

Huom. Ke 10.4. tietokoneharjoitus on siirretty. Uusi aika on **ma 8.4. klo 16–18**.

Käynnistä MATLAB ja aseta työhakemistoksi haluamasi hakemisto. Muista komento `diary` ja käskyjen selaus nuoli ylös.

3.1 Luo matriisi

$$C = \begin{bmatrix} 16 & 14 & 4 \\ -13 & -2 & 22 \end{bmatrix}$$

ja

- Etsi C :n singulaariarvohajotelma (SVD) $C = USV^T$ missä U ja V ovat ortogonaalisia ja S on diagonaalinen¹. Tarkista saamasi hajotelman oikeellisuus kertomalla. Vihje: `help svd`.
- Poimi C :n singulaariarvot matriisista S vektoriin e komennolla `diag`.
- Vertaa suurinta singulaariarvoa matriisin C normiin. Vihje: `norm` laskee matriisnormin.
- Merkitään $S = \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix}$ missä 0:t ovat kukin sopivan kokoisia nollamatriiseita² ja D on diagonaalinen neliömatriisi, jonka diagonaali-alkiot ovat nolosta poikkeavia.
Merkitään $S^+ = \begin{bmatrix} D^{-1} & 0 \\ 0 & 0 \end{bmatrix}$. Tällöin C :n pseudoinverssi C^+ saadaan lasketta kaavalla $C^+ = VS^+U^T$. Laske C^+ :n arvo tätä kaavaa käyttäen. Vihjeitä: käytä edellä laskemaasi vektoria e hyväksesi. Komennolla `diag` saat myös muodostettua diagonaalimatriisin. Mieti, miten saat laskettua diagonaalimatriisin käänteismatriisin.
- Vertaa saamaasi tulosta komennon `pinv` antamaan pseudoinverssiin.
- Tarkasta, että $CC^+C = C$ ja $C^+CC^+ = C^+$. Laske vielä matriisitulot CC^+ ja C^+C .

3.2 Kopioi tiedosto `0:\Visible2Everyone\MATA123_2013\ex3_2.mat` työhakemistoosi. Lataa MATLAB:iin muuttuja x komennolla `load ex3_2`. Luo työtilaan matriisi

$$A = \begin{bmatrix} -1.6 & -0.6 \\ 0.7 & 0.9 \end{bmatrix}$$

Katso, mitä matriisi A tekee muuttujan x sarakkeiden muodostamalle pistejoukolla komentamalla

¹Huom! Tässä S ei ole neliömatriisi, kuten LAG2-monisteessa.

²”Sopiva” koko voi olla myös esim. $0 \times 2 \dots$

```
>> plot(x(1,:),x(2:,:),'k.')
>> v = A*x;
>> hold on; axis equal
>> plot(v(1,:),v(2:,:),'r.')
```

Kokeile, mitä matriisi tekee pala palalta: Määritä ensin matriisin A singulaariarvo-hajotelma $A = USV^T$ käyttäen komentoa `svd`. Laske sitten muuttujat $y = V^T x$, $z = Sy$ ja $w = Uz$. Piirrä kaikki nämä välivaiheet kuvaan komentamalla

```
>> figure; hold on; axis equal;
>> plot(y(1,:),y(2:,:),'g.')
>> plot(z(1,:),z(2:,:),'b.')
>> plot(w(1,:),w(2:,:),'m.')
```

Laske `cond(A)` ja tarkastele, miten tämä liittyy singulaariarvoihin. Aseta lopuksi toinen singulaariarvo nollassa komentamalla `>>S(2,2)=0` ja muodosta matriisi $B = USV^T$. Tutki vaiheittain, millainen kuvaus nyt syntyy. Laske myös `cond(B)`.

3.3 Avaa MATLAB:in editoriin uusi tiedosto `leikkuri.m` komennolla `edit leikkuri`. Kirjoita tiedostoon rivit

```
function a = leikkuri(b)
N = length(b);
a = zeros(1,N);
for k=1:N
    if b(k) > 5
        a(k) = 5;
    elseif b(k) >= 0
        a(k) = b(k);
    else
        a(k) = 0;
    end
end
```

ja tallenna tiedosto. Selvitä funktion toiminta kokeilemalla mitä funktio antaa, kun syötät sille vektorin z , jonka luot komennolla `z=-5:10`. Kokeile vielä, mitä komento `z == leikkuri(z)` laskee. Vihje: `help eq`.

3.4 Muokkaa edellisen tehtävän funktiota siten, että se toimii matriiseille B , ja sen palauttamassa matriisissa A on alkiot $a_{i,j}$, joille pätee $a_{i,j} = b_{i,j}$ kun $b_{i,j} \in [0, 5]$, $a_{i,j} = 0$ kun $b_{i,j} < 0$, ja $a_{i,j} = 5$ kun $b_{i,j} > 5$. Tallenna muokkaamasi funktio nimellä `leikkuri2.m`. Testaa funktiota matriisilla `B=meshgrid(-5:10)`. Piirrä funktion palauttama matriisi komennolla `mesh, surf` ja `imagesc`.

3.5 Tutki esim. matriisille $B = \text{floor}(5 \cdot \text{randn}(5))$, mitä tekevät komennot

```
>> 0 <= B & B <= 5  
>> 0 <= B <= 5
```

Miksi komennot palauttavat eri tuloksen? Kuinka saat laskettua MATLAB:illa niiden B :n alkioden lukumäärän, jotka osuvat välille $[0, 5]$? Entäpä miten löydät ne matriisin B indeksit (i, j) , joille ehto $0 \leq b_{i,j} \leq 5$ pätee? Vihje: `sum`, `find`.

Kuten tässäkin, monessa tapauksessa MATLAB:illa voi välttää silmukat ja eksplisiittiset ehtolauseet. Funktion `leikkuri2.m` tehtävän tekee myös esim. komento

```
>> min(5, max(B,0))
```

Vertaa ylläolevan komennon ja luomasi funktion `leikkuri2.m` nopeuseroa suurilla matriiseilla.

3.6 Monissa simulaatioissa tarvitaan satunnaislukuja. MATLAB:issa voi luoda satunnaisia matriiseja esim. komennolla `rand` ja `randn`. Luo 100×100 satunnaismatriisi B (muista puolipiste), jonka alkiot ovat tasaisesti jakautuneet välille $[-10, 10]$ ja samankokoinen satunnaismatriisi C , jonka alkiot ovat standardinormaalijakaumasta. Piirrä kuvat matriiseista komennolla `imagesc`. Tutki komennolla `hist`, milä matriisien alkioden jakaumat näyttävät.

Testaa myös seuraavaa koodia:

```
yn = randn(10000,1);  
yu = rand(10000,1);  
x = min(yn):.2:max(yn);  
subplot(1,2,1)  
hist(yn,x)  
title('Normally distributed random numbers')  
subplot(1,2,2)  
hist(yu,25)  
title('Uniformly distributed random numbers')
```

3.7 (*) Arvo satunnaisesti 1000 pistettä, jotka ovat tasajakautuneet neliöön $[-1, 1] \times [-1, 1]$. Piirrä pisteet kuvaan. Laske yksikköympyrään osuvien pisteiden suhteellinen osuus kerrottuna neljällä? Testaa pisteiden eri lukumäärillä. Piirrä yksikköympyrän sisään osuvat pisteet punaisella ja ulkopuolelle osuvat pisteet sinisellä. Anna kuvalle, akseleille ja pisteille selitteet (`title xlabel ylabel legend`).