Mátrixműveletek SCILAB programcsomag segítségével

A Scilab mátrixalapú programcsomag, így minden változót mátrixnak tekint. Először a mátrixok és a vektorok (amelyek tulajdonképpen speciális mátrixok) megadásának legegyszerűbb módjait ismertetjük.

1. Mátrix (vektor) megadása

A mátrix megadása legegyszerűbben úgy végezhető, hogy szögletes zárójelek " [" és "] " közé soronként beírjuk a mátrix elemeit úgy, hogy az egyes elemeket szóközzel választjuk el egymástól, a mátrix sorának végére érve pontosvesszőt (;) teszünk, majd hasonlóan végezzük a többi sor beírását. A sorvektor és az oszlopvektor is egy-egy speciális mátrix, amelynek egy sora ill. egy oszlopa van. Egyébként a skalárt is mátrixként kezeli a Scilab, de ennél nem használjuk a szögletes zárójeleket.

Az $A = \begin{bmatrix} 6 & 8 & 2 \\ 4 & 1 & 1 \end{bmatrix}$ mátrix megadása a következő módon történik:

–>A=[6 8 2; 4 1 1]
A =
6. 8. 2.
4. 1. 1.

Az $y = (2, 5)$ vektor sorvektorként történő bevitele a következő módon történik:

–>y=[2 5]
y =
2. 5.

Az $x = (2, 3, -1)$ vektor oszlopvektorként történő bevitele a következő módon történik:

–>x=[2;3;-1]
x =
2.
3.
- 1.

Mátrixokat megadhatunk beépített függvényekkel is, ilyenek a zeros(), ones(), eye(), rand() függvények, amelyekkel rendre zérus mátrixot, csupa 1-es számot tartalmazó mátrixot, egységmátrixot és megadott eloszlású, véletlenszerű számokat tartalmazó mátrixot hozhatunk létre. Egy 3x3 méretű egységmátrixot az alábbi módon adhatunk meg:

–>eye(3,3)
ans =
1. 0. 0.
0. 1. 0.
0. 0. 1.

A mátrix elemeire kerek zárójellel lehet hivatkozni. Az A mátrix i-edik sorának j-edik elemére való hivatkozás: A(i,j) például

–>A(3,1)
ans =
4.

Fontosnak tartjuk még a számsorozat képzését, amelyet kettőspont (:) operátor használatával valósíthatunk meg a következő módon:

ke:nov:be vagy **[ke:nov:be]**

ahol "ke" a sorozat kezdő eleme, "be" a befejező eleme, "nov" pedig a növekmény, pl:

–>-2:0.5:2
ans =
- 2. - 1.5 - 1. - 0.5 0. 0.5 1. 1.5 2.

–>sorozat=[-2:0.5:2]
sorozat =

- 2. - 1.5 - 1. - 0.5 0. 0.5 1. 1.5 2.

A kapott számsorozat tehát egy sorvektor.

Megjegyezzük, hogy a mátrix sorának elemeit helyköz helyett vesszővel (,) is elválaszthatjuk. Az is lehetséges, hogy minden sor után pontosvessző helyett ENTER-t ütünk. Lehet vegyesen is használni a szóközt és a vesszőt, mint ahogy az alábbi példák mutatják. Javasoljuk az olvasónak, hogy a számára leginkább kényelmes adatmegadás mellett döntsön, a keverés kevésbé átlátható.

–>B=[2,3 5
–>6 1,0]
B =
2. 3. 5.
6. 1. 0.

–>c=[2
–>4;6]
c =
2.
4.
6.

2. Mátrix transzponálása

A mátrix transzponálását az aposztróf (') jel segítségével végezhetjük. Ezzel lehet például sorvektorból oszlopvektort (vagy fordítva) készíteni, hasznos művelet akkor is, ha egy sorozatot oszlopvektorként akarunk használni.

–>B=[2,3,5
–>6,1,0]
B =
2. 3. 5.
6. 1. 0.

–>B'
ans =
2. 6.
3. 1.
5. 0.

–>sorozat'
ans =
- 2.
- 1.5
- 1.
- 0.5
0.
0.5
1.
1.5
2.

3. Mátrix-mátrix szorzása

Az alábbiakban a 2x3 méretű C mátrixnak és a 3x3 méretű D mátrixnak a CD szorzatát számítjuk ki. A két mátrix az adott sorrendben összeszororható, mert csatlakozók, az első helyen álló C mátrix oszlopmérete megegyezik a második helyen álló D mátrix sorméretével. Az eredménymátrix sormérete egyenlő lesz az első helyen álló C mátrix sorméretével, oszlopmérete pedig a második helyen álló D mátrix oszlopméretével lesz egyenlő. A szorzási művelet jele a csillag (*) jel, amelyet a két mátrix közé teszünk:

–>C=[2,3,5;4,-1,1]

C =
2. 3. 5.
4. - 1. 1.

–>D=[5,2,-2;3,0,1;4,2,-1]
D =
5. 2. - 2.
3. 0. 1.
4. 2. - 1.

–>C*D
ans =
39. 14. - 6.
21. 10. - 10.

4. Mátrix-vektor szorzása

Tekintsük az előző C mátrixot és szorozzuk be jobbról a korábban megadott $x = (2, 3, -1)$ oszlopvektorral, majd szorozzuk be balról a korábban megadott $y = (2, 5)$ sorvektorral:

A Cx művelet elvégzése:
–>C*x
ans =
8.
4.

Az yC művelet elvégzése:
–>y*C
ans =
24. 1. 15.

5. Vektorok skaláris (sorvektor*oszlopvektor) és diadikus szorzata (oszlopvektor*sorvektor)

–>s=[2;3;-1]
s =
2.
3.
- 1.

–>r=[1,5,2]
r =
1. 5. 2.

–>t=[3,4]
t =
3. 4.

–>r*s
ans =
15.

–>s*t
ans =
6. 8.
9. 12.
- 3. - 4.

–>s'*r'
ans =
15.

–>t'*r
ans =
3. 15. 6.
4. 20. 8.

6. Mátrix determinánsának meghatározása
A mátrix determinánsának kiszámítására a **det()** függvény szolgál.

–>G=[2,3,4
–>2,-1,5
–>1,-2,4]
G =
2. 3. 4.
2. - 1. 5.
1. - 2. 4.

–>det(G)
ans =
- 9.

7. Mátrix inverzének meghatározása

A mátrix inverzének meghatározását a Scilab beépített **inv()** nevű függvényével végezhetjük el. Legyen az F mátrix az alábbi utasítással megadva:

–>F=[1,2,3;2,5,6;3,4,10]
F =
1. 2. 3.
2. 5. 6.
3. 4. 10.

–>G=inv(F)
G =
26. - 8. - 3.
- 2. 1. 0.
- 7. 2. 1.

8. Lineáris egyenletrendszer megoldása

Legyen a megoldandó lineáris egyenletrendszer az alábbi

$$\begin{aligned} x_1 + 4x_2 + 3x_3 &= 3 \\ 2x_1 + \ x_2 - 4x_3 &= 9 \\ x_1 + \ x_2 + \ x_3 &= 2 \end{aligned}$$

Először az együtthatómátrix elemeit és a jobboldalt kell megadni egy mátrixba ill. egy oszlopvektorba. Legyen a mátrix azonosítója A, az oszlopvektoré pedig b.
–>A=[1 4 3;2 1 -4;1 1 1]
A =
1. 4. 3.
2. 1. - 4.
1. 1. 1.

–>b=[3;9;2]
b =
3.
9.
2.

A lineáris egyenletrendszer megoldását kétféleképpen is végezhetjük, első mód az $x = A^{-1}b$ ismert formula alkalmazása, a második mód pedig a Scilab lineáris egyenletrendszert megoldó beépített függvényének az alkalmazása. A beépített függvény neve: **linsolve()**, amelynek két paramétere van, az első az együtthatómátrix (A), a második pedig a jobboldali vektor (-1) -szerese ($-b$), ugyanis ez a függvény az $Ax + b = 0$ alakú lineáris egyenletrendszert oldja meg.

```
–>x=inv(A)*b
x =
2.
1.
- 1.

–>y=linsolve(A,-b)
y =
2.
1.
- 1.
```