

Matlab

Bevezető

MATLAB-MATtrix LABoratory

-a rendszert úgy szerkesztették meg, hogy a mátrixokkal való műveletek elvégzése nagyon egyszerű legyen.

-interaktív módon használható, ebben eltér az eddig ismert programozási nyelvektől(C++, Java)

-az utasítás beírása után (a speciális MATLAB prompt helyén) azonnal választ, eredményt kapunk.

Sikeres MATLAB programozáshoz:

- Ismerni kell a pontos alakját az utasításoknak
- Ki kell fejleszteni egy logikus tervet sajátos feladatok megoldásához

Indítás:Windows- icon vagy UNIX- matlab beírás

Több ablak jelenik meg

Lezárás:**File** menüből *Exit MATLAB* vagy prompt jel után írd *quit* v. *exit*.

>> 2 + 3 <Enter>

-a parancs csak akkor kerül végrehajtásra, ha <Enter> követi

>> 2 * 3 <Enter>

>> 1/2 <Enter>

>> 2^ 3<Enter>

>> 2\1 <Enter> 1/2 megfelelője

Mi történik 1/0 esetén? Próbálják ki $13 + Inf$ illetve $29/Inf$, $0/0$ -NaN

Matlab Help használata

help-alkönyvtárak listája

help alkönyvtár-megtalálható az ott lévő MATLAB kulcsszavak listája témák szerint csoportosítva, pl. `help elfun`

help kulcsszó megkapjuk a konkrét kulcsó MATLAB jelentését, pl. `help eye`

Ha a megoldandó feladatot ismerjük, de a kulcsszót nem, akkor **lookfor**(keresd!) parancs.

pl. `lookfor integr-integrálási` lehetőségekről akarunk érdeklődni.

helpwin parancs hatására külön ablakban jelenik meg az alkönyvtárak listája, egérrel kattintva közvetlenül bontjuk ki az információkat.

helpdesk-részletes leírásokat tartalmazó HTML fájlokból tanulmányozhatjuk a MATLAB-ot.

1. Változók neve, deklarálása

M-ban a változók nevét 19 jelből (betű v. szám) állíthatjuk össze, az első jel betű kell legyen. Különbség van a kis és nagy betűk között.

Lefoglalt nevek (nem használhatóak):

-**ans**-névnélküli kifejezés neve

-**eps**- gépi relatív pontosság értéke az adott számítógépen

-**realmax**-az adott számítógépen a legnagyobb lebegőpontos szám értéke

-**pi**-az egységkör területe kb. 16 számjegyes pontossággal

-**inf**-az $1/0$ osztások eredménye, a M nem áll meg nullával való osztás esetén

-**NaN**-a $0/0$ és inf/inf -féle osztások eredménye, a M nem áll meg ilyenre

-**i,j**-mindkettő képzetes egység, a M-ban komplex számokat is használhatunk

-**clock**- sorvektor, melynek tartalma a pillanatnyi év, hó,nap, óra, perc, másodperc

-**cputime**-a M indítása óta eltelt idő másodpercekben

-**tic**-0-ra állítja a kezdőértéket a *toc* számára

-**toc**-a legutolsó *toc* utasítás óta eltelt idő másodpercekben

2. Adattípusok

- alapvető adattípus a mátrix amit nem kell külön deklarálni
- négyzetes vagy téglalap alakú, valós vagy komplex, teljesen kitöltött vagy ritka
- mátrixindex kezdőértéke mindig 1!!!!

$a = [1 \ 3 \ 4; 2 \ -1 \ 0; 3 \ -1 \ 7]$

$inv(a), det(a), size(a), rank(a)$

$A = [2 \ 4 \ 1; -4 \ 1 \ 9; 0 \ -2 \ 3]$

A' - A transzponáltja

$c = a * A$

$d = a. * A$

$e = a.^2$

$f=abs(a), m=mean(a), g=geomean(a), s=sum(a), p=prod(a), \max(a), \min(a), \text{diag}(a)$

$\text{triu}(a)$

$m = [2 \ 3 \ 5; 0 \ 7 \ 1; 8 \ 10 \ -1]$

$m(2,1)$

$m(:,1)$ - m első oszlopa

$m(:,j)$ - m j -dik oszlopa

$m(2,:)$ - m 2-dik sora

$m(i,:)$ - m i -dik sora

$m(i, 1 : 3)$ vagy $m(i, 1 : \text{end})$

$m(2 : 3, 2 : 3)$

Sajátos mátrixok

$\text{eye}(8), \text{eye}(5, 7), \text{zeros}(5, 7), \text{ones}(5, 7), \text{magic}(3), \text{hilb}(3), \text{pascal}(4)$

(toeplitz, hankel, vander) -vektorok megadása

$b = [1 \ 2 \ 3]$ vagy $b = [1, 2, 3]$ -sorvektor

$B = [4; 5; 6]$ vagy $B = [4 \ 5 \ 6]'$ -oszlopvektor

$b * B$

$c = [4 \ 5 \ 6]$ vagy $c = B'$ - B transzponáltja

$c1 = b. * B$

$c2 = b.^2$

$c3 = b.^B$

$d = 1 : 6$ -felsorolás, (lépés=1)

$e = 1 : 2 : 10$ -kezdőérték:lépés:végsoérték

$f = 10 : -1 : 1$

$\exp(b), \exp(1) = e, \text{sqrt}(b)$

$g = [-1 \ 0 \ -5], \text{abs}(g), \text{length}(g)$

$\text{mean}(b), \text{geomean}(b), \text{sum}(b), \text{prod}(b)$

Karakterláncok Matlab-ban

```
>> s='Ez egy szoveg'
>> size(s) ans= 1 13
Konverziós függvények:
-numerikus adatokat karakterláncokká konvertálhatunk int2str,
num2str, mat2str függvényekkel
Például:>> sint=int2str([pi,eps;pi,eps]);
>> sint =
    3    0
    3    0
>> intdim=size(sint)→ intdim=2 4
```

3. Aritmetikai műveletek

$A + B$, $A - B$, $A * B$, $A ./ B$ -elemenkénti szorzás
 $A ./ B$ -jobboldali osztás, eredménye Y , amelyre $Y * B = A$
 $A \setminus B$ -baloldali osztás, eredménye X , amelyre $A * X = B$

Relációs operátorok

$< \rightarrow <$
 $\leq \rightarrow \leq$
 $> \rightarrow >$
 $\geq \rightarrow \geq$
egyenlőség $\rightarrow ==$
 $\neq \rightarrow \neq$

Mátrixokra is alkalmazható, ekkor elemenként végzi.

Logikai operátorok

1. Negáció: \sim
2. Logikai szorzás, és: $\&$ mely akkor igaz, ha mindkét operandus igaz
3. Logikai összeadás, vagy: \mid mely akkor hamis ha mindkettő hamis.

Formátum

A számok kinyomtatásához a Matlab a 'short' formátumot használja, azaz 4 tizedesjegyre kerekít, ezen változtathatunk:

- format bank: 2 tizedesjegyre kerekít
- format long: 14 v. 15 tizedesjegyre kerekít
- format long e: 16 értékes jegyre kerekít, exponenciális szorzót is készít

-format compact: sűrített forma

Használjátok a M helpjét, visszakapcsolni format short utasítással lehet.

Grafika

Egyszerű grafikákat a plot(x,y) utasítással lehet, ahol x, y ugyanolyan méretű vektorok(lehetnek mátrixok is). Ekkor az (x_i, y_i) koordinátájú pontokat rajzolja ki.

Példa: Ábrázoljuk a $\sin(x)$ függvényt a $[0,10]$ intervallumon

4. Feltétel vizsgálat

-if utasítás

if változó(tömb), utasítás(ok), *end*

-változótömb lehet logikai és aritmetikai kifejezés is, melynek van aktuális értéke az *if* utasítás végrehajtásakor. Logikai kifejezés esetén ha értéke 0 v. tömb esetén legalább egy nullát tartalmaz akkor az *if*-ben megadott utasítások nem lesznek végrehajtva. Ha értéke 1, v. tömb esetén egy érték sem nulla, akkor végrehajtja az utasításokat.

1. Példa. `>> a = 2`

`>> if a == 2 b = 10; end`

`>> b`

`b=10`

-tömb esetén

`>> a1 = [2 < 10 3 == 4]`

`a1 = 1 0`

`>> if a1 b = 11; end`

`>> b`

`b=10.`

if változó(tömb), utasítás(ok)1;

else utasítások2;

end

2. Példa. `>> Am = [2 0; 2 -1]`

`>> Am`

```
>> if Am b = 13; else b = 14; end
>> b
b = 14
```

Egymásba ágyazott if szerkezetek

```
if kifejezés1 utasítások1;

    if kifejezés2 utasítások2;

        else utasítások3;
    end
end
```

3. Példa. >> a = 2

```
>> if a == 2,
if a > 3 b = 10;
else b = 0;
end else
b = 12;
end
>> b
b = 0.
```

Ciklus szervezés

-for és while utasítások

-for: előre meghatározott változó az ú. n. ciklusváltozó alapján szervezi a ciklust előre meghatározott lépésszámmra.

-while: nem kell rögzíteni a ciklus ismétlődésének számát, a ciklus befejezését a feltétel(ek) vizsgálatával vezéreljük.

for változó=kifejezés utasítás(ok) end

-ahol a változó a ciklusváltozót jelenti.

-a kifejezésben meg kell adni a ciklusváltozó kezdeti értékét, a ciklusváltozó lépésenkénti növekményét és a ciklusváltozó utolsó értékét. A növekmény lehet negatív is.

A kifejezés szintaktikája:

kezdő érték: növekmény: utolsó érték

Ha növekmény nincs, akkor 1.

4. Példa. *Feltöltünk egy A sorvektort az első tíz természetes szám harmadik hatványával.*

```
>> A = [];
>> for i = 1 : 10; A(i) = i ^ 3; end
>> A
```

-for ciklusok egymásba ágyazhatók

5. Példa. 5x4-es A mátrixot feltölteni $A(i, j) = \frac{1}{i+j}$.

```
>> A = [];
for i = 1 : 5
for j = 1 : 4
A(i, j) = 1/(i + j);
end
end
>> A
```

FELADAT: készítsd el a $\sin(x)$ értékeinek táblázatát $(x, y) \in [-1, 1]$ tartományon, 0.1 pontosságú rácson.

```
>> for i = -1 : 0.1 : 1; ss = [ss, sin(i)] end
```

vagy

```
>> for i = 1 : -0.1 : -1; sss = [sss, sin(i)] end
```

Ha a ciklusváltozó értelmezési intervallumát előre nem tudjuk, a ciklus szervezését a **while** utasítással oldhatjuk meg.

```
while változó, utasítás1, utasítás2, .. end
```

-a változó értékét legtöbbször valamilyen logikai kifejezés kiértékelésével kapjuk meg. A while ciklus addig ismétlődik, amíg a változó értéke nulla nem lesz. Ha ez nem áll elő, akkor végtelen ciklust kapunk.

5. Matlab függvények, m kiterjesztésű fájlok használata

A Matlab rendszer függvényekből épül fel : beépített függvények és saját függvények

A beépített függvények programlistáját nem lehet megnézni, viszont az m-fájlban tárolt függvények programlistája tanulmányozható.

Ha olyan függvényt akarunk használni, amely a rendszerben nincs, akkor magunk is írhatunk m-fájlokat (ne adjunk olyan nevet, amely a rendszerben már van, mert ekkor bizonytalanná válik a rendszer működése).

Definiálunk egy m-kiterjesztésű fájlt, amelynek neve ugyanaz, amilyen néven az új függvényt használni szeretnénk.

A függvény első sorában meg kell adni, ha a szóbanforgó fájl egy függvény, milyen formában kapjuk végrehajtás után az eredményt, a nevét és milyen adatokra van szükség.

Példa

```
function [maxnorm,norm2]=norma(x)
% a függvény az x vektor normáját számolja
% maxnorm -maximum norma
% norm2- euklideszi norma
n=length(x);
maxnorm=max(abs(x));
ss=0;
for i=1:n
ss=ss +x(i) ^ 2;
end;
norm2=ss;
```

A program használata:

```
>> x = [11 -2 3 8 5 6]
```

```
>> [a,b]=norma(x);
```

A függvény programjában minden változó lokális változó.

Az m fájlok futását vezérlő utasítások

- pause* felfüggeszti az m fájl futását billentyű lenyomásig
- pause(n)* n másodpercre függeszti fel
- pause off* a programban az utasítást követő *pause* parancsokat figyelmen kívül hagyja
- pause on* hatástalanítja a *pause off* utasítást
- break* *for* vagy *while* ciklusok futását megszakítja
- return* megállítja az m fájl futását és visszatér oda ahonnan az m fájlt meghívtuk
- error(szöveg)* az m fájl végrehajtása befejeződik és a hibaüzenetként megadott szöveg kiíratásra kerül a képernyőre
- nargin, nargout* (pl. *nargin('norma')*, *nargout('norma')*)

6. A függvények mint más függvények paramétere

A M hasonlóan a magasszintű programozási nyelvekhez megengedi, hogy egy függvény paramétere függvénynév legyen.

Eval és feval utasítások

-*eval*-paraméterrel vagy paraméterekkel együtt szövegesen megadott függvény kiértékelését végzi el a megadott paraméterekre

```
>> valt = 'sin(2.3)';
```

```
>> eval(valt) → ans = 0.7457
```

-ha a függvény paramétere nem konkrét érték hanem változó, akkor a változónak értéket kell adni:

```
>> x = 3;
```

```
>> valt = ('sin(x)');
```

```
>> eval(valt)
```

-(*feval*)-csak a függvény nevét kell megadni szövegesen, amely lehet beépített vagy m-fájlban megadott függvény, de a függvény paramétereinek értékét a *feval* utasításban fel kell sorolni a függvény neve után

```
>> valt = 'cos';
```

```
>> feval(valt, 2) → ans = -0.4161 vagy
```

```
>> x = [1 2 3 4];
```

```
>> feval(valt, x)
```

Példa: Egy később megadott függvény $[a, b]$ intervallumon való értékeinek tabellázása n alappontban.

Eval utasítással:

```
function y=tablazat(ff,a,b,n)
```

```
% az ff függvény értékei az  $a + i(b - a)/n, i = 0, \dots, n$  pontokban
```

```
% az eredmény az y=[x,fx] táblázatban jelenik meg
```

```
h=(b-a)/n; x=a:h:b; z=eval(ff);y=[x,z];
```

Feval utasítással

```
function y=tablazat1(ff,a,b,n)
```

```
% az ff függvény értékei az  $a + i(b - a)/n, i = 0, \dots, n$  pontokban
```

```
% az eredmény az y=[x,fx] táblázatban jelenik meg
```

```
h=(b-a)/n; x=a:h:b;z=feval(ff,x);y=[x,z];
```

Az m-fájl alkalmazása

```
>>tablazat('sin(x)',0,1,6) >>tablazat1('sin',0,1,6)
```