

Memento kurzus: Matlab bevezető

Segédanyag: Molnár Tamás, átdolgozta: Hajdu Dávid

2020. október 2.

Tartalomjegyzék

1. Bevezetés	1
1.1. A Matlab kezelőfelülete	1
1.2. Matlab script futtatása	2
2. Alapvető parancsok	2
2.1. Alapműveletek, változók definiálása	2
2.2. Vektorok, mátrixok, tömbök	3
2.3. Függvények	5
2.4. Logikai műveletek	5
2.5. Ciklusok	6
2.6. Ábrázolás	6
3. Gyakorló feladat	7
4. Hasznos linkek	7

1. Bevezetés

A Matlab név a *Matrix Laboratory* kifejezés rövidítéséből ered. A Matlab egy olyan programrendszer, amely numerikus számítások elvégzésére és az eredmények megjelenítésére szolgál, miközben alapvetően mátrixok (tömbök) műveleteivel operál. A hatékony numerikus számítás mellett számos előnye és felhasználási lehetősége van, tulajdonképpen egy egyszerű, könnyen tanulható, script írásra alkalmas nyelv.

1.1. A Matlab kezelőfelülete

A Matlab kezelőfelületén a felső menüsávot leszámítva négy fontos ablak van, amit használni fogunk (ld. 1. ábra):

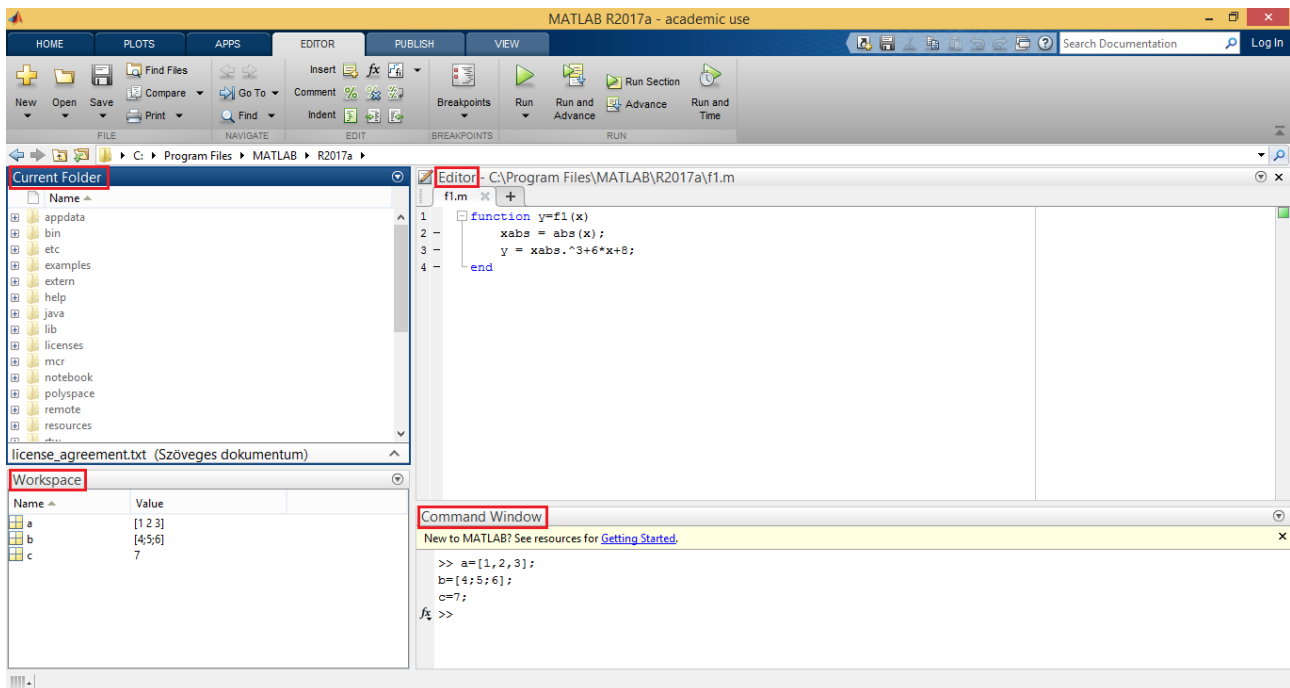
Command Window Parancsok közvetlen beírására és futtatására szolgál

Workspace A parancsok által definiált változók értéke a itt látható.

Editor Scriptfájlok szerkesztésére szolgáló beépített editor

Current Folder A "Working Directory" tartalmát mutató felület

Hosszabb kódot célszerű nem a Command Window-ban, hanem külön, egy .m kiterjesztésű fájlban írni, amelyet az Editor-ban szerkeszthetünk. E fájl elérési útját a Current Folder ablakban kell megadni.



1. ábra. A Matlab kezelőfelülete.

1.2. Matlab script futtatása

Az .m fájlban megírt Matlab script-et több féleképpen futtathatjuk:

- A teljes fájlt:
 - felső menüsorban az Editor fülön található Run gombbal (gyors gombja: **F5**)
 - a fájlnev Command Window-ba történő beírásával
 - a Profiler futtatása a az összes parancs futásidejét is mutatja
- Tetszőleges kódrészleteket ki lehet jelölni, és a jobb gombbal az Evaluate Selection opciót választva (gyors gombja: **F9**)
- A kódot dupla százalékjellel (%%) részekre tagolva, azokat külön futtatva jobb kattintás után az Evaluate Current Section opcióra kattintva (gyors gombja: **Ctrl+Enter**)

A futtatni nem kívánt kódrészeket ideiglenes "kikommentezhetjük", ha a sor elejére százalék jelet (%) teszünk. A futtatást megszakítani a Command Window-ba kattintva **Ctrl+C** billentyűkombinációval tudjuk.

Szerkesztés közben az Editor a jobb oldalon pirossal jelzi a szintaktikailag hibás sorokat (error) és a nem hatékony kódsorokra való figyelmeztetéseket (warning).

2. Alapvető parancsok

A Matlab részletes súgóval rendelkezik, melyet az F1 gombbal lehet előhívni – ez hasznos segítséget nyújthat bármilyen script írásakor. Egy adott parancs súgója közvetlenül is előhívható a parancs szövegében tetszőleges helyre kattintva F1 gombbal. **Ctrl+D** billentyűkombinációval a kívánt függvényen belül állva megnyithatjuk a matlab kódfájlt is. Egyéb segítséget és leírást kaphatunk a command window-ba való begépeléskor:

- `help sqrt` - rövid leírás az `sqrt` függvény használatához
- `doc sqrt` - részletes leírás az `sqrt` függvény használatához

2.1. Alapműveletek, változók definiálása

A legegyszerűbb műveletek működését közvetlenül a Command Window-ba írva is tesztelhetjük.

```
2+4
4*5
```

A legtöbb alapvető műveletre létezik beépített függvény, ezeket kis betűvel írva hívhatjuk meg, argumentumaikat kerek zárójelek közt megadva. Továbbá egyes változók (pl. i , π) előre definiáltak.

```
sqr(2)
sin(pi/4)
exp(i*pi)
```

Változók definiálásához az egyenlőség jelet használhatjuk. A változóknak komplex értéket is adhatunk, az imaginárius egységet egyaránt jelölhetjük i -vel és j -vel is és a szorzásjelet nem kötelező elé kitenni. Értékadás után a változók megjelennek a Workspace listájában és lehet velük további műveleteket végezni. A kódsort pontos vesszővel zárva a Matlab elvégzi a kívánt műveletet, de futtatás után nem jelenik meg a számítás eredménye a Command Window-ban.

```
a=3;
b=5;
a/b
c=2+4i;
d=4-3j;
```

A számítások elvégzése után a Workspace tartalmát egy .mat kiterjesztésű fájlba menthetjük ki a `save` paranccsal, melyet a `load` paranccsal lehet később visszaolvasni. A Workspace változók törlésére a `clear` illetve a `clear all` parancsok szolgálnak. Ezen kívül a Command Window tartalma is törölhető a `clc` paranccsal.

```
save('proba.mat')
clear a
clear all
load('proba.mat')
clc
```

2.2. Vektorok, mátrixok, tömbök

Matlab programban különös jelentősége van a vektorok, mátrixok, illetve tömbök megadásának. Vektorok és mátrixok létrehozhatók beépített paranccsal (pl. nullvektor, egységmátrix) vagy egyéb szabály szerint (pl. kettősponttal számtani sorozatból).

```
zeros(5)
zeros(5,1)
ones(1,7)
eye(4)
a=1:10
b=1:0.1:2
```

A vektorokat, mátrixokat elemenként is megadhatjuk. A Matlab megkülönböztet sor és az oszlopvektorokat, ezekre egy soros, illetve egy oszlopos mátrixként is tekinthetünk. A mátrixok sorait pontos vesszővel (vagy Enterrel sortöréssel) választjuk el, a soron belül az egyes elemeket (oszlopokat) vesszővel tagoljuk. E szabály szerint már meglévő mátrixokat is összefűzhetünk.

```
c=[1,5,7]
d=[4;-6;8]
e=[1,-2;-3,4]
f=[5,8
    7,6];
[e,f]
[e;f]
```

• 1. Feladat – ld. a feladatlapon •

A mátrixok adott indexű elemeire úgy hivatkozhatunk, hogy az indexeket zárójelben, vesszővel elválasztva adjuk meg. Egy adott indexhez tartozó összes elemet (pl. a mátrix egy teljes sorát vagy oszlopát) kettőspont segítségével választhatjuk ki. Több sor vagy oszlop kijelöléséhez az indexek vektorait is megadhatjuk zárójelben. Az utolsó elemet / sort / oszlopot az **end** szóval választhatjuk ki (az utolsó előtti pedig **end-1** kifejezéssel). Figyeljünk, hogy a C-kódokkal ellentétben a tömbök indexelése 1-gyel kezdődik, nem 0-val.

```
c(2)
A = rand(4)
A(1,1)
A(2,:)
A(:,3)
A(3:4,2:4)
A(:,end)
```

A Matlab értelmezi a mátrixok és a skalárok közötti műveleteket, ezeket a mátrix minden elemére egyesével elvégzi. Hasonlóan, a logikai műveleteket illetve az egyszerű beépített függvények műveleteit is elemenként végzi el. Logikai műveleteket akár a mátrixok indexeinél, adott feltételt teljesítő elemek kiválasztásakor is használhatunk.

```
c+1
abs(d)
d>0
d(d>0)
```

Mátrixok összeszorzására a ***** karakter szolgál, ez sorvektor és oszlopvektor összeszorzásakor skaláris szorzatot jelent. Ha két mátrix között elemenként szeretnénk műveletet végezni, akkor a műveleti jel elé pontot teszünk (pl. elemenkénti szorzás: **.***, elemenkénti osztás: **./**). Mátrixok transzponálására a **'** karakter szolgál, pont nélkül a **'** karakter konjugált transzponáltat jelent. A fontosabb mátrixműveletek (pl. sajátérték számítás, determináns számítás, invertálás) mind megtalálható a Matlabban beépített függvényként.

```
c*d
e*f
e.*f
g=[1-i,2+2i;0.5-5i,4+0.3i];
g.'
g'
cross(c,d)
h = [1,-3,5; 2,5,6; 7,-8,0];
eig(h)
det(h)
inv(h)
c/h
h\d
```

• 2. Feladat – ld. a feladatlapon •

• 3. Feladat – ld. a feladatlapon •

2.3. Függvények

Egy nagyobb feladat elvégzését célszerű részekre bontani, és az egyes rész feladatok végrehajtására külön függvényt írni.

A függvényeket megadhatjuk a **function** paranccsal, melyet **end** paranccsal zárunk. Ehhez célszerű a függvényt egy külön .m fájlban létrehozni (ugyanabban a mappában, ahol a többi .m fájl is elhelyezkedik), de a függvényeket egymáson belül, beágyazva is létrehozhatjuk (nested function). A függvény bemeneteit zárójelben felsoroljuk, kimenetét pedig egyenlőség jel elé írjuk.

```
% az f1.m fájlban
function y = f1(x)
    xabs = abs(x);
    y = xabs.^3+6*x+8;
end
```

Fontos, hogy a **function** és **end** sorok között (azaz a függvényen belül) definiált belső változók nem jelennek meg a Workspace listájában és nem kerülnek elmentésre, csak a függvény kimeneteit tárolja a Matlab. A függvényt ezután a nevével és zárójelben a bemeneteivel hívhatjuk meg.

```
% az eredeti .m fájlban
f1(-5)
f1(0:2:10)
```

A függvények visszatérési értékét elmenthetjük változóknak, többkimenetű függvény esetén változók vektorát adhatjuk meg (az elmenteni nem kívánt kimeneteket a ~ karakterrel elhagyhatjuk).

```
[cmax,maxindex] = max(c)
[~,maxindex] = max(c)
```

A **function** parancs használata helyett anonim függvény definiálása is lehetséges a @ karakterrel és zárójelben az argumentumaival.

```
f2 = @(x,y)x.^2+y.^2;
f2(1:5,7:11)
```

Sok bemenő paraméter esetén érdemes a paramétereket egy struktúrába rendezni és a struktúrát adni meg bemenetként a függvénynek. A struktúra egyes mezőire ponttal hivatkozhatunk.

```
param.a=1;
param.b=5;
param.c=-2;
f4 = @(t,param)par.a*sin(t)+par.b*cos(t)+par.c;
f4(pi/4,param)
```

2.4. Logikai műveletek

Logikai műveleteket az **if**, **elseif**, **else** parancsokkal végezhetünk, melyeket az **end** paranccsal zárunk.

```
ertek = 8;
if ertek>5
    disp('nagy')
elseif ertek>2
    disp('kozepes')
else
    disp('kicsi')
end
```

Hasonló esetszétválasztás a **switch**, **case**, **otherwise**, **end** parancsokkal is lehetséges.

2.5. Ciklusok

Ismételt feladatok elvégzésére használhatjuk a `for` és a `while` ciklusokat. Mindkét parancsot `end` sorral zárjuk, a `for` után egyenlőség jel mögé írjuk a futó változó lehetséges értékeinek vektorát, míg a `while` után azt a logikai feltételt írjuk, amelynek nem teljesülése a ciklus végét jelenti.

```
% a vektor elemeinek feltöltése egyenként
clear all
```

```
tic
for i = 1:1e7
    a(i)=i^2;
end
toc
```

```
tic
b=zeros(1,1e7);
for i = 1:1e7
    b(i)=i^2;
end
toc
```

A memória előzetes lefoglalásával lényegesen gyorsabb az adatok feltöltése. A Matlab az Editor oldalsó szélén a Warningok között figyelmeztet is erre.

- 4. Feladat – ld. a feladatlapon •
- 5. Feladat – ld. a feladatlapon •

2.6. Ábrázolás

Új ábrát a `figure` parancs segítségével hozhatunk létre. Egyváltozós skalár függvény ábrázolására a `plot` parancs szolgál, melynek argumentumában először az ábrázolandó pontok x koordinátáinak vektorát és y koordinátáinak vektorát adjuk meg, ezután következnek a megjelenítési beállítások (pl. vonaltípus és szín). Ha az y koordinátákhoz több soros mátrixot adunk meg, akkor a Matlab a mátrix egyes soraiból külön görbéket készít. Ábrasorozatot a `subplot` paranccsal készíthetünk, itt meg kell adni hányszor hány ábrát szeretnénk egymás mellett látni és hányadik az aktuálisan ábrázolni kívánt diagram. Ha egy meglévő ábrára további görbéket szeretnénk rajzolni, a `hold on` parancsot használhatjuk.

```
t=0:2*pi/100:2*pi;
figure
subplot(1,2,1)
plot(t,cos(t),'r')
hold on
plot(t,1./t,'b.-')
```

Számos további megjelenítési beállítás létezik (pl. tengelyfeliratok: `xlabel`, `ylabel`, ábracím: `title`, ábrázolás tartománya: `axis`, egyéb tengely beállítások: `set(gca,...)`).

```
xlabel('t')
ylabel('x')
title('Idofuggveny')
axis([0 2*pi -1.5 1.5])
set(gca,'FontSize',12)
```

Térbeli görbe a `plot3` paranccsal rajzolható.

```
subplot(1,2,2)
plot3(cos(2*t),sin(2*t),t)
```

Felület a `surf` paranccsal ábrázolható, annak szintvonalai pedig a `contour` paranccsal érhetők el.

```
x=-2:0.1:2;
y=-4:0.2:4;
[X,Y] = meshgrid(x,y);
Z=X.^2/4+Y.^2/9;
% Z(X,Y) felület abrazolasa
figure;
surf(X,Y,Z)
pbaspect([1 2 1])
figure;
contour(X,Y,Z,'ShowText','on')
pbaspect([1 2 1])
```

A létrehozott ábrák bezárhatók a `close all` paranccsal.

- 6. Feladat – ld. a feladatlapon •
- 7. Feladat – ld. a feladatlapon •

3. Gyakorló feladat

Készítsünk numerikus szimulációt egy matematikai inga mozgásáról! Az inga mozgását leíró nem-lineáris differenciálegyenlet:

$$\ddot{\varphi}(t) + \sin \varphi(t) = 0, \quad (1)$$

ahol φ az inga függőlegestől mért szöghelyzete radiánban. Az $\mathbf{y}(t) = [\varphi(t) \ \dot{\varphi}(t)]^T$ állapotvektor bevezetésével a mozgásegyenlet elsőrendű alakba írható:

$$\begin{bmatrix} \dot{y}_1(t) \\ \dot{y}_2(t) \end{bmatrix} = \begin{bmatrix} y_2(t) \\ -\sin y_1(t) \end{bmatrix}. \quad (2)$$

A mozgásegyenlet megoldását számítsuk ki a $t \in [0, 10\pi]$ időintervallumon a $\varphi(0) = 30^\circ$, $\dot{\varphi}(0) = 0$ (azaz $\mathbf{y}(0) = [\pi/6 \ 0]^T$) kezdeti feltétellel. A mozgásegyenlet megoldásához használjuk a beépített `ode45` függvényt! Az `ode45` alkalmazásához a súgóban találunk segítséget (gyorsgombja: F1). Ábrázoljuk a kapott $\varphi(t)$ függvényt!

A szimuláció ciklusos ismétlésével vizsgáljuk meg, hogy változik a lengések T periódusideje a $\varphi(0)$ kezdeti kitérítés függvényében! A szimulációk során legyen $\varphi(0) = 2^\circ, 4^\circ, \dots, 174^\circ, 176^\circ$.

4. Hasznos linkek

- Matlab tutorial: <https://www.mathworks.com/support/learn-with-matlab-tutorials.html>
- Matlab help: <https://www.mathworks.com/help/matlab/>