

## 5. GYAKORLAT

**FÜGGVÉNYVIZSGÁLAT, LINEÁRIS  
EGYENLETRENDSZEREK****FÜGGVÉNYVIZSGÁLAT, NEVEZETES PONTOK, HATÁROZOTT  
INTEGRÁL****Minta feladat**

Ábrázoljuk az  $f: x \rightarrow 12 \cdot \cos(0,07x) \cdot \sin(1,2x) + 1$  függvényt a  $[20, 27]$  intervallumban, majd határozzuk meg a nevezetes pontjait! (Zérushelyek, minimum- és maximumhelyek.)

Először a d:\munka könyvtárban hozzuk létre a megszokott módon az f3.m fájlt (saját függvény), majd ábrázoljuk a függvényünket.

```
function y = f3(x)
    y=12*cos(0.07*x).*sin(1.2*x) + 1;
end
```

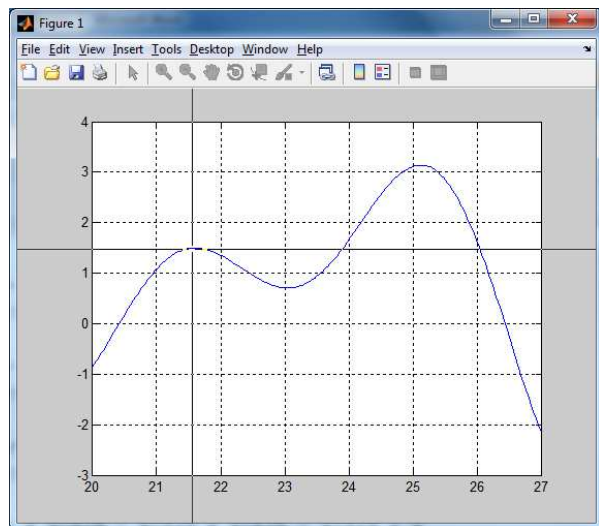
```
>> fplot('f3', [20 27]), grid on
```

A következő lépés az adatok egérrel történő leolvasása, majd a kapott közelítő értékek pontosítása a megfelelő függvénnyel (fzero, fminbnd).

Létező grafikon esetén a

[p q] = ginput

utasítás után az ábránkon egérrel egy pontra kattinthatunk, majd ezt a pozíciót Enterrel nyugtázzhatjuk. A két koordináta a p és q változókhoz rendelődik. Ha a ginput(n) alakot használjuk, akkor a p és q vektorok n eleműek lesznek, azaz folyamatosan n darab pont érzékelését végezhetjük el.



Az előbbi ábráról olvassuk be a következő pontok közelítő értékeit ilyen sorrendben:

- a két zérushely
- a két lokális maximumpont
- az egy lokális minimumpont!

Ellenőrizzük, hogy a p és q vektorok (nagyjából) megfelelő értékeket kaptak-e!

A letapogatott pontok koordinátáit a következők szerint pontosítjuk.

A zérushelyek keresését az `fzero('függvény', hol)` paranccsal végezhetjük el.

```
>> zhx = [fzero('f3', p(1)) fzero('f3', p(2))], zhy = [0 0]
```

A minimumhely kereséséhez az `fminbnd('függvény', alsó határ, felső h.)` parancsot használjuk.

```
>> minx = fminbnd('f3', p(5)-0.5, p(5)+0.5), miny = f3(minx)
```

Maximumhely kereséshez az `fminbnd` keresést a  $-f(x)$  függvényre alkalmazzuk.

```
>> maxx(1)=fminbnd('-f3(x)',p(3)-0.5,p(3)+0.5), maxy(1)=f3(maxx(1))  
>> maxx(2)=fminbnd('-f3(x)',p(4)-0.5,p(4)+0.5), maxy(2)=f3(maxx(2))
```

A pontokat feltesszük fekete, piros és zöld körökkel a grafikonra.

```
>> hold on  
>> plot(zhx,zhy,'ko'), plot(maxx, maxy,'ro'), plot(minx, miny,'go')
```

Tegyük fel az ábrára jelmagyarázatot! (A megadási sorrend az ábraelemek sorrendje.)  
Adjunk az ábrának címet is!

```
>> title('f(x) = 12*cos(0.07*x)*sin(1.2*x) + 1')
```

A határozott integrál (függvény alatti terület) kiszámítására – numerikusan – a `quad('függvény', alsó határ, felső határ, pontosság)` parancs szolgál.

```
>> integral = quad('f3', zhx(1), zhx(2), eps)
```

Ezt is kiírjuk az ábrára a `text(x, y, 'szöveg')` parancs segítségével.

```
>> text(20.5, -2.5, ['Zérushelyek közötti határozott integrál: '  
num2str(integral)])
```

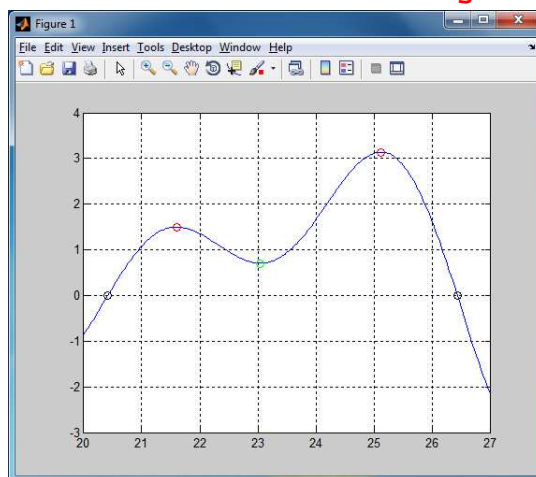
% Itt a karakterláncot egy  
sorvektorban adtuk meg  
szeletenként.

Végül mentjük el az ábránkat `abra1.jpg` néven.

```
>> print -djpeg90 -r300 abra1  
% az aktuális könyvtárba
```

A `print` utasítás paraméterei (lásd `help print`):

- `-d` után a típus jön (ps, psc, eps, epsc, jpeg<nn>, tiff, png, ...)
- `-r` után a dot/inch



## LINEÁRIS EGYENLETRENDSZEREK – ALAPESET

### Feladat

Oldjuk meg a következő lineáris egyenletrendszert:

$$\begin{aligned}6x_1 - 6x_2 - 2x_3 + 5x_4 &= 20 \\ -x_1 + 8x_2 + 5x_3 + 5x_4 &= -15 \\ 2x_1 - 9x_2 - 9x_3 - 3x_4 &= 22 \\ -4x_1 - x_2 + 3x_3 + 10x_4 &= -10\end{aligned}$$

Tipp: Először az inverz mátrixos módszerrel dolgozunk. Kövessük az alábbi lépéseket!  
Az  $Ax = b$  lineáris egyenletrendszer teljes táblázatát az M.dat fájlból tölthetjük be (M mátrix).

```
>> load M.dat M % vagy: load M.dat
```

Az M mátrix első 4 oszlopát töltsük az A mátrixba, utolsó oszlopát a b vektorba!  
Számítsuk ki az A mátrix determinánsát, rangját! (Ellenőrizzük, hogy a mátrix valóban invertálható.)

Számítsuk ki az A mátrix inverzét!

Határozzuk meg az  $x = \text{inv}(A)*b$  megoldásvektort!

A megoldás ellenőrzéseként számítsuk ki az  $Ax - b$  eltérést!

Ezután alkalmazzuk a balosztós módszert. Eszerint  $x = A \backslash b$ . Itt is végezzünk ellenőrzést.  
(Megj.: A pszeudoinverzes számolás is jó eredményt ad. Ezt önálló feladatként érdemes ellenőrizni.)

## SZÖVEGES FELADATOK

### Feladat

Határozzuk meg a Matlab segítségével a következő probléma megoldását!

A Szigorlatra Igyekvő Diákok Szalonjában (SZIDSZ) négyféle, egyenként 100-100 grammos arcpakolás van. A Gizi-pakolás 12 g tojást, 27 g fürjfület, 50 g gélt és 11 g sódert tartalmaz. A Béla-pakolás 34 g tojást, 8 g fürjfület és 58 g sódert vegyít. A Maris-pakk 65 g fürjfület, 21 g gélt és 14 g sódert tartalmaz. A Dove-kence 53 g tojás, 30 g fürjfül és 17 g sóder bedarált elegye. A raktárban 532 g tojás, 987 g fürjfül, 689 g gél és 792 g sóder várja, hogy felhasználják. Melyik pakolásból hányat készíthetünk, ha minden nyersanyagot el akarunk használni? Mennyi lesz a bevételünk, ha pakolások rendre 1100, 1200, 1300, 1400 forintba kerülnek? (A feladat eredete: Sotepedia, op.kut minták)

Tipp:

Az adatok bevitele után ellenőrizzük, hogy a feladat megoldása valóban egyértelmű!

Most az inverz mátrixos és a balosztós módszer egyaránt használható.

Az egyenletrendszer Excel környezetben történő megoldását mutatja a következő ábra.

The screenshot shows an Excel spreadsheet with the following data:

|    | A       | B        | C        | D        | E        | F     | G       | H |
|----|---------|----------|----------|----------|----------|-------|---------|---|
| 1  |         | Gizi     | Béla     | Maris    | Dove     |       | Készlet |   |
| 2  | Tojás   | 12       | 34       | 0        | 53       |       | 532     |   |
| 3  | Fürjfül | 27       | 8        | 65       | 30       |       | 987     |   |
| 4  | Gél     | 50       | 0        | 21       | 0        |       | 689     |   |
| 5  | Sóder   | 11       | 58       | 14       | 17       |       | 792     |   |
| 6  |         |          |          |          |          |       |         |   |
| 7  | Eár     | 1100     | 1200     | 1300     | 1400     |       |         |   |
| 8  |         |          |          |          |          |       |         |   |
| 9  |         | Inverz   |          |          |          |       | Mo.     |   |
| 10 |         | 0,00456  | -0,0071  | 0,02311  | -0,0017  |       | 10      |   |
| 11 |         | -0,00427 | -0,00395 | -0,00131 | 0,02029  |       | 9       |   |
| 12 |         | -0,01087 | 0,01691  | -0,00741 | 0,00404  |       | 9       |   |
| 13 |         | 0,02058  | 0,00414  | -0,0044  | -0,01263 |       | 2       |   |
| 14 |         |          |          |          |          |       |         |   |
| 15 | Haszon  | 11000    | 10800    | 11700    | 2800     | 36300 |         |   |

**Feladat**

Határozzuk meg a Matlab segítségével a következő probléma megoldását!

Adél három almát, 12 banánt és egy sárgadinnyét vesz 2 Euro 36 centért. Béla 12 almát és két sárgadinnyét vesz 5 Euro 26 centért. Cecília két banánt és három sárgadinnyét vesz 2 Euro 77 centért. Mennyibe kerül egy-egy darab gyümölcs?

**LINEÁRIS EGYENLETRENDSZEREK – SZINGULÁRIS ESETEK****Feladat**

Oldjuk meg a következő mátrixokkal megadott egyenletrendszereket!

1. A.dat b.dat – egyértelműen megoldható
2. As.dat bs.dat – As ugyan négyzetes, de szinguláris
3. At.dat bt.dat – túlhatározott (több egyenlet, mint ismeretlen)

As:

$$\begin{aligned} x_1 + 5x_2 + 13x_3 + x_4 + 4x_5 &= 19 \\ 2x_1 + 3x_2 + 12x_3 - 2x_4 - 3x_5 &= 4 \\ -4x_1 + 7x_2 + 3x_3 - 5x_4 + x_5 &= -35 \\ 3x_1 + x_2 + 4x_3 + 2x_5 &= 14 \\ 2x_1 + 16x_2 + 32x_3 - 6x_4 + 4x_5 &= 2 \end{aligned}$$

At:

$$\begin{aligned} x_1 + 5x_2 + 13x_3 &= 6 \\ 2x_1 + 3x_2 + 12x_3 &= 5 \\ -4x_1 + 7x_2 + 3x_3 &= 3 \\ 3x_1 + x_2 + 4x_3 &= -1 \\ -2x_1 + 4x_2 + x_3 &= 2 \end{aligned}$$

Tipp: Az első egyenletrendszert megoldását már ismerjük.

A másodikonál azonnal látjuk, hogy a rang csak 4, ill. a determináns 0, *inverz tehát nem létezik* (az  $A_s$  mátrix összefüggő). Azt is tudjuk ellenőrizni, hogy az  $[A_s \ b_s]$  rendszer rangja is 4, ill. hogy az eltolt rendszer determinánsa is 0 (és annak a rangja is 4):

```
>> rank([As bs])  
>> As2= [As(:,2:5), bs], det(As2), rank(As2)
```

Ebből látjuk, hogy a teljes rendszer is összefüggő (ellenőrizhető, hogy esetünkben az utolsó egyenlet az első négy összege). Mint tudjuk, ekkor végtelen sok megoldás létezik.

A megoldásnál a balasztós eljárás itt is alkalmazható, és a pszeudoinverzrel is dolgozhatunk (`pinv` parancs). Ellenőrizzük, hogy mindkét módszer jó megoldást ad! (*De ezek nem azonosak!*) A `pinv` paranccsal kapott megoldás optimális abban a tekintetben, hogy minimális vektorhosszú. (Lásd `norm` parancs.)

A túlhatározott egyenletrendszer ugyanabba a kategóriába sorolható, mint a négyzetes, ellentmondásos rendszer.

Ellenőrizhetjük, hogy az együtthatómátrix rangja kisebb a konstansoszloppal bővített rendszer rangjánál.

```
>> rank(A_t), rank([A_t b_t])
```

(Ha a rendszerünk négyzetes, akkor ellentmondásos esetben az eredeti rendszer determinánsa 0, az eltolt rendszeré ugyanakkor nem 0 – persze nálunk most determináns nem számolható).

Ilyenkor nem létezik megoldás, csak arra van reményünk, hogy az  $Ax - b = 0$  esethez „közel kerülünk”. Célunk tehát olyan  $x$  vektor előállítása, amelyre  $\|Ax - b\|$  minimális.

Most is használható a balasztós módszer és a pszeudoinverzes módszer is. (Mindkét esetben ugyanazt a „megoldást” kapjuk.)

Ellenőrizzük az eredményt visszaszorzással! Számoljuk ki az  $\|Ax - b\|$  vektornormát is!

```
>> norm(A_t*x_t - b_t)
```

## HATÉKONYSÁGI ELEMZÉS

### Mintafeladat

Most összehasonlítjuk a balasztós és az inverz mátrixos megoldás hatékonyságát.

Megoldjuk a  $Tx = b$  egyenletrendszert, ahol  $T$  egy  $3000 \times 3000$  méretű speciális mátrix (Töplitz-mátrix):

```
>> n=3000; T = toeplitz(n:-1:1); b = linspace(1,2,n)';  
>> tic; x = T\b; toc
```

Majd másképp:

```
>> tic; x = inv(T)*b; toc
```

Melyik megoldási metodika a gyorsabb?

## OTTHONI MUNKA

### Feladat (saját függvény)

Végezzük el a teljes függvényvizsgálatot a Matlab segítségével a megismert módon egy alkalmas gyakorló függvényre (választhatunk a matek tankönyvünkből is)! Térjünk ki a következőkre: értelmezési tartomány, szakadási helyek, zérushelyek, monotonitás (deriváltak), nevezetes pontok. Készítsünk ábrát a tanult módon. (Kiegészítésként érdemes megnézni a szimbolikus extra óra anyagát is.)

Például legyen a függvény  $x^3 - 3x^2 + x + 1$ , az ábrázolást pedig a  $[-1, 3]$  intervallumban végezzük el.

### Feladat (3d grafika)

Keressünk a sűgóban/interneten példákat a kétváltozós függvények ábrázolására (többdimenziós grafika)! Próbáljuk ki őket, és módosítsunk egyes paramétereket!

(Pl.  $fk = @(x,y) x.^2 + y.^2$ , ezcontourf(fk), axis square)

### Mintafeladat (implicit megadású függvények)

Rajzoljuk ki az  $x^2 + y^2 = 1$  síkegyenlettel adott kört az ezplot utasítással úgy, hogy a tengelyeket  $-1,1$ -től  $+1,1$ -ig skálázzuk!

```
>> f = inline('x^2+y^2=1'), argnames(f)
>> k=1.1; ezplot(f, [-k k -k k]), axis square
% inline megadás
>> k=1.1; ezplot('x^2+y^2=1', [-k k -k k]), axis square
% sztringes megadás
```

### Mintapéllda (implicit megadású függvények)

Rajzoljuk ki az  $x^3 + y^3 = 3a \cdot x \cdot y$  implicit egyenlettel adott Descartes-levelet!

```
>> a = 2; % ekkor 3*a értéke 6 lesz
>> f = inline('x^3+y^3=6*x*y'), argnames(f), ezplot(f, [-2*a 3*a -2*a 3*a]), axis square
```

### Feladat (lineáris egyenletrendszer, egyértelmű eset)

Gyakoroljuk más négyzetes, egyértelmű megoldással rendelkező egyenletrendszerekkel is a balosztós, ill. az inverzes megoldást. Ilyen példákat találhatunk pl. a „Solver minták” Excel gyakorló feladatban.

### Feladat (lineáris egyenletrendszer, nem egyértelmű eset)

A fenti túlhatározott egyenletrendszerre próbáljuk ki, hogy a „megoldás” az At\eye(sorok száma) paranccsal is meghatározható.

A fenti szinguláris egyenletrendszert úgy is oldjuk meg, hogy vegyünk fel egy olyan egyenletrendszert, amely eggyel kevesebb egyenletből áll ( $As_2$  mátrix), majd az  $As_2'/(As_2*As_2')$  képlettel számított kváziinverz segítségével dolgozzunk!

**Feladat (nagy méretű lineáris egyenletrendszer, időmérés)**

Végezzünk a fentihez hasonló hatékonysági elemzést alkalmasan megválasztott, különböző méretű tesztmátrixokkal. Mérjük le a végrehajtási időket, és ábrázoljuk az eredményt szép ábrán (lásd előadás)!

