritmice pe ambele axe (așa cum este necesar la calculul caracteristicii amplitudine - pulsație) și a doua, în coordonate logaritmice pe axa Ox și liniare pe axa Oy. Rezultatele grafice furnizate de această secvență de program sînt prezentate în figurile 1 și 2.

Exemplul 3 (Calculul Spectrului de Putere al unui semnal simulat)

```

» w1 = 2 * pi / 20; w2 = 2 * pi / 40;
» t = (1: 256)';
» u = sin(w1*t) + sin(w2*t + pi/4) + (rand(t) - 0.5);

```

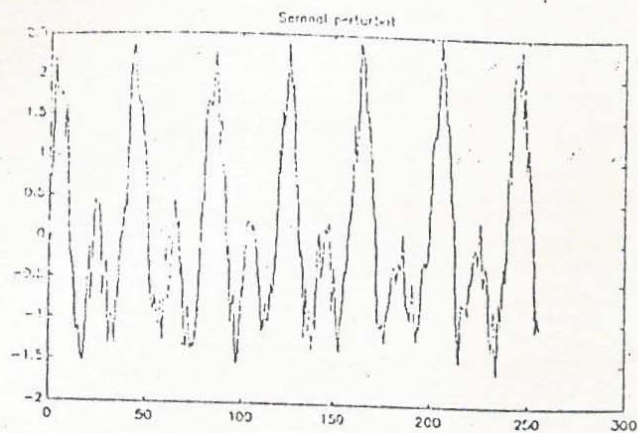


Fig. 3a

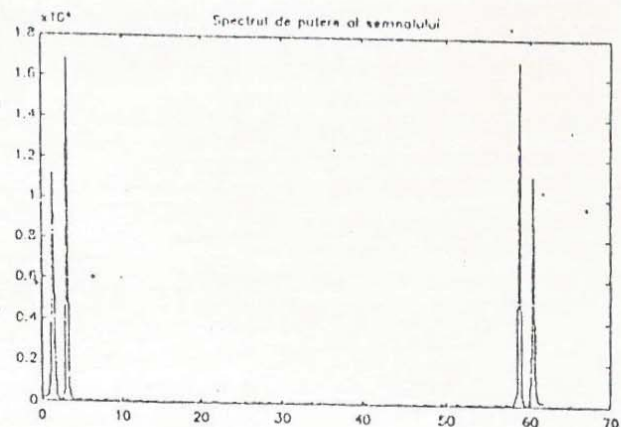


Fig. 3b

```

» plot(t,u,'-w'), title (' Semnal perturbat ')
» U = fft(u);

```

```

» Su = real(U).^ 2 + imag(U).^ 2;

```

```

» plot(2*pi/26*(0:255),Su,'-w')

```

```

» title (' Spectrul de putere al semnalului')

```

În liniile 1 și 2 se definesc două pulsații, w1 și w2 și un vector, t, reprezentînd momentele de timp la care este definit semnalul u; în linia a 3-a este simulat semnalul u ca fiind suma sinusoidelor de pulsații w1 și w2 și perturbat de un semnal aleator, uniform distribuit, de medie zero. În linia a 4-a semnalul simulat este plotat, iar în linia a 5-a se trece la calculul Transformatei Fourier Rapide, U (de dimensiune 256, prima putere a lui doi mai mare sau egală cu numărul de puncte disponibile ale semnalului u), utilizînd funcția fft; se calculează în linia a 6-a spectrul de putere estimat al semnalului Su, modulul la pătrat al valorilor transformatei U iar în linia a 7-a acest spectru este afișat. Rezultatele grafice obținute cu acest program sînt

plotate în figura a 3-a.

Exemplul 4 (Plotarea în perspectivă a unor suprafețe de nivel)

```
» x1 = -8:0.5:8 ;  
» [ x y ] = meshdom(x1,x1);  
» R=sqrt(x.^2 + y.^2) +eps;  
» Z = sin(R). / R  
» mesh(Z)
```

În acest exemplu este ilustrată posibilitatea generării de imagini tridimensionale în perspectivă, fiind arătată suprafața rezultată prin rotirea graficului funcției sinc în jurul axei Oz, pentru o regiune pătrată din planul Oxy. Graficul rezultat este prezentat în figura a 4-a.

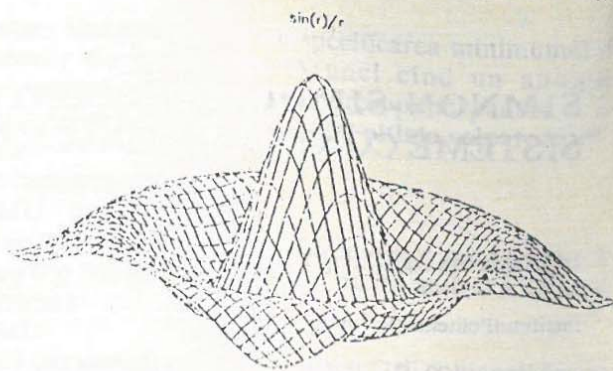


Fig. 4