

4. GYAKORLAT

MÁTRIXFÜGGVÉNYEK, SAJÁT FÜGGVÉNYEK, GRAFIKA 1.

NEVEZETES MÁTRIXOK (FOLYT. A MÚLT ÓRÁRÓL)

hilb(n) – $n \times n$ méretű Hilbert-mátrix, amelynek elemei a természetes számok reciprocai a következő szabály szerint:

$$h(i, j) = 1/(i + j - 1)$$

A törtalakot (karakterláncként) a `rats(hilb(n))` szolgáltatja. A Hilbert-mátrixok determinánsai egyre kisebbek! (Gyorsan csökkenő sorozat...)

invhilb(n) – a Hilbert-mátrix inverze, amelynek elemei egyre nagyobb egészeket tartalmaznak.

```
>> H3 = hilb(3), d3 = det(H3), T3 = rats(H3)
```

Feladatok

Állítsuk elő az 5×5 -ös Hilbert-mátrixot! Mennyi a 3. oszlop elemeinek az összege?

Melyik Hilbert mátrixnak lesz először kisebb a determinánsa, mint az *eps* érték?

Állapítsuk meg az `hilb(4)`, `hilb(5)`, `hilb(6)` mátrixok inverzeinek legnagyobb elemét!

(Tipp: a `max` függvény hasonlóan működik, mint a `sum`.)

pascal(n) – a binomiális együtthatókból képezett mátrix, amelynek anti-átlóiban az $(a + b)$ egyre növekvő egészhatványainak együttható-sorozata van.

Feladatok

Állapítsuk meg az $(a + b)^7$ kifejtésében az $a^5 b^2$ tag együtthatóját!

Melyik a Pascal-háromszög 8. sorának 2. legnagyobb eleme?

(Itt a hagyományos háromszöges elrendezést követjük, a háromszög csúcsában található egy darab 1-es elemet 0. sornak tekintjük.)

Feladat

Építsük fel az egységmátrix, a csupa nulla mátrix, a csupa egyes mátrix, a Pascal-mátrix, a Hilbert-mátrix és egyszerű mátrixműveletek (transzponálás is megengedett) segítségével a következő mátrixokat:

W =

2	0	0	1	1
0	2	0	1	1
0	0	2	1	1
1	1	1	0	0
1	2	3	0	0
1	3	6	0	0

Z =

1.0000	0.5000	0.3333	0.2500	1.0000
0.5000	0.3333	0.2500	0.2000	1.0000
0.3333	0.2500	0.2000	0.1667	1.0000
0.2500	0.2000	0.1667	0.1429	1.0000
0	0.5000	0	0	0
0	0	0.5000	0	0
0	0	0	0.5000	0
0	0	0	0	0.5000

magic(n) – mágikus négyzet, ahol n legalább 3 és az elemek mind különbözők 1-től n^2 -ig. Az elemek összege a sorokban, oszlopokban, a főátlóban és az antiátlóban megegyezik.

Feladat: Mennyi a 4×4-es mágikus mátrix egy oszlopának elemösszege?

MÁTRIXFÜGGVÉNYEK

A mátrixmanipulációs függvények (tükrözés, forgatás, eltolás) látványos lehetőségeket adnak a múlt órán megismerteken túl további származtatott mátrixok előállítására, ill. újabb, bővített „építkezésre”.

Az általunk felhasznált „repertoár”: flipud, fliplr, rot90, circshift; továbbá a diag, a tril és a triu.

Feladat

Hozzunk létre olyan 5×5-ös méretű mátrixot, amely az antiátlójában csupa 5-ös elemet tartalmaz, a többi pozícióban pedig 0 áll.

Feladat

Készítsük el az 5×5-ös mágikus mátrixot!

Forgassuk el 90 fokkal az óramutató járásával ellentétesen, majd transzponáljuk!

Adjuk meg a 3. sor 2. elemét!

Igazoljuk kétféle módon is, hogy a mágikus mátrix sor- és oszlopösszege megegyezik.

Kidolgozott mintafeladat

Rakjuk össze a speciális mátrixfüggvények segítségével előállított almátrixokból a következő mátrixot!

0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Megoldás

A mátrixunk 4 blokkra bontható, ezek mérete 3×3, 3×4; 4×3, 4×4.

A nem négyzetes blokkok zérusmátrixok, a négyzetes mátrixoknak pedig csak az anti-diagonálisa tartalmaz egyeseket. Ez utóbbiakat az egységmátrix tükrözésével, vagy forgatásával állíthatjuk elő. A klasszikus összerakós megoldás:

```
>> A = [flipud(eye(3)), zeros(3,4); zeros(4,3), rot90(eye(4))]
```

Egy másik megoldás:

```
>> AE4 = rot90(eye(4)); A(1:3,1:3) = AE4(2:4,1:3); A(4:7,4:7) = AE4
```

Itt felhasználtuk azt, hogy a célmátrix fel nem töltött elemei automatikusan nullák lesznek. Még rövidebben (mátrixmanipulációs függvényekkel):

```
>> A = flipud(circshift(eye(7),4))
% A 7x7 egységmátrix sorait 4-el ciklikusan előre toltuk, majd
% tükröztük, vagy:
>> A = circshift(flipud(eye(7)),3)
% először tükrözzük, azután shiftelünk
```

Feladatok

1. Oldjuk meg a kidolgozott mintafeladatot úgy is, hogy nem az antiátlókban, hanem a főátlókban vannak az egyes elemek.
2. Állítsuk elő a következő (még nem létező) mátrixot elemeire történő értékadásokkal! (lehetőleg a legkevesebbel)

0	10	0	0	0	0	0	0	0	0
0	0	10	0	0	0	0	0	0	0
0	0	0	10	0	0	0	0	0	0
0	0	0	0	10	0	0	0	0	0
0	0	0	0	0	10	0	0	0	0
0	0	0	0	0	0	10	0	0	0
0	0	0	0	0	0	0	10	0	0
0	0	0	0	0	0	0	0	10	0
0	0	0	0	0	0	0	0	0	10
0	0	0	0	0	0	0	0	0	0

Segítség:

a./ clear A; A(1:9, 2:10) = 10*eye(9) az utolsó sor nélküli eredményt adja. Hogyan jöhet létre az utolsó sor?

b./ Shifteljük az egységmátrix 10-szeresét függőlegesen negatív irányba, vagy vízszintesen pozitív irányba, és a felesleges elemet nullázzuk ki!

SÁVMÁTRIXOK, ALSÓ- ÉS FELSŐHÁROMSZÖG MÁTRIXOK

A diag parancs felhasználásával kiemelhetjük egy mátrix főátlóját vagy valamelyik mellékátlóját, ill. vektor felhasználásával diagonális mátrixot készíthetünk (az adott vektor a főátlóba vagy valamelyik mellékátlóba elhelyezhető).

Feladat

Legyen $D = [1:4; 11:14; 21:24; 31:34]$.

Vegyük ki D főátlóját, majd készítsük el azt a mátrixot, amely csak D főátlójának elemeit tartalmazza, és a többi eleme 0.

Mit eredményez a $\text{diag}(D, -1)$ és a $\text{diag}(\text{diag}(D, -1), -1)$ kifejezés?

Az előző mintára állítsuk elő azt a kifejezést, amelyik olyan mátrixot generál, amelyben csak a D mátrix felső mellékátlójának elemei nem 0 értékűek!

Feladat

Ellenőrizzük, hogy a $\text{diag}(\text{diag}(D, -1), -1) + \text{diag}(\text{diag}(D, 1), 1) + \text{diag}(\text{diag}(D))$ kifejezés a D mátrix olyan sávmátrixú részét emeli ki, amelyben csak a főátló és a két első mellékátló elemei szerepelnek.

Mit állít elő a $\text{diag}(\text{diag}(D, -1), 1) + \text{diag}(\text{diag}(D, 1), -1) + \text{diag}(\text{diag}(D))$ kifejezés?

Feladat

Állítsunk elő a diag és a ones függvények használatával egy olyan 6×6 -os mátrixot, amelynek főátlója csupa 8-as, felső mellékátlójában minden elem 3-as és az alsó mellékátlóban 1-esek vannak! (Vigyázat: az első mellékátlókban csak 5-5 elem van!)

A triu és a tril parancs felső- ill. alsóháromszög mátrixot készít az alapmátrixból.

Feladat

Állítsuk elő D alsó- és felsőháromszög mátrixát!

Mit eredményez a $\text{triu}(D, 1)$ és a $\text{tril}(D, -1)$ kifejezés?

Ez alapján állítsuk elő D -t oly módon, hogy összerakjuk (most: mátrixösszeadással!) a főátlójából, az alsó- és a felsőháromszög mátrixából!

A PLOT UTASÍTÁS

A plot utasítás a legegyszerűbb esetben (x, y) pontpárok összekötött megjelenítésére szolgál (a pontok koordinátáit vektorok tartalmazzák). A szintaktika: $\text{plot}(x, y)$.

Feladat

Ábrázoljuk a $[0, 0]$ és $[1, 1]$ pontok által meghatározott szakaszt!

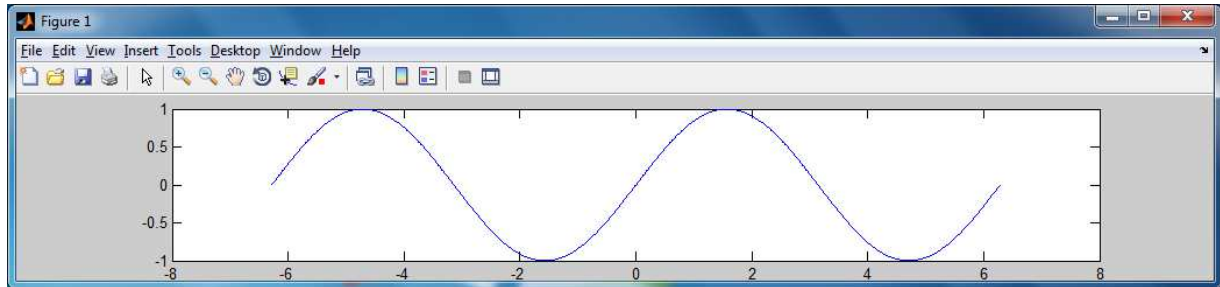
Először az alapértelmezett színt használjuk, utána legyen zöld, majd fekete a vonal.

Függvényábra készítésénél úgy indulunk el, hogy egy vektorba legyártjuk az alappontokat (linspace parancs vagy $:$ operátor), majd erre húzzuk rá a függvényt.

Példa

Rajzoljuk ki a $\sin(x)$ függvény grafikonjának pontjait a $[-2\pi, 2\pi]$ intervallumban 1001 pont segítségével!

```
>> x = linspace(-2*pi, 2*pi, 1001); plot(x, sin(x))
```



Az alappontok megfelelően sűrű előállítására kulcsfontosságú, anélkül a grafikonunk nem lesz korrekt. **F:** Nézzük meg, hogy mi történik, ha az x sorozat csak 11 elemű!

Több rajz egy ábrán a hold on/off parancsokkal jeleníthető meg. A hold on kiadása után minden ábra egymásra kerül mindaddig, amíg a hold off parancsot ki nem adjuk.

Feladat

Ismételjük meg az előző két grafikon kirajzolását, de most már egy közös ábrán! A vonalak színe legyen különböző (pl. piros és kék)! (Próbáljuk ki a vonalstílus megváltoztatását is.)

Tipp: hold on és hold off között gyártunk le a megfelelő x vektorokat (pl. x1 és x2 néven), és adjuk ki a rajzoló utasításokat.

Ismételjük meg az előző kirajzoló utasításokat úgy, hogy a vonalvastagságot is változtatjuk, és a vonalszínt az RGB skálán állítjuk be.

```
>> x1 = linspace(...); plot(x1,sin(x1),'LineWidth',1,'Color',[1 0 0]);
```

Megjegyezzük, hogy a hold utasítás nélkül is lehet több grafikont egy ábrára tenni. A szintaktika: plot(x1, y1, string1, x2, y2, string2, ...), ahol a string1 pl. 'r' lehet. Próbáljuk ki!

OTTHONI MUNKA

Feladatok (nevezetes mátrixok)

a./ Előkészítésként nézzük meg a súgóban a chol parancs leírását!

Határozzuk meg a 6×6-os Pascal-mátrix Cholesky-felbontását (felső háromszög mátrix, U).

Ellenőrizzük, hogy U^*U valóban kiadja a Pascal-mátrixot!

b./ Határozzuk meg a pascal(5) és a pascal(6) mátrixok determinánsát! Milyen következtetés adódik ebből az inverz mátrixok elemeire? (Ellenőrizzük!)

Feladat

Készítsünk egyedi tervezésű mátrixokat a kidolgozott mintafeladathoz hasonlóan a Matlab speciális mátrixfüggvényeinek felhasználásával.

Próbáljunk ki olyan építési feladatokat is, hogy az összerakott mátrix valamely részmátrixát felülírjuk egy új mátrixdarabbal. Pl.

```
W =  
    2    0    0    1    1  
    0    2    0    1    1  
    0    0    2    1    1  
    1    1    1    0    0  
    1    2    3    0    0  
    1    3    6    0    0
```

Felülírással módosítva:

```
W =  
    2.0000    0    0    1.0000    1.0000  
    0    1.0000    0.5000    0.3333    1.0000  
    0    0.5000    0.3333    0.2500    1.0000  
    1.0000    0.3333    0.2500    0.2000    0  
    1.0000    2.0000    3.0000    0    0  
    1.0000    3.0000    6.0000    0    0
```

Feladat (üres mátrix/vektor)

Milyen méretű lesz az $A = [\text{eye}(3); 6:3]$ mátrix?

Próbáljuk ki a következő előállításokat, és hasonlítsuk össze a létrejövő objektumok méreteit!

```
>> c = [10:0], isempty(c), c_meret = size(c), c_hossz = length(c)  
>> d = [1:0]', isempty(d), d_meret = size(d), d_hossz = length(d)  
>> e = [], isempty(e), e_meret = size(e), e_hossz = length(e)
```

Feladat

Tekintsük a jegyzetben szereplő, ill. a gyakorlat könyvtárában adott tridiag.m fájlt, amely egy speciális sávmátrixot állít elő (főátló + első mellékátlók). Az .m fájl segítségével adjuk meg egyetlen Matlab-utasítással azt a 20×20 -as mátrixot, amelynek

- bal felső 10-szer 10-es blokkja olyan tridiagonális mátrix, amelynek főátlójában -2 -esek, két mellékátlójában pedig 1 -esek állnak (tridiag[1, -2, 1] típusú);
- jobb alsó 10-szer 10-es blokkja tridiag[2, -4, 2] típusú;
- a többi eleme pedig 0.

