



Matlab 5. előadás

Grafika, függvényábrázolás

Dr. Szörényi Miklós,
Dr. Kallós Gábor

2017–2018





Tartalom

- Motivációs példák
- Grafika – áttekintés
- Plot parancs, függvényábrázolás
 - Plot beállítások
 - Animáció
- Subplot és fplot
- Egy- és többváltozós függvények ábrázolásának (további) lehetőségei
 - Szimbolikus megadású függvények, ezplot
 - Többváltozós függvények, 3D grafika
- Függvényvizsgálat
 - Zérushelyek, metszéspontok
 - Lokális maximum- és minimumhelyek
 - További lehetőségek



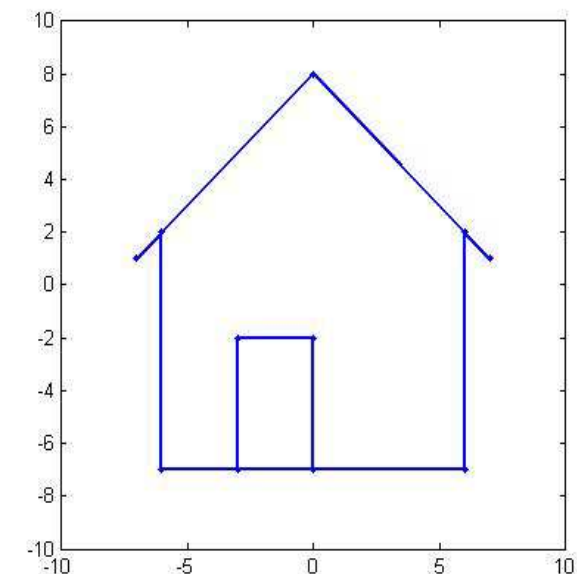
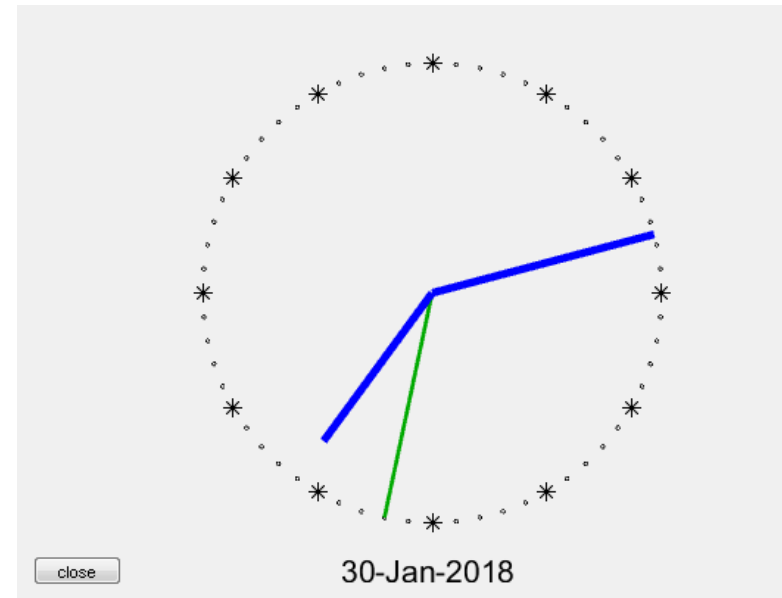


Példák

- clockex: analóg óra
 - Grafikus objektumokat használ
 - Készítő: Cleve Moler
 - Részletek: clockex.m fájl
- Házikó transzformációi
 - Síkbeli pontsorozatot összekötünk
 - „Csinosítva”: dot2dot.m fájl
 - Az ábra még hangolható
 - Transzformációk mátrixszal
 - Lásd köv. slide

Bemutató

```
>> X = [-6 -6 -7 0 7 6 6 -3 -3 0 0;  
... -7 2 1 8 1 2 -7 -7 -2 -2 -7]  
% a házikó pontjai, ellenőrizzük!  
>> plot(X(1,:),X(2,:))  
% pontok kirajzolása  
>> X(:,end+1) = X(:,1);  
% korrekció (összekötés)
```



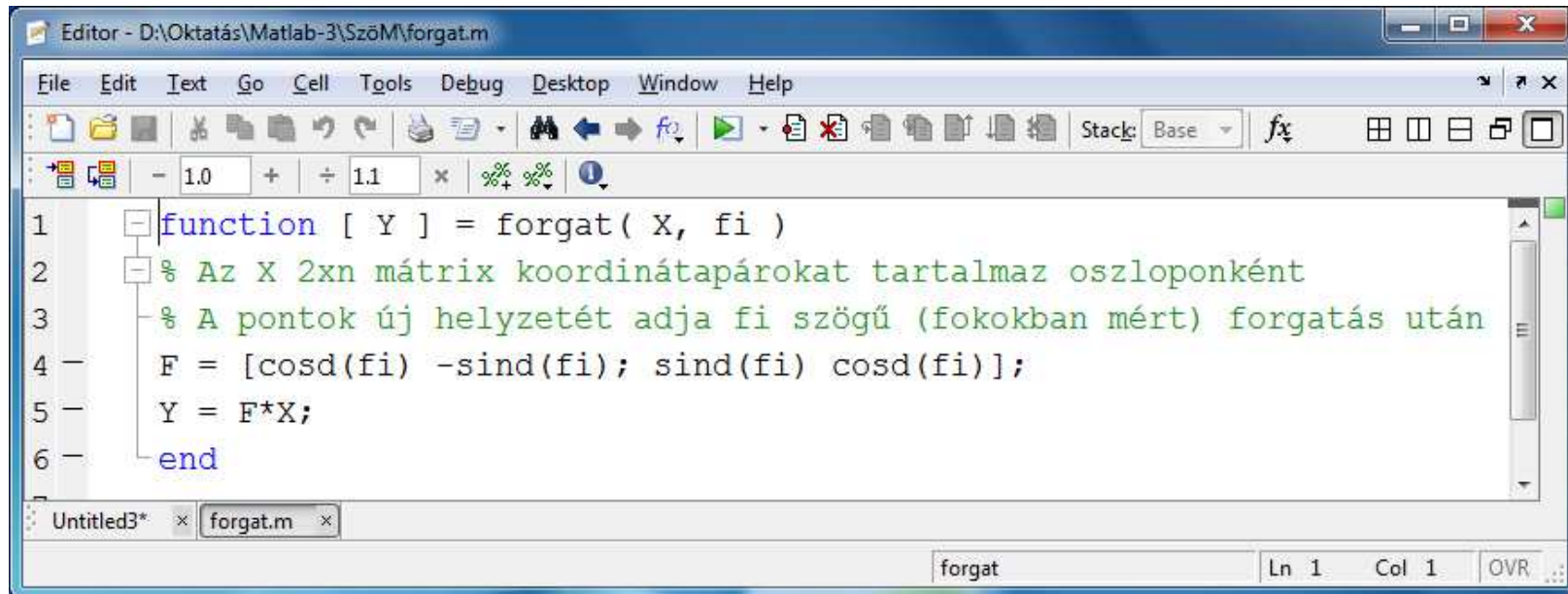
Példák (folyt.)

(Házikó trafói, folyt.)

- Síkbeli pontsorozat összekötés (már néztük)
- Transzformációk mátrixszal
 - Forgatás (lásd saját fv.)
- Új pozícióban kirajzolás
- Bemutató (folyt.)

- Hívás:

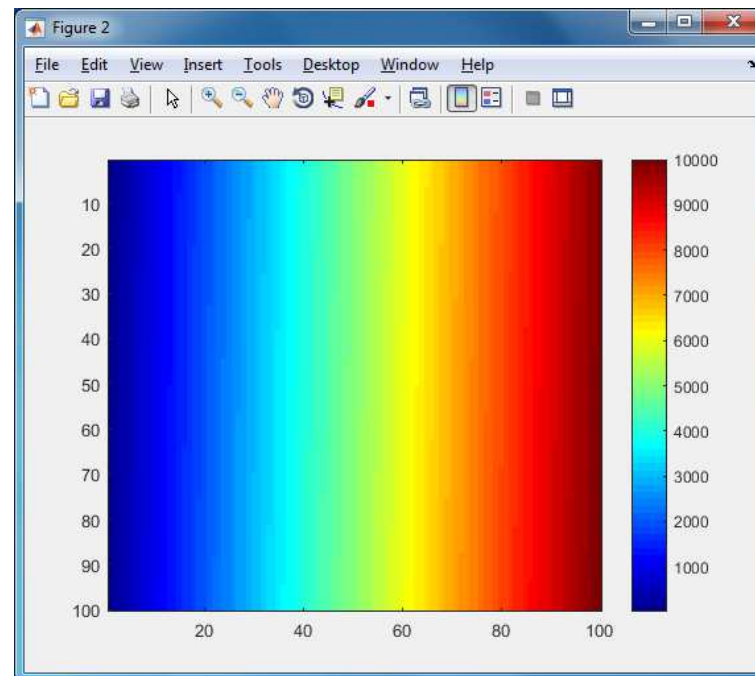
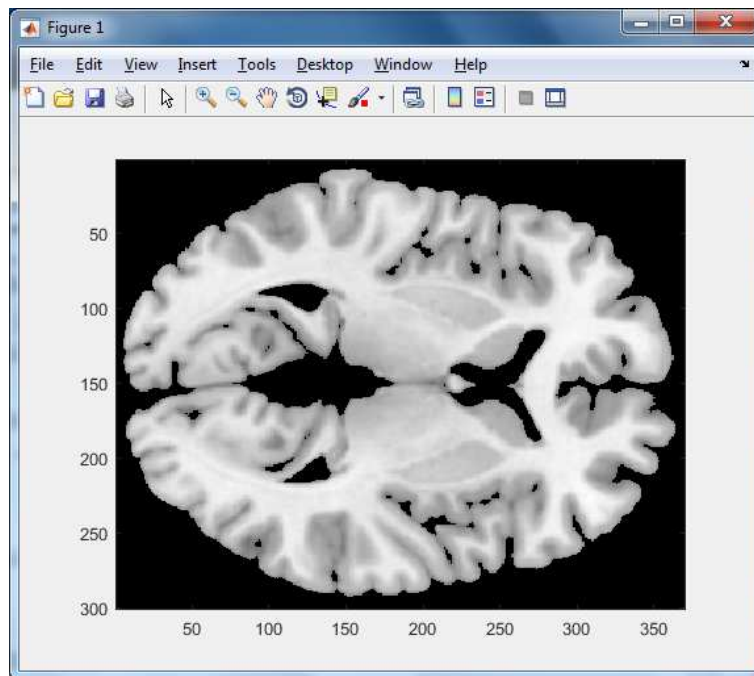
```
>> for i=0:360 dot2dot(forgat(X,i)), pause(0.1), end;  
% függvények egymásba is ágyazhatók
```



```
Editor - D:\Oktatás\Matlab-3\SzöM\forgat.m  
File Edit Text Go Cell Tools Debug Desktop Window Help  
- 1.0 + ÷ 1.1 x % % %  
1 function [ Y ] = forgat( X, fi )  
2 % Az X 2xn mátrix koordináta párokat tartalmaz oszloponként  
3 % A pontok új helyzetét adja fi szögű (fokokban mért) forgatás után  
4 F = [cosd(fi) -sind(fi); sind(fi) cosd(fi)];  
5 Y = F*X;  
6 end  
Untitled3* x forgat.m x  
forgat Ln 1 Col 1 OVR
```

Grafika – áttekintés

- A Matlab a rajzokat grafikus ablakban készíti el
 - Válgathatunk a már létező, vagy még nem létező ablakok között
 - Utóbbi esetben új ablak jön létre
 - Parancs: *figure(n)*
- Egy grafikus ablakban sok objektum helyezhető el, amelyek mindegyike hangolható
 - Lásd még: *hold on*, ill. *subplot*
- Többféle rajzoló utasítás használható





Plot

- Síkbeli pontokat a *plot* paranccsal rajzoltathatunk
- Megadandók az egyes pontok x ill. y koordinátái
 - Külön vektorban! (1. és 2. param.)
 - Egyszerű példa: `plot([0 1], [0 1])`
- Alapértelmezés szerint ...
 - ... a pontok folytonos vonallal összekötöttek
 - ... nincs jelölő
 - (... a vonal színe először kék, majd piros stb.)
- A parancs (opcionális) harmadik paraméterében opciókat adhatunk át
 - Ezek az alapértelmezést módosítják
 - Három különböző tulajdonságot adhatunk meg egy néhány karakter hosszú kódértékekből összeállított *sztringben*
 - Vonalstílus, szimbólum, szín
 - A tulajdonságok együtt vagy külön is megadhatók
- Példa
 - `>> x = rand(1,3), y = rand(1,3), x(1,4) = x(1,1),
y(1,4) = y(1,1), plot(x,y, '*-.b')`





Plot

- A plot parancs lehetséges harmadik paraméterei
 - Összefoglaló táblázat

kód	vonal stílusa	kód	szimbólum	kód	szín
-	folytonos (alapértelmezés)	+	+	r	piros
--	szaggatott	o	kör	g	zöld
:	pontozott	*	*	b	kék
-.	pont-vonal	.	pont	c	cián
		x	x	m	magenta
		s	négyzet	y	sárga
		d	rombusz	k	fekete
		^	háromszög fel	w	fehér
		v	háromszög le		
		>	háromszög jobbra		
		<	háromszög balra		
		p	ötágú csillag		
		h	hatágú csillag		





Plot

- Példa a marker szimbólumok használatára

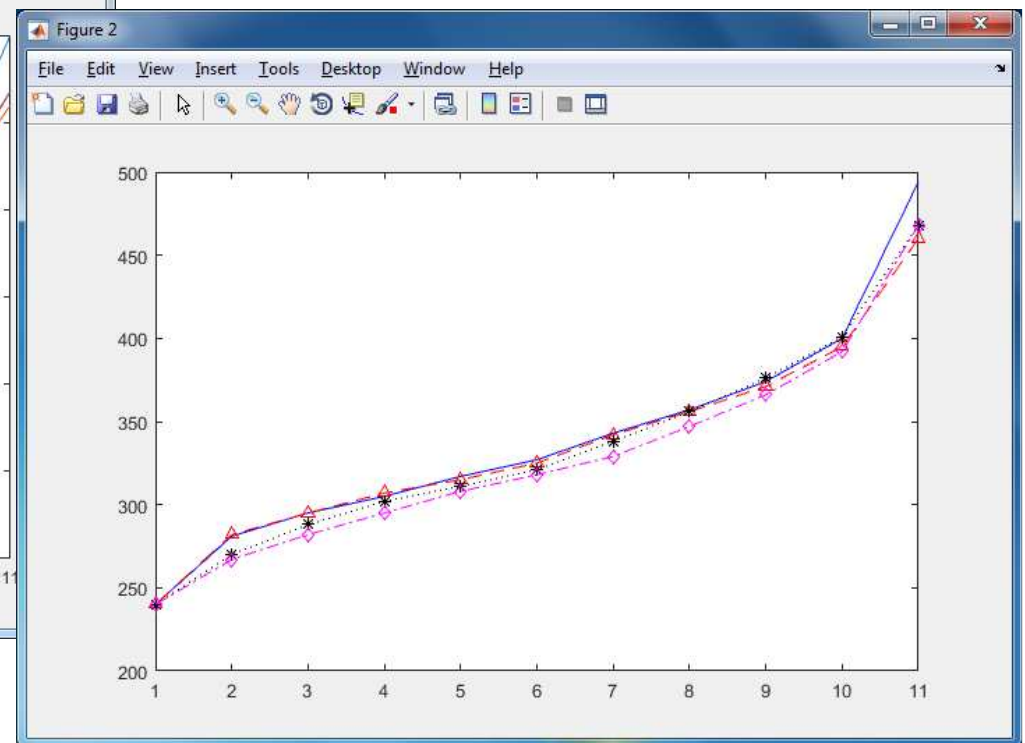
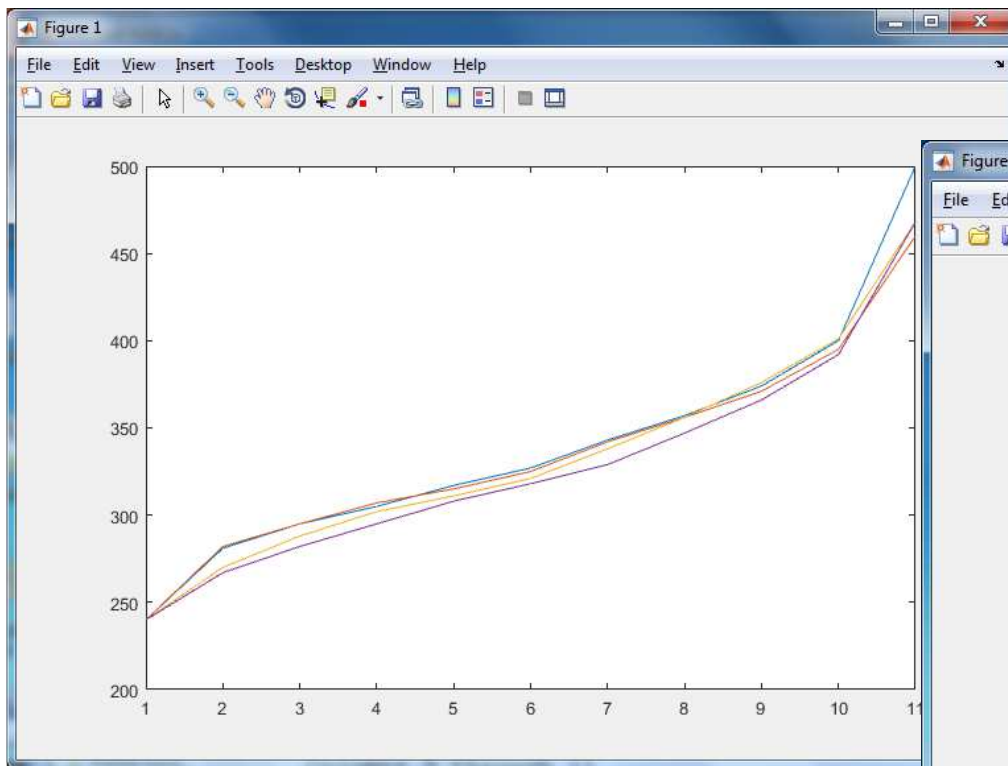
- Csoportok (decilisekre bontva) és pontok

- Szintaktika (részlet, 2. ábra)

```
>> plot(x, mi, 'b-')
```

```
>> plot(x, gi, 'r^--')
```

```
>> plot(x, vi, 'k*:')
```





Plot

További plot lehetőségek

- Finomabb beállítások (szintaxis):
`plot(x, y, tulajdonság neve, tulajdonság értéke)`
- Több tulajdonságot is megadhatunk név–érték párokat ismételve
- Néhány fontosabb:

Color	A vonal színe	Színkód (lásd előző táblázat), vagy RGB kódolással: 3 számérték (vektorban), 0 és 1 közötti valós számok
LineStyle	A vonal stílusa	Mint a fenti táblázatban: - - : - .
LineWidth	A vonal vastagsága	Az érték megadható

- További tulajdonságok (pl. jelölőre vonatkozó beállítások, színek):
lásd súgó, ill.
`h = plot(x, y, ...), get(h), set(h, 'tulajdonság neve', tulajdonság értéke)`

- Példa

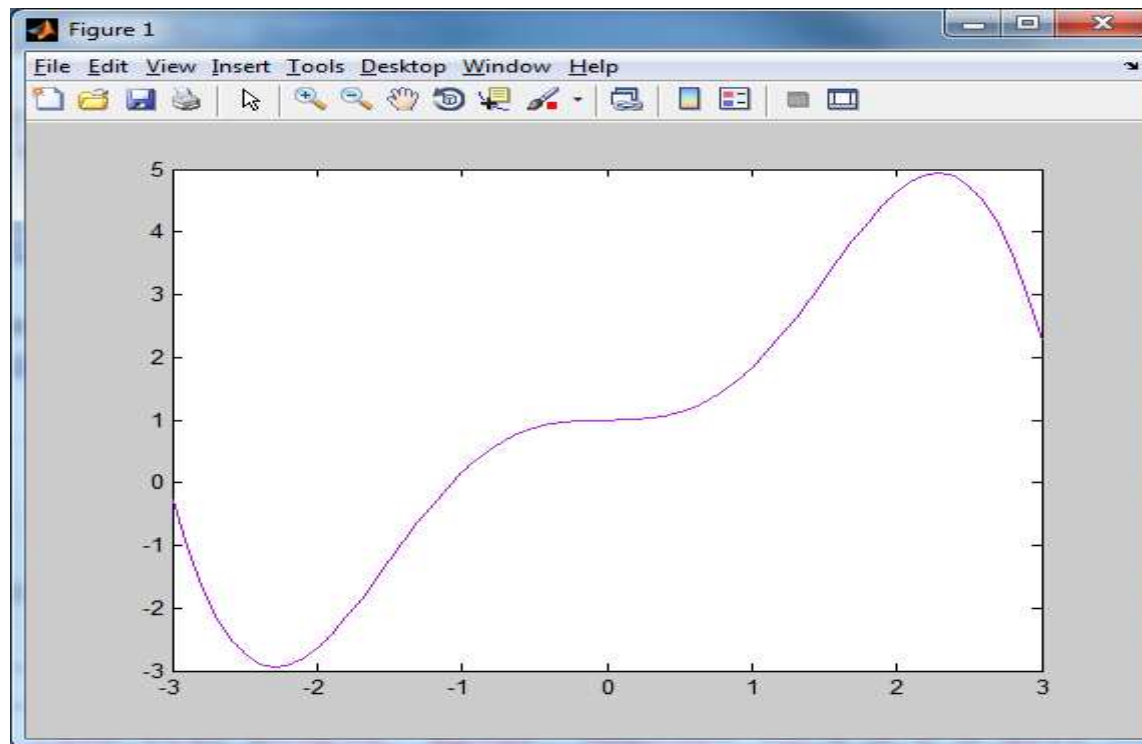
```
>> x1 = linspace(-2*pi, 2*pi, 101);  
plot(x1, sin(x1), 'LineWidth', 2, 'Color', [.8 .3 0]);
```



Függvényábrázolás plot-tal

- „Kotta”, az előzőek szerint:
 - Alappontok (vektor) és fv.definíció elkészítése (valamely megismert módon)
 - Rajzoló utasítás megadása az esetleges opciókkal
 - Ha több rajzot szeretnénk egy ábrára helyezni, akkor ezt is beállítjuk (*hold on*)
- Példa (korábbi anonimus függvény)

```
>> fw = @(x) x.^2.*sin(x)+1  
>> s = -3:0.1:3; plot(s, fw(s), 'Color', [0.7 0.1 0.9])
```

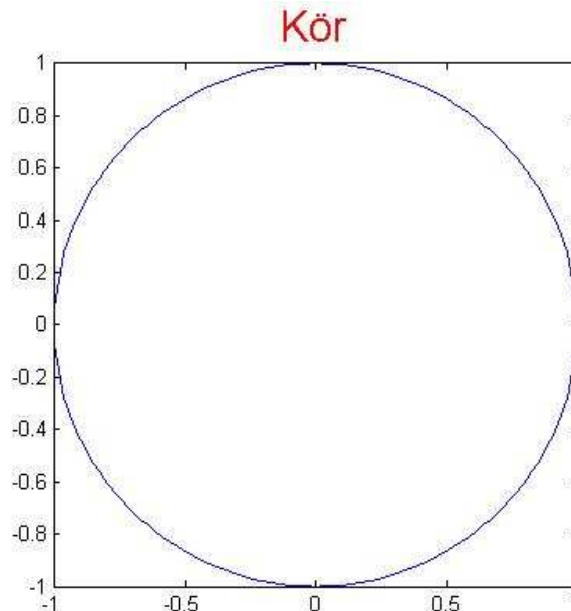




Függvényábrázolás plot-tal

- A koordinátpárokat komplex számsorral is megadhatjuk (paraméteres ábrázolás)
- Példa

```
z = exp(i*linspace(0, 2*pi, 181)); plot(z);  
axis square;  
% az ábra méretezése négyzetes  
h = title('Kör', 'Color', 'R');  
% felirat az aktív rajzra  
% az objektum mutatót feljegyezzük  
get(h);  
% megjelenítjük az objektum hangolható tulajdonságait  
set(h, 'FontSize', 20) % betűméret beállítása
```





További rajz parancsok

A rajzok készítéséhez használható további fontosabb parancsok (hangolás, kieg.)

- **clf**: grafikus ablak tartalmának törlése
- **axis([xmin xmax ymin ymax])**: a koordinátarendszer határainak beállítása a megadott értékekre
- **axis auto**: a határok visszaállítása az automatikus beállításra
- **axis equal**: az x és y irányú egység azonos hosszúságú
- **axis off/on**: a koordinátarendszert elrejt, bekapcsolja
- **grid on/off**: rács be- és kikapcsolása
- **title('cím')**: grafika címének magadása
- **xlabel('felirat'), ylabel('felirat')**: x/y-tengely felirata
- **text(x,y,'string')**: adott pozícióra kiír egy szöveget
- **legend('string', ..., pozíció)**: jelmagyarázat megadása
 - A poz. újabb Matlab verziókban szövegesen adandó meg (!), régebben számmal
- **hold on/off**: a meglevő ábrába helyezi el a következő fv.rajzot, vagy újba
- **figure(n)**: új ablak létrehozása, illetve váltás egy létező ablakra, ahol *n* az ablak mutatója (poz. egész)
- **gcf**: az aktuális ablak mutatóját adja vissza

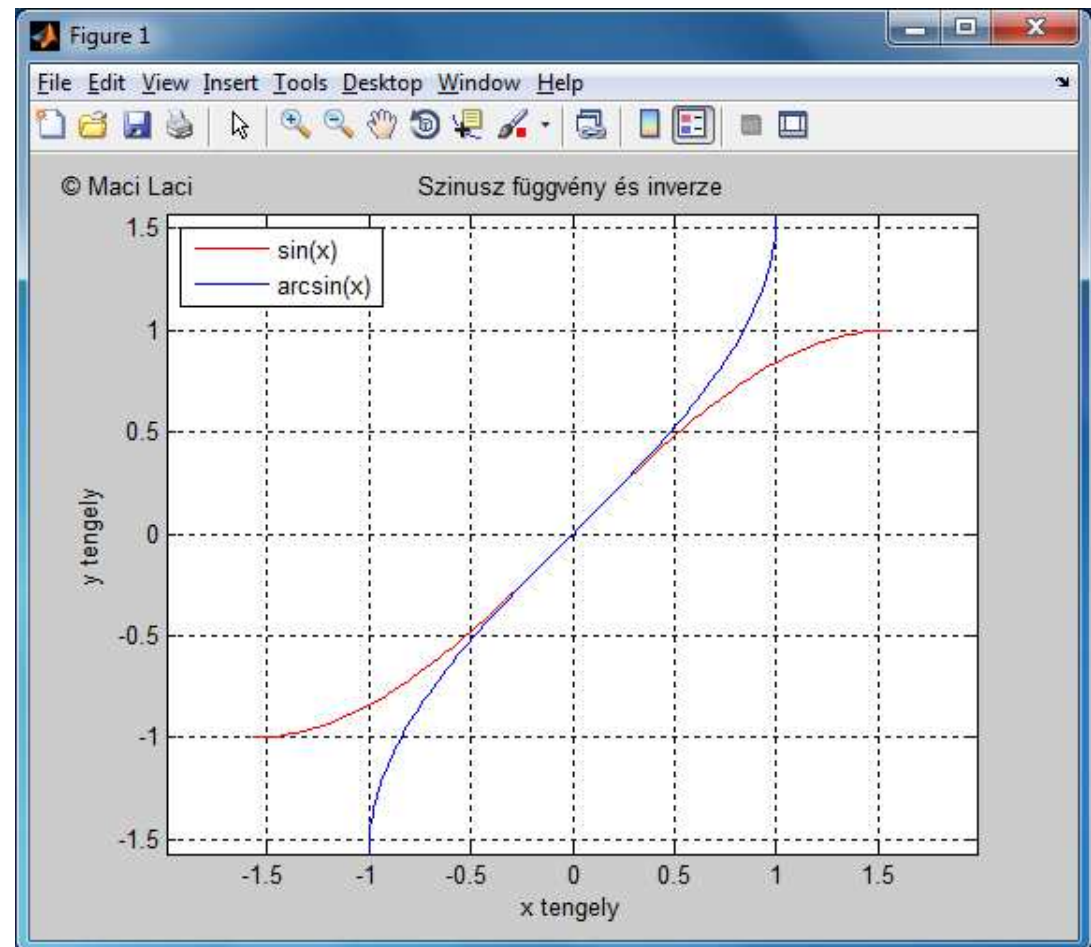


Függvényábrázolás plot-tal

- Összetett plot példa (függvény és inverze)

```
>> x = linspace(-pi/2, pi/2, 101); y = sin(x);  
>> plot(x, y, 'r'); hold on; plot(y, x, 'b'); axis auto  
>> xlabel('x tengely'), ylabel('y tengely')  
>> title('Szinusz  
függvény és inverze')  
>> text(-2.5, 1.7,  
    '@ Maci Laci')  
% szöveg, koordinátával  
>> legend('sin(x)', ...  
    'arcsin(x)', 'Location',  
    'northwest');  
% pozíció megadása  
% régebben 1-4  
>> grid on,  
axis equal, hold off;
```

- A jelmagy. színezése a plotok sorrendjét követi





Animáció plot-tal

- Animáció lehetősége: a rajzot mindig újrarajzoljuk és közben szüneteket tartunk
- Egyszerű példa – egységvektor körbeforgatása

```
>> x = [0 1; 0 0];
% 1 hosszú vektor végpontjai az x tengelyen
>> fok = 2;
>> T = [cosd(fok) -sind(fok); sind(fok) cosd(fok)];
% forgatás mátrixa, 2 fokonként megyünk
>> for fi = 0:fok:360
    plot(x(1,:),x(2,:),'.-'), axis([-1 1 -1 1]);
    % rajz
    pause(0.1); % várakozás
    x = T*x;
    % az x vektor új pozíciója
end
% ciklus vége
```
- Most nem kell *hold on* parancs!
 - Minden lépésben új vásznat rajzolunk (ugyanoda), a régit pedig eldobjuk

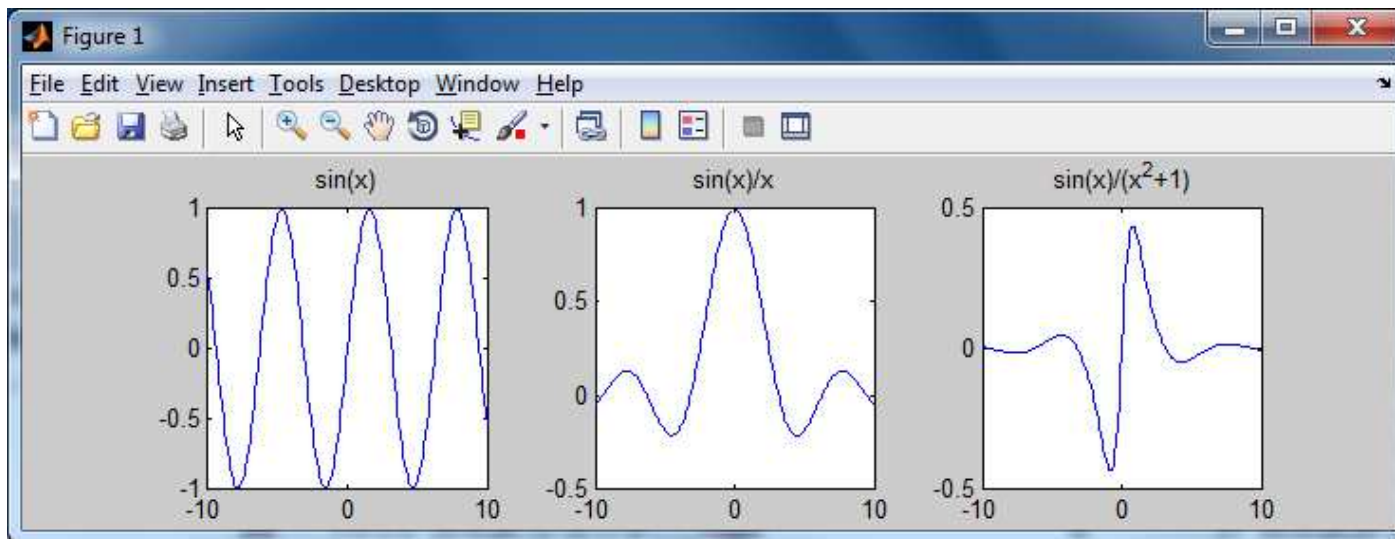




Subplot

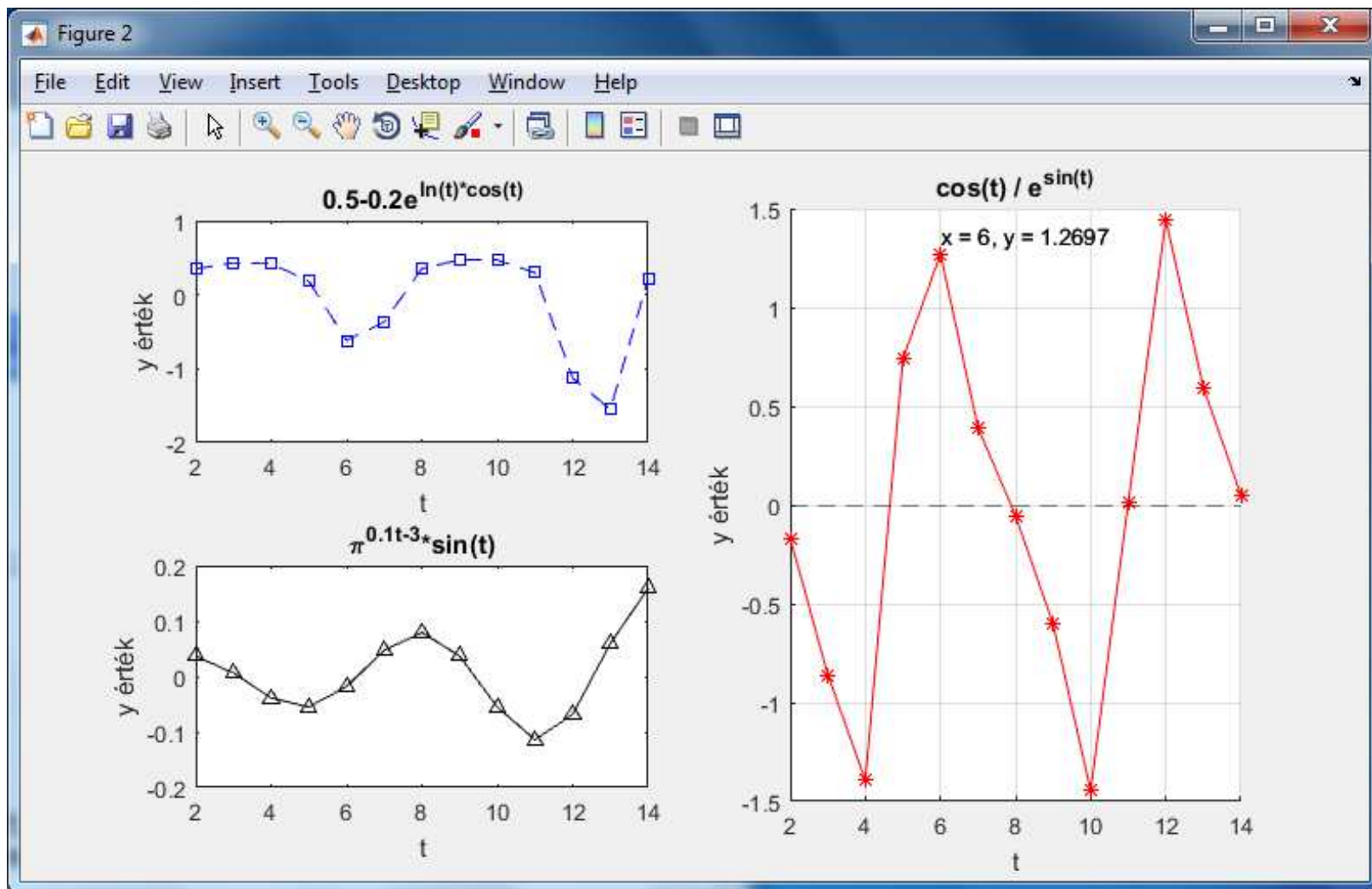
- *Subplot*: egy képtáblázat megfelelő cellájába illesztjük be a köv. plot rajzot
- Szintaktika: a `subplot(n, m, p)` az n sorú és m oszlopú képtáblázat p -edik, sorfolytonosan számozott celláját aktiválja. A plot itt helyezi el a köv. képet.
- Példa: egy 1×3 méretű képtáblázatba három függvényrajzot helyezünk el

```
x = -10:0.1:10; figure(1),  
y = sin(x); subplot(1,3,1); plot(x,y); axis square,  
title('sin(x)');  
y = sin(x)./x; subplot(1,3,2); plot(x,y); axis square,  
title('sin(x)/x');  
y = sin(x)./(x.^2+1); subplot(1,3,3); plot(x,y); ...
```



Subplot

- Összetett subplot példa (M. D.)





Subplot

- A mögöttes kód (részlet, elemzés Hf.)

```
t = 2:14;  
subplot(221);  
plot(t, 0.5-0.2*exp(log(t).*cos(t)), 'bs--');  
xlabel('t'); ylabel('y érték');  
title('0.5-0.2e^{ln(t)*cos(t)}');  
xlim([t(1) t(end)]);  
subplot(223);  
plot(t, pi.^(0.1*t-3).*sin(t), 'k^--');  
xlabel('t'); ylabel('y érték');  
title('\pi^{0.1t-3}*sin(t)');  
xlim([t(1) t(end)]);  
subplot(2,2,[2 4]);  
hold on;  
plot(t, cos(t)./exp(sin(t)), 'r*-');  
plot([t(1) t(end)], [0 0], 'k--');  
xlabel('t'); ylabel('y érték');  
title('cos(t) / e^{sin(t)}');  
xlim([t(1) t(end)]);  
...
```





Fplot

- Az *fplot* utasítás segítségével ismert/megadott egyváltozós függvények grafikonját rajzoltathatjuk ki
- Eltérés a *plot* utasítástól: a függvény változó, automatikus osztással kerül kiértékelésre
 - A gyorsabb függvényérték-változásoknál sűrűbben, máshol ritkábban
 - x alappontokat így külön nem kell készítenünk
 - Ezáltal a grafikus kép is lehet jóval kifejezőbb (eml.: „rossz” plot)
- Példa

```
function y = fw(t);  
    y = t.*exp(-t.^2).*sin(4*t);  
end
```
- Hívás:

```
>> fplot('fw', [0 pi])  
% a függvény nevét aposztrófok határolják, vagy:  
>> fplot(@fw, [0 pi])
```
- Jellegzetességek
 - *fplot*-nál nem kötelező pontozott műveleteket használni, *de javasolt* (warning az új verziókban)
 - Aposztrófos megadás: 2017 utáni Matlab verziókban már nem lesz támogatott, javasolt helyette a *@*-ra áttérés!
 - Anonymus (és inline) megadásnál is használhatunk *fplot* rajzolást
 - Az intervallumhatárokat nem kötelező megadni, *de javasolt* (régebben kötelező volt)

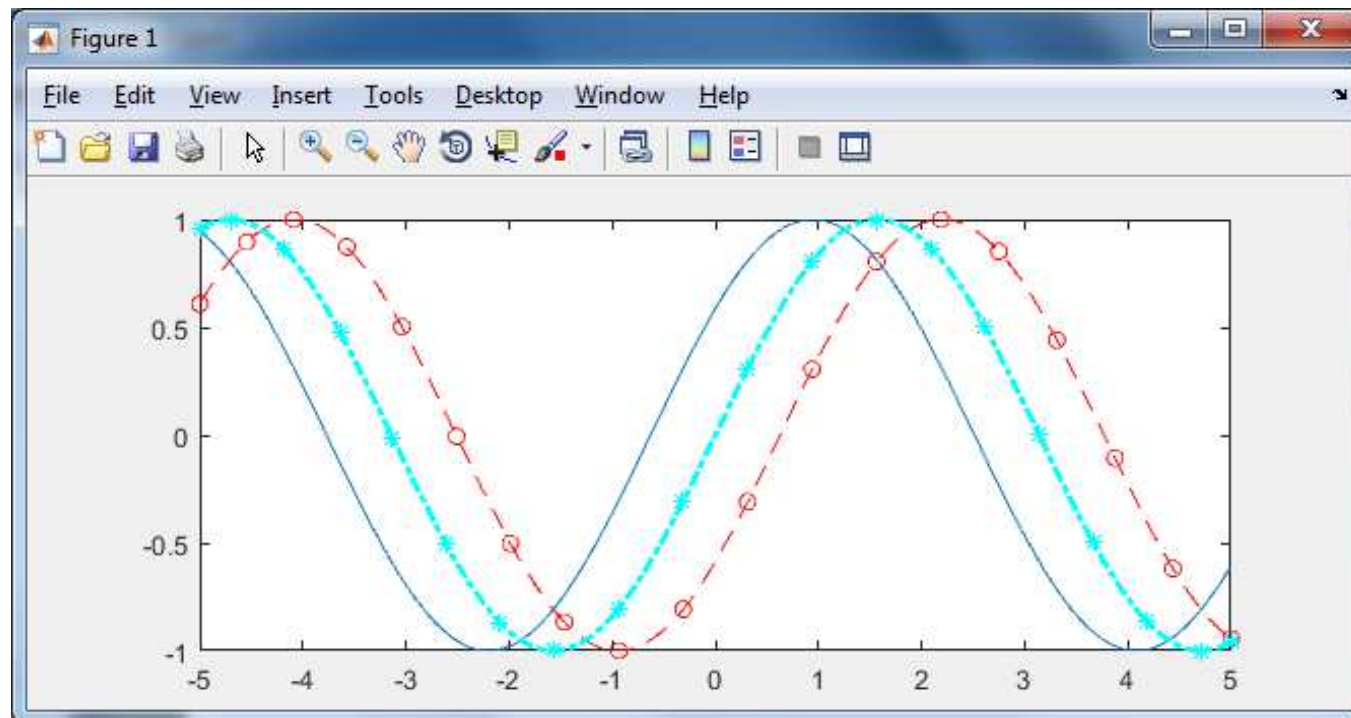


Fplot

- Az fplot-nál is ugyanúgy szabályozható a vonalszín, -stílus és a jelölők használata, mint a plot esetén

- Példa

```
>> fplot(@(x) sin(x+pi/5)); % sima rajz  
>> hold on  
>> fplot(@(x) sin(x-pi/5), '--or');  
>> fplot(@(x) sin(x), '-.*c', 'Linewidth', 2)  
% pontozott vonal jelölővel, vastagabb
```



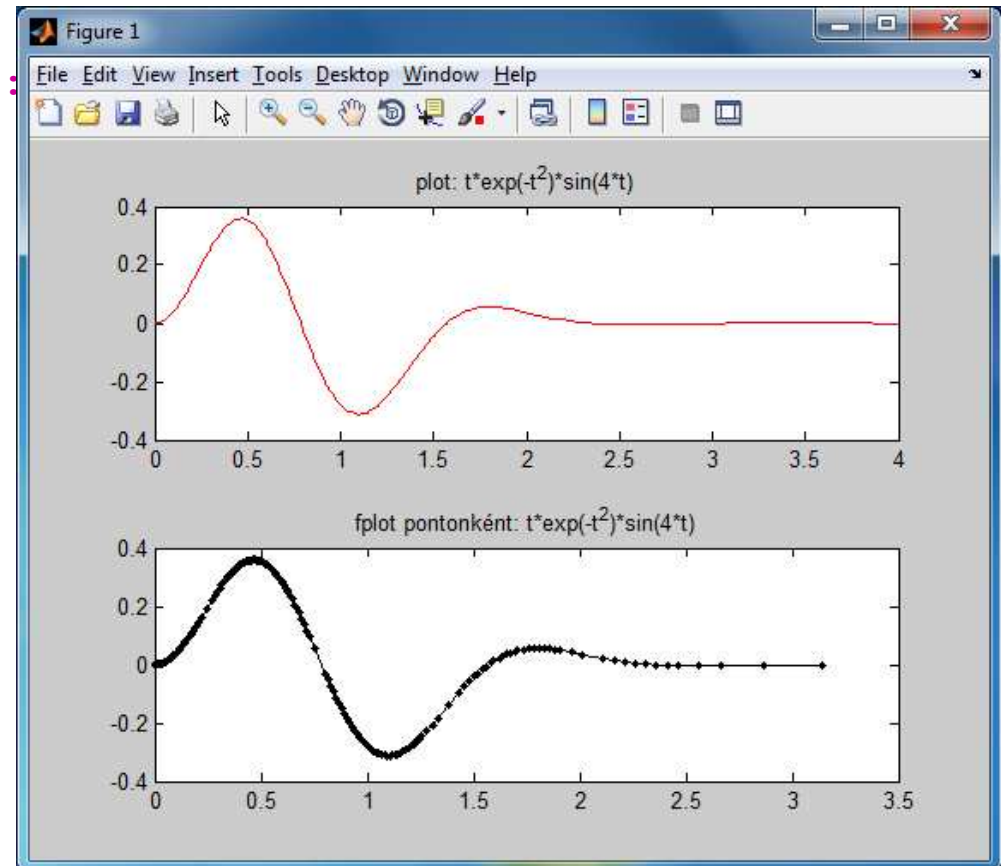
Fplot

Érdekesség: az *fplot* belső működése

```
>> subplot(2,1,1); fplot('fw',[0,4], 'r');  
% sima vonalas függvényrajz  
>> title('plot: t*exp(-t^2)*sin(4*t)')  
>> [x,y] = fplot('fw', [0, pi]); subplot(2,1,2);  
plot(x,y,'k.-');  
>> title('fplot pontonként: t*exp(-t^2)*sin(4*t)')
```

Magyarázat

- Ha az *fplot* hívás $[x \ y]$ visszaadott értékeit eltároljuk, akkor a grafikon nem jelenik meg
- Így az *fplot*hoz szükséges x, y párokat feljegyezve, egy *plot*tal csak az x, y párokkal adott pontsorozatot rajzoltatjuk ki
- + összehasonlító ábra: *sima fplot*



Ezplot

- **ezplot („easy-plot”):** egyszerű segédeszköz a közvetlen megadású egyváltozós, az implicit megadású és a paraméteresen adott függvények ábrázolásához
 - Az ábrázolási intervallumhatárok elhagyhatók, ekkor azokat a Matlab automatikusan generálja
 - A függvény automatikusan ábracímet is készít

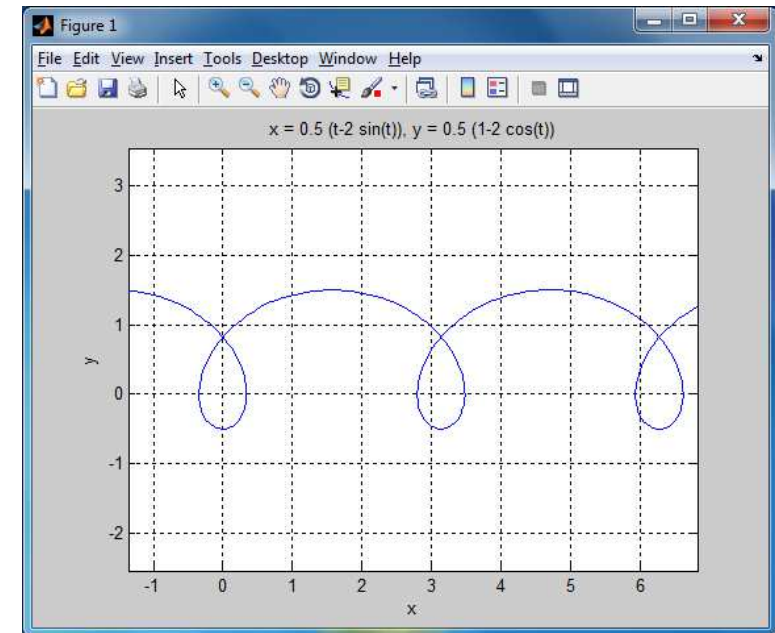
Példák

- Egyváltozós függvény


```
>> ezplot('fw')
>> ezplot('sin(x)/x', [-4*pi 4*pi])
```
- Implicit megadású függvény:


```
>> ezplot('(x-1)^2/3^2 + (y-2)^2/2^2 = 1', [-3 5, -1 5]),
grid on % ellipszis
```
- Paraméteresen adott függvény:


```
>> ezplot('0.5*(t-2*sin(t))', '0.5*(1-2*cos(t))',
[-3 15]), grid on % ciklois
>> ezplot('t*cos(t)', 't*sin(t)', [0 4*pi]), grid on
% archimédeszi spirális
```





Ezplot

- Szimbolikusan adott függvényeket az ez kezdetű parancsokkal lehet kirajzoltatni (*ezplot*, *ezcontour*, *ezsurf*, *ezplot3* stb.)
- A szimbolikus megadás nagy előnye: a függvények *formálisan deriválhatók*, *integrálhatók* stb.
- Most csak az *ezplot* ilyen típusú használatát nézzük meg

- Példa

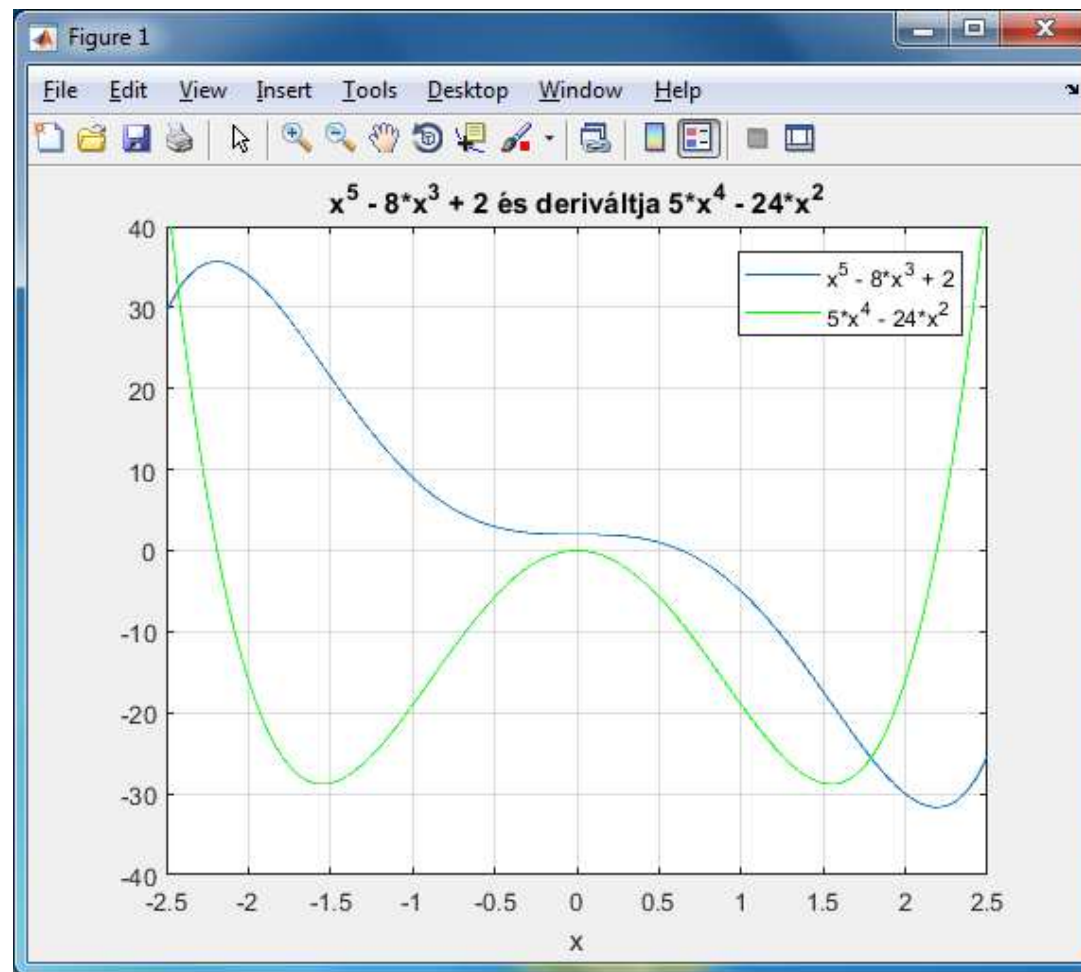
```
syms x; % x szimbolikus változó
fp = x^5-8*x^3+2 % fp függvény (szimbolikusan)
fp_diff = diff(fp, x)
% fp függvény deriváltja (szimbolikusan)
ezplot(fp, [-2.5 2.5]), grid on, hold on % fp rajz
h = ezplot(fp_diff, [-2.5 2.5]);
% fp_diff rajz, objektumkezelő feljegyzés
set(h, 'Color', 'g') % objektum színének változtatása
axis([-2.5 2.5 -40 40]) % intervallumhatárok beállítása
title('x^5 - 8*x^3 + 2 és deriváltja 5*x^4 - 24*x^2')
legend('x^5 - 8*x^3 + 2', '5*x^4 - 24*x^2', 'Location',
'northeast')
% cím, jelmagyarázat (jobb felső sarok)
```





Ezplot

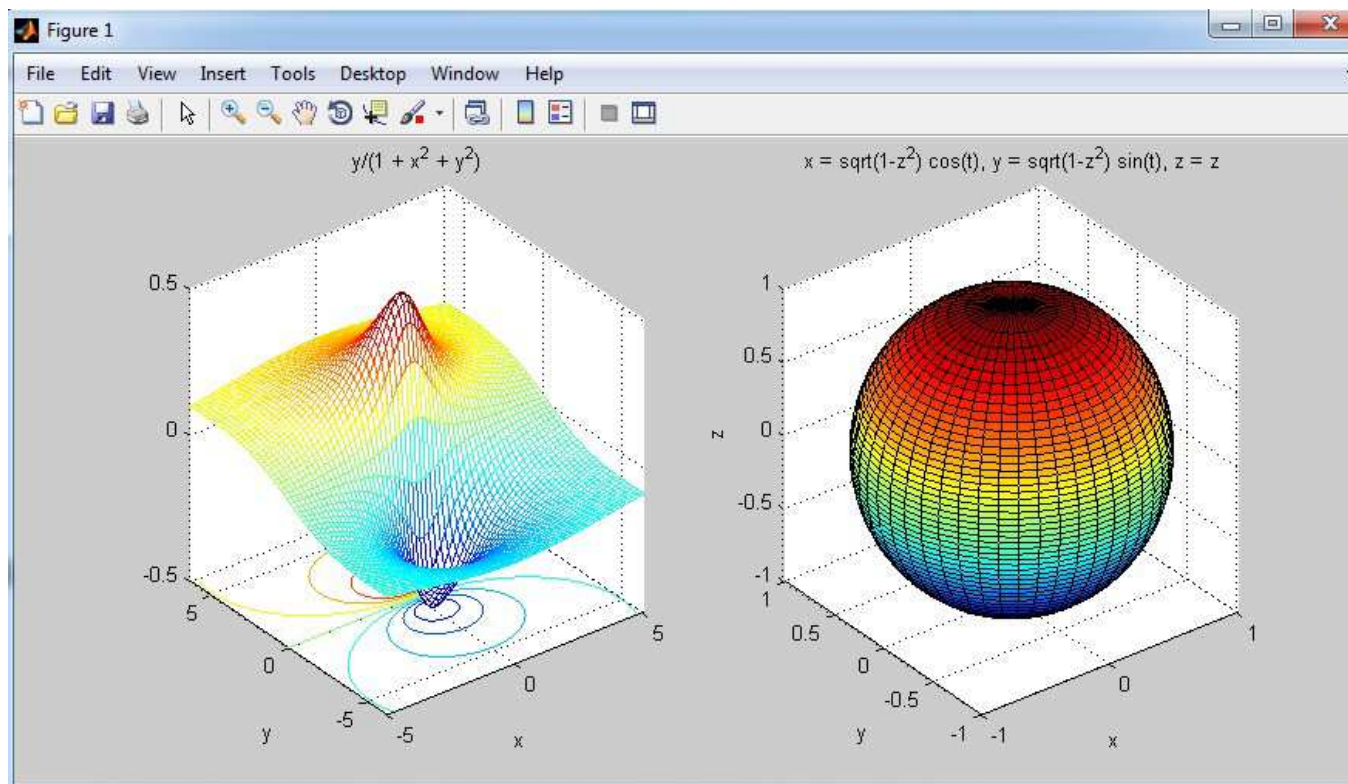
- Az eredmény (szép rajz):



Többváltozós függvények ábrázolása

- Szintén egyszerűen lehet ábrázolni kétváltozós függvényeket és térbeli felületeket a megfelelő ez kezdetű függvénnyel
- Ábrázoljuk az $f(x, y) = y/(1 + x^2 + y^2)$ függvényt a $-5 < x < 5$, $-2\pi < y < 2\pi$ tartomány felett, majd ábrázoljuk az egységsugarú gömböt is!

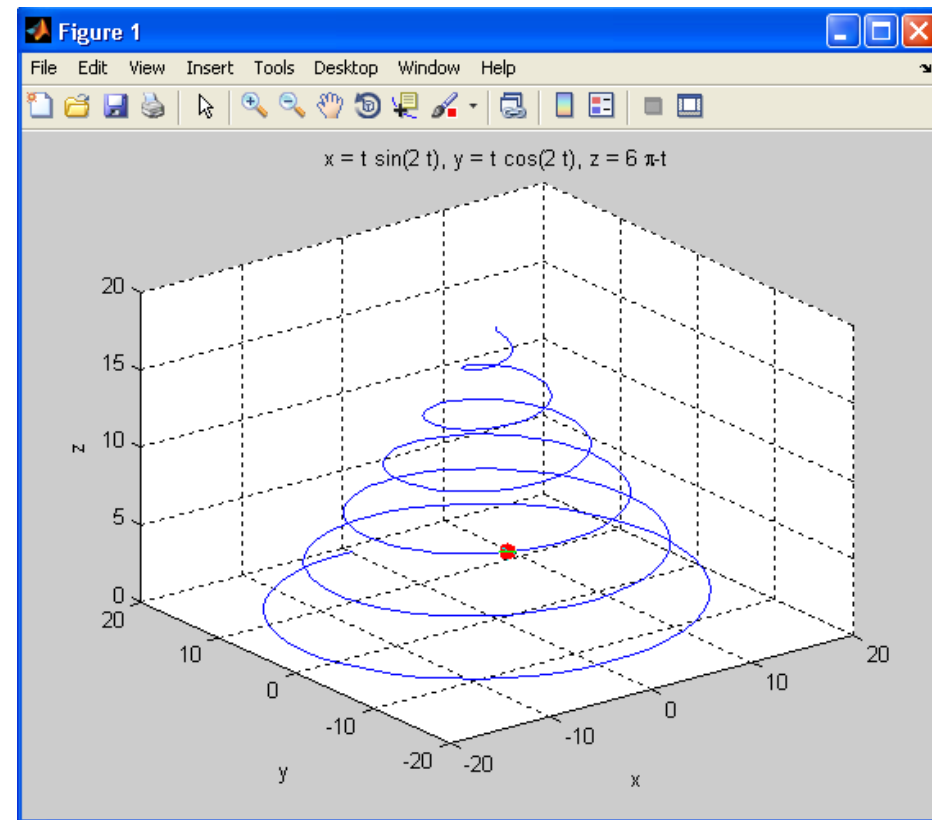
```
>> subplot(1,2,1), ezmesh('y/(1 + x^2 + y^2)', [-5,5, -  
2*pi,2*pi])  
>> subplot(1,2,2), s='sqrt(1-z^2)*'; h=ezsurf([s  
'cos(t)'], [s 'sin(t)'], 'z', [0 2*pi, -1 1])
```



Többváltozós függvények ábrázolása

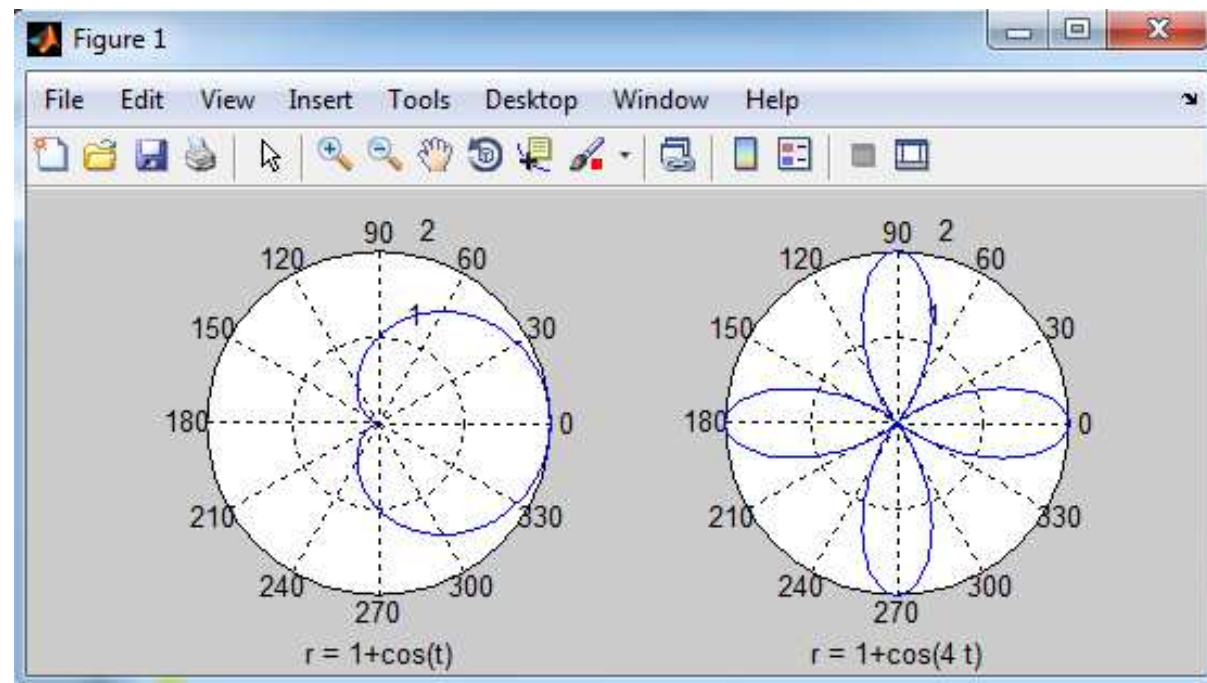
- A paraméteresen adott térgörbék rajzolásához az `ezplot3(funx,funy,funz,[tmin,tmax])` függvény használható
- Az *animate* opcióval (ez extra paraméter) a térgörbén végigfutó pontot animálhatjuk is
- Példa: az $x = t \sin(2t)$, $y = t \cos(2t)$, $z = 6\pi - t$ egyenletű térgörbe egy álló kúp felületén körkörösleg legördülő pont pályáját írja le
- Ábrázoljuk és animáljuk!

```
>> ezplot3('t*sin(2*t)', 't*cos(2*t)', '6*pi-t', [0,6*pi],  
          'animate')
```
- (További többdim. példák: lásd sűgó)



Polárkoordinátás függvények ábrázolása

- Az $r = r(\varphi)$ polárkoordinátás megadás gyakran hasznos lehet
- Ilyenkor a síkgörbe paraméteres alakja:
 $x(\varphi) = r(\varphi) \cdot \cos(\varphi), \quad y(\varphi) = r(\varphi) \cdot \sin(\varphi)$
- Ezek megjelenítésére az `ezpolar(r(t), [a, b])` parancs alkalmas
- Jelenítsük meg az $r = 1 + \cos(t)$ és az $r = 1 + \cos(4t)$ függvényeket!
 - Ha az intervallumot nem adjuk meg, akkor $0 \leq t \leq 2\pi$ lesz
- ```
>> subplot(1,2,1), ezpolar('1+cos(t)'), subplot(1,2,2),
ezpolar('1+cos(4*t)')
```





## Ez-parancsok, kiegészítés

### Kiegészítő megjegyzés

- Az ez-parancsokat a 2017 utáni Matlab verziókban már nem fogják támogatni, ezért célszerű (lassan, folyamatosan) átszokni a megfelelő „kiváltó” parancsokra!
  - Lásd súgó
- Kiváltási példák
  - `>> fplot(@fw)`
  - `>> fimplicit(@(x,y) (x-1).^2/3^2 + (y-2).^2/2^2 - 1, [-3 5, -1 5]), grid on`  
% átalakítás kellett, a kifejezésben is
  - `>> fmesh(@(x,y) y./(1 + x.^2 + y.^2), [-5,5,-2*pi,2*pi])`  
% pontozott műv. nélkül warning
  - `>> fplot3(@(t) t.*sin(2*t), @(t) t.*cos(2*t), @(t) 6*pi-t, [0,6*pi])` % az utóbbi 3 fv. 2016-tól használható  
% animáció nélkül

### ezplot

(Not recommended) Easy-to-use function plotter

[collapse all in page](#)



ezplot is not recommended. Use [fplot](#) instead.

### ezmesh

(Not recommended) Easy-to-use 3-D mesh plotter

[collapse all in page](#)

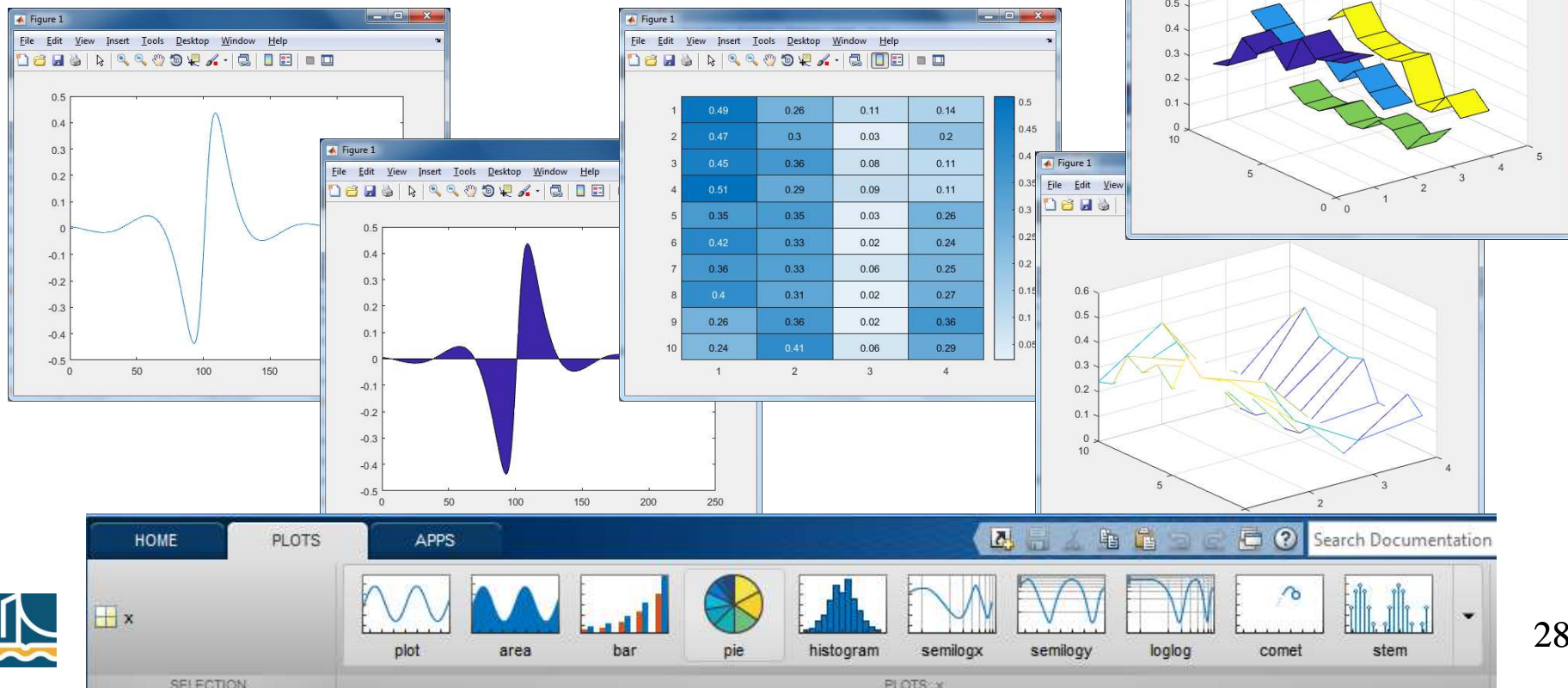


ezmesh is not recommended. Use [fmesh](#) instead.



## Gyors grafika

- A menüszalag (Plot, gyors eszköztár) azonnali lehetőséget biztosít a direkt rajzolásra
- Egyszerűen kiválasztjuk az objektumot és az alkalmazni kívánt rajztípust!
  - Mások a lehetőségek egy- és többváltozós esetben
- Szemléltető példák





## Függvényvizsgálat

### Klasszikus függvényvizsgálat

- Lépések a „nagykönyv” szerint:
  - Értelmezési tartomány meghatározása
  - Zérushelyek, határértékek
  - Monotonitás, szélsőértékek (első derivált vizsg.)
  - Konvexitás, inflexiós pontok (második deriv. vizsg.)
  - Értékkészlet, függvényábra
- A Matlab mindegyik ponthoz *rendkívül széles körű támogatást* ad! (megfelelő parancsok)
  - De bizonyos matematikai tudás nem nélkülözhető...
  - Példa (tudjuk)  

```
>> x=-pi:0.01:pi; plot(x, tan(x))
```
- Specialitások
  - Zérushelyek, lokális szélsőértékek keresése: megfelelően közeli pontból indulva (adott intervallumon)
  - Két függvény metszéspontja: *a különbségfüggvény zérushelye*
  - Lokális maximum megkeresése: az ellentett fv. lokális minimumát keressük
  - A függvényábra bővíthető az új nevezetes pontokkal (tudjuk)





## Függvényvizsgálat

### Zérushelyek, metszéspontok

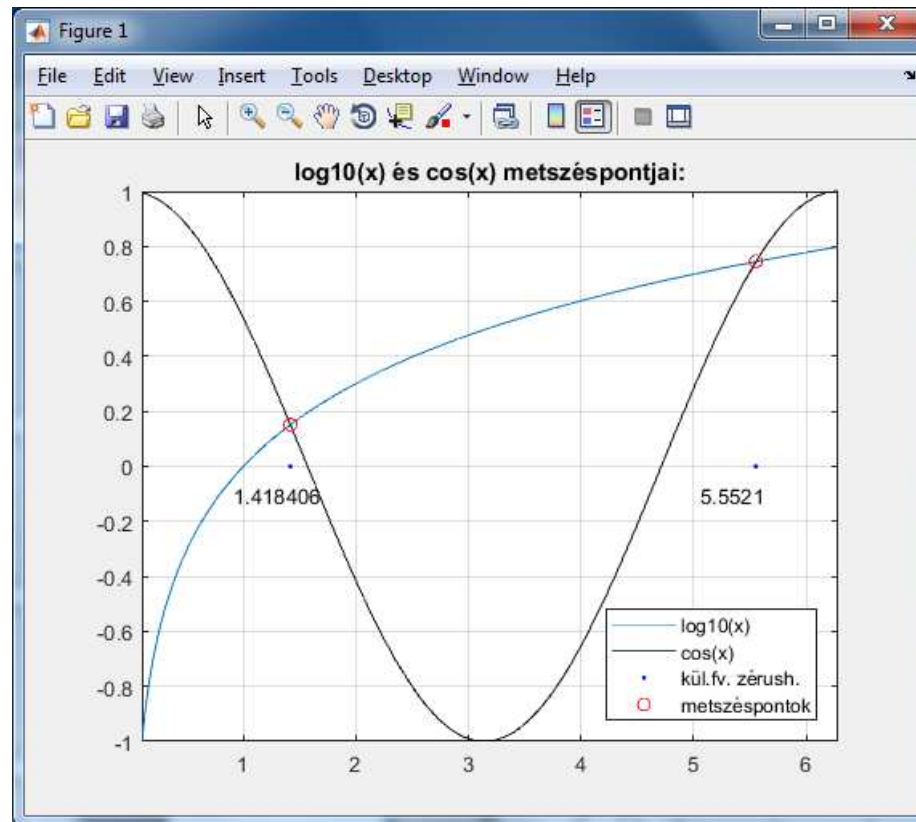
- Két függvény metszéspontjának, vagy egy függvény zérushelyeinek a meghatározása:
  - Ábrázolás megfelelő méretű intervallumon
  - Metszéspont (zérushely) keresés egy alkalmas közeli pontból (*fzero*), ábrázolás
- Példa: Határozzuk meg a  $\log_{10}x$  és  $\cos(x)$  függvények metszéspontjait a  $[0 \ 2\pi]$  intervallumban!
- ```
>> fplot(@log10, [0.1 2*pi]), hold on,
fplot(@cos, [0.1 2*pi], 'k'), grid on,
title('log10(x) és cos(x) metszéspontjai:')
fd = @(x) log10(x)-cos(x) % különbségfüggvény
>> x = [fzero(fd, 2) fzero(@(x) log10(x)-cos(x), 6)]
% zérushelyek keresése
>> plot(x, fd(x), '.b'), plot(x, cos(x), 'or')
% kék pont: a kül.fv zérushelyei, piros kör: metszéspontok
>> text(x(1)-0.5,-0.1,sprintf('%f',x(1))),
text(x(2)-0.5,-0.1,num2str(x(2)))
legend('log10(x)', 'cos(x)', 'kül.fv.
zérush.', 'metszéspontok', 'Location', 'southeast')
% jobb alsó sarokba jelmagyarázat
>> xd = x; % zérushelyek feljegyzése integráláshoz
>> quad(fd, xd(1), xd(2)) % num. integrálás, ld. később is
```



Függvényvizsgálat

Zérushelyek, metszéspontok (folyt.)

- Az eredmény:



- Megj.: Az *fzero* nemlin. kifejezések zérushelyének meghatározására nagyon jó eszköz! (Algoritmus, működés: lásd még súgó)





Függvényvizsgálat

Szélsőértékek

- Egyváltozós függvények minimumhelyét az *fminbnd* függvény segítségével határozhatjuk meg

- Legyen *f1(x)* definíciója *m*-fájlban:

```
function y = f1(x)
    y = x.^5-8*x.^3+2;    % a függvényt már láttuk
end %f1
```

- Először rajzolunk:

```
>> fplot(@f1, [-2.5 2.5]), grid on, hold on
```

- *fminbnd* paraméterek: a vizsgált függvény, az intervallum alsó és felső határa

- A zérushely közelítő helye letapogatható a grafikonról (data cursor, ill. *ginput*, lásd még: gyak.)

- A 4. paraméter az opciók beállítására szolgál, és az *fminbnd* valójában három értéket képes visszaadni

- ```
>> [mnx mny kod] = fminbnd('f1', 2, 2.5, optimset('TolX',
eps)), plot(mnx,mny,'ob') % minimumpont: kék kör
% az optimset beállítás az fzero-nál is használható
```







## Függvényvizsgálat

### Szélsőértékek (folyt.)

- A kód lehetséges értékei az *fminbnd* parancsnál
  - 1: az *fminbnd* konvergált az érvényes opciók mellett (minden OK)
  - 0: elértük a lépésszámkorlátozást
  - -1: függvény miatti terminálódás (pl. negatívból gyökvonás)
  - -2: inkonzisztens határok
- Függvény maximumának meghatározására *nincs külön parancs* (!), ezt a -1-szeresének minimumaként keressük:
- ```
>> mxx = fminbnd(@(x) -f1(x), -3, -2), mxy = f1(mxx),  
plot(mxx, mxy, 'or')  
% anonimus megadás ismert függvényekre hivatkozva!  
% de jó '-f1(x)' is  
>> hx=text(mxx+.2, mxy, sprintf('(% .2f, % .2f)', mxx, mxy));  
set(hx, 'fontweight', 'b', 'color', 'r')  
% piros és bold lesz a koordináta pár  
>> hn=text(mnx-.4, mny-3, sprintf('(% .2f, % .2f)', mnx, mny));  
set(hn, 'color', 'b') % kék koordináták  
>> title('f_1(x) = x^5-8*x^3+2 szélsőértékei')
```

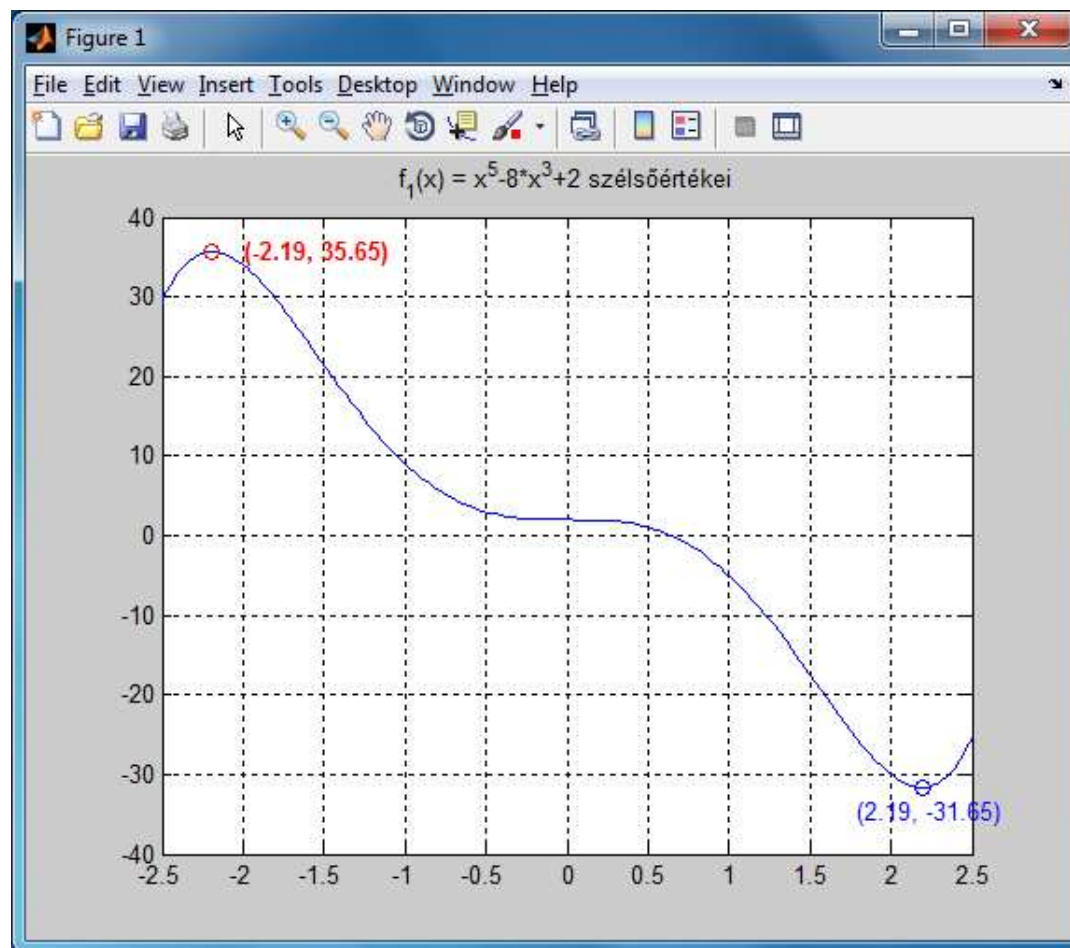




Függvényvizsgálat

Szélsőértékek (folyt.)

- A kész ábra
 - Figyeljük meg a felrakott elemeket!



Függvényvizsgálat

További Matlab támogatás a függvényvizsgálathoz

- Határértékek
 - Ábráról, ill. numerikusan (teszt a pont környékén)
 - Szimbolikus *limit* parancs (lásd szimbolikus óra)
- Monotonitás
 - Lehet: a szélsőértékek ismeretében, a fv. rajz alapján
 - Szimbolikus deriválással (*diff*)
 - Elemezhetjük a numerikus differenciavektort is (lásd később)
- Konvexitás, inflexiós pontok
 - Szimbolikusan, a második deriv. vizsgálatával
- Értékkészlet
 - Függvényábráról
 - *min*, *max* paranccsal a függvény kif. alapján (eml. optimalizációs fel.ok)
- Példa
 - ```
>> syms x, limit(1/x, x, 0) % nem létezik
```
  - ```
>> limit(1/x, x, 0, 'right'), limit(1/x, x, 0, 'left')
```
 - ```
>> ezplot('1/x', [-6 6 -8 8]) % ellenőrzés
```
  - ```
>> hold on, fplot('0', 'k--'),
```
 - ```
fplot('1/x', 'k--', [-100000*eps 100000*eps])
```

