

Numerikus matematika

Gazdaságinformatikus BSc

Baran Ágnes

Labor

A tananyag elkészítését az EFOP-3.4.3-16-2016-00021 számú projekt támogatta. A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósult meg.

Tartalom

- 1 Lebegőpontos számok (Octave/Matlab bevezető)
- 2 Octave/Matlab alapok 1.
- 3 Normák, kondíciós számok
- 4 Lineáris egyenletrendszerek
- 5 Octave/Matlab alapok 2.
- 6 Legkisebb négyzetes közelítések
- 7 Interpoláció
- 8 Numerikus integrálás
- 9 Nemlineáris egyenletek
- 10 Optimalizálás

Lebegőpontos számok

Lebegőpontos számok

1. feladat.

Legyen $a = 2$, $t = 4$, $k_- = -5$, $k_+ = 5$.

(a) Írja fel a következő számok lebegőpontos alakját.

30, 26, 0.875, 0.5625, 1.625, 2.75

(b) Hány pozitív normalizált lebegőpontos szám ábrázolható ilyen jellemzők mellett?

(c) Mi lesz a legnagyobb ábrázolható szám és a legkisebb pozitív normalizált lebegőpontos szám?

(d) Mi lesz az alábbi számokhoz rendelt lebegőpontos szám szabályos kerekítés, illetve levágás esetén?

0.1, 0.4, 0.3, $\frac{1}{3}$, 0.7, $\frac{1}{32}$

2. feladat

- (a) Ábrázolja számegyenesen az $a = 2$, $t = 4$, $k_- = -3$, $k_+ = 2$ jellemzők mellett felírható összes pozitív normalizált lebegőpontos számot.
- (b) A fenti számábrázolási jellemzők mellett mennyi lesz M_∞ , ε_0 és ε_1 értéke?
- (c) Ezen jellemzők mellett mondjon példát olyan $x, y > 0$ lebegőpontos számokra, melyekre $x + y < M_\infty$ teljesül, de $x + y$ nem lebegőpontos szám.
- (d) Mondjon példát olyan $x, y > 0$, $x \neq y$ lebegőpontos számokra, melyekre $fl(x - y) = 0$ teljesül.
- (e) Mondjon példát olyan $x, y > 0$ lebegőpontos számokra, melyekre $fl(x + y) = x$ teljesül.

Octave/Matlab bevezető

Részletes dokumentáció:

- Octave:
`https://www.gnu.org/software/octave/`
- Matlab:
`https://www.mathworks.com/help/matlab/`

A parancsablakba utasításokat gépelhetünk, pl:

```
>> 3+4
```

```
ans =
```

```
7
```

```
>> 3*1.5
```

```
ans =
```

```
4.5000
```

```
>> cos(0)
```

```
ans =
```

```
1
```

Ha másképp nem rendelkezünk, akkor az eredmény az `ans` nevű változóba kerül.

Használhatunk más változókat is, pl.:

```
>> a=3+4
```

```
a =
```

```
7
```

```
>> a=3; b=4; c=a+b
```

```
c =
```

```
7
```

Ha egy értékadó utasítást pontosvesszővel zárunk le, akkor az értékadás végrehajtódik, de az eredmény nem jelenik meg a parancsablakban. Pl.:

```
>> a=3; b=4; c=a+b;
```

A változó értékét ekkor is megkérdezhetjük, nevének begépelésével:

```
>> c
```

```
c =
```

```
7
```

Változónevek

- Betűvel vagy aláhúzással (Octave), illetve betűvel (Matlab) kell kezdődniük, tartalmazhatnak betűket, számokat, aláhúzást.
Megkülönbözteti a kis- és nagybetűket. Ne használjunk ékezetes betűket!
- Nem lehetnek változónevek az Octave/Matlab kulcsszavai (pl. `if`, `end`, `stb`), az `iskeyword` utasítással felsoroltathatjuk ezeket a kulcsszavakat.
- Figyeljünk rá, hogy ne használjuk változónévként függvények neveit (pl. `cos`, `size`, `stb`). Ha nem vagyunk biztosak benne, hogy egy név létezik-e már, akkor az `exist` függvénnyel ellenőrizhetjük (pl. `exist cos`)
- A `clear` utasítással törölhetünk változókat (pl. `clear a,b` törli az `a` és `b` változókat). A `clear all` utasítással minden változó törlődik.

Összehasonlító operátorok

Értékük 1 (igaz), vagy 0 (hamis).

- $a < b$ Igaz, ha a kisebb, mint b
- $a \leq b$ Igaz, ha a nem nagyobb, mint b
- $a > b$ Igaz, ha a nagyobb, mint b
- $a \geq b$ Igaz, ha a nem kisebb, mint b
- $a == b$ Igaz, ha a egyenlő b -vel
- $a \sim b$ Igaz, ha a nem egyenlő b -vel
- $a != b$ (Csak Octave-ban!) Igaz, ha a nem egyenlő b -vel

Ha a és b azonos méretű mátrixok, akkor az összehasonlítás elemenként történik (és a visszaadott érték egy logikai tömb)

m-fájlok

Az Octave/Matlab futtatható állományai az m-fájlok.

- Nyissunk meg a szerkesztőablakban egy új fájlt.
- Írjuk ide a programunkat

A megjegyzéseinket

- ▶ Octave-ban % vagy # jel mögött,
- ▶ Matlab-ban % jel mögött

helyezhetjük el.

Több sorból álló blokkot a %{ és %} jelek közé helyezve tehetünk megjegyzésbe.

Itt is figyeljünk a sorvégi pontosvesszőkre, ha egy értékadó utasítás végén lemarad, akkor annak eredménye futás közben megjelenik a parancsablakban.

- Mentsük el a fájlt.
- Futtassuk a programunkat.

for-ciklus

```
for ciklusvaltozo=vektor  
    utasitasok  
end
```

Példák

```
s=0;  
for i=1:100  
    s=s+i;  
end
```

```
a=[4 2 -1 5];  
s=0;  
for i=a  
    s=s+1/i;  
end
```

```
s=100;  
for i=98:-2:2  
    s=s+i;  
end
```

while-ciklus

```
while logikai kifejezés  
    utasítások  
end
```

Példa

```
k=10;  
F=1;  
while k>1  
    F=F*k;  
    k=k-1;  
end
```

3. feladat

- (a) Vizsgálja meg számítógépén a $0.4 - 0.5 + 0.1 == 0$ logikai kifejezés értékét! Magyarázza meg a tapasztalt jelenséget! Mi lesz a $0.1 - 0.5 + 0.4 == 0$ logikai kifejezés értéke?
- (b) Mit tapasztal, ha az alábbi kódokat lefuttatja?

```
a=0;  
for i=1:5  
    a=a+0.2;  
end  
a==1
```

```
a=1;  
for i=1:5  
    a=a-0.2;  
end  
a==0
```

4. feladat

- (a) Írjon egy kódot a gépi epszilon meghatározására.
- (b) Olvassa el az Octave/Matlab eps függvényének help-jét. Nézze meg az eps (azaz az eps(1)) értékét.

5. feladat

Vizsgálja meg számítógépén a $2^{66} + 1 == 2^{66}$, $2^{66} + 10 == 2^{66}$, $2^{66} + 100 == 2^{66}$, $2^{66} + 1000 == 2^{66}$ és $2^{66} + 10000 == 2^{66}$ logikai kifejezések értékét! Keresse meg azt a legkisebb $n > 0$ számot, melyre a $2^{66} + n == 2^{66}$ logikai kifejezés értéke hamis. Mennyi az eps(2^66) értéke?

6. feladat

- (a) Az alábbi algoritmus végrehajtása után mennyi az x elméleti, illetve a gépi számítás után adódó értéke?

```
x=1/3;  
for i=1:40  
    x=4*x-1;  
end
```

- (b) Az alábbi algoritmus elméletileg minden $x \geq 0$ esetén az x eredeti értékét adja vissza. Vizsgálja meg mi történik a gyakorlatban, ha az algoritmust $x = 1000$, $x = 100$ kezdőértékkel futtatja! Mi az oka a tapasztalt jelenségnek?

```
for i=1:60  
    x=sqrt(x);  
end  
for i=1:60  
    x=x^2;  
end
```

7. feladat

Legyen $x = (10^{200}, 1)$. Számítsa ki gépén az x vektor normáját az alábbi két (matematikailag ekvivalens) módon. Magyarázza meg a tapasztalt jelenséget!

(a)

$$\|x\| = \sqrt{x_1^2 + x_2^2}$$

(b)

$$c = \max\{|x_1|, |x_2|\}, \quad \|x\| = c \cdot \sqrt{\left(\frac{x_1}{c}\right)^2 + \left(\frac{x_2}{c}\right)^2}$$

8. feladat

Írassa ki a `realmax` és `realmin` értékét. Vizsgálja meg a `realmin('single')` és `realmax('single')` értékeket is.

Octave/Matlab alapok 1.

Vektorok Octave/Matlab-ban

Megkülönbözteti a sor- és oszlopvektorokat

Sorvektorok

Az $a = (-1.2, 3.1, 4.7, 1.9)$ vektor megadása elemei felsorolásával:

- az elemeket vesszővel választjuk el:

`a = [-1.2, 3.1, 4.7, 1.9]`

- vagy az elemeket szóközzel választjuk el:

`a = [-1.2 3.1 4.7 1.9]`

A vektorkoordináták számozása 1-gyel kezdődik, $a(i)$ az a vektor i -edik koordinátája.

`length(a)` az a vektor koordinátáinak száma

`a = []` üres vektor

Vektorok, mint szabályos sorozatok

A kettőspont operátorral

- a $b = (1, 2, 3, 4, 5)$ vektor:

$$b = 1:5$$

- a $c = (5, 4, 3, 2, 1)$ vektor:

$$c = 5:-1:1$$

- a $d = (2, 2.2, 2.4, 2.6, 2.8, 3)$ vektor

$$d=2:0.2:3$$

Általában:

$$x=elsoelem:lepeskoz:utolsoelem$$

ahol a lépésköz negatív is lehet, vagy

$$x=elsoelem:utolsoelem$$

akkor a lépésköz 1.

Vektorok, mint szabályos sorozatok

A linspace függvénnyel:

- az $e = (1, 1.2, 1.4, 1.6, 1.8, 2)$ vektor
`e=linspace(1,2,6)`
- egy 100 koordinátából álló f vektor
`f=linspace(1,2)`

Általában:

`x=linspace(elsoelem,utolsoelem,elemekszama)`

ahol a koordináták egyforma lépésközzel követik egymást, vagy

`x=linspace(elsoelem,utolsoelem)`

akkor a koordináták száma 100.

Oszlopvektorok megadása

- elemeinek felsorolásával (a vektor koordinátáit pontosvesszővel választjuk el)

`m=[-3;0;7]`

- egy sorvektor transzponálásával: `n=[1 -2 4 -1]'`
(**valójában a ' jel konjugált transzponáltat eredményez, a konjugálás nélküli transzponálás: `a.'` vagy `transpose(a)`**)

`x(i)` és `length(x)` az `x` vektor i -edik koordinátája és az `x` vektor koordinátáinak száma (ugyanúgy mint a sorvektoroknál)

`size(x)` az `x` vektor mérete (sorvektoroknál az `[1 length(x)]` vektor, oszlopvektoroknál a `[length(x) 1]` vektor)

Vektorok konstruálása más vektorokból

- $[a \ b]$ két sorvektor egymás után fűzése
- $[m;n]$ két oszlopvektor egymás után fűzése
- $[-4 \ a \ 3 \ -1]$ sorvektor bővítése újabb elemekkel
- $[1;m;-3]$ oszlopvektor bővítése újabb elemekkel
- $h(2:4)$ a h vektor 2., 3. és 4. koordinátájából álló vektor
- $h([1 \ 4 \ 5])$ a h vektor 1., 4. és 5. koordinátájából álló vektor
- $h(2)=[]$ elhagyja a h vektor 2. koordinátáját
- $h([2 \ 4])=[]$ elhagyja a h vektor 2. és 4. koordinátáját

Fontos! Ha $a=[-1 \ 3 \ 2]$ akkor az $a(6)=4$ utasítás eredménye az $a=[-1 \ 3 \ 2 \ 0 \ 0 \ 4]$ vektor (a legkisebb olyan vektor, amelyben van értelme a $a(6)=4$ utasításnak, a nemdefiniált elemeket 0-kal tölti fel. **Megváltozik a vektor mérete, erre nem figyelmeztet!**)

Néhány hasznos függvény

- `min(x)` és `max(x)` az x vektor legkisebb és legnagyobb eleme
- `sort(x)` az x elemeit növekvő sorrendbe rendezi
- `sort(x, 'descend')` az x elemeit csökkenő sorrendbe rendezi
- `flip(x)` az x elemeit fordított sorrendben sorolja fel
- `length(x)` az x vektor elemeinek a száma
- `sum(x)` az x vektor elemeinek összege
- `prod(x)` az x vektor elemeinek szorzata
- `mean(x)` az x vektor elemeinek átlaga
- `x(3)` az x vektor harmadik eleme
- `x(1:3)` az x vektor első három eleme
- `x(3:end)` az x vektor minden elemei a harmadiktól az utolsóig

Műveletek vektorokkal

Ha a és b két ugyanolyan méretű vektor, akkor

- $a+b$ ill. $a-b$ a két vektor összege és különbsége
- $x=a+1$ egy ugyanolyan méretű vektor, mint a , $x_i = a_i + 1$
- $x=a.^2$ egy ugyanolyan méretű vektor, mint a , $x_i = a_i^2$.
- $x=a.*b$ egy ugyanolyan méretű vektor, mint a és b , $x_i = a_i b_i$
- $x=a./b$ egy ugyanolyan méretű vektor, mint a és b , $x_i = \frac{a_i}{b_i}$
- $x=1./a$ egy ugyanolyan méretű vektor, mint a , $x_i = \frac{1}{a_i}$
- $\text{dot}(a,b)$ az a és b skaláris szorzata

Fontos! A műveleti jel előtti pont a művelet elemenkénti végrehajtását eredményezi

$\sin$, $\cos$, $\tan$, $\exp$, $\log$, sqrt , abs , stb. mind elemenként hajtódik végre.

NaN : Not a Number (pl. $0/0$, Inf/Inf)

1. feladat

(a) Az elemek egyenkénti begépelése nélkül állítsa elő az alábbi vektorokat!

(1) $a = (0, 1, \dots, 30)$

(2) $b = (2, 4, 6, \dots, 100),$

(3) $c = (2, 1.9, 1.8, \dots, 0.1, 0)$

(4) $d = (0, 3, 6, \dots, 27, 30, -100, 30, 27, \dots, 6, 3, 0)$

(5) $e = (\frac{1}{2}, \frac{1}{3}, \dots, \frac{1}{20})$

(6) $f = (\frac{1}{2}, \frac{2}{3}, \dots, \frac{19}{20})$

(b) Legyen x egy adott 100 elemű sorvektor. Az x vektorból állítsa elő azt az y vektort, melynek elemei

(1) az x vektor elemei fordított sorrendben felsorolva,

(2) az x vektor első 5 eleme,

(3) az x vektor elemei ugyanolyan sorrendben, kivéve az x 4. elemét

(4) az x vektor elemei ugyanolyan sorrendben, kivéve az x 5., 72. és 93. elemét

(5) az x vektor páratlan sorszámú elemei

(6) az x vektor 2., 5., 17. és 81. eleme.

2. feladat

Legyen x egy adott sorvektor. A `for` utasítás használata nélkül az x vektorból állítsa elő azt az y vektort, melynek i -edik eleme

(1) $x(i) + 2$

(2) $x(i)^2$

(3) $1/x(i)$

(4) $\sin(x(i)^3 - 1)$

(5) $x(i) - i$

Mátrix megadása elemenként

$A = [1, 2, 3; 4, 5, 6; 7, 8, 9]$ vagy $A = [1 \ 2 \ 3; \ 4 \ 5 \ 6; \ 7 \ 8 \ 9]$
eredménye:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

(Az egy sorban álló elemeket vesszővel vagy szóközzel, a sorokat pontosvesszővel választjuk el.)

A mátrixelemek számozása $(1, 1)$ -gyel kezdődik.

$A(i, j)$ a mátrix (i, j) -edik eleme.

Mátrixok megadása

Vektorok összefűzésével

Ha $a=[1 \ -2 \ 0]$; $b=[2 \ -11 \ 7]$; $m=[-3;0;7]$; $n=[1; \ -2; \ 0]$; akkor $B=[a;b]$ eredménye:

$$B = \begin{pmatrix} 1 & -2 & 0 \\ 2 & -11 & 7 \end{pmatrix}$$

$C=[a' \ \ b']$ és $D=[m \ \ n]$ eredménye:

$$C = \begin{pmatrix} 1 & 2 \\ -2 & -11 \\ 0 & 7 \end{pmatrix} \quad D = \begin{pmatrix} -3 & 1 \\ 0 & -2 \\ 7 & 0 \end{pmatrix}$$

Mátrixok bővítése

Az előbb definiált mátrixokkal, vektorokkal:

$E=[A; a]$ vagy $E=[A; [1, -2, 0]]$ eredménye

$$E = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 1 & -2 & 0 \end{pmatrix}$$

Tehát: *[mátrix „sortörés” (azaz ;) sorvektor]*

Az $F=[A \ m]$ vagy $F=[A, \ m]$ eredménye

$$F = \begin{pmatrix} 1 & 2 & 3 & -3 \\ 4 & 5 & 6 & 0 \\ 7 & 8 & 9 & 7 \end{pmatrix}$$

Tehát: *[mátrix szóköz vagy vessző oszlopvektor]*

Mátrixok bővítése

$G=[C \ D]$ és $H=[C;D]$ eredménye

$$G = \begin{pmatrix} 1 & 2 & -3 & 1 \\ -2 & -11 & 0 & -2 \\ 0 & 7 & 7 & 0 \end{pmatrix} \quad H = \begin{pmatrix} 1 & 2 \\ -2 & -11 \\ 0 & 7 \\ -3 & 1 \\ 0 & -2 \\ 7 & 0 \end{pmatrix}$$

$C(4,5)=9$ eredménye:

$$C = \begin{pmatrix} 1 & 2 & 0 & 0 & 0 \\ -2 & -11 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9 \end{pmatrix}$$

Megváltozik a mátrix mérete, erre nem figyelmeztet!

Hivatkozás elemekre, sorokra, oszlopokra, részmátrixokra

- `size(A)` az A mátrix mérete (egy kételemű sorvektor)
- `length(A)` egy skalár: `max(size(A))`
- `A(i,j)` az A mátrix (i,j) -edik eleme
- `A(i,:)` egy sorvektor, az A mátrix i -edik sora
- `A(:,j)` egy oszlopvektor, az A mátrix j -edik oszlopa
- `A(2:3,:)` az A mátrix 2. és 3. sora
- `A([1 2 4],:)` az A mátrix 1., 2. és 4. sora
- `A(:, [1 3])` az A mátrix 1. és 3. oszlopa
- `A(2:3, [1 3])` az A mátrix 2. és 3. sorának és 1. és 3. oszlopának metszetéből álló mátrix

Mátrixok „átszabása”

Sorok, oszlopok elhagyása mátrixokból

- $A(i, :) = []$ az i -edik sor elhagyása
- $A(:, j) = []$ a j -edik oszlop elhagyása
- $A([1 \ 3], :) = []$ az 1. és 3. sor elhagyása
- $A(:, [1 \ 3]) = []$ az 1. és 3. oszlop elhagyása

Sor- és oszlopcsere

Az i -edik és j -edik sor illetve oszlop cseréje:

$$A([i, j], :) = A([j, i], :), \text{ ill. } A(:, [i, j]) = A(:, [j, i])$$

Mátrixból vektor

$A(:)$ az A mátrix elemei oszlopfolytonosan felsorolva

Néhány beépített mátrix

<code>eye(n)</code>	az $n \times n$ -es egységmátrix
<code>eye(n,m)</code>	az $n \times m$ -es egységmátrix
<code>ones(n)</code>	a csupa 1-esből álló $n \times n$ -es mátrix
<code>ones(n,m)</code>	a csupa 1-esből álló $n \times m$ -es mátrix
<code>zeros(n)</code>	a csupa 0-ból álló $n \times n$ -es mátrix
<code>zeros(n,m)</code>	a csupa 0-ból álló $n \times m$ -es mátrix

Néhány hasznos függvény

- `numel(A)` az A elemeinek száma
- `size(A)` az A mérete
- `length(A)` egyenlő `max(size(A))` értékével

Műveletek vektorok és mátrixok között

Legyen A és B két mátrix (melyek akár vektorok is lehetnek), c egy skálár.
Az

$$A+B, \quad A-B, \quad c \cdot A, \quad A \cdot B, \quad A^2$$

műveletek a hagyományos, lineáris algebrában értelmezett műveletek, feltéve, hogy A és B mérete megfelelő. Az

$$A + c$$

művelet eredménye: az A minden eleméhez hozzáadunk c -t. Az

$$A/B \quad \text{és} \quad A \setminus B$$

műveletek eredménye $A \cdot B^{-1}$ és $A^{-1} \cdot B$.

Műveletek vektorok és mátrixok között

Elemenkénti művelet

A műveleti jel előtti `.` jel a művelet elemenkénti végrehajtását eredményezi:

Az $A.*B$ mátrix ij -edik eleme $a_{ij} * b_{ij}$,

az $A.^2$ mátrix ij -edik eleme a_{ij}^2 ,

az $A./B$ mátrix ij -edik eleme a_{ij}/b_{ij} .

A beépített függvények általában hívhatók mátrix argumentummal is, pl. $\sin(A)$, $\log(A)$, $\exp(A)$, $\text{abs}(A)$, stb. Ilyenkor a függvény a mátrix minden elemére végrehajtódik.

3. feladat

Legyen $x = [-1 \ 4 \ 0]$, $y = [3 \ -2 \ 5]$
és $A = [-3 \ 1 \ -4; 6 \ 2 \ -5]$. Döntse el, hogy az alábbi utasítások közül melyik végrehajtható. Ha nem végrehajtható, akkor magyarázza meg miért, ha végrehajtható, akkor fogalmazza meg mi lesz az eredmény!

(1) $z = [x, y]$

(2) $z = [x; y]$

(3) $z = [x', y']$

(4) $z = [x'; y']$

(5) $z = [A; x]$

(6) $z = [A, x]$

(7) $z = [x; A; y]$

(8) $z = [A'; x]$

(9) $z = [A', x]$

(10) $z = [A', x']$

(11) $x + y$

(12) $x + y'$

(13) $A + y$

(14) $A + 2$

(15) $x ./ y$

(16) $A \wedge 2$

(17) $A. \wedge 2$

4. feladat

Legyen

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{pmatrix}$$

Konstruálja meg (az elemek felsorolása nélkül) azt a B mátrixot, melyet úgy kapunk, hogy

- (1) elhagyjuk az A mátrix első sorát,
- (2) elhagyjuk az A mátrix 2. és 4. oszlopát,
- (3) elhagyjuk az A mátrix utolsó sorát és oszlopát
- (4) kétszer egymás mellé írjuk az A mátrixot,
- (5) transzponáljuk az A mátrixot,
- (6) felcseréljük az A mátrix 2. és 4. oszlopát
- (7) négyzetre emeljük az A elemeit

- (8) az A minden elemét megnöveljük 3-mal
- (9) A minden elemének vesszük a négyzetgyökét
- (10) A minden elemének vesszük a szinuszát
- (11) az A első sorának második elemét kicseréljük -2 -re
- (12) az A 2. sorát kicseréljük a $[-1 \quad 0 \quad -2 \quad 3]$ vektorra

5. feladat

- Egy rövid utasítás segítségével állítsa elő az alábbi mátrixot!

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 20 & 18 & 16 & 14 & 12 & 10 & 8 & 6 \\ 2 & 4 & 8 & 16 & 32 & 64 & 128 & 256 \end{pmatrix}$$

- Az előző feladat A mátrixával vizsgálja meg az alábbi utasítások eredményét!

(1) `sum(A)`

(2) `sum(A,2)`

(3) `reshape(A,6,4)`

(4) `max(A)`

(5) `max(A, [], 2)`

(6) `max(A,2)`

(7) `flipud(A)`

(8) `fliplr(A)`

(9) `size(A)`

(10) `length(A)`

Néhány, lineáris algebrában hasznos függvény

- $\det(A)$ az A determinánsa
- $\text{inv}(A)$ az A inverze
- $\text{dot}(a,b)$ az a és b vektorok belsőszorzata
- $\text{norm}(A)$ az A 2-normája (mátrixok és vektorok esetén is)
- $\text{norm}(A, \infty)$ az A ∞ -normája (mátrixok és vektorok esetén is)
- $\text{norm}(A, 1)$ az A 1-normája (mátrixok és vektorok esetén is)

Az $Ax = b$ lineáris egyenletrendszer megoldása (részletesen ld. később):

$$x=A \backslash b$$

Néhány hasznos függvény

diag

- $\text{diag}(a)$
ahol a egy vektor, egy négyzetes mátrixszal tér vissza, főátlójában az a vektorral
- $\text{diag}(a,k)$
ahol a egy vektor, k egy egész, egy olyan mátrixszal tér vissza, aminek a k -adik átlója az a vektor. A 0. átló a főátló, onnan felfelé egyesével nő, lefelé egyesével csökken az átlók sorszáma.
- $\text{diag}(A)$
ahol A egy mátrix (nem feltétlenül négyzetes) egy oszlopvektorral tér vissza, az A főátlóbeli elemeivel
- $\text{diag}(A,k)$
ahol A egy mátrix, k egy egész, egy oszlopvektorral tér vissza, az A mátrix k -adik átlójának elemeivel.

Néhány hasznos függvény

tril és triu

- `tril(A)`
Az A mátrix alsóháromszög részével tér vissza (a főátló és az alatta álló elemek, a többi 0)
- `triu(A)`
Az A mátrix felsőháromszög részével tér vissza (a főátló és a felette álló elemek, a többi 0)
- `tril(A,k)`
Az A mátrix k -adik átlójában és az alatta álló elemekkel tér vissza (a többi 0)
- `triu(A,k)`
Az A mátrix k -adik átlójában és a felette álló elemekkel tér vissza (a többi 0)

Függvények írása

Az Octave/Matlab függvények szerkezete:

```
function kimenovaltozok=fvneve(bemenovaltozok)
    utasitasok
end
```

Fontos! A fenti függvényt fvneve.m néven kell elmenteni.

Példák.

```
function y=masodf(x)
    y=2*x.^2-3*x+5;
end
```

Ekkor a $y=masodf(x)$ utasítás eredménye a $2x^2 - 3x + 5$ kifejezés értéke, ahol x akár vektor is lehet, ebben az esetben a függvény elemenként hajtódik végre és y is vektor (ezt az teszi lehetővé, hogy a fv-ben minden művelet végrehajtható vektorokra is, mivel a négyzetreemelés jele elé tettük a $.$ jelet)

Normák, kondíciós számok

1. feladat

Írjon 1-1 Octave/Matlab függvényt az 1 -, 2 -, ∞ -vektornormák számítására.

2. feladat

Írjon 1-1 Octave/Matlab függvényt az 1 - és ∞ -mátrixnormák számítására.

3. feladat

Olvassa el a `norm` és `cond` függvények help-jét.

4. feladat

Oldja meg Octave/Matlab-bal az $Ax = b$ lineáris egyenletrendszert, ha

$$A = \begin{pmatrix} 1 & 0.99 \\ 0.99 & 0.98 \end{pmatrix}, \quad b = \begin{pmatrix} 1.99 \\ 1.97 \end{pmatrix}.$$

Tegyük fel, hogy b helyett

$$b + \delta b = \begin{pmatrix} 1.98 \\ 1.98 \end{pmatrix}$$

ismert. Oldja meg az $Ay = b + \delta b$ egyenletrendszert is. Számítsa ki a megoldásvektor, illetve a jobboldali vektor relatív hibáját ∞ -normában. Határozza meg $\text{cond}_{\infty}(A)$ értékét! (Használja a beépített `cond` függvényt.)

5. feladat

Számítsa ki a 6×6 -os Hilbert-mátrix kondíciószámát! (Használja a `cond` és `hilb` beépített függvényeket!) Legyen B egy 6×6 -os véletlen mátrix (használja a `rand` függvényt), számítsa ki B kondíciószámát is (végezzen több kísérletet)!

6. feladat

Írjon egy függvényt, mely adott n pozitív egész szám esetén előállítja azt az $A \in \mathbb{R}^{n \times n}$ mátrixot, melyre

$$a_{ij} = \begin{cases} 1, & \text{ha } i = j, \\ -1, & \text{ha } i < j, \\ 0, & \text{egyébként,} \end{cases}$$

teljesül.

7. feladat

Állítsa elő a következő $A \in \mathbb{R}^{100 \times 100}$ mátrixot illetve $b \in \mathbb{R}^{100}$ vektort, és a backslash operátort használva oldja meg az $Ax = b$ egyenletrendszert. Ezután perturbálja a b vektort, pl. 1 helyett legyen $b(100) = 1.00001$ és oldja meg a rendszert újra. Számítsa ki az A kondíciós számát.

$$a_{ij} = \begin{cases} 1, & \text{ha } i = j, \\ -1, & \text{ha } i < j, \\ 0, & \text{egyébként,} \end{cases} \quad b = (-98, -97, \dots, 0, 1)^T.$$

Megjegyzés

Nagy mátrixok esetén a `cond` függvény helyett használjuk a `condest` függvényt, amely az 1-normában vett kondíciós szám becslését adja (anélkül, hogy kiszámítaná A^{-1} -et.)

Lineáris egyenletrendszerek, mátrixok felbontása

1. feladat

Határozza meg az alábbi mátrixok inverzét Gauss-Jordan eliminációval!

(a)

$$A = \begin{pmatrix} 2 & 2 & 3 \\ 1 & -1 & 0 \\ -1 & 2 & 1 \end{pmatrix},$$

(c)

$$C = \begin{pmatrix} -2 & 1 & 4 \\ 1 & 0 & 3 \\ -1 & 2 & -3 \end{pmatrix},$$

(b)

$$B = \begin{pmatrix} 3 & -1 & 2 \\ 1 & -3 & -4 \\ 2 & 2 & 5 \end{pmatrix},$$

(d)

$$D = \begin{pmatrix} -1 & 2 & 1 \\ 2 & -3 & 1 \\ 1 & 1 & 2 \end{pmatrix}.$$

2. feladat

Oldja meg LU-felbontással az $Ax = b$ egyenletrendszert! Határozza meg az A mátrix determinánsát!

(a)

$$A = \begin{pmatrix} -2 & 2 & 1 \\ 6 & -3 & -4 \\ -4 & 1 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ -8 \\ 4 \end{pmatrix},$$

(b)

$$A = \begin{pmatrix} -2 & 1 & 4 & -2 \\ 2 & -4 & -1 & 1 \\ -4 & 8 & 6 & -3 \\ -6 & 3 & 8 & -3 \end{pmatrix}, \quad b = \begin{pmatrix} 5 \\ 1 \\ 3 \\ 16 \end{pmatrix},$$

(c)

$$A = \begin{pmatrix} 2 & -4 & 2 \\ -4 & 6 & -1 \\ 1 & 0 & -2 \end{pmatrix}, \quad b = \begin{pmatrix} -2 \\ 5 \\ 0 \end{pmatrix},$$

(d)

$$A = \begin{pmatrix} -4 & -4 & 2 \\ -2 & -7 & 3 \\ 2 & 12 & -5 \end{pmatrix}, \quad b = \begin{pmatrix} -2 \\ 6 \\ -13 \end{pmatrix},$$

(e)

$$A = \begin{pmatrix} -1 & -2 & 0 & -3 \\ 1 & -1 & 2 & 2 \\ 2 & -2 & 6 & 5 \\ 0 & -6 & -2 & -2 \end{pmatrix}, \quad b = \begin{pmatrix} -2 \\ 5 \\ 14 \\ 0 \end{pmatrix},$$

3. feladat

Oldja meg az $Ax = b$ és $Ax = c$ lineáris egyenletrendszereket LU-felbontással!

$$A = \begin{pmatrix} -3 & 1 & 0 \\ 12 & -6 & 2 \\ -6 & 4 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 5 \\ -28 \\ 12 \end{pmatrix}, \quad c = \begin{pmatrix} -6 \\ 18 \\ -15 \end{pmatrix}.$$

4. feladat

Határozza meg az alábbi mátrixok Cholesky-felbontását! Számítsa ki a mátrixok determinánsát!

$$A = \begin{pmatrix} 4 & -2 & -4 \\ -2 & 10 & 5 \\ -4 & 5 & 9 \end{pmatrix}$$

$$B = \begin{pmatrix} 4 & -4 & -4 & 0 \\ -4 & 13 & 7 & 6 \\ -4 & 7 & 6 & -1 \\ 0 & 6 & -1 & 17 \end{pmatrix}$$

$$C = \begin{pmatrix} 9 & -3 & -3 \\ -3 & 5 & 5 \\ -3 & 5 & 9 \end{pmatrix}$$

5. feladat

Oldja meg az $Ax = b$ lineáris egyenletrendszert Cholesky-felbontással!

(a)

$$A = \begin{pmatrix} 9 & -6 & -6 & 3 \\ -6 & 8 & 6 & -6 \\ -6 & 6 & 9 & -10 \\ 3 & -6 & -10 & 18 \end{pmatrix}, \quad b = \begin{pmatrix} 36 \\ -38 \\ -47 \\ 58 \end{pmatrix}$$

(b)

$$A = \begin{pmatrix} 9 & -6 & 3 \\ -6 & 20 & -14 \\ 3 & -14 & 14 \end{pmatrix}, \quad b = \begin{pmatrix} -21 \\ 34 \\ -34 \end{pmatrix}$$

Lineáris algebra Octave/Matlab-bal

Példa

Oldjuk meg az $Ax = b$ lineáris egyenletrendszert, ha

$$A = \begin{pmatrix} -2 & -1 & 4 \\ 2 & 3 & -1 \\ -4 & -10 & -5 \end{pmatrix}, \quad b = \begin{pmatrix} 3 \\ 1 \\ -12 \end{pmatrix}$$

Megoldás. Használjuk a backslash operátort!

```
>>A=[-2 -1 4; 2 3 -1; -4 -10 -5];
```

```
>>b=[3; 1; -12];
```

```
>>x=A\b
```

```
x=
```

```
3
```

```
-1
```

```
2
```

Ügyeljünk rá, hogy a b oszlopvektorként legyen megadva!

Ha az egyenletrendszer kibővített mátrixával meghívjuk az `rref` függvényt:

```
>>rref([A b])
```

```
ans=
```

```
1 0 0 3
```

```
0 1 0 -1
```

```
0 0 1 2
```

akkor láthatjuk, hogy a Gauss-Jordan elimináció eredményeként valóban így állítható elő a b vektor az A oszlopvektoraiból, amelyek lineárisan függetlenek, tehát a megoldás egyértelmű.

Példa

Oldjuk meg az $Ax = b$ lineáris egyenletrendszert, ha

$$A = \begin{pmatrix} -4 & -4 & 2 \\ -2 & -7 & 3 \\ 2 & 12 & -5 \end{pmatrix}, \quad b = \begin{pmatrix} -2 \\ 6 \\ -13 \end{pmatrix}$$

Megoldás. Próbálkozzunk ismét a backslash operátorral!

```
>>A=[-4 -4 2; -2 -7 3; 2 12 -5];
```

```
>>b=[-2; 6; -13];
```

```
>>x=A\b
```

```
warning: matrix singular to machine precision
```

```
x =
```

```
1.93162
```

```
-1.27350
```

```
0.31624
```

Az Octave arra figyelmeztetett, hogy a mátrix szinguláris (valóban, $\det(A) = 0$), de ellenőrizhetjük, hogy $Ax=b$ kerekítési hiba nagyságrendű, azaz x -et tekinthetjük megoldásnak.

Próbálkozzunk az `rref` függvénnyel!

```
>>rref([A b])
```

```
ans=
```

```
1.0000      0 -0.1000  1.9000
      0 1.0000 -0.4000 -1.4000
      0      0      0      0
```

Azt látjuk, hogy a mátrix oszlopvektorai lineárisan függőek, de a b vektor benne van az oszlopvektorok által felfeszített térben. Tudjuk, hogy ilyenkor az egyenletrendszernek végtelen sok megoldása van, ezek közül egy:

$$x = \begin{pmatrix} 1.9 \\ -1.4 \\ 0 \end{pmatrix}$$

Ha az egyenletrendszer összes megoldását szeretnénk tudni, akkor használjuk a `null` függvényt, amely előállítja a nulltér egy bázisát:

```
>>p=null(A)
p=
-0.092450
-0.369800
-0.924500
```

Ezek szerint a lineáris egyenletrendszer általános megoldása:

$$\begin{pmatrix} 1.9 \\ -1.4 \\ 0 \end{pmatrix} + \lambda p$$

ahol $\lambda \in \mathbb{R}$. (A kapott x megoldás a $\lambda = -0.34207$ konstanshoz tartozik.)

Példa

Oldjuk meg az $Ax = b$ lineáris egyenletrendszert, ahol

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{pmatrix}, \quad b = \begin{pmatrix} 4 \\ 6 \\ 9 \\ 12 \end{pmatrix}$$

Megoldás. A backslash operátorral azt kapjuk, hogy

```
>>x=A\b
```

```
x=
```

```
1.0000
```

```
2.7000
```

Könnyen látható, hogy ez **nem megoldása** az egyenletrendszernek.

Az `rref` függvénnyel:

```
>>rref([A b])  
ans=  
    1  0  0  
    0  1  0  
    0  0  1  
    0  0  0
```

láthatjuk, hogy az alaplátrix rangja 2, a kibővített mátrixé 3, az egyenletrendszer **ellentmondásos**.

Ellentmondásos lineáris egyenletrendszerek esetén a backslash operátor egy olyan x vektort ad vissza, melyre az Ax és b vektorok eltérése euklideszi normában a legkisebb (azaz $\|Ax - b\|_2$ minimális). Ilyenkor azt mondjuk, hogy x az egyenletrendszer legkisebb négyzetes értelemben vett megoldása.

6. feladat

Oldja meg Octave/Matlab-bal az $Ax = b$ lineáris egyenletrendszert, ahol

(a)

$$A = \begin{pmatrix} 2 & -3 & 1 & 1 \\ -1 & 3 & 4 & 7 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 5 \end{pmatrix}$$

(b)

$$A = \begin{pmatrix} 2 & 1 \\ -3 & 4 \\ 5 & -1 \end{pmatrix}, \quad b = \begin{pmatrix} -5 \\ 24 \\ -23 \end{pmatrix}$$

(c)

$$A = \begin{pmatrix} 2 & 1 & 5 & 0 \\ -3 & 4 & -13 & 22 \\ 5 & -1 & 16 & -14 \\ 1 & 1 & 2 & 2 \end{pmatrix}, \quad b = \begin{pmatrix} 12 \\ 81 \\ -33 \\ 15 \end{pmatrix}$$

Hasznos: ha az x racionális elemű vektor koordinátáit nem tizedestört alakban akarjuk látni, akkor használhatjuk a `rats(x)` utasítást, vagy a kiíratás formátumát állítsuk át: `format rat`

Több jobboldali vektor

Példa

Oldjuk meg az $Ax = b$ és $Ax = c$ egyenletrendszereket, ha

$$A = \begin{pmatrix} -2 & -1 & 4 \\ 2 & 3 & -1 \\ -4 & -10 & -5 \end{pmatrix}, \quad b = \begin{pmatrix} 3 \\ 1 \\ -12 \end{pmatrix}, \quad c = \begin{pmatrix} 17 \\ 1 \\ -42 \end{pmatrix}$$

Megoldás. Mivel a két rendszer mátrixa azonos, ezért megoldhatjuk őket egyszerre.

```
>>A=[-2 -1 4; 2 3 -1; -4 -10 -5];
```

```
>>b=[3; 1; -12]; c=[17; 1; -42];
```

```
>>x=A\[b c]
```

```
x=
```

```
    3    -2  
   -1     3  
    2     4
```

Több jobboldali vektor

Nagyméretű mátrixok esetén a futási időt jelentősen befolyásolhatja, hogy az azonos mátrixszal adott rendszereket egyszerre, vagy külön-külön oldjuk meg:

```
>> A=rand(10000);  
>> b=ones(10000,1);  
>> c=zeros(10000,1);  
>> tic;x=A\[b,c];toc  
Elapsed time is 6.116513 seconds.  
>> tic;x=A\b; x2=A\c; toc  
Elapsed time is 11.571959 seconds.
```

(A fenti eredmény egy Intel Core i5-4590 processzorral, 7.7 GiB memóriával rendelkező gépen született).

LU-felbontás

$$[L,U,P]=lu(A)$$

Ekkor

- L : alsóháromszög mátrix, átlójában csupa 1-es
- U : felsőháromszög mátrix
- P : permutációs mátrix

úgy, hogy $PA = LU$.

$$[L1,U1]=lu(A)$$

Ekkor $A = L1 \cdot U1$ úgy, hogy $U1$ megegyezik az előző U mátrixszal és $L1 = P^T L$.

Cholesky-felbontás

- `chol(A)`
elkészíti az A mátrix Cholesky-felbontását, a felbontásban szereplő felsőháromszög mátrixszal tér vissza.
- `chol(A, 'lower')`
elkészíti az A mátrix Cholesky-felbontását, a felbontásban szereplő alsóháromszög mátrixszal tér vissza.

Ha az A nem pozitív definit, akkor nem létezik a felbontás, hibaüzenetet kapunk.

A felbontás létezéséhez a mátrixnak szimmetrikusnak kell lennie, ezt nem vizsgálja, de első esetben csak az A felső-, a másodikban az alsóháromszög részét használja.

Több jobboldali vektor

Ha több lineáris egyenletrendszert kell megoldanunk, ahol a mátrix azonos, a jobboldali vektorok különbözőek, de a jobboldali vektorok nem állnak egyszerre rendelkezésre, akkor a következő utasításokat használjuk:

Egyetlen egyszer, a rendszerek megoldása előtt készítsük el a mátrix LU-felbontását:

```
>> [L,U]=lu(A);
```

Ahányszor egy újabb b jobboldali vektor rendelkezésünkre áll, adjuk ki az

```
>> x=U\ (L\b);
```

utasítást, amivel megkapjuk az adott jobboldali vektor esetén a rendszer megoldását.

Mátrix inverze Octave/Matlab-bal

Az `inv` függvénnyel számítható. Ha a mátrix nem négyzetes, vagy a determinánsa 0 (vagy 0-hoz közeli), akkor hibaüzenetet, illetve figyelmeztetést kapunk.

**Nagyméretű mátrixok inverzének kiszámítása túl költséges lehet.
Csak akkor számoljuk ki, ha ténylegesen szükségünk van az inverzre.**

Pl. az $Ax = b$ négyzetes mátrixú lineáris egyenletrendszer megoldása $x = A^{-1}b$ módon kb háromszor annyi műveletbe kerül, mint az $x = A \backslash b$ megoldás.

7. feladat

Tekintsük a következő mikrogazdasági modellt: földművesek, állattenyésztők és bányászok egy-egy csoportja rendre gabonát, húst és szenet "állít elő". Az előállított termék egy részét minden csoport maga használja fel, egy részét a többiek veszik igénybe, egy részét pedig külső piacon értékesíti. Az alábbi táblázatban látható, hogy az egyes csoportoknak egy egységnyi áru előállításához hány egységre van szükségük a többi nyersanyagból, illetve hány egység a külső igény. Határozza meg az előállított termékek mennyiségét úgy, hogy minden igény ki legyen elégítve, és ne keletkezzen felesleg.

	földműves	állattenyésztő	bányász	külső igény
növény	0.1	0.7	0.1	2
hús	0.2	0.1	0.3	3
szén	0.4	0.3	0.1	5

8. feladat

Legyen $A = \text{pascal}(10)$ (azaz A a 10×10 -es Pascal mátrix, ami egy szimmetrikus, pozitív definit mátrix), $x = \text{ones}(10, 1)$ és definiálja a b vektort úgy, hogy $b = A * x$. Oldja meg az $Ax = b$ rendszert az `lu`, `chol` és `A\b` utasításokat alkalmazva (használjon „format long”-ot)!

Ritka mátrixok

A sparse függvény

```
>> A=[-1.1 0 0 2; 0 0 2 0; 0 -1 0 1;0 0 0 3]
```

A =

-1.1000	0	0	2.0000
0	0	2.0000	0
0	-1.0000	0	1.0000
0	0	0	3.0000

```
>> S=sparse(A)
```

S =

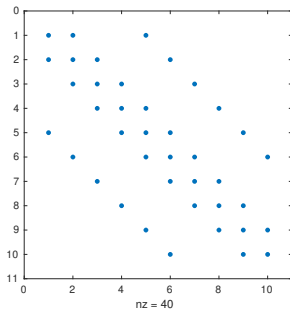
(1,1)	-1.1000
(3,2)	-1.0000
(2,3)	2.0000
(1,4)	2.0000
(3,4)	1.0000
(4,4)	3.0000

Definiáljunk egy olyan 10×10 -es ritka mátrixot, melyben csak 5 átlóban vannak 0-tól különböző elemek:

```
>> d=ones(10,1);  
>> S=spdiags([d d -4*d d d],[-4 -1 0 1 4],10,10);
```

Megnézhetjük a nemnulla elemek elhelyezkedését:

```
>> spy(S)
```



A nemnulla elemek száma: `nnz(S)`

Hasonlítsuk össze egy nagyméretű ritka mátrix esetén a tárigényt a különböző tárolási módok esetén:

```
>> d=ones(10000,1);  
>> S=spdiags([d d -4*d d d],[-4000 -1 0 1 4000],10000,10000);  
>> F=full(S);  
>> whos S F
```

Vizsgáljuk meg egy mátrix-vektor szorzás futási idejét:

```
>> x=rand(10000,1);  
>> tic;b=S*x;toc  
>> tic;b=F*x;toc
```

és egy lineáris egyenletrendszer megoldásának futási idejét

```
>> tic;y=F\b;toc  
>> tic;y=S\b;toc
```

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!

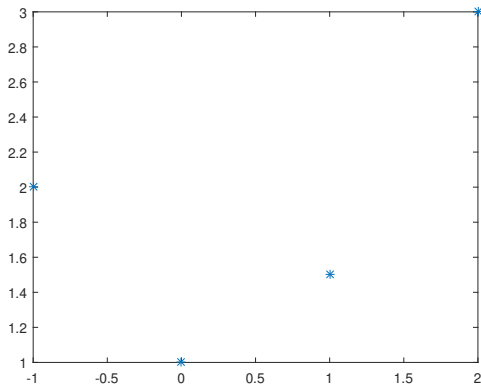
```
>> y=[2, 1, 1.5, 3];
```

(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.

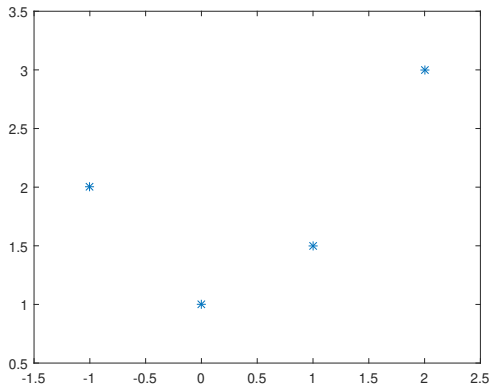
Az `axis` beállítja a tengelyek határait.

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

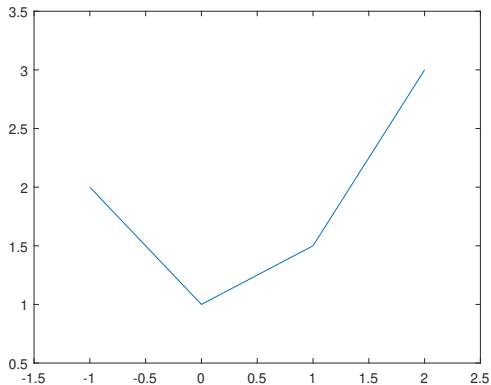
Markerek

- * 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

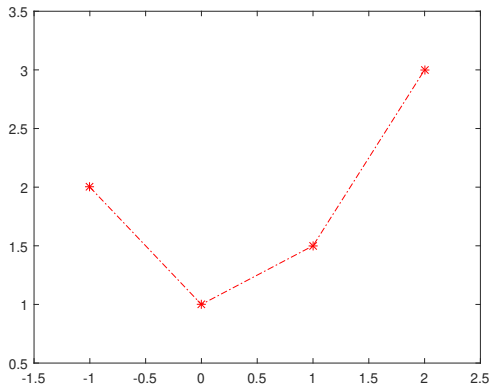
Színek

- 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);
```

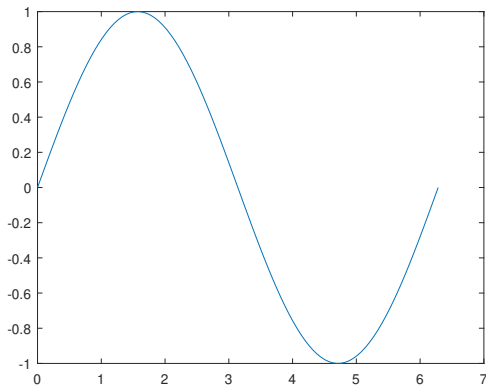
vagy

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

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

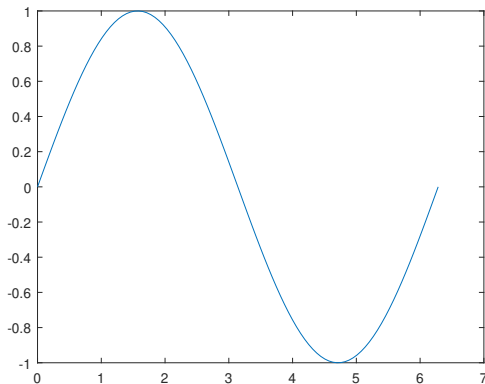
Minden pontban számítsuk ki a függvény értékét és rajzoltassuk ki a pontokat!

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



Az fplot függvény

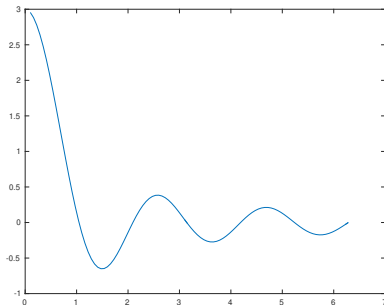
```
figure;  
fplot('sin',[0,2*pi])
```



Példa

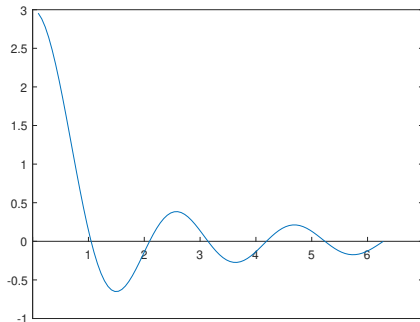
Rajzoltassuk ki az $f(x) = \frac{\sin(3x)}{x}$ függvényt a $[0.1, 2\pi]$ intervallumon!

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

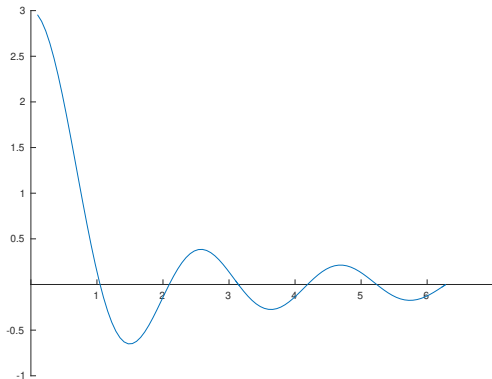


A tengelyek pozícionálása

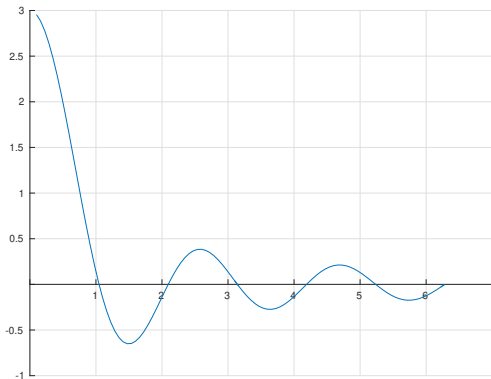
```
graphics_toolkit gnuplot %csak Octave-ban  
x=linspace(0.1,2*pi);  
y=sin(3*x)./x;  
figure; plot(x,y)  
set(gca,'xaxislocation','origin')
```



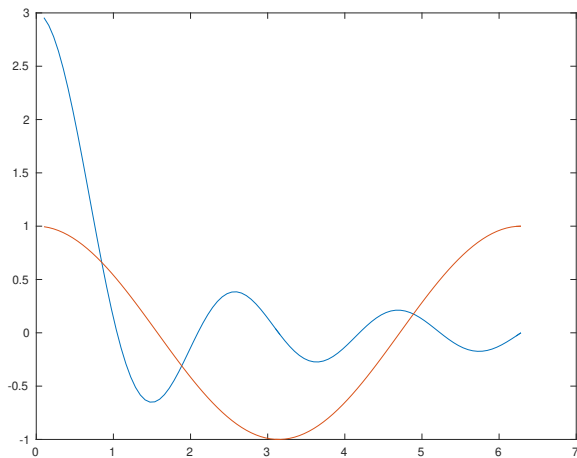
```
graphics_toolkit gnuplot %csak Octave-ban  
x=linspace(0.1,2*pi);  
y=sin(3*x)./x;  
figure; plot(x,y)  
set(gca,'axislocation','origin')  
box off
```



```
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)
```

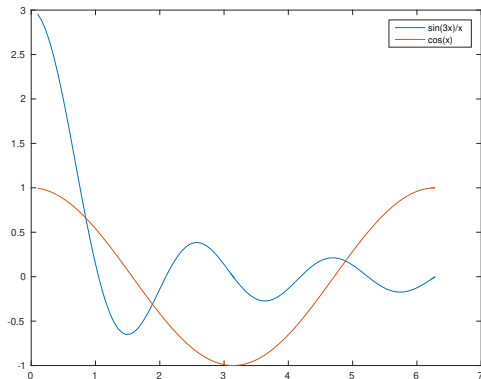
vagy

```
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

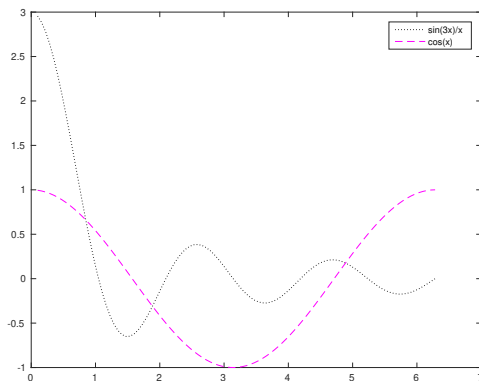
Több függvény egy ábrán, legend box

```
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)')
```

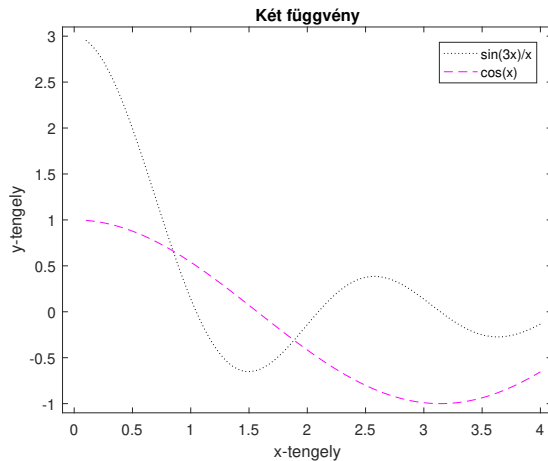


Vonaltípus és szín megadása

```
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

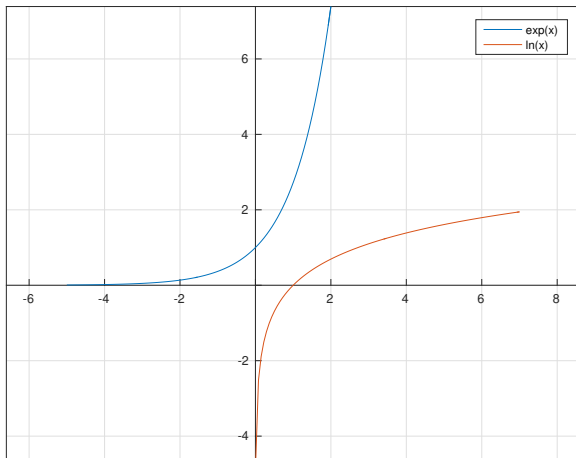
Néhány hasznos utasítás

- `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)')
```



Legkisebb négyzetes közelítések

1. feladat

Határozza meg az alábbi adatokra legkisebb négyzetes értelemben legjobban illeszkedő egyenes egyenletét.

$$(a) \begin{array}{c|ccccc} t_i & 0 & 1 & 2 & 3 & 4 \\ \hline f_i & 3 & \frac{5}{2} & \frac{1}{2} & 1 & 1 \end{array}$$

$$(b) \begin{array}{c|ccccc} t_i & 0 & 1 & 1 & 2 & 3 \\ \hline f_i & 1 & 2 & \frac{5}{2} & 3 & 5 \end{array}$$

$$(c) \begin{array}{c|cccc} t_i & 3 & 3 & 3 & 3 \\ \hline f_i & 2 & 1 & 2 & 2 \end{array}$$

Legkisebb négyzetes közelítések, polinom illesztése

Példa

Határozzuk meg az alábbi adatokat legkisebb négyzetes értelemben legjobban közelítő egyenest!

t_i	1	1.1	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2
f_i	8	8.9	9	9.8	10	11	11.5	11.5	12.5	13	13.7	14

Megoldás. Használjuk a polyfit függvényt!

```
p=polyfit(t,f,m)
```

megadja a (t_i, f_i) adatokra legkisebb négyzetes értelemben legjobban illeszkedő legfeljebb m -edfokú polinom együtthatóit a főegyütthatóval kezdve.

```
>> t=[1 1.1 1.1:0.1:2];  
>> f=[8 8.9 9 9.8 10 11 11.5 11.5 12.5 13 13.7 14];  
>> p=polyfit(t,f,1)  
p=  
5.8235 2.5338
```

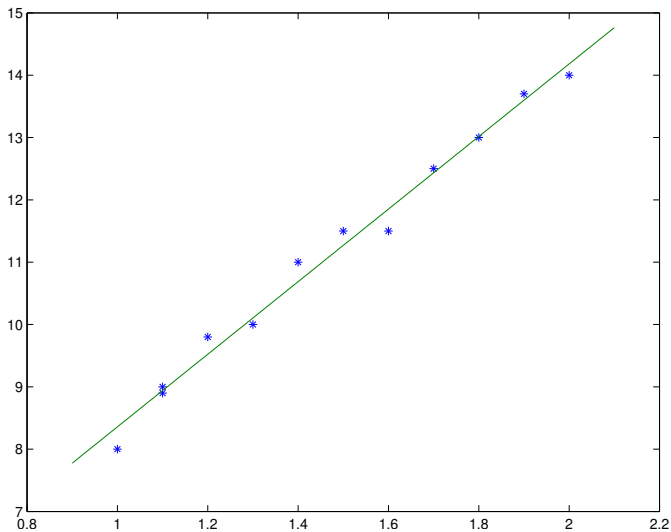
A keresett egyenes egyenlete:

$$f(t) = 5.8235t + 2.5338$$

Ha ábrázolni szeretnénk az adatokat és az illesztett egyenest:

```
>> xx=linspace(0.9,2.1);  
>> yy=polyval(p,xx);  
>> figure; plot(t,f,'*',xx,yy)
```

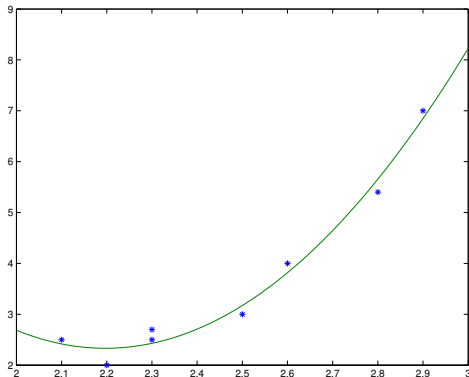
A polyval függvény a p együtthatójú polinom értékeit adja az xx-ben adott helyeken.



2. feladat

Határozzuk meg az alábbi adatokat legkisebb négyzetes értelemben legjobban közelítő másodfokú polinomot!

t_i	2.1	2.2	2.3	2.3	2.5	2.6	2.8	2.9
f_i	2.5	2	2.5	2.7	3	4	5.4	7



Legkisebb négyzetes közelítések

Példa

Határozzuk meg az alábbi adatokat legkisebb négyzetes értelemben legjobban közelítő

$$F(t) = x_1 + x_2 \cos(\pi t) + x_3 \sin(\pi t)$$

alakú modell paramétereit!

t_i	0.1	0.5	1.2	1.5	2	2.1	2.4	3	3.2
f_i	3.9	2.6	-0.8	0.3	3.2	3.8	3.2	-0.7	-0.9

Megoldás. A paramétereket az

$$A^T A x = A^T f$$

Gauss-féle normálegyenlet megoldása szolgáltatja.

$$A^T A x = A^T f$$

ahol

$$A = \begin{pmatrix} 1 & \cos(\pi t_1) & \sin(\pi t_1) \\ 1 & \cos(\pi t_2) & \sin(\pi t_2) \\ \vdots & & \\ 1 & \cos(\pi t_9) & \sin(\pi t_9) \end{pmatrix}, \quad f = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_9 \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

Állítsuk elő a megadott adatokból az A mátrixot:

```
>> t=[0.1 0.5 1.2 1.5 2 2.1 2.4 3 3.2]';
>> f=[3.9 2.6 -0.8 0.3 3.2 3.8 3.2 -0.7 -0.9]';
>> A=[ones(9,1), cos(pi*t), sin(pi*t)];
```

Ügyeljünk rá, hogy a t és az f oszlopvektor legyen!

Oldjuk meg a normálegyenletet!

```
>> x=(A'*A)\(A'*f)
```

```
x =
```

```
1.4372
```

```
2.0310
```

```
1.1711
```

A legjobban illeszkedő adott alakú modell tehát:

$$F(t) = 1.4372 + 2.0310 \cos(\pi t) + 1.1711 \sin(\pi t)$$

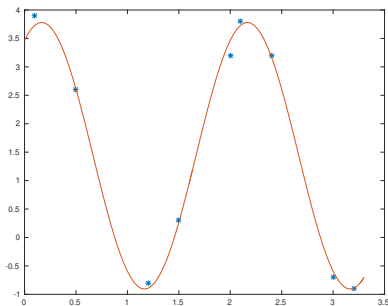
Ábrázoljuk az adatokat és az illesztett modellt!

```
>> xx=linspace(0,3.3);
```

```
>> yy=x(1)+x(2)*cos(pi*xx)+x(3)*sin(pi*xx);
```

```
>> figure; plot(t,f,'*',xx,yy)
```

```
t=[0.1 0.5 1.2 1.5 2 2.1 2.4 3 3.2]';  
f=[3.9 2.6 -0.8 0.3 3.2 3.8 3.2 -0.7 -0.9]';  
A=[ones(9,1), cos(pi*t), sin(pi*t)];  
x=(A'*A)\(A'*f);  
xx=linspace(0,3.3);  
yy=x(1)+x(2)*cos(pi*xx)+x(3)*sin(pi*xx);  
figure; plot(t,f,'*',xx,yy)
```



Feladatok

- (3) Határozza meg az alábbi adatokat négyzetesen legjobban közelítő egyenes egyenletét!

t_i	1	1.2	1.4	1.4	1.5	1.7	1.9	2
f_i	6.2	7	8	7.9	8.4	9.2	10	10.6

- (4) Határozza meg az alábbi adatokat négyzetesen legjobban közelítő harmadfokú polinomot!

t_i	0.5	0.8	1.1	1.3	1.5	1.7	1.9	2.1	2.3
f_i	2.5	2.3	1.8	1.3	0.9	0.4	0.1	-0.05	-0.01

- (5) Határozza meg az alábbi adatokat legjobban közelítő

$$F(t) = a + \frac{b}{t}$$

alakú modell paramétereit!

t_i	1	1.2	1.4	1.4	1.5	1.7	1.9	2	2.1	2.2
f_i	4.2	3.8	3.4	3.3	3.3	3	2.8	2.8	2.75	2.7

Feladatok

- (6) Határozza meg az alábbi adatokat négyzetesen legjobban közelítő

$$F(t) = x_1 \sin(t) + x_2 \sin(2t) + x_3 \sin(3t)$$

alakú modell paramétereit!

t_i	0.1	0.5	1.2	1.5	2	2.1	2.4	3	3.2	3.4	3.8	4	4.2	4.6	5
f_i	1	4.1	3	1	-1.5	-1.6	-1.7	-0.4	0.1	0.7	1.6	1.8	1.6	0.2	-2.5

- (7) Határozza meg az alábbi adatokat négyzetesen legjobban közelítő

$$F(t) = x_1 + x_2 \ln(t)$$

alakú modell paramétereit!

t_i	0.1	0.5	1.2	1.5	2	2.1	2.4	3	3.2
f_i	-0.6	1.5	2.5	2.9	3.2	3.3	3.5	3.8	3.9

8. feladat

Egy fél méter magas, téglatest alakú víztartályt egyenletes sebességgel töltenek fel vízzel. Amikor a tartályban 3 cm magasan áll a víz Péter elhatározza, hogy megméri a vízszint változását az idő függvényében. A következő méréseket végezte:

t_i (min)	0	2	4	6	8	10	12
f_i (cm)	3	4	5	5.5	6.5	7	8

Becsülje meg milyen magasan lesz a víz 20 perccel azután, hogy Péter elindította a mérést! Mikor indították el a tartály feltöltését? Kb mikor lesz tele a tartály?

9. feladat

Egy ipari mérlegen egy nagyobb mennyiségű gabona van, amit valaki egyenletes sebességgel lapátol a mérlegről zsákokba. Miután elkezdte a munkát, időnként megnézzük mennyit mutat a mérleg. Az alábbi értékeket láttuk:

idő (min)	1	15	20	28
tömeg (kg)	980	605	470	250

Becsüljük meg mennyi ideig tart, amíg az összes gabonát zsákokba rakja, illetve eredetileg mennyi gabona volt a mérlegen.

Interpoláció

Lagrange-interpoláció

Példa

Határozzuk meg a $(-2, -5)$, $(-1, 3)$, $(0, 1)$, $(2, 15)$ pontokra illeszkedő minimális fokszámú polinomot!

Megoldás. Készítsük el az osztott differenciák táblázatát!

Az első két oszlopba az alappontok és a megfelelő függvényértékek kerülnek:

-2	-5
-1	3
0	1
2	15

Számítsuk ki az elsőrendű osztott differenciákat!

-2	-5	$\frac{3-(-5)}{-1-(-2)} = 8$
-1	3	$\frac{1-3}{0-(-1)} = -2$
0	1	$\frac{15-1}{2-0} = 7$
2	15	

Számítsuk ki a másodrendű osztott differenciákat!

$$\begin{array}{r|l}
 -2 & -5 \\
 & 8 \\
 -1 & 3 \\
 & -2 \\
 0 & 1 \\
 & 7 \\
 2 & 15
 \end{array}
 \quad
 \begin{array}{l}
 \frac{-2-8}{0-(-2)} = -5 \\
 \frac{7-(-2)}{2-(-1)} = 3
 \end{array}$$

Számítsuk ki a harmadrendű osztott differenciát!

-2	-5			
		8		
-1	3		-5	
		-2		
0	1		3	
		7		
2	15			

$$\frac{3 - (-5)}{2 - (-2)} = 2$$

A táblázat felső élet használva írjuk fel a polinomot!

-2	-5			
		8		
-1	3		-5	
		-2		2
0	1		3	
		7		
2	15			

$$L_3(x) = -5 + 8(x+2) - 5(x+2)(x+1) + 2(x+2)(x+1)x$$

Megj.:

A Lagrange-polinom nem függ az adatok sorrendjétől, így választhattuk volna a táblázat alsó élét is:

-2	-5			
		8		
-1	3		-5	
		-2		2
0	1		3	
		7		
2	15			

$$L_3(x) = 15 + 7(x - 2) + 3(x - 2)x + 2(x - 2)x(x + 1)$$

Mindkét esetben

$$L_3(x) = 2x^3 + x^2 - 3x + 1$$

Lagrange-interpoláció

1. feladat

Írja fel az alábbi pontokra illeszkedő minimális fokszámú polinomot!

(a) $(-3, -6), (-2, -17), (-1, -8), (1, -2), (2, 19),$

(b) $(-3, -31), (-2, -8), (1, 1), (2, 24),$

(c) $(-2, -13), (-1, -4), (1, 2),$

(d) $(-2, -5), (-1, 3), (0, 1), (2, 15),$

(e) $(-1, 4), (1, 2), (2, 10), (3, 40),$

(f) $(-2, 38), (-1, 5), (1, -1), (2, -10), (3, -7),$

(g) $(-2, -33), (-1, -2), (1, 6), (2, 7), (3, -18).$

2. feladat

Horner-algoritmussal határozza meg a $p(-3)$ értéket, ha

$$p(x) = -x^5 + 3x^3 - 4x^2 - 3x + 5$$

Lagrange-interpoláció Octave/Matlab-bal

A polyfit függvény

`polyfit(x,f,n-1)` Ha x és f n -elemű vektorok, akkor megadja annak a legfeljebb $(n-1)$ -edfokú polinomnak az együtthatóit, amely illeszkedik az (x_i, f_i) , $i = 1, \dots, n$ adatokra.

Példa

Határozzuk meg a $(-2, -5)$, $(-1, 3)$, $(0, 1)$, $(2, 15)$ pontokra illeszkedő minimális fokszámú polinomot!

Megoldás.

```
>>x=[-2, -1, 0, 2];  
>>f=[-5, 3, 1, 15];  
>>p=polyfit(x,f,3)  
p=  
    2.0000    1.0000   -3.0000    1.0000
```

Ábrázoljuk a pontokat és az illesztett függvényt!

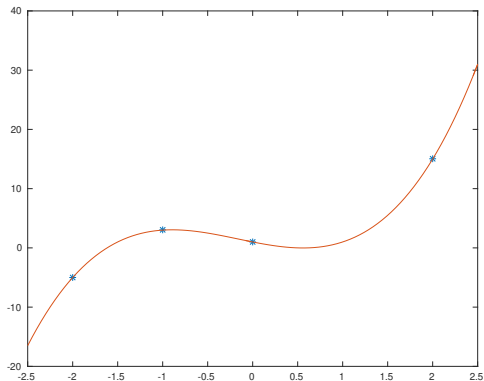
```
x=[-2, -1, 0, 2];  
f=[-5, 3, 1, 15];  
p=polyfit(x,f,3);  
xx=linspace(-2.5,2.5);  
yy=polyval(p,xx);  
figure; plot(x,f,'*',xx,yy)
```

A polyval függvény:

```
yy=polyval(p,xx);
```

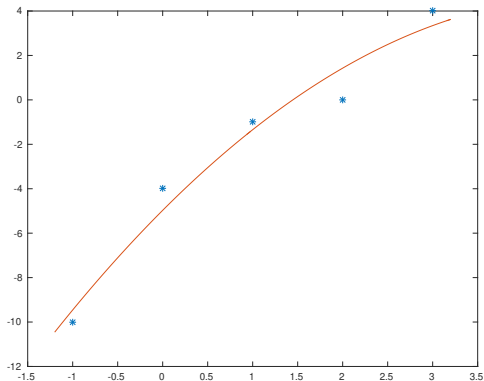
a p együtthatójú polinom értékeit adja az xx vektor koordinátaiban.
(p-ben a polinom együtthatói a főegyütthatóval kezdve szerepelnek)

```
x=[-2, -1, 0, 2];  
f=[-5, 3, 1, 15];  
p=polyfit(x,f,3);  
xx=linspace(-2.5,2.5);  
yy=polyval(p,xx);  
figure; plot(x,f,'*',xx,yy)
```



Fontos! Ha a `polyfit` függvényben nem megfelelően írjuk elő a polinom fokszámát, akkor a polinom nem feltétlenül illeszkedik az adatokra.

```
x=[-1 0 1 2 3]; f=[-10 -4 -1 0 4]; p=polyfit(x,f,2);  
xx=linspace(-1.2,3.2); ff=polyval(p,xx);  
figure; plot(x,f,'*',xx,ff)
```



3. feladat

Közelítse az

$$f(x) = e^x - \sin(\pi x)$$

függvényt a $[0, 1]$ intervallumon egy másodfokú polinommal. Ábrázolja az eredeti és az illesztett függvényt közös ábrán.

4. feladat

Tudjuk, hogy egy test méterben számolva s_0 utat tett meg, egyenletes v_0 (m/s) sebességgel, majd ezután egyenletesen gyorsítani kezdett a (m/s²) gyorsulással. A gyorsulás kezdetétől számítva a 2., 4. és 5. másodperc végén az összes megtett út rendre 16, 38 és 52 m. Határozza meg s_0 , v_0 és a értékét.

5. feladat

Rajzoltassuk ki közös ábrára az alábbi 3 függvényt:

- az

$$f(x) = \frac{1}{1 + 25x^2}$$

függvényt a $[-1, 1]$ intervallumon

- az f függvény

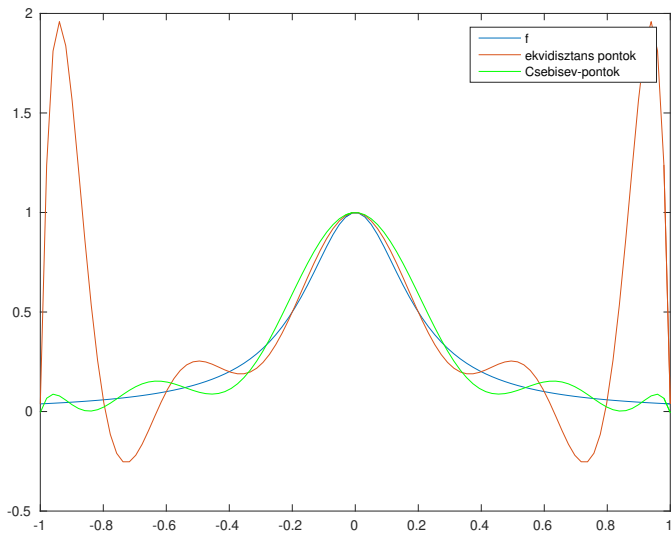
$$-1, -0.8, -0.6, \dots, 0.6, 0.8, 1$$

egyenlő lépésközű (ekvidisztáns) alappontokhoz tartozó
Lagrange-polinomját

- az f függvény

$$x_k = \cos\left(\frac{2k-1}{22}\pi\right), \quad k = 1, 2, \dots, 11$$

alappontokhoz (Csebisev-pontok) tartozó Lagrange-polinomját.



Hermite-interpoláció

Példa

Határozzuk meg az alábbi adatokra illeszkedő minimális fokszámú polinomot!

x_i	-2	-1	1
$f(x_i)$	-10	-2	2
$f'(x_i)$	-20	10	10
$f''(x_i)$		-16	

Megoldás. Az illeszkedési feltételek száma: $m = 7$, így az Hermite-polinom legfeljebb 6-odfokú lesz.

Készítsük el az osztott differenciák táblázatát!

A kiinduló adatok:

x_i	-2	-1	1
$f(x_i)$	-10	-2	2
$f'(x_i)$	-20	10	10
$f''(x_i)$		-16	

-2	-10		
		-20	
-2	-10		
-1	-2		
		10	
-1	-2		-8
		10	
-1	-2		
1	2		
		10	
1	2		

Számoljuk ki a hiányzó értékeket!

-2	-10		
		-20	
-2	-10		
-1	-2		
		10	
-1	-2		-8
		10	
-1	-2		
1	2		
		10	
1	2		

A hiányzó elsőrendű osztott differenciák:

-2	-10		
		-20	
-2	-10		
		8	
-1	-2		
		10	
-1	-2		-8
		10	
-1	-2		
		2	
1	2		
		10	
1	2		

A hiányzó másodrendű osztott differenciák:

-2	-10		
		-20	
-2	-10		28
		8	
-1	-2		2
		10	
-1	-2		-8
		10	
-1	-2		-4
		2	
1	2		4
		10	
1	2		

A harmadrendű osztott differenciák:

-2	-10			
		-20		
-2	-10		28	
		8		-26
-1	-2		2	
		10		-10
-1	-2		-8	
		10		2
-1	-2		-4	
		2		4
1	2		4	
		10		
1	2			

A negyedrendű osztott differenciák:

-2	-10				
		-20			
-2	-10		28		
		8		-26	
-1	-2		2		16
		10		-10	
-1	-2		-8		4
		10		2	
-1	-2		-4		1
		2		4	
1	2		4		
		10			
1	2				

Az ötödrendű osztott differenciák:

-2	-10					
		-20				
-2	-10		28			
		8		-26		
-1	-2		2		16	
		10		-10		-4
-1	-2		-8		4	
		10		2		-1
-1	-2		-4		1	
		2		4		
1	2		4			
		10				
1	2					

A hatodrendű osztott differencia:

-2	-10								
		-20							
-2	-10		28						
		8		-26					
-1	-2		2		16				
		10		-10		-4			
-1	-2		-8		4		1		
		10		2		-1			
-1	-2		-4		1				
		2		4					
1	2		4						
		10							
1	2								

-2	-10						
		-20					
-2	-10		28				
		8		-26			
-1	-2		2		16		
		10		-10		-4	
-1	-2		-8		4		1
		10		2		-1	
-1	-2		-4		1		
		2		4			
1	2		4				
		10					
1	2						

$$H(x) = -10 - 20(x+2) + 28(x+2)^2 - 26(x+2)^2(x+1) \\ + 16(x+2)^2(x+1)^2 - 4(x+2)^2(x+1)^3 \\ + 1(x+2)^2(x+1)^3(x-1)$$

4. feladat

Határozza meg az alábbi adatokra illeszkedő minimális fokszámú polinomot!

(a)

x_i	-1	1	2
$f(x_i)$	4	6	94
$f'(x_i)$	9	17	213

(b)

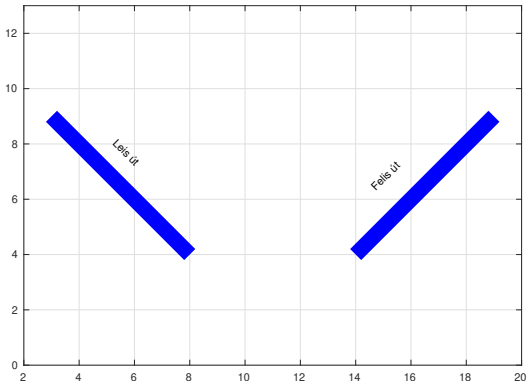
x_i	-2	-1	1
$f(x_i)$	13	3	7
$f'(x_i)$	-31	14	18
$f''(x_i)$		-40	

6. feladat

Írja fel az $f(x) = \cos(x) - 3x$ függvény $x_0 = 0$ -beli érintőjének az egyenletét!

7. feladat

Az ábrán látható két útszakasz (Leis út, Felis út) egymáshoz közelebbi végei között szeretnénk utat építeni úgy, hogy az így kapott út menetében ne legyen törés. Adja meg a hiányzó útszakasz nyomvonalát leíró függvényt!



Spline interpoláció Octave/Matlab-bal

Példa

Határozzuk meg az alábbi adatokhoz tartozó harmadfokú spline-t!

x_i	-2	-1	0	1	2	3
S	4	1	7	4	12	9
S'	15					8

Megoldás. Használjuk a spline függvényt!

`p=spline(x,y)`

Előállítja a szakaszonként harmadfokú spline együtthatóit. Itt x az alappontok vektora, az y vektor első és utolsó koordinátája a két végpontban adott deriváltérték, a többi koordináta a függvényértékek.

```
>>x=-2:3; y=[15 4 1 7 4 12 9 8]; p=spline(x,y)
p =
  scalar structure containing the fields:
    form = pp
    breaks =
        -2   -1    0    1    2    3

    coefs =
        19.0000   -37.0000    15.0000     4.0000
       -12.0000    20.0000    -2.0000     1.0000
        11.0000   -16.0000     2.0000     7.0000
       -12.0000    17.0000     3.0000     4.0000
        15.0000   -19.0000     1.0000    12.0000

    pieces = 5
    order = 4
    dim = 1
```

A spline együtthatói: p.coefs

Figyeljünk arra, hogy a polinomok együtthatóit a részintervallumok kezdőpontjaihoz viszonyítva kapjuk!

Az 5 illesztett polinom:

$$p_1(x) = 19(x+2)^3 - 37(x+2)^2 + 15(x+2) + 4$$

$$p_2(x) = -12(x+1)^3 + 20(x+1)^2 - 2(x+1) + 1$$

$$p_3(x) = 11x^3 - 16x^2 + 2x + 7$$

$$p_4(x) = -12(x-1)^3 + 17(x-1)^2 + 3(x-1) + 4$$

$$p_5(x) = 15(x-2)^3 - 19(x-2)^2 + (x-2) + 12$$

Ellenőrizzük az illeszkedési feltételeket!

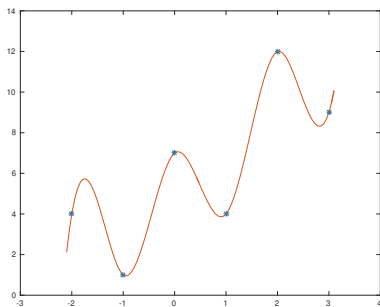
Ha nem az együtthatókat szeretnénk tudni, hanem a spline értékét valamely pont(ok)ban, akkor

```
yy=spline(x,y,xx)
```

ahol x és y az előbbi vektorok, xx azon pontok vektora, ahol a helyettesítési értéket keressük. Ekkor yy -ba kerülnek a kiszámolt függvényértékek.

```
>> x=-2:3;  
>> y=[15 4 1 7 4 12 9 8];  
>> xx=linspace(-2.1,3.1);  
>> yy=spline(x,y,xx);  
>> plot(x,y(2:end-1),'*',xx,yy)
```

```
x=-2:3;  
y=[15 4 1 7 4 12 9 8];  
xx=linspace(-2.1,3.1);  
yy=spline(x,y,xx);  
plot(x,y(2:end-1),'*',xx,yy)
```



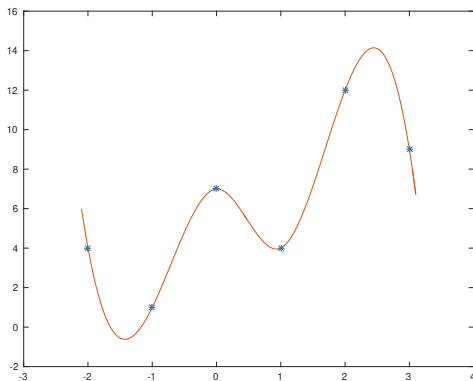
Az így kapott függvény teljesíti az illeszkedési feltételeket, és az első két deriváltja folytonos.

Megjegyzés

Ha a spline függvényt olyan x és y vektorokkal hívjuk, amelyek ugyanannyi koordinátát tartalmaznak, akkor a hiányzó két feltételt az Octave/Matlab azzal helyettesíti, hogy az első és utolsó két részintervallum találkozásánál a harmadik deriváltat is folytonosnak tekinti.

```
x=-2:3;  
y=[4 1 7 4 12 9];  
xx=linspace(-2.1,3.1);  
yy=spline(x,y,xx);  
plot(x,y,'*',xx,yy)
```

```
x=-2:3;  
y=[4 1 7 4 12 9];  
xx=linspace(-2.1,3.1);  
yy=spline(x,y,xx);  
plot(x,y,'*',xx,yy)
```



8. feladat

Rajzoltassuk ki közös ábrán az alábbi 3 függvényt:

- az

$$f(x) = \frac{1}{1 + 25x^2}$$

függvényt a $[-1, 1]$ intervallumon

- az f függvény

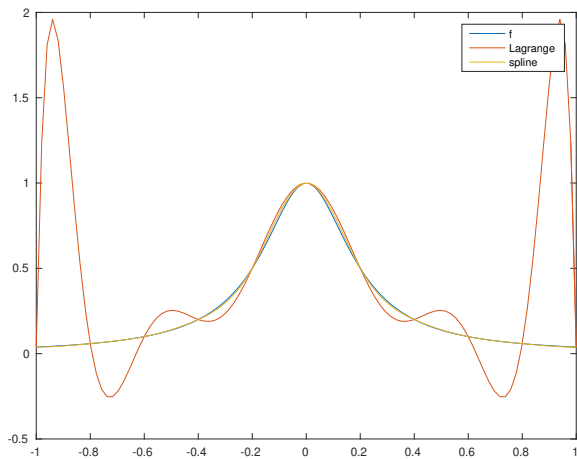
$$-1, -0.8, -0.6, \dots, 0.6, 0.8, 1$$

egyenlő lépcsőközű (ekvidisztáns) alappontokhoz tartozó
Lagrange-polinomját

- az f függvény

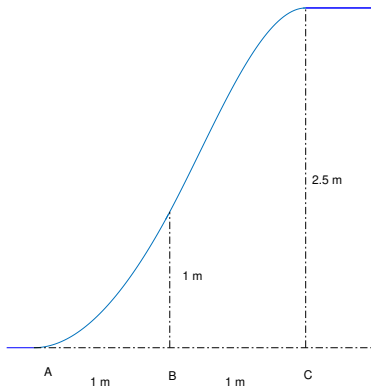
$$-1, -0.8, -0.6, \dots, 0.6, 0.8, 1$$

alappontokhoz tartozó harmadfokú spline polinomját. (A
végpontokban a deriváltértékeket tekintjük 0-nak.)



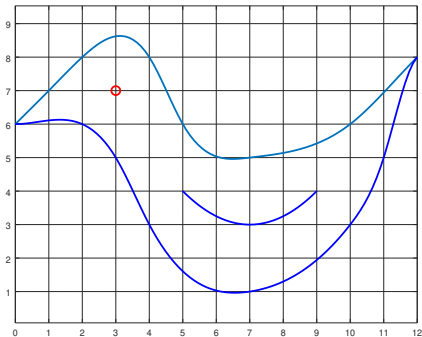
9. feladat

Az ábrán látható csúszda csúszófelületét szeretnénk elkészíteni két darabból úgy, hogy az A és C helyeken simán csatlakozzon a vízszintes felületekhez, illetve a két lemez is minél simábban csatlakozzon egymáshoz B -ben. Írja fel azt a függvényt, ami a csúszófelület lefutását modellezi!

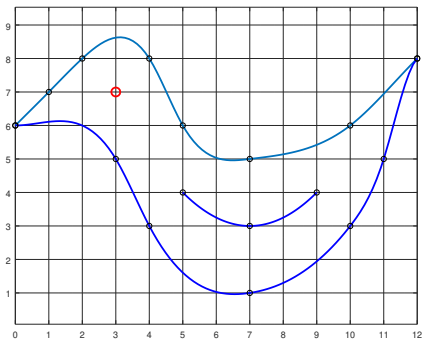


10. feladat (szorgalmi)

Készítse el Octave-val az ábrán látható rajzot.



Útmutatás: használja a bejelölt (egész koordinátájú) pontokat és a spline függvényt.



Numerikus integrálás

Anoním függvények, function handle

Függvényeket definiálhatunk parancssorban is:

```
>> f1= @(x) x.*sin(x);
```

Ilyen módon az $f1(x) = x \sin(x)$ függvényt definiáltuk, hívása pl.:

```
>> y=f1(pi/4)
y=
    0.5554
```

A @ szimbólum után zárójelben szerepelnek a függvény változói (most x), ezt követi a függvény (ez egy ú.n. anoním függvény). Az = baloldalán szereplő változó (most $f1$) egy ú.n. „function handle” típusú változó lesz.

Akár többváltozós függvényeket is megadhatunk így:

```
>> f2= @(x,y) x.^2+x.*y-y+3;
```

Ekkor pl.

```
>> z=f2(2,-1)
```

```
z=
```

```
6
```

A function handle tárolja azon változók értékét is, amelyek szükségesek a függvény kiértékeléséhez:

```
>> a=2.5; b=3;
```

```
>> f3= @(x) a*sin(x)+b*cos(x);
```

```
>> y=f3(-4)
```

```
y=
```

```
-0.0689
```

```
>> clear a b
```

```
>> y=f3(-4)
```

```
y=
```

```
-0.0689
```

1. feladat

Egészítse ki az alábbi kódrészletet úgy, hogy I az f függvény $[a, b]$ intervallum feletti határozott integráljának közelítése legyen

(a) összetett trapéz-képlettel,

(b) összetett Simpson-képlettel,

úgy, hogy az $[a, b]$ intervallumot m részre osztjuk.

```
function I=myinteg(a,b,m)
```

```
    f=@(x) ...
```

```
end
```

2. feladat

Az előző kódok segítségével közelítse az

$$\int_0^1 e^{-x^2} dx$$

integrál értékét $m = 3, 6, 12$ esetén.

Becsülje meg hány részintervallumra van szükség, ha $\varepsilon = 10^{-3}$ pontossággal szeretnénk tudni az integrál értékét.

Numerikus integrálás Octave-val

Egyváltozós függvények integrálására pl a quad függvényt használhatjuk.

Példa

Octave segítségével számítsuk ki az

$$\int_0^3 x\sqrt{1+x}dx$$

integrál értékét!

Megoldás.

```
>> f= @(x) x.*sqrt(1+x);  
>> quad(f,0,3)  
ans=  
    7.7333
```

A quad függvény hívása:

```
>> quad(fv,xmin,xmax)
```

ahol `fv` az integrálandó függvény (`fv` egy `function handle` típusú változó, vagy egy sztring, mely egy függvény neve), `xmin` és `xmax` az alsó és felső határ.

A quad függvény az `fv` függvényt vektor argumentummal fogja meghívni. Figyeljünk rá, hogy az `fv` ennek megfelelően legyen megadva (elemenkénti operátorok!).

A quad függvény Gauss-kvadratúrát használ (egy olyan interpolációs kvadratúraképlet, melynek alappontjai nem ekvidisztánsak, hanem bizonyos polinomok zérushelyei), és alapértelmezésként 10^{-8} abszolút, és 10^{-8} relatív hibával számítja ki az integrál értékét.

A hibahatárok átállíthatóak:

```
>> quad(f,0,3,[1e-10 1e-10])
```

A vektor első eleme az abszolút, második a relatív hiba.

Az előző példában nem feltétlenül szükséges létrehozni a `f` változót:

```
>> quad(@(x) x.*sqrt(1+x),0,3)
```

Ha a függvényt korábban egy `m`-fájlban definiáltuk, pl.

```
function y=myfnc(x)
    y=x.*sqrt(1+x)
end
```

akkor a `quad` függvénynek átadhatjuk a függvény nevét is (function handle-ként, vagy sztringként):

```
>> quad(@myfnc,0,3)
```

vagy

```
>> quad('myfnc',0,3)
```

Hasonló a helyzet az Octave beépített függvényeivel:

```
>> quad(@sin,0,pi)
```

```
>> quad('sin',0,pi)
```

Improprius integrálok

- Az integrálás határai lehetnek $-\infty$ és ∞ is:

```
>> f= @(x) exp(-x);  
>> quad(f,0,inf)  
ans = 1.0000
```

- Az sem probléma, ha a függvény az intervallum végpontjaiban nincs értelmezve:

```
>> f= @(x) 1./sqrt(1-x.^2);  
>> quad(f,-1,1)  
ans=  
3.1416
```

Ha nem ismert a függvény

Előfordulhat, hogy nem ismerünk egzakt képletet az integrálandó függvényre, csak bizonyos pontokban ismerjük az értékeit. Ilyenkor a trapz Octave-függvényt használhatjuk.

Példa

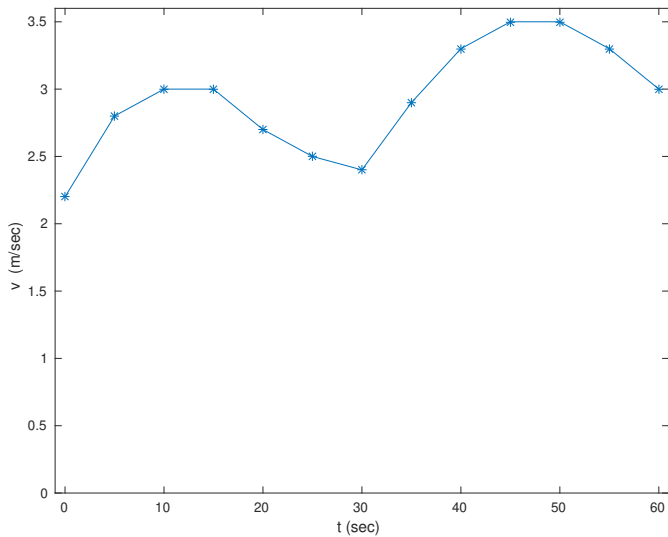
Egy jármű sebességét 1 percen keresztül mértük 5 másodperces időközönként:

t (sec)	0	5	10	15	20	25	30	35	40	45	50	55	60
v (m/sec)	2.2	2.8	3	3	2.7	2.5	2.4	2.9	3.3	3.5	3.5	3.3	3

Becsüljük meg a jármű által megtett utat!

Megoldás. Tudjuk, hogy az a idő alatt megtett út:

$$S = \int_0^a v(t) dt$$



A trapz függvény segítségével az integrál becslése:

```
>> x=0:5:60;  
>> f=[ 2.2 2.8 3 3 2.7 2.5 2.4 2.9 3.3 3.5 3.5 3.3 3];  
>> trapz(x,f)  
ans =  
    177.5000  
  
>> y=cumtrapz(x,f);
```

Ekkor

$$y = (0, 12.5, 27, 42, 56.25, 69.25, 81.5, 94.75, 110.25, 127.25, 144.75, 161.75, 177.5)$$

Az y vektor i -edik koordinátája az i -edik időpillanatig megtett utat mutatja.

3. feladat

Octave segítségével számítsa ki az alábbi határozott integrálok értékét!

(a)

$$\int_{-\pi/2}^{\pi/2} x \sin(x^2) dx$$

(b)

$$\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx$$

(c)

$$\int_{-1}^1 \sqrt{1-x^2} dx$$

4. feladat

Közelítse az

$$\int_0^{10} x \sin(5x) dx$$

integrált a quad függvénnyel, illetve a trapz függvénnyel úgy, hogy alappontoknak az

- `xi=0:10` pontokat
- `xi=[0 0.5:9.5 10]` pontokat

választja. Próbálja megmagyarázni a tapasztalt jelenséget (ábrázolja az integrálandó függvényt a megadott intervallum felett). Növelje az alappontok számát a trapz függvény esetén.

Felületek ábrázolása Octave-ban

Ha egy kétváltozós függvényt az $[a, b] \times [c, d]$ tartomány felett szeretnénk ábrázolni, akkor vegyünk fel osztópontokat az $[a, b]$ és $[c, d]$ intervallumban és a `meshgrid` függvény segítségével készítsük el az ezek által meghatározott rácspontokat. A rácspontokban számítsuk ki az ábrázolni kívánt függvény értékeit és ábrázoljuk a `mesh` vagy `surf` függvények segítségével.

Példa a rács előállítására:

Ha a tartomány $[0, 15] \times [0, 10]$, akkor

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

Ekkor `X` és `Y` is 11×16 -os mátrix.

```
>> 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)

Felületek ábrázolása Octave-ban

Példa

Ábrázoljuk az

$$f(x, y) = xe^{-(x^2+y^2)}$$

függvényt a $T = [-2, 2] \times [-1, 1]$ tartomány felett!

Megoldás.

Vegyünk fel sok pontot a $[-2, 2]$ és a $[-1, 1]$ intervallumban!

```
>> x=linspace(-2,2);
```

```
>> y=linspace(-1,1);
```

“Rácsozzuk be” a tartományt!

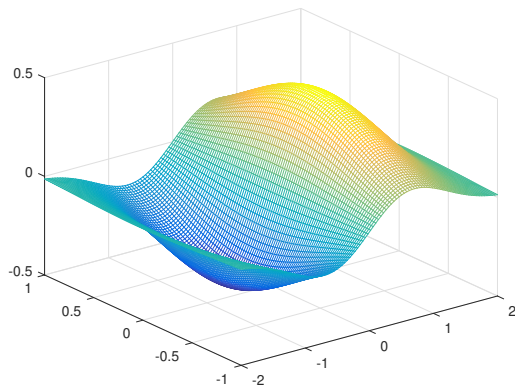
```
>> [xx,yy] = meshgrid(x,y);
```

Számítsuk ki a rácspontokban a függvény értékét!

```
>> zz = xx.*exp(-xx.^2-yy.^2);
```

Felületek ábrázolása Octave-ban

```
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)
```



Kétváltozós függvények numerikus integrálása Octave-val

Használjuk a `dblquad` függvényt!

Példa

Számítsuk ki a

$$\int_{-2}^2 \int_{-1}^1 x e^{-x^2-y^2} dy dx$$

integrál értékét!

Megoldás.

```
>> f = @(x,y) x.*exp(-x.^2-y.^2)
>> dblquad(f,-2,2,-1,1)
ans =      3.1812e-19
```

Megjegyzés: Az integrál kiszámítása `dblquad` függvénnyel bizonyos esetekben sok időt vehet igénybe.

5. feladat

Számítsa ki a következő integrálok értékét!

(a)

$$\int_0^{\pi/4} \int_{-\pi/3}^0 2y \sin x \cos^2 x dy dx$$

(b)

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{1}{2\pi} e^{-\frac{x^2+y^2}{2}} dx dy$$

Nemlineáris egyenletek

1. feladat

Egészítse ki a lent adott Octave-függvényt úgy, hogy az f függvény $[a, b]$ -beli gyökének felező-módszerrel történő e pontosságú közelítését és az ehhez szükséges lépésszámot adja vissza (ha $f(a) \cdot f(b) < 0$). `maxit` az elvégzett lépések számának maximumát jelöli.

```
function [x,k]=felezo(a,b,e,maxit)
```

```
    f=@(x)
```

```
    if f(a)*f(b)>0
```

```
        error(' ... ')
```

```
    end
```

```
end
```

2. feladat

Tesztelje az előző kódját az alábbi függvényekkel:

(a) $f(x) = x^3 - 3x - 2$ a $[0, 3]$ intervallumon

(b) $f(x) = x^3 - 3x + 2$ a $[0, 3]$ intervallumon

3. feladat

Az előző kód mintájára írjon egy kódot, mely húrmódszerrel közelíti a függvény gyökét.

4. feladat

Írjon egy Octave-függvényt, mely megadott x_0 kezdőpont, `maxit` maximális iterációszám, ε pontosság esetén egy adott f függvény gyökét közelíti Newton-módszerrel, a gyök közelítésével és az elvégzett iterációk számával tér vissza, illetve ha az algoritmus nem konvergál, vagy egy Newton-lépés nem definiált, akkor a megfelelő hibaüzenettel.

5. feladat

Az előző függvénnyel közelítse a 2. feladat (a) és (b) részében megadott függvények gyökeit az $x_0 = 1.5$ kezdőpontból indulva. Hasonlítsa össze az elvégzett lépések számát a két f függvény esetén. Hasonlítsa össze a lépésszámot a felező-, illetve húrmódszer esetén szükséges lépésszámokkal is. Magyarázza meg a tapasztalt jelenséget!

6. feladat

Alkalmazzuk a Newton-módszert az $f(x) = x^3 - 5x$ függvény gyökének közelítésére az $x_0 = 1$ pontból indulva!

7. feladat

Közelítse $\sqrt{5}$ értékét Newton-módszerrel.

8. feladat

Mutassa meg, hogy az $3x^3 - 12x + 4 = 0$ egyenletnek van gyöke a $[0, 1]$ intervallumban. Vizsgálja meg az $x_0 \in [0, 1]$,

$$x_{k+1} = \frac{3x_k^3 + 4}{12}, \quad k = 0, 1, \dots$$

eljárás konvergenciáját! Írjon egy Octave-kódot, amely kiszámolja az iteráció első néhány lépését! Módosítsa a kódot úgy, hogy olyan k értékre álljon le, amelyre $|x_k - x_{k-1}| < \varepsilon$, ahol $\varepsilon > 0$ adott.

9. feladat

Mit mondhatunk az $x_0 \in [-\pi/2, \pi/2]$, $x_{k+1} = \frac{1}{3} \cos(x_k)$ eljárás konvergenciájáról?

10. feladat

Tekintsük a

$$\begin{aligned} -4x_1 + \cos(2x_1 - x_2) &= 3 \\ -3x_2 + \sin x_1 &= 2 \quad x_1, x_2 \in [-\pi, \pi] \end{aligned}$$

egyenletrendszert. Mit mondhatunk az egyenletrendszer megoldhatóságáról, illetve az

$$\begin{aligned} x_1^{(k+1)} &= -\frac{3}{4} + \frac{1}{4} \cos(2x_1^{(k)} - x_2^{(k)}) \\ x_2^{(k+1)} &= -\frac{2}{3} + \frac{1}{3} \sin x_1^{(k)} \end{aligned}$$

($k = 0, 1, 2, \dots$) eljárás konvergenciájáról?

11. feladat

Próbálja meg az

$$xe^x - 1 = 0, \quad x \in [0.25, 1]$$

egyenlet gyökét az

$$x_0 = 0.5, \quad x_{k+1} = g(x_k), \quad k = 0, 1, \dots$$

fixpont-iterációval közelíteni úgy, hogy

(a) $g(x) = e^{-x}$

(b) $g(x) = \frac{1+x}{1+e^x}$

(c) $g(x) = x + 1 - xe^x$

Mit tapasztal?

12. feladat

Írjon egy Octave-függvényt, mely egy adott $g : \mathbb{R}^n \rightarrow \mathbb{R}^n$ függvény esetén a $g(x) = x$ egyenlet gyökét közelíti fixpont-iterációval. A függvény bemenő paraméterei a kezdővektor, a pontosság és a maximális iterációszám legyenek.

13. feladat

Közelítse a 10. feladatban adott egyenletrendszer gyökét az előző kóddal.

14. feladat

Közelítse az

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0.1x_1^2 + 0.1x_2^2 + 0.1x_3^2 \\ 0.1x_1 + 0.1x_2 + 0.1x_3 \\ 0.1x_1x_2x_3 + 0.3 \end{pmatrix}$$

egyenletrendszer $[0, 1] \times [0, 1] \times [0, 2]$ -beli gyökét fixpont-iterációval!

Nemlineáris egyenletek megoldása Octave-val

Például az `fsolve` függvényt használhatjuk.

```
[xopt,fopt] = fsolve(F,x0)
```

Az $F(x) = 0$ egyenlet gyökét közelíti az x_0 kezdőértékből indulva, ahol F akár egy többváltozós függvény is lehet.

x_{opt} : a gyök közelítése,

f_{opt} : a függvény értéke x_{opt} -ban

Példa

Közelítsük a $\cos(x) = 3x$ egyenlet megoldását az `fsolve` segítségével.

Írjuk át az egyenletet $F(x) = 0$ alakba:

$$\cos(x) - 3x = 0.$$

Hívjuk meg az `fsolve` függvényt pl $x_0 = 0$ kezdőértékkel.

```
>> F=@(x) cos(x)-3*x;  
>> [xopt,fopt]=fsolve(F,0)  
xopt = 0.31675  
fopt = -2.8390e-09
```

Példa

Közelítsük a 10. feladatban adott egyenletrendszer megoldását az `fsolve` segítségével.

```
>> F=@(x) [-4*x(1)+cos(2*x(1)-x(2))-3; -3*x(2)+sin(x(1))-2];  
>> [xopt,fopt]=fsolve(F,[0;0])
```

```
xopt =
```

```
-0.50406
```

```
-0.82766
```

```
fopt =
```

```
-3.9646e-09
```

```
3.1555e-10
```

Polinomok gyökeinek közelítésére a roots függvényt használhatjuk:

```
r=roots(p)
```

ahol a p vektorban a polinom együtthatóit kell felsorolni a főegyütthatóval kezdve.

Példa

Közelítsük a $p(x) = 2x^3 - x + 1$ polinom gyökeit.

```
>> p=[2 0 -1 1]
```

```
>> r=roots(p)
```

```
r =
```

```
-1.00000 + 0.00000i
```

```
0.50000 + 0.50000i
```

```
0.50000 - 0.50000i
```

15. feladat

- (a) Közelítse a $3x = \cos(x)$ egyenlet gyökeit!
- (b) Közelítse a $3x^3 - 12x + 4 = 0$ egyenlet gyökeit!
- (c) Közelítse az $e^x = \sin(x)$ egyenlet gyökeit!
- (d) Közelítse az $\ln(x) = 2 - x$ egyenlet gyökét!
- (e) Közelítse a $\cos^2(x) + 2\sin(x) = 2$ egyenlet gyökét!
- (f) Közelítse az $x^4 - x^3 - 2x^2 - 2x + 4 = 0$ egyenlet gyökeit!

16. feladat

Közelítse az alábbi egyenletrendszer gyökét a $[-1, 1]^2$ tartományon.

$$\sin(x_1 + 2x_2) + x_1x_2 = 0$$

$$\cos(x_2 - 1) - \sin(x_1) = 0$$

Optimalizálás

Egyváltozós függvények minimalizálása Octave-val

Az `fminbnd` függvény:

```
[xopt,fopt]=fminbnd(f,a,b)
```

Az f függvény $[a, b]$ intervallumbeli lokális minimumhelyének közelítését adja.

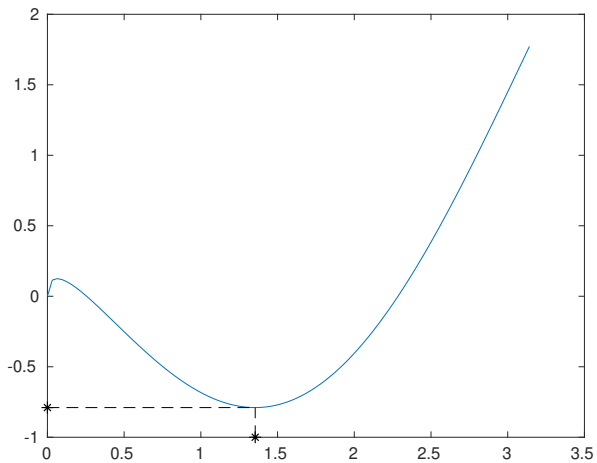
`xopt`: a lokális minimumhely közelítése

`fopt`: a függvény értéke az `xopt` helyen

Példa

Keressük meg az $f(x) = \sqrt{x} - 2\sin(x)$ függvény $[0, \pi]$ -beli minimumhelyét.

```
f=@(x) sqrt(x)-2*sin(x);  
[xopt,fopt]=fminbnd(f,0,pi)  
xopt = 1.3543  
fopt = -0.78957
```



Az `fminsearch` és `fminunc` függvények:

- `[xopt,fopt]=fminsearch(f,x0)`
- `[xopt,fopt]=fminunc(f,x0)`

Az f függvény lokális minimumhelyének közelítését adja az x_0 kezdőpontból indulva.

Mindkettő alkalmas többváltozós függvények minimalizálására is.

```
f=@(x) sqrt(x)-2*sin(x);  
[xopt,fopt]=fminsearch(f,0.5)  
    xopt =    1.3542  
    fopt =   -0.78957  
xopt,fopt]=fminunc(f,0.5)  
    xopt =    1.3543  
    fopt =   -0.78957
```

1. feladat

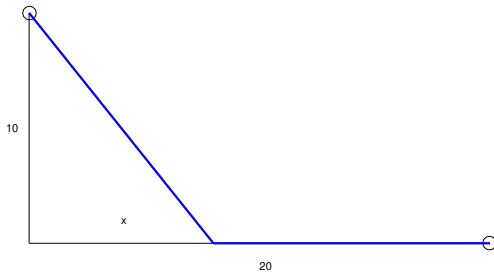
- (a) Határozza meg az $f(x) = x^2 + \cos(3x)$ függvény összes $[0, 6]$ intervallumbeli lokális minimumhelyét.
- (b) Határozza meg az $f(x) = x^2 + \cos(3x)$ függvény összes $[0, 6]$ intervallumbeli lokális maximumhelyét.

2. feladat

- (a) Határozza meg az $f(x) = \sin(2x) \sin(3x)$ függvény összes $[0, 5]$ intervallumbeli lokális minimumhelyét.
- (b) Határozza meg az $f(x) = \sin(2x) \sin(3x)$ függvény összes $[0, 5]$ intervallumbeli lokális maximumhelyét.

3. feladat

A parttól 10 km-re fekvő sziget áramellátását szeretnénk biztosítani egy olyan áramellátó központból, amely közvetlenül a parton helyezkedik el, 20 km-re a partnak a szigethez legközelebbi pontjától. Ha 250 ezer Ft-ba kerül 1 km víz alatti vezeték elhelyezése, és 100 ezerbe 1 km vezeték telepítése a szárazföldön, akkor határozzuk meg a minimális költségű útvonalat.



4. feladat

Egy 1 l űrtartalmú, henger alakú konzervdobozt szeretnénk készíteni. Határozza meg a doboz méreteit úgy, hogy adott vastagságú lemezből készítve a lehető legkevesebb anyagra legyen szükség az elkészítéséhez.

5. feladat

Egy 15 cm-szer 20 cm-es kartonlapból egy fedél nélküli dobozt szeretnénk hajtogatni (a lap 4 sarkából 1-1 négyzetet kivágva, a keletkező „füleket” felhajtva). Adja meg a doboz méretét úgy, hogy annak térfogata maximális legyen.

6. feladat

Egy 30 cm széles lemezből szeretnénk csatornát hajtogatni úgy, hogy a lemez két szélén 10-10 cm-t valamilyen szögben felhajtunk. Határozza meg a szöget úgy, hogy a csatornába a lehető legtöbb víz férjen.

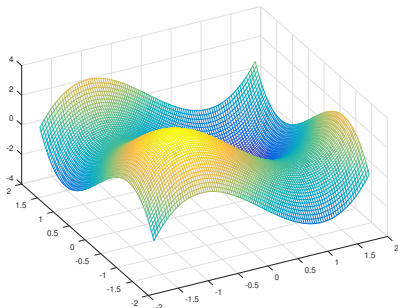


Felületek ábrázolása

Példa

Rajzoltassuk ki az $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, $f(x) = x_1^3 + x_2^3 - 3x_1 - 3x_2$ felületet a $[-2, 2] \times [-2, 2]$ tartomány felett.

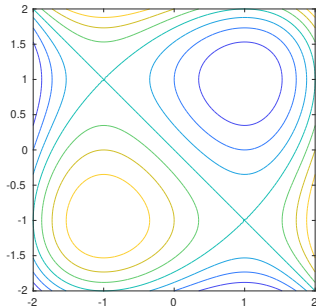
```
xx=linspace(-2,2);  
yy=xx;  
[X,Y]=meshgrid(xx,yy);  
Z=X.^3+Y.^3-3*X-3*Y;  
figure; mesh(X,Y,Z)
```



Példa

Rajzoltassuk ki az $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, $f(x) = x_1^3 + x_2^3 - 3x_1 - 3x_2$ felület szintvonalait a $[-2, 2] \times [-2, 2]$ tartomány felett.

```
xx=linspace(-2,2);  
yy=xx;  
[X,Y]=meshgrid(xx,yy);  
Z=X.^3+Y.^3-3*X-3*Y;  
figure; contour(X,Y,Z)  
axis equal
```



A szintvonalakra ráírthatjuk a „magassági számokat” is:

```
xx=linspace(-2,2);  
yy=xx;  
[X,Y]=meshgrid(xx,yy);  
Z=X.^3+Y.^3-3*X-3*Y;  
figure; contour(X,Y,Z,'ShowText','on')  
axis equal
```

Többváltozós függvények minimalizálása

Példa

Határozzuk meg az $f(x) = x_1^3 + x_2^3 - 3x_1 - 3x_2$ függvény gradiensét és a stacionárius pontjait.

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{pmatrix} = \begin{pmatrix} 3x_1^2 - 3 \\ 3x_2^2 - 3 \end{pmatrix}$$

Stacionárius pont: ahol $\nabla f(x) = 0$.

A függvénynek 4 stacionárius pontja van:

$$(-1, -1), \quad (-1, 1), \quad (1, -1), \quad (1, 1)$$

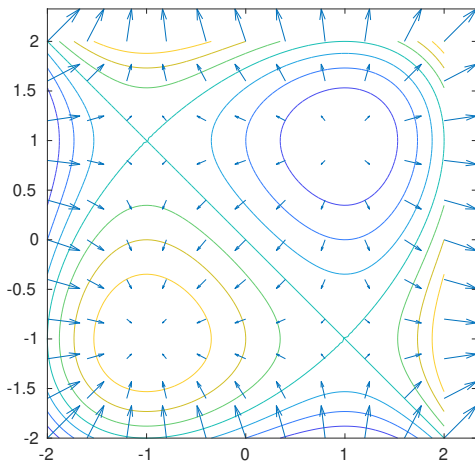
Példa

Rajzoltassuk ki az $f(x) = x_1^3 + x_2^3 - 3x_1 - 3x_2$ függvény szintvonalait és a gradiens mezőt a $[-2, 2] \times [-2, 2]$ tartomány felett.

Figyeljünk rá, hogy a gradiens mező rajzolásához nagyobb beosztású rácsot használjunk.

```
%a szintvonalak
xx=linspace(-2,2); yy=xx;
[X,Y]=meshgrid(xx,yy);
Z=X.^3+Y.^3-3*X-3*Y;
figure; contour(X,Y,Z)
axis equal

%a gradiensmezo
xx=linspace(-2,2,11); yy=xx;
[X,Y]=meshgrid(xx,yy);
dX=3*X.^2-3;
dY=3*Y.^2-3;
hold on; quiver(X,Y,dX,dY)
```



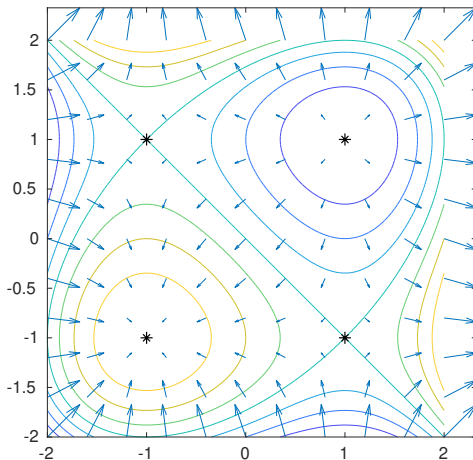
Az $f(x) = x_1^3 + x_2^3 - 3x_1 - 3x_2$ függvény szintvonalai és a gradiensmező.

A gradiensmező kirajzolásához használhatjuk az Octave gradient függvényét is. (Ekkor nem kell kiszámolnunk a gradienst, az Octave numerikusan közelíti azt)

```
%a szintvonalak
xx=linspace(-2,2); yy=xx;
[X,Y]=meshgrid(xx,yy);
Z=X.^3+Y.^3-3*X-3*Y;
figure; contour(X,Y,Z)
axis equal

%a gradiensmezo
xx=linspace(-2,2,11); yy=xx;
[X,Y]=meshgrid(xx,yy);
Z=X.^3+Y.^3-3*X-3*Y;
[dX,dY]=gradient(Z);
hold on; quiver(X,Y,dX,dY)
```

Tegyük rá az ábrára a stacionárius pontokat is!

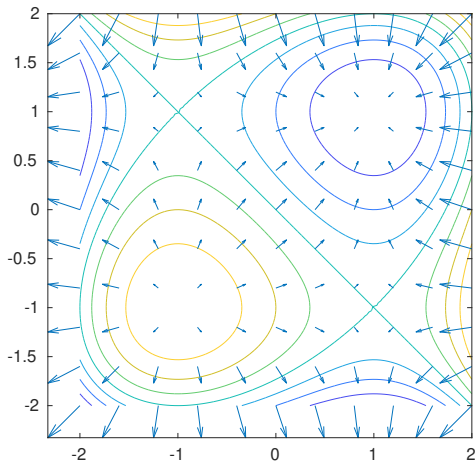


Az előző ábrán megfigyelhetjük, hogy

- a gradiensvektor merőleges az adott pontbeli szintvonalra
- a vektorok hossza a gradiens nagyságát, az iránya a gradiens irányát mutatja
- a stacionárius pontokban a gradiensvektor hossza 0

A gradiensvektor az adott pontban a legmeredekebb emelkedés irányába mutat, a (-1) -szerese (a negatív gradiens) pedig a legmeredekebb csökkenés irányába.

Ha a gradiensmező helyett a negatív gradiensmezőt rajzoltatjuk ki, akkor a nyilak a csökkenés irányába mutatnak.



Az $f(x) = x_1^3 + x_2^3 - 3x_1 - 3x_2$ függvény szintvonalai és a negatív gradiensmező.

Példa

Határozzuk meg az $f(x) = x_1^3 + x_2^3 - 3x_1 - 3x_2$ függvény stacionárius pontjainak típusát!

A függvény Hesse-mátrixa:

$$H(x) = \begin{pmatrix} 6x_1 & 0 \\ 0 & 6x_2 \end{pmatrix}$$

Ha $x = (-1, -1)$, akkor

$$H(x) = \begin{pmatrix} -6 & 0 \\ 0 & -6 \end{pmatrix},$$

így $\Delta_1 < 0$, $\Delta_2 > 0$, tehát ez a pont lokális maximumhely.

Ha $x = (-1, 1)$, akkor

$$H(x) = \begin{pmatrix} -6 & 0 \\ 0 & 6 \end{pmatrix},$$

így $\Delta_1 < 0$, $\Delta_2 < 0$, tehát ez a pont nyeregpont.

Ha $x = (1, -1)$, akkor

$$H(x) = \begin{pmatrix} 6 & 0 \\ 0 & -6 \end{pmatrix},$$

így $\Delta_1 > 0$, $\Delta_2 < 0$, tehát ez a pont nyeregpont.

Ha $x = (1, 1)$, akkor

$$H(x) = \begin{pmatrix} 6 & 0 \\ 0 & 6 \end{pmatrix},$$

így $\Delta_1 > 0$, $\Delta_2 > 0$, tehát ez a pont lokális minimumhely.

7. feladat

Számolja ki az alábbi függvények gradiensét, a stacionárius pontjaikat, és határozza meg a stacionárius pontok típusát. Rajzoltassa ki a felületeket, illetve egy másik ábrán a függvény szintvonalait és a **negatív** gradiens mezőt.

(a) $f(x_1, x_2) = 10 - x_1^2 - x_2^2$

(b) $f(x_1, x_2) = x_1^2 x_2 - 2x_1 x_2^2 + 3x_1 x_2 + 2$

(c) $f(x_1, x_2) = x_1^3 - x_1^2 x_2^2 - x_1 + x_2^2$

8. feladat

Határozza meg azt a téglatestet, melynek térfogata 1000 cm^3 és éleinek összhossza minimális.

9. feladat

Egy cég 20000 \$-t költött egy új termék kifejlesztésére. A termék előállítási költsége darabonként 2 \$. Egy piackutató szerint, ha a cég R \$-t költene reklámra, és ezután a terméket darabonként A \$-ért árulná, akkor a kereslet

$$2000 + 4\sqrt{R} - 20A$$

darab lenne. Mennyit érdemes reklámra költeni, és milyen áron érdemes kínálni a terméket, ha a hasznót maximalizálni szeretnék?

Többváltozós függvények minimalizálása Octave-val

Az `fminsearch` vagy az `fminunc` függvényeket használhatjuk.

Példa: Keressük meg az $f(x) = x_1^3 + x_2^3 - 3x_1 - 3x_2$ függvény egy lokális minimumhelyét.

Mindkét függvény hívásához meg kell adnunk a minimumhely egy kezdeti közelítését.

```
>> f=@(x) x(1)^3+x(2)^3-3*x(1)-3*x(2);  
>> [xopt,fopt]=fminsearch(f,[0.5,0.5])
```

```
xopt =  
    0.99996    1.00000
```

```
fopt =  
   -4.0000
```

Az `fminunc` függvénnel:

```
>> f=@(x) x(1)^3+x(2)^3-3*x(1)-3*x(2);  
>> [xopt,fopt]=fminunc(f,[0.5,0.5])
```

```
xopt =  
    1.0000    1.0000
```

```
fopt = -4.0000
```

10. feladat

Rajzoltassa ki a megadott tartomány felett az alábbi kétváltozós függvényeket, a szintvonalait, a negatív gradiensmezőt és közelítse az adott tartományon belül a minimumhelyüket.

- $f(x_1, x_2) = \frac{1}{6}x_1^3 - x_1 + \frac{1}{4}x_1x_2^2$, ha $x \in [-2.5, 2.5]^2$
- $f(x_1, x_2) = \sin(x_1) \cos(x_2)$, ha $x \in [0, 2\pi) \times [0, 2\pi)$
- $f(x_1, x_2) = x_2(1 - x_1^2 - x_2^2)$, ha $x \in [-1.5, 1.5]^2$

11. feladat

Egy téli üdülőövezetben a mentőhelikopter bázisállomását úgy szeretnénk elhelyezni, hogy az n adott síközponttól mért legnagyobb távolsága minimális legyen. Írjon egy Octave-függvényt, melynek input paramétere az az $A \in \mathbb{R}^{n \times 2}$ mátrix, melynek soraiban a síközpontok koordinátái találhatóak, output paramétere pedig a mentőhelikopter bázisállomásának koordinátáit tartalmazó kételemű vektor.

12. feladat

Adott egy bolthálózat n üzletének elhelyezkedése. Helyezzük el az áruraktárt úgy, hogy az üzletektől vett távolságainak összege minimális legyen. Írjon egy Octave-függvényt, melynek input paramétere az az $A \in \mathbb{R}^{n \times 2}$ mátrix, melynek soraiban az üzletek koordinátái találhatóak, output paramétere pedig az áruraktár koordinátáit tartalmazó kételemű vektor.

13. feladat

Írjon egy Octave-függvényt, mely egy adott f függvény minimumhelyét közelíti Newton-módszerrel egy x_0 kezdővektorból indulva ε pontossággal, legfeljebb `maxit` lépést végezve. A függvény output paraméterei a minimumhely közelítése és az elvégzett lépések száma legyen.