

6. GYAKORLAT

LINEÁRIS ALGEBRAI ALKALMAZÁSOK

Többféle lineáris algebrai alkalmazást már korábban tanultunk és gyakoroltunk (pl. transzformációs mátrixok, lineáris egyenletrendszerek).

Most néhány további olyan témakört érintünk, amelyek a mérnöki gyakorlatban fontosak lehetnek.

SPECIÁLIS VEKTORSZORZATOK

Eml. (előadás): Matlabban a skaláris szorzat, a vektoriális szorzat és a vegyesszorzat meghatározására léteznek megfelelő beépített parancsok (*dot*, *cross* és ezek kombinációja). Két oszlopvektor hajlásszöge a *subspace* függvénnyel számolható.

Feladat

Hány fokos szöget zárnak be az $a = [1 \ 0]$ és $b = [1 \ 1]$ vektorok?

Tipp: Használjuk először a *subspace* függvényt. A visszatérési értéket radiánban kapjuk meg. Próbáljuk ki az arkuszkoszinuszos képletet is!

Ellenőrizzük, hogy sorvektorokra nem kapunk helyes eredményt!

Feladat

Határozzuk meg az $a = [1, 3, 2]$, $b = [-1, 5, 2]$ vektorokra merőleges olyan c vektort, amelynek a hossza éppen 2!

Tipp: Az $a \times b$ vektor nyilván merőleges a -ra és b -re (*cross* fv.). Azt kell még biztosítani, hogy a szorzatvektor hossza 2 legyen. Ehhez használjuk a (kettes, euklideszi) normát.

NORMÁK, KONDÍCIÓSZÁM

Eml. (előadás): A vektor- és mátrixnormák többek között lineáris egyenletrendszerek megoldásának érzékenységi vizsgálatára használhatók, de a hagyományos euklideszi távolság is egy spec. norma. Idézzük fel önállóan a norma tulajdonságait!

Feladat

Készítsünk egy x sorvektort, amely 0-tól 1-ig lépdél 0,1-es lépésközzel. Határozzuk meg x egyes, kettes és végtelen normáját (*norm* parancs). A képletek alapján (előadás) ellenőrizzük – ha kell, megfelelő számolással –, hogy valóban a várt eredményt kaptuk.

Feladat

Ellenőrizzük az $A = [1 \ 2; 3 \ 4]$ és $B = [2 \ 3; 4 \ 5]$ mátrixok felhasználásával a következő mátrixnorma tulajdonságokat (az 1-es norma az oszlopösszegnorma, a végtelen norma a sorösszegnorma), ill. még a háromszög-egyenlőtlenséget is:

$$\|A \cdot B\| \leq \|A\| \|B\|$$

$$\|A\|_1 = \max_j \sum_{i=1}^m |a_{ij}|$$

$$\|A\|_\infty = \max_i \sum_{j=1}^n |a_{ij}|$$

Feladat

Legyen $A = \text{eye}(10) \cdot 0.01$, majd készítsünk egy olyan B mátrixot, amelynek főátlójában az 1..10 sorozat szerepel. Mennyi A és B determinánsa, rangja?

Mekkora a kondíciós számuk?

Hogyan készítenénk ez alapján olyan mátrixot, amelynek nagy a kondíciós száma (azaz gyengén kondicionált)?

Tipp: Diagonális mátrixokra a kondíciós szám a $|\max|$ és $|\min|$ elemek hányadosa.

(Egység mátrixokra ez pont 1.)

Feladat

A Hilbert mátrixok gyengén kondicionált mátrixok. Tekintsük a következő egyenletrendszert:

$H \cdot x = b$, ahol $H = \text{hilb}(4)$, $b = [13/6, 7/6, 49/60, 19/30]'$.

Ennek elméletileg pontos megoldása, amiről behelyettesítéssel meg is győződhetünk:

$x = [2, -1, 2, 0]'$.

Módosítsuk a H mátrixot úgy, hogy a 2. sor 1. elemét 1 ezrelékkal megnöveljük. Mennyi lesz a megoldásvektor elemeinek abszolút-értékbeli változása? (Mennyi a legnagyobb változás?)

Tipp: Az egyenletrendszerek megoldását a $H \backslash b$ formula segítségével állítsuk elő!

SAJÁTÉRTÉK, SAJÁTVEKTOR

Eml. (ea.): A sajátérték feladat legegyszerűbb megfogalmazása az, hogy az $A \cdot x = \lambda \cdot x$ probléma megoldásait keressük (amikor a mátrix csak nyújt). Négyzetes mátrix sajátértékeit a karakterisztikus polinomjának ($\det(A - \lambda \cdot E) = 0$ megoldásával) a zérushelyei adják. Matlabban a `poly(A)` hívás egy sorvektort ad, elemei a karakterisztikus polinom együtthatói.

Mintafeladat

Határozzuk meg az $A = [5 \ 1 \ -2 \ 1; 1 \ 6 \ -1 \ 1; -2 \ -1 \ 4 \ 2; 1 \ 1 \ 2 \ 8]$ szimmetrikus mátrix sajátvektorait és sajátértékeit! Ellenőrizzük a sajátértékegyenlet teljesülését is!

>> [U L] = eig(A), ellenorzes = A*U - U*L

Határozzuk meg az A mátrix karakterisztikus polinomját! Helyettesítsük be a sajátértékeket a karakterisztikus polinomba, és így ellenőrizzük, hogy ezek valóban kielégítik a karakterisztikus egyenletet!

Tipp: A `polyval` parancs polinom kiértékelésére szolgál, adott helyen/helyeken.

>> p = poly(A), l = eig(A), polyval(p, l)

Feladat

Legyen $B = [6, -1; 8, 2]$. Valósak-e a B mátrix sajátértékei?

Tipp: Használjuk a megoldáshoz a *poly* és a *roots* párost, ill. az *eig* parancsot!

Az *eig* parancs segítségével készítsük el a mátrixhoz az U és Λ mátrixokat úgy, hogy U oszlopvektorai az 1-re normált sajátvektorokat adják, a Λ mátrix főátlóbeli elemei pedig a sajátértékeket. Ellenőrizzük, hogy ekkor $U \cdot \Lambda \cdot \text{inv}(U)$ az eredeti mátrixot adja!

Feladat

Határozzuk meg a szokásos módon az $A = [-4, -3, -7; 2, 3, 2; 4, 2, 7]$ mátrix sajátvektorait és sajátértékeit! (S és L mátrixok.)

a.) Igaz-e, hogy az $S^{-1} \cdot A \cdot S$ mátrix diagonálmátrix és a főátlójában éppen a sajátértékeket tartalmazza?

b.) A sajátértékeknek a karakterisztikus polinomba való behelyettesítésével ellenőrizzük, hogy a sajátértékek kielégítik a karakterisztikus egyenletet!

Feladat

Valós szimmetrikus mátrix minden sajátértéke valós és a sajátvektorainak mátrixa ortogonális mátrix, azaz bármely két különböző sajátvektor skaláris szorzata nulla. Ellenőrizzük ezt az $A = [-3, 1, -1; 1, -5, 1; -1, 1, -3]$ mátrixra!

OTTHONI MUNKA

Részben kidolgozott mintafeladat (LU felbontás, Gauss-elimináció)

Legyen $A = [1 \ 1 \ 1; 1 \ -1 \ 1; 1 \ 1 \ -1]$ és $b = [6; 2; 0]$. Ekkor az $Ax = b$ egyenletrendszer egyértelműen megoldható, és a megoldás $x = [1; 2; 3]$.

A Matlab beépített *lu* parancsa előállítja az eredeti A mátrix olyan felbontását, amelyre $L \cdot U = A$, és U tartalmazza a Gauss-elimináció során kapott végső együtthatókat, L pedig a megoldás során alkalmazott szorzókat.

Ellenőrizzük a Matlab segítségével, hogy akkor is ugyanezek az együtthatók és szorzók jönnek ki, ha mi oldjuk meg az egyenletrendszert.

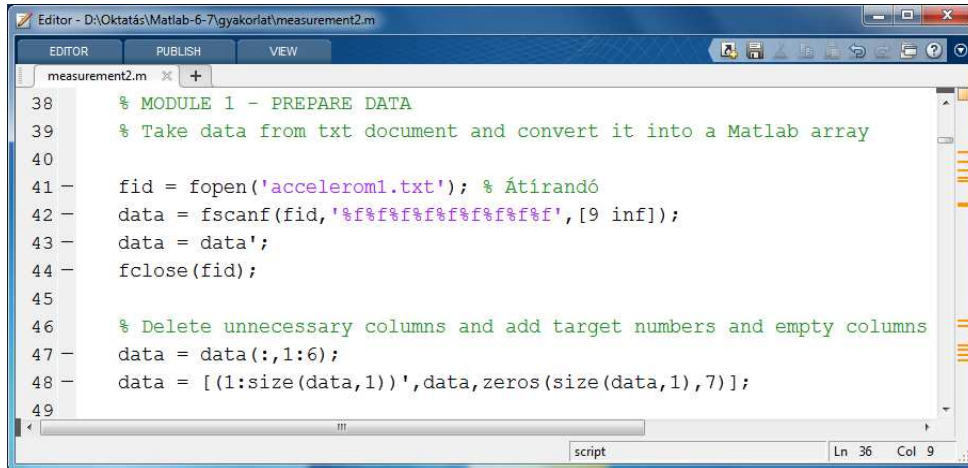
```
>> [L, U] = lu(A), L*U % LU felbontás
>> A_uj = A; A_uj(2,:) = A_uj(2,:)-A_uj(1,:)
>> b_uj = b; b_uj(2,:) = b_uj(2,:)-b_uj(1,:) % 2. sor
% Hasonlóan a 3. sorra ...
```

Feladat (Audis mérési feladat része)

Az előadáson bemutatott script a Laser Handyscannerrel mért adatok feldolgozására szolgál.

A mérésnél kocka alakú gyorsulásmérőkön (kb. 2 cm-es oldalúak) mérési *pontokat* helyeztünk el, egy műszeren 3 db-ot. Emellett még minden gyorsulásmérő közvetlen szomszédságában is van még egy mérési pont az alapsíkon (és esetleg máshol is lehetnek ilyenek).

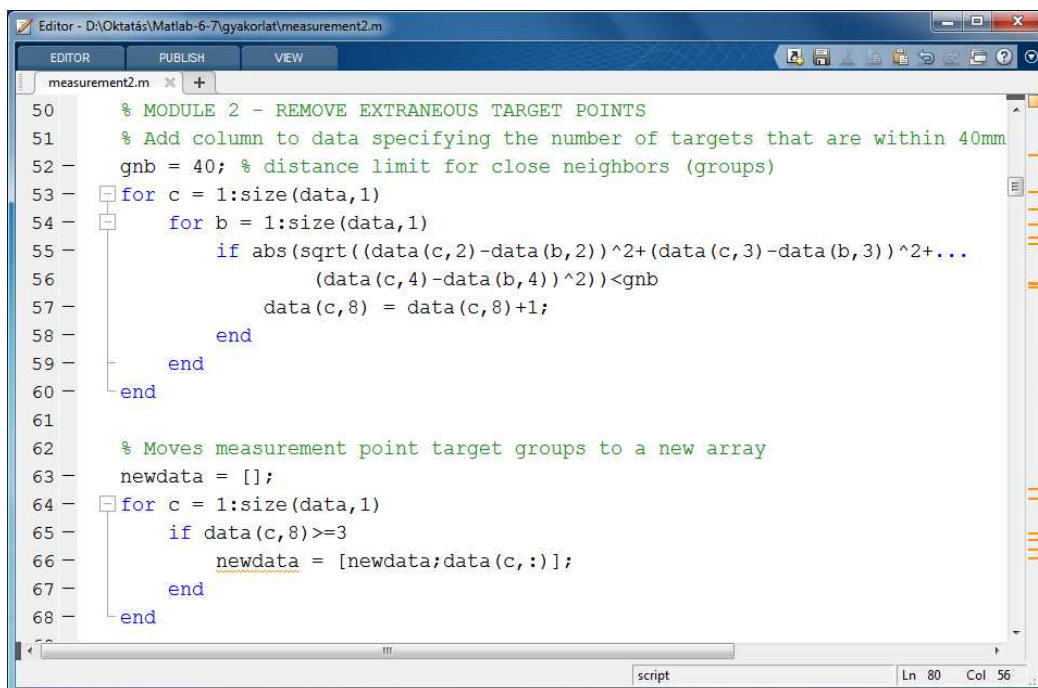
A feldolgozás első lépése a mérési pontokhoz tartozó adatok beolvasása *fájlból egy adattömbbe* (data), majd a felesleges oszlopok törlése (az első 6 oszlop marad meg; térbeli koordináták és normálvektor komponensek).



```
38 % MODULE 1 - PREPARE DATA
39 % Take data from txt document and convert it into a Matlab array
40
41 fid = fopen('acceleroml.txt'); % Átírandó
42 data = fscanf(fid, '%f%f%f%f%f%f', [9 inf]);
43 data = data';
44 fclose(fid);
45
46 % Delete unnecessary columns and add target numbers and empty columns
47 data = data(:,1:6);
48 data = [(1:size(data,1))', data, zeros(size(data,1),7)];
49
```

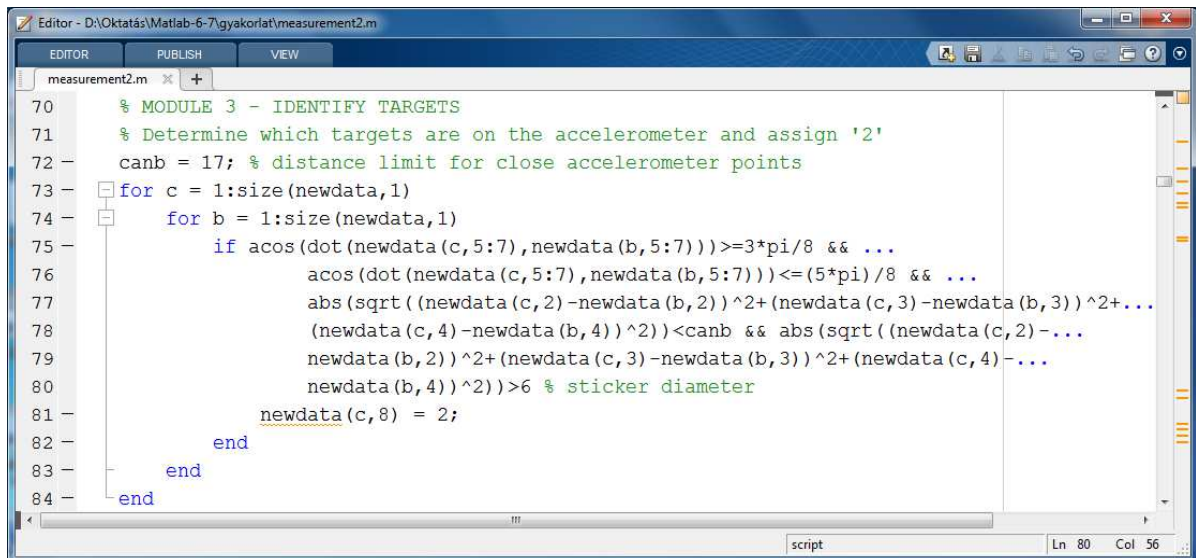
Szintén itt még sorszámokat adunk a mérési pontokhoz (új 1. oszlop), és a későbbiekben szükséges új oszlopokkal (7 db) bővítjük az adattömböt, ezeket egyelőre csupa 0-val töltjük fel (Module 1).

Ezután következik a pontokhoz *a közeli szomszédok számának a meghatározása* (8. oszlop; euklideszi távolság, ha 40 mm-nél kisebb), és a legalább három közeli szomszéddal rendelkező pontok átpakolása (másolása) egy új adattömbbe (newdata). Ezek a pontok *helyezkedhetnek el a gyorsulásmérőkön* (Module 2).



```
50 % MODULE 2 - REMOVE EXTRANEOUS TARGET POINTS
51 % Add column to data specifying the number of targets that are within 40mm
52 gnb = 40; % distance limit for close neighbors (groups)
53 for c = 1:size(data,1)
54     for b = 1:size(data,1)
55         if abs(sqrt((data(c,2)-data(b,2))^2+(data(c,3)-data(b,3))^2+...
56                 (data(c,4)-data(b,4))^2))<gnb
57             data(c,8) = data(c,8)+1;
58         end
59     end
60 end
61
62 % Moves measurement point target groups to a new array
63 newdata = [];
64 for c = 1:size(data,1)
65     if data(c,8)>=3
66         newdata = [newdata;data(c,:)];
67     end
68 end
```

A következő lépésben meghatározzuk a *ténylegesen a gyorsulásmérőkön elhelyezkedő* pontokat. Ehhez közeli pontpáronként közbezárt szöget számolunk (képlet: $\text{acos}(\text{dot}(x, y))$), megnézzük, hogy a vizsgált pontok távolsága kisebb-e, mint 17 mm, de ugyanakkor nagyobb-e, mint 6 mm (ne legyen ugyanaz a két pont). A megtalált pontoknál a 8. oszlop mérőszámát 2-re állítjuk át.

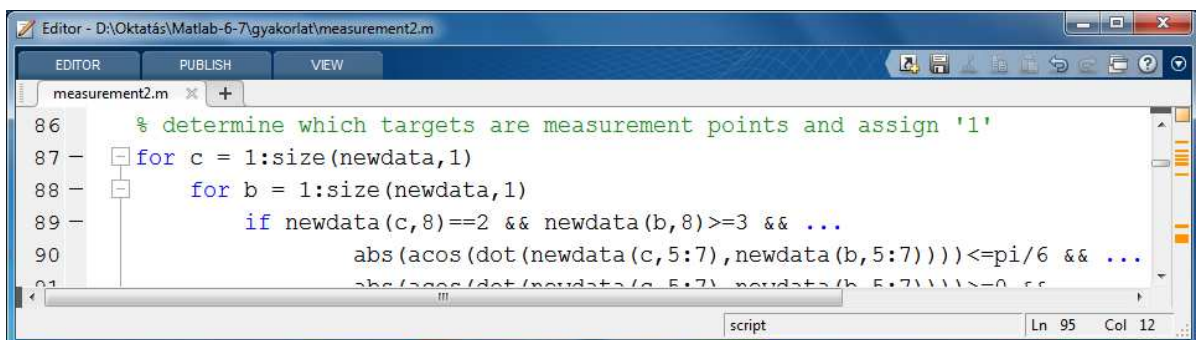


```

70 % MODULE 3 - IDENTIFY TARGETS
71 % Determine which targets are on the accelerometer and assign '2'
72 canb = 17; % distance limit for close accelerometer points
73 for c = 1:size(newdata,1)
74     for b = 1:size(newdata,1)
75         if acos(dot(newdata(c,5:7),newdata(b,5:7)))>=3*pi/8 && ...
76             acos(dot(newdata(c,5:7),newdata(b,5:7)))<=(5*pi)/8 && ...
77             abs(sqrt((newdata(c,2)-newdata(b,2))^2+(newdata(c,3)-newdata(b,3))^2+...
78                 (newdata(c,4)-newdata(b,4))^2))<canb && abs(sqrt((newdata(c,2)-...
79                 newdata(b,2))^2+(newdata(c,3)-newdata(b,3))^2+(newdata(c,4)-...
80                 newdata(b,4))^2))>6 % sticker diameter
81             newdata(c,8) = 2;
82         end
83     end
84 end
  
```

Az arkuszkoszinuszos képletben most a hosszal nem kell osztani, mert úgyis normálvektorokkal dolgozunk.

Ezután következik a *gyorsulásmérők tetején elhelyezkedő ún. releváns mérési pontok* detektálása.



```

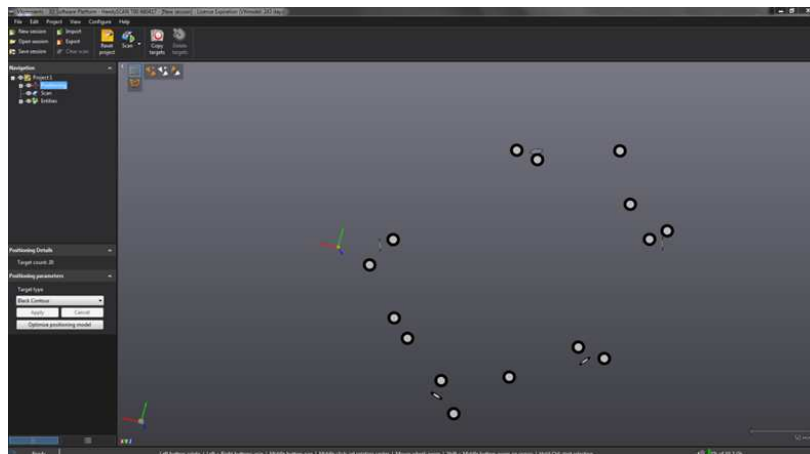
86 % determine which targets are measurement points and assign '1'
87 for c = 1:size(newdata,1)
88     for b = 1:size(newdata,1)
89         if newdata(c,8)==2 && newdata(b,8)>=3 && ...
90             abs(acos(dot(newdata(c,5:7),newdata(b,5:7))))<=pi/6 && ...
91             abs(acos(dot(newdata(c,5:7),newdata(b,5:7))))>=pi/3
  
```

Itt a kódnak *csak egy részletét közöltük*, és a hiányzó rész megírása a feladat.

Az előző részlettel azonos módon közeli pontpárokat vizsgálunk. Ha az éppen aktuális pont azonosítója 2, párjái pedig legalább 3 (azaz *gyorsulásmérőn levő pontot hasonlítunk az alapsíkon levő közelivel*), továbbá a bezárt szögük 0 fok (kis hibával, az arkuszkoszinuszos képletet használjuk), illetve a távolságuk kisebb mint 35 mm, akkor az aktuális pont azonosítóját 1-re állítjuk.

Írjuk meg a hiányzó pár sort! (Ezzel Module 3 is teljes lesz).

Próbáljuk ki a saját script helyességét az „accelerom1.txt” fájlra, amely 5 db gyorsulásmérőn elhelyezett 3-3 db mérési pont, és a mellettük elhelyezett 1-1 segédpont beszkenelt adatait tartalmazza (összesen 20 db mérési pont, ebből 5 releváns)! A munkánk akkor sikeres, ha a script lefutásának végén pontosan 5 db 1-es lesz a 8. oszlopban.



accelerom1-r.txt - Jegyzetmő									
Fajl	Szerkesztés	Formátum	Nézet	Súgó					
-33.12076569	-24.18719292	312.00549316	-0.23165	0.02458	-0.97249	1.0000000000	2	0	
-67.25471497	16.93524742	322.37088013	-0.23948	0.03371	-0.97032	1.0000000000	2	0	
-24.26832008	-7.56352854	284.40209961	-0.25879	0.03910	-0.96514	1.0000000000	2	0	
27.69557190	26.24225044	301.14297485	-0.24441	0.05441	-0.96814	1.0000000000	2	0	
32.11973572	39.75262833	298.20214844	-0.25620	0.06968	-0.96411	1.0000000000	2	0	
-50.04188919	189.53259277	299.27999878	-0.24700	0.03630	-0.96834	1.0000000000	2	0	
61.93379974	76.90521240	292.30899048	-0.26909	0.03330	-0.96254	1.0000000000	2	0	
41.91696930	94.99169159	272.22964478	-0.26743	0.01031	-0.96352	1.0000000000	2	0	
-120.94751518	48.14112473	310.43109131	-0.26040	0.03809	-0.96475	1.0000000000	2	0	
-24.77285004	199.93563843	319.56393433	-0.24745	0.04753	-0.96773	1.0000000000	2	0	
-121.95680237	180.83961487	343.12814331	-0.23768	0.03800	-0.97060	1.0000000000	2	0	
-151.62631226	149.73809814	323.17065430	-0.23318	0.03284	-0.97188	1.0000000000	2	0	
-102.79859161	221.34751892	340.03533936	-0.24730	0.03472	-0.96832	1.0000000000	2	0	
53.69538116	90.64099121	279.67037964	0.90657	-0.33639	-0.25490	1.0000000000	2	0	
-21.07922363	-18.04066467	292.85809326	0.24072	-0.96544	-0.09986	1.0000000000	2	0	
-135.08549500	51.98793793	340.79226685	-0.23914	0.04406	-0.96999	1.0000000000	2	0	
-125.82482147	41.23564529	321.32360840	-0.73151	-0.65688	0.18278	1.0000000000	2	0	
-15.53639221	167.42768860	350.94619751	-0.24466	0.04760	-0.96844	1.0000000000	2	0	
-43.98385620	199.23507690	310.04501343	0.35283	0.93476	-0.04175	1.0000000000	2	0	
-159.10328674	151.96470642	335.63305664	-0.92914	0.26774	0.25496	1.0000000000	2	0	

Feladat (sajátérték, sajátvektor)

Tekintsük a következő 2D transzformációs mátrixokat: helybenhagyás, SWAP12, forgatás α fokkal, tükrözés x tengelyre, tükrözés y tengelyre. Ellenőrizzük, hogy milyen sajátértékek és sajátvektorok léteznek! (Hány valós sajátérték van?)
Végezzünk hasonló elemzést 3D-ben is!

Feladat (kondíciószám)

Igazolható, hogy bármely invertálható mátrix kondíciószáma nem lehet kisebb 1-nél. Ellenőrizzük ezt a tulajdonságot néhány ilyen mátrixra!

