

Alkalmazott matematika

Baran Ágnes

Octave/Matlab alapok 2.

Egy egyszerű ábra

Példa

Rajzoltassuk ki a $(-1, 2)$, $(0, 1)$, $(1, 1.5)$, $(2, 3)$ pontokat a síkon!

1. lépés: Soroljuk fel egy változóban a pontok első koordinátáit!

```
>> x=[-1, 0, 1, 2];
```

2. lépés: Soroljuk fel egy másik változóban a pontok második koordinátáit!

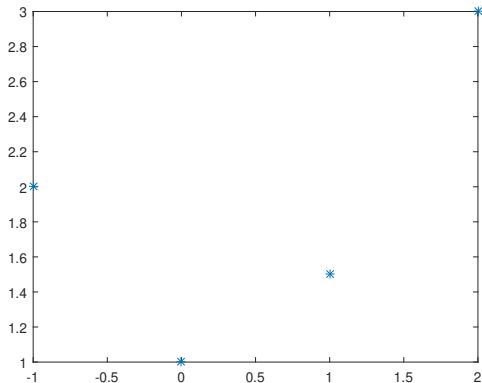
```
>> y=[2, 1, 1.5, 3];
```

(Figyeljünk rá, hogy “tizedesvessző” helyett “tizedespont” szerepel)

3. lépés: A plot függvény segítségével rajzoltassuk ki a pontokat!

```
>> plot(x,y,'*')
```

```
% kirajzolunk 4 pontot  
x=[-1, 0, 1, 2];  
y=[2, 1, 1.5, 3];  
plot(x,y,'*')
```



Az elkészült programunkat könnyen módosíthatjuk. Pl.

```
% kirajzolunk 4 pontot  
figure  
x=[-1, 0, 1, 2];  
y=[2, 1, 1.5, 3];  
plot(x,y,'*')  
axis([-1.5 2.5 0.5 3.5])
```

A `figure` utasítás hatására egy új grafikus ablak nyílik. Ennek hiányában, ha van megnyitott grafikus ablak, akkor abba készíti el az ábrát, annak korábbi tartalmát felülírva.

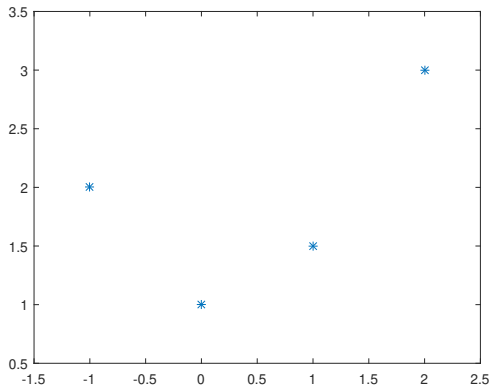
Az `axis` beállítja a tengelyek határait.

A `plot` függvényről (ill. hasonlóan bármely más beépített függvényről) a parancsablakba a

```
>> help plot
```

utasítást gépelve tudhatunk meg többet.

```
% kirajzolunk 4 pontot  
figure  
x=[-1, 0, 1, 2];  
y=[2, 1, 1.5, 3];  
plot(x,y,'*')  
axis([-1.5 2.5 0.5 3.5])
```



A plot függvény

- `plot(x,y)`
ábrázolja azokat a síkbeli pontokat, melyeknek első koordinátája az x , második az y változóban szerepel, és összeköti őket.
- `plot(x,y,'szin típus')`
ábrázolja a pontokat, a megadott típusú markerrel, illetve vonaltípussal, a megadott színnel.

Vonaltípusok

- - folyamatos vonal
(alapértelmezés)
- : pontozott vonal
- - - szaggatott vonal
- -. szaggatott-pontozott vonal

A plot függvény

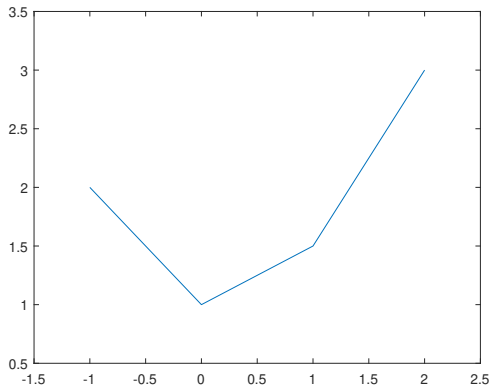
Markerek

- * csillag
- o kör
- + összeadás jel
- x kereszt
- s négyzet
- d rombusz
- p ötszög
- h hatszög
- < balra mutató háromszög
- > jobbra mutató háromszög
- ^ felfele mutató háromszög
- v lefele mutató háromszög

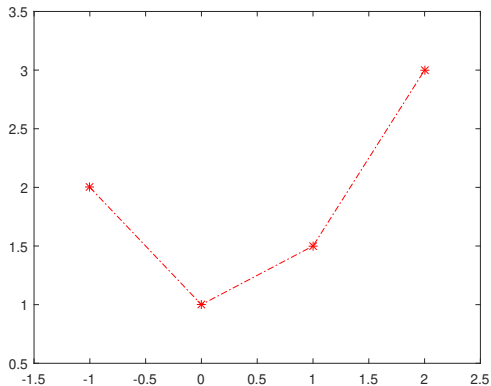
Színek

- b kék
- r piros
- g zöld
- k fekete
- w fehér
- y sárga
- m magenta
- c cián

```
% kirajzolunk 4 pontot  
figure  
x=[-1, 0, 1, 2];  
y=[2, 1, 1.5, 3];  
plot(x,y)  
axis([-1.5 2.5 0.5 3.5])
```




```
% kirajzolunk 4 pontot  
figure  
x=[-1, 0, 1, 2];  
y=[2, 1, 1.5, 3];  
plot(x,y,'-.r*')  
axis([-1.5 2.5 0.5 3.5])
```



Függvények ábrázolása

Példa

Rajzoltassuk ki az $f(x) = \sin(x)$ függvényt a $[0, 2\pi]$ intervallumon!

Függvényeket úgy ábrázolhatunk, hogy a függvénygörbe nagyon sok pontját kirajzoltatjuk.

Vegyünk a $[0, 2\pi]$ intervallumon sok pontot, pl:

```
>> x=linspace(0,2*pi,50);
```

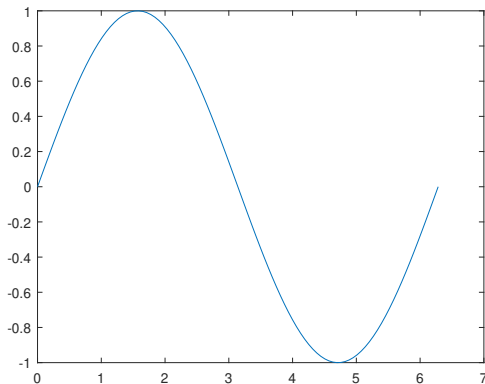
vagy

```
>> x=linspace(0,2*pi);
```

Az első esetben 50, a másodikban 100 egyforma lépésközű pontot kapunk a $[0, 2\pi]$ intervallumon.

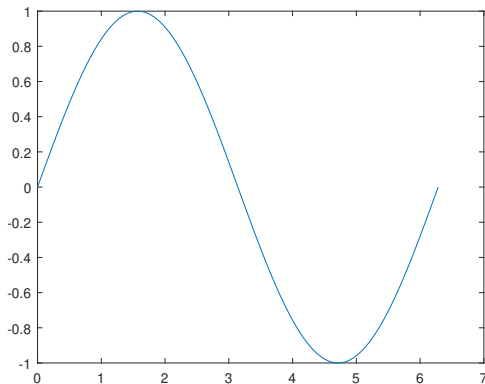
Minden pontban számítsuk ki a függvény értékét és rajzoltassuk ki a pontokat!

```
x=linspace(0,2*pi);  
y=sin(x);  
figure; plot(x,y)
```



Az fplot függvény

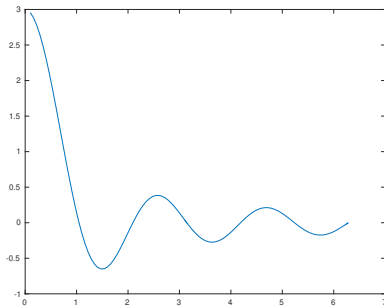
```
figure;  
fplot('sin',[0,2*pi])
```



Példa

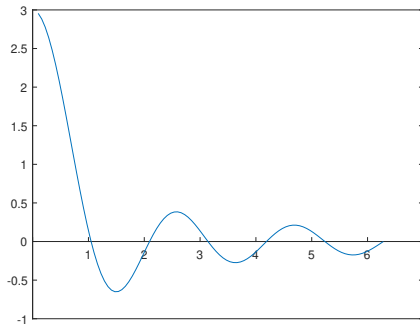
Rajzoltassuk ki az $f(x) = \frac{\sin(3x)}{x}$ függvényt a $[0.1, 2\pi]$ intervallumon!

```
x=linspace(0.1,2*pi);  
y=sin(3*x)./x;  
figure; plot(x,y)
```

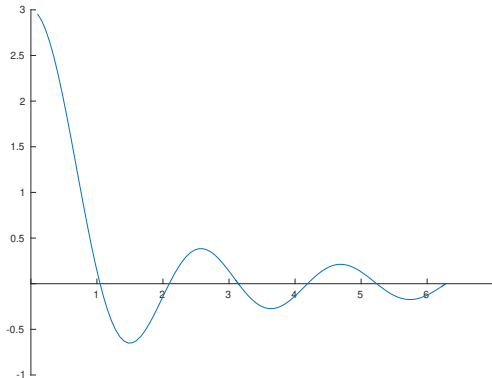


A tengelyek pozícionálása

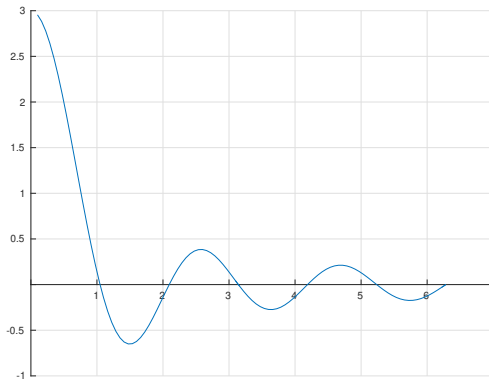
```
graphics_toolkit gnuplot %csak Octave-ban  
x=linspace(0.1,2*pi);  
y=sin(3*x)./x;  
figure; plot(x,y)  
set(gca,'xaxislocation','origin')
```



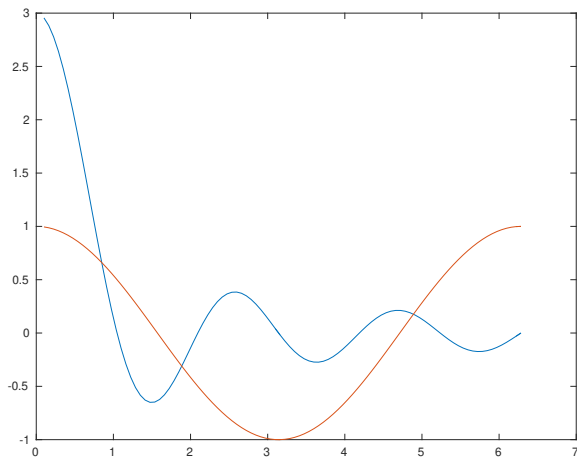
```
graphics_toolkit gnuplot %csak Octave-ban  
x=linspace(0.1,2*pi);  
y=sin(3*x)./x;  
figure; plot(x,y)  
set(gca,'xaxislocation','origin')  
box off
```



```
graphics_toolkit gnuplot %csak Octave-ban  
x=linspace(0.1,2*pi);  
y=sin(3*x)./x;  
figure; plot(x,y)  
set(gca,'xaxislocation','origin')  
box off; grid on
```



Több függvény egy ábrán



Több függvény egy ábrán

```
x=linspace(0.1,2*pi);  
y=sin(3*x)./x;  
z=cos(x);  
figure; plot(x,y,x,z)
```

vagy

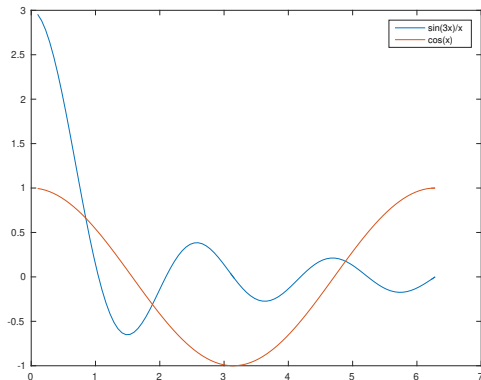
```
x=linspace(0.1,2*pi);  
y=sin(3*x)./x;  
z=cos(x);  
figure; plot(x,y)  
hold on;  
plot(x,z)  
hold off;
```

- hold on

bekapcsolja a „rárajzoló” üzemmódot: az aktuális figure-ablakba rajzol, az ottani eredeti ábra meghagyásával

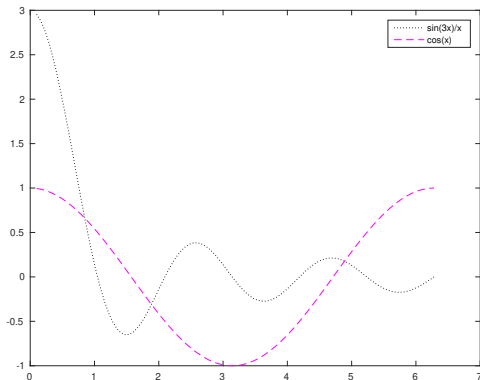
Több függvény egy ábrán, legend box

```
x=linspace(0.1,2*pi);  
y=sin(3*x)./x;  
z=cos(x);  
figure; plot(x,y,x,z)  
legend('sin(3x)/x','cos(x)')
```

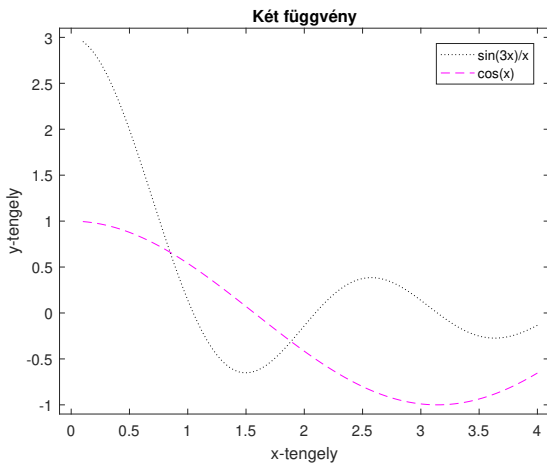


Vonaltípus és szín megadása

```
x=linspace(0.1,2*pi);  
y=sin(3*x)./x;  
z=cos(x);  
figure; plot(x,y,'k:',x,z,'m--')  
legend('sin(3x)/x','cos(x)')
```



Cím, tengelyek, legendbox



Cím, tengelyek, legendbox

```
x=linspace(0.1,2*pi);  
y=sin(3*x)./x;  
z=cos(x);  
figure; plot(x,y,'k:',x,z,'m--')  
axis([-0.1 4.1 -1.1 3.1]);  
xlabel('x-tengely')  
ylabel('y-tengely');  
title('Két függvény');  
legend('sin(3x)/x','cos(x)');
```

- `title('az abra cime')`
az ábra címe
- `xlabel('szöveg')` illetve `ylabel('szöveg')`
a tengelyek feliratozása

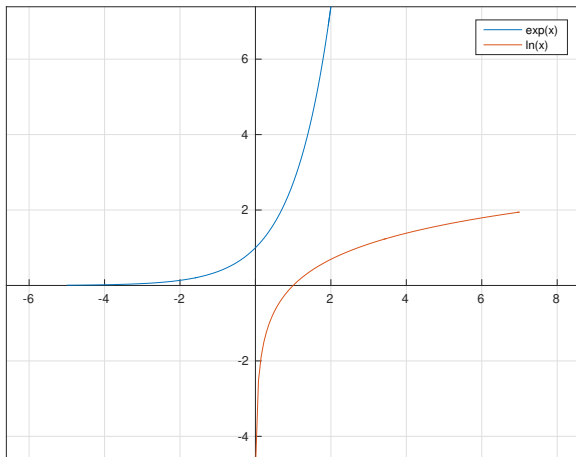
Néhány hasznos utasítás

- `axis tight`
a tengelyek határait úgy állítja be, hogy az ábra kitöltse a dobozt
- `axis equal`
minden tengelyen ugyanazt az egységet használja
- `axis square`
egyforma hosszú tengelyeket használ
- `axis off`
nem jeleníti meg a tengelyeket
- `box off`
nem jelenik meg a doboz határolóvonala
- `grid on`
berácsozza az ábrát
- `close all`
bezár minden látható figure ablakot

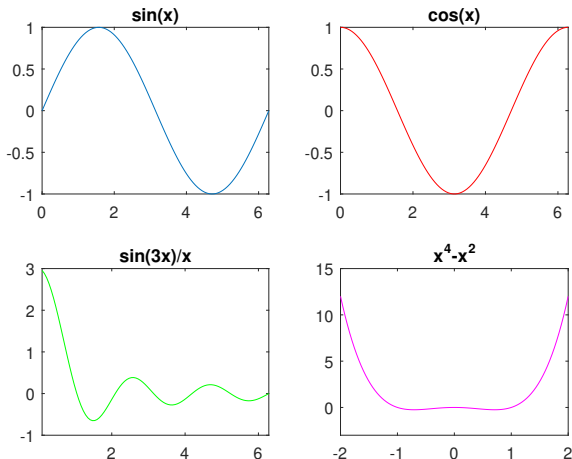
Példa

Ábrázoljuk az $f(x) = e^x$ és $g(x) = \ln(x)$ függvényeket egy ábrán!

```
graphics_toolkit gnuplot %csak Octave-ban
x1=linspace(-5,2); y1=exp(x1);
x2=linspace(0.01,7); y2=log(x2);
figure; plot(x1,y1,x2,y2);
set(gca,'xaxislocation','origin')
set(gca,'yaxislocation','origin')
axis equal; grid on;
legend('exp(x)', 'ln(x)')
```

Részábrák

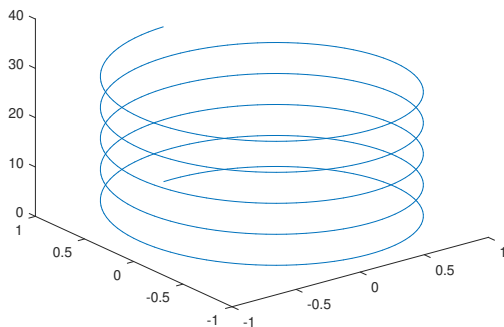


Az $m \times n$ részábrát tartalmazó ábra k -adik részábrájára vonatkozó utasítások a `subplot(m,n,k)` után következnek:

```
figure
subplot(2,2,1)
x=linspace(0,2*pi);
plot(x,sin(x),'k')
axis([0,2*pi,-1,1]);
title('sin(x)')
subplot(2,2,2)
plot(x,cos(x),'r')
axis([0,2*pi,-1,1]);
title('cos(x)')
subplot(2,2,3)
x=linspace(0.1,2*pi);
plot(x,sin(3*x)./x,'g')
title('sin(3x)/x')
subplot(2,2,4)
x=linspace(-2,2);
plot(x,x.^4-x.^2,'m')
title('x^4-x^2')
```

Vonalak 3 dimenzióban

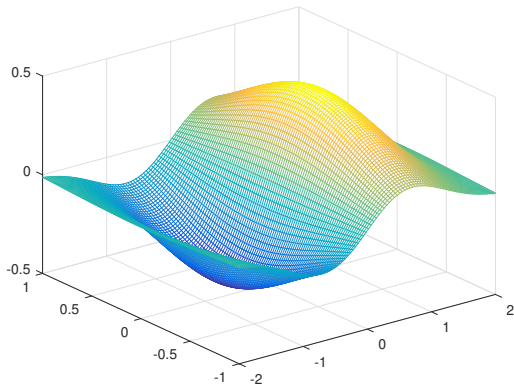
```
t = 0:pi/50:10*pi;  
st = sin(t);  
ct = cos(t);  
figure; plot3(st,ct,t)
```



Felületek (fmesh függvénnel)

Példa

Ábrázoljuk az $f(x, y) = xe^{-x^2-y^2}$ függvényt a $[-2, 2] \times [-1, 1]$ tartomány felett!



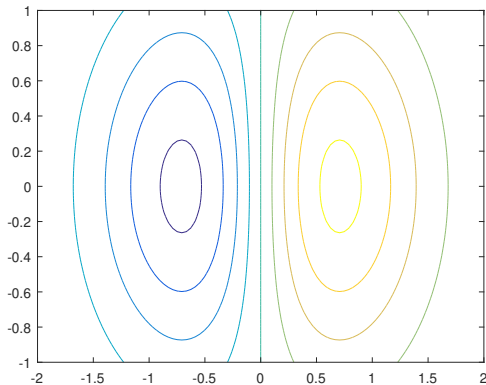
Felületek (fmesh függvénnyel)

- Definiáljuk a függvényt egy function handle típusú változóként.
- Hívjuk meg az `fmesh` függvényt, első argumentumában az ábrázolni kívánt függvénnyel, második argumentumában (opcionális) a változókra vonatkozó határokkal. Ha a második argumentum hiányzik, akkor a tartomány a $[-5, 5] \times [-5, 5]$.

```
f = @(x,y) x.*exp(-x.^2-y.^2);  
figure; fmesh(f,[-2,2,-1,1])
```

Szintvonalak (fcontour függvénnel)

```
f = @(x,y) x.*exp(-x.^2-y.^2);  
figure; fcontour(f,[-2,2,-1,1])
```



Felületek (mesh függvénnyel)

A **mesh** használatához előbb “be kell rácsoznunk” a sík egy tartományát, pl.:

```
>> x=0:15; y=0:10;  
>> [X,Y]=meshgrid(x,y);
```

Ekkor X és Y is 11×16 -os mátrix:

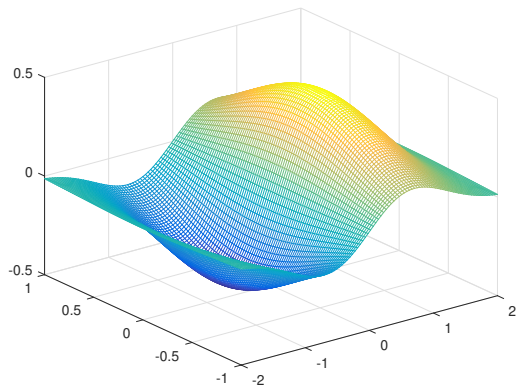
$$X = \begin{bmatrix} 0 & 1 & \dots & 14 & 15 \\ 0 & 1 & \dots & 14 & 15 \\ \vdots & & & & \\ 0 & 1 & \dots & 14 & 15 \end{bmatrix} \quad Y = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ 1 & 1 & \dots & 1 & 1 \\ \vdots & & & & \\ 10 & 10 & \dots & 10 & 10 \end{bmatrix}$$

(Az X és Y mátrixokat „egymásra helyezve” megkapjuk az összes lehetséges (x_i, y_j) párt)

Ezután kiszámoljuk az (X_i, Y_i) pontokban a függvény értékét és ábrázoljuk (pl a **mesh** vagy **surf** függvénnyel)

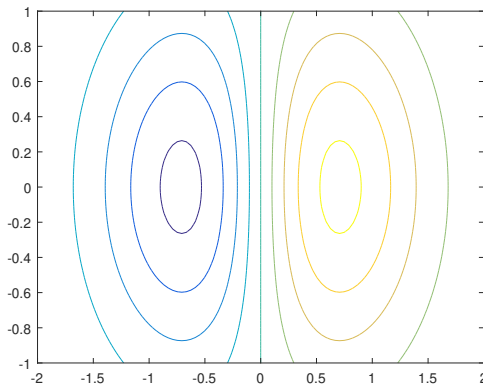
Felületek (mesh függvénynel)

```
x=linspace(-2,2);  
y=linspace(-1,1);  
[xx,yy] = meshgrid(x,y);  
zz = xx.*exp(-xx.^2-yy.^2);  
figure; mesh(xx,yy,zz)
```



Szintvonalak (contour függvényrel)

```
x=linspace(-2,2);  
y=linspace(-1,1);  
[xx,yy] = meshgrid(x,y);  
zz = xx.*exp(-xx.^2-yy.^2);  
figure; contour(xx,yy,zz)
```



Felületek és szintvonalak

```
x=linspace(-2,2);  
y=linspace(-1,1);  
[xx,yy] = meshgrid(x,y);  
zz = xx.*exp(-xx.^2-yy.^2);  
figure; meshc(xx,yy,zz)
```

