

Mátrixot a következőképpen adhatunk meg:

- Az elemeket szögletes zárójelek között soroljuk fel;
- a sorokat szóközzel vagy vesszővel, az oszlopokat pontosvesszővel választjuk el.

Pl. az

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

mátrix megadása:

» A = [1 2 3; 4 5 6; 7 8 9]

A képernyőn ezt a következő formában látjuk:

A =

1 2 3

4 5 6

7 8 9

A mátrixot kiírathatjuk a nevének begépelésével. Bármelyik elemére is rákérdezhetünk, pl. A(2,3) a második sor harmadik elemét jelenti. Bármelyik mátrixelemnek megváltoztathatjuk az értékét. Ha pl. a harmadik sor harmadik elemét 9-ről 0-ra szeretnénk cserélni, akkor a következőt írjuk be:

A(3,3)=0

Az eredmény:

A =

1 2 3

4 5 6

7 8 0

Mátrixot is transzponálhatunk az aposztróf jellel. Vigyázat: Ha az A mátrixnak komplex elemei is vannak, akkor a vessző nemcsak transzponálja a mátrixot, hanem konjugálja is az elemeket. Ennek elkerülésére a vessző elé írunk pontot: A.'

Nagyobb méretű vektorok vagy mátrixok megadásánál nem érdemes kézzel beírni minden egyes elemet, vannak egyéb lehetőségek is. Tegyük fel, hogy egy 51 elemű vektort szeretnénk készíteni, amelyben az elemek 0-tól 5-ig következnek egymás után 0,1 lépésközzönként. Ezt a

```
» x = 0:0.1:5
```

sor beírásával tehetjük meg a legegyszerűbben.

Olyan vektor megadásához, amelynek a szomszédos elemei egyenlő távolságra vannak egymástól, használható a `linspace` parancs is. Az `y = linspace(a,b)` parancs beírására egy 100 elemű `y` sorvektor keletkezik, amelynek első eleme `a`, és utolsó eleme `b` (tehát a végpontok is beleszámítanak a 100-ba). Ha nem 100 osztáspontra van szükségünk, hanem `n`-re, akkor `y = linspace(a,b,n)` használható. Pl.

```
» theta = linspace(0,2*pi,101)
```

a $[0, 2\pi]$ intervallumot 100 egyenlő részre darabolja, és így egy 101 elemű vektort készít.

Rézmátrixok készítése

A kettőspont arra is használható, hogy segítségével egy mátrixból rézmátrixokat vegyünk ki. Az `A(m:n,k:l)` az `A` mátrix azon elemeiből áll, amelyek az `m`-edik és `n`-edik sor, valamint a `k`-adik és `l`-edik oszlop között helyezkednek el. `A(m,:)` az `A` `m`-edik sorát, `A(:,m)` az `A` `m`-edik oszlopát oszlopát adja meg. `A(:,2:3)` az `A` mátrix második és harmadik oszlopából álló rézmátrixot jelenti. `A(end,:)` az `A` mátrix utolsó sorát, `A(:,end)` pedig az utolsó oszlopát adja.

Pl.

```
» A=[1 2 3;4 5 6;7 8 9]
```

Készítsük el `B` néven `A`-nak azt a rézmátrixát, amely a második és harmadik sorában valamint a első és második oszlopában lévő elemeket tartalmazza:

```
» B = A([2 3],[1 2])
```

`B =`

```
4 5
```

```
7 8
```

Sorokat ill. oszlopokat fel is cserélhetünk, pl.

```
» C = A([2 1 3],:)
```

```
C =
```

```
4 5 6
```

```
1 2 3
```

```
7 8 9
```

Egy mátrixból vektort készíthetünk az

```
» A(:)
```

paranccsal, amely oszlopvektort készít az A mátrix egymás utáni oszlopaiból. Az eredménye tehát:

```
ans =
```

```
1
```

```
4
```

```
7
```

```
2
```

```
5
```

```
8
```

```
3
```

```
6
```

```
9
```

Próbáljuk ki az

```
» A(end:-1:1,end)
```

 parancsot is!

Egy mátrixból törölhetünk sort vagy oszlopot is az [] üres vektor alkalmazásával:

```
» A(3,:) = []
```

```
A =
```

```
1 2 3
4 5 6
```

Itt a harmadik sort töröltük. Visszaállítani a következőképpen lehet:

```
» A = [A(1,:);A(2,:);7 8 9]
```

vagy egyszerűen

```
» A = [A;7 8 9]
```

Ha nem vagyunk benne biztosak, hogy hány sora ill. oszlopa van egy mátrixnak, akkor erre a `size` paranccsal kérdezhetünk rá.

```
» size(A)
ans =
3 3
```

Ez azt jelenti, hogy 3 sora és három oszlopa van az A mátrixnak. (Az első a sorok száma.) Az

```
» [m,n]=size(A)
```

sor beírására a sorok száma belekerül az m, az oszlopok száma az n nevű változóba.

Mátrixok összefűzése

Mátrixokból mint blokkokból nagyobb méretű mátrixokat is összerakhatunk. Pl.

```
» A = [1 2 3; 4 5 6; 7 8 9]

» B = [A 10*A; -A [1 0 0; 0 1 0; 0 0 1]]
B =
1 2 3 10 20 30
4 5 6 40 50 60
7 8 9 70 80 90
-1 -2 -3 1 0 0
-4 -5 -6 0 1 0
-7 -8 -9 0 0 1
```

Speciális mátrixok

A Matlabban léteznek utasítások speciális mátrixok generálására. Ezek:

- `eye(n)`: $n \times n$ -es identitásmátrix;
- `eye(m,n)`: $m \times n$ -es mátrix 1-esekkel a „főátlóban”, a többi helyen nullával;
- `zeros(m,n)`: $m \times n$ -es nullmátrix;
- `ones(m,n)`: csupa 1 elemű $m \times n$ -es mátrix;
- `rand(m,n)`: $m \times n$ -es véletlen elemű mátrix;

Hasznos még a `diag(A)` parancs, ez az A mátrix főátlójában lévő elemekből vektort készít. Ha pedig v egy vektor, akkor `diag(v)` olyan diagonális mátrixot készít, amelynek a főátlójában v elemei vannak. További speciális mátrixokról a `help elmat` parancsra kaphatunk felvilágosítást.