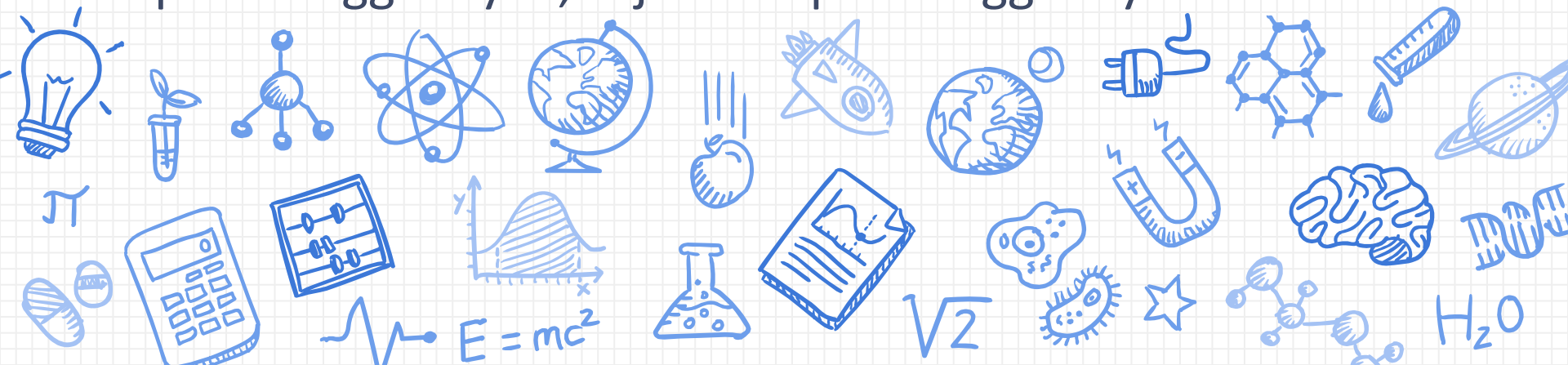


MATLAB 2017

1. gyakorlat

adminisztráció, kalkulátor,
beépített függvények, saját szkript és függvény írása



Adminisztráció

Gyakorlatvezetők és elérhetőségek:

- ✘ Makra Ákos -- akosmakra@gmail.com
- ✘ Reguly István -- reguly.istvan@itk.ppke.hu
- ✘ Zsedrovits Tamás -- zsedrovits.tamas@itk.ppke.hu

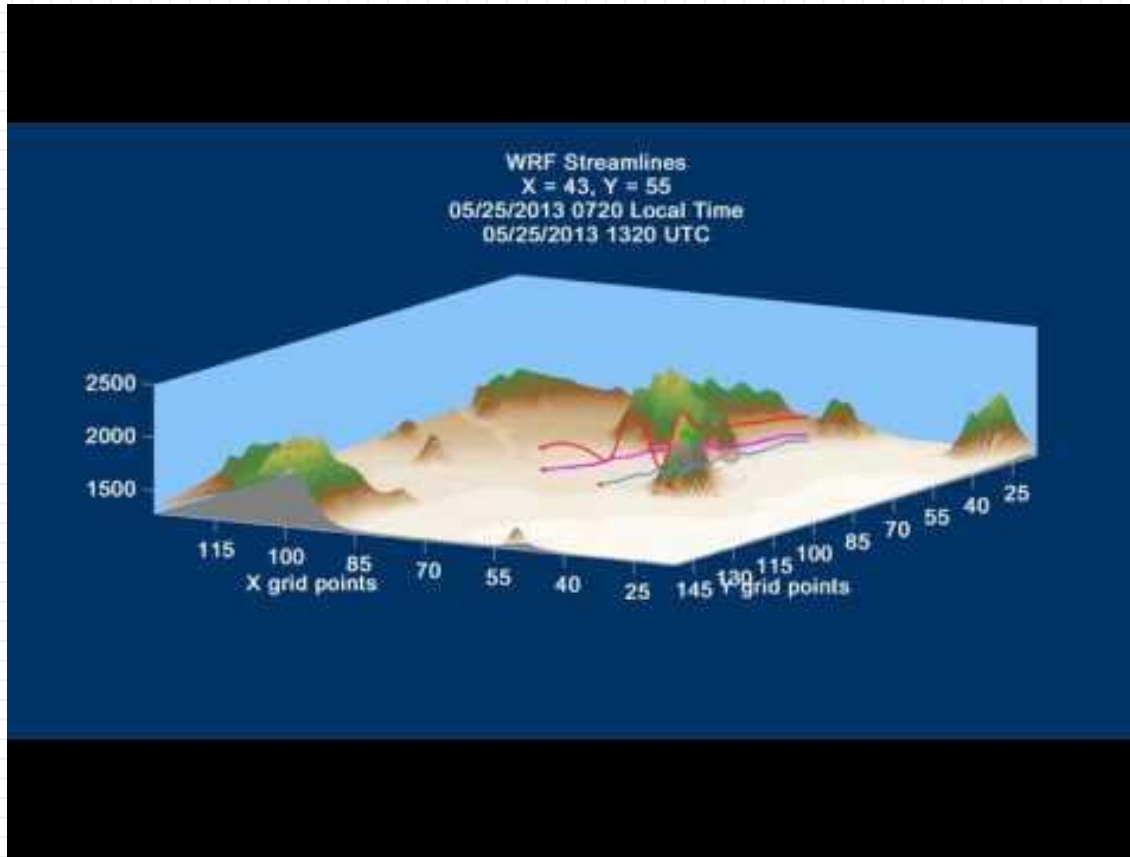
Kérjük, hogy minden levél tárgyában szerepeljen a
“**MATLAB**” szó.

Követelmények és tematika: a tárgy wiki-oldalán
<https://wiki.itk.ppke.hu/twiki/bin/view/PPKE/Matlab>



- ✗ MATrix LABoratory (MathWorks)
- ✗ teljes szoftver csomag, ami nagyban megkönnyíti és felgyorsítja az algoritmusfejlesztést
- ✗ hatékony numerikus megoldók mátrixműveletekre optimalizálva, saját (egyszerűen használható) szkript nyelv, könnyű adatmegjelenítés, rengeteg toolbox

MATLAB



Video courtesy of
Dr. Zachariah
Silver
zsilver@nd.edu





Parciális Differenciálegyenletek megoldása és vizualizációja (Vortex shedding)

MATLAB

Miért jó nekünk?

- ✗ megkönnyíti a munkánkat,
- ✗ sok tárgy épít rá,
- ✗ **MI**: a legegyszerűbb első lépés algoritmusfejlesztésben (optimális kódhoz később pl. C, C++),
- ✗ **MB**: alapvető programozási platform, hatékonysága sok mindenre elég.



Margószéli megjegyzés -

- ## hogyan állítsuk a betűméretet nagyobbra?

felső szalag felületen a
“Home” fülön →
Preferences → a felugró
ablak bal oldali listájában
“MATLAB” / “Fonts”
menüpont → jobb oldalon
feltűnő panelen “Desktop
code font” méretét állítsuk
nagyobb (pl. 10).

- ✗ a Command Window-ban dolgozunk (interaktív parancsértelmező)
 - ✗ változók megadás egyből értékadásnál (nem kell előre deklarálni a típust, sőt, a típust nagyon ritkán adjuk meg közvetlenül)
- ```
pl a = 1; (és nem unsigned int a = 1;)
```

# A MATLAB mint számológép

- ✗ beépített konstansok (i, j, pi, ans, inf, -inf, nan, eps)
- ✗ operátorok (+, -, \*, /, ^, :, ==)
- ✗ beépített függvények (sin, cos, tan, atan, sqrt, exp, power, pow2, log, log10, exp, factorial, factor, primes, round, floor, ceil, abs, datestr(clock), min, max)
- ✗ egyéb (., ;, ..., %, clear X, clc, save asdf.mat a, load asdf.mat, format)

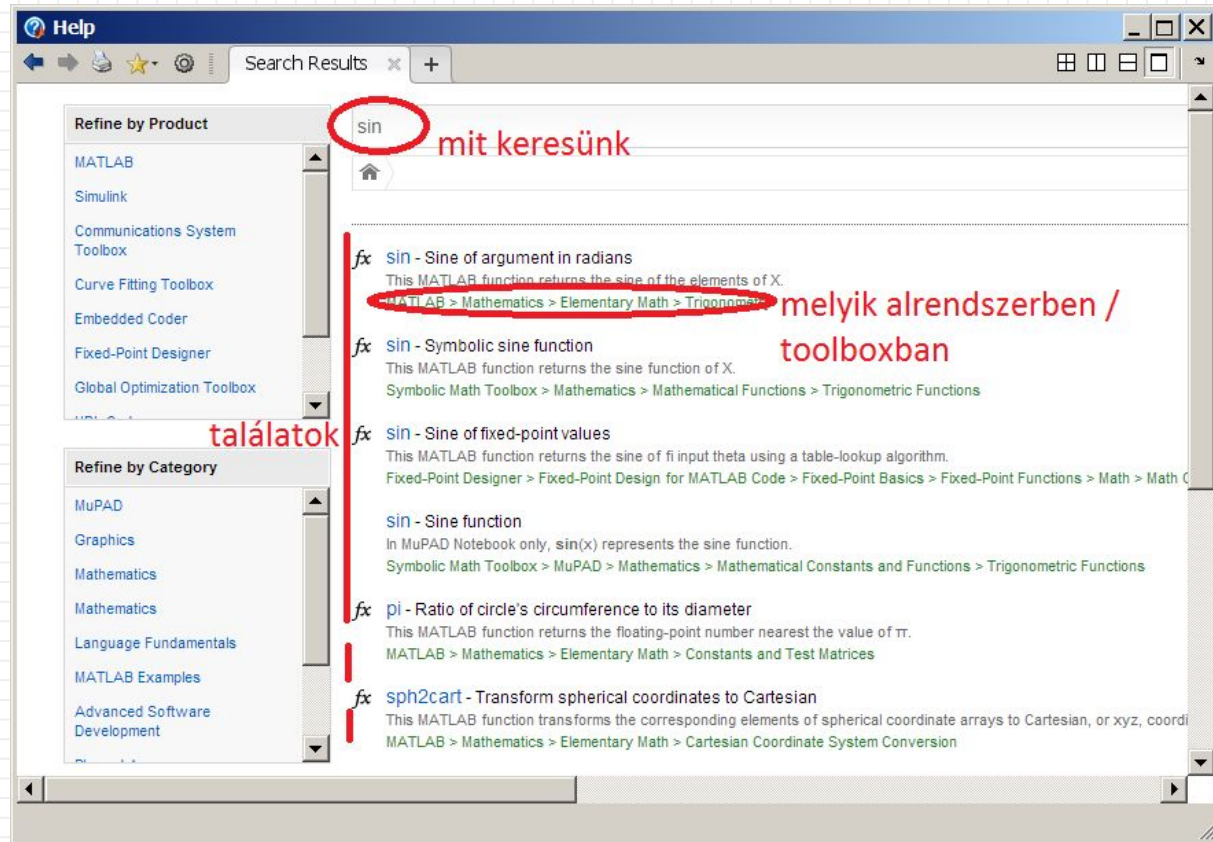
## Help használat példa – sin

Előhívás:

✗ jobb felső  
részen  
fehér  
körben  
kérdőjel

VAGY

✗ F1



# Help használat példa – sin

Help

sin x +

Documentation Center

MATLAB

Getting Started with MATLAB

Examples

Release Notes

Functions

Language Fundamentals

Mathematics

Elementary Math

Arithmetic

Trigonometry

Functions

sin

sind

asin

asind

sinh

asinh

cos

cosd

acos

acosc

cosh

acosh

tan

tand

atan

atand

atan2

atan2d

tanh

atanh

csc

cscd

acsc

acscd

csch

Search Documentation

MATLAB Mathematics Elementary Math Trigonometry

**sin**

Sine of argument in radians

**Syntax**

$Y = \sin(X)$

**Description**

$Y = \sin(X)$  returns the sine of the elements of X. The **sin** function operates element-wise on arrays. The function accepts both real and complex inputs. For real values of X in the interval  $[-\text{Inf}, \text{Inf}]$ , **sin** returns real values in the interval  $[-1, 1]$ . For complex values of X, **sin** returns complex values. All angles are in radians.

**Examples**

Plot Sine Function

Sine of Vector of Complex Angles

**Input Arguments**

$X$  — Input angle in radians  
scalar value | vector | matrix | N-D array

**Output Arguments**

$Y$  — Sine of input angle  
scalar value | vector | matrix | N-D array

**More About**

Sine Function

**See Also**

asin | asin | sind | asinh

**rovid leírás**

**szintaxis**

**részletes leírás**

**példák**

**paraméter magyarázat**

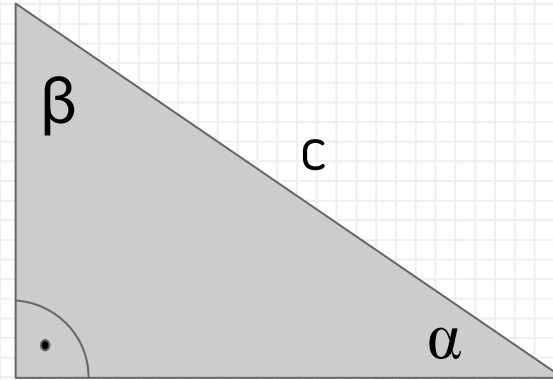
**elméleti / részletező megjegyzések**

**kapcsolódó tartalmak**

Was this topic helpful? Yes No

# A MATLAB mint számológép használata

Legyen adott a következő derékszögű háromszög:  $a=3$



## Számítsuk ki:

- ✗  $c$  értékét, majd  $b=5$
- ✗  $a$  értékét  $a$  és  $b$  felhasználásával, majd
- ✗  $a$  értékét  $b$  és  $c$  felhasználásával.

## Mit tapasztalunk? Egyenlő a két érték?





## A MATLAB mint számológép használata

---

Megoldás:

```
a = 3;
```

```
b = 5;
```

```
c = sqrt(a^2 +
b^2);
```

```
alpha1 = atan(a/b);
```

```
alpha2 = acos(b/c);
```

```
alpha1 == alpha2
```

```
% nem lesz igaz, az
oka: veges
```

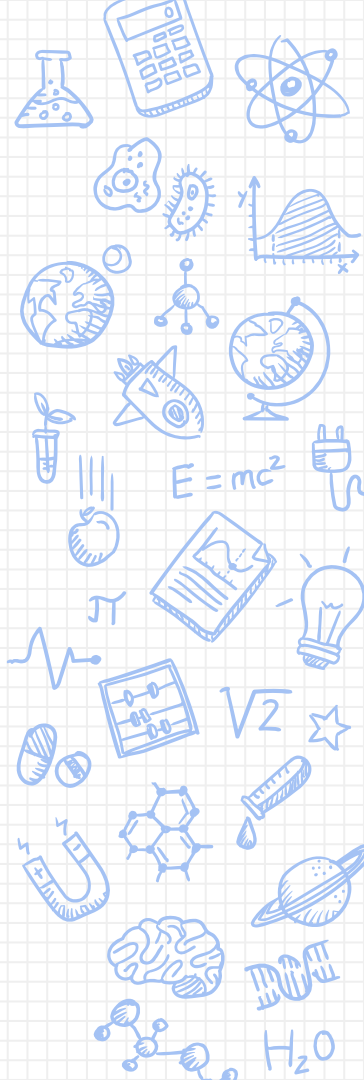
```
% reprezentacio,
kerekitesi hiba
```

```
% lehetséges javítás:
```

```
abs(alpha1 - alpha2)
<= eps
```

```
% igaz lesz,
```

```
% mernokileg egyenlo
```



- ✗ az Editor ablakban készülnek;
- ✗ `.m` kiterjesztés, megkötések:
  - ✗ NE kezdődjön számmal,
  - ✗ NE legyen benne space;
- ✗ parancsok szekvenciális sora (ugyanúgy értelmezi a MATLAB, mintha a Command Window-ba gépelnék a sorokat).

# MATLAB szkriptek

új fájl

futtatás (F5)

az Editor fül aktív

maga a szerkesztőablak (amiben a szkriptek és függvények készülnek)

```
1 KinectHandles=mxNiCreateContext();
2
3
4 figure('Position',[200 200 930 320]);
5 I = mxNiPhoto(KinectHandles);
6 I = permute(I,[3 2 1]);
7 D = mxNiDepth(KinectHandles);
8 D = permute(D,[2 1]);
9
10 mxNiChangeRegisteringMode(KinectHandles);
11 DI_reg = mxNiRegistered(KinectHandles);
12 mxNiChangeRegisteringMode(KinectHandles);
13 DI_reg = permute(DI_reg,[2 1]);
14
```

Microsoft (R) Manifest Tool version 6.3.9600.16384

Copyright (c) Microsoft Corporation 2012.

All rights reserved.

del C:\Users\m\AppData\Local\Temp\mex\_212518894609\_5584\mxNiRegistered.obj mxNiRegistered.exp mxNiRegi

MEX completed successfully.

>>

>> modalitas\_illesztes\_probaja

After initialization:

Successful started color stream

Successful started depth stream

Sikerult, off-rol reg-ze

Sikerult, reg-rol off-ra

f>

>>

<

| Name          | Value                  | Min      |
|---------------|------------------------|----------|
| D             | 480x640 uint16         | 0        |
| D2            | 480x640 uint8          | 0        |
| d2max         | 93                     | 93       |
| d2min         | 0                      | 0        |
| DI_reg        | 480x640 uint16         | 0        |
| DI_reg2       | 480x640 uint8          | 1        |
| h1            | 174.0016               | 174.0... |
| h2            | 179.0016               | 179.0... |
| h3            | 184.0016               | 184.0... |
| I             | 480x640x3 uint8        | <Too ... |
| I2            | 480x640 uint8          | 1        |
| KinectHandles | [8791424647312 8791... | 0        |
| nld_idx       | 480x640 logical        |          |

# MATLAB szkriptek - pelda.m

```
% peldaszkrípt - nincs bemenő paraméter
clear variables;
tmp = pi;
% kiírás a konzolra (Command Window):
disp('Lefutott');
```

Mentsük el a fentebbi sorokat az alábbi fájl névvel: `pelda.m`

## Kapcsolódó kérdések:

1. mit látunk, ha beírjuk a konzolba az alábbi parancsot: `help`  
`pelda`
2. futtassuk le!
3. mi van a Workspace-ben?



## saját függvények MATLAB-ban

---

### szkriptekhez hasonlóan:

- ✗ az Editor ablakban készülnek, `.m` kiterjesztésűek,
- ✗ ugyanúgy nem kezdődhet számmal, és nem lehet space a nevében,
- ✗ ne egyezzen meg beépített függvény nevével (pl. ~~plot.m~~);

### szkriptektől eltérően:

- ✗ saját munkatérbe dolgozik - a függvényben használt változók itt jönnek létre a függvény futása során, majd a függvény terminálásakor meg is szűnnek;
- ✗ fejléc formája: `function` kulcsszóval kezdődik, `end` zárja le;
- ✗ opcionálisan lehetnek bemenő paraméterek és visszatérési értékek,  
pl: `function [visszateresi_param1, vp2] =  
pelda_fv(bemeneti_p1, bp2)`



## saját függvények MATLAB-ban – peldafv.m

---

```
function kimenet = peldafv(bemenet1,
bemenet2)
% Ez a peldafuggveny help bejegyzese.
% A ket bemeno parameter tetszoleges szam,
% a kimenet ezek osszege.
 kulonbseg = bemenet1 - bemenet2;
 kimenet = bemenet1 + bemenet2;
end
```

Mentsük el a fentebbi sorokat az alábbi fájl névvel:  
peldafv.m



## saját függvények MATLAB-ban – peldafv.m

---

### Feladatok:

1. mit ír ki a `help peldafv`?
2. hívjuk meg a konzolon a függvényünket:  
`osszeg = peldafv(2.3, 5.6);`
3. mi van a Workspace-ben?







## saját függvények MATLAB-ban – példafv2.m

---

### Feladatok:

1. hívjuk meg a konzolon a függvényünket:

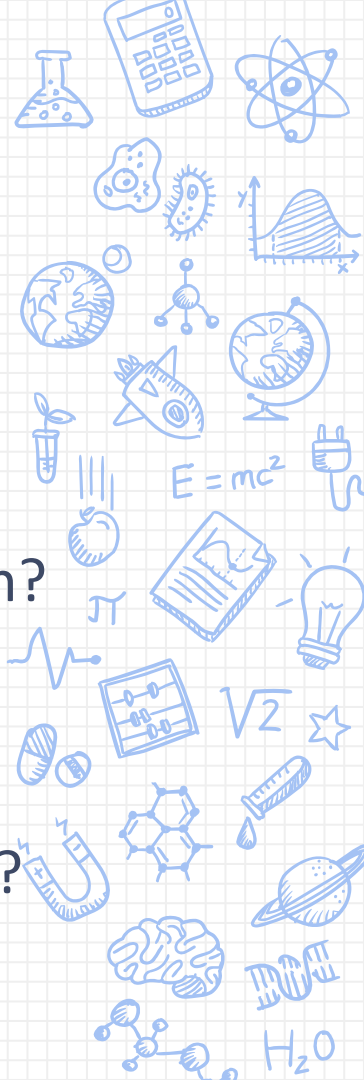
```
osszeg = példafv2(2.3, 5.6);
```

2. Mit ad vissza a függvény, mi van a Workspace-ben?

3. Hívjuk meg a konzolon a függvény a következőképp:

```
[osszeg szorzat] = példafv2(2.3, 5.6);
```

4. Mit ad vissza a függvény, mi van a Workspace-ben?



-

# MATLAB 2017

## 2. gyakorlat

vektorok, 2D ábrázolás



## Vektorok, alpműveletek

✗ sor- és oszlopvektor létrehozása:

✗ `sorv = [2 1 4 3];`

✗ `oszlopv = [6; 5; 7; 9];`

✗ transzponált: `sorv = sorv';`

✗ összefűzés: `ofv = [sorv, oszlopv'];`

✗ indexelés:

✗ **1-től kezdődik!**

✗ speciális index: `end` -- ez mindig az utolsó elem indexét jelenti, pl. `ofv(end)` ;

|   |   |   |   |
|---|---|---|---|
| 2 | 1 | 4 | 3 |
|---|---|---|---|

|   |
|---|
| 6 |
| 5 |
| 7 |
| 9 |

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 2 | 1 | 4 | 3 | 6 | 5 | 7 | 9 |
|---|---|---|---|---|---|---|---|

↑  
1

↑  
end



## Vektorok, alpműveletek

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 2 | 1 | 4 | 3 | 6 | 5 | 7 | 9 |
|---|---|---|---|---|---|---|---|

- ✗ indexelés:
  - ✗ adott elem lekérése: `elem = sorv(3) ;`
  - ✗ adott elem felülírása: `sorv(3) = 44 ;`
  - ✗ `length` - hossz lekérdezése
- ✗ `size` - méret lekérdezése, annyi elemű vektorral tér vissza, ahány dimenziós a mátrix, de legalább 2 - mivel a vektorra is mátrixként tekint



## Vektorok, alpműveletek

✗ `ones`, `zeros` - csupa egyes vagy csupa nullás számjegyből álló tömb létrehozása

✗ ha **1 paraméterrel** használjuk, akkor négyzetes mátrixszal tér vissza: `ones(2)`

|   |   |
|---|---|
| 1 | 1 |
| 1 | 1 |

✗ ha **2 vagy több paraméterrel**, akkor rendre az egyes dimenziók mérete az egyes paraméterek szerint:  
`zeros(1, 3)`

|   |   |   |
|---|---|---|
| 0 | 0 | 0 |
|---|---|---|

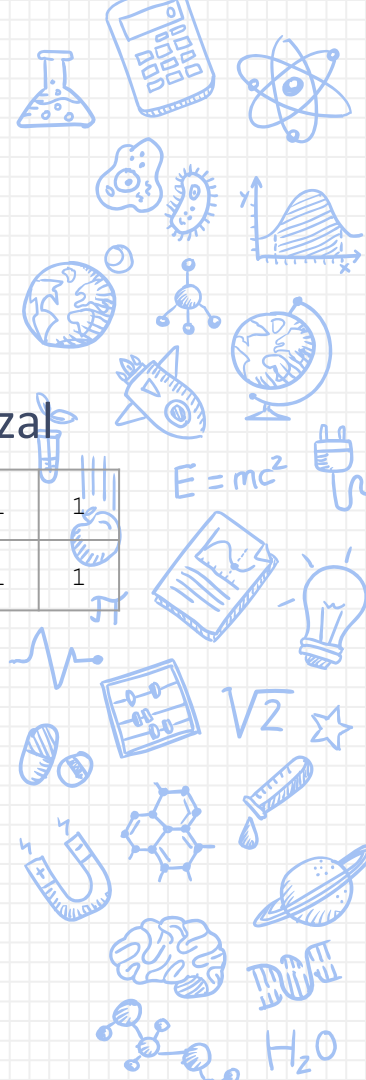
✗ `rand` - véletlen számokból álló tömb generálása

✗ `eye` - egységmátrix generálása

✗ `diag`:

✗ adott diagonálissal rendelkező mátrix előállítása,

✗ egy mátrix diagonálisának kinyerése





## Vektorok, alpműveletek

✗ `sum, prod, mean`

✗ `min, max:`

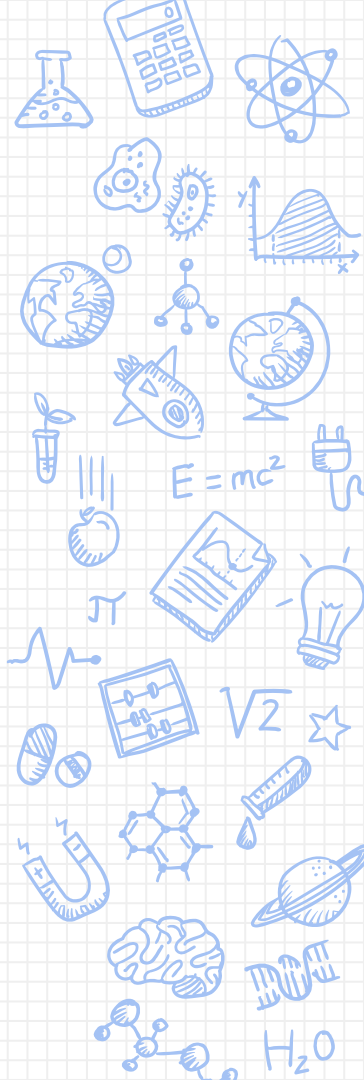
```
min([2 7 1 4 6])
→ ans = 1
```

✗ ha **1 visszatérési értéket** várunk, vagy nem is mentünk semmit → **szélsőérték**

✗ ha **2 visszatérési értéket** várunk → **szélsőérték és annak indexe**

```
[a, b] = min([2 7 1 4 6])
→ a = 1
→ b = 3
```

✗ **elemenkénti műveletvégzés** - `e.*f`,  
`g.^2`, stb.





## Vektorok, alpműveletek

---

- ✗ `find` - paraméterben megadott logikai feltétel alapján keresés, azokkal a skalár indexekkel tér vissza, amely indexű elemek a feltételnek eleget tesznek
- ✗ logikai műveletek és indexelés:
  - ✗ logikai feltétel alapján (pl.  $a < 1.2$ ,  $b \sim 3$ , stb)
  - ✗ az **eredeti vektor hosszával megegyező logikai vektort** kapunk vissza, ami
  - ✗  $\{0, 1\}$  számok valamilyen sorozata, ahol
    - 1 jelentése: igaz az adott elemre a feltétel
    - 0 jelentése: nem igaz az adott elemre a feltétel



## Vektorok, alpműveletek

- ✗ logikai műveletek és indexelés:
  - ✗ logikai feltétel alapján (pl.  $a < 1.2$ ,  $b \sim 3$ , stb)
  - ✗ az **eredeti vektor hosszával megegyező logikai vektort** kapunk vissza, ami
    - 1 jelentése: igaz az adott elemre a feltétel
    - 0 jelentése: nem igaz az adott elemre a feltétel

```
alma = [23 14 31 17 42];
indexek1 = find(alma>24) % → [3, 5]
alma(indexek1) % → [31, 42]
```

23>24 14>24 31>24 17>24 42>24

```
indexek2 = alma>24 % → [0, 0, 1, 0, 1]
alma(indexek2) % → [31, 42]
```





## 2D ábrázolás

✗ `figure` - új ábra létrehozása

```
alma = [23, 14, 31, 17, 42];
plot(alma);
```

✗ `plot(x), plot(x, y)` - adatsor kirajzolása

✗ **egyargumentumú** esetben: a vektor értékei a vektorindexek függvényében

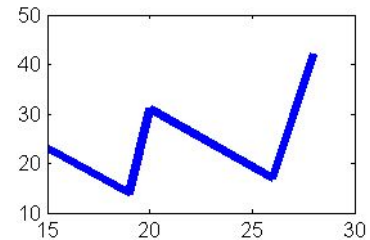
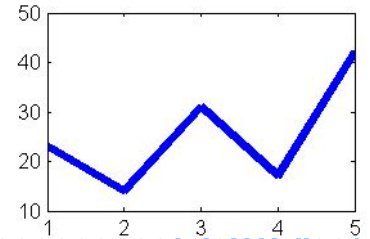
✗ **kétagumentumú** esetben:  $x \rightarrow$  helyek,  $y \rightarrow$  értékek

```
korte = [15, 19, 20, 26, 28];
plot(korte, alma);
```

✗ `stem(x), stem(x, y)` - pálcikadiagram, paraméterezés hasonlóan

✗ `stairs` - lépcsődiagram

✗ `hold on/off` - egy rajzfelületre több görbét is kirajzolhatunk (felülírás az alapértelmezett)



## 2D ábrázolás

- ✗ subplot - az ábránkat alábrákra oszthatjuk szabályos rácsos felosztással

```
subplot(2, 3, 4)
```

sorok száma

oszlopok száma

index (sorfolytonosan)

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |



- ✗ rajzolás több cellába:

```
subplot(2, 3, [1, 4])
```



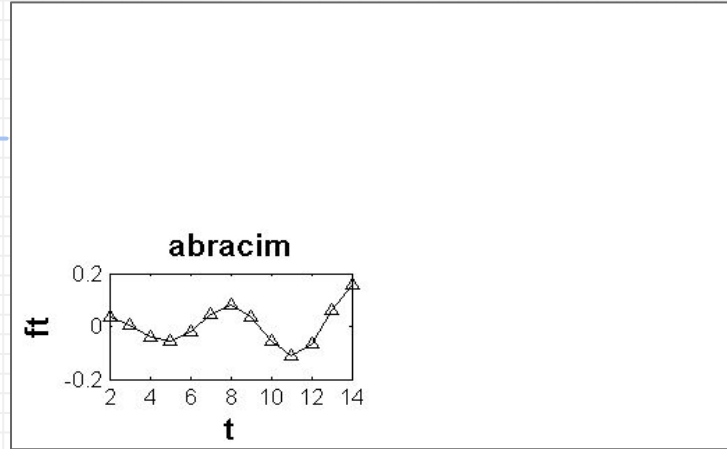
- ✗ grid on/off - a grafikonhoz hátsó hálóbeosztás kérhető
- ✗ legend - adatsorok felirata

- ✗ linespec: nem konkrét parancs, hanem a vonalak részletes beállításának gyűjteménye, kifejezetten a Help-ben megnézni
- ✗ xlim, ylim, box - tengelyhatárok
- ✗ text, xlabel, ylabel, title - csak úgy szöveg, x- és y-tengely felirat, ábracím
- ✗ paraméterezés pl. 'FontSize', 14, 'FontWeight', 'bold'

Hozzunk létre egy  
alábrát az alábbiak  
szerint:

- ✗ az ábra bal alsó negyedébe kerüljön,
- ✗ a  $[2, 14]$  közötti értelmezési tartományon
- ✗ az alábbi értékeket határozzuk meg:

$$t \rightarrow \pi^{0.1t-3} \sin(t)$$



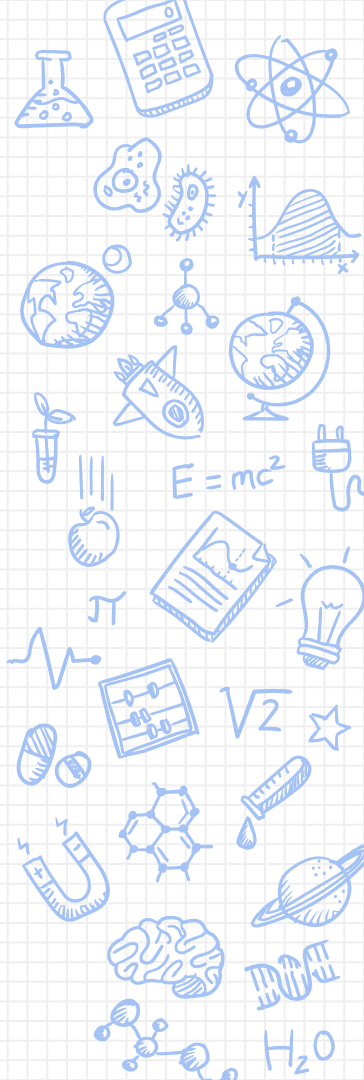
```
t = 2:14;
ft = pi.^(0.1*t - 3) .* sin(t);
figure;
subplot(2, 2, 3);
plot(t, ft, 'k^-');
xlim([t(1), t(end)]);
xlabel('t', 'FontSize', 14,...
 'FontWeight', 'bold');
ylabel('ft', 'FontSize', 14,...
 'FontWeight', 'bold');
title('abracim', 'FontSize', 14,...
 'FontWeight', 'bold');
```

## Feladatok

---

- ✗ A feladatgyűjtemény **2.1 - 2.7** feladatai egyetlen függvényben, melynek neve ez legyen:  
`gyak2_[NEPTUN].m` (Az egyes feladatoknál kért indexeket/indexvektorokat, értékeket/értékvektorokat kérjük kimenetként + ábrák, váz a weboldalon)
- ✗ Az órán ismertetett parancsok kikeresése és tanulmányozása a Help-ben

Amivel nem végzel / nem végzünk, azt otthon kell befejezni, ez a házi feladat is egyben. A határidő vasárnap (február 26.) éjfél. Feltöltés: [users.itk.ppke.hu/~zseta/matlab2017](https://users.itk.ppke.hu/~zseta/matlab2017)





# MATLAB 2017

## 3. gyakorlat

Formázott kiíratás, mátrixok, vezérlőszervezetek





## Formázott kiíratás

### ✗ formatSpec

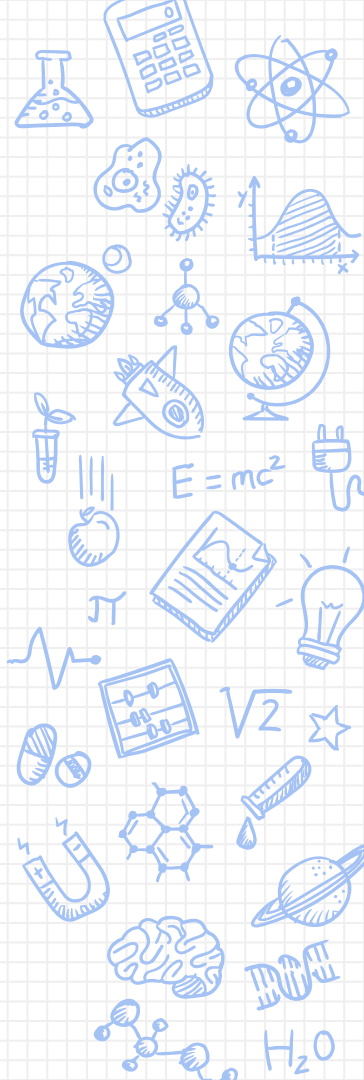
```
fprintf('Teszt: \n\tEgész: %2.0f\n\tTizedes: ...
%2.4f\n\tNormál: %e\n', 10*pi, 10*pi, 10*pi)
```

sortörés, tabulálás

változó hosszúságú  
paraméterlista

- % - mező nyitása
- 2 - minimum mező szélesség
- . - decimális pont
- 4 - tizedesjegyek száma
- f - fixpontos típus

```
Teszt:
Egész: 31
Tizedes: 31.4159
Normál: 3.141593e+01
```



## Formázott kiírás

- ✗ `format` - lebegőpontos számok kijelzésének pontossága

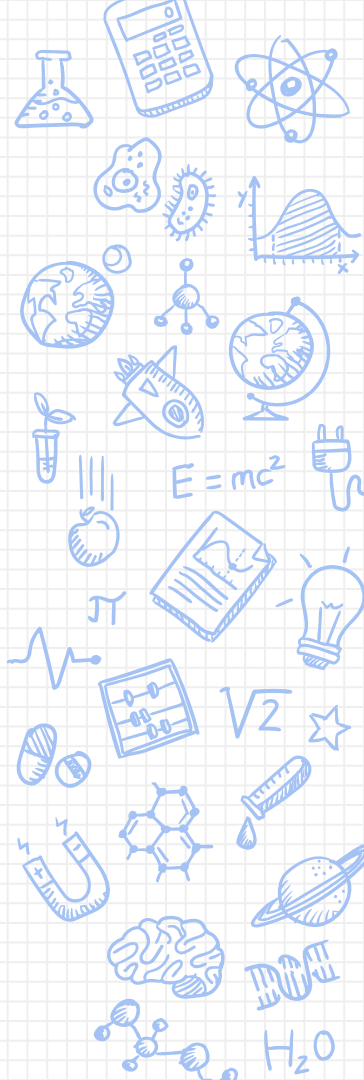
```
format short; disp(pi/100) % → 0.0314
format long; disp(pi/100) % → 0.031415926535898
format shortE; disp(pi/100) % → 3.1416e-02
```

- ✗ `num2str` - szám → string konverzió

```
strPi=num2str(pi,3);
disp(strPi); % → 3.14
```

- ✗ `mat2str` - mátrix → string konverzió

```
strVector=mat2str(1:5);
disp(strVector); % → [1 2 3 4 5]
```



# Mátrixok, alapműveletek

## ✖ mátrix létrehozás, transzponálás, összefűzés, indexelés

```
alma = [11 22 33; 44 55 66];
tmp1 = alma(1, 2); % → 22
tmp2 = alma(2, :); % → 44
```

|   |    |    |    |
|---|----|----|----|
| 1 | 11 | 22 | 33 |
| 2 | 44 | 55 | 66 |

tartomány kijelölés:

kettőspont operátorral

alma(2, 2:3)

|    |    |    |
|----|----|----|
| 11 | 22 | 33 |
| 44 | 55 | 66 |

teljes tartomány - pl összes oszlop:

alma(2, :)

sorrend:

1. sor
2. oszlop
3. mélység
4. ...

```
korte = [77; 88];
```

```
[alma, korte]
```

```
[alma, korte; korte', korte']
```

|    |    |    |    |    |
|----|----|----|----|----|
| 77 | 11 | 22 | 33 | 77 |
| 88 | 44 | 55 | 66 | 88 |

```
idx = [1 3 4];
```

```
alma(idx)
```

```
idx2 = logical([1 0 1; 1 0 0]);
```

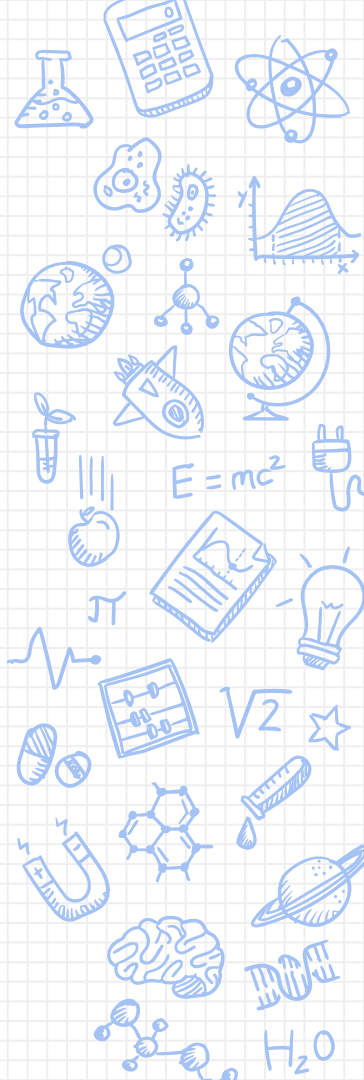
```
alma(idx2)
```

|    |    |    |    |
|----|----|----|----|
| 11 | 22 | 33 | 77 |
| 44 | 55 | 66 | 88 |
| 77 | 88 | 77 | 88 |

|    |
|----|
| 11 |
| 44 |
| 33 |

|    |    |    |
|----|----|----|
| 11 | 22 | 55 |
|----|----|----|

|   |   |   |
|---|---|---|
| 1 | 0 | 1 |
| 1 | 0 | 0 |



## Mátrixok, alpműveletek

- ✗ `size`, `ones`, `zeros`, `rand`, `diag`, `eye`
- ✗ `squeeze` - az 1 kiterjedésű (*singleton*) dimenziók eltávolítása

```
dio = ones(2, 1, 3); % 2 sor, 1 oszlop, 3 mélység
squeeze(dio)
```

|   |   |   |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 1 | 1 |

- ✗ `reshape` - átméretezés  
`B = reshape(A, m, n)`  
B-nek legyen m sora, n oszlopa;  
A-ból az elemek oszlopirányban/  
oszlopfolytonosan választódnak ki

- ✗ `numel` - a mátrix összes elemének száma

- ✗ `sum`, `prod`, `mean`, `min`, `max` - közös bennük: dimenzióként működnek (az első: oszlopirányban)

- ✗ logikai műveletek, logikai indexelés, elemenkénti műveletvégzés

```
alma = [1 2 3; 4 5 6]; % soronkénti
max(alma, [], 2) % maxkeresés
```

```
alma = [1 2 3; 4 5 6];
sum(alma)
```

|   |   |   |
|---|---|---|
| 5 | 7 | 9 |
|---|---|---|

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |



## Vezérlőszervezetek – elágazás

### IF

```
if feltetel
 parancsok
end
```

### ELSE

```
if feltetel
 parancsok1
else
 parancsok2
end
```

### ELSEIF

```
if feltetel1
 parancsok1
elseif feltetel2
 parancsok2
else
 parancsok3
end
```

- ✗ nincs zárójelezés, az adott blokk végét az end jelzi
- ✗ a feltétel logikai értékét vizsgálja (ami nem 0, az igaz)
- ✗ relációs operátorok: ==, ~=, <, >, <=, >=
- ✗ logikai operátorok: &, &&, |, ||, ~, xor, all, any



```
for n = 1:10
 parancsok
end
```

- ✗ ismeretlen számú iteráció elvégzésére, egy feltétel teljesüléséig
- ✗ nincs zárójelezés, a blokk végét az `end` jelzi
- ✗ nincs explicit módon megadott ciklusváltozó
- ✗ figyeljünk a végtelen ciklus elkerülésére!

```
while feltetel
 parancsok
end
```



```
in = input('Irj be egy számot: ');

switch in
 case 0
 disp('Nullat irtal be');
 case 1
 megoldas = 2*pi*exp(3.2);
 disp(megoldas);
 case {2, 5}
 disp('A bemenet 2 vagy 5');
 otherwise
 disp('Nem jo erteket adtal meg...');
end
```

- ```
gyak3_f31_[NEPTUN].m
gyak3_f32_[NEPTUN].m
```

$$E = mc^2$$

-

MATLAB 2017

4. gyakorlat

Lineáris egyenletrendszerek, leképezések



Lineáris egyenletrendszerek

$$\begin{cases} ax + by + cz = d \\ ex + fy + gz = h \\ ix + jy + kz = l \end{cases}$$



$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \end{cases}$$

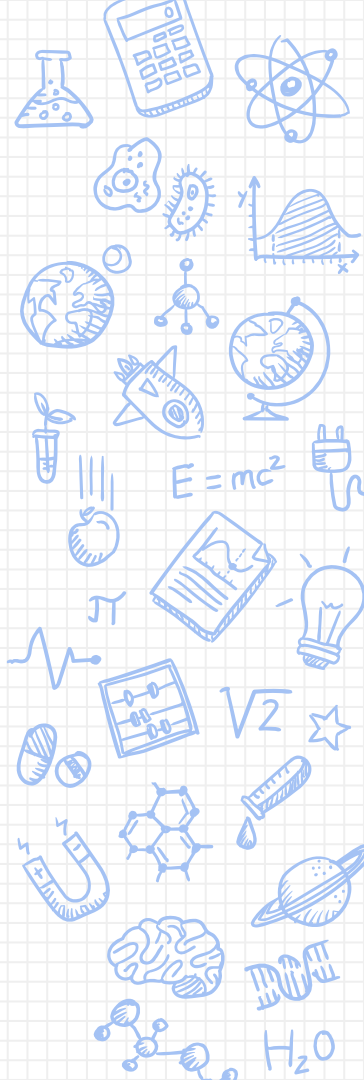

$$\underline{A} \underline{x} = \underline{b}$$



$$\underline{A}^{-1} \underline{A} \underline{x} = \underline{A}^{-1} \underline{b}$$



$$\underline{x} = \underline{A}^{-1} \underline{b}$$





Bal és jobb osztás, inverz

- ✗ A mátrixok körében a szorzás nem kommutatív, ezért kétféle osztás definiálható:

- ✗ bal osztás: $A \setminus B$ azt az X mátrixot jelenti, amelyre $A * X = B$ azaz $A \setminus B = A^{-1}B$

- ✗ jobb osztás: A / B azt az Y mátrixot jelenti, amelyre $Y * B = A$ azaz $A / B = AB^{-1}$

- ✗ MATLAB-ban van beépített inverz számoló: `inv(...)`, de ha kifejezetten lineáris egyenletrendszert akarunk megoldani, akkor sokkal célszerűbb a 'bal osztás' operátorával dolgoznunk:

$$A * x = b$$

$$x = A \setminus b$$

~~$$x = \text{inv}(A) * b$$~~

miért? *Gyorsabb, pontosabb* (Gauss-eliminációt használ az inverz létrehozása nélkül). MATLAB help-ben: `mldivide (\)`

Lineáris leképezések

- ✗ minden mátrix egy lineáris leképezésnek tekinthető;
- ✗ a gyakorlaton csak 2D leképezésekről lesz szó, pl: adott egy téglatest koordinátaival, nagyítsuk fel az y-tengely mentén

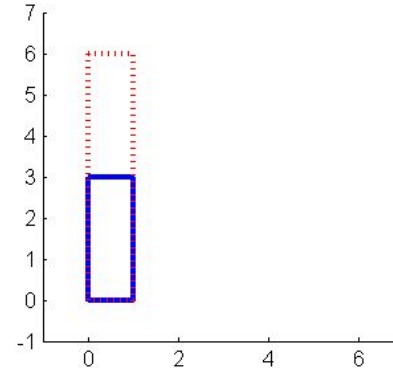
```
A = [1 0; 0 2];  
vonal = [0 3; 1 3; ...  
         1 0; 0 0; 0 3];  
vonal2 = vonal*A;
```

```
figure;  
hold on;  
plot(vonal(:, 1), vonal(:, 2), 'LineWidth', 3);  
plot(vonal2(:, 1), vonal2(:, 2), 'r:', 'LineWidth',  
3);  
xlim([-1 7]);  
ylim([-1 7]);
```

0	6
1	6
1	0
0	0
0	6

0	3
1	3
1	0
0	0
0	3

1	0
0	2



-



Feladatok

- ✗ egyrészt az új anyagrészhez: a feladatgyűjtemény **4.1 - 4.4** feladatai, melyeknek neve ez legyen, rendre:
gyak4_f41_[NEPTUN].m, ...,
gyak4_f44_[NEPTUN].m
(természetesen szögletes zárójelek nélkül).

Amivel nem végzel / nem végzünk, azt otthon kell befejezni, ez a házi feladat is egyben. A határidő vasárnap (március 12.) éjfél.

Feltöltés: users.itk.ppke.hu/~zseta/matlab2017/HF04/



5. gyakorlat

Polinomok, deriválás, integrálás, anonim függvények



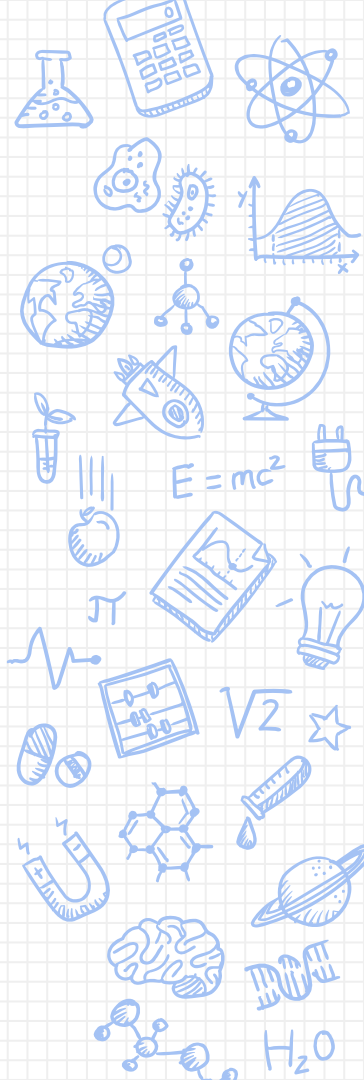
- $$\begin{aligned} \times \quad P_1(x) &= x^2 + 2x + 3 && \rightarrow P1 = [1 \ 2 \ 3]; \\ \times \quad P_2(x) &= 10x^3 + 4x^2 + 5x - 7 && \rightarrow P2 = [10 \ 4 \ 5 \ -7]; \\ \times \quad P_3(x) &= 3x^4 + 5x^2 - 12 && \rightarrow P3 = [3 \ 0 \ 5 \ 0 \ -12]; \end{aligned}$$

P_1, P_2, P_3 itt egy egyszerű sorvektor!

Polinomok

Az általunk megadott vektorokat a MATLAB megfelelő beépített függvényei fogják polinomként értelmezni:

- ✗ gyökök kiszámítása: `roots(P);`
- ✗ kiértékelés adott pontban: `y0 = polyval(P, x0);`
- ✗ kiértékelés sok pontban: `x = -1:0.01:1;`
`y = polyval(P, x);`
- ✗ polinom létrehozása a gyökeiből: `r = [-1 1];`
`P = poly(r);`
- ✗ polinom illesztése adatsorra:
`P = polyfit(x_t, y_t, fokszam);`

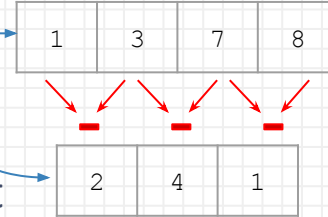


- ✗ **diszkrét értékeken történik, ezért fontos a jelek felbontása** (alacsony felbontással mintavételezve egy folytonos függvényt nem kapunk "kellőképpen" folytonos értékeket);
- ✗ a differenciálhatóság feltétele általában a függvény folytonossága (vannak persze kivételek);
- ✗ MATLAB-ban teljes folytonosságról nem beszélhetünk (minimális lépésköz ϵ_s), de megfelelő felbontást választva jó közelítéssel **numerikusan** is kiszámítható a derivált;
- ✗ **MATLAB-ban a derivált mindig differencia-hányados.**

Numerikus deriválás – differencia hányados számítása

- ✗ `diff(vektor);` % előállítja az elemenként vett különbségeket

```
alma = [1 3 7 15];  
diff(alma)
```

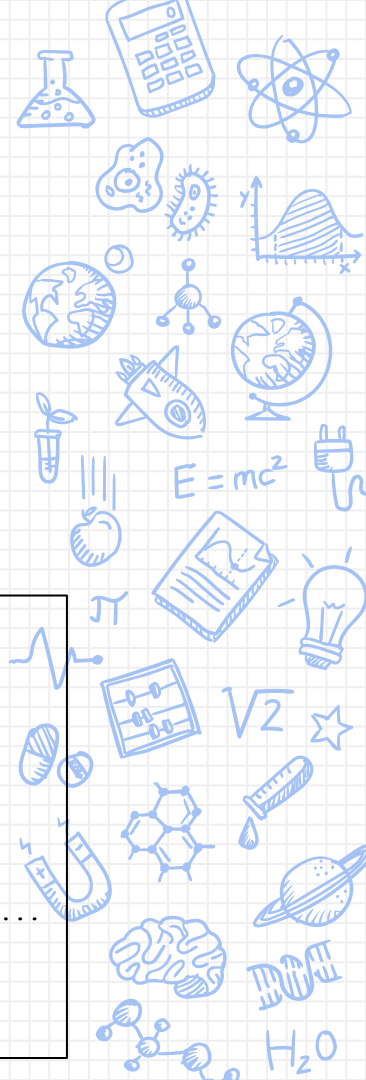


- ✗ állítsuk elő az értékkészlet és az értelmezési tartomány adatsorainak különbségét
`dy = diff(y); dt = diff(t);`

- ✗ és ezeknek vegyük a hányadosát:

```
differentiahanyados = dy ./ dt;
```

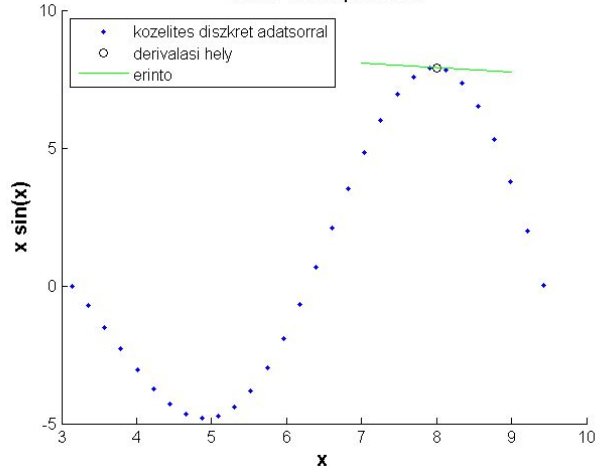
- ✗ megjegyzés: vegyük észre, hogy a különbségvektoroknak eggyel kevesebb eleme van: `length(t)` vs. `length(dt)` ezért ne is próbáljuk az értékkülönbségeket az eredeti ÉT-tartományhoz hozzárendelni pl kirajzoláskor: ~~`plot(t, dy, 'r.')`~~ helyett `plot(t(1:end-1), dy, 'r.')`



Deriválás adott pontban

- ✗ ha a vizsgált függvény képlettel felírható (pl. polinom, szögfüggvény, stb.), akkor adott hosszúságú és felbontású mintavétel nélkül is megadható egy adott pontban a derivált;
- ✗ ehhez a vizsgált pont tetszőlegesen kicsi környezetét megvizsgáljuk

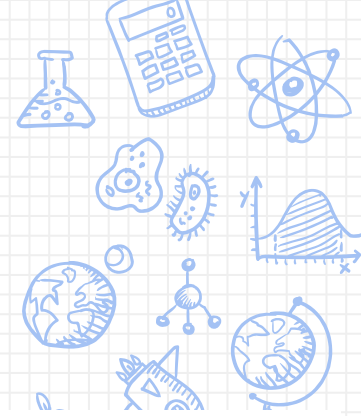
Erinto adott pontban



```
x = linspace(pi, 3*pi, 30); y = x .* sin(x);
x_p = 8; y_p = x_p*sin(x_p);
x_t = [8-1e-6, 8+1e-6]; y_t = x_t .* sin(x_t);
dx = diff(x_t); dy = diff(y_t);
differenciahányados = dy/dx;

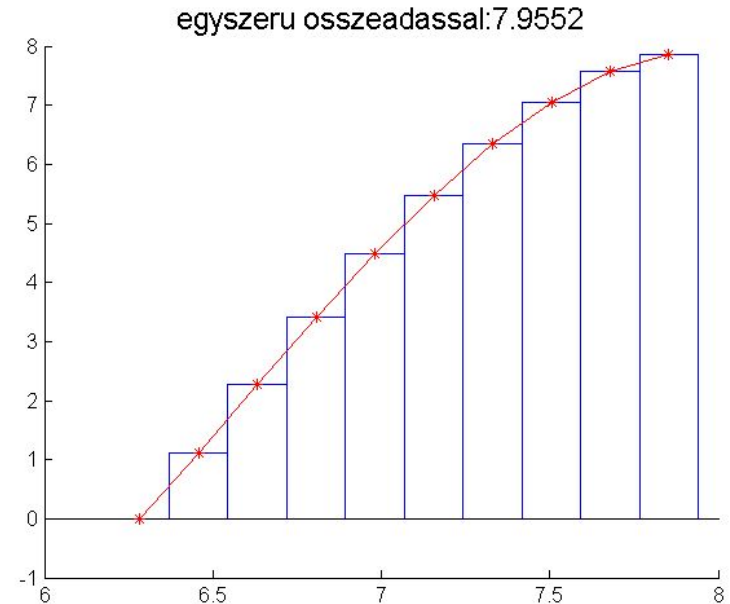
Figure; hold on;
plot(x, y, 'b. '); plot(x_p, y_p, 'ko');
plot([x_p-1, x_p+1], [y_p-differenciahányados, ...
    y_p+differenciahányados], 'g');
legend('kozelites diszkrét adatsorral', 'derivalasi hely', ...
    'Erinto', 'Location', 'NorthWest');
xlabel('x', 'FontSize', 12, 'FontWeight', 'bold');
ylabel('x sin(x)', 'FontSize', 12, 'FontWeight', 'bold');
title('Erinto adott pontban', 'FontSize', 14);
```

- ✘ a deriváláshoz hasonlóan lehet vektorértékek és megadott függvény alapján is integrálni (*integrál := a függvényértékek és az x-tengely közötti területe részek előjeles összege*);
- ✘ lehetőségek:
 - a. egyszerű összeadással,
 - b. trapékszabály alkalmazásával,
 - c. megadott függvény alapján



Numerikus integrálás --- a) egyszerű összeadással

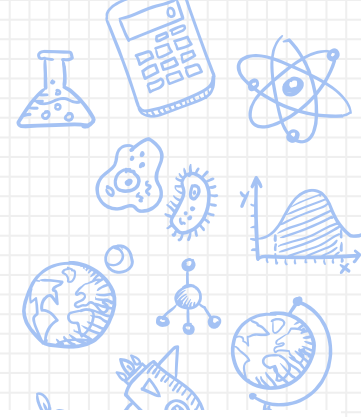
```
x = linspace(2*pi, 2.5*pi, 10);  
y = x .* sin(x);  
% egyszeru osszeadassal:  
szelesseg = x(2)-x(1);  
osszeg_1 = szelesseg*sum(y);  
  
figure;  
hold on;  
bar(x, y, 1, 'w', 'EdgeColor', 'b', ...  
    'LineStyle', '-');  
plot(x, y, 'r*-');  
title(strcat('egyszeru osszeadassal: ', ...  
    num2str(osszeg_1, 5)), 'FontSize', 14);
```



$$T_{trapez} = (q - p) \frac{f(p) + f(q)}{2}$$

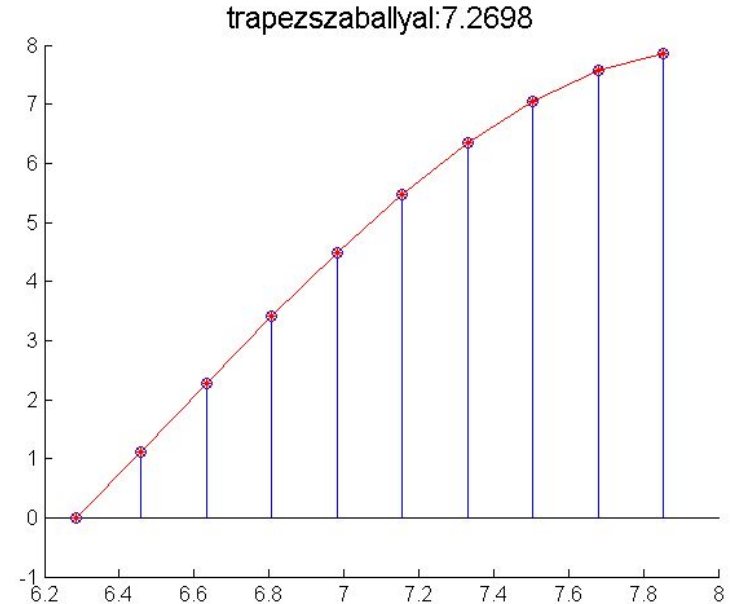
A graph illustrating the trapezoidal rule. The area under a curve is approximated by a trapezoid. The x-axis is labeled with p and q , and the y-axis with $f(p)$ and $f(q)$. The trapezoid is shaded orange.

$$\begin{aligned}\int_a^b f(x) dx &\approx \frac{b-a}{2N} \sum_{n=1}^N (f(x_n) + f(x_{n+1})) \\ &= \frac{b-a}{2N} [f(x_1) + 2f(x_2) + \dots + 2f(x_N) + f(x_{N+1})]\end{aligned}$$



Numerikus integrálás --- b) trapézsabállyal

```
x = linspace(2*pi, 2.5*pi, 10);  
y = x .* sin(x);  
% trapezszabállyal:  
osszeg_2 = trapz(x, y);  
  
figure;  
hold on;  
stem(x, y);  
plot(x, y, 'r*-');  
title(strcat('trapezszabállyal: ', ...  
    num2str(osszeg_2, 5)), 'FontSize', 14);
```



Anonim függvények

- ✗ a MATLAB lehetőséget ad arra, hogy függvényeket “tároljunk” változóknban, (ha azok kellőképpen egyszerűek);
- ✗ a konstrukció:

`fv = @(x) x+3;`

függvéynév -
változónév

bemenő paraméter(ek)től
függő kifejezés,
a függvény törzse

bemenő paraméter(ek)

```
>> fv = @(x) sin(x)-2*x^2+3*x
fv =
      @(x) sin(x)-2*x^2+3*x

>> fv(3)
ans =
    -8.8589
```

```
>> P = [2 0 3];
>> fv2 = @(x) polyval(P, x)
fv2 =
      @(x) polyval(P,x)

>> fv2(10)
ans =
    203
```



Numerikus integrálás --- c) függvényel

```
x = linspace(2*pi, 2.5*pi, 10);  
fv = @(t) t .* sin(t);  
% függvényel:  
osszeg_3 = integral(fv, x(1), x(end));  
  
fprintf(['Elojeles osszegek:\n\tegyszeru osszeadassal: %.4f\n\ttrapezszaballyal: ' ...  
        '%.4f\n\tfuggvennyel: %.4f\n'], osszeg_1, osszeg_2, osszeg_3);
```

```
Elojeles osszegek:  
egyszeru osszeadassal: 7.9552  
trapezszaballyal: 7.2698  
fuggvennyel: 7.2832
```

megjegyzés:
régebbi MATLAB-ban és/vagy linux-os
*kiadás alatt: **integral** helyett **quad***



Numerikus integrálás --- összegzés

- ✗ integrálásnál a **sima összeadást** lehetőleg **ne használjuk**;
- ✗ tetszőleges **vektoros adatsorokhoz**: **trapézszabály**;
- ✗ anonim **függvényekkel felírható görbékhez**: **integral** függvény.



- ```
gyak5 f51 [NEPTUN].m ...
```

```
gyak5 f57 [NEPTUN].m
```

(természetesen szögletes zárójelek nélkül).

- ✗ a diasorban ismertetett parancsok kikeresése és tanulmányozása a Help-ben

Amivel nem végzel / nem végzünk, azt otthon kell befejezni, ez a házi feladat is egyben. A határidő vasárnap (március 19.) éjfél.

Feltöltés: [users.itk.ppke.hu/~zseta/matlab2017/HF05](https://users.itk.ppke.hu/~zseta/matlab2017/HF05)

# MATLAB 2017

## 6. gyakorlat

# Differenciálegyenletek



- ✗ **Diffegyenlet: Olyan egyenlet, amelyben az ismeretlen egy függvény, és szerepel benne ennek az ismeretlen függvénynek valamely deriváltja is.**
- ✗ **A diffegyenlet rendje: az ismeretlen függvény legmagasabb fokú deriváltjának fokszáma (első és másodrendűről lesz szó).**
- ✗ **MATLAB-ban a diffegyenletek megoldása numerikus integrálással történik.**



**X**

**X**

**X**

a függvény valamilyen deriváltját tartalmazó függvény

$$f'(t) = g(f(t), t)$$

## Szemléletesen

- ✗ Tegyük fel, hogy egy hegyi úton sétálunk, és az aktuális magasságunk az előrehaladás közben változik.
- ✗ A magasságunkat felírhatjuk pl. az idő, a hosszúsági és szélességi kör vagy a megtett út függvényében is.
- ✗ A megtett út függvényében a magasságra a következő összefüggés írható fel:

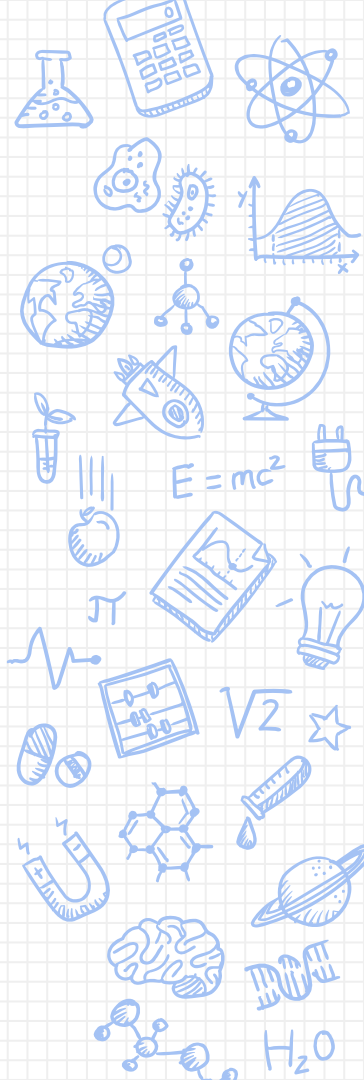
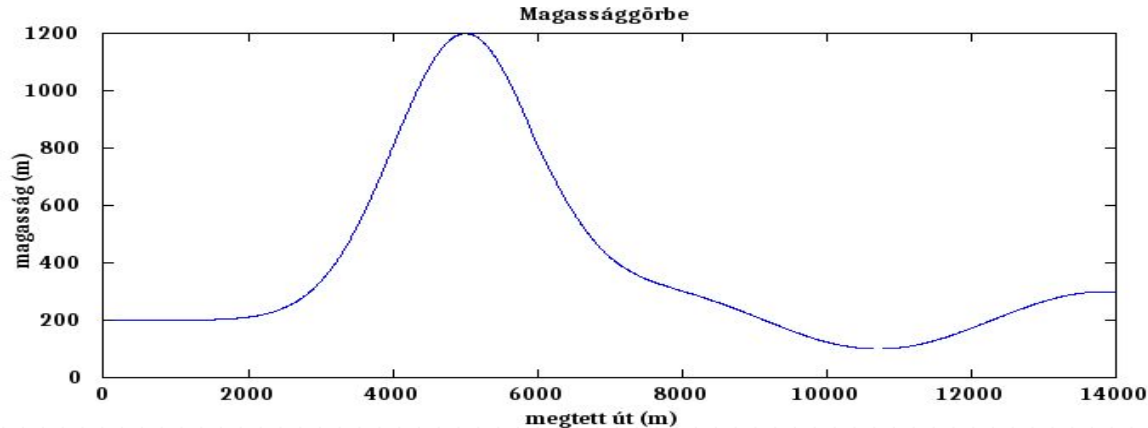
$$y = y(x),$$

ahol  $x$  az út,  $y$  pedig a magasság.



## Szemléletesen

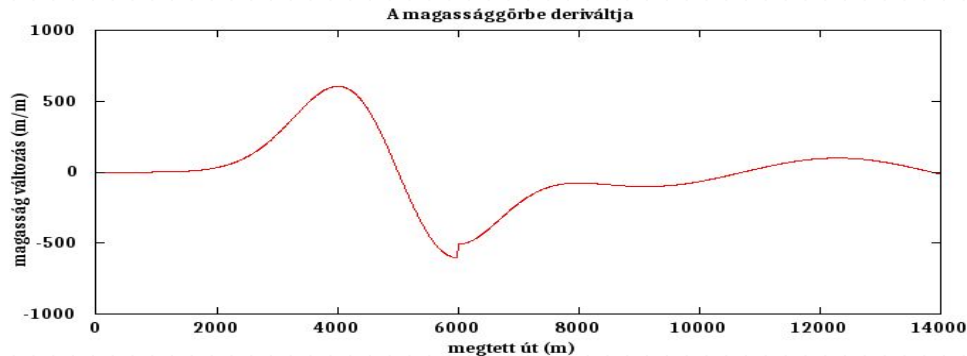
- ✗ Ha van nálunk magasságmérő vagy GPS vevő, az előrehaladás közben elegendő ponton felírva az aktuális magasságértékeket megkapjuk az  $y = y(x)$  összefüggés értékeit (pl. a lenti ábra szerint).



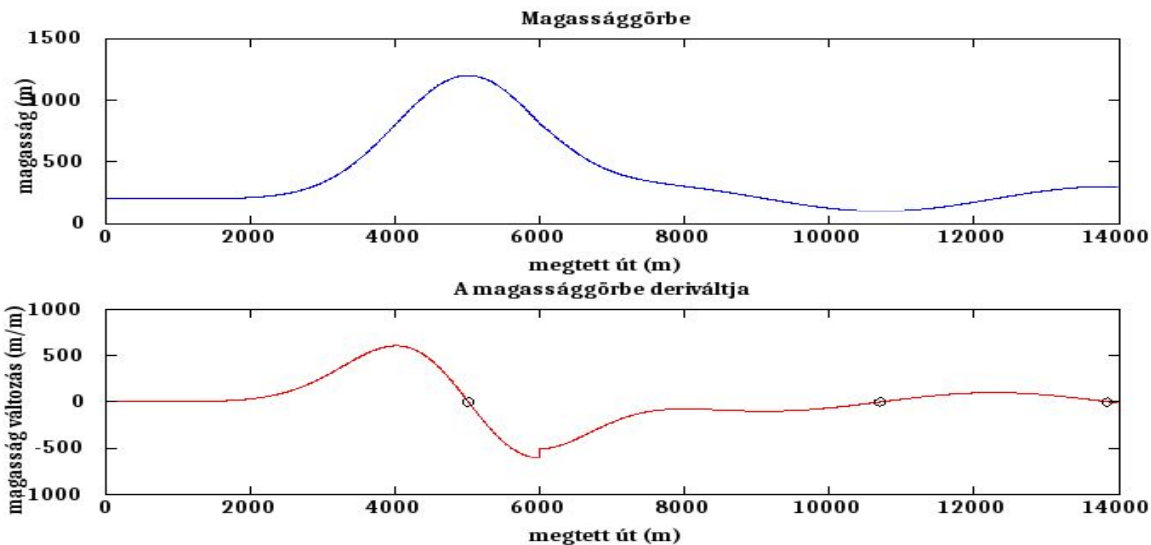
- $$\frac{y(x+h) - y(x)}{h} = 0.05$$
- ✗ ahol az összefüggés bal oldala az út **meredeksége**  $x$  és  $x+h$  között.

## Szemléletesen

- ✗ Tegyük fel, hogy az út mentén pár méterenként találunk egy ilyen táblát, melyek a magasság változásának közelítő értékeit adják meg a megtett út függvényében.
- ✗ Ha ezeket az értékeket felírjuk, megkapjuk a  $dy/dx$  összefüggést, ami  **$y$  deriváltja** lesz.



- 



- ✗ Legyen adott a következő elsőrendű differenciálegyenlet:  
 $y'(t) = 2y(t)$ .
- ✗ Adjuk meg  $y(t)$  értékeit a  $t = [0, 3]$  intervallumon,  $y(0) = 1$  kezdeti érték esetén!

## 1. példa – explicit Euler

maga a diffegyenlet:  $\dot{y}(t) = f(t, y(t))$

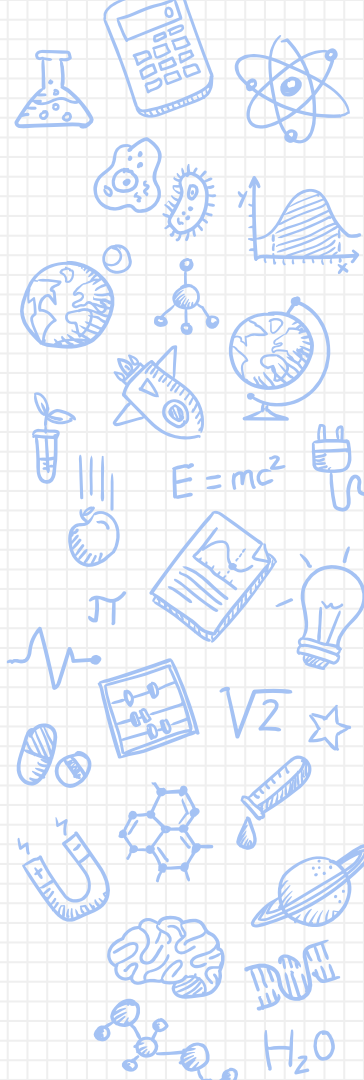
kezdeti érték:  $y(t_0) = y_0$

lépesköz:  $h$

időskala (n+1)-edik tagja:  $t_{n+1} = t_n + h$

ahol a megoldás:  $y_{n+1} = y_n + hf(t_n, y_n)$

```
function [t_out, y_out] = explicitEuler(F, tspan, y0)
% Egyszerű differencialegyenlet-megoldó, explicit Euler
% módszer alapján.
% Csak szemleltetési célú, ne használjuk később, mert
% pontatlan.
% F: deriváltfüggvény
% tspan: időskala (első és utolsó pontja)
% y0: kezdeti érték
...
end
```





- ✗ **F** a deriváltfüggvény értékeit tartalmazza, a **t** paraméter a beépített megoldók miatt (**ode45**, **ode23**, **ode15**) kell.

$$F = @ (t, y) \quad 2 * y;$$

## 1. példa – explicit Euler

- ✗ Írjunk egy saját differenciálegyenlet megoldó eljárást (`explicitEuler`), amely az Euler módszert alkalmazva,  $\mathbf{F}$  numerikus integrálásával kiszámolja  $\mathbf{y}(t)$  értékeit a fent megadott intervallumon és kezdeti értékkel.
- ✗ 200 lépéssel dolgozzunk, így az integrálás lépésközét  $(\text{intervallum\_hossza})/200$ -nak válasszuk meg.
- ✗ **FONTOS:** a saját megoldó csak szemléltetési célt szolgál, a későbbi feladatok megoldásakor **mindig** a beépített `ode45` megoldót használjuk!



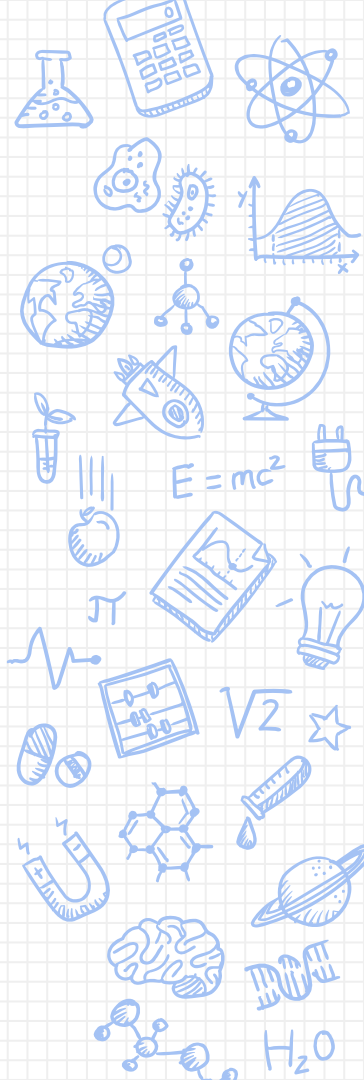
## 1. példa – explicit Euler

✗ Hívjuk meg a függvényt és rajzoljuk ki az eredményt!

```
% függvény definíció
F = @(t,y) 2*y;

% megoldás
[t1,y1] = explicitEuler(F,[0 3],1);

% rajzoljuk ki
figure(1); hold on;
plot(t1,y1,'r-');
```



## 1. példa – explicit Euler

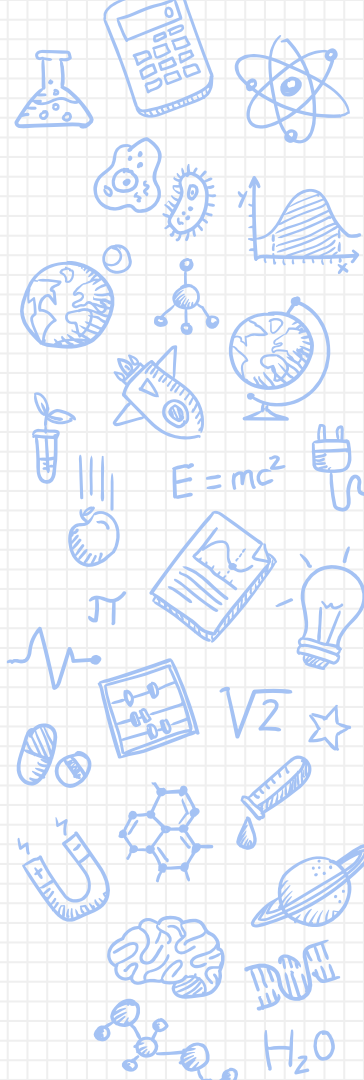
- ✗ Nézzük meg ugyanezt a beépített `ode45` megoldó használatával is:

```
% beépített megoldó eljárás
[t45,y45] = ode45(F,[0 3],1);

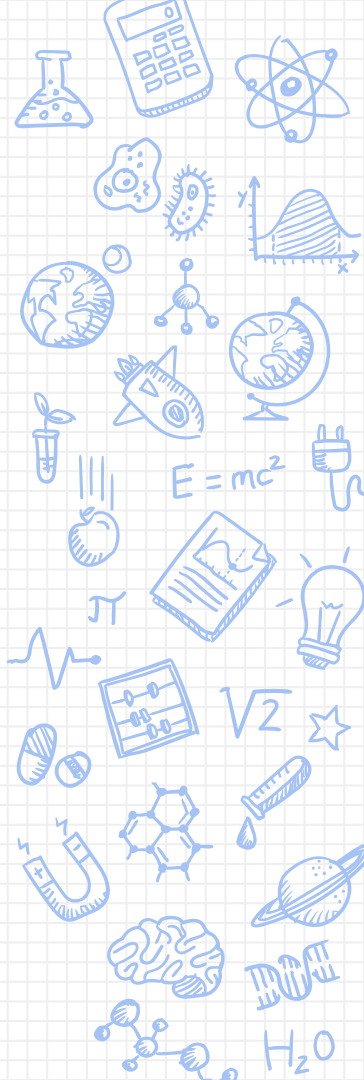
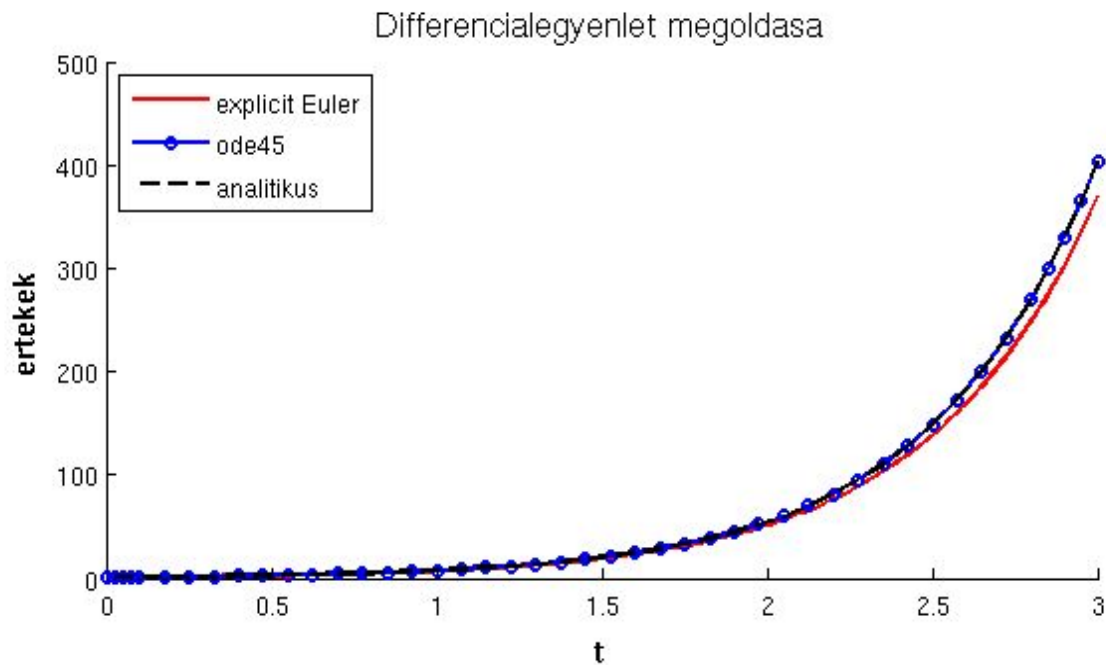
% rajzoljuk ki
plot(t45,y45,'bo-');
```

- ✘ Analízisből ismert, hogy az  $y'(t) = 2y(t)$  differenciálegyenlet megoldása  $y(t) = e^{2t}$ , ezért ellenőrzésként rajzoljuk ki ezt is:

```
plot(t45,exp(2*t45),'k--','LineWidth',2);
```



## 1. példa – explicit Euler



## Konklúzió – explicitEuler vs ode45

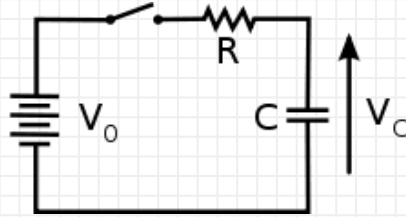
---

- ✗ A beépített `ode45` megoldó nem lineárisan osztja el a "mintavételi" időpontokat (ezért kell a  $t$  paraméter a deriváltfüggvény megadásánál).
- ✗ A lépésköz meghatározása minden esetben egy előre meghatározott pontosság elérése érdekében történik.
- ✗ A legtöbb problémára az `ode45` a legjobb választás, ezért ezt fogjuk használni.



## 2. példa – áramkör (egyváltozós, elsőrendű)

- x** Vegyünk egy egyszerű töltőáramkört az alábbi ábra alapján:



- ✗ ahol  $V_0 = 2 \text{ V}$ ,  $R = 1 \text{ k}\Omega$ ,  $C = 500 \text{ }\mu\text{F}$  és tudjuk, hogy  $\tau = RC$  (időállandó).
- ✗  $t = 0$ -ban a kapacitáson nincs töltés és a kapcsoló nyitva van

- x**  $t = 0$ -ban a kapacitáson nincs töltés és a kapcsoló nyitva van

## 2. példa – áramkör (egyváltozós, elsőrendű)

- ✗ A kapcsoló bekapcsolásakor a kapacitáson átfolyó áram alakulása az alábbi differenciálegyenlettel írható le, ( $V_0/R$  kezdeti értékkel):

$$i'(t) = -\frac{1}{\tau} i$$

- ✗ Analitikus alakban pedig az alábbi képlettel adható meg:

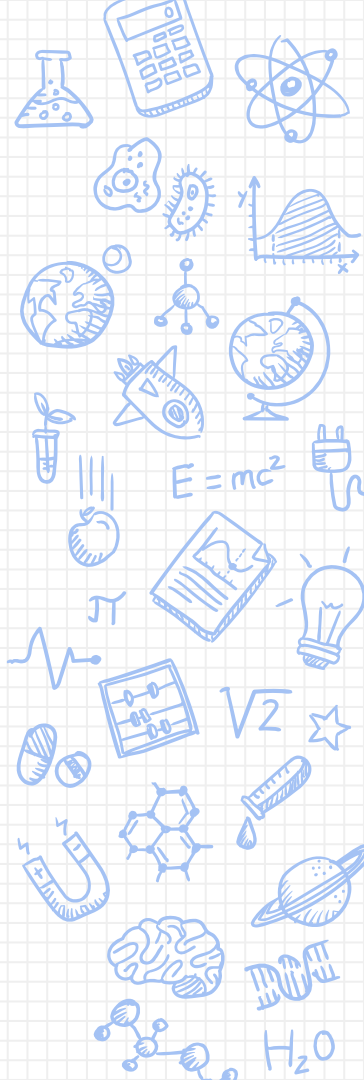
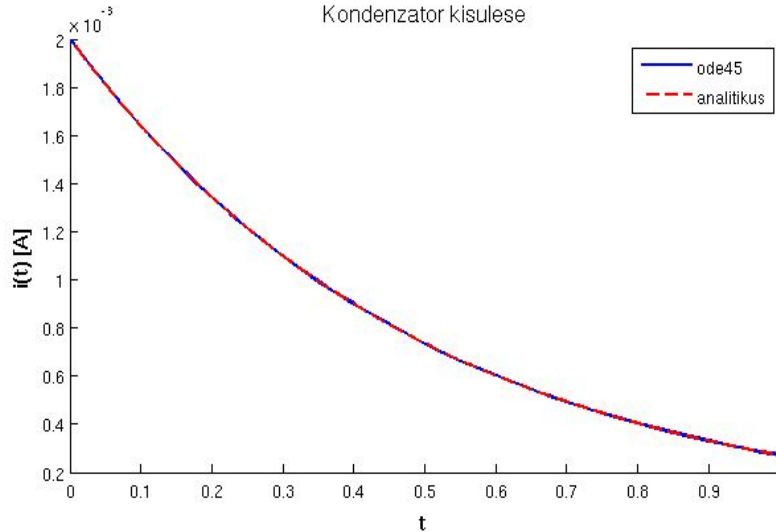
$$i(t) = \frac{V_0}{R} e^{-\frac{t}{\tau}}$$



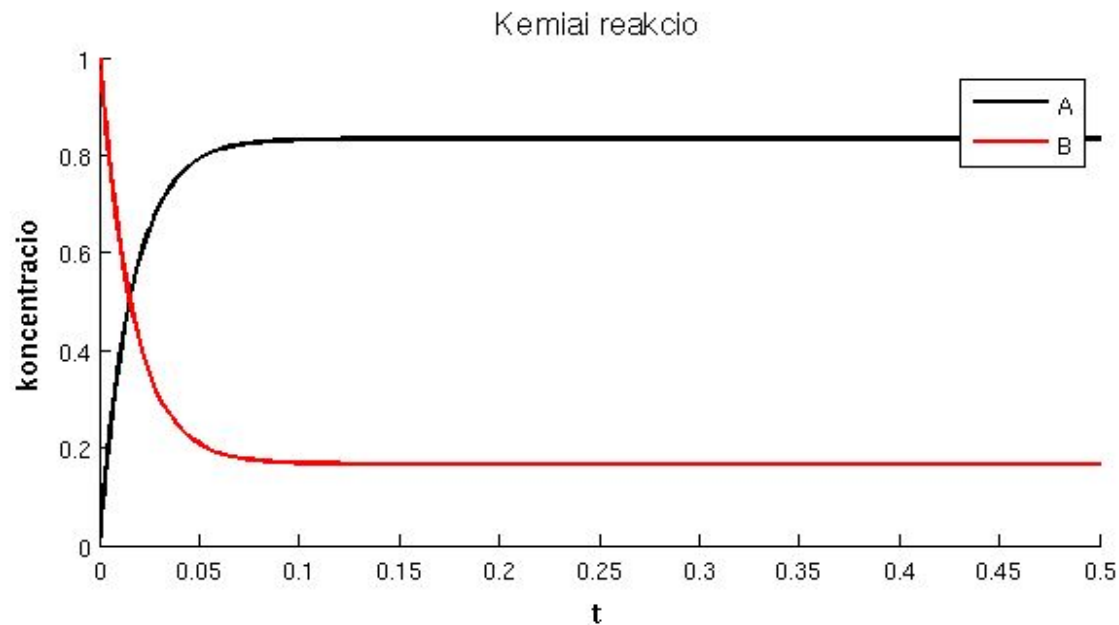


## 2. példa – áramkör (egyváltozós, elsőrendű)

- ✗ Számítsuk ki (diffegyenlettel és analitikusan) és ábrázoljuk a fent leírt áramkörben a kapacitás áramának időbeli változását:





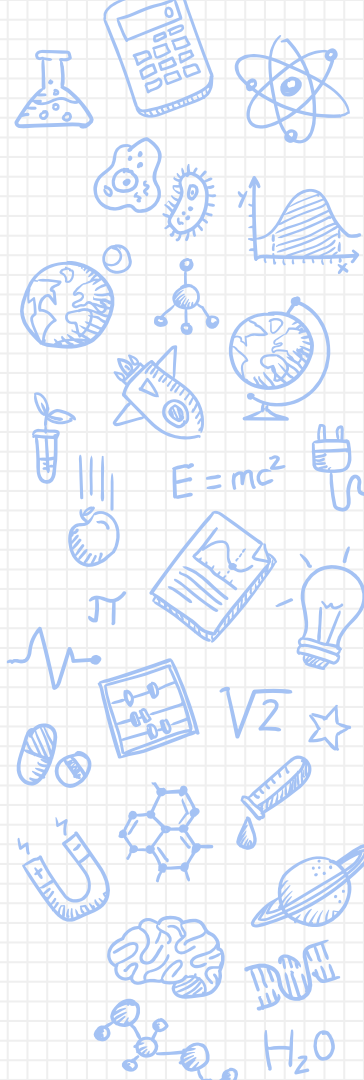


#### 4. példa – rezgőmozgás (egyváltozós, másodrendű)

- ✗ Rezgőmozgás során az erők egyensúlyát az alábbi összefüggés adja meg:

$$mx'' + Dx + Cx' = F$$

- ✗ ahol  $x$  a test kitérése,  $m$  a test tömege,  $D$  a rugóállandó,  $C$  a csillapítási tényező,  $F$  pedig külső erő.
- ✗ Az állapotvektor  $[y_1, y_2]$  legyen:  $y_1 = x$  (kitérés)  
 $y_2 = x'$  (sebesség)
- ✗ Ekkor a másodrendű egyenlet két elsőrendűvel megoldható.



## 4. példa – rezgőmozgás (egyváltozós, másodrendű)

```
function ydot=rugoe egyenlet(t, x, m, D, C, F)
% Masodrendu diffegyenlet megoldasa:
% szetszedjuk ket elsorendure
% Az allapotvektor y=[y1;y2] alaku,
% ahol y1=kiteres, y2=sebesseg:
% Az allapotvektor derivaltjai
ydot = zeros(2,1);
% ahol ydot(1) maga a sebesseg
ydot(1)=x(2);
% es ydot(2) pedig a gyorsulasra
% rendezett egyenlet
ydot(2)=-D/m*x(1) -C/m*x(2) +F/m;
end
```

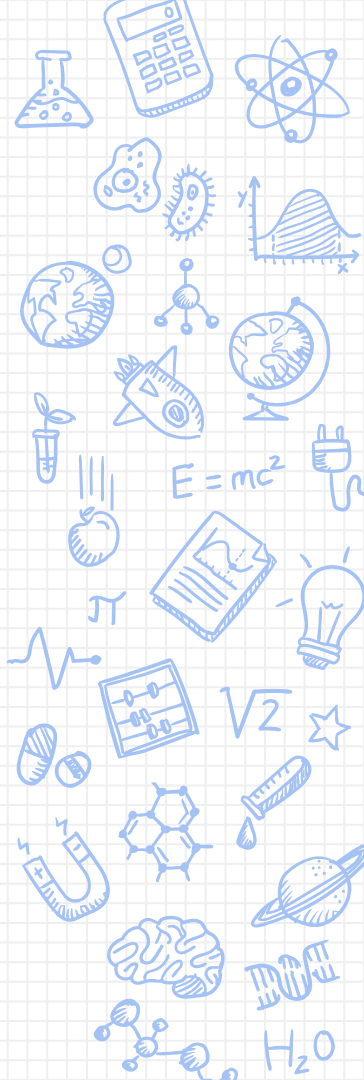
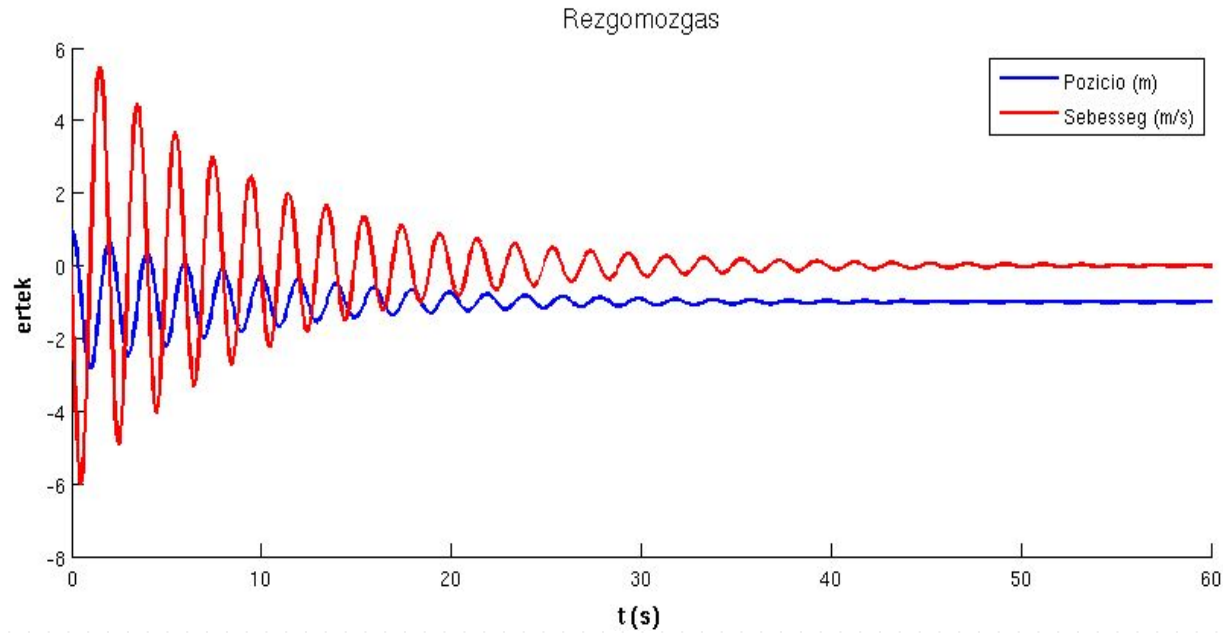




#### 4. példa – rezgőmozgás (egyváltozós, másodrendű)

- ✗ Számítsuk ki az alábbi paraméterekkel rendelkező rendszer rezgőmozgásának időbeli lefutását a  $t = [0 \ 60]$  intervallumon:
  - $m = 1 \text{ [kg = Ns}^2/\text{m]}$
  - $D = 10 \text{ [N/m]}$
  - $C = 0.2 \text{ [Ns/m]}$
  - $F = -10 \text{ [N]}$
- ✗ Ábrázoljuk a rugóra rögzített test kitérésének és sebességének időbeli változását.

## 4. példa – rezgőmozgás (egyváltozós, másodrendű)





## Feladatok

---

- ✗ a feladatgyűjtemény **6.1, 6.2** feladatai, melyeknek neve ez legyen, rendre:  
gyak6\_f61\_[NEPTUN].m  
gyak6\_f62\_[NEPTUN].m  
(természetesen szögletes zárójelek nélkül).
- ✗ a diasorban ismertetett parancsok kikeresése és tanulmányozása a Help-ben

Amivel nem végzel / nem végzünk, azt otthon kell befejezni, ez a házi feladat is egyben. A határidő vasárnap (április 2.) éjfél.

Feltöltés: [users.itk.ppke.hu/~zseta/matlab2017/HF06](https://users.itk.ppke.hu/~zseta/matlab2017/HF06)



# MATLAB 2017

## 7. gyakorlat

### 3D ábrázolás





## 3D ábrázolás

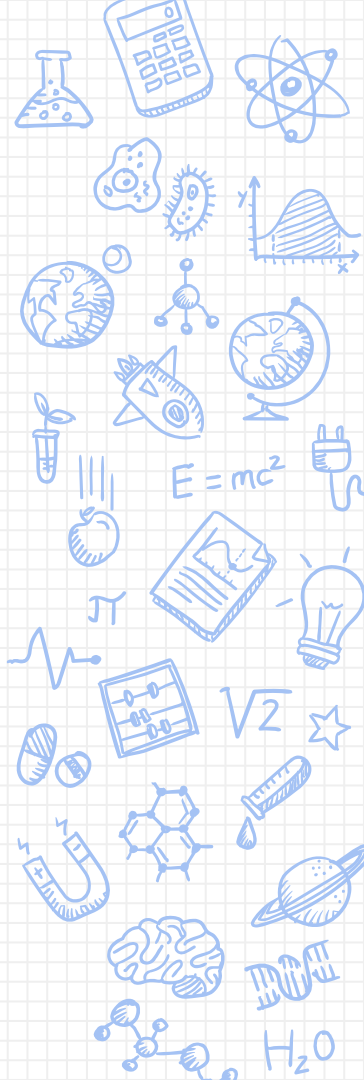
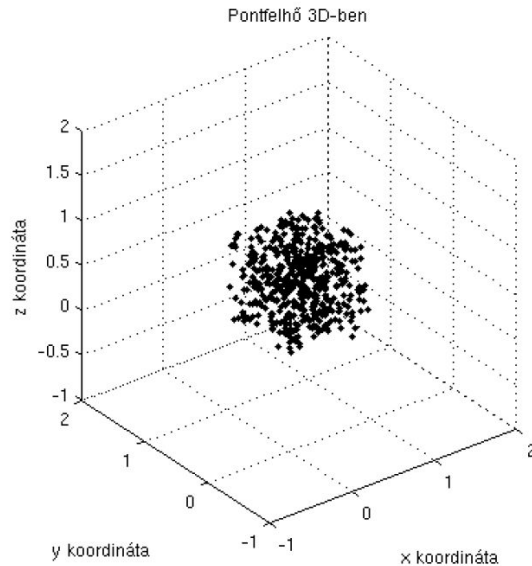
---

- ✗ A MATLAB beépített függvényekkel lehetőséget biztosít pontfelhők, görbék és felületek hatékony térbeli ábrázolására.
- ✗ A *plot*-hoz hasonlóan ezek is **értelmezési tartomány(ÉT) értékkészlet(ÉK)** alakban működnek, ahol **ÉT** és **ÉK** is lehet  $\mathbf{R}$ ,  $\mathbf{R}^2$ ,  $\mathbf{R}^3$  (az ábrázolt alakzat függvényében, részletek később)
- ✗ jellemzően 3D-ben ábrázolt alakzatok:
  - ✗ pontfelhők( $x, y, z$  koordináták)
  - ✗ parametrikus térgörbék( $\mathbf{R} \rightarrow \mathbf{R}^3$ )
  - ✗ felületek( $\mathbf{R}^2 \rightarrow \mathbf{R}$ )
  - ✗ vektormezők( $\mathbf{R}^2 \rightarrow \mathbf{R}^2$ ,  $\mathbf{R}^3 \rightarrow \mathbf{R}^3$ )

- ✗ Hozzunk létre egy 0 és 1 közötti véletlen értékeket tartalmazó  $500 \times 3$  elemű mátrixot!
- ✗ A kapott értékeket 500 db 3-dimenziós pontként értelmezve jelenítsük meg a mátrix által leírt pontfelhőt!
- ✗ új parancs: `plot3`

# Megoldás

```
Pontfelho = rand(500,3);
figure(1)
plot3(Pontfelho(:,1),Pontfelho(:,2),Pontfelho(:,3), 'k.')
title('Pontfelho 3D-ben')
xlabel('x koordinata')
ylabel('y koordinata')
zlabel('z koordinata')
xlim([-1 2]);ylim([-1 2]); zlim([-1 2])
grid on
```



## Felületek ábrázolása

- ✗  $\mathbb{R}^2 \rightarrow \mathbb{R}$  leképezés
- ✗ az értelmezési tartományhoz kell egy térháló, amelyen kiszámoljuk az adott felület értékeit
- ✗ **meshgrid** – általunk megadott tartományon, tetszőleges felbontással létrehoz egy térhálót
- ✗ rajzoló parancsok: **surf**, **mesh**, **contour**, **surfc**, **meshc**, **quiver**

## Szemléletesen

```
>> [X,Y] = meshgrid(2:4,1:5)
```

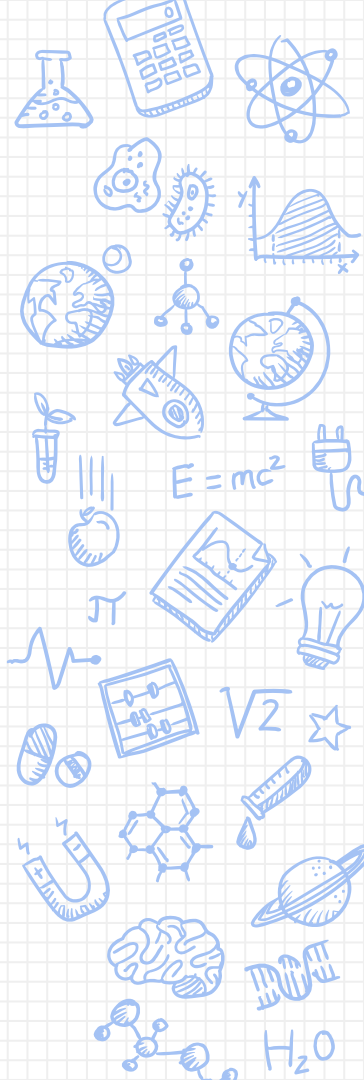
X =

|   |   |   |       |       |       |
|---|---|---|-------|-------|-------|
| 2 | 3 | 4 | (2,1) | (3,1) | (4,1) |
| 2 | 3 | 4 | (2,2) | (3,2) | (4,2) |
| 2 | 3 | 4 | (...) |       |       |
| 2 | 3 | 4 |       |       |       |
| 2 | 3 | 4 |       |       |       |

Y =

|   |   |   |       |       |       |
|---|---|---|-------|-------|-------|
| 1 | 1 | 1 | (...) |       |       |
| 2 | 2 | 2 |       |       |       |
| 3 | 3 | 3 | (2,5) | (3,5) | (4,5) |
| 4 | 4 | 4 |       |       |       |
| 5 | 5 | 5 |       |       |       |

Az adott rácsponton lévő x és y **értékeket** adja vissza X-ben és Y-ban.





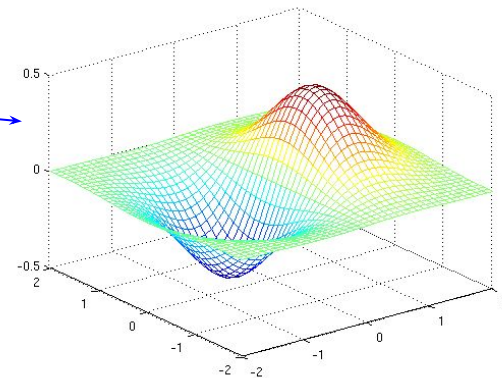
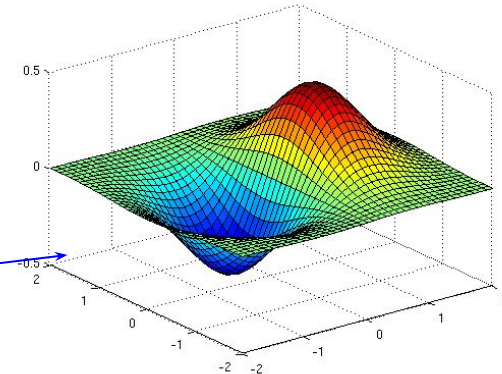


## Megoldás

```
% a térháló létrehozása
[X,Y] = meshgrid(-2:0.1:2);
% a függvény értékeinek
% kiszámítása
Z = X.*exp(-X.^2-Y.^2);

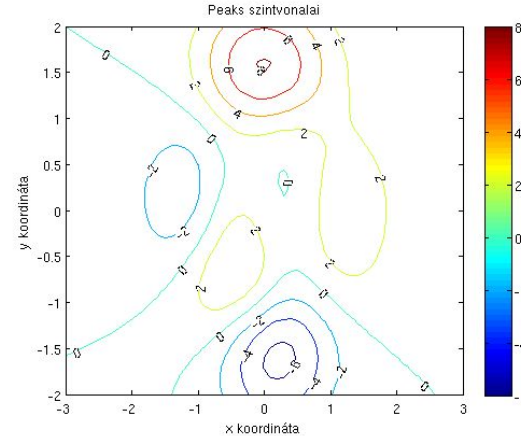
% kirajzolás
figure;
surf(X,Y,Z);

figure;
mesh(X,Y,Z);
```

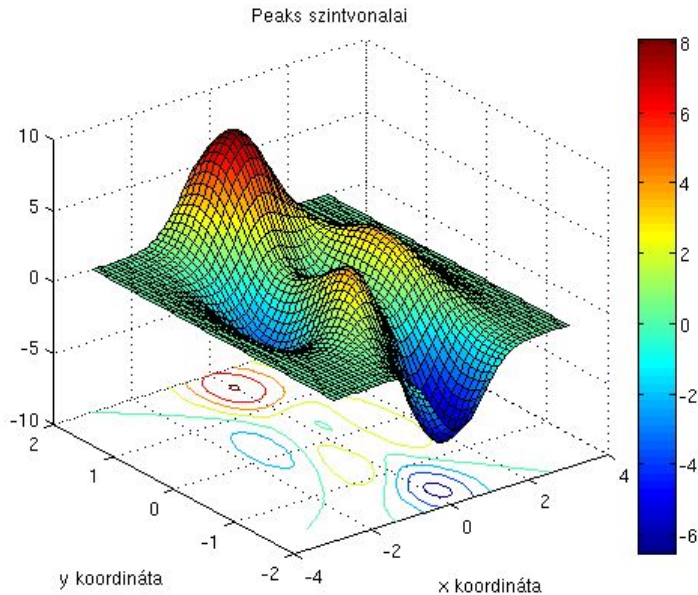


- ✗ Egy felület azonos nagyságú (azonos “magasságon” lévő) értékeit tartalmazó görbe.
- ✗ csak szintvonal kirajzolása: **contour**, (`contour3`, `contourf`)
- ✗ felület és szintvonal kirajzolása: **surf**, **meshc**
- ✗ szintvonalak feliratozása: **clabel**
- ✗ színskála: **colorbar**

- ```
[X Y] = meshgrid (-3:0.1:3, ...  
-2:0.1:2);  
Z = peaks(X,Y);  
C = contour(X,Y,Z);  
colorbar  
clabel(C)  
title('Peaks szintvonalai')  
xlabel('x koordinata')  
ylabel('y koordinata')
```



- ```
[X Y] = meshgrid(-3:0.1:3,...
-2:0.1:2);
surf(X,Y,Z);
title('Peaks szintvonalai')
xlabel('x koordinata')
ylabel('y koordinata')
colorbar
```





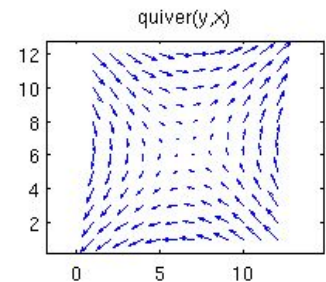
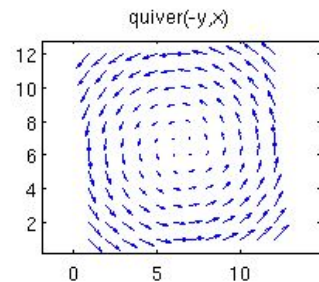
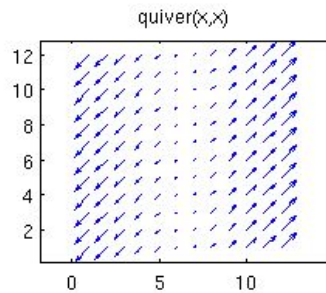
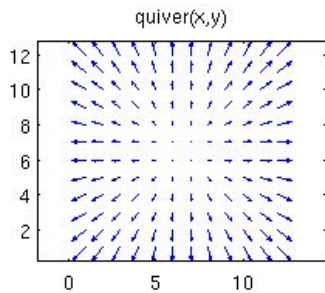
### 3.3 példa - view használata

- ✗ A view parancs használatával állítsuk be az előző ábrán a nézőpontot  $Az=0$ ,  $El=30$ -ra! (“azimuth” és “elevation” szögek)
  - ✗ Azimuth: z-tengely körüli horizontális forgatás fokban, negatív y tengelytől kezdődik, óramutató járásával ellenkező irányban forgat
  - ✗ Elevation (magasság):
    - + az ábrát fölülről nézzük
    - - az ábrát alulról nézzük
- ✗ Írjunk egy ciklust, amelyben 0.1 másodperces késleltetéssel elforgatjuk a nézőpontot  $Az=0$ ,  $El=30$  és  $Az=180$ ,  $El=30$  között, a lépéseket egyenletesen elosztva!

```
El = 30;
for Az = 0:180
 view(Az,El)
 pause(0.1)
end
```

## 4. példa - vektormezők

- ✗ a legtöbbször  $\mathbb{R}^2 \rightarrow \mathbb{R}^2$  vagy  $\mathbb{R}^3 \rightarrow \mathbb{R}^3$
- ✗ vektormező kirajzolása: **quiver**, quiver3



```
[x,y] = meshgrid(-1.1:.2:1.1);
subplot(221);
quiver(x,y);
axis equal;
title('quiver(x,y)');
subplot(222);
quiver(x,x);
axis equal;
title('quiver(x,x)');
subplot(223);
quiver(-y,x);
axis equal;
title('quiver(-y,x)');
subplot(224);
quiver(y,x);
axis equal;
title('quiver(y,x)');
```



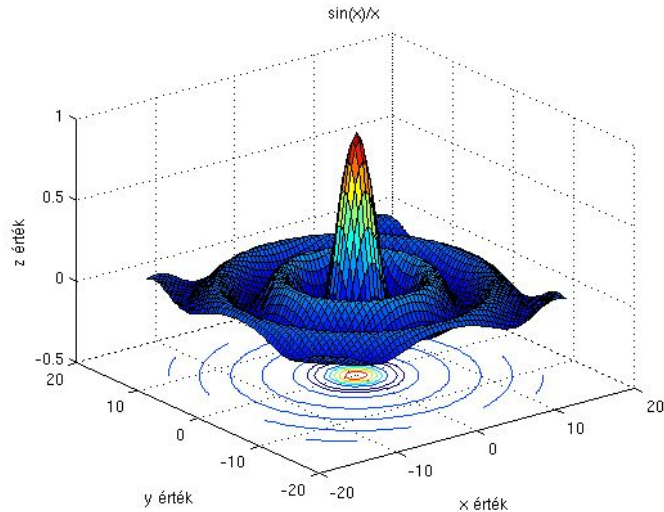
# Gyakorlófeladatok megoldással





## Gyakorló Példa 1.

Ábrázoljuk a  $\sin(x)/x$  függvényt a szintvonalalaival együtt a  $[-15, 15] \times [-15, 15]$  intervallumon 0.5-ös felbontással, körkörösén minden irányban!



```
% térháló létrehozása
[X,Y] = meshgrid(-15:0.5:15);

% a körkörös irány miatt
% a sugár létrehozása
R = sqrt(X.^2+Y.^2) + eps;

% a fv. értékeinek kiszámítása
Z = sin(R)./R;

figure(7);
surf(X,Y,Z);

title('sin(x)/x');
xlabel('x érték');
ylabel('y érték');
zlabel('z érték');
```

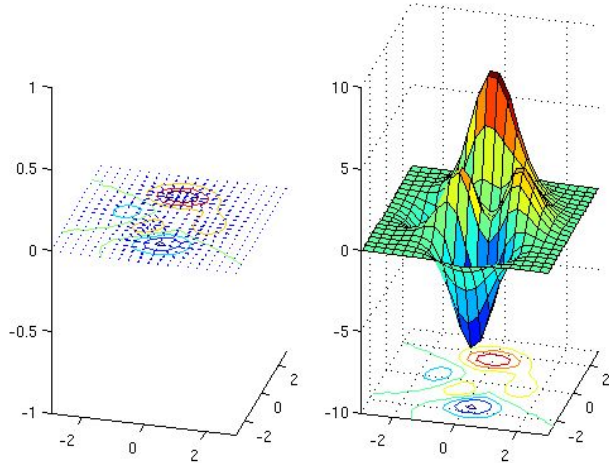




## Gyakorló Példa 2. – gradiens mező

Számítsuk ki a **peaks(20)** parancs által megadott felület gradiens mezőjét!

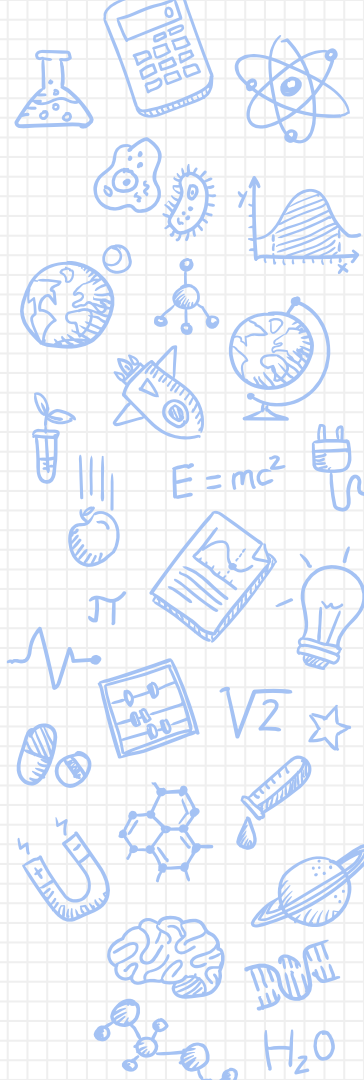
Ábrázoljuk a felületet és a gradiens mezőt egymás mellett elhelyezkedő subplotokon, mindegyiken kirajzolva a szintvonalakat is!



```
% a felület pontjainak létrehozása
[X,Y,Z] = peaks(20);
% gradiens kiszámítása
[PX,PY] = gradient(Z);
```

```
% kirajzolás
figure
subplot(121)
hold on
% a gradiens ábrázoló vektormező
quiver(X,Y,PX,PY);
% ugyanitt a szintvonalak
contour(X, Y, Z);
axis([min(min(X)) max(max(X))
min(min(Y)) max(max(Y))]);
view([15 15]);
```

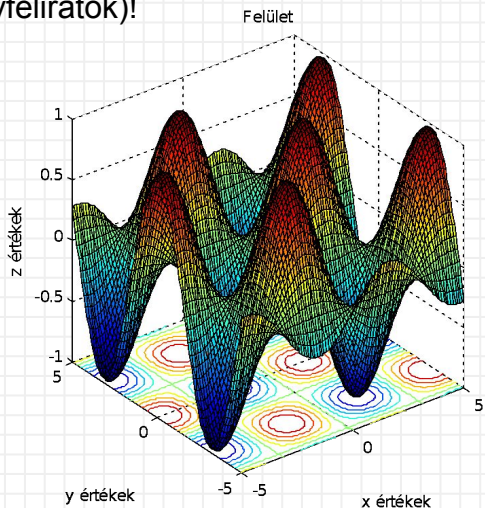
```
subplot(122);
% felület + szintvonalak
surf(X, Y, Z);
axis([min(min(X)) max(max(X))
min(min(Y)) max(max(Y))]);
view([15 15]);
```



## Gyakorló Példa 3.

Ábrázold a  $z = \sin(x)\cos(y)$  képlettel megadott felületet a szintvonalalaival együtt a  $[-5; 5] \times [-5; 5]$  tartományon, 0.1-es lépésközzel!

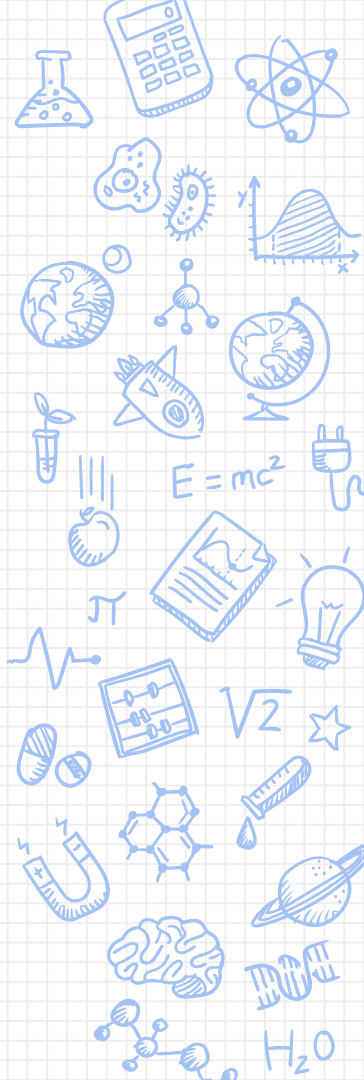
Az ábrát feliratozd megfelelően (cím, tengelyfeliratok)!



```
[X,Y] = meshgrid(-5:0.1:5);
Z = sin(X).*cos(Y);
```

```
figure;
clf;
surf(X,Y,Z);
```

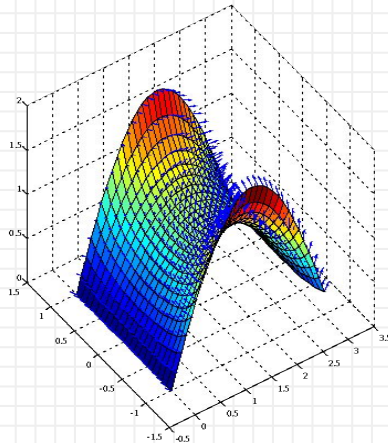
```
axis square
title('Felület');
xlabel('x értékek');
ylabel('y értékek');
zlabel('z értékek');
```



## Gyakorló Példa 4.

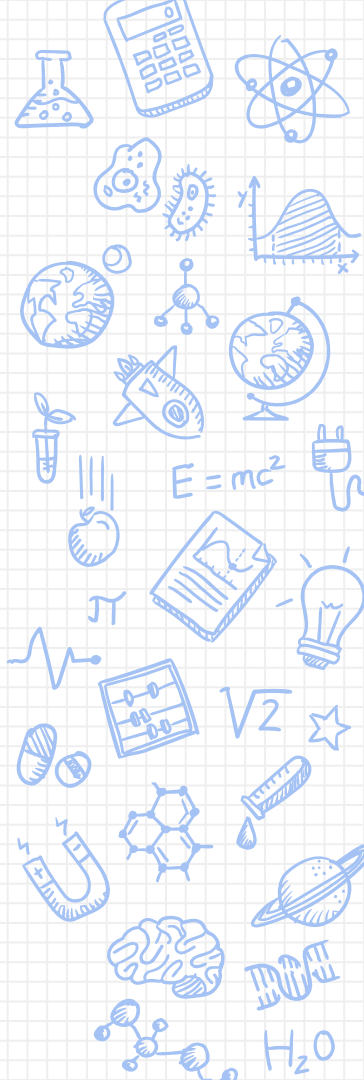
A `quiver3` és a `surfnorm` parancsok Help bejegyzései alapján ábrázold az alábbi felületet és az egyes rácspontokra eső normálvektorokat a  $[0;3] \times [-1;1]$  tartományon, 0.1-es lépésközzel!

$$z = \frac{\sin(x)}{\cos(y)}$$



```
% a térháló pontjainak létrehozása
[X,Y] = meshgrid(0:0.1:3,-1:0.1:1);
% a felület pontjainak kiszámítása
Z = sin(X)./cos(Y);
% normálvektorok kiszámítása
[U,V,W] =surfnorm(X,Y,Z);
```

```
% ábrázolás
figure;
% normálvektorok térben
quiver3(X,Y,Z,U,V,W);
hold on
% felület
surf(X,Y,Z);
% nézet
view(-35,45)
axis square
hold off
```



## Feladatok

---

- ✗ a feladatgyűjtemény **7.1**, **7.2** feladatai, melyeknek neve ez legyen, rendre:  
gyak7\_f71\_[NEPTUN].m  
gyak7\_f72\_[NEPTUN].m  
(természetesen szögletes zárójelek nélkül).
- ✗ a diasorban ismertetett parancsok kikeresése és tanulmányozása a Help-ben

Amivel nem végzel / nem végzünk, azt otthon kell befejezni, ez a házi feladat is egyben. A határidő vasárnap (április 9.) éjfél.

Feltöltés: [users.itk.ppke.hu/~zseta/matlab2017/HF07](https://users.itk.ppke.hu/~zseta/matlab2017/HF07)



## 8. gyakorlat

# Cellatömbök, struktúrák, fájlműveletek



## Cellatömbök

- ✗ Olyan **adattípus**, melyet **különböző típusú és/vagy méretű** változók **tárolására** használhatunk.
- ✗ Üres cellatömb létrehozása: `C = cell(2,2);`
- ✗ Cellatömb elemeinek indexelése a `{ }` zárójelekkel, rész-cellatömb indexelése a `( )` zárójelekkel történik.
- ✗ Cellatömb létrehozása ismert elemekkel:

`C = {1, 2; 3, 4}`

- ✗ Fontos, hogy **az egyes cellákban teljesen eltérő elemeket tárolhatunk**, akár további cellatömböket is.
- ✗ Cellatömbök megjelenítése: `cellplot`



## Cellatömb létrehozás, és ábrázolás

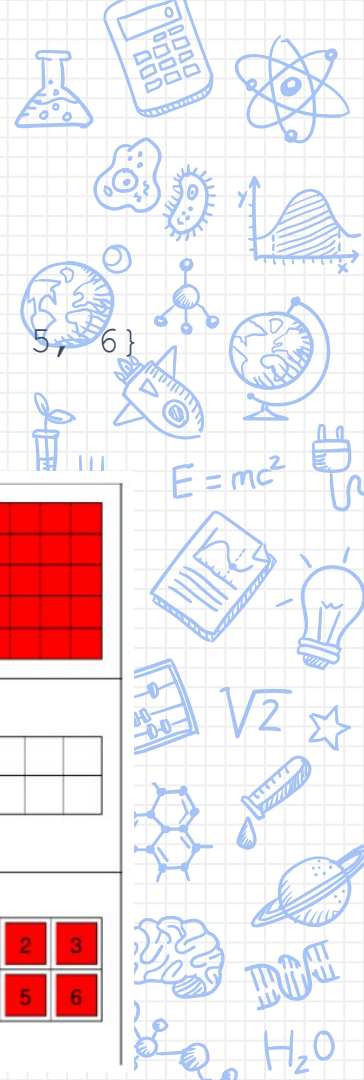
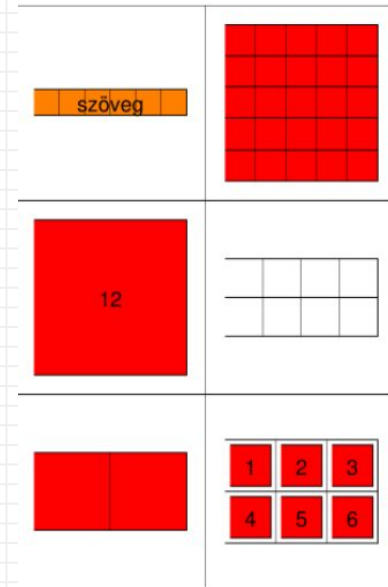
- ✗ Hozzunk létre egy 3×2 méretű üres cellatömböt!
- ✗ Az egyes cellákat oszlopfolytonosan töltjük fel az alábbi elemekkel:
- ✗ 'szöveg' , 12, [8.13, 14.86], magic(5), cell(2,4), {1, 2, 3; 4, 5, 6}
- ✗ Jelenítsük meg a cellatömb tartalmát a cellplot parancs használatával!
- ✗ Próbálgassuk az egyes elemek indexelését!

```
% cellatömb létrehozása
C1 = cell(3,2);

% feltöltés oszlopfolytonosan
C1{1,1} = 'szöveg';
C1{2,1} = 12;
C1{3,1} = [8.13, 14.86];
C1{1,2} = magic(5);
C1{2,2} = cell(2,4);
C1{3,2} = {1, 2, 3; 4, 5, 6};

% másik lehetőség
C2 = {'szöveg', magic(5); 12, cell(2,4); [8.13 14.86], {1, 2, 3; 4, 5, 6}};

% megjelenítés
cellplot(C1);
title('C1 tartalma');
figure;
cellplot(C2);
title('C2 tartalma');
```



## Struktúrák

- ✗ A cellatömbökhöz hasonlóan **különböző típusú** és/vagy **méretű** változók tárolására használható **adattípus** azzal a különbséggel, hogy a tárolt adatok **névvel ellátott mezőkbe** kerülnek (hasonlóan pl a C nyelv struct típusához).
- ✗ Az adatok címezése a `változónév.mezőnév` minta alapján történik

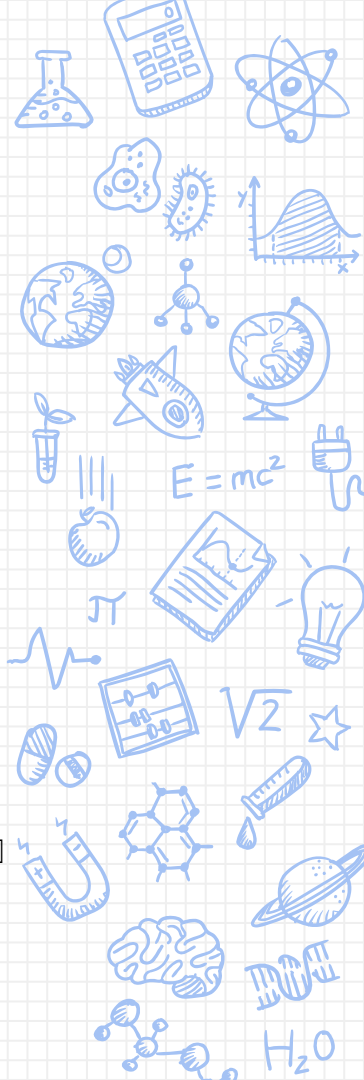
### Struktúra létrehozása

```
hallgato.nev = 'Kis Pista';
hallgato.szul_datum = '1992.03.24.';
hallgato.evfolyam = 3;
hallgato.osztalyzatok = [2 4 3 5 3 4 4 3];
```

### Struktúra/mezők lekérdezése

```
>> hallgato
hallgato =
 nev: 'Kis Pista'
 szul_datum: '1992.03.24.'
 evfolyam: 3
 osztalyzatok: [2 4 3 5 3 4 4 3]
```

```
>> hallgato.nev
ans =
Kis Pista
```





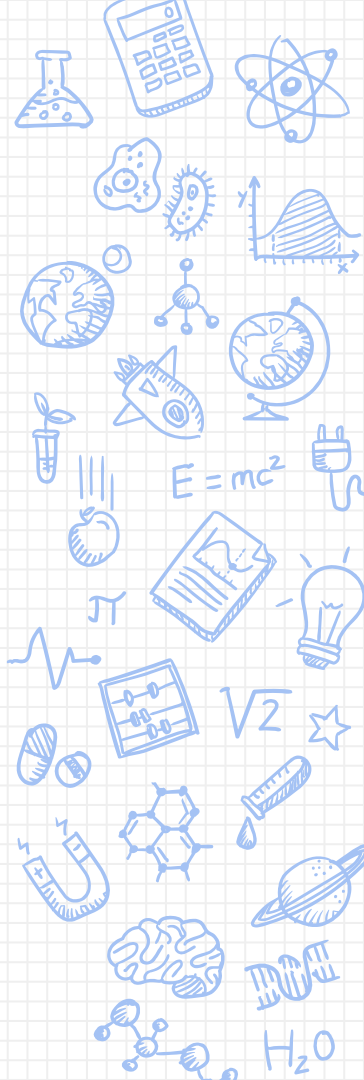
## Fájlkezelés

- ✗ Többféle módon lehet MATLAB-ban fájlokat írni/olvasni, melyek közül az alábbi három módszerről lesz szó:
- ✗ fájl írás/olvasás **formázott szövegként**: `fprintf`, `fscanf`, `textscan`
  - ✗ Használható logok készítésére, vagy pedig egyszerű adatok olvasható tárolására, látszik a formátum
- ✗ fájl írás/olvasás **binárisan**: `fwrite`, `fread`, `save`, `load`
  - ✗ Adatok kompakt tárolása, csak programmal olvasható, a formátum nem feltétlenül derül ki a fájlból
  - ✗ A MATLAB beépített tárolási módja is bináris fájlokat generál/olvas be (.mat)
- ✗ fájl írás/olvasás **táblázatként**: `writetable`, `readtable` <- jövő órán
  - ✗ Táblázatok hatékony kezeléséhez, nagyobb mennyiségű, rendezett adathoz praktikus
  - ✗ Többféle formátumot támogat: .txt, .dat, .csv, .xls, .xlsx, .ods



## Elérési útvonalak

- ✗ A kiíráshoz kell lenni fog a cél fájl elérési útvonala stringként megadva:
- ✗ Az egyes platformokon más a fájl elválasztó karakter ('\' vagy '/'), ami csapatmunka esetén problémás lehet. Erre nyújt megoldást a **filesep** parancs, amely megadja az aktuális platformon használt elválasztó karaktert.
- ✗ Windows alatt pl.: `DIR_PATH = 'D:\myfolder\file_op\'`;
- ✗ Linux alatt pl.: `DIR_PATH = '/home/myfolder/file_op/'`;
- ✗ Aktuális könyvtár lekérdezése: `pwd`
- ✗ Fájlok listázása: `dir`
  - ✗ Szűrők pl.: `matFilesInfo = dir([pwd filesep '*.mat'])`;
- ✗ Fájl nyitás és zárás: `fopen`, `fclose`



# Fájlba írás formázott szöveggént

## X `fprintf(fileID, formatString, data)`

```
% kiirando jel generalasa
% mintaveteli frekvencia
Fs=1000; % Hz
% idovektor
t=0:1/Fs:10; % s
% a szinusz frekvenciaja
f=5; % Hz
% szinusz jelalak
s=sin(2*pi*f*t);

% az adott helyhez kepest vett relativ cimzes
% (lehet abszolut is, teljes utvonallal)
fs = filesep;
DIR_PATH=['.' fs 'files' fs];

% a kiirando fajl neve
filename='szinusz_fprintf.txt';

% a fajl eleresi utvonala (ha csak a
fajlnevet
% adom meg, az aktualis könyvtarba menti)
file_path=[DIR_PATH filename];
```

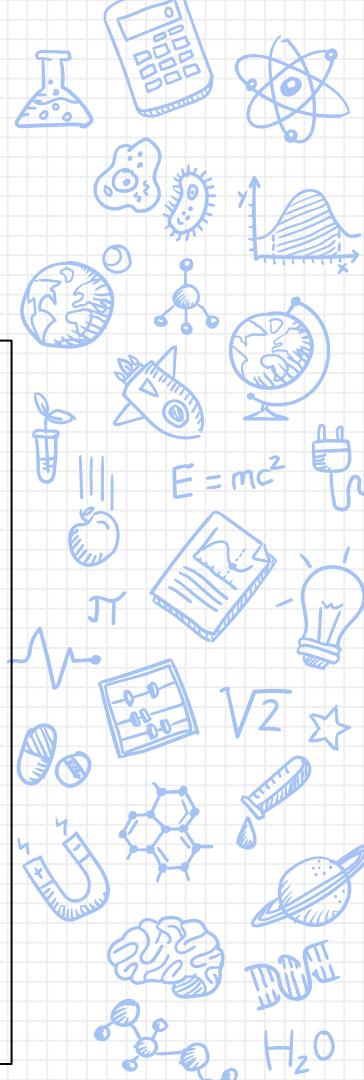
```
% fajl megnyitasa irasra, ez utan az fid-val
% hivatkozok erre a fajlra
fid=fopen(file_path,'w');

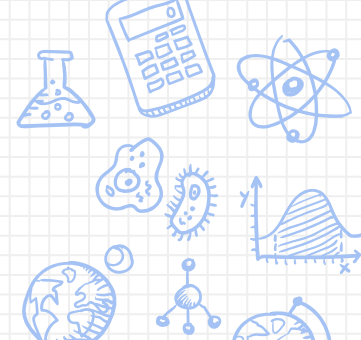
% header irasa, hogy tudjuk, mi van a fajlban
% pl.: Szinusz (Fs = 1000)
fprintf(fid,['Szinusz (Fs = ' num2str(Fs) ')
\n']);
fprintf(fid,'t sin \n');

% az iras utan nyitva marad a fajl
% ezert irhatok meg bele formazott adatot

% irjuk bele a kiszamolt szinusz fuggvenyt
fprintf(fid,'%5.2f %4.4f\n',[t;s]);

% fajl bezarasa - FONTOS, ne felejtsek el!
fclose(fid);
```





## Fájlból olvasás formázott szöveggént (fscanf)

**X** `fscanf(fileID, formatString, outputSize)`

```
% a beolvasando fajl neve (amit az elobb
kiirtunk)
filename='szinusz_fprintf.txt';

% a fajl eleresi utvonala
file_path=[DIR_PATH filename];

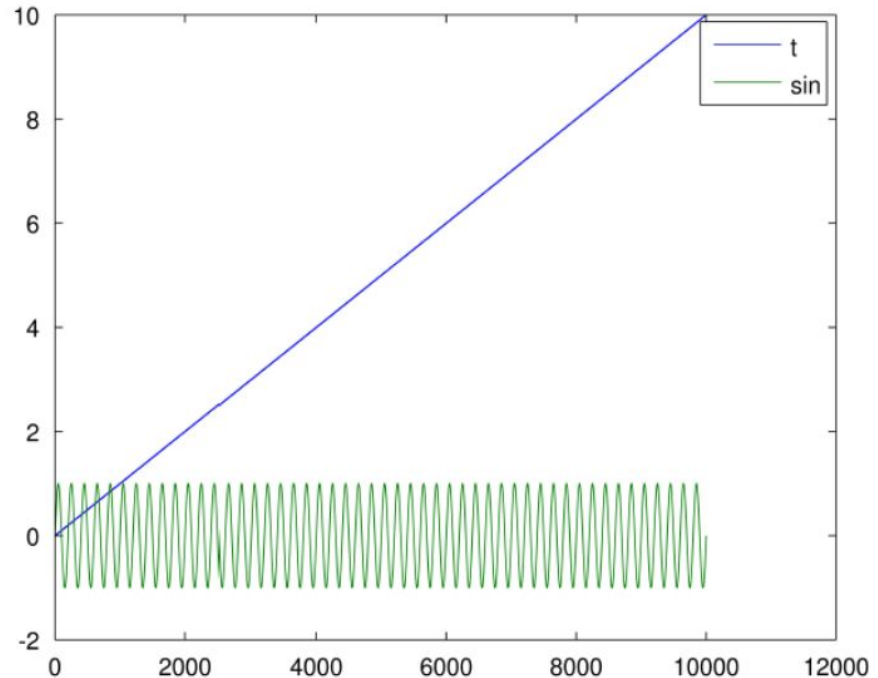
% fajl megnyitasa olvasasra
fid=fopen(file_path);

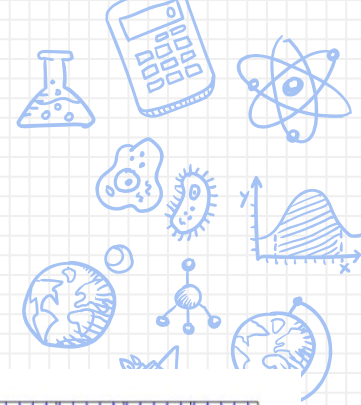
% fajl beolvasasa formazott szoveggent
% eloszor a header beolvasasa stringkent
Sheader=fscanf(fid,'%s',6);

% majd az adatok beolvasasa lebegopontos szamkent
Sdata=fscanf(fid,'%f %f',[2 inf]);

% fajl bezarasa
fclose(fid);

% beolvasott adatok kirajzolasa
figure;
plot(Sdata');
legend('t','sin');
```





## Fájlból olvasás formázott szöveggént (textscan)

### X `textscan(fileID, formatSpec, nTimes)`

```
% a textscan-nek lehet regularis kifejezest is adni (lásd
HELP)
% a beolvasando fajl neve (meg mindig ugyanaz)
filename='szinusz_fprintf.txt';

% a fajl eleresi utvonala
file_path=[DIR_PATH filename];

% fajl megnyitasa olvasasra
fid=fopen(file_path);

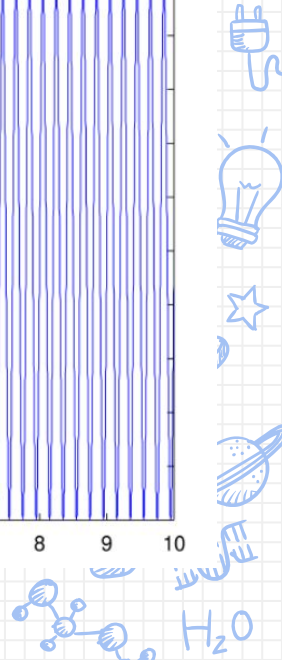
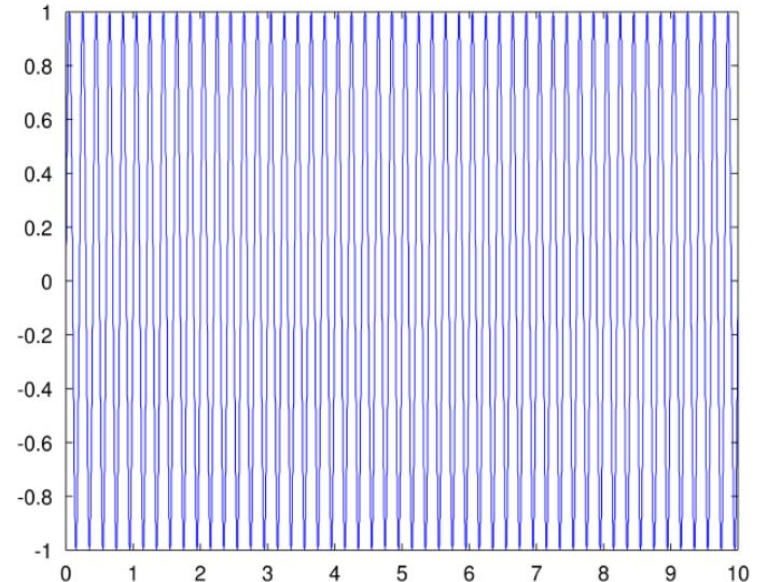
% fajl beolvasasa cellatombbe header beolvasasa
Sheader_ts=textscan(fid,'%s',6);

% adatok beolvasasa
Sdata_ts=textscan(fid,'%f %f');

% fajl bezarasa
fclose(fid);

% beolvasott adatok kirajzolasa
tt = Sdata_ts{1};
data = Sdata_ts{2};

figure;
plot(tt,data);
```



```
X fwrite(fileID, data, precision)
```

```
% a kiirando fajl neve
filename='szinusz_fwrite.bin'; % nem kotelezo a
bin
```

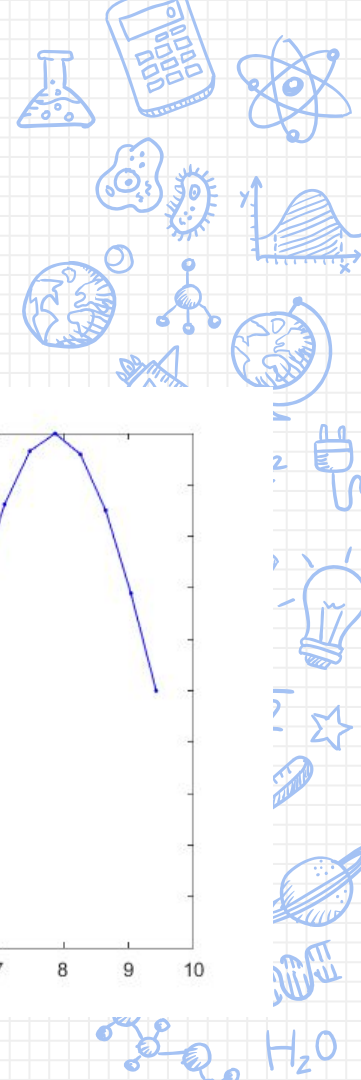
```
fs = filesep;
dir_path = ['.', fs];
file_path=[dir_path, filename];
```

```
% fajl megnyitása irasra, ez utan az
% fid-val hivatkozok erre a fajlra
fid=fopen(file_path,'w');
```

```
t = linspace(0.1, 3*pi, 25);
y = sin(t);
```

```
% adatok kiirasa binarisan, double-kent
fwrite(fid,[t' y'],'double');
% fajl bezarasa, valtozok torlese
fclose(fid);
```

|         |      |      |      |      |      |      |      |      |
|---------|------|------|------|------|------|------|------|------|
| 0000000 | 999a | 9999 | 9999 | 3fb9 | 4f3a | 7666 | 441d | 3fdf |
| 0000010 | 1c07 | 4333 | 10ea | 3fec | 8839 | e599 | 3fe2 | 3ff4 |
| 0000020 | 826e | a999 | 7750 | 3ffa | be51 | 36cc | 575f | 4000 |
| 0000030 | bb6c | 18cc | 7316 | 4003 | b886 | facc | 8ecc | 4006 |
| 0000040 | b5a1 | dccc | aa83 | 4009 | b2bc | becc | c63a | 400c |
| 0000050 | afd6 | a0cc | e1f1 | 400f | 5678 | 4166 | 7ed4 | 4011 |
| 0000060 | 5505 | b266 | 0caf | 4013 | 5392 | 2366 | 9a8b | 4014 |
| 0000070 | 521f | 9466 | 2866 | 4016 | 50ad | 0566 | b642 | 4017 |
| 0000080 | 4f3a | 7666 | 441d | 4019 | 4dc7 | e766 | d1f8 | 401a |
| 0000090 | 4c55 | 5866 | 5fd4 | 401c | 4ae2 | c966 | edaf | 401d |
| 00000a0 | 496f | 3a66 | 7b8b | 401f | 23fe | 55b3 | 84b3 | 4020 |
| 00000b0 | 2345 | 0e33 | 4ba1 | 4021 | 228c | c6b3 | 128e | 4022 |
| 00000c0 | 21d2 | 7f33 | d97c | 4022 | cb2c | cb8b | 8eae | 3fb9 |
| 00000d0 | e198 | b947 | 0982 | 3fde | b2a5 | a45f | 9a8b | 3fe8 |
| 00000e0 | b0e0 | a17f | 856c | 3fee | ccd4 | 6f6b | e392 | 3fef |
| 00000f0 | 312e | ee58 | 80ca | 3fec | 42f2 | a097 | de4b | 3fe4 |
| 0000100 | 3fc3 | bd52 | 3edd | 3fd4 | 40f8 | bf0b | odd4 | bfb1 |
| 0000110 | 1dfd | 8f70 | 2314 | bfdc | 73cb | 2a3e | e87a | bfe7 |
| 0000120 | 5fd7 | 17a0 | 2f0c | bfee | 9258 | 1b28 | f5c3 | bfef |
| 0000130 | fdb9 | a6a0 | f8d6 | bfec | 2bd3 | 4813 | aa4d | bfe5 |
| 0000140 | a12e | 92e3 | 41fa | bfd6 | 8ad1 | f3b9 | 1041 | 3fa1 |
| 0000150 | 9144 | b5db | 34a5 | 3fda | 7c8f | e9d4 | 2f9b | 3fe7 |
| 0000160 | 45bc | d771 | d015 | 3fed | 53c3 | bc52 | fedc | 3fef |
| 0000170 | eaaf | e338 | 68a4 | 3fed | 319b | 7793 | 7025 | 3fe6 |
| 0000180 | 243f | ccb8 | 3ec2 | 3fd8 | 8a0a | 4c9e | 7939 | 3cba |
| 0000190 |      |      |      |      |      |      |      |      |



## Fájlból olvasás bináris formában

**X** `fread(fileID, size, precision)`

```
% a beolvasando fajl neve
filename='szinusz_fwrite.bin';

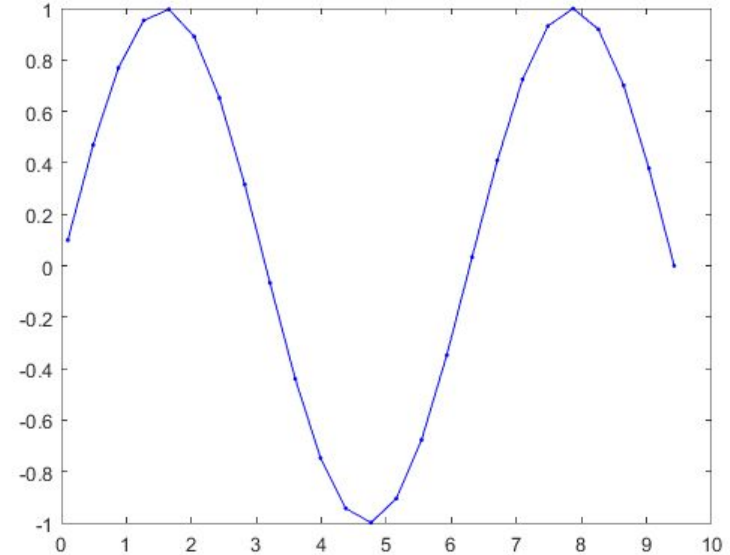
fs = filesep;
dir_path = ['. ', fs];
file_path=[dir_path, filename];

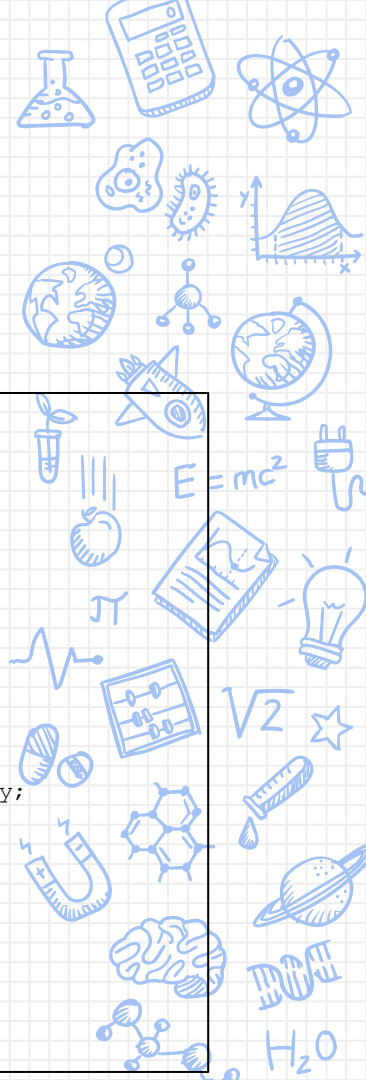
% nyitas, olvasas
fid=fopen(file_path, 'r'); % vagy siman
fid=fopen(file_path);

data = fread(fid, inf, 'double');
data = reshape(data, length(data)/2, 2);

fclose(fid);

figure;
plot(data(:, 1), data(:, 2), 'b.-');
```





## Mentés/betöltés MATLAB formátum esetén

---

**X** `save(filePath, variableNames, ...)`  
**X** `load`

```
% csak hogy tiszta legyen a munkater
clear;
% a kiirando fajl neve
filename='sajat_matlab_archivum.mat';
fs = filesep;
dir_path = ['. ', fs];
file_with_path=[dir_path, filename];
t = linspace(0.1, 3*pi, 25);
y = sin(t);
dummy1 = rand(20, 2);
% most mi van a munkaterben?
% a valtozok kimentese: save utasitassal
save(file_with_path, 't', 'y')
% valtozok torlese a munkaterbol clear('t', 'y', 'dummy1'); % lehetne meg a formaja: clear t, y;
% most mi van a munkaterben?
% valtozok szelektiv betoltese -- ha tudod elore mi van benne...
load(file_with_path, 't');
% most mi van a munkaterben?
load(file_with_path, 'y');
% most mi van a munkaterben? - lehetett volna egyszerre mindent: load(file_with_path);

figure;
plot(t, y, 'b.-');
```





## Feladatok

---

- ✗ a feladatgyűjtemény **8.1 - 8.4** feladatai, melyeknek neve ez legyen, rendre:  
gyak8\_f81\_[NEPTUN].m  
...  
gyak8\_f84\_[NEPTUN].m  
(természetesen szögletes zárójelek nélkül).
- ✗ az ismertetett parancsok kikeresése és tanulmányozása a Help-ben

Amivel nem végzel / nem végzünk, azt otthon kell befejezni, ez a házi feladat is egyben. A határidő vasárnap (április 30.) éjfél.

Feltöltés: [users.itk.ppke.hu/~zseta/matlab2017/HF08](https://users.itk.ppke.hu/~zseta/matlab2017/HF08)

# MATLAB 2017

## 9. gyakorlat

Táblázatok, fájlműveletek, képmentés



## Táblázatok

- ✗ Olyan **adattípus**, melyet **különböző típusú** és/vagy **méretű** változók, valamint meta adatok rendszerezett **tárolására** használhatunk.
- ✗ Az oszlopokba sorolt adatok tárolására a legmegfelelőbb, amire példa a vesszővel tagolt fájlok (csv), valamint a táblák (xls,xlsx...)
- ✗ Megadható oszlop-, sornév, leírás, oszlop leírás, valamint mértékegység az oszlopokhoz.
- ✗ Üres táblázat: `t = table;`
- ✗ Oszlopnevekkel:

```
t=table(v1,v2,v3,'VariableNames',{'o1','o2','o3'})
```

```
t=table(1,1,1,'VariableNames',{'o1','o2','o3'})
```

t =

| o1 | o2 | o3 |
|----|----|----|
| 1  | 1  | 1  |

- X Adatok indexelése táblázatban:
  - X `()` - táblázatot ad vissza
  - X `{}` - homogén adatok esetén tömböt ad vissza
  - X oszlopnévvel - vektort ad vissza
- X Táblázat létrehozása tömbökből - fontos az adatok elrendezése
  - X `array2table`, `cell2table`, `struct2table`
- X Tömbök létrehozása táblázatból - fontos az adatok fajtája
  - X `table2array`, `table2cell`, `table2struct`
- X Táblázatok összefűzése, műveletei
  - X Táblázat tulajdonságai: `height`, `width`, `istable`
  - X Adatokon végzett műveletek: `summary`, `ismember`, `sortrows`, `ismissing`, `varfun`, `rowfun`, `standardizeMissing`, `unique`
  - X Táblázatok műveletei: `intersect`, `union`, `join`, `setdiff`, `setxor`, `innerjoin`, `outerjoin`



## Fájl írás/olvasás táblázatként

- ✗ Különböző fájl típusok: .txt, .csv, .dat, .xls, .xlsx, .ods, ...
- ✗ `t=readtable(fileName);`
- ✗ Megadható még:
  - ✗ `FileType` - fájl típusa: text - tagolt szöveg, vagy spreadsheet - táblázat
  - ✗ `Delimiter` - elválasztó karakter
  - ✗ `Format` - formázó string, ami alapján megy a beolvasás
  - ✗ `Range` - beolvasást megadó téglalap alakú cellatartomány
  - ✗ `ReadVariableNames` - első oszlop használata címsornak
- ✗ Földrajzi adatokat tartalmazó táblázat betöltése, amiben oszlop nevek is megtalálhatók:

```
ft=readtable('foldrajzi.xls','ReadVariableNames',true)
;
summary(ft)
```



## Fájl írás/olvasás táblázatként

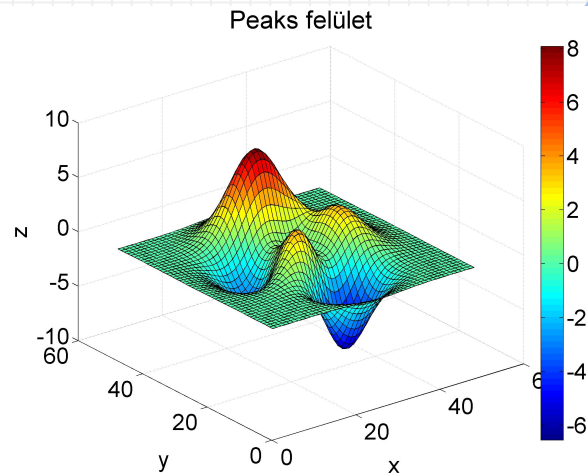
---

- ✗ Különböző fájl típusok: .txt, .csv, .dat, .xls, .xlsx, .ods, ...
- ✗ `writetable(t, fileName);`
- ✗ Megadható még:
  - ✗ `FileType` - fájl típusa: text - tagolt szöveg, vagy spreadsheet - táblázat
  - ✗ `Delimiter` - elválasztó karakter
  - ✗ `Sheet` - lap megadása szövegesen, vagy az indexével
  - ✗ `Range` - kiírást megadó téglalap alakú cellatartomány
  - ✗ `WriteVariableNames` - első oszlop használata címsornak
- ✗ Földrajzi adatok táblázatos kiírása, oszlop nevekkal:  
`writetable(ft, 'foldrajzi.xlsx', 'WriteVariableNames', true)`
- ✗ Földrajzi adatok kiírása, csv-be oszlop nevekkal, pontosvesszővel tagoltan:  
`writetable(ft, 'foldrajzi.csv', 'WriteVariableNames', true, 'FileType', 'text', 'Delimiter', ';')`

## Ábrák mentése

- ✗ A MATLAB sokféle formátumot támogat - céltól függően más-más formátum a jó
- ✗ Fontos a megfelelő méretű szöveg, és a megfelelő vastagságú vonalak kiválasztása!
- ✗ Kép mentése figure ablakból: `print`
  - ✗ LaTeX - `print`(figure\_handler, '-depsc2', '-r300', picture\_name)
  - ✗ Word - `print`(figure\_handler, '-dpng', '-r300', picture\_name)
  - ✗ Felületek -  
`print`(figure\_handler, format, '-zbuffer', '-r300', picture\_name)
  - ✗ Figure handler - figure\_handler = **figure**;
  - ✗ Aktuális tengelyek - axes\_handler = **gca**;

```
f1=figure;
surf(peaks)
title('Peaks felület','FontSize',22)
xlabel('x','FontSize',20)
ylabel('y','FontSize',20)
zlabel('z','FontSize',20)
colorbar
set(gca,'FontSize',20)
print(f1, '-dpng', '-zbuffer', '-r300', 'surf1.png')
```





## Feladatok

---

- ✗ a feladatgyűjtemény **9.1 - 9.4** feladatai, melyeknek neve ez legyen, rendre:

gyak9\_f91\_[NEPTUN].m

...

gyak9\_f94\_[NEPTUN].m

(természetesen szögletes zárójelek nélkül).

- ✗ az ismertetett parancsok kikeresése és tanulmányozása a Help-ben

Amivel nem végzel / nem végzünk, azt otthon kell befejezni, ez a házi feladat is egyben. A határidő vasárnap (május 7.) éjfél.

Feltöltés: [users.itk.ppke.hu/~zseta/matlab2017/HF09](https://users.itk.ppke.hu/~zseta/matlab2017/HF09)