



Matlab 6. előadás

Lineáris algebrai alkalmazások

Dr. Szörényi Miklós,
Dr. Kallós Gábor

2017–2018





Tartalom

- Matematikai alapok
- Transzformációs mátrixok
 - Swap mátrixok
 - Forgatómátrixok
 - Titkosítás
- Vektoralgebra
 - Vektorok skaláris szorzata, hajlásszöge
 - Vektorok vektoriális szorzata
 - Vektorok vegyesszorzata, függetlensége
- Lineáris egyenletrendszerek
 - Áttekintés
 - Alapeset, szöveges feladatok
 - Túlhatározott rendszerek
 - Összefüggő rendszerek
 - Időmérés
- Normák, kondíciószám
- Sajátérték, sajátvektor



Matematikai alapok (lineáris algebra)

Vektortér, bázis, lineáris transzformációk

Gyors elméleti összefoglaló a mátrixokhoz
(bizonyítások nélkül)

Lineáris tér: az a halmaz, melynek elemein (vektoroknak nevezzük) értelmezve van a "nyújtás" és "összeadás" művelete (lineáris műveletek) és ezek nem vezetnek ki a halmazból

Lineáris tér bázisa: a tér vektorainak egy legszűkebb (minimális elemszámú) részhalmaza, melynek elemeiből a tér többi eleme összeadásokkal és nyújtásokkal (lineáris kombináció) elérhető. Egy bázis elemei lineárisan függetlenek, azaz semelyikük sem állítható elő a többi elem lineáris kombinációjaként. Egy bázis elemeinek száma a tér dimenziója.

Vektor koordinátái: bármely vektor bázisvektoros előállításában szereplő nyújtási együtthatókat az adott bázisra vonatkozó koordinátáknak nevezzük.

Lineáris transzformáció: a lineáris tér elemein értelmezett olyan leképezés, mely a lineáris műveletekkel sorrendben felcserélhető, azaz teljesen mindegy, hogy például előbb transzformálok és azután nyújtok, vagy fordítva

Mátrix: egy lineáris térbeli A lineáris transzformációt $\underline{A}$ számtáblázat reprezentálja, melynek oszlopaiba a bázisvektorok transzformáltjainak koordinátái kerülnek. Minden mátrixelem pozícióját a sor ill. oszlopbeli elhelyezkedése (sorindex, oszlopindex) jellemzi

Oszlopvektor: a tér egy vektorának koordinátáit egyoszlopos mátrixba írjuk

A : z tengely körüli β szögű forgatás

β_1 60 fok

$\underline{A} = \begin{bmatrix} 0,5 & -0,86603 & 0 \\ 0,866025 & 0,5 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

β_2 -15 fok

$\underline{B} = \begin{bmatrix} 0,965926 & 0,258819 & 0 \\ -0,25882 & 0,965926 & 0 \\ 0 & 0 & 1 \end{bmatrix}$



Matematikai alapok (transzform. mátrixok)

■ Lineáris transzformációk, transzformációs mátrixok* (forrás: Kiss Emil, ELTE)

Lineáris transzformációk.

Definíció

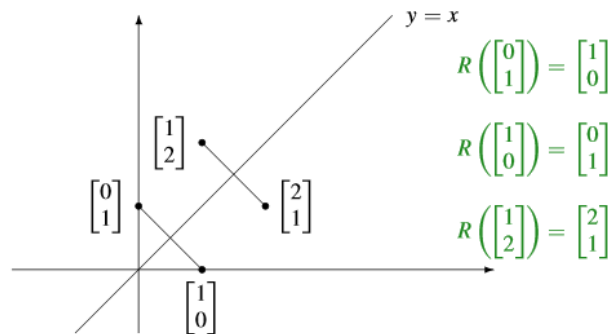
T test. Az $A : T^n \rightarrow T^n$ függvény *lineáris transzformáció*, ha tetszőleges $v, w \in T^n$ vektorra és λ skalárra teljesül, hogy

$$A(v + w) = A(v) + A(w) \text{ és } A(\lambda v) = \lambda A(v).$$

Vagyis A *összegtartó* és *skalárszoros-tartó*.

Példa

Legyen R az $y = x$ egyenesre való tükrözés.



Vektor képe általános transzformációnál.

Kérdés

Legyen $A : T^2 \rightarrow T^2$ lineáris transzformáció.

Ha $A\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) = \begin{bmatrix} a \\ b \end{bmatrix}$ és $A\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} c \\ d \end{bmatrix}$, akkor $A\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = ?$

$$A\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = A\left(\begin{bmatrix} x \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ y \end{bmatrix}\right) = A\left(\begin{bmatrix} x \\ 0 \end{bmatrix}\right) + A\left(\begin{bmatrix} 0 \\ y \end{bmatrix}\right),$$

hiszen A *összegtartó*. A *skalárszoros-tartás* miatt ez

$$\begin{aligned} A\left(x \begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) + A\left(y \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) &= xA\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) + yA\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) = \\ &= x \begin{bmatrix} a \\ b \end{bmatrix} + y \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} ax \\ bx \end{bmatrix} + \begin{bmatrix} cy \\ dy \end{bmatrix} = \begin{bmatrix} ax + cy \\ bx + dy \end{bmatrix}. \end{aligned}$$

Tehát minden vektor képét ki tudjuk számolni, ha ismerjük a, b, c, d értékét.

Definíció

A fenti A mátrixa $[A] = \begin{bmatrix} a & c \\ b & d \end{bmatrix}$. (Az oszlopok $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ és $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ képei.)

Definíció

$$\begin{bmatrix} a & c \\ b & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax + cy \\ bx + dy \end{bmatrix} \text{ (mátrix és vektor szorzata).}$$

Következmény: Vektor képének kiszámítása: $A(v) = [A]v$.

A forgatás képlete.

Példa

Az origó körüli α szögű forgatás mátrixa $[F] = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix}$.

Példa kompozícióra.

Kérdés

Melyik transzformációt kapjuk, ha először tükrözzük az $y = x$ egyenesre, majd forgatunk az origó körül 90° -kal? Ez a két transzformáció *kompozíciója* (egy más utánja).

$$\begin{bmatrix} x \\ y \end{bmatrix} \mapsto R\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} y \\ x \end{bmatrix}.$$

Mivel $[F] = \begin{bmatrix} \cos 90^\circ & -\sin 90^\circ \\ \sin 90^\circ & \cos 90^\circ \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$, ezért $\begin{bmatrix} y \\ x \end{bmatrix}$ képe

$$F\left(\begin{bmatrix} y \\ x \end{bmatrix}\right) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} = \begin{bmatrix} 0y + (-1)x \\ 1y + 0x \end{bmatrix} = \begin{bmatrix} -x \\ y \end{bmatrix}.$$

Az eredmény az y -tengelyre való tükrözés (**Házi Feladat**).

A kompozíció mátrixa.

Definíció

Ha $A, B : T^2 \rightarrow T^2$ transzformációk, akkor *kompozíciójuk* $(A \circ B)(v) = A(B(v))$ tetszőleges $v \in T^n$ esetén. *Összetett függvény:* először B -t, utána A -t alkalmazzuk.

Tétel

$$[A] = \begin{bmatrix} a & c \\ b & d \end{bmatrix}, [B] = \begin{bmatrix} a' & c' \\ b' & d' \end{bmatrix} \implies [A \circ B] = \begin{bmatrix} aa' + cb' & ac' + cd' \\ ba' + db' & bc' + dd' \end{bmatrix}.$$

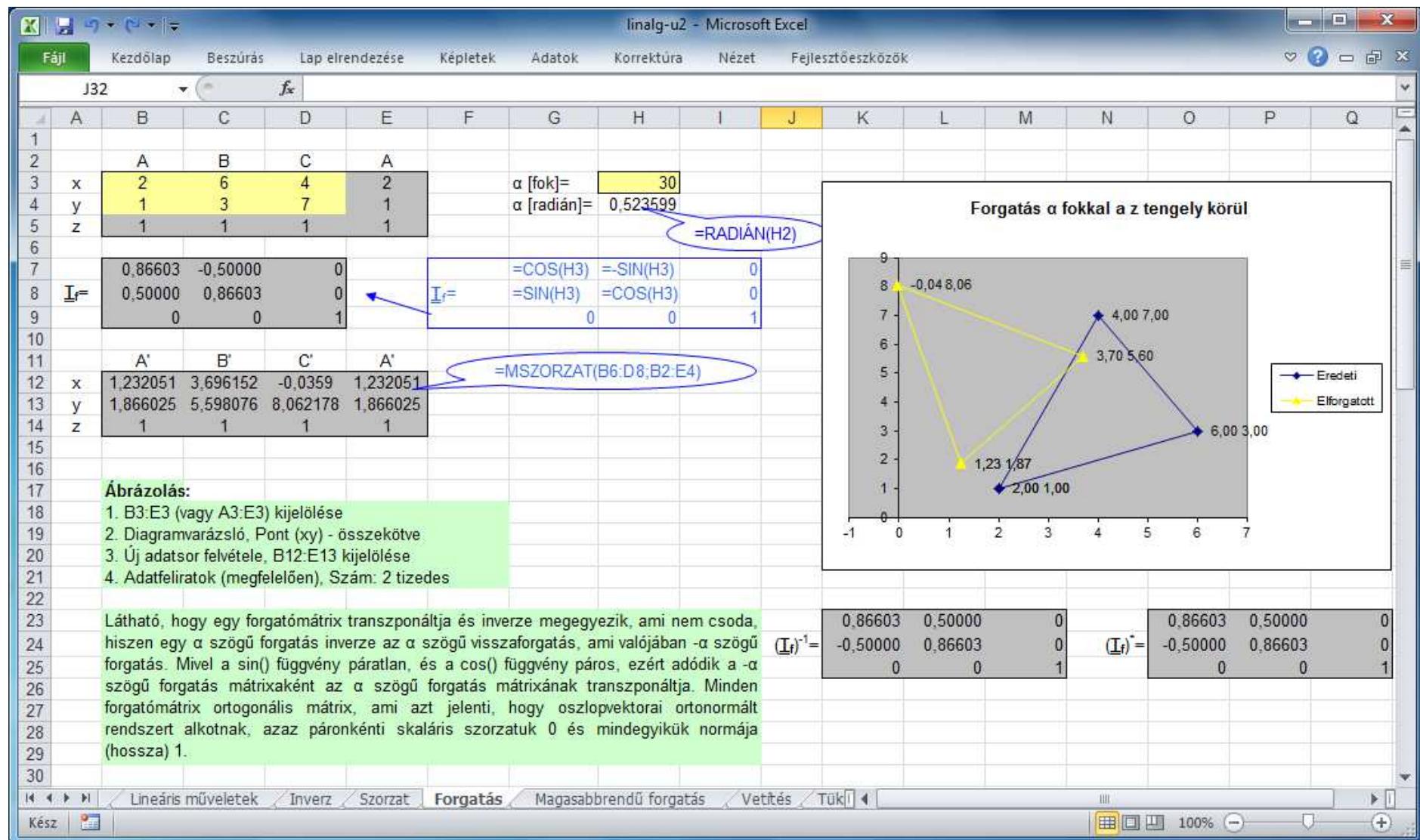
Transzformációs mátrixok

- Adott egy $n \times n$ -es transzformációs $\mathbf{A}$ mátrix, és egy megfelelő $(n \times 1)$ -es $\mathbf{x}$ vektor
 - A transzf. mátrix oszlopai a bázis egységvektorok képeit tartalmazzák (lásd: Swap)
- Az $\mathbf{x}$ vektorra alkalmazzuk az $\mathbf{A}$ transzformációt.
Ekkor az eredményvektort $\mathbf{y} = \mathbf{A}(\mathbf{x}) = \mathbf{A} \cdot \mathbf{x}$ módon számolhatjuk.
 - Megj.: egyszerre több vektorral is lehet
- Példa: Swap (csere) mátrix (ábra most: Excel; Mszorzat függvénnyel)
 - Mit kapunk, ha a transzformációt még egyszer elvégezzük?
 - Nézzük meg u.ezt az SW13 = [0 0 1; 0 1 0; 1 0 0] és az SW23 = ... mátrixokkal is!
- Az egységmátrix is transzformációs mátrix
 - Mi a megvalósított művelet?
- Alkalmazási példa: Projektormátrix (korábbi ea.)

	A	B	C	D	E	F	G	H	I	J	K	L	M
1													
2													
3		0	1	0			1	3			2	2	
4	SW12	1	0	0		Pontok	2	2		Eredmény	1	3	
5		0	0	1			3	1			3	1	
6													

Transzformációs mátrixok

- Alkalmazás: Forgatómátrixok (2D, 3D; gyak. is)





Transzformációs mátrixok

Alkalmazás: Titkosítás

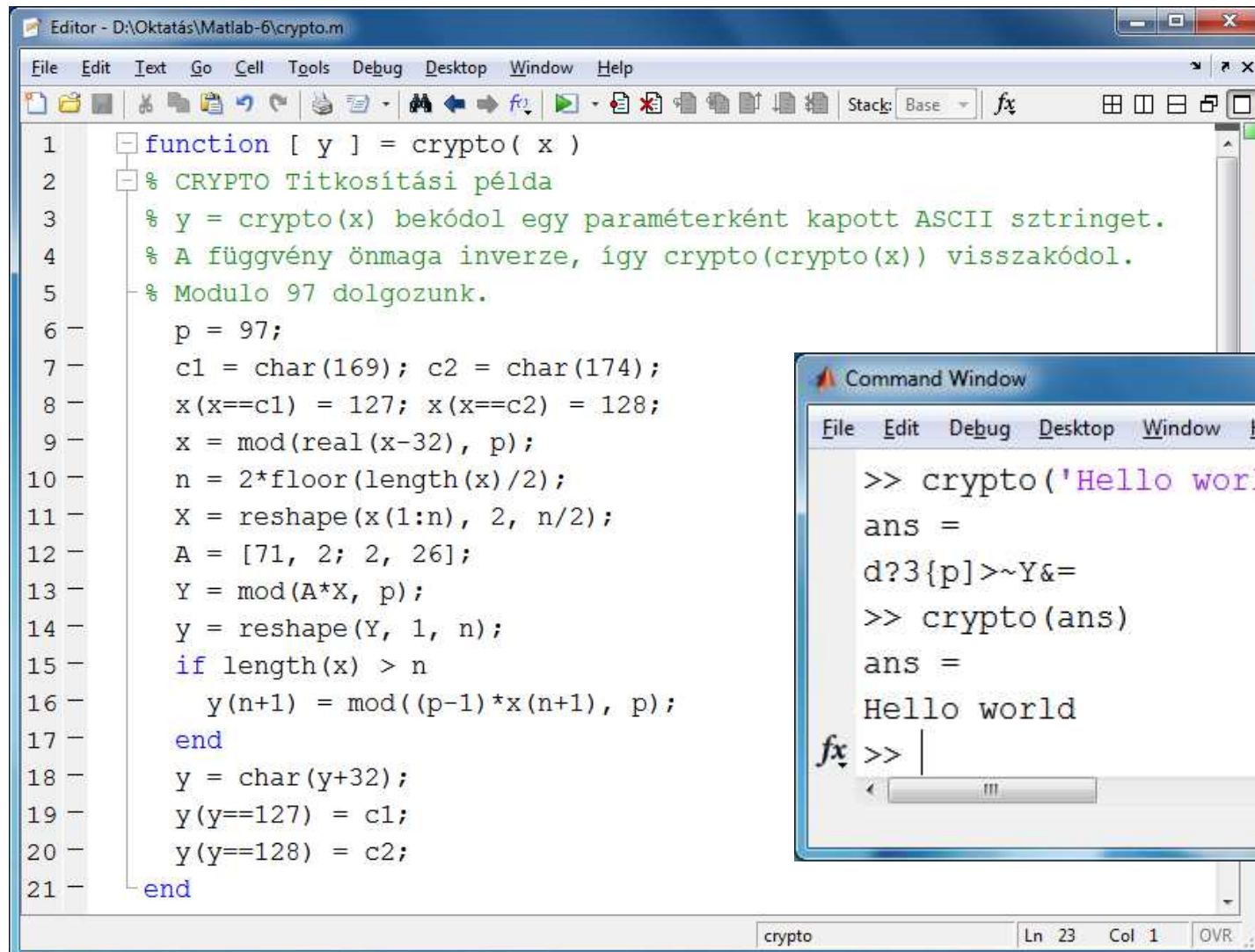
- A (be)kódolás (encoding) $y = A \cdot x \bmod p$ módon történik, ahol p alkalmas prím, A pedig olyan speciális mátrix, amelynek inverze önmaga $\bmod p$ (visszakódoláshoz, decoding)
 - Azaz A négyzete az egységmátrix
- Kódolandó karakterek: 32-től 127-ig (95 db nyomtatható jel, az utolsó a DEL)
`>> x = reshape(32:127, 32, 3)', c = char(x)`
- Legyen $p = 97$, kódolható karakterek: $0:p-1$ tartományban (+ 2 extra jel lehet)
- Jó A mátrix: $A = \begin{bmatrix} 71 & 2 \\ 2 & 26 \end{bmatrix}$
 - Ellenőrzés
`>> E = mod(A*A, p)`
- Kódolható: egy (vagy több) 2×1 -es x vektor, pl.
`>> x = 'TV', x', x = x'+0-32`
- Kódolás (eredm.: '1U')
`>> A*x, y = mod(A*x, p), char(y+32)` $\begin{pmatrix} 71 & 2 \\ 2 & 26 \end{pmatrix} \cdot \begin{pmatrix} 52 \\ 54 \end{pmatrix} \bmod 97 = \begin{pmatrix} 17 \\ 53 \end{pmatrix}$
- Visszakódolás (eredm.: 'TV')
`>> A*y, z = mod(A*y, p), char(z+32)`



Transzformációs mátrixok

Titkosítás (folyt.)

- Script, és futtatása



The screenshot shows the MATLAB environment. The Editor window displays a script named 'crypto.m' with the following code:

```
1 function [ y ] = crypto( x )
2 % CRYPTO Titkosítási példa
3 % y = crypto(x) bekódol egy paraméterként kapott ASCII sztringet.
4 % A függvény önmaga inverze, így crypto(crypto(x)) visszakódol.
5 % Modulo 97 dolgozunk.
6 p = 97;
7 c1 = char(169); c2 = char(174);
8 x(x==c1) = 127; x(x==c2) = 128;
9 x = mod(real(x-32), p);
10 n = 2*floor(length(x)/2);
11 X = reshape(x(1:n), 2, n/2);
12 A = [71, 2; 2, 26];
13 Y = mod(A*X, p);
14 y = reshape(Y, 1, n);
15 if length(x) > n
16     y(n+1) = mod((p-1)*x(n+1), p);
17 end
18 y = char(y+32);
19 y(y==127) = c1;
20 y(y==128) = c2;
21 end
```

The Command Window shows the execution of the script:

```
>> crypto('Hello world')
ans =
d?3{p}>~Y&=
>> crypto(ans)
ans =
Hello world
```

The status bar at the bottom of the Editor window indicates the current position is Ln 23, Col 1.



Vektoralgebra

- Megj.: valós koordinátájú sor- és oszlopvektorokkal dolgozunk
- Skaláris szorzat
 - Euklideszi skalárszorzat (szorzatösszeg), ahol x és y azonos méretű vektorok (nem csak a geometriai vektorok terében!)
 - Megvalósítás a Matlabban: `dot(x, y)` vagy `sum(x.*y)`
 - Példa
`>> a = [3, 4, 0], b = [-4, 3, 0], dot(a, b) % merőlegesek`
- Vektoriális szorzat
 - Vektoriális-szorzat, ahol a és b pontosan 3-elemű vektorok, eredmény az $S(a,b)$ síkra merőleges vektor lesz
 - Megvalósítás a Matlabban: `cross(a, b)`
 - Példa
`>> a = [1 0 0], b = [0 1 0], cross(a, b)`

```
Command Window
File Edit Debug Desktop Window Help
>> a = [1 0 0], b = [0 1 0], cross(a,b)
a =
     1     0     0
b =
     0     1     0
ans =
     0     0     1
fx >> |
```





Vektoralgebra

- Vegyesszorzat
 - Három térbeli vektor vegyesszorzata; a három vektor által kifeszített paralelepipedon előjeles térfogata
 - Ha ez 0, akkor a vektorok összefüggők
 - Megvalósítás a Matlabban:
`dot(cross(a, b), c)` vagy `det([a; b; c])` ill. `det([a, b, c])`
 - Példa

```
>> a = rand(1, 3), b = rand(1, 3), c = rand(1, 3),  
dot(cross(a, b), c), det([a; b; c])
```
- Vektorok hajlásszöge
 - Matlabban a `subspace(x, y)` paranccsal határozható meg, az eredmény a két **oszlopvektor** „szöge” radiánban (magasabb dimenzióban is!)
 - Sorvektorokra nem kapunk helyes eredményt
 - Példa

```
>> subspace(a', b'), subspace(a, b)
```
 - Használhatjuk ehelyett az `acos(dot(x, y)/(norm(x)*norm(y)))` kifejezést, ahol a `norm(x) = sqrt(dot(x, x))` kifejezés az euklideszi norma, azaz az x vektor „hossza”, ez sorvektorokra is működik





Lineáris egyenletrendszerek

Áttekintés

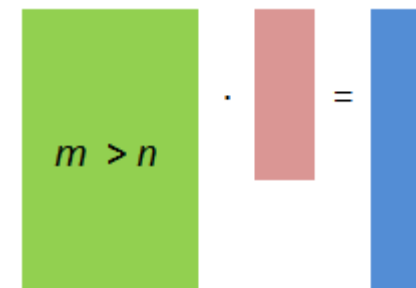
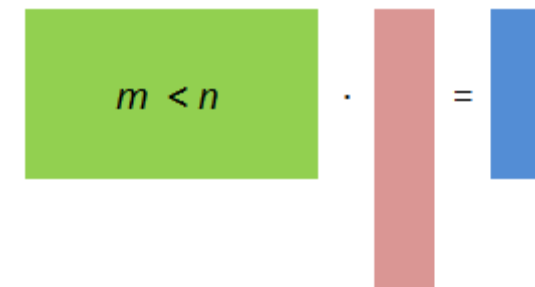
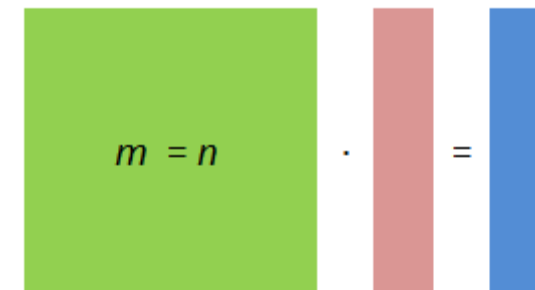
- Fontos és gyakori feladat, sokszor „előkerül”
 - Nagyon jó Matlab támogatás, kiválóan illeszkedik eddigi tanulmányainkhoz
- Rövid felírás: $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$
 - n db ismeretlen, m db egyenlet
 - Együtthatómátrix, ismeretlenek vektora, konstansvektor
- Klasszikus osztályozás
 - Négyzetes rendszer
 - A megoldások száma $\mathbf{A}$ -tól függ (Lehet 1, végtelen sok vagy 0)
 - Alulhatározott eset
 - Kevesebb egyenlet, mint ismeretlen
 - Végtelen sok megoldás
 - Túlhatározott eset
 - Több egyenlet, mint ismeretlen
 - Általában nincs megoldás
 - Leggyakrabban minimális $\mathbf{A} \cdot \mathbf{x} - \mathbf{b}$ hibavektort keresünk $\Rightarrow$ norma fogalma (vektorokra, mátrixokra)

$$a_{11} * x_1 + a_{12} * x_2 + \dots + a_{1n} * x_n = b_1$$

$$a_{21} * x_1 + a_{22} * x_2 + \dots + a_{2n} * x_n = b_2$$

...

$$a_{m1} * x_1 + a_{m2} * x_2 + \dots + a_{mn} * x_n = b_m$$



Lineáris egyenletrendszerek

Alapeset

- A független egyenletek (sorok/oszlopok) száma megegyezik az ismeretlenek számával, azaz $\det(A) \neq 0$ vagy $\text{rank}(A) == \text{length}(A)$
 - A megoldás ekkor *egyértelmű*
- Ekkor $x = A \backslash b$ vagy $x = \text{inv}(A) * b$ vagy $x = \text{pinv}(A) * b$
- Példa

```
>> A = [3 4 1; -1 2 5; 7 -2 3],  
b = [18 8 32]', rang = rank(A),  
x = A\b, A*x == b
```
- Az $A \backslash b$ megoldási módszer gyorsabb
 - Lásd: gyakorlat
 - Itt Gauss-elimináció hajtódik végre részleges főelem-kiválasztással, *sőt ha A szimmetrikus és pozitív definit, akkor a Cholesky-felbontás kerül alkalmazásra (lásd még: sugó)

```
Command Window
File Edit Debug Desktop Window Help
>> A = [3 4 1; -1 2 5; 7 -2 3]
A =
     3     4     1
    -1     2     5
     7    -2     3
>> b = [18 8 32]'
b =
    18
     8
    32
>> det(A)
ans =
    188
>> x = A\b
x =
    4.0000
    1.0000
    2.0000
>> A*x==b
ans =
     1
     1
     1
fx >>
```





Lineáris egyenletrendszerek

Alapeset (folyt.)

- Tipikus alkalmazás: az együtthatómátrixot külső fájlból olvassuk be, gyakran oly módon, hogy az A mátrix és b vektor közös táblázatban adott; ennek mérete: $n \times (n + 1)$
 - Ekkor a tényleges megoldás előtt szétdarabolás is szükséges, a már ismert módon
- Lehetőség több egyenletrendszer közös megoldására:
Ha az A eh.mátrixú egyenletrendszert több különböző b vektor esetén akarjuk megoldani, akkor ezeket egy B mátrixba foglalhatjuk:
 $AX = B \Rightarrow X = A \backslash B$
 - A mo-ok X mátrixának ugyanannyi oszlopvektora van, mint a B-nek
- Megj.: a fenti esetek persze úgy is alkalmazhatók, hogy nem egyértelmű a mo. (lásd lent)

Variable Editor - M

File Edit View Graphics Debug Desktop Window Help

Stack: Base No valid pl...

M <4x5 double>

	1	2	3	4	5	6
1	6	-6	-2	5	20	
2	-1	8	5	5	-15	
3	2	-9	-9	-3	22	
4	-4	-1	3	10	-10	
5						



Lineáris egyenletrendszerek

Alapeset – szöveges feladatok

- Tipikusan: keverési feladatok, szállítási feladatok stb.
- Megoldás a megszokott módon
 - Infók kihámozása, egyenletrendszer megadása, megoldás
 - Ellenőrzés

lin prog - Microsoft Excel

	A	B	C	D	E	F
1		Pók	Hernyó	Sárkány	Eladólány	Készlet
2	Fej	1	1	7	1	201
3	Láb	8	12	4	2	1880
4	Szem	2	1	14	3	281
5	Haj	0	5	20	333	1361
6						
7	Eredm	48	123	4	2	
8						
9	Det	-17536				

fx {=TRANSPONÁLÁS(MSZORZAT(INVERZ.MÁTRIX(

```

Command Window
File Edit Debug Desktop Window Help
>> A = [1 1 7 1; 8 12 4 2; 2 1 14 3; 0 5 20 333] ^
A =
    1     1     7     1
    8    12     4     2
    2     1    14     3
    0     5    20    333

>> det(A)
ans =
   -17536

>> b = [201 1880 281 1361]';
>> x = A\b
x =
   48.0000
  123.0000
    4.0000
    2.0000

>> A*x
ans =
   1.0e+003 *
    0.2010
    1.8800
    0.2810
    1.3610

fx >>
  
```

**Lineáris egyenletrendszerek**

$$\begin{pmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 4 \\ 6 \end{pmatrix} \quad \begin{pmatrix} 10 & -7 & 0 \\ 0 & -0,1 & 6 \\ 0 & 2,5 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 6,1 \\ 2,5 \end{pmatrix}$$

Alapeset – a megoldás részletei

- A Gauss-eliminációról általában
- Példa: a megoldandó rendszer
 - $A = [10 \ -7 \ 0; \ -3 \ 2 \ 6; \ 5 \ -1 \ 5]$, $b = [7; \ 4; \ 6]$
- 1. lépés – $0,3 \times$, ill. $0,5 \times$ első egyenl. +/-
- 2.a) lépés – egyenletcsere (pivotálás)
- 2.b) lépés – $0,04 \times$ második egyenl. +
- 3. lépés – ismeretlenek meghatározása

$$\begin{pmatrix} 10 & -7 & 0 \\ 0 & 2,5 & 5 \\ 0 & -0,1 & 6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 2,5 \\ 6,1 \end{pmatrix}$$

$$\begin{pmatrix} 10 & -7 & 0 \\ 0 & 2,5 & 5 \\ 0 & 0 & 6,2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 2,5 \\ 6,2 \end{pmatrix}$$

$$2,5x_2 + 5 \cdot 1 = 2,5 \quad x_2 = -1 \quad 10x_1 + (-7)(-1) = 7 \quad x_1 = 0 \quad x_3 = 1 \quad \begin{pmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{pmatrix} \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 7 \\ 4 \\ 6 \end{pmatrix}$$

- 4. lépés – ellenőrzés

- Bevezethetők

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 0,5 & 1 & 0 \\ -0,3 & -0,04 & 1 \end{pmatrix} \quad U = \begin{pmatrix} 10 & -7 & 0 \\ 0 & 2,5 & 5 \\ 0 & 0 & 6,2 \end{pmatrix} \quad P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

- Szorzók (L), végső együtthatók (U), pivotálás v. permutációk (P)
- Itt érvényes $L \cdot U = P \cdot A \quad L \cdot y = P \cdot b, \quad U \cdot x = y$
- Sőt, ha $P = E$, akkor $x = U \setminus (L \setminus b)$

- Ellenőrzés

>> $[L, U, P] = \text{lu}(A), \quad L^*U$

% Az lu eredetileg a LAPACK csomag része (lásd súgó)





Lineáris egyenletrendszerek

A megoldás részletei – *numerikus hibák hatása

$$\begin{pmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 4 \\ 6 \end{pmatrix}$$

- Példánkban: nincs pivotálás, ill. rosszul csináljuk

- Tfh. olyan gépen dolgozunk, amely 5 decimális jegy pontosan számol

- Az eredeti és az új rendszer

- A megoldás mindkét esetben $x = [0; -1; 1]$

$$\begin{pmatrix} 10 & -7 & 0 \\ -3 & 2,099 & 6 \\ 5 & -1 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 3,901 \\ 6 \end{pmatrix}$$

- 1. lépés – $0,3 \times$, ill. $0,5 \times$ első egyenl. +

- 2. lépés – $2500 \times$ második egyenl. +

$$\begin{pmatrix} 10 & -7 & 0 \\ 0 & -0,001 & 6 \\ 0 & 2,5 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 6,001 \\ 2,5 \end{pmatrix}$$

- 3. lépés – ismeretlenek meghatározása

- Kerekítési hibák

- $6,001 \cdot 2500 = 1,50025 \cdot 10^4$, kerekítve: $1,5002 \cdot 10^4$, + 2,5 helyett 2

- Így az egyenletünk

$$(5 + 2500 \cdot 6)x_3 = 2,5 + 2500 \cdot 6,001$$

- Ebből (x_3)

- Visszahelyettesítés

$$(5 + 1,5 \cdot 10^4)x_3 = 2 + 1,5002 \cdot 10^4$$

$$x_3 = \frac{1,5004 \cdot 10^4}{1,5005 \cdot 10^4} = 0,99993$$

- Ebből (x_2)

- Végül az első egyenletből

$$-0,001x_2 = 6 \cdot 0,99993 = 6,001$$

- Ebből (x_1)

$$x_2 = \frac{1,5 \cdot 10^{-3}}{-1 \cdot 10^{-3}} = -1,5$$

- Végül a $[-0,35; -1,5; 0,99993]$ rossz megoldást kapjuk

$$10x_1 + (-7) \cdot (-1,5) = 7$$

$$x_1 = -0,35$$





Lineáris egyenletrendszerek

Szinguláris esetek

- Ilyenkor tipikusan kevesebb fgtlen egyenletünk van, mint ismeretlen, azaz: $\det(A) == 0$ és $\text{rank}(A) < \text{length}(A)$
- Két eset különböztethető meg
 - $\text{rank}([A \ b]) > \text{rank}(A)$, azaz b nincs az A mátrix oszlopvektorai által meghatározott lin. térben. Ekkor a szinguláris eh-mátrixú egyenletrendszer *ellentmondásos*, és nincs megoldás.
 - Az A , ill. $[A \ b]$ mátrixok rangja megegyezik, de kevesebb az ismeretlenek számánál. Ekkor az egyenletrendszer *összefüggő*, és végtelen sok mo. van.
 - Azaz: egy vagy több ismeretlen szabadon megválasztható
- Mit lehet tenni? (megoldási lehetőségek)
 - Pontos megoldás nem kapható, de törekszünk arra, hogy $\|Ax - b\|$ a *lehető legközelebb legyen 0-hoz*.
Megvalósítás Matlabban: $x = \text{pinv}(A)*b$ ($x = A \backslash b$ is jó, ha működik)
 - A végtelen sok megoldás közül *egy speciálisat* meghatározunk. Ilyenkor a legrövidebb vektorhosszú mo.vektort szoktuk megadni.
Megvalósítás Matlabban: $x = \text{pinv}(A)*b$
($x = A \backslash b$ is ad(hat) mo-t, de nem a „legjobbat”)



Lineáris egyenletrendszerek

Szinguláris esetek (folyt.)

- Az esetek szétválasztása technikailag: $\text{rank}(A)$, $\text{rank}([A \ b])$
 - Példa
 $A = [3 \ 4 \ 1; -1 \ 2 \ 5; 2 \ 6 \ 6]$, $b = [18 \ 8 \ 8]'$, $\text{rank}(A)$, $\text{rank}([A \ b])$
- Másik lehetőség: az eltolt rendszer vizsgálata; pl. 3×3 -as eset:
 $\text{rank}([A(:,2:3) \ b])$ vagy $\det([A(:,2:3) \ b])$
 - Összefüggő esetben az eltolt $\det == 0$, ellentmondásos esetben nem
 - Ez a technika – $\det.$ vizsgálattal – Excelben is jól alkalmazható, persze nemcsak az első oszlop elhagyásával
- Esetszétválasztás példa és megoldás Excelben →

kész_új [kompatibilis üzemmód] - Microsoft Excel

F6: =NÉGYZETÖSSZEG(H2:H5)

	A	B	C	D	E	F	G	H	I
1								x	
2		2	-2	4	1	5		1	0
3		2	2	-2	4	8		1	-2
4		4	1	2	-4	7		1	-4
5		8	1	4	1	20		1	-6
6								4	
7									
8	det(A)=		-1,06581E-14						
9									
10	det([A(:,2:4) b])=		0						
11									

kész_új [kompatibilis...]

I9: f_x

	G	H	I	J	K
1		x		Ax - b	
2		2,136678		-8,88178E-16	
3		0,726644		0	
4		0,358131		-2,66454E-15	
5		0,747405		-3,55271E-15	
6		5,780277			
7					



Lineáris egyenletrendszerek

Szinguláris esetek (folyt.)

- A megoldás vizsgálata az összefüggő esetben

```
>> A = [3 4 1; -1 2 5; 2 6 6], b = [18 8 26]', rank(A),  
rank([A b])  
>> w = A\b, A*w, norm(w)  
% balosztóval van eredmény, de figyelmeztetéssel (warning)  
>> y = pinv(A)*b, A*y, y_norma = norm(y)  
% a kváziinverzrel kapott megoldás (a leg)rövidebb  
vektorhosszú, nincs warning
```
- A független egyenletek számának lecsökkentése a rangig

```
>> A(3,:)=[], b(3)=[], rank(A), rank([A b])  
% töröljük az utolsó egyenletet (első kettőből jön)  
>> z = A\b, A*z, z_norma = norm(z)  
% a balosztó is ad megoldást, a normája nem minimális  
>> x = pinv(A)*b, A*x, x_norma = norm(x)  
% a kváziinverzes megoldás a fent számolttal egyezik,  
normája minimális  
>> v = A'/(A*A')*b  
% ekkor a kváziinverzes megoldás így is számítható
```





Lineáris egyenletrendszerek

Túlhatározott eset

- (Logikailag tekinthetjük az ellentmondásos esethez tartozónak)
- Ilyenkor tipikusan több egyenletünk van, mint ismeretlen
 - (Pl. polinomiális interpoláció helyett alacsonyabb fokú regressziót kérünk)
- Az A mátrix rangja megegyezik az ismeretlenek számával, a kibővített $[A \ b]$ mátrix rangja pedig eggyel nagyobb
- Ekkor is azt a „megoldást” keressük, amelyre az $\|Ax - b\|$ normában mért eltérés minimális
- Megvalósítás Matlabban: $x = \text{pinv}(A)*b$ vagy $x = A \backslash b$ (ha működik)
 - Ekkor valójában a Moore-Penrose kváziinverz (pseudoinverz) kiszámítása történik
 - Érdekes, hogy ekkor a balosztó a kváziinverzet adja!
- Példa

```
>> A = [3 4 1; -1 2 5; 7 -2 3; 3 5 1], b = [18 8 32 6]'  
>> rank(A), rank([A b])  
>> x = A\b, pinv(A)*b, (A'*A)\A'*b  
% az utolsó a pseudoinverz (lásd korábban)  
>> [A*x b], tavolsag = norm(A*x-b)
```





Lineáris egyenletrendszerek

Nagyméretű lineáris egyenletrendszerek megoldási ideje – összehasonlítás

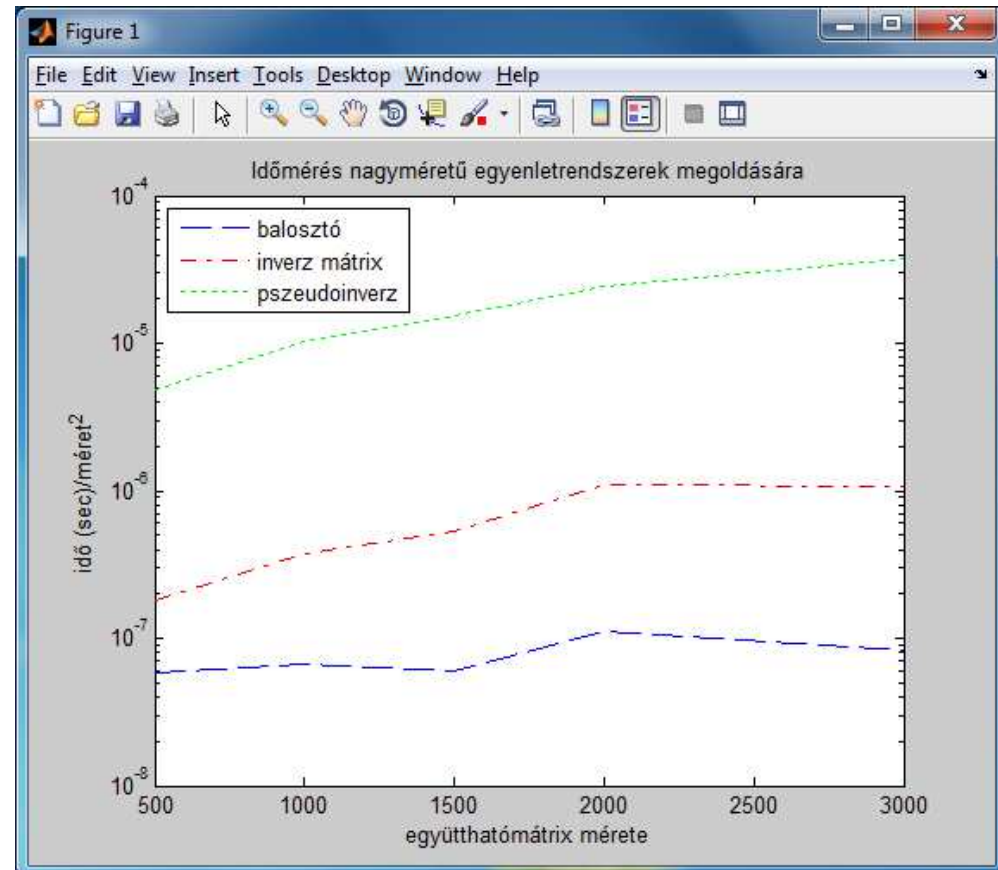
■ Időmérési példa

- Táblázat, ábra
- Teszteléshez:
Töplitz-mátrixok
- Időmérés az ismert módon
- Adatok most Excelből (vessző ...)

■ „Kulturált” (szép) ábra

- Különböző színek
- Különböző vonalstílusok (!)
- Az időadatok osztva n^2 -tel
- Fél-logaritmikus skálázás
- Jelmagyarázat
- Tengelyfeliratok

Méret/Végrehajtási idő (mp)	$A \setminus b$	$\text{inv}(A) * b$	$\text{pinv}(A) * b$
500×500	0,014702	0,044624	1,214775
1000×1000	0,066429	0,372239	10,342868
1500×1500	0,134975	1,185211	34,722225
2000×2000	0,449540	4,347214	96,005524
3000×3000	0,735710	9,694462	339,125561





Normák, kondíciósám

- A gyakorlatban a lineáris egyenletrendszer mátrixa és a jobb oldali konstansvektor általában nem pontosan ismert
 - Mérési (esetleg: leolvasási) hibák, kerekítési pontatlanságok stb.
- Fontos kérdés: a megoldás érzékenységének vizsgálata
 - Mennyire befolyásolja a megoldást, ha (kisebb mértékben) megváltoznak $\mathbf{A}$ és $\mathbf{b}$ elemei?
- Egyszerű megfontolás: az érzékenység erősen függ a konkrét rendszertől (!)
 - Ha $\mathbf{A}$ szinguláris, akkor egyes $\mathbf{b}$ vektorokra nincs megoldás, másokra végtelen sok lesz
 - Ilyenkor kis változás is nagyon jelentősen módosítja az eredményt
 - Másrészt, ha $\mathbf{A}$ közel egységmátrix, akkor $\mathbf{b}$ és $\mathbf{x}$ is közel ugyanaz
 - Ilyenkor kis változások az eredményben is csak kis változást okoznak
 - A pivotálás hatásán is érdemes elgondolkodni...
- Rajzos példa: kis változásra nem érzékeny (jól kondicionált) és kis változásra nagyon érzékeny (rosszul kondicionált) lineáris egyenletrendszer kétdimenzióban





Normák, kondíciós szám

- Egzakt mérőszám az érzékenységre: vektor- és mátrixnorma (kondíciós szám)
- Definíció: a norma egy $\|\cdot\| : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$ leképezés, amely minden vektorra (mátrixra), ill. vektorpárra (mátrixpárra) teljesíti a következő tulajdonságokat
 - Definit(ség): nemnegatív, és csak akkor 0, ha a vektor nullvektor $\|x\| > 0$, ha $x \neq 0$ és $\|0\| = 0$,
 - Homogenitás: skalárszoros normája egyenlő a vektor normájával, $\times$ a skalár abszolút értékével $\|c \cdot x\| = |c| \cdot \|x\|$, minden c skalárra,
 - Háromszög-egyenlőtlenség $\|x + y\| \leq \|x\| + \|y\|$
- Ezt a definíciót a vektorok, ill. mátrixok vektorterében többféle módon lehet teljesíteni
 - A bevezetett többféle norma azonban konvergens vektor- ill. mátrixsorozatokra ekvivalens (0-hoz tartó eltéréssorozat)
- Leggyakrabban használt vektor- ill. mátrixnormák (itt: vektorokra nézzük)
 - 1-es norma (Manhattan-norma)
 - 2-es norma (euklideszi távolság)
 - Hasonlóan: p-es (p-edik) norma
 - Végtelen norma (Csebisev-norma)
- Normák a Matlabban

```
>> x = (1:4)/5, norm1 = norm(x,1)
>> norm2 = norm(x), norminf = norm(x,inf)
```

$$\|x\|_1 = \sum_{i=1}^n |x_i|$$

$$\|x\|_2 = \left(\sum_{i=1}^n |x_i|^2 \right)^{1/2}$$

$$\|x\|_\infty = \max_i |x_i|$$





Normák, kondíciós szám

- Ha egy x vektort transzformálunk ($\mathbf{A}$ mátrix), akkor az új Ax vektor normája nagyon eltérhet x -től. Ez a változás mutatja azt az érzékenységet, amit keresünk

- A max és min itt minden $x \neq 0$ vektorra előállítandó

- Ha $\mathbf{A}$ szinguláris, akkor $m = 0$

$$M = \max \frac{\|Ax\|}{\|x\|}, \quad m = \min \frac{\|Ax\|}{\|x\|}$$

- Az M/m hányadost nevezzük $\mathbf{A}$ kondíciós számának (szemléletes jelentés: kb. ennyiszeres hibát okozhat a zavar)

- A pontos érték (persze) függ a használt vektornormától, de elsősorban a nagyságrend fontos

- Mivel $M \geq m$, ezért nyilvánvaló, hogy $\kappa(\mathbf{A}) \geq 1$

- Speciális mátrixok kondíciószáma

- $\kappa(\mathbf{E}) = 1$, $\kappa(c\mathbf{A}) = \kappa(\mathbf{A})$

- Diagonális mátrixokra: a $|\max|$ és $|\min|$ elemek hányadosa

$$\kappa(\mathbf{A}) = \frac{\max \frac{\|Ax\|}{\|x\|}}{\min \frac{\|Ax\|}{\|x\|}}$$

- A kondíciós szám mátrixnormával is bevezethető

- Ekkor

- Azaz

$$\|\mathbf{A}\| = \max \frac{\|Ax\|}{\|x\|}, \quad \|\mathbf{A}^{-1}\| = \frac{1}{m}$$

- A Matlab beépített függvényei a számolásra

- `cond(A) == cond(A, 2)`

- `cond(A, 1)`, `cond(A, inf)`, `rcond(A)` (becslés $1/\text{cond}(A, 1)$ -re)

$$\kappa(\mathbf{A}) = \|\mathbf{A}\| \|\mathbf{A}^{-1}\|$$

- Példa

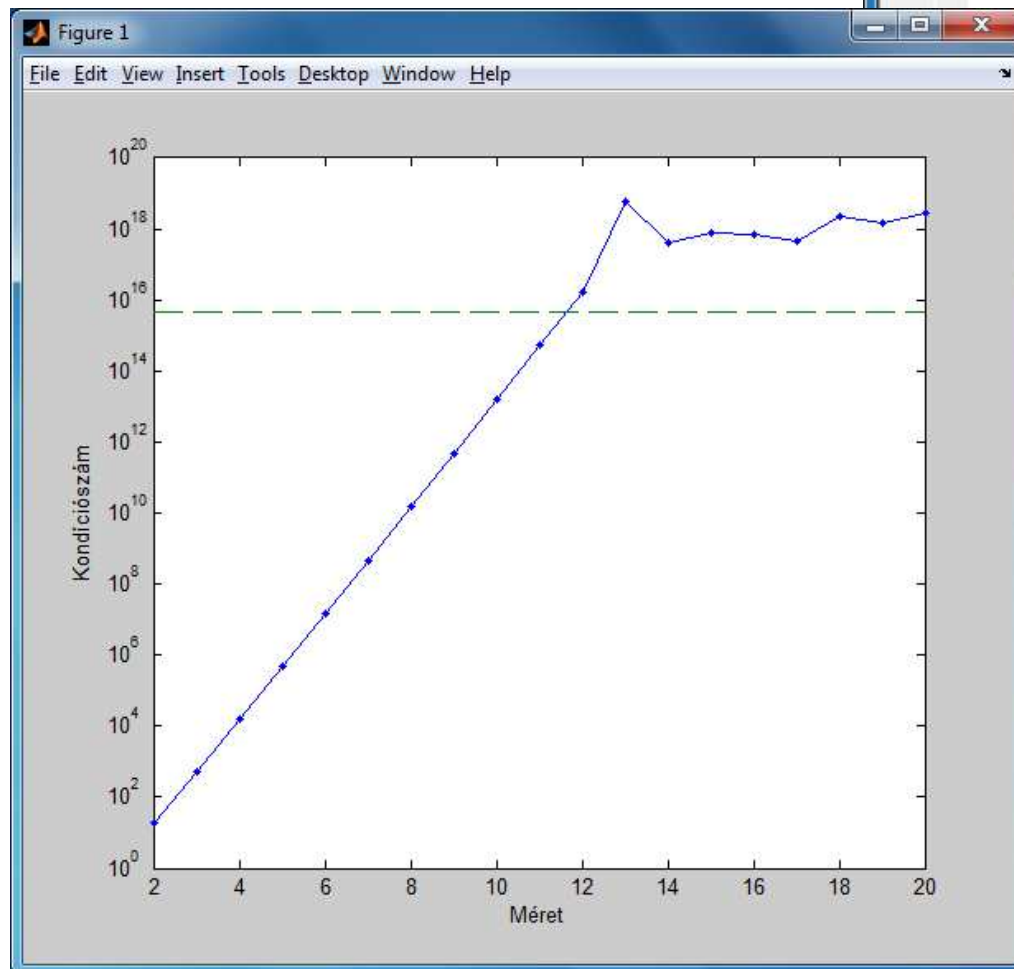
```
>> A=[4 5 6; 5 6 8; 1 7 7], norm(A)*norm(inv(A)), cond(A)
```

```
>> 1/(norm(A,1)*norm(inv(A),1)), rcond(A)
```



Kondíciósám

- Eml.: A Hilbert-mátrixok gyengén kondicionáltak (nagy cond értékek)
- Hilbert-mátrixok sorozatát vizsgáljuk (2-től 20-ig)
 - 12 felett pontatlan az eredmény



```
Editor - D:\Oktatás\Matlab-6\hilbert_kond.m
File Edit Text Go Cell Tools Debug Desktop Window Help
1 nmax = 20; inveps(1:nmax) = 1/eps;
2 H = hilb(nmax);
3 for n = 2:nmax
4     cond_h(n) = cond(H(1:n, 1:n));
end
semilogy(2:n, cond_h(2:n), '.-', ...
          2:n, inveps(2:n), '--')
xlabel('Méret')
ylabel('Kondíciósám')
```



Sajátérték, sajátvektor

- Egy λ konstanst és egy $x \neq 0$ vektort az **A** négyzetes mátrixhoz tartozó sajátértéknek (eigenvalue) és sajátvektornak (eigenvector) nevezzük, ha kielégítik az $A \cdot x = \lambda \cdot x$ egyenletet
 - (Ebben az esetben A csak nyújt)
 - Pontosan n db sajátérték van, és mindegyikhez legalább egy sajátvektor tartozik
 - Lehetnek (köztük) komplexek!
- *Egy σ konstanst és az $u, v \neq 0$ vektort az **A** négyzetes mátrixhoz tartozó szinguláris értéknek és szinguláris vektoroknak nevezünk, ha kielégítik az $A \cdot v = \sigma \cdot u$, és $A' \cdot u = \sigma \cdot v$ egyenleteket
 - Itt A' a szokásos konjugált-transzponált
- Megvalósítás Matlabban: eig (sajátérték), ill. svd (szing. érték) parancs
- *A kondíciószám felírható a legnagyobb és a legkisebb szinguláris érték hányadosaként is, és a szinguláris értékek pont $A'A$ sajátértékei
 - Példa

```
>> cond(A)
>> sqrt(max(abs(eig(A'*A))))/sqrt(min(abs(eig(A'*A))))
>> max(svd(A))/min(svd(A))
```
- A sajátérték/s.vektor sok gyakorlati mérnöki feladat megoldásánál fontos szerepet játszik
 - Pl.: híd terheléspróbája, megnyúlások alapján rezonancia-frekvenciák számítása





Sajátérték, sajátvektor

- Egy A négyzetes mátrix sajátértékeit a *karakterisztikus polinomjának* a zérushelyei adják
 - Ezek a $\det(A - \lambda \cdot E) = 0$ megoldásával kaphatók meg
 - A sajátérték-egyenlet: $(A - \lambda \cdot E) \cdot x = 0, x \neq 0$
 - Komplex értékeket is kaphatunk (valós szimmetrikus esetben nem)
- Egyszerű megoldás a Matlabban: *poly*, ill. a *roots* parancsok
 - A *poly(A)* hívás egy sorvektort ad, elemei a karakterisztikus polinom eh.i
 - A *roots* hívás a gyököket szolgáltatja
- Példa

```
>> A = [4 5 6; 5 6 8; 1 7 7], a = poly(A)
>> lambda = roots(a), poly(lambda)
```
- A sajátért. probléma megoldásra beépített „kincstári” függvény az *eig*
- Az $[U \text{ Lambda}] = \text{eig}(A)$ hívásnál az U mátrix oszlopvektorai az 1-re normált sajátvektorokat adják, a Lambda mátrix főátlóbeli elemei pedig a sajátértékeket
- Példa

```
>> [U Lambda] = eig(A), A*U, U*Lambda
>> U*Lambda*inv(U) % A-t kapjuk (sajátérték-felbontás)
% Nem szinguláris szimmetrikus mátrixra U*Lambda*U' is A-t adja
```





Sajátérték, sajátvektor

- Sajátérték-felbontás (eigenvalue decomposition)
 - Legyenek $\lambda_1, \lambda_2, \dots, \lambda_n$ az A mátrix sajátértékei, legyenek $x_1, x_2, \dots, x_n$ hozzájuk tartozó sajátvektorok
 - Helyezzük el a sajátértékeket az adott sorrendben egy Λ mátrix főátlójában, a sajátvektorokat pedig egy X mátrixban (a j -edik oszlopban legyen x_j)
 - Ekkor $A \cdot X = X \cdot \Lambda$
 - Ha a sajátvektorok fgtlen rdsz-t alkotnak (ez nem mindig igaz), akkor létezik X^{-1} , azaz $A = X \cdot \Lambda \cdot X^{-1}$ (sajátérték-felbontás)
 - *Ilyenkor A sok tulajdonsága Λ -val vizsgálható, pl. $A^n = X \cdot \Lambda^n \cdot X^{-1}$ teljesül
- *Szinguláris felbontás (singular value decomposition, SVD)
 - Mint az előbb, írjuk mátrixokba a szinguláris értékeket (Σ – csak a főátlóban vannak nem 0 elemek), és a megfelelő szinguláris vektorokat (U és V)
 - Ekkor $A \cdot V = U \cdot \Sigma$ és $A' \cdot U = V \cdot \Sigma'$
 - Itt U és V teljesíti: $U' \cdot U = E$, $V' \cdot V = E$ (ortogonálisak)
 - Egy mátrix ortogonális, ha az inverze megegyezik a transzponáltjával (lásd: linalg. öf.)
 - Ebből $A = U \cdot \Sigma \cdot V'$
 - A szinguláris felbontás nemcsak négyzetes A mátrixokra értelmezhető
- Megvalósítás a Matlabban: *eig* és *svd* parancsok



Sajátérték, sajátvektor

Példa (kis négyzetes mátrixra)

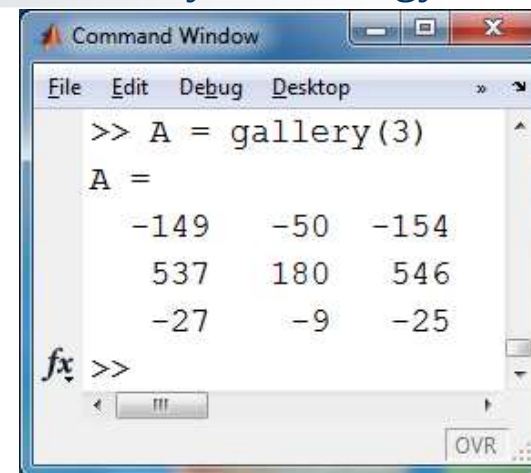
- Legyen $A = \text{gallery}(3)$
- Ekkor $\det(A - \lambda \cdot E) = \lambda^3 - 6\lambda^2 + 11\lambda - 6 = (\lambda - 1)(\lambda - 2)(\lambda - 3)$
- Így tehát a sajátértékek: 1, 2, 3

```
>> A = gallery(3)  
>> a = poly(A), lambda = roots(a) % karakt. polinommal  
>> B = sym(A), syms l, det(B-l*eye(3)) % szimbolikusan
```
- Sajátvektorok

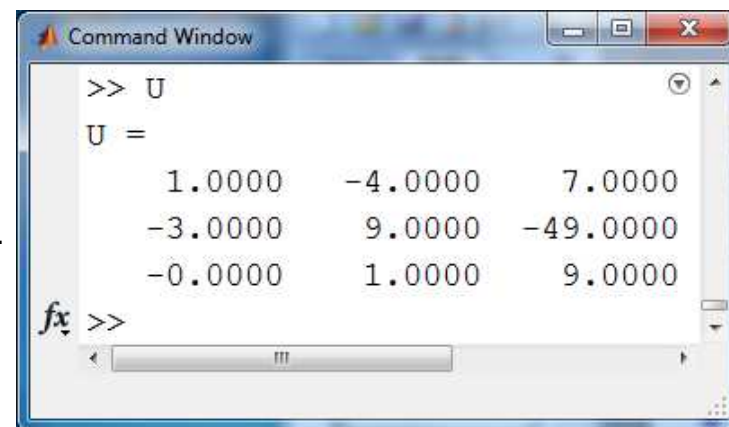
```
>> [U L] = eig(A)
```
- U módosítható úgy, hogy (1,1) és (3,2) eleme 1 legyen, (3,3) eleme 9 legyen
 - Így egész koordinátájú sajátvektorok, $\det(U) = 1$
- Ekkor U inverze is egész elemeket tartalmaz, és $A = U \cdot L \cdot U^{-1}$

```
>> inv(U), U*L*inv(U)
```
- Szinguláris felbontás

```
>> [X, S, V] = svd(A)  
>> X*S*V' % A szinguláris értékek között nagy a különbség!
```



```
Command Window  
File Edit Debug Desktop  
>> A = gallery(3)  
A =  
-149    -50   -154  
 537    180    546  
 -27     -9    -25  
fx >>
```



```
Command Window  
>> U  
U =  
 1.0000   -4.0000   7.0000  
 -3.0000   9.0000  -49.0000  
 -0.0000   1.0000   9.0000  
fx >>
```

Sajátérték, sajátvektor

*Érzékenységi elemzés

- A sajátértékek érzékenysége a sajátvektor mátrix kondíciós számával becsülhető

- Példa 1.

```
>> A = gallery(3)
>> [U L] = eig(A), cond(U)
% Itt még pontos választ kapunk
```

- Példa 2.

```
>> A = gallery(5)
>> B = sym(A), det(B-lambda*eye(5))
% Ebből lambda_i = 0, pontos szimbolikus eredmény
>> [U L] = eig(A)
% Numerikus hibával terhelt eredmény
>> cond(U) % gyengén kondicionált
>> e = eig(A)
>> plot(real(e), imag(e), 'r*', 0, 0, 'ko')
>> axis(0.1*[-1 1 -1 1]), axis square
% az ötszög alakzat hiba
```

