

# MATLAB 2024

## 3. téma

Vezérlőszervezetek, formázott kiíratás,  
polinomok, 2D ábrázolás



## Vezérlőszervezetek - elágazás

---

### IF

```
if feltetel  
    parancsok  
end
```

### ELSE

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

### ELSEIF

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

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

## Vezérlőszervezetek - switch

---

### SWITCH

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

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

## Vezérlőszervezetek - ciklusok

### FOR

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

- X ismert számú iteráció elvégzésére
- X nincs zárójelezés, a blokk végét az `end` jelzi
- X a **ciklusváltozót sorvektorként** (vagy más sor elrendezésű változóként) definiáljuk, a ciklus törzsében értéke az aktuális elemnek megfelelő érték
- X a ciklusváltozó tetszőleges sorvektor lehet (pl. `[1 3 7 5 6]`)

### WHILE

```
while feltétel  
    parancsok  
end
```

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

## Mátrixban keresés C++-hoz képest

---

**C++:**

```
vector<vector<int>> matrix;  
// *** mátrix feltöltése ***  
  
// Minimumkeresés:  
int matmin = matrix[0][0];  
for(size_t i = 0; i < matrix_s; i++)  
    for(size_t j = 0; j < matrix[i].size(); j++)  
        if (matrix[i][j] < matmin)  
            matmin = matrix[i][j];  
  
cout << endl << matmin;
```

**Matlab:**

```
m = randn(10);  
min(m,[],'all');
```

## Formázott kiíratás

---

**X** `disp` - szöveg vagy változótartalom egyszerű kiíratása

```
disp(ones(2)./[2 3; 4 5])  
% → 0.5000    0.3333  
%    0.2500    0.2000
```

**X** `fprintf` - szöveg kiíratása formázott-beágyazott mezőkkel (vagy fájlba, vagy a konzolra); különösen fontos: **formatSpec** rész a help-ben

**X** `sprintf` - mint `fprintf`, csak string-változóba “nyomtat”

**X** `rats` - racionális számmal közelít, bizonyos/adott tolerancia mellett

```
disp(rats(ones(2)./[2 3; 4 5]))  
% → 1/2    1/3  
%    1/4    1/5
```

## Formázott kiíratás

X 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  
e - normál alak

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

## Formázott kiíratás

---

### X `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
```

### X `num2str` - szám → string konverzió

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

### X `mat2str` - mátrix → string konverzió

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



## Polinomok

---

- ✗ sok függvény és valós folyamat leírható magasabb rendű polinomokkal
- ✗ MATLAB-ban a polinomokat az együtthatóvektorokkal reprezentáljuk:

$$\text{✗ } P_1(x) = x^2 + 2x + 3 \quad \rightarrow \text{P1} = [1 \ 2 \ 3];$$

$$\text{✗ } P_2(x) = 10x^3 + 4x^2 + 5x - 7 \quad \rightarrow \text{P2} = [10 \ 4 \ 5 \ -7];$$

$$\text{✗ } P_3(x) = 3x^4 + 5x^2 - 12 \quad \rightarrow \text{P3} = [3 \ 0 \ 5 \ 0 \ -12];$$

P1, P2, P3 itt egy egyszerű sorvektor!

## Polinomok

---

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

**X** gyökök kiszámítása:

```
roots (P) ;
```

**X** kiértékelés adott pontban:

```
y0 = polyval (P, x0) ;
```

**X** kiértékelés sok pontban:

```
x = -1:0.01:1;
```

```
y = polyval (P, x) ;
```

**X** polinom létrehozása a gyökeiből:

```
r = [-1 1] ;
```

```
P = poly (r) ;
```

**X** polinom illesztése adatsorra:

```
P = polyfit (x_t, y_t, fokszam) ;
```

## 2D ábrázolás

---

### Az ábrák hierarchiája

#### X Ábra -- figure

- X Lényegében a fölnyíló ablak, amin megjelennek a grafikonok
- X új ábra létrehozása -- **figure**

#### X Tengelyek -- axes

- X egy ábrán több alábrát (tengelypárok) is megjeleníthetünk (de egyet mindenképp)
- X tengelypárokat explicit módon szabályos rácsos felosztással a **subplot** függvénnyel adhatunk hozzá az ábrához
- X implicit módon grafikon rajzolásakor jöhet létre tengelypár

#### X Grafikon -- line

- X grafikont a megjelenítéssel együtt ad hozzá
- X az aktív alábrához adja hozzá a grafikont
- X ha nem adtunk hozzá mi magunk alábrát, akkor ez hozzáad egyet

## 2D ábrázolás

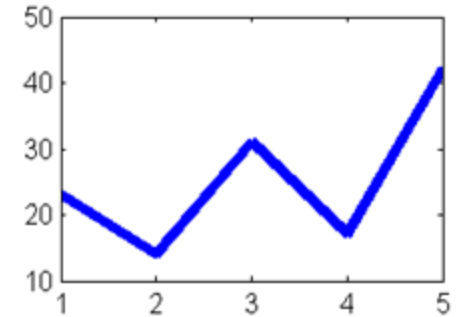
```
alma = [23, 14, 31, 17, 42];  
plot(alma);
```

X figure - új ábra létrehozása

X plot(x), plot(x, y) - adatsor kirajzolása

X egyargumentumú esetben: a vektor értékei a vektorindexek függvényében

X kétargumentumú esetben:  $x \rightarrow$  helyek,  $y \rightarrow$  értékek

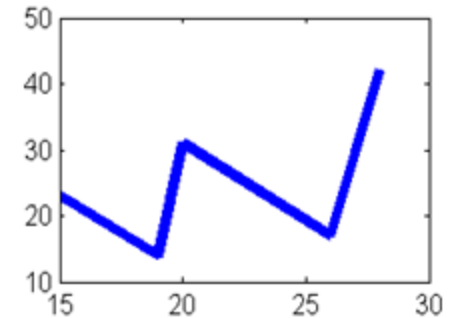


```
korte = [15, 19, 20, 26, 28];  
plot(korte, alma);
```

X stem(x), stem(x, y) - pálcikadiagram, paraméterezés hasonlóan

X stairs - lépcsődiagram

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



## 2D ábrázolás

Az ábrák tulajdonságai változókbán tárolhatók

**X** `f = figure;`

**X** `get(f);` - az ábra tulajdonságok lekérése

**X** `pl1 = plot(rand(1, 6))` - a grafikon tulajdonságai

**X** `a = get(f, 'CurrentAxes')` - a tengely tulajdonságai

**X** `get(a, 'Children')` - a grafikon tulajdonságai a tengely „gyereke”

```
Alphamap: [1×64 double]
BeingDeleted: off
BusyAction: 'queue'
ButtonDownFcn: ''
Children: [0×0 GraphicsPlaceholder]
Clipping: on
CloseRequestFcn: 'closereq'
Color: [1×3 double]
Colormap: [256×3 double]
ContextMenu: [0×0 GraphicsPlaceholder]
CreateFcn: ''
CurrentAxes: [0×0 GraphicsPlaceholder]
CurrentCharacter: ''
CurrentObject: [0×0 GraphicsPlaceholder]
CurrentPoint: [0 0]
DeleteFcn: ''
DockControls: on
```

`a =`

**Axes** with properties:

```
XLim: [1 6]
YLim: [2.0000e-01 1]
XScale: 'linear'
YScale: 'linear'
GridLineStyle: '-'
Position: [1.3000e-01 1.1000e-01 7.7500e-01 8.1500e-01]
Units: 'normalized'
```

`pl1 =`  
**Line** with properties:

```
Color: [0 4.4700e-01 7.4100e-01]
LineStyle: '-'
LineWidth: 5.0000e-01
Marker: 'none'
MarkerSize: 6
MarkerFaceColor: 'none'
XData: [1 2 3 4 5 6]
YData: [5.9889e-01 9.0106e-01 9.3938e-01 2.2118e-01 4.8267e-01 3.7601e-01]
ZData: [1×0 double]
```

## 2D ábrázolás

X subplot - az ábránkat alábrákra oszthatjuk szabályos rácsos felosztással

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

1	2	3
4	5	6



sorok száma

oszlopok száma

index (sorfolytonosan)

X rajzolás több cellába:

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



X grid on/off - a grafikonhoz hátsó hálószerkeztés kérhető

X legend - adatsorok felirata

X get(a, 'Type') - a subplot axes típusú

## 2D ábrázolás

---

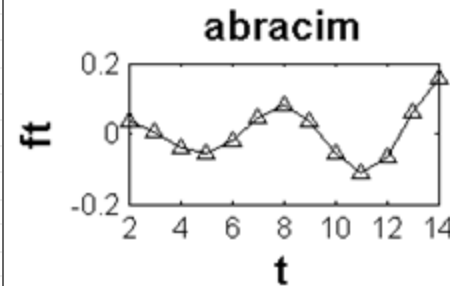
- X **linespec**: nem konkrét parancs, hanem a vonalak részletes beállításának gyűjteménye, kifejezetten érdemes a Help-ben megnézni
- X **xlim, ylim, box** - tengelyhatárok
- X **text, xlabel, ylabel, title** - csak úgy szöveg, x- és y-tengely felirat, ábracím
  - X paraméterezés pl. `'FontSize', 14,`  
`'FontWeight', 'bold'`
- X `xline(3.14); yline(120);` - vízszintes és függőleges vonal adott pozícióban

## 2D ábrázolás

Hozzunk létre egy alábrát az alábbiak szerint:

- X az ábra bal alsó negyedébe kerüljön,
- X a  $[2, 14]$  közötti értelmezési tartományon
- X 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

---

X [P-ITMAT-0014] Bevezetés a Matlab programozásba

X <https://moodle.ppke.hu/course/view.php?id=1514>

X 03 Feladatok → 2D ábrázolás

→ Szenzorstatisztika

→ Polinom