MATLAB 2024
4. téma

Lineáris egyenletrendszerek, spektrális elemzés, leképezések

Lineáris egyenletrendszerek

$$\left|\begin{array}{l} ax+by+cz=d \\ ex+fy+gz=h \\ ix+jy+kz=l \end{array}\right| \longrightarrow \left|\begin{array}{l} a_{11}x_1+a_{12}x_2+a_{13}x_3=b_1 \\ a_{21}x_1+a_{22}x_2+a_{23}x_3=b_2 \\ a_{31}x_1+a_{32}x_2+a_{33}x_3=b_3 \end{array}\right|$$

$$\underline{\boldsymbol{A}}\,\underline{\boldsymbol{x}}=\underline{\boldsymbol{b}}$$

$$\underline{\boldsymbol{A}}^{-1}\underline{\boldsymbol{A}}\,\underline{\boldsymbol{x}}=\underline{\boldsymbol{A}}^{-1}\underline{\boldsymbol{b}}$$

$$\boxed{\underline{\boldsymbol{x}}=\underline{\boldsymbol{A}}^{-1}\underline{\boldsymbol{b}}}$$

Bal és jobb osztás, inverz

- A mátrixok körében a szorzás nem kommutatív, ezért kétféle osztás definiálható:
  - bal osztás: `B\A` azt az `X` mátrixot jelenti, amelyre `B*X=A` azaz **`B\A`** $= B^{-1}A = X$
  - jobb osztás: `A/B` azt az `Y` mátrixot jelenti, amelyre `Y*B=A` azaz **`A/B`** $= AB^{-1} = Y$
- MATLAB-ban van beépített inverz számoló: `inv(...)`, de ha kifejezetten lineáris egyenletrendszert akarunk megoldani, akkor sokkal célszerűbb a ‘bal osztás’ operátorával dolgoznunk:

`A*x = b`

~~`x = inv(A)*b`~~

`x = A\b`

miért? *Gyorsabb, pontosabb* (Gauss-eliminációt használ az inverz létrehozása nélkül). MATLAB help-ben: `mldivide` (\)

Lineáris leképezések

- minden mátrix egy lineáris leképezésnek tekinthető;
- a gyakorlaton csak 2D leképezésekről lesz szó, pl: adott egy téglatest koordinátáival, amit nagyítsunk fel az y-tengely mentén

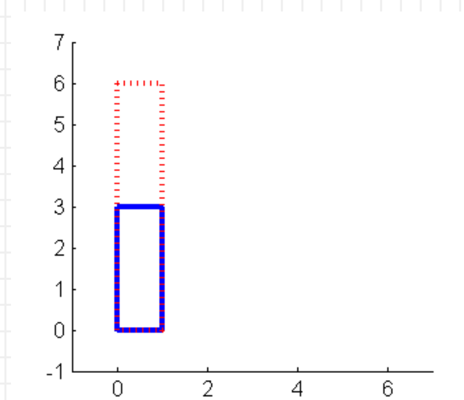
```
A = [1 0; 0 2];
vonal = [0 3; 1 3;...
    1 0; 0 0; 0 3];
vonal2 = vonal*A;
```

```
figure;
hold on;
plot(vonal(:, 1), vonal(:, 2),
'LineWidth', 3);
plot(vonal2(:, 1), vonal2(:, 2),
'r:', 'LineWidth', 3);
xlim([-1 7]);
ylim([-1 7]);
```

0	6
1	6
1	0
0	0
0	6

0	3
1	3
1	0
0	0
0	3

1	0
0	2

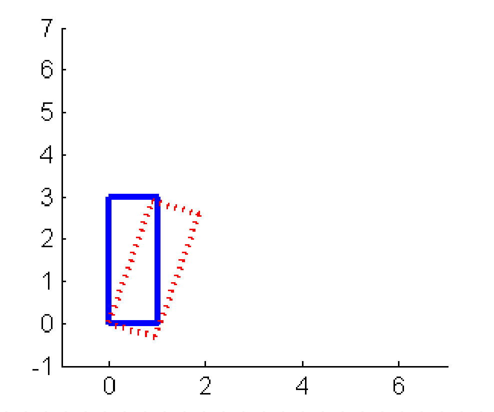


2D forgatás

- 2D esetben forgatásról a rajzolási síkra merőleges (általában Z) tengely körül beszélhetünk;
- forgatási transzformációs mátrix (**ϕ** szöggel):

$$\underline{\boldsymbol{R}}(\phi)=\begin{bmatrix}\cos(\phi) & -\sin(\phi)\\ \sin(\phi) & \cos(\phi)\end{bmatrix}$$

```
angInRad = 0.3;
B = [cos(angInRad), -sin(angInRad);...
      sin(angInRad), cos(angInRad)];
vonal3 = vonal*B;
```



Sajátérték, sajátvektor

Egy **A** mátrix **λ** sajátértékei és **v** jobb oldali sajátvektorai a

$$\boldsymbol{\lambda v} = \boldsymbol{Av}$$

lineáris egyenletrendszer megoldásai. Ugyanígy az **u** bal oldali sajátvektorok a $\boldsymbol{\lambda u} = \boldsymbol{uA}$

lineáris egyenletrendszer megoldásai.

```
e = eig(A);
D = eig(A, 'matrix');
```

- **eig - sajátérték, és sajátvektor számolása**
  - egy visszatérési értékkel: paraméterként átadott mátrix sajátértékeit tartalmazó oszlopvektor (második paraméterként opcionálisan megadhatjuk, hogy mátrix formában adja vissza a sajátvektorokat, és ekkor diagonális mátrixot kapunk)
  - két visszatérési értékkel:
    - V mátrix oszlopai: a (normált hosszúságú) jobb oldali sajátvektorok
    - D mátrix: a megfelelő sajátértékeket a főátlójában tartalmazó diagonális mátrix `[V,D] = eig(A);`
  - három visszatérési értékkel:
    - U mátrix oszlopai a megfelelő (normált hosszúságú) bal oldali sajátvektorok

```
[V,D,U] = eig(A);
```

Komplex sajátértékek és sajátvektorok

- A legtöbb mátrix sajátértékei ill. sajátvektorainak elemei között van komplex. Egy komplex szám valós és képzetes részét a **real** és **imag** függvények segítségével kaphatjuk meg.

```
real([3 + 5j 2; 4 + 0j 6 + 1j])
imag(3 + 5j) % -> ans = 5
```

3	2
4	6

- Egy adott skalárról vagy mátrixról az **isreal** függvény segítségével dönthetjük el, hogy valós szám(okat tartalmaz) (nem elemenként működik!).

```
isreal([3 + 0j 4 + 1j]) % -> ans = 0
```

Feladatok

- [P-ITMAT-0014] Bevezetés a Matlab programozásba
  - https://moodle.ppke.hu/course/view.php?id=1514
  - 04 Feladatok → Gyümölcsök
    → Transzformáció
    → Egyenletrendszer
    → Áramkör
    → Korcsoporteloszlás
    → Sztochasztikus