### A számok kiíratásának formátuma

Alapértelmezésben a Matlab négy tizedesjegy pontossággal írja ki az eredményeket, pl.

```
» x=2/3
x =
0.6667
```

A format paranccsal átállíthatjuk a kiíratás formátumát. Ha több tizedesjegyet szeretnénk látni, akkor használjuk a format long vagy format long e parancsot. A szokásos formátum a format short vagy a format parancsra visszaáll. Fontos, hogy mindez nem érinti a számítások pontosságát, azokat a Matlab alapértelmezésben duplapontosan végzi. Ha kevésbé szellősen akarunk írni a parancsablakban, akkor használjuk a format compact parancsot.

Vegyük észre, hogy eddig a Matlab minden értékadás után kiírta a változó értékét a képernyőre. Ez nem mindig kényelmes, különösen, ha több soros vagy oldalas kimenetről van szó. Elkerülésére csak annyit kell tennünk, hogy a parancs után pontosvesszőt (;) teszünk.

Ha kíváncsiak vagyunk egy változó értékére, akkor a prompt után egyszerűen írjuk be a nevét, pl.

```
» x
x = 0.6667
```

A parancsablakban egy sorba több utasítást is begépelhetünk, ha azokat vesszővel (,) vagy pontosvesszővel (;) választjuk el. A vessző használatakor az eredmény kiíródik a képernyőre, míg a pontosvessző használatakor nem. Pl.

» a=5; b=6, c=7 b = 6 c = 7

A Matlab természetesen fejben tartja az összes létrehozott változót, de mi erre nem biztos, hogy képesek vagyunk. A who illetve a whos utasítások segítségével kilistáztathatjuk az összes használatban lévő változót. (A whos utasítás részletesebb listát ad.) Adott változó törlésére a clear valtozonev, az összes változó törlésére pedig a clear all utasítás szolgál. Ezt a parancsot érdemes kiadnunk, ha új számolásba fogunk, nehogy egy változó régi értékével végezzük el a számolásokat. Ha munkánkat befejeztük, az eredményt tartalmazó változó(ka)t elmenthetjük későbbi felhasználás céljára. Erre a **save** utasítás szolgál. Például a

#### »save result a b

a result.mat állományba menti el az a és b változókat. A save utasítással elmentett változókat a load utasítással tölthetjük be. Itt a változók neveit nem kell megadni. A

#### »load result

a result.mat állományba tett változókat tölti be. Elmenthetjük a teljes munkaterület tartalmát is (összes létrehozott változónkat). Ekkor a **save** utasítás után nem írunk változónevet, csak az állomány nevét. Ha ez utóbbit is elhagyjuk, akkor változóink automatikusan a matlab.mat állományba kerülnek. (Hasonló eredményre vezet, ha a parancsablak File menüjének Save Workspace As parancsát választjuk.)

Arra is lehetőség van, hogy a képernyő teljes tartalmát elmentsük. Ehhez a munka megkezdésekor adjuk ki a diary on parancsot, a végén pedig gépeljük be:

# » diary fajlnev

ahol a fajlnev szabadon választott. Ennek hatására az aktuális könyvtárba kerül a parancsablak tartalma (a diary on kiadásától kezdve) fajlnev néven. Ha a naplókészítést fel akarjuk függeszteni, akkor gépeljük be a diary off parancsot, ha pedig újra akarjuk indítani, a diary on parancs használható. A létrejövő fájl txt kiterjesztésű, bármilyen szövegszerkesztővel megnyitható.

#### Egyéb utasítások

- A parancsablak tartalma a clc paranccsal törölhető.
- Éppen futó számítás a CTRL+C billentyűkombináció megnyomásával szakítható meg.
- Ha egy kifejezés nem fér ki egy sorba, akkor ... beírásával folytatható a következő sorban.

# A Matlab helpje

A Matlab online dokumentációja elérhető a Help menüből. Ezenkívül bármilyen létező parancsról felvilágosítást kaphatunk, ha a parancsablakba begépeljük:

#### » help utasitas

Ha nem tudjuk, mi a neve annak az utasításnak, amire szükségünk lenne, akkor a lookfor parancsot használjuk, amely kilistázza a szóba jöhető Matlab-utasításokat. Például tegyük fel, hogy szeretnénk invertálni egy mátrixot, de nem tudjuk, hogy a Matlabban milyen utasítás van erre. A

## » help inverse

nem ad ki semmit, mert inverse nevű utasítás nincs. A

## » lookfor inverse

parancsra viszont részletes listát kapunk rövid leírással azokról az utasításokról, amelyek inverz számításával kapcsolatosak. Ezek közül könnyen kiválaszthatjuk a mátrixinvertálásra használható parancsot.

# Kilépés a Matlabból

A Matlabból a quit parancs begépelésével vagy a File  $\rightarrow$  Exit MATLAB menüpont kiválasztásával léphetünk ki.

# Beépített függvények

A Matlab számos elemi függvényt ismer. Ezekről felvilágosítást kaphatunk a help elfun parancs beírásával. Speciális függvényekről a help specfun tájékoztat. A beépített függvények listája igen terjedelmes. Az olyan alapvető függvények, mint pl. sin(x), cos(x), tg(x), exp(x), ln(x) neve a Matlabban a sin, cos, tan, exp, log, ahol az argumentumba írhatunk számot, vektort vagy mátrixot (lásd később) is. A Matlab néhány nevezetes konstans értékét is ismeri, pl. pi a  $\pi$  szám, i vagy j a képzetes egység, Inf a végtelent jelenti, és létezik egy NaN (not a number) nevű konstans is, amit matematikailag értemezhetetlen művelet (pl. 0/0, Inf-Inf) beírására kapunk.

## Példák:

1. Számítsuk ki a Matlabban az  $y = e^{-a} sin(x) + 10\sqrt{y}$  kifejezés értékét a = 5, x = 2 és y = 8 esetén.

» a = 5; x = 2; y = 8; » y = exp(-a)\*sin(x)+10\*sqrt(y) y = 28.2904 2. Számítsuk ki a 142 természetes és tízes alapú logaritmusát!

A természetes alapú logaritmus kiszámítása:

```
» log(142)
ans =
4.9558
```

A tízes alapú logaritmus:

```
» log10(142)
ans =
2.1523
```

3. Számítsuk ki  $\sin(\frac{\pi}{4})$  ill.  $e^{10}$  értékét!

```
» sin(pi/4)
ans =
0.7071
```

```
» exp(10)
ans =
```

2.2026e+004

# Megjegyzések:

- Beépített függvényt csak egy kifejezés jobb oldalán használjunk. Ne adjunk értéket olyan nevű változónak, amely valamilyen beépített függvényt jelent.
- A képzetes egységre használatos i ill. j változó átdefiniálható, ezeket a betűket előszeretettel használják ciklusváltozóként is (lásd később). Az ebből adódó zűrzavar

elkerülésére javasolható, hogy ciklusváltozóként inkább a kettőzött ii ill. jj nevű változókat használjuk.

## Vektorok és mátrixok a Matlabban

A Matlab széleskörű használhatóságának titka, hogy vektorokkal és mátrixokkal ugyanolyan könnyen végezhetünk műveleteket, mint számokkal. Bármely változó értéke (komplex elemű) vektor ill. mátrix is lehet.

Egy sorvektor elemeit szögletes zárójelben, szóközzel vagy vesszővel elválasztva adhatjuk meg, pl. adjuk meg a következő v sorvektort: v = (1, 4, 7, 10, 13)

» v = [1 4 7 10 13] v = 1 4 7 10 13

Oszlopvektort hasonlóan készíthetünk, de az elemeket pontosvesszővel választjuk el:

```
» w = [1;4;7;10;13]
w =
1
4
7
10
13
```

Sorvektorból transzponálással is készíthetünk oszlopvektort (vagy fordítva), erre az aposztróf szolgál:

```
» w =v'
w =
1
4
7
10
13
```

A v vektor *i*-edik koordinátáját v(i) adja meg (az indexelés 1-gyel kezdődik). Pl.

» w(2)= ans = 4

Több elemből álló blokkok lekérdezésére a kettőspont operátort használhatjuk, pl. a fenti v vektor első három elemét megkapjuk, ha beírjuk:

» v(1:3) ans = 1 4 7

Vagy, ha a v vektor elemei a harmadik elemtől az utolsóig érdekelnek bennünket:

```
» v(3,end)
ans =
7 10 13
```

ahol end az utolsó elemet jelenti.

Vektor hosszára a length ill. a size paranccsal kérdezhetünk rá. Az előbbi csak a vektor elemszámát adja meg, az utóbbi mátrixnál is használható, és a sorok és oszlopok számáról tájékoztat.

```
» length(v)
ans =
5
% size(v)
ans =
1 5
```

Az utóbbi sor alapján 1-szer 5-ös tömbbel, azaz sorvektorral van dolgunk.