# mlteht/mlBasic, Matlab-perusteita

Tässä luvussa on tehtäviä Matlab-perusharjoitteluun:
Vektoreilla ja matriiseilla operointia, pisteittäiset ja "matriisimaiset" laskutoimitukset, vektoriajattelu, vektori- ja matriisifunktioita.
Myös grafiikkaa, vaikka sille on varattu oma sektionsa.

---

mlBas000

**Matlab-pikaohje**

(a) Matlab-työ aloitetaan pienimuotoisilla kokeiluilla antamalla **komentoikkunassa** komentoja tyyliin
`>> v=1:10;`

(b) **Skripti**: Vakavampi Matlab-työ tehdään kirjoittamalla editorilla tiedostoon, esim. `ajo.m`. Tiedosto kannattaa rakentaa muotoon:

```
%% Otsikko
%
%% Kappale 1
% Selitystä
% ...
komento       % Kappaleen 1 SUORITUS: CTR-ENTER
komento
 ...
%% Kappale 2
%
komento       % Kappaleen 2 SUORITUS: CTR-ENTER
komento
 ...
%%
```

Julkaisu: *publish*

(c) Komennon suorittama tulos tulee ruudulle ENTER-painalluksen jälkeen (kuvat erilliseen ikkunaan). Jos haluat estää tulostuksen, päätä komento puolipisteeseen. Jos myöhemmin haluat katsoa muuttujan sisällön, kirjoita sen nimi (ilman puolipistettä). Jos muuttuja on suuri matriisi, kannattaa ensin katsoa sen koko `size(A)` tai sen jotain osaa, esim. `A(1:10,1:10)`. Tai klikkaa "workspace"-ikkunan muuttujaikonia.

(d) Edellisen komennon tulos on muuttujassa `ans`. Yleensä on suositeltavaa antaa tulokselle oma nimi tyyliin `nimi= ...`

(e) Nuoliylös-näppäimellä (↑) voi selata aikaisempia komentoja. Huomaa myös **Command history**.
Käytä ahkerasti komentoja `help, doc`.

(f) `format long` : Tulostetaan enemmän numeroita (n. 16). Laskutarkkuuteen tämä ei vaikuta.
`format rational` laskee rationaaliluvuilla.
`format short`: Paluu oletustulostukseen.

(g) Matriisi saadaan aikaan tyyliin: `A=[2 4 3;0 1 -1;3 5 7]`. Vektori saadaan näin: `v=[1 2 3]`. Pystyvektorissa käytetään erottimena puolipistettä (tietysti, vrt. matriisi A yllä). Matriisikertolaskun merkki on `*`

(h) Matriisin `A` transpoosi: `A'` (reaalisessa tapauksessa).

(i) Kokonaislukuvektori: Esim `1:10` tai `1:2:20`. Myös `linspace`. Pystyvektoriksi transponoimalla.

(j) `A(i,j)` A:n alkio (i,j).
`A(2,:)` A:n 2. rivi
`A(:,3)` A:n 3. sarake
`A(1:4,1:4)` osamatriisi
Matriisin osaa voi päivittää, vaikkapa:
`A(1:4,1:4)=ones(4,4)` tai
`A(2,:)=A(2,:)-2*A(:,1)` (Gaussin rivioperaatio).

(k) Matriisien liittäminen: Jos $A$:lla ja $B$:llä on yhtä monta riviä, ne voidaan liittää peräkkäin:
`[A b]`
(tai `[A, b]`). Jos yhtä monta saraketta, niin allekkain: `[A;B]`

(l) Laskutoimitukset tarkoittavat matriisilaskua. Siis esim.
`A*B, A^p` (jälkimmäinen mahdollinen vain neliömatriisille)

(m) Vektorien ja matriisien (samankokoisten) pisteittäinen eli alkioittainen laskenta tapahtuu lisäämällä eteen piste. Esim: `u=[1 2 3], v=[-2 -2 -2], u.*v`.
Toinen operandi voi olla skalaari. Siten esim. vektorin $u$ kaikki komponentit voidaan korottaa toiseen komennolla `u.^2`
(Ei siis tarvitse tehdä: `u.^(2*ones(size(u)))`, joka tietysti toimii.)

(n) Piirtämistä varten muodostetaan $x$-vektori, joka edustaa diskretoitua $x$-akselia ja lasketaan a.o. funktion arvo vektoriin $y$.
**Piirtoesim:**

```
x=linspace(-pi,pi); y=sin(x); plot(x,y)
```

**Huom!** Matlab-funktioita voi yleensä soveltaa vektoriin ja tulokseksi saadaan funktion arvojen muodostama vektori. Laskutoimitukset $+, -$ operoivat vastinalkioittain ("pisteittäin"). Koska kerto- ja jakolasku sekä potenssiin korotus `^` on varattu matriisilaskutoimituksille, on "pisteittäin"operoitaessa lisättävä piste (.) ao. laskutoimitusmerkin eteen. $(+, -)$ merkkien eteen ei saa lisätä, ne ovat jo valmiiksi pisteittäisiä.)

Jos haluamme muodostaa vaikkapa funktion $x^2$ arvot annetun $x$-vektorin pisteissä ja $x$-vektorina olkoon välin [-1,1] diskretointi 60:een osaan, voimme laskea ja piirtää näin: `x=linspace(-1,1,60); y=x.^2; plot(x,y)` . Toinen tapa diskretoida on (:), esim: x=a:h:b;
jossa siis annetaan askeleen pituus `h` (askelten lukumäärän sijasta).

Kts. `help plot, help :, help colon`

(o) **Yhden rivin funktion määritys, funktiokahva**

```
>> f=@(x) x.*exp(x) % Määrittely
>> f(-1:.1:1)       % Käyttö
>> g=@(x,y) x.^2 + y.^2   % Määrittely
>> x=[.1 .2 .3]; y=1:3; g(x,y)  % Käyttö
```

(p) **Lineaarisen yhtälösysteemin ratkaisu**
Yhtälösysteemi: $A\,x = b$

```
>> A=rand(3,3);
>> b=ones(3,1);
>> x=A\b
>> [A*x b]       % Tarkistus
```

(q) 3d-piirto: Pintojen ja korkeuskäyrien piirtämiseksi tarvitaan korkeusarvojen matriisi xy-tason pistehilan päällä. Se aikaansaadaan helpoimmin (ja rutiininomaisesti) `meshgrid`-komennolla. Jos haluaisimme piirtää vaikkapa funktiopinnan $f(x, y) = \sin x \cos y$ neliössä $[-\pi, \pi] \times [-2\pi, 2\pi]$, ja hilapisteitä olisi x-suunnassa 25 ja y-suunnassa 50 kpl., tehtäisiin näin:

```
>> x=linspace(-pi,pi,25);
>> y=linspace(-2*pi,2*pi,50);
>> [X,Y]=meshgrid(x,y);
>> Z=sin(X).*cos(Y);
>> mesh(x,y,Z)    % Rautalankakuva
>> surf(x,y,Z)    % Kaunis pintakuva (myos surfl, surfc, colorbar,...)
>> contour(x,y,Z) % Korkeusk. piirros
```

Lue myös Lyhyen Matlab-oppaan kohdasta:

http://math.aalto.fi/ apiola/matlab/opas/lyhyt/grafiikka.html#surf

**Tiedoston Latex-koodi:**
../mlteht/mlBasic/mlBas000.tex

**Avainsanat:** Matlab perusteet, harjoitus-pikaohje, harjoitusohje

---

mlBas001

Olkoon `z = [0 -1 2 4 -2 1 5 3]`, ja `J = [5 2 1 6 3 8 4 7]`.
Mitä syntyy seuraavilla Matlab-komennoilla (sijoitetaan tilan säästämiseksi useita samalle riville.)

```
x = z', A = x*x', s = x'*x, w = x*J,
length(x), length(z)
size(A), size(x), size(z), size(s)
```

Suorita doc length, doc size, tai etsi Matlabin Help index:n avulla (lisä)tietoa komennoista.

**Vaativuus:** 1-
**Tehtävän Latex-koodi:**
../mlteht/mlBasic/mlBas001.tex

**Ratkaisu:**
m-tiedosto

**Avainsanat:**Matlabperusteet,mlBasic, vektoriperusteet,sisätulo,ulkotulo
**Matlabfunktioita:** *, length,size

---

mlBas002
Muodosta vektori, joka koostuu parillisista kokonaisluvuista välillä $[21, 66]$.

**Vaativuus:** 1-
**Tehtävän Latex-koodi:**
../mlteht/mlBasic/mlBas002.tex

**Ratkaisu:**
pdf-muodossa
m-tiedosto

**Avainsanat:**Matlabperusteet,mlBasic, vektoriperusteet
**Matlabfunktioita:** help colon (:)

---

mlBas003.tex
Olkoon `x=[2 5 1 6 7 4 3 2 1 11]`.

(a) Lisää jokaiseen alkioon luku 12

(b) Lisää 3 parittomien indeksien osoittamiin alkioihin.

(c) Laske vektorin alkioiden neliöjuuri.

(d) Laske vektorin alkioiden neliöt ja neliösumma.

**Vihje:**
Helppo tapa vektorin v indeksivektorin muodostamiseen:
`ind = 1:length(v)`
Miten siis parittomat indeksit?
Summaus sujuu helposti: `help sum`

**Vaativuus:** 1-
**Tehtävän Latex-koodi:**
../mlteht/mlBasic/mlBas003.tex

**Ratkaisu:**
../mlteht/mlBasic/ratkaisut/mlBas003R.m (m-tiedosto)

**Avainsanat:**Matlabperusteet,mlBasic, vektoriperusteet, indeksointi, vektorien muodostus,vektorioperaatio
**Matlabfunktioita:** help colon (:),length,sqrt

---

mlBas004

Olkoon `x = [3 2 6 8 0 -1]' ja y=[4 1 3 5 0 0]'`

(a) Lisää vektorin x alkioiden summa vektoriin y

(b) Korota vektorin x alkiot vektorin y vastinalkioiden osoittamiin potensseihin.

(c) Jaa y:n jokainen alkio vektorin x vastinalkiolla.

(d) Kerro x:n jokainen alkio y:n vastaavalla alkiolla ja talleta tulos muuttujaan z

**Vihje:** Tässä harjoitellaan aritmetiikkaa vektorilausekkeilla. Muista piste (.) laskuoperaation edessä (paitsi $+, -$). Summaukseen: `help sum`
Lue: `help NaN` ja `help inf` .
**Huomaa:** Matlab:lle $0^0 = 1$ (eikä `NaN`)

**Vaativuus:** 1
**Tehtävän Latex-koodi:**
../mlteht/mlBasic/mlBas004.tex

**Ratkaisu:**
../mlteht/mlBasic/ratkaisut/mlBas004R.m (m-tiedosto)

**Avainsanat:**Matlabperusteet,mlBasic, vektoriperusteet, indeksointi, vektorien muodostus,vektorioperaatio
**Matlabfunktioita:** help colon (:),length,sqrt

mlBas005
Muodosta vektori x, joka koostuu alkioista:

(a) $2, 4, 6, 8, \ldots, 20$

(b) $10, 8, 6, 4, 2, 0, -2, -4, \ldots, -10$

(c) $1, 1/2, 1/3, 1/4, 1/5, \ldots, 1/10$

(d) $0, 1/2, 1/3, 1/4, 1/5, \ldots, 1/10$

**Vihje:** (c) ja (d): Muista pisteittäinen jakolasku: `skalaari./vektori`. Voit selkeyttää komentamalla:
`format rational`
Paluu oletusformaattiin: `format`

**Vaativuus:** 1
**Tehtävän Latex-koodi:**
../mlteht/mlBasic/mlBas005.tex

**Ratkaisu:**
../mlteht/mlBasic/ratkaisut/mlBas005R.m (m-tiedosto)

**Avainsanat:**Matlabperusteet,mlBasic, vektoriperusteet, indeksointi, vektorien muodostus,vektoriaritmetii
**Matlabfunktioita:** help colon (:), help format

---

mlBas006
Määrittele vektorit

```
x = [1 2 3 4 5]
y = [0 2 4 6]
z = [-4 -2 0 2 4 ]
```

Kokeile seuraavia laskutoimituksia/komentoja:

```
x.*z
x*z'
x*z              % Miksi virhe ?

x.^2             % Mika vektori?
x^2              % Miksi virhe ?

sqrt(x*x')
sqrt(sum(x.^2))  % Miksi sama tulos kuin edella?
norm(x)
```

**Vihje:** (c) ja (d): Muista pisteittäinen jakolasku: `skalaari./vektori`. Voit selkeyttää komentamalla:
`format rational`
Paluu oletusformaattiin: `format`

**Vaativuus:** 1
**Tehtävän Latex-koodi:**
../mlteht/mlBasic/mlBas006.tex

**Ratkaisu:**
../mlteht/mlBasic/ratkaisut/mlBas006R.m (m-tiedosto)
../mlteht/mlBasic/ratkaisut/mlBas006R.txt (m-tiedoston kopio .txt [että kaikki selaimet ymmärtää])

**Avainsanat:**Matlabperusteet,mlBasic, vektoriperusteet, indeksointi, vektorien muodostus,vektoriaritmetiik matriisitulo, vektorinormi
**Matlabfunktioita:** sqrt, (.*), (*), norm

---

mlBas007
Määrittele matriisit

$$u = \begin{bmatrix} 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} \quad v = \begin{bmatrix} 0 \\ 2 \\ 4 \\ 6 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 3 & 6 \\ 1 & 4 & 9 \end{bmatrix} \quad A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix} \quad B = \begin{bmatrix} 3 & 4 & 5 & 6 \\ 2 & 1 & 0 & -1 \\ 5 & 4 & 2 & 0 \\ 1 & 2 & 1 & 1 \end{bmatrix}$$

Selvitä (ilman Matlabia), mitkä seuraavista laskutoimituksista on määritelty, ja kerro sanallisesti, mitä ne tekevät. Tarkista MATLAB:lla.

```
A*C   C*A    C^2    C.^2   A^2    A.^2
```

**Vihje** Tee skripti, jossa kukin laskutoimitus on omana %%-merkeillä erotettuna lohkonaan tyyliin:

```
%%
A*C % Lyhyt selitys
%%
C*A % Lyhyt selitys
%%
...
```

Vie kursori kuhunkin lohkoon vuorollaan ja `CTR-ENTER`, ja seuraa MATLAB-komentoikkunan tapahtumaa.

**Vaativuus:** 1
**Tehtävän Latex-koodi:**
../mlteht/mlBasic/mlBas007.tex

**Ratkaisu:**
../mlteht/mlBasic/ratkaisut/mlBas007R.m (m-tiedosto)

**Avainsanat:**Matlabperusteet,mlBasic,taulukko-operaatiot, matriisitulo, vektorinormi
**Matlabfunktioita:** (.*), (*)

---

mlBas008

Tehtävän mlBas0081 lyhennetty versio.

Määrittele matriisit
$A = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 3 & 6 \\ 1 & 4 & 9 \end{bmatrix}$ $B = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix}$,
Kokeile ja selitä:

```
A*B  B*A  A^2  A.^2  B^2  B.^2
```

**Vihje** Tee skripti, jossa kukin laskutoimitus on omana %%-merkeillä erotettuna lohkonaan tyyliin:

```
%%
A*C % Lyhyt selitys
%%
C*A % Lyhyt selitys
%%
...
```

Vie kursori kuhunkin lohkoon vuorollaan ja `CTR-ENTER`, ja seuraa MATLAB-komentoikkunan tapahtumaa.

**Vaativuus:** 1
**Tehtävän Latex-koodi:**
../mlteht/mlBasic/mlBas008.tex

**Ratkaisu:**
../mlteht/mlBasic/ratkaisut/mlBas008R.m (m-tiedosto)

**Avainsanat:**Matlabperusteet,mlBasic,taulukko-operaatiot, matriisitulo
**Matlabfunktioita:** (.*), (*)

mlBas0081

Olkoot
$A = \begin{bmatrix} 10 & -3 \\ 4 & 2 \end{bmatrix}$, $B = \begin{bmatrix} 1 & 0 \\ -1 & 2 \end{bmatrix}$, $v = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, $w = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$
Laske sekä käsin (ainakin nyt joku) että Matlabilla:

(a) $v^T w$, (b) $v\, w^T$, (c) $A\, v$, (d) $A^T v$
(e) $AB - BA$, (f) Vektori x, jolle $Ax = v$

**Vihje** Tee skripti, jossa kukin laskutoimitus on omana %%-merkeillä erotettuna lohkonaan tyyliin:

```
%% Matriisien muodostus
A =[..;..], B=...
%% (a)

%% (b)
...

%%
...
```

Vie kursori kuhunkin lohkoon vuorollaan ja `CTR-ENTER`, ja seuraa MATLAB-komentoikkunan tapahtumaa.

**Vaativuus:** 1
**Tehtävän Latex-koodi:**
../mlteht/mlBasic/mlBas0081.tex

**Ratkaisu:** (Tuskin tarpeen)

**Avainsanat:**Matlabperusteet,mlBasic,taulukko-operaatiot, matriisitulo
**Matlabfunktioita:** (.*), (*), Lineaarinen yhtälöryhmä `(\)`

---

mlBas0082.tex

(a) Määrittele $5 \times 5$-nollamatriisi $A$.

(b) Muuta A:n 1. rivin alkot ykkösiksi.

(c) Muuta A:n 5. rivin alkiot viitosiksi.

(d) Muuta A:n 5. sarake muotoon $[1, 2, 3, 4, 5]^T$.

**Vaativuus:** 0+
**Tehtävän Latex-koodi:**
../mlteht/mlBasic/mlBas0082.tex

**Avainsanat:**Matlabperusteet,mlBasic, kaksoispiste (:), building matrices, updating rows and columns of a matrix

**Matlabfunktioita:** kaksoispiste, colon(:), zeros

---

mlBas009

(a) Avaa uusi m-tiedosto (skripti) vaikkapa alkulukuja.m . Kirjoita siihen komennot, joilla saat selville kaikkien korkeintaan N:n suuruisten alkulukujen lukumäärän ja summan. Laske lisäksi lukumäärän suhde kaikken lukujen $\leq N$ lukumäärään, ja myös sama summille.

(b) Tee edellisen pohjalta funktio tiedostoon *alkulukulasku.m* :

```
function [summa, lkm]=alkulukulasku(N)
...
summa =... ;
lkm =...;
end
```

**Vihje (a):** Aloita tiedosto näin:

```
%% Selita, mita skripti tekee ja vaikka oma nimi, pvm. ym.
N = 100
alkuluvut= ...
lkm = ...
summa= ...
...
```

**Vaativuus:** 1
**Tehtävän Latex-koodi:**
../mlteht/mlBasic/mlBas009.tex

**Ratkaisu:**
../mlteht/mlBasic/ratkaisut/mlBas009R.m (m-tiedosto)

**Avainsanat:**Matlabperusteet,mlBasic,skripti, funktiotiedosto,m-tiedosto, alkuluvut, primes
**Matlabfunktioita:** primes, sum

---

mlBas010.tex

(a) Miten kääntäisit vektorin v alkiot vastakkaiseen järjestykseen kaksoispisteen (:) avulla?

(b) Entä matriisin A sarakkeet, vastaavasti rivit?

(c) Miten limität ("merge") kaksi samanpituista vektoria $u$ ja $v$?
Tarkoitus on siis muodostaa vektori $w = [u_1, v_1, u_2, v_2, \ldots]$

**Vihje** (a-b) Voit verrata Matlabfunktioin: `fliplr, flipud` "LeftRight, UpDown"(Mieti ensin itse!)
(c) Liitä vektorit allekkain ja jonouta näin saatu 2-rivinen matriisi sarakkeittain (ovelaa). (Sarakkeittain jonoutus matriisille $A$ saadaan näin: `A(:)`.)
**Vaativuus:** 1
**Tehtävän Latex-koodi:**
../mlteht/mlBasic/mlBas010.tex

**Ratkaisu:**
../mlteht/mlBasic/ratkaisut/mlBas010R.m (m-tiedosto)

**Avainsanat:**Matlabperusteet,mlBasic,fliplr,flipud, kaksoispiste (:), kaanteinen jarjestys, matriisin jonoutus, "merge", limitys
**Matlabfunktioita:** fliplr,flipud, kaksoispiste, colon (:)

---

mlBas011
**Matriisin kokoaminen osista, lohkomatriisit, skriptit**

Tutustu helpin avulla funktioihin: `eye, ones, zeros, diag, size`.
Aloita sitten hommat avaamalla uusi skripti-tiedosto, jonne kirjoitat kommentit ja komennot.

Olkoot $Y_{n\times k}$ ja $N_{n\times k}$ ykkösistä ja vastaavasti nollista koostuvia $n \times k$-matriiseja ja olkoon $I_{n\times n}$ yksikkömatriisi. Muodosta seuraavat (lohko)-matriisit esim. arvoilla $n = 4, k = 3$. Rakenna skripti siten, että näitä on helppo muutella.

$$A = \begin{bmatrix} I_{n\times n} & Y_{n\times k} \\ N_{k\times n} & I_{k\times k} \end{bmatrix}, \quad B = \begin{bmatrix} N_{n\times n} & I_{n\times n} \\ -I_{n\times n} & N_{n\times n} \end{bmatrix}$$

(a) Poimi $A$ :n pää- ja sivulävistäjä. **Neuvo:** Jälkimmäisessä on hyötyä vaikkapa `fliplr`-komennosta.

(b) Poimi B:n "alalävistäjät", jotka alkavat 4:n askeleen päässä päälävistäjästä 1. vaaka- ja 2. pystysuunnassa. (Edelleen: `help diag`).

Lopuksi voit käyttää `publish`-komentoa dokkarin aikaansaamiseksi.

**Vaativuus:** 1
**Tehtävän Latex-koodi:**
../mlteht/mlBasic/mlBas011.tex

**Avainsanat:** Matlabperusteet,mlBasic,diag, fliplr,flipud,Lohkomatriisit, skriptit

**Matlabfunktioita:** diag, fliplr,flipud, kaksoispiste, colon (:)

---

mlBas012.tex
Taikaneliön saa komennolla magic(n) . Muodosta muutamalla pienehköllä n:n arvolla matriisin M=magic(n) rivisummat, sarakesummat, lävistäjäsumma ja sivulävistäjäsumma.

Taikaneliöillä on mielenkiintoinen historia. Ne tunnettiin Kiinassa 2000 vuotta e.a.a. `http://www.mathworks.com/moler/intro.pdf` Kts. Molerin kirjan introsta s. 18 alk. Myös Matlab:n dokumentaatiosta.

**Vaativuus:** 1-
**Tehtävän Latex-koodi:**
../mlteht/mlBasic/mlBas012.tex

**Avainsanat:** Matlabperusteet,mlBasic,taikanelio, magic, rivisummat, sarakesummat,diag
**Matlabfunktioita:** diag,magic )

---

mlBas013
Kun *Newtonin menetelmää* sovelletaan yhtälöön $x^2 - a = 0$, saadaan iteraatiojono

$$x_0 = a, x_{n+1} = \frac{1}{2}(x_n + \frac{a}{x_n}),$$

joka suppenee kohti lukua $\sqrt{a}$. Kirjoita MATLAB-skripti, jolla voit tarkastella tätä suppenemista, kun $a = 5$. Alkuarvona voit tässä tapauksessa käyttää vaikka lukua $a$.
Anna tuloksena taulukko T, jossa on sarakkeet (vain numeeriset, ei otsikoita):

$$\begin{array}{ccc}
n & x(n) & \text{virhe} \\
0 & a & \sqrt{a}-1 \\
1 & x(1) & \sqrt{a}-x(1) \\
\vdots & \vdots & \vdots \\
N & x(N) & \sqrt{a}-x(N)
\end{array}$$

**Vihje**
Suppenemisen tutkimisessa kannattaa käyttää `for` -luuppia. Se toimii syntaksilla

```
for k = 1:N-1
    x(k+1)=...
    virhe(k+1)=...
end
```

Oikean lopetusehdon muodostamiseen `while`-rakenne on parempi, mutta tämä nyt ensi harjoitteluun etenkin, kun suppeneminen on hyvin nopeaa.

Vrt. teht. **mlBas013a**, jossa pyydetään `while`-ratkaisua.

(a) Laske vain $x$-vektori for-luupissa ja muodosta taulukko T liittämällä 3 saraketta vierekkäin. (Yksinkertaisempi ja "Matlabmaisempi" tapa)

(b) Rakenna taulukko T rivi riviltä suoraan `for`-silmukassa. (Tämäkin on opettavaista, hiukan joudut "indeksishiftaukseen" erityisesti, koska 0 on kielletty indeksi matriiseissa.)

**Jatkovihje**

(a)-tapauksessa kannattaa alustaa (pysty)vektori:
`x=zeros(N+1,1)`

(b)-tapauksessa kannattaa alustaa matriisi:`T=zeros(N+1,3)`;

Matlab ei alustusta vaadi, mutta isoilla datoilla alustaminen tehostaa huomattavasti, koska muuten tulkki joutuisi allokoimaan uutta muistitilaa joka kierroksella.

Tulostustarkkuuden säätö: `format long` (ei vaikuta laskentatarkkuuteen).

**Huom: Indeksointi alkaa 1:stä**

**Vaativuus** 1+
**Tehtävän Latex-koodi:**
../mlteht/mlBasic/mlBas013.tex

**Ratkaisu:**
../mlteht/mlBasic/ratkaisut/mlBas013R.m (m-tiedosto)
../mlteht/mlBasic/ratkaisut/mlBas013R.txt (m-tiedoston kopio .txt-tyyppisenä)

**Avainsanat:**Matlabperusteet,mlBasic,iteraatio, iteration, for-loop, Newtonin menetelma neliojuuren laskentaan
**Matlabfunktioita:** for, ones, sqrt

---

mlBas013a [Vrt. mlBas013, ota tarvittaessa ensin. ]

Kun *Newtonin menetelmää* sovelletaan yhtälöön $x^2 - a = 0$, saadaan iteraatiojono

$$x_0 = a, x_{n+1} = \frac{1}{2}(x_n + \frac{a}{x_n}),$$

joka suppenee kohti lukua $\sqrt{a}$. Alkuarvona käytetään tässä siis lukua $a$.

Kirjoita funktio *newtsqrt*, joka palauttaa Newtonin menetelmällä lasketun likiarvon syötteenä annetulle luvulle $a$. Käytä `while`-rakennetta ja lopetusehtoa, joka vertaa uutta iteraatiojonon arvoa edelliseen. (Ei siis vertailua Matlabilla laskettuun lukuun $\sqrt{a}$, jota toki kannattaa käyttää tarkistukseen.)

**Vihje**

```
function x=newtsqrt(a)
% Lasketaan neliöjuuri a Newtonin menetelmällä.
% Lopetusehto: Peräkk. termien suhteellinen ero <= 10*eps.
x=a;
xuusi=...; % Hiukan epäelegantisti tehdään yksi erillinen iteraatio.
           % Elegantimpi tapa neuvottu alla:
tol=10*eps; % Toleranssi hiukan (dekaadin) päälle kone-epsilonin
while abs(xuusi-x) >=x*tol
  x=xuusi;
  xuusi=...;
end;
end
```

```
x=-Inf;   % Elegentti alustus, jolloin ei tarvita erillistä ekaa kierrosta.
xuusi=a;
```

Ohjelmankehityksessä kannattaa sijoittaa aluksi kommentiksi otsikkorivi ja sopivat muut rivit ja ajaa kommentoimattomia komentoja skriptinä. Voit jakaa %%-merkeillä osa-alueisiin. (Tämä siltä varalta, ettet ensikirjoittamalla saisi funktiotasi virheettömäksi.) Toki debuggeri on myös käytössä, mutta skriptikehittely lienee useimmille mieluisampaa.

**Vaativuus** 1+
**Tehtävän Latex-koodi:**
../mlteht/mlBasic/mlBas013a.tex

**Ratkaisu:**

../mlteht/mlBasic/ratkaisut/mlBas013aR.m (m-tiedosto)
../mlteht/mlBasic/ratkaisut/newtsqrt.m (m-tiedosto, sama koodi kuin edellä, luonnollinen nimi ja sopivin help-tekstein täydennettynä)

**Avainsanat:**Matlabperusteet,mlBasic,iteraatio, iteration, while-rakenne, Newtonin menetelmä neliojuuren laskentaan
**Matlabfunktioita:** while, sqrt

---

mlBas014 (Matlab), mplBas014 (Maple)
Maple [Mathematica] , Matlab (erityisesti b)-kohta).

Tarkastellaan funktiota

$$f(x) = 1 + \frac{sin(x)}{1 + x^2}.$$

(a) **Maple:**
Määrittele f lausekkeeksi, laske f:n arvo pisteessä $x = -2.0$ ja piirrä kuvaaja välillä $[-5, 5]$.

**Matlab:**
Tee vastaava asia Matlabilla, kirjoita skripti. Huomaa, että Matlabissa täytyy ensin antaa x:lle numeerinen (vektori)arvo. (Tai käyttää symbolic toolboxia, jolloin komennetaan ensin: `syms x`.)

(b) Tee samat asiat, mutta nyt määrittelemällä f funktioksi.

### Vihje

a)

```
  Maple                                 Matlab:

> f:=1-...                            >> x=...
> subs...                             >> f=...
> plot                                >> plot
```

b)

```
   Maple                               Matlab

> f:=x->1-...                        >> f:=@(x) 1-...
```

**Vaativuus:** 1
**Tehtävän Latex-koodi:**
../mlteht/mlBasic/mlBas014.tex

**Ratkaisut:**

**Matlab:** ../mlteht/mlBasic/ratkaisut/mlBas014R.pdf pdf-muodossa
../mlteht/mlBasic/ratkaisut/mlBas014R.m (m-tiedosto)

**Maple:** ../mplteht/mplBasic/ratkaisut/mplBas014R.pdf pdf-muodossa
../mplteht/mplBasic/ratkaisut/mplBas014R.mw (mw-tiedosto)

**Avainsanat:**Matlabperusteet,mlBasic,Mapleperusteet,mplBasic, lauseke, funktio
**Matlabfunktioita:** (syms)
**Maplefunktioita:** subs, eval

---

mlBas016
Avaa Matlabin `FILE`-valikosta uusi m-tiedosto ja valitse "skripti". Talleta nimelle `cossinplot.m`. Kirjoita tai kopioi tiedostoon täällä oleva teksti
`http://www.cs.cornell.edu/cv/Books/SCMV/Mfiles/chap1.htm#SinePlot`
Voit kopioida sen myös tästä:

```
%% Script File: SinePlot
% Displays increasingly smooth plots of sin(2*pi*x).
close all   % Suljetaan mahd. avatut grafiikkaikkunat.
for n = [4 8 12 16 20 50 100 200 400]
   x = linspace(0,1,n);
   y = sin(2*pi*x);
   plot(x,y)
   title(sprintf('Plot of sin(2*pi*x) based upon n = %3.0f points.',n))
   pause(1)
end
```

Suorita komennot

- copy/paste:lla istuntoon tai
- editorin vihreällä nuolella tai F5:llä tai CTR-ENTER tai
- kirjoittamalla Matlab-istuntoon tiedoston nimi: `cossinplot`

Muuta pause-komento muotoon `pause()`, jolloin komentojono jää odottamaan ENTER-painallusta. Voit kirjoittaa ennen `pause`- komentoa kehoituksen tyyliin `disp('Paina ENTER:iä jatkaaksesi')`. Samalla voit editoida n-vektoria loppupäästä lyhyemmäksi.

Näin pääset hallitummin katsomaan tilannetta.

**Vaativuus:** 1-
**Tehtävän Latex-koodi:**
../mlteht/mlBasic/mlBas016.tex

**Avainsanat:**Matlabperusteet,mlBasic,Skripti, komentotiedosto, kuva, piirto,plot
**Matlabfunktioita:** plot, linspace

**Lähde:** [CvL] C. van Loan: Introduction to Scientific computing with Matlab.

---

mlBas017
Vahvista numeerisesti uskoasi matemaattiseen totuuten siitä, että summa

$$p(n) = \sum_{k=0}^{n} \frac{1}{16^k} \left( \frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6} \right)$$

suppenee kohti arvoa $\pi$ kun $n \to \infty$.

(a) Suorita tehtävä vektoroidusti muodostamalla vektori k kaksoispiste (:) - operaattorilla. Muodosta pisteittäistä vektoriaritmetiikkaa käyttäen ensin termien jono: *termivektori* ja laske summa funktiolla *sum*.
Muodosta myös kumulatiivinen summa, jolloin voit tarkkailla suppenemista, funktio on *cumsum*. Tee taulukko, jossa on indeksisarake ja osasummasarake.

(b) Voit harjoitella myös "epämatlabmaista" skalaariohjelmointia ja `for`-luuppia

(c) Ohjausrakenteiden harjoittelemiseksi voit tehdä saman myös *while*-rakenteella, joka sopii yleensä parhaiten iteratiivisiin algoritmeihin. Lopetusehto saadaan virhetoleranssista, eikä kokeilemalla, kuten for-silmukassa.

Kirjoita skriptiksi, jossa voit vaihdella parametria n, tottakai!

**Vaativuus:** 1+
**Tehtävän Latex-koodi:**
../mlteht/mlBasic/mlBas017.tex

**Ratkaisu:**
../mlteht/mlBasic/ratkaisut/mlBas017Rvekt.m
../mlteht/mlBasic/ratkaisut/mlBas017Rfor.m
while-versio puuttuu, teepä itse!

**Avainsanat:**Matlabperusteet,mlBasic, vektoriaritmetiikka, kaksoispiste (:), pisteittäinen, pointwise, sum, cumsum, for, while
**Matlabfunktioita:** sum, cumsum, for, while, kaksoispiste, colon (:)

---

mlBas018
Huom: Tehtävä on varsin tarkkaan neuvottu. Pituus ei merkitse vaikeutta.

Tutkitaan heitetyn pallon lentorataa MATLABilla. Aloita luomalla m-tiedosto johon kirjoitat tarvittavat komennot.

1. Teemme seuraavat lähtöoletukset:

   i Pallon korkeus $h$ heittohetkellä on $1.5m$

   ii Putoamiskiihtyvyys $g$ on $9.8m/s^2$

   iii Pallon vauhti $v$ heittohetkellä on $4m/s$

   iv Pallon etenemisvektorin suunta $\theta$ on $45^o$

   Kirjoita oletukset skriptiisi.

2. Luo vektori `t`, jossa on 1000 tasaisin välein valittua arvoa väliltä $[0, 1]$.

3. Kuvataan muuttujalla $x$ pallon etäisyyttä heittäjästä (mitattuna maan pinnalla) ja muuttujalla $y$ pallon korkeutta, seuraavat yhtälöt kuvaavat muuttujien riippuvuutta ajasta ja oletetuista parametreista.

   (a)
   $$x(t) = v\cos\left(\theta\frac{\pi}{180}\right)t.\text{Muunnetaan kulma radiaaneiksi}$$

   (b)
   $$y(t) = h + v\sin\left(\theta\frac{\pi}{180}\right)t - \frac{1}{2}gt^2.$$

   Kirjoita annettujen yhtälöiden ja määrittelemiesi arvojen avulla vektorit `x` ja `y`.

4. Arvioidaan hetkeä jolloin pallo putoaa maahan, ja sen lentämää matkaa: etsi ensimmäinen indeksi, jolla pallon korkeus $y$ muuttuu negatiiviseksi (käytä funktiota `find`). Pallon lentämä etäisyys on vektorin $x$ arvo tässä indeksissä, lentoaika on vektorin $t$ arvo tässä indeksissä. Tulosta sekä lentomatka että -aika näkyviin ruudulle.

5. Piirretään pallon lentorata: piirrä kuva, jossa pisteiden x-koordinaatit ovat vektorissa $x$, ja y-koordinaatit vektorissa $y$. Tämän jälkeen piirrä nolla-taso näkyviin katkoviivalla.

**Vihje**

**Vaativuus:** 1+
**Tehtävän Latex-koodi:**
../mlteht/mlBasic/mlBas018.tex

**Ratkaisu:**
../mlteht/mlBasic/ratkaisut/html/mlBas018R.html
../mlteht/mlBasic/ratkaisut/html/mlBas018R.pdf
../mlteht/mlBasic/ratkaisut/mlBas018R.m

**Avainsanat:**Matlabperusteet,mlBasic,find
**Matlabfunktioita:** find, linspace, plot

---

mlBas019

Olkoon

$$g(x) = \begin{cases} 0, x \leq 0 \\ x, x > 0 \end{cases}$$

Määrittele funktio g Matlab-funktioksi (m-tiedostoon).

**Vihje** Voit käyttää funktioita `zeros` ja `max` .

Ehkä vieläkin elegantimmin näin:
Mieti, millä saat aikaan yksikköaskelfunktion (Heavisiden funktion), joka saa negatiivisilla arvon 0 ja positiivisilla 1. (Tähän riittää 3 merkkiä.) Sillä kerrot funktion $y = x$.

**Vaativuus:** 1
**Tehtävän Latex-koodi:**
../mlteht/mlBasic/mlBas019.tex

**Ratkaisu:**
../mlteht/mlBasic/ratkaisut/mlBas019R.html
../mlteht/mlBasic/ratkaisut/mlBas019R.m (m-tiedosto)

**Avainsanat:** Matlabperusteet,mlBasic, ramppifunktio, paloittain määrittely, zeros, max **Matlabfunktioita:** zeros, max

---

mlBas020
(Maple ja Matlab)

Määritä seuraavat summat:

$$\sum_{k=1}^{1000} k \quad \text{ja} \quad \sum_{k=1}^{\infty} \frac{1}{k^2}.$$

**Vihje** Maple: Kokeile edelliseen sekä `sum` että `add` - komentoja, jälkimmäiseen vain `sum`.

Matlab: Muodosta vektori `1,2,...1000` ja sitten vain `sum`. Jälkimmäisessä voit laskea muutamalla, toinen toistaan suuremmalla arvolla. (Numeerisesti et tietenkään voi summata äärettömyyksiin.)

**Vaativuus:** 1
**Tehtävän Latex-koodi:**
../mlteht/mlBasic/mlBas020.tex

**Avainsanat:**Matlabperusteet,mlBasic, kaksoispiste (:), sum, cumsum
**Matlabfunktioita:** sum, cumsum
**Maplefunktioita:** sum, add

---

mlBas021
Esitä yhden rivin Matlab-komento, jolla saat selville vektorin tai matriisin niiden alkioiden lukumäärän, jotka ovat $> 5$.

Testaa ainakin näille:

a) `A=1:10`

b) `B=[1 2 3;4 5 6;7 8 9]`

c) `C=10*rand(6,6)`

d) `D=ones(4,4)`

**Vaativuus:** 1
**Tehtävän Latex-koodi:**
../mlteht/mlBasic/mlBas021.tex

**Ratkaisu:**
../mlteht/mlBasic/ratkaisut/html/mlBas021R.html
../mlteht/mlBasic/ratkaisut/mlBas021R.m (m-tiedosto)

**Avainsanat:**Matlabperusteet,mlBasic, $>$
**Matlabfunktioita:** $>$, colon (:)

---

**1.** On esitetty, että jatkuva funktio $f : [0, \infty) \to [0, \infty)$, joka toteuttaa ehdot

1. $f(2x) = 2f(x)$, and
2. $f(1) = c$

on aina muotoa $f(x) = cx$. Tälle on kuitenkin esitetty seuraava vastaesimerkki:

$$f(x) = 2^{-n}x^2 + 2^{n+1}, x \in [2^n, 2^{n+1}),$$

$n = \ldots, -2, -1, 0, 1, 2, \ldots$. Kirjoita funktio $f$ MATLABissa ja piirrä sen kuvaaja.

**Vihje:** Tehtävän tarkoitus on opastaa MATLABin katto- ja lattiafunktioiden käytössä. Nämä ovat nimeltään `floor` ja `ceil` - katso tarkempia tietoja MATLABin helpistä.

mlBas023.tex
Piirrä funktiot $x^4$ ja $2^x$ samaan kuvaan ja selvitä, miten monessa pisteessä kuvaajat leikkaavat.

**Vaativuus:** 1
**Tehtävän Latex-koodi:**
../mlteht/mlBasic/mlBasBas.tex

**Avainsanat, keywords:** Matlab perusteet, Matlab basics

mlBas024.tex

Opettajalle: Voit tietysti ottaa sopivia osia tai keksiä lisää tilanteen mukaan.

**Kolmion ala** voidaan lausua sivujen $a, b, c$ avulla *Heron'n* kaavalla:

$$A = \sqrt{(p-a)(p-b)(p-c)},$$

missä $p = \frac{a+b+c}{2}$ (piirin puolikas).

**osa A skripti**

Kirjoita skripti `kolmio.m`, joka päättelee ja laskee seuraavaa:

1. Päättelee annettujen kolmen luvun $a, b, c$ perusteella, voidaanko muodostaa kolmio, jonka sivujen pituudet ovat nuo luvut. Jos voidaan, antaa luvun 1, muuten 0.
2. Laskee kolmion alan, kun sivujen pituudet annetaan, jos edellä saatiin 1.
3. Tulostaa tektin (a) "tasakylkinen", (b) "tasasivuinen", (c) "suorakulmainen", tapauksen mukaan.
4. Piirtää kolmion, vaikkapa niin, että pisin sivu alkaa O:sta ja on x-akselilla. Voit ratkaista ympyröiden leikkauspisteet sopivalla ratkaisijalla tai piirtää ympyrät ja poimia leikkauspisteen kuvasta `ginput`:n avulla.

**osa B funktio**

Kirjoita funktio mikakolmio:

```
function [onkokolmio,kolmiontyyppi,pinta_ala] = mikakolmio(a,b,c)
%   Laskee ...
%   Esim:
%   [onko,tyyppi,ala] = mikakolmio(2,3,2)
%
%   onko =
%        1
%   tyyppi =
%   taskylkinen
%   ala =
%       1.9843
...
```

Sisällytä myös piirto.

**Vaativuus:** 2

**Tehtävän Latex-koodi:**

../mlteht/mlBasic/mlBas040.tex

**Ratkaisu:**
../mlteht/mlBasic/ratkaisut/mlBas040R.m

**Lisätietoa Hero:sta**

"The formula is credited to Hero (or Heron) of Alexandria, who was a Greek Engineer and Mathematician in 10 – 70 AD." Poiminta viitteestä:
https://www.mathsisfun.com/geometry/herons-formula.html

**Avainsanat, keywords:** Matlab perusteet, Matlab basics, ohjelmointi, programming, kolmion pinta-ala, area of triangle, sort,if-then-else-elseif, functionfile, ginput

---

mlBasyyy.tex
Source: Matlab group in Facebook

HIUKAN EPÄSELVÄ TEHTÄVÄ, vaatinee lisätietoa. – "UNCLEAR"

Given two lists of numbers:

- $1, 2, 3, \ldots, 9$
- $2, 3, 4, 8, 9, 10, 13, 14, 15.$

Sort the lists into $3 \times 3$ matrices $A$ and $B$ to
(a) maximize and
(b) to minimize $det(A\,B)$

(Perhaps needs some more explanation (?))

**Vaativuus:** 2 (?)
**Tehtävän Latex-koodi:**
../mlteht/mlLinalg/mlBas030.tex

**Avainsanat, keywords:** Matlab perusteet, Matlab basics

---

## mlComplAnal

Tehtäviä kompleksilukujen ja -funktioiden käsittelyyn Matlab:lla.

---

mlCA001.tex
Ensiapuohjeita

- Sijoitus muuttujaan esim: `>> z=(1+i)/(1-2*i)`
  Puolipiste lopussa estää tulostuksen.
- Muuttujan sisällön näet kirjoittamalla sen nimen ilman puolipistettä

- Lopeta puolipisteeseen, jos komentosi tuottaa dataa paljon, isojen tulosteiden vilistäminen ruudulla on pelkkää kärsimystä.
- Jos kuitenkin unohdit, niin `CTR-C`
- Vektorin voi muodostaa esim. näin: `>> vektori=[-1,0,4,5.2-3.45*i]`
  Pitempiin vektoreihin: `help colon, help linspace`
- Kun teet aritmetiikkaa vektoreille tyyliin
  `>> x=linspace(0,1); y=(x.^2).*sin(x);`
  niin **muista piste** ja ymmärrä. (Tässä sulut vain selvennykseksi.)
- Kaikki vähänkin vakavampi työskentely kannattaa tehdä avaamalla editorilla tekstitiedosto, jonne kerätään Matlab-ajon komennot (ja mahdollisesti tärkeimmät tulosteet).
- Matriisi muodostetaan tyyliin `>> A=[1 2 3;4 5 6]`
  2. rivi: `A(2,:)`, 3. sarake: `A(:,3)` . Kokeile ihmeessä!

---

mlCA002

Sijoita muuttujalle $z$ arvo $2 + 3i$. Laske Matlabilla $z$:n reaaliosa, imaginaariosa,liittoluku, moduli (itseisarvo) ja argumentti (napakulma). Piirrä pisteet $z$ ja $\overline{z}$. Anna sopiva axis-komento, jotta saat paremman koordinaatiston. Piirrä sitten samaan kuvaan jana origosta pisteeseen z punaisella värillä ja origosta pisteeseen $\overline{z}$ sinisellä.

**Vihje:**
Sopivia Matlab-komentoja: `real, imag, conj, abs, angle, sign, plot, axis, hold on`
Tutki kutakin helpillä tyyliin `help real` jne.

Opiskele aihetta "kompleksilukujen piirtäminen":

http://math.aalto.fi/∼apiola/matlab/opas/lyhyt/grafiikka.html#complexplot

Tässä lyhyesti:

- `plot(z)` piirtää pisteen $z$ tasoon, jos $z$ on kompleksiluku.
- Yleisemmin, jos $z$ on kompleksivektori, `plot(z)` piirtää $z$-vektorin pisteiden kautta murtoviivan.
- Jos halutaan pelkät pisteet, niin `plot(z,'o')` (tai esim `plot(z,'xb')` jos sinisellä ristillä).

**Vaativuus:** 1
**Tehtävän Latex-koodi:**
../mlteht/mlComplAnal/mlCA002.tex

**Ratkaisu:**
../mlteht/mlComplAnal/ratkaisut/mlCA002R.html

**Avainsanat:**Matlabperusteet,mlComplAnal, kompleksiluvut
**Matlabfunktioita:**`real, imag, conj, abs, angle, sign, plot`

---

mlCA003.tex
Olkoon $z$:lla vaikkapa arvo $2 + 3i$ (yhtä hyvin joku muu). Suorita komennot ja selvitä aina itsellesi, mitä kukin tekee.

```
>> z=sign(z)
>> w=[z,z^2,z^3,z^4,z^5,z^6,z^7,x^8]
>> plot(w)
>> axis equal
```

Piirrä samaan kuvaan säteet origosta kuhunkin $w$-vektorin pisteeseen vaikka punaisella.
Vihje: Muista `hold on` ja sitten vaan annat plot:lle argumentiksi 2:n pituisia vektoreita ja käytät komentoeditoria (napsuttelet nuolinäppäintä ja editoit).

Mitä havaitset, jos ajattelet kompleksiluvun potensseja (vaikkapa De Moivreta)?

Matlab-tekniikkaa:
- Miksi ei tarvitse kirjoittaa `z.^2` ?
- Entä jos haluaisit piirtää potenssit `1:100` ? Miksi nyt täytyy kirjoittaa `.^` ?

Suorita Matlab:ssa: `help arith`
Lue:
`http://math.aalto.fi/~apiola/matlab/opas/ei-niin-lyhyt/osa2.html#luku4` matriisi- ja taulukko-operaatioesimerkit ja
`http://math.aalto.fi/~apiola/matlab/opas/lyhyt/perusteet.html#sec:matriisilaskenta`

**Vaativuus:** 1
**Tehtävän Latex-koodi:**
../mlteht/mlComplAnal/mlCA003.tex

**Ratkaisu:**
../mlteht/mlComplAnal/ratkaisut/mlCA003R.html (html)

**Avainsanat:**Matlabkompleksianalyysi,mlComplAnal,kompleksiluvun potenssit, De Moivre
**Matlabfunktioita:** sign

---

mlCA004.tex
Muodosta vektori, jossa on luvut $w_{n,k} = \sqrt[n]{1}, k = 0 \ldots n - 1$. n:n arvolla 10.

a) Käytä juurikaavoja.
b) Ratkaise (numeerisesti) polynomiyhtälö $z^n = 1$ ( `help roots`)

Piirrä yksikköympyrä ja samaan kuvaan kaikki ykkösen n:nnet juuret vaikka `'o'`-merkillä), missä vaikkapa $n = 10$. Kokeile eri n:n arvoilla.

Kirjoita Matlabin editorilla tiedosto `h1teht3.m` (tms.) ja tee siitä Matlab skripti tyyliin:

```
%% Harjoitus 1 tehtävä 3 , tiedosto h1teht3.m
% Nimi ja opno (harjoitukseksi myöhempiin tarpeisiin)
%
n=10    % Tätä voit vaihdella.
k=0:n-1 % kokonaislukuvektori.
w=...   % w-vekori, jossa nuo n:nnet juuret.
```

Skripti, jossa on Matlab-komentoja (ja selityksiä `%`-merkin takana), suoritetaan Matlabissa komentamalla `h1teht3` tai leikkaus/liimaus- menetelmällä. Nykyversioissa koko skripti tai sen osa suoritetaan suoraan Matlabin editorista CTR-ENTER-näppäilyllä.

**Vaativuus:** 1
**Tehtävän Latex-koodi:**
../mlteht/mlComplAnal/mlCA004.tex

**Ratkaisu:**
../mlteht/mlComplAnal/ratkaisut/mlCA004R.html

**Avainsanat:**Matlabkompleksianalyysi,mlComplAnal,kompleksijuuret
**Matlabfunktioita:** roots,plot

---

mlCA005.tex
Puhdas kynä/paperi-tehtävä. Hahmottele seuraavat alueet kompleksitasossa (z-tasossa) ja niiden kuva- alueet kuvauksessa $w = e^z$. Kiinnitä myös huomiota siihen, kuuluuko reuna mukaan. Tässä $z = x + iy$
(a) $-1 < x < 1, -\pi < y < \pi$.
(b) $0 \le y \le \pi/2$. (Mieti, kuuluuko 0 mukaan kuvaan.)
(c) $\pi < y \le 3\pi$.
(d) $\ln 3 < x < \ln 5$.

**Vaativuus:** 1
**Tehtävän Latex-koodi:**
../mlteht/mlComplAnal/mlCA005.tex

**Ratkaisu:**
../mlteht/mlComplAnal/ratkaisut/mlCA005R.html (myös tehtävään mlCA006)

**Avainsanat:** kompleksianalyysi,ComplAnal
**Matlabfunktioita:**

---

mlCA006.tex
Avaa
http://www.math.hut.fi/teaching/k3/03/L/CA1.html

ja sieltä kohta demoexp. Voit käytellä skriptiä leikkaamalla/liimaamalla omaan Matlab-työtiedostoosi ja sieltä komentoikkunaan. Tässä vaiheessa ei ole välttämätöntä ymmärtää kaikkia skriptin Matlab-komentoja, kunhan näet, miten sitä modifiomalla voit tehdä haluamiasi juttuja.

(a) Suorita ensin komennot ja katso, että saat samanlaiset kuvat kuin CA1.html:ssä.
(b) Havainnollista exp-funktiota katsomalla joidenkin suorakulmioalueiden kuvautumista.
(c) Voit myös helposti muuttaa skriptiä yleisluontoisemmaksi ottamalla x-vektorin pituuden n käyttöön ja kirjoittamalla vastaavanlaisen for-silmukan kuin y-vektorille on tehty.

**Vaativuus:** 1+
**Tehtävän Latex-koodi:**
../mlteht/mlComplAnal/mlCA006.tex

**Ratkaisu:**
../mlteht/mlComplAnal/ratkaisut/mlCA006R.html

**Avainsanat:**Matlabkompleksianalyysi,mlComplAnal,exp-funktio,eksponenttifunktio
**Matlabfunktioita:** linspace, exp, meshgrid,min,max,length

---

mlCA007.tex
(Puhdas käsinlasku)
Kompleksiluvulla $e^{i\alpha}$ kertominen suorittaa kierron kulman $\alpha$ verran. Kyseessä on tason $\mathbb{R}^2$ lineaarikuvaus, jolla niin ollen on matriisiesitys. (Muistele 1-kurssien asioita, lineaarikuvauksia käsitellään tälläkin kurssilla lähemmin.)

Johda kiertokuvauksen matriisiesitys muodostamalla tulo $w = e^{i\alpha}z$, $z = x + iy = re^{i\Theta}$

Ohje: Ei tarvitse muuta kuin kirjoittaa $e^{i\alpha}(x+iy)$ muotoon $Re+iIm$ ja samaistaa kompleksiluku $x + iy$ pystyvektorin $[x, y]^T$ kanssa.

Opetus: Kiertokuvauksia (ja eräitä muitakin tason lineaarikuvauksia) voidaan käsitellä erityisen kätevästi kompleksiaritmetiikan avulla. Matriisilaskujen sijasta voidaan harrastaa kompleksiaritmetiikkaa.

Seuraavassa tehtävässä harrastetaan tätä oikein olan takaa.

Olennaista on, että käytössä on ohjelma, joka osaa laskea kompleksisilla vektoreilla. Matlab on tällainen (myös Maple ja Mathematica).

**Vaativuus:** 1
**Tehtävän Latex-koodi:**
../mlteht/mlComplAnal/mlCA007.tex

**Avainsanat:**Matlabkompleksianalyysi,mlComplAnal,lineaarikuvaus, kiertokuvaus
**Matlabfunktioita:**

---

mlCA008
Alla on versio kuuluisan matemaatikon *Arnoldin* ns. kissaa, jonka toteutamme Matlabilla varsin

yksinkertaisena tyylitelmänä. Harjoituksen ajatuksena on demonstroida kompleksiaritmetiikan mahdollisuuksia kiertokuvauksien käsittelyssä. Samalla saamme rutiinia niin Matlabissa kuin yleensäkin kompleksiluvuilla laskemisessa.

Käsittelemme tässä kissaa tavallisuudesta poiketen kompleksilukuvektorina.

Tehtävässä ei ole jätetty juurikaan itse keksittävää. Niinpä jos aika on tiukalla, tämä tehtävä sopii oikein hyvin omatoimisesti läpikäytäväksi vaikka kotona.

Selvitä itsellesi juurta jaksain, mitä kussakin vaiheessa tehdään. Kissa ja sen pyöritys on pelkkää kompleksiaritmetiikkaa.

Komennot on annettu "ideointiyyliin", siksi koodia on niin paljon.

Suorita ensin nämä komennot: (Huomaa puolipisteen käyttö, jos dataa on "vähänkin paljon".)

```
clf              % clear graphics
t=0:pi/100:2*pi; % tai esim. t=linspace(0,2*pi);
paa=.1*exp(i*t);
plot(paa)
axis equal
hold on
silmat=[-0.05+i*.055, 0.05+i*.055]
plot(silmat,'+')
nena=.02*i+.003*exp(i*t);% t-vektori muodostettiin yllä.
plot(nena,'r')
%nena=.02*i
%plot(nena,'o')  % Tämä olisi ''laiskan miehen nenä''
% Suuksi sopiva ympyrän kaari välillä (-2*pi/3,-pi/3)
phi=linspace(-2*pi/3,-pi/3);
suu=0.05*exp(i*phi);
plot(suu,'.')
korvat=[.1*exp(i*pi/6),.1*exp(i*5*pi/6)]
plot(korvat,'o') % ''Laiskan miehen korvat''
```

Tässä on peruskissa.

Nyt ryhdymme pyörittelemään kissaparkaa. Päätä ei tarvitse pyörittää, ympyrä ei pyöritettäessä miksikään muutu. Riittää, kun pyöritämme suuta, nenää, silmiä ja korvia.

Kootaan ensin kissan osat yhteen vektoriin ja tehdään äskeinen uudestaan kissavektorilla.

```
figure(1)   % Tätä tarvitaan vain palattaessa takaisin kuvasta 2.
% Ajatellaan, että figure(1) on z-taso ja figure(2) w-taso.
clf              % Grafiikan putsaus
t=0:pi/100:2*pi; % Syytä tehdä uudestaan, vanha t voisi olla jo ihan muuta
paa=.1*exp(i*t);    % vaikkei näillä komennoilla satukaan.
plot(paa)
axis equal
```

```
hold on
zkissa=[silmat,nena,korvat,suu];
plot(zkissa,'or')  % Pelkät pisteet merkillä 'o' värillä 'r'
```

Avataan uusi grafiikkaikkuna ja piirretään siihen kierrettyjä kissoja.

```
figure(2)  % w-taso
clf
wkissa=exp(i*pi/4)*zkissa
paa=.1*exp(i*t);
plot(paa)
axis equal
hold on
plot(wkissa,'or')
shg     % show graphics
```

Nyt voit jatkaa kissaleikkiä alla olevaan tyyliin tai jotenkin muuten. Helpointa ja opettavaisinta on kirjoittaa Matlabin editorilla alla olevat rivit (ja mahdollisesti myös zkissarivit) omien mieltymystesi mukaan modifioiden tiedostoon `kissa.m`. Kun tiedosto on polun varrella, voit sanoa istunnossasi `kissa`, editoida `kissa.m`:ää ja taas komentaa kissaa.

```
clf
alpha=pi/4;  % Kiertokulma, helppo muutella.
wkissa=exp(i*pi/4)*wkissa
paa=.1*exp(i*t);
plot(paa)
axis equal
hold on
plot(wkissa,'or')
shg
```

Seuraavaksi jo sormet syyhyävät kissan käsittelyyn matriisina ja kiertomatriisilla kerrottuna (vrt. [D]-teht. 5). Maltamme kuitenkin mielemme ja jätämme seuraavaan kertaan.

Palannemme kissaan muutenkin yleisempien lineaarikuvausten yhteydessä, tällöin saatamme kaltoin kohdella sitä paljon enemmän, anteeksi kissa!

**Vaativuus:** 1
**Tehtävän Latex-koodi:**
../mlteht/mlComplAnal/mlCA008.tex

**Ratkaisu:**
Tehtävä on tyyppiä: “Tee, mitä käsketään”, ja ihastele/ihmettele/ota opiksi.

**Avainsanat:**Matlabkompleksianalyysi,mlComplAnal, Arnoldin kissa, lineaarikuvaus, kiertokuvaus kompleksiaritmetiikan avulla

**Matlabfunktioita:**exp,plot,hold on

---

mlCA009.tex
Matlab on ennenkaikkea matriisikieli. Jos kompleksiaritmetiikka sujuu käteväti, niin samoin on laita matriisilaskujen. Katsotaan siksi vielä, miten edellinen hoidettaisin matriisioperaatioin. **(Vrt. teht. 4)** HUOM HA:lle
Näin saadaan malli myös yleisemmille linaarikuvauksille, joihin liittyviä tehtäviä mlLinalg-hakemistossa.

Tällä kerralla esitämme kissan tavanomaisemmassa muodossa kaksirivisenä reaalisena matriisina, jossa kukin sarake edustaa kissan pistettä $\mathbb{R}^2$:ssa.

Olkoon `zkissa` kuten edellä.

```
alpha=pi/4;   % muuttele tarpeen mukaan.
A=[cos(alpha), -sin(alpha);sin(alpha),cos(alpha)] % Kiertomatriisi.
zkissa=[real(zkissa);imag(zkissa)]; % zkissasamaistus C <-> R^2
wkissa=A*zkissa; % Mahtavan kätevää on tämäkin. Kun zkissapisteet ovat
            % matriisin sarakkeina, niin kertomalla kiertomatriisilla A,
            % saadaan wkissapisteiden muodostama matriisi.
figure(1); clf
plot(zkissa(1,:),zkissa(2,:),'o') % Tämä on kaikkein tavallisin plot-
                                  % komennon muoto, kun data on reaalista.
axis equal
figure(2); clf
plot(wkissa(1,:),wkissa(2,:),'*r')
```

```
>> z=2+3*i
>> plot(z)
>> plot(z,'*')
>> axis([0 4 0 4])
>> hold on
>> plot([0 z])
>> abs(z)
>> angle(z)
>> atan(3/2)
```

**Vaativuus:** 1
**Tehtävän Latex-koodi:**
../mlteht/mlComplAnal/mlCA008.tex

**Ratkaisu:**
Tehtävä on tyyppiä: "Tee, mitä käsketään", ja ihastele/ihmettele/ota opiksi.

**Avainsanat:**Matlabkompleksianalyysi,mlComplAnal, Arnoldin kissa, lineaarikuvaus, kiertokuvaus

**Matlabfunktioita:**exp,plot,hold on

---

**2.** Jos $f$ on analyyttinen, sille pätee Cauchyn integraalikaava, josta muuttujanvaihdolla saadaan

$$f(z_0) = \frac{1}{2\pi i}\int_{|z-z_0|}\frac{f(z)}{z-z_0}dz = \frac{1}{2\pi i}\int_0^{2\pi}\frac{f(z_0-re^{it})}{z_0+re^{it}-z_0}dt$$
$$= \frac{1}{2\pi i}\int_0^{2\pi} f(z0+re^{it})dt.$$

Gaussin keskiarvoperiaatteen nojalla analyyttisen funktion arvo pisteessä $z_0$ on laskettavissa ottamalla integraalikeskiarvo reunan ylitse. Intuitiivisesti keskiarvo on aina arvojen maksimin alapuolella.

Tätä ajatusta noudattaen päästään seuraavaan: olkoon $f: U \to \mathbb{C}$ ei-vakio analyyttinen funktio alueessa $U$. Tällöin $z \mapsto |f(z)|$ ei periaatteen mukaan saavuta suurinta arvoaan alueessa $U$. Tutkitaan asiaa kokeellisesti:

1. Piirrä funktion $f(z) = |e^z|$ kuvaaja alueessa $[-1,1]\times[-i,i]$.
2. Piirrä funktion $f(z) = |\log(z)|$ kuvaaja alueessa $[1,10]\times[i,10i]$
3. Tutki vielä kuvauksen $z \mapsto |z^3|$ käyttäytymistä alueessaa $[-1,1]\times[-i,i]$.

Kuinka maksimiperiaate ilmenee näiden funktioiden tapauksessa?

Maksimiperiaate pätee myös harmonisille funktioille: jos $f = u+iv$ on analyyttinen funktio, joka (vähintään lokaalisti) saadaan annetusta harmonisesta funktiosta $u$, niin funktion

$$F(z) = e^{f(z)}$$

avulla saadaan maksimiperiaate pätemään funktiolle $u$, sillä $|F(z)| = e^u$. Tutki maksimiperiaatten toteutumista eksponenttifunktion reaali- ja imaginääriosille seuraavasti:

```
t  = -1:0.1:1;
[x, y] = meshgrid(t,t);
u = exp(x).*cos(y);
mesh(x,y,u);
v = exp(x).*sin(y);
mesh(x,y,v)
```

**Vihje:**

mlCAxxx.tex
Template file

Let

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

What can you say ...

**Vaativuus:** 1 - 3+
**Tehtävän Latex-koodi:**
../mlteht/mlLinalg/mlCAxxx.tex

**Avainsanat, keywords:** Kompleksianalyysi, Complaex analysis, Matlab

---

## mlCurveFit

mlCF000.tex
**Interpolaatio ja pienimmän neliösumman sovitus, PNS, LSQ**

Lineaarialgebra-osioissa on aiheesta myös joitakin perustehtäviä.

Tähän tiedostoon kootaan laskaripaperiin sopivia pikaohjeita ja mielen virkistyksiä.

- .
- ..
- ...

---

mlCF001
Johdatusta interpolaatioon.

Eräässä (keskilännen) kaupungissa tilastoitiin 3:n eri viikon aikana alkoholin kokonaiskulutus ja rattijuopumuspidätykset seuraavan taulukon mukaisesti.

| Viikko | Alkoholin kulutus(litraa) | RJ-pidätykset |
|---|---|---|
| 1 | 400 000 | 237 |
| 2 | 930 000 | 845 |
| 3 | 1 704 000 | 1356 |

Arvioi interpolaatiopolynomin avulla RJ-pidätysten lukumäärä viikolla, jolla kulutus olisi 1 230 000 litraa.

(a) Olkoon $p(x) = ax^2 + bx + c$. Muodosta annetun datan perusteella yhtälöryhmä, jossa tuntemattomina ovat kertoimet $a, b, c$. Kirjoita yhtälöryhmä matriisimuotoon: $Ax = B$ ja ratkaise Matlab:n takakenolla (\).

(b) Käytä polynomin arvon laskentaan `polyval`-funktiota

(c) Piirrä datapisteet ja interpolaatiopolynomin kuvaaja, sekä merkitse rinkulalla kysytty piste.

**Vihje**: Vastaisuuden varalle voit tutustua myös funktiohin `vander` ja `polyfit`.

Interpolaatiopolynomeja voidaan muodostaa tehokkaammin ja nokkelammin mm. *Lagrangen* ja *Newtonin* menetelmillä. Nämä voidaan helposti toteuttaa Matlab-ohjelmina. Palataan tuonnempana.

**Vaativuus** 1+

**Tehtävän Latex-koodi:**
../mlteht/mlCurveFit/mlCF001.tex

**Avainsanat:** mlCurvefit, interpolaatio, interpolation, lineaarinen yhtälöryhmä, Käyrän sovitusta/interpolaatiota Matlab:lla ,mlCurveFit,mlCF
**Matlabfunctions:** "Takakeno" (\) $A\backslash b$, polyval, (polyfit)
**Viitteitä:**

---

mlCF01.tex [Maple: ../../mplteht/mplCurveFit/mplCF01.tex]

**Opettajalle:** Tehtävä soveltuu hyvin Maple-ratkaisutekniikan opetteluun. Matlabin kohdalla kyse on yhtälöryhmän muodostamisesta (käsin) ja ratkaisemisesta Matlab:lla.

**Hermiten interpolaatio:** Interpolaatioehdoissa esiintyy myös derivaattoja.

**Huom:** 5 ehtoa ja 5 tuntematonta kerrointa $\Longrightarrow$ järkevän tuntuinen tehtävä. Yleisesti "järkevälläkään" Hermiten interpolaatiotehtävällä ei aina ole yksikäsitteistä ratkaisua (kuten ei neliömatriisin määräämällä lineaarisella yhtälöryhmälläkään – siitähän on kyse). Pelkkiä funktion arvoja koskevalla interpolaatiotehtävällä aina on (koska "Vandermonden neliömatriisi" on aina ei-singulaarinen).

**Vihje:**
Kirjoita käsin yhtälöryhmä muotoon $Ax = B$. Tuntemattomina ovat siis etsittävän polynomin kertoimet.

**Vaativuus:** 1+
**Tehtävän Latex-koodi:**
../mlteht/mlCurveFit/mlCF01.tex

**Ratkaisu:**
../mplteht/mplCurveFit/ratkaisut/mplCF01R.pdf (Tästä Maple-ratkaisusta näkyy oikea lopputulos.)

**Avainsanat:** Käyrän sovitusta/interpolaatiota Matlab:lla ,mlCurveFit,mlCF, lineaarinen yhtälöryhmä

**Matlabfunktioita:** `A\b` "backslash" , polyval

---

Oletetaan, että meille on annettu dataa muodossa $(x_k, y_k).k = 1 \ldots m$, johon muodustuu kaksi murtopisteen erottamaa lineaarista suuntausta. Esimerkiksi

```
x=-2:0.1:4; y=0.2*sin(3*x);
y(x<1)=y(x<1)+0.5*(x(x<1)-1);
y(x>=1)=y(x>=1)+2*(x(x>=1)-1);
```

muodostaa selvän murtopisteen kohtaan $x = 1$. Intuitiivisesti tuntuu selvältä, että tällaiseen dataan kannattaa sovittaa PNS-suoran sijaan paloittain lineaarinen funktio, ts. ”suora murtopisteellä.”

Kirjoita ohjelma joka tekee tämän: ohjelman tulee valita murtopiste $(s, t)$ tasosta hiiren klikkauksen perusteella (kts. vihje) ja sovittaa paloittain lineaarisen funktion dataan tätä murtopistettä käyttäen, ts. sovittaa suoran

$$y = k_1 x + b_1, x < s$$

pisteisiin $(x_k, y_k), x_k < s$ ja suoran

$$y = k_2 x + b_2, x > s$$

pisteisiin $(x_k, y_k), x_k > s$.

**Vihje:**
Tehtävän keskeinen osa on murtopisteen valinta ja datapisteiden suodatus.

Murtopisteen valintaan kannattaa käyttää `ginput` funktiota, joka valitsee klikatun pisteen kuvasta tyyliin

```
[x y] = ginput(1);
```

Datan suodatukseen kannattaa käyttää MATLABin loogista indeksointia: esimerkiksi valitaan kaikki vektorin  pisteet, jotka ovat pienempiä kuin 5.

```
a = b(b<5);
```

**Vaativuus:** 2
**Tehtävän Latex-koodi:**
../mlteht/mlCurveFit/mlCF017.tex

**Ratkaisu:**
../mlteht/mlCurveFit/ratkaisut/html/mlCF017R.html (publish: m->html)
../mlteht/mlCurveFit/ratkaisut/mlCF017R.m (m-tiedosto)

**Avainsanat:** Käyrän sovitusta/interpolaatiota Matlab:lla ,mlCurveFit,mlCF

**Matlabfunktioita:** polyfit, polyval

---

mlCF02 (Perus-Matlabtekniikkaa polynomi-interpolaatiotehtävään)
Muodosta interpolaatiopolynomi pisteistölle, joka saadaan laskemalla funktion $f(x) = \cos(1 +$

$x^2$) arvot tasavälisessä x-pisteistössä, jossa on 7 pistettä välillä $[0, 3]$. Piirrä samaan kuvaan funktio, datapisteet (rinkuloilla) ja interpolaatiopolynomi.

**Vihje**
help (tai doc) polyfit, polyval .
Sinun on tiedettävä, mikä on polynomin asteluku.
Tarkistus: Kulkeeko polynomi kaikkien datapisteiden kautta.

**Vaativuus:** 1+
**Tehtävän Latex-koodi:**
../mlteht/mlCurveFit/mlCF02.tex

**Ratkaisu:**
../mlteht/mlCurveFit/ratkaisut/html/mlCF02R.html
../mlteht/mlCurveFit/ratkaisut/mlCF02R.m

**Avainsanat:** Käyrän sovitusta/interpolaatiota Matlab:lla ,mlCurveFit,mlCF

**Matlabfunktioita:** polyfit, polyval

---

mlCF03 [Interpolaatiokertoimet lineaarisesta yhtälöryhmästä]
Kirjoita funktio, jonka otsikko ja "help-kommentit" voisivat olla:

```
function [kertoimet,condnr]=vandinterp(xdata,ydata)
% Funktio laskee interpolaatiopolynomin kertoimet Vandermonden
% systeemin ratkaisulla ja palauttaa my\"os cond-luvun.
%  Esim:
%  xdata=0:5; ydata=xdata.*sin(xdata);
%  [c,cnr]=vandinterp(xdata,ydata);
```

Laske vaikkapa kommenttiesimerkin tapaus ja vertaa polyfit-funktion antamiin kertoimiin. Piirrä data ja interpolaatiopolynomi. Käytä arvojen laskentaan polyval-funktiota.

(Opettajalle: **Helpotettu versio:** Jätetään cond pois.)

**Vihje**
Olkoon $p(x) = a_0 + a_1 x + \ldots + a_n x^n$ etsitty polynomi.
Määrätään tuntemattomat kertoimet interpolaatioehtojen $p(x_k) = y_k, k = x_0, \ldots x_n$ avulla saatavasta lineaarisesta yhtälösysteemistä ratkaisemalla.
**Huom:** Matlab:ssa polynomi esitetään kerroinvektorina:
$[a_n, a_{n-1}, \ldots, a_0]$ . Muista myös, että luku 0 ei kelpaa vektorin indeksiksi, toki tässä tehtävässä ei liene mitään tarvetta indeksointiin.

Kirjoita yhtälöryhmä tässä yleisessä muodossa ja tee ensin Matlab-skripti tyyliin

```
xd=...;
yd=...;
A=...;  % Yht.ryhman matriisi, help/doc vander
a=      % Ratkaisuna saatava kerroinvektori, help slash (a=A\...)d
x=linspace(alkup,loppup); % x-pisteet piirt. varten
y=polyval(...);           % Polynomin arvot x-pisteissa
...
plot(xd,yd,'o')
hold on
plot(x,y)
grid on
```

Tarkistus: Kulkeeko polynomi kaikkien datapisteiden kautta.
Kun skripti toimii, tee sen pohjalta pyydetty funktio.

Sovella funktiotasi johonkin tämän tehtäväkokoelman interpolaatiotehtävään.

**Vaativuus:** 2
**Tehtävän Latex-koodi:**
../mlteht/mlCurveFit/mlCF03.tex

**Ratkaisu:**
../mlteht/mlCurveFit/ratkaisut/vandinterp.m Funktio vandinterp
../mlteht/mlCurveFit/ratkaisut/html/mlCF03R.html Ajotiedosto html-muodossa (publish)
../mlteht/mlCurveFit/ratkaisut/mlCF03R.m (Ajo-m-tiedosto)

**Avainsanat:** Käyrän sovitusta/interpolaatiota Matlab:lla ,mlCurveFit,mlCF, Lineaarinen yhtälöryhmä, Vandermonden matriisi

**Matlabfunktioita:** A_ "backslash", polyval

---

mlCF03a
Tee Vandermonden matriisi useallakin eri tavalla.

... Tarkennetaan tehtävää ...

**Vaativuus:** 1+
**Tehtävän Latex-koodi:**
../mlteht/mlCurveFit/mlCF03a.tex

**Avainsanat:** Käyrän sovitusta/interpolaatiota Matlab:lla ,mlCurveFit,mlCF, Vandermonden matriisi

**Matlabfunktioita:** polyfit, polyval, vander

---

mlCF04
Eräs kemiallinen koe tuotti seuraavat datapisteet:

```
tdata=[-1 -0.960 -0.86 -0.79 0.22 0.5 0.93];      % Aikapisteet
ydata= [ -1 -0.151 0.894 0.986 0.895 0.5 -0.306]; % Reaktiotulokset
```

Tarkoitus on estimoida reaktiotulosfunktion $y(t)$ arvoja välillä $[-1, 1]$

(a) Piirrä datapisteet.

(b) Muodosta interpolaatiopolynomi ja piirrä samaan kuvaan.

(c) Muodosta asteita 2,3,4 olevat PNS-polynomit ja piirrä samaan kuvaan

(d) Sovita vielä splini ja piirrä samaan.

(e) Asettele grafiikkaikkuna paremmaksi vaikka tyyliin:

```
a=min(xdata)-0.1;b=max(xdata)+0.1;
c=min(ydata)-0.1;d=max(ydata)+0.1;
axis([a b c d])
```

Käytä myös `legend`-komentoa ja kokeile `grid on`

**Vaativuus:** 2-
**Tehtävän Latex-koodi:**
../mlteht/mlCurveFit/mlCF04.tex

**Avainsanat:** Käyrän sovitusta/interpolaatiota Matlab:lla ,mlCurveFit,mlCF

**Matlabfunktioita:** polyfit, polyval,"backslash"

---

mlCF05

a) Piirrä suorat $y = 0.2 - 0.5x, y = 0.2 + 2x, y = -0.3 + 1.5x$ ja $y = 0.95x$.

b) Määritä tason piste $(x, y)$, joka on PNS-mielessä lähinnä suorien "leikkauspistettä", ts. piste, jonka suorista laskettujen etäisyyksien neliösumma on minimaalinen.

**Vihje:**
Kirjoita suorien yhtälöt ensin muotoon $a_i x + b_i = y$, $i = 1 \dots 4$. Muodosta $4 \times 2$ - matriisi A tuntemattomien $x$ ja $y$ kertoimista ja vektori $b$ vakiotermeistä.
Vaikealta kuullostava tehtävä muuttuu helpoksi, koska Matlabin "takakeno" on ohjelmoitu ratkaisemaan ylimääräytyvä yhtälösysteemi PNS-mielessä.

Samaan kuvaan piirtäminen onnistuu komennolla `hold on`.
Pisteen $xy$ piirto punaisella rinkulalla menee näin (jos ratkaisusi on vektorissa $xy$):

```
plot(xy(1),xy(2),'or')
```

**Vaativuus:** 2
**Tehtävän Latex-koodi:**
../mlteht/mlCurveFit/mlCF05.tex

**Ratkaisu:**
../mlteht/mlCurveFit/ratkaisut/html/mlCF05R.html (Publish: m-tied.-> html)
../mlteht/mlCurveFit/ratkaisut/mlCF05R.m (Matlab:n m-tiedosto)

**Avainsanat:** Käyrän sovitusta Matlab:lla, PNS-sovitus/LSQ-fit, CurveFit,mlCF

**Matlabfunktioita:** "backslash", "takakeno"

---

mlCF07.tex/mplCF07.tex // Matlab,Maple,[Mathematica]

W.A Mozartin(1756-1791) sävellyksiä indeksoidaan Köchel-luvuilla, jotka ilmaisevat teosten sävellysjärjestyksen. Alla on eräitä Köchel-lukuja, ja vastaavien teosten sävellysvuosia.

```
 Number    Year
  1        1761
  75       1771
 155       1772
 219       1775
 271       1777
 351       1780
 425       1782
 503       1786
 575       1789
 626       1791
```

Käyttäen tätä dataa, arvioi teoksen Sinfonia Concertanten sävellysvuosi, kun tiedetään, että sen Köchel-numero on 364.

**Vihje:**
Piirrä ensin datapisteet tasoon, ja päätä millaista menetelmää kannattaa käyttää. Epäilemättä sopivan asteista PNS-polynomia. Suorita joitakin sovituksia, ja tarkista sitten tulos vaikka Wikipediasta.

**Vaativuus:** 1+
**Tehtävän Latex-koodi:**
../mlteht/mlCurveFit/mlCF07.tex

**Ratkaisu:**
../mlteht/mlCurveFit/ratkaisut/mlCF07R.pdf (Matlab:n m-tiedosto -> pdf (publish))
../mlteht/mlCurveFit/ratkaisut/mlCF07R.m (Matlab:n m-tiedosto)

**Avainsanat:** Käyrän sovitusta Matlab:lla ,mlCurveFit,mlCF

**Matlabfunktioita:** polyfit, polyval, “backslash”

---

mlCF09

a) Luo dataa seuraavalla skriptillä:

```
r = 0.5+0.5*rand(10,1);
theta =2*pi*rand(10,1)
x = 3*r.*cos(theta);
y = 3*r.*sin(theta);
```

ja piirrä data pisteittäin.

b) Sovitamme dataan ympyrän muodossa $(x-c_1)^2+(y-c_2)^2=r^2$. Ympyrän sovituksessa etsitään kahta arvoa: ympyrän keskipistettä $(c_1,c_2)$, ja sen sädettä $r$. Helpoimmin sovitus onnistuu huomaamalla, että $(x-c_1)^2+(y-c_2)^2=r^2 \Leftrightarrow 2xc_1+2yc_2+(r^2-c_1^2-c_2^2)=x^2+y^2$. Asettamalla $c_3=r^2-c_1^2-c_2^2$, saadaan yhtälö muotoon

$$2xc_1+2yc_2+c_3=x^2+y^2.$$

Tälle yhtälölle voidaan tehdä vaadittu datan sovitus, ja ratkaista arvot $(c_1,c_2,c_3)$, jonka jälkeen $c_3$sta ratkaistaan $r$.

**Vihje:**
Pisteittäinen piirtäminen onnistuu komennolla `plot(x,y,'.')`. Ympyrän, jonka keskipiste on $(x,y)$ ja säde $r$, voi piirtää komennolla `plot(x+r*cos(0:0.02:2*pi,y+r*sin(0:0.02:pi)))`.

**Vaativuus:** 2
**Tehtävän Latex-koodi:**
../mlteht/mlCurveFit/mlCF09.tex

**Ratkaisu:**
../mlteht/mlCurveFit/ratkaisut/html/mlCF09R.html (Publish: m-tied.-> html)
../mlteht/mlCurveFit/ratkaisut/mlCF09R.m (Matlab:n m-tiedosto)

**Avainsanat:** Käyrän sovitusta Matlab:lla PNS-sovitus/LSQ-fit, mlCurveFit,mlCF

**Matlabfunktioita:** polyfit, polyval, “backslash”

---

mlCF11
Maple tai Matlab
Tutkitaan nk. Rungen ilmiötä. Laske funktion $g(x)=1/(1+x^2)$ arvoja tasaisin välein väliltä

$[-5, 5]$, ja tee näihin pisteisiin perustuva polynomi-interpolaatio. Piirrä sekä $g(x)$ että $P(x)$ samaan kuvaan. Mitä huomaat, kun valittujen datapisteiden määrää tihennetään?

Kokeile interpolointia silloin, kun datapisteitä ei valita tasavälisesti, vaan ne valitaan Chebyshev-pisteiden

$$x_j = 5\cos(\frac{j\pi}{N}), j = 0 \ldots N$$

mukaan.

**Vihje:**
Polynomi-interpolaatio kannattaa tehdä MATLAB-funktiolla `polyfit`.
Funktio $g$ kannattaa määritellä funktiokahvan avulla: `g = @(x)1./(1+x.^2)`.
Tasavälisiä pisteistä saa funktiolla `linspace`
Chebyshev-pisteet muodostat tietysti vektorikomennoin (for-silmukka ankarasti kielletty).

Sopii aivan yhtä hyvin Maplelle.

**Vaativuus:** 2
**Tehtävän Latex-koodi:**
../mlteht/mlCurveFit/mlCF11.tex

**Ratkaisu:**
../mlteht/mlCurveFit/ratkaisut/html/mlCF11R.html (Publish: m-tied.-> html)
../mlteht/mlCurveFit/ratkaisut/mlCF11R.m (Matlab:n m-tiedosto)

**Avainsanat:** Käyrän sovitusta/interpolaatiota Matlab:lla ,mlCurveFit,mlCF, Rungen ilmiö, Chebycheff/Tsebyseff-pisteet

**Matlabfunktioita:** polyfit, polyval

**Opettajalle:** Tästä voi myös tehdä pikku tutkielman, jossa dokumentointi tehdään viimeisen päälle. Kuvissa voi käyttää selventäviä komentoja, kuten **legend**, taustaa voi hakea vaikka Wikipediasta, jne.

---

mlCF13.tex/mplCF13.tex
H2T14.tex/ Matlab,Maple,[Mathematica]

Yhdysvaltojen perustuslaki vaatii, että maassa suoritetaan joka kymmenes vuosi väestönlaskenta. Ohessa on väestönlaskennan tuloksia sadoissa miljoonissa asukkaissa viime vuosisadalta.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1900 | 1910 | 1920 | 1930 | 1940 | 1950 | 1960 | 1970 | 1980 | 1990 |
| 76 | 92 | 106 | 122 | 132 | 150 | 179 | 203 | 226 | 248 |

(a) Tee polynomi-interpolointi datalle, ja ennusta väestön määrä vuonna 2010. Kuinka ennusteesi suhtautuu laskennan todelliseen tulokseen: 308,745,538 laskettua asukasta?

**Vihje:** Tässä polyfit tuottaa liian paljon virhettä (johtuen ison Vandermonden matriisin häiriöalttiudesta), mikä näkyy heti kuvasta. (Miksiköhän?) Tehtävä on paremmin skaalattu, jos käytetään Lagrangen menetelmää. Tarvittava funktio *polyfit* on Molerin kokoelmassa: http://www.mathworks.se/moler/ncmfilelist.html ja myös tässä:
../mlteht/mlCurveFit/ratkaisut/polyinterp.m

**Opettajalle:** Tähän kohtaan voidaan myös ottaa ihan itse tehty Lagrangen interpolaatio, hts. teht. mlCF13a.

(b) Sovita eriasteisia PNS-polynomeja.

**Vaativuus:** 2-
**Tehtävän Latex-koodi:**
../mlteht/mlCurveFit/mlCF13.tex

**Ratkaisu:**
../mlteht/mlCurveFit/ratkaisut/html/mlCF13R.html (Publish: m-tied.-> html)
../mlteht/mlCurveFit/ratkaisut/mlCF13R.m (Matlab:n m-tiedosto)
../mlteht/mlCurveFit/ratkaisut/polyinterp.m (Funktio "polyinterp", lataa tästä)

**Viitteitä:**
Num. Comp. with Matlab, interpolation

**Avainsanat:** Käyrän sovitusta/interpolaatiota Matlab:lla ,mlCurveFit,mlCF, USA census, USA:n väestölaskenta

**Matlabfunktioita:** polyfit, polyval

---

mlCF13a
Lagrangen interpolaation ohjelmointi

Kirjoita funktio L, joka implementoi Lagrangen kertojapolynomin, ja tee sitä apuna käyttäen funktio *LagInterp*, joka muodostaa Lagrangen interpolaatiopolynomin. Kutsu voisi olla:

y=LagInterp(xdata,ydata,x), missä siis x on laskentapisteiden vektori.

Tätä voidaan ohjeistaa lisää tarpeen mukaan.

**Vaativuus:** 2+
**Tehtävän Latex-koodi:**
../mlteht/mlCurveFit/mlCF13a.tex

**Ratkaisu:**
Hiukan liiankin elegantti ratkaisu on Molerin:
../mlteht/mlCurveFit/ratkaisut/polyinterp.m
Helpommin vaheistettavia "omaleimaisia" ratkaisuja tulee (on jo, kunhan haetaan)

**Avainsanat:** Käyrän sovitusta/interpolaatiota Matlab:lla ,mlCurveFit,mlCF, Lagrangen interpolaatio, ohjelmointi

**Matlabfunktioita:** Oma LagInterp, polyval

---

mlCF15.tex [Maple: ../../mplteht/mplCurveFit/mplCF03.tex]

**Opettajalle**:
*(a)-kohta sopii ensitutustumiseen.*
*(b)-kohta on sikäli huono, että virhetermin suuruusluokka on toisesta maailmasta (opettavaista kylläkin, mutta alkajaisiksi vaatii ainakin varoituksen).*
*Lisää tehtävän opetuksia ratkaisutiedostoissa.*

(a) Muodosta interpolaatiopolynomi pisteistölle, joka saadaan laskemalla funktion $\cos(1+x^2)$ arvot tasavälisessä x-pisteistössä, jossa on 7 pistettä välillä $[0, 3]$.Piirrä samaan kuvaan funktio, datapisteet ja interpolaatiopolynomi.

(b) Arvioi (Lagrangen) interpolaatiokaavan virhetermin avulla interpolaatiovirheen yläraja yo. välillä ja vertaa todelliseen.

**Lause** Olkoot $x_0, x_1, \ldots, x_n$ erilliset pisteet ja $f$ $(n+1)$ kertaa jatkuvasti derivoituva funktio $x_k-$pisteet sisältävällä välillä. Jos $p_n$ on (1-käs) dataan $(x_k, f(x_k))$ liittyvä interpolaatiopolynomi, niin

$$f(x) - p_n(x) = \frac{f^{n+1}(\xi)}{(n+1)!}(x-x_0)(x-x_1)\cdots(x-x_n).$$

**Vihje:**
Tässä on mahdollista harrastaa Maplen ja Matlabin yhteistyötä. Virhekaavan derivaatta muodostetaan tietysti Maplella ja lauseke sievennetään. Itse asiassa piirtämällä ja poimimalla kuvasta maksimipisteen koordinaatit, saadaan riittävän hyvä arvio.
Toinen mahdollisuus on käyttää Matlabin symbolic toolboxia.

Tulotermin voisi hoitaa tehokkaimmin Matlabissa ottamalla tiheän diskretoinnin ja käyttämällä max-funktiota. Maplessakin on max-funktio, lakenta on Matlabissa tehokkaampaa.
Miten tulotermi lasketaan Matlabissa? Vaikka tähän tapaan:
1. `x=linspace(....,N)`
2. Tedään matriisi X, jossa x-vektoreita allekkain n+1 kpl.
3. Tehdään matriisi X0, jossa rivit

```
x0 x0 ... x0   N kpl.
x1 x1 ... x1   N kpl.
 ...
xn xn ... xn   N kpl.
```

Nämä syntyvät vaikka `meshgrid`-komennolla tai ulkotuloilemalla ykköspystyvektorilla.
4. Vähennetään matriisit ja `prod())`. Sitten vain `abs` ja `max` kehiin.

Tosi Matlabmaista! (Ei moitita, vaikka tekisit for-loopin, vain 8 kertaa käydään, mutta hyvä ymmärtää Matlabin hienoa matriisiajattelua, muistiahan ei nykyisin tarvitse säästellä.)

**Vaativuus:** 2+
**Tehtävän Latex-koodi:**
../mlteht/mlCurveFit/mlCF15.tex

**Ratkaisu:**
../mlteht/mlCurveFit/ratkaisut/html/mlCF15R.html (Publish: m-tied.-> html)
../mlteht/mlCurveFit/ratkaisut/mlCF15R.m (Matlab:n m-tiedosto)

**Avainsanat:** Käyrän sovitusta/interpolaatiota Matlab:lla ,mlCurveFit,mlCF, interpolaatiovirhe, Lagrange

**Matlabfunktioita:** polyfit, polyval

---

mlCF16.tex
Kuvitteellinen koe tuotti seuraavat tulokset. Tulosten perusteella tehtiin hypoteesi, jonka mukaan pisteet noudattelevat paloittain vakiota funktiota, jolla on yksi murtopiste, toisin sanoin, funktio, joka koostuu kahdesta vakio-osasta . Testaa hypoteesi sovittamalla paloittain vakio funktio dataan käyttämällä pienimmän neliösumman menetemää.

| t | b |
|---|---|
| 0.0 | 0.9 |
| 0.1 | 1.01 |
| 0.2 | 1.05 |
| 0.3 | 0.97 |
| 0.4 | 0.98 |
| 0.5 | 0.95 |
| 0.6 | 0.01 |
| 0.7 | -0.1 |
| 0.8 | 0.02 |
| 0.9 | -0.1 |
| 1.0 | 0.0 |

**Vihje:** Tehtävä kannattaa aloittaa graafisella tarkkailulla, ja määritellä silmämääräinen murtopiste. Tämän jälkeen on helppo muodostaa minimoitavat yhtälöt.

**Vaativuus:** 2
**Tehtävän Latex-koodi:**
../mlteht/mlCurveFit/mlCF16.tex

**Avainsanat:** Käyrän sovitusta/interpolaatiota Matlab:lla ,mlCurveFit,mlCF

**Matlabfunktioita:** polyfit, polyval

---

mlCF21.tex (Kirjasta *Fröberg*: Numerical Analysis 1985)
Joulukuun 1–28 päivänä 1981 aurinko laski *Lund*:ssa klo 15.30:n ja 15.45:n välillä seuraavan

taulukon mukaisesti, missä x tarkoittaa päivää ($1 \leq x \leq 28$) ja y minuuttimäärää klo 15.30:n jälkeen, jolloin aurinko laski.

| x | y | x | y |
|---|---|---|---|
| 1 | 8 | 19-21 | 3 |
| 2 | 7 | 22-23 | 4 |
| 3 | 6 | 24 | 5 |
| 4-5 | 5 | 25 | 6 |
| 6-7 | 4 | 26-27 | 7 |
| 8-9 | 3 | 28 | 7 |
| 10-18 | 2 | | |

Data voidaan esittää varsin hyvin 2. asteen polynomilla. Määritä kertoimet $a_0, a_1, a_2$, piirrä data ja polynomi. Minä päivänä auringonlaskuaika saavuttaa miniminsä mallin mukaan.

**Vaativuus:** 1+
**Tehtävän Latex-koodi:**
../mlteht/mlCurveFit/mlCF21.tex

**Avainsanat:** Käyrän sovitusta/interpolaatiota Matlab:lla ,mlCurveFit,mlCF

**Matlabfunktioita:** polyfit, polyval

---

mlCF30.tex [Moler NCM Exe 5.12 pp. 164-165]

Planeetan rata, neliömuoto

Tarkastellaan neliömuotoa:

$$q(x, y) = a\,x^2 + b\,x\,y + c\,y^2 + d\,x + e\,y + f.$$

Tason pistejoukko $\{(x, y) \mid q(x, y) = 0\}$ on *kartioleikkaus.*

Yhtälö voidaan normeerata jakamalla jollain nollasta poikkeavalla kertoimella.

Kartioleikkaus voidaan piirtää tyyliin:

```
>> x=linspace(xa,xb,M); y=linspace(ya,yb,N);
>> [X,Y]=meshgrid(x,y);
>> Z=a*X.^2 + b*X.*Y + ... ;
>> contour(x,y,Z,[0 0])  % Kts. help contour, miksei yksi 0 toimi.
```

Kartioleikkauksen tyyppi määräytyy diskriminantin $b^2 - 4ac$ merkistä.

(A)
Piirrä kutakin kolmea tyyppiä oleva kartioleikkaus valitsemalla kertoimet sopivasti.

(B)
Planeetat kiertävät tunnetusti ellipsirataa. Tässä on suoraan Matlab-muodossa annettu planeetan tasossa mitatut koordinaattipisteet:

```
>> x=[1.02 .95 .87 .77 .67 .56 .44 .30 .16 .01]':
>> y=[.39 .32 .27 .22 .18 .15 .13 .12 .13 .15]';
```

(a) Määritä neliömuodon kertoimet PNS sovituksella annettuun dataan asettamalla yksi kerroin arvoon 1 ja ratkaisemalla $10 \times 5$ yhtälöryhmä PNS-mielessä.
Piirrä neliömuoto ja datapisteet samaan kuvaan.

(b) Tämä tehtävä on lähellä singulaarista. Tutki, mitä pieni häiriö datassa vaikuttaa tulokseen lisäämällä pieni, välillä $[-0.0005, 0.0005]$ tasajakautunut satunnaisvirhe datapisteisiin. Piirrä alkuperäinen rata ja häiritsemällä saatu rata samaan kuvaan.
Kommenttisi kertoimien ja kuvan häiriintymisestä!

**Vihje**:

**Vaativuus:** 3 (sopii projektiteht.)
**Tehtävän Latex-koodi:**
../mlteht/mlCurveFit/mlCF30.tex

**Avainsanat:** Käyrän sovitusta Matlab:lla ,mlCurveFit,mlCF, neliomuoto, quadratic form, planetary orbit, planeetan rata, häiriöalttius

**Matlabfunktioita:** polyfit, polyval, meshgrid

---

## mlDiffint

mlDi013.tex
Määritä funktion $f(x) = \arcsin(2x\sqrt{1-x^2})$ suurin ja pienin arvo välillä $[-1, 1]$.

**Vihje:**
arcsin on Mathematicassa `ArcSin`, Maplessa `arcsin` ja Matlabissa `asin`.
Käytä symboliohjelmissa perinteistä “diffistekniikkaa” kuvan kanssa, Matlab:ssa raakaa “numeronmurskausta” tyyliin: `linspace, plot, zoom`, uusi `linspace` kapeammalla välillä, `find`, ...

**Vaativuus:** 1+
**Tehtävän Latex-koodi:**
../mlteht/mlDiffint1/mlDi013.tex

**Ratkaisu:**
../mlteht/mlDiffint1/ratkaisut/mlDi013R.pdf

../mlteht/mlDiffint1/ratkaisut/mlDi013R.m

**Avainsanat:** mlDiffint1,Matlabdiffint1, Differentiaali- ja integraalilaskentaa Matlab:lla, calculus1, analysis1, looginen indeksointi, minimi/maksimi

**Matlabfunktioita:** find, fminsearch

---

mlDi030.tex

Muista, että funktion $f$ derivaatta pisteessä $x_0$ määritellään seuraavasti:

$$f'(x_0) = \lim_{h \to 0} \frac{f(x_0 + h) - f(x_0)}{h}.$$

Kuinka laskisit derivaatan numeerisesti? Kirjoita MATLAB funktio joka laskee annetun funktion derivaatan.

Kokeile laskea funktion $f(x) = \sin(x)$ ja vertaa saamaasi tulosta derivaatan tarkkaan arvoon $f'(x) = \cos(x)$. Tuottaako pienempi $h$:n arvo parempia tuloksia?

Keksitkö keinoa jolla laskea toinen derivaatta numeerisesti? Entä vektorifunktioiden derivointi?

**Vihje** Muista, että alkioittaiset operaatiot ilmoitetaan pisteellä.

**Huom!** Numeerinen derivointi on numeerisesti epästabiili ongelma johtuen siitä, että jatkuvan funktion erotusosamäärä on rajalla tyyppiä $\frac{0}{0}$. Jos $h$ on lähellä likulukulaskennan suhteellista virherajatarkkuutta, Matlab:ssa muuttuja `eps`, niin tulos voi mennä kohti metsikköä. Tätäkin voit tutkia lähemmin.

**Opettajalle:** Tehtävästä kannattaa poimia tarpeen mukaan ja sopivasti muokaten soveltuvia osia.

**Vaativuus:** 1+...2
**Tehtävän Latex-koodi:**
../mlteht/mlDiffint1/mlDi030.tex

**Avainsanat:** mlDiffint1,Matlabdiffint1, Differentiaali- ja integraalilaskentaa Matlab:lla, calculus1, analysis1, differenssiapproksimaatio, erotusosamaara, erotusosamäärä, numeerinen derivaatta

**Matlabfunktioita:** diff

---

mlDi05.tex

**Ohjelmat**:
Maple,Mathematica , Matlab (erityisesti b)-kohta).

(Kurssi: 2012 kevät H/H2T15.tex)

Laske integraali

$$\int_0^{2\pi} \frac{\cos x}{13 - 12\cos 2x}\,dx$$

a) symbolisesti, b) numeerisesti. Piirrä integroitavan funktion kuvaaja. Mikä itse asiassa on integraalin arvo?

**Vihje**:

**Mathematica**:
Symbolinen integrointi tapahtuu funktiolla `Integrate`, numeerinen funktiolla `NIntegrate`. Jälkimmäisessä sovelletaan suoraan jotakin numeerisen integroinnin menetelmää, jonka valintaan myös käyttäjä voi vaikuttaa. Ks. dokumentaatiota, erityisesti Implementation Notes.

**Maple:**
Symbolinen integrointi tapahtuu funktiolla `int`, numeerinen funktiolla `int(...,type=numeric) tai evalf(Int(...))`. Jälkimmäisessä sovelletaan suoraan jotakin numeerisen integroinnin menetelmää, jonka valintaan myös käyttäjä voi vaikuttaa.

**Matlab**:
Integrandi määritellään funktioksi (helpoimmin funktiokahvaksi). Sitten `help integral, doc integral`, myös `quad`-alkuiset. Voidaan myös integroida symbolisesti (symbolic toolbox tarvitaan, on Aallossa): syms x, int(f(x),a,b)

**Vaativuus:** 1+
**Tehtävän Latex-koodi:**
../mlteht/mlDiffint1/mlDi05.tex

**Ratkaisu:**
../mlteht/mlDiffint1/ratkaisut/mlDi05R.pdf
../mlteht/mlDiffint1/ratkaisut/mlDi05R.m

**Avainsanat:** mlDiffint1,Matlabdiffint1, Differentiaali- ja integraalilaskentaa Matlab:lla, calculus1, analysis1

**Matlabfunktioita:** integral, quad.., syms x, int(f(x),a,b)

---

mlDi051.tex [Moler NCM Probl. 6.6 p. 180]

“Error function”, *erf* määritellään integraalina:

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt.$$

Tunnetusti integrointia ei voida suorittaa ns. “suljetussa muodossa”, mutta sille on kehittyneissä matemaattisissa ohjelmistoissa luotettava numeerinen laskentaohjelmansa, niin myös Matlab:ssa.

Käytä Matlab:n numeerista integrointifunktiota *integral* taulukoidaksesi *erf*-funktion arvoja $x$:n arvoilla $0.1, 0.2, \ldots 1.0$. Vertaa Matlab:n *erf*-funktion arvoihin samoissa pisteissä.

**Vihje**: help integral (help quad), help erf

**Vaativuus:** 2
**Tehtävän Latex-koodi:**
../mlteht/mlDiffint1/mlDi051.tex

**Avainsanat:** mlDiffint1,Matlabdiffint1, Differentiaali- ja integraalilaskentaa Matlab:lla, numeerinen integrointi, erf

**Matlabfunktioita:** integral, erf

---

mlDixxx.tex
Template file
Let

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

What can you say ...

**Vaativuus:** 1 - 3+
**Tehtävän Latex-koodi:**
../mlteht/mlLinalg/mlDixxx.tex

**Avainsanat, keywords:** Matlab, Diffint1, yhden muuttujan diffint, differentiqal calculus, functions of one variable

---

-e

## mlVektori

**3.** Muista, että Jacobin matriisi koostuu vektori- tai skalaariarvoisen funktion $F$ ensimmäisistä osittaisderivaatoista:

$$J_F = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \ldots & \frac{\partial y_1}{\partial x_1} \\ \vdots & \ldots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \ldots & \frac{\partial y_1}{\partial x_n} \end{bmatrix}.$$

Kuinka laskisit numeerisen Jacobin matriisin?

**Vihje:** Muista, että numeerisesti

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h},$$

Harjoitellaan usean muuttujan funktioiden piirtämistä. MATLABissa tämä tehdään määrittelemällä ensin piirtoalueen peittävä diskretointi, tai hila. Tämä määritellään komennolla `meshgrid`. Luodaan esimerkiksi suorakulmion $[1,2] \times [3,4]$ peittävä hila:

```
t1 = 1:0.2:2;t2 = 3:0.2:4;
[x y] = meshgrid(t1,t2);
```

Nyt muuttujissa `x` ja `y` ovat hilapisteiden $x-$ ja $y$-koordinaatit. Niillä voidaan suorittaa laskutoimituksia kuten tavallisillakin muuttujilla; on vain pidettävä mielessä että kyseessä ovat nyt matriisit, eli yleensä haluamme valita alkioittaiset operaatiot matriisioperaatioiden sijaan.

Pintojen piirtäminen tapahtuu komennoilla `surf` ja `mesh`. Lasketaan ja piirretään funktion

$$f(x,y) = x\,y^2$$

kuva:

```
z = x.*y;
% Huomaa alkoittainen kertolasku
surf(x,y,z);
```

.

Piirrä näillä keinoin seuraavien funktioiden kuvat

- $f(x,y) = \sqrt{1 - x^2 - y^2}$ alueessa $[-\frac{1}{2}, \frac{1}{2}] \times [-\frac{1}{2}, \frac{1}{2}]$
- $f(x,y) = x^3 - 3xy$ alueessa $[-1,1] \times [-1,1]$.

Toinen yleinen tapa kuvata usean muuttujan funktioita on käyttää tasa-arvokäyriä. Näitä piirretään komennolla `contour`, syntaksi on sama kuin tavallisilla piirtokomennoilla. Kokeile piirtää myös edellisten funktioden tasa-arvokäyrät.

**4.** Vektorianalyysissä on todistettu että funktion tasarvo-käyrä on aina gradientin normaali. Tutkitaan tätä nyt käytännössä. Piirrä funktiolle

$$f(x, y) = e^{-x^2-y^2}$$

tasa-arvokäyrät haluamassasi hilassa käyttäen komentoa `contour` (komennon syntaksi on sama kuin tavallisten piirtokomentojen). Tämän jälkeen laske funktion numeerinen gradientti tässä hilassa funktiolla `gradient`. Funktio palauttaa tässä tapauksessa kaksi matriisia, joissa on pisteittäiset numeeriset estimaatit arvoille

$$\frac{\partial f}{\partial x} \text{ ja } \frac{\partial f}{\partial y}.$$

Gradientit kannattaa piirtää vektorikenttänä, MATLABin tapauksessa komennolla `quiver`. Esimerkiksi, jos halutut osittaisderivaatat ovat matriiseissa `dx` ja `dy` niin piirto tapahtuisi komennolla `quiver(x,y,dx,dy)`. Lopuksi piirrä kuvat näkyviin päällekäin komentamalla:

```
contour(x,y,z)
hold on
quiver(x,y,dx,dy)
```

Tutki kuvasta, ovatko gradienttinuolet suorassa kulmassa tasa-arvokäyriä vastaan.

**5.** Funktio

```
function Jf = numjaco(F,m,x,n)
% f is a function with m components,
% x is a vector with n components,
% the result is an m by n matrix.
Jf = ones(m,n);   h = 1e-4;
f = fcnchk(F);
for j =1:n
    e = zeros(n,1); e(j) = 1;
    Jf(:,j) = (f(x+h*e)-f(x-h*e) )/(2*h);
end;
```

laskee vektorikentän $F$ numeerisen Jacobin matriisin, olettaen että käytetty kanta on standardi. Jos vektorikentässä on vain yksi komponentti, kyseessä on kyseisen funktion gradienttivektori.

Kopioi edellä oleva funktio tiedostoon, ja määrittele funktiot

- $f(x,y) = x^2y$
- $g(x,y) = xe^{x^2+y}$

ja laske niiden numeeriset gradientit pisteissä $(1,2)$ ja $(2,-4)$. Vertaa tulosta tarkkoihin arvoihin:

$$\nabla f = \begin{bmatrix} 2xy \\ x^2 \end{bmatrix} \nabla g = \begin{bmatrix} e^{x^2+y} + 2x^2e^{x^2+y} \\ xe^{x^2+y} \end{bmatrix}$$

**6.** Piirrä funktion $f(x,y) = e^{-x^2-y^2}$ määrittelemä pinta $(x,y)$-tason neliön $[-2,2] \times [-2,2]$ alueella. Laske tämän funktion integraali yli kyseisen neliön. Kyseisen integraalin arvo koko tason ylitse on $\pi$.

**Vihje:** Pinnan piirtämiseen tarvitset sopivan alueen peittävän hilapisteistön. Näitä luodaan MATLABissa komennolla `meshgrid`, katso `help meshgrid`. Pinta piirretään komennolla `mesh` tai `surf`, katso dokumentaatiota.

Integraalin laskemiseen kannattaa käyttää funktiota `dblquad`. Katso kutsumisohjeet dokumentaatiosta. `dblquad`in ensimmäisen argumentin tulee olla funktio: tässä kannattaa käyttää MATLABin anonyymifunktioita. Esimerkiksi $\int_0^3 \int_2^3 xy\,dx\,dy$ = `dblquad(@(x,y)x.*y,2,3,0,1)`. Jälleen kerran, tutki dokumentaatiota `function_handle`.

mlDiV11.tex

Laske funktion $f(x,y) = 3 + \cos(x) + \cos(y)$ määräämän pinnan ja suorakulmion $[0,\pi] \times [0,2\pi]$ väliin jäävä tilavuus. Piirrä havainnollistava kuva.

**Vihje:** Tilavuuteen tarvitset kaksinkertaista integraalia `dblquad`, pinta piirretään komennoilla `meshgrid` ja `mesh`.

**7.** Piirrä funktion $f(x,y) = y^2 - x^2$ tasa-arvokäyrät ja gradienttivektorikenttä, ja totea, että gradientti on aina kohtisuorassa tasa-arvokäyrää vasten.

**Vihje:** Laske gradientti funktiolla `gradient`, ja piirrä tasa-arvokäyrät komennolla `contour`. Gradienttivektorikentän saat piirrettyä komennolla `quiver`.

**8.** Määritä pinnan $z = -x^2 - y^2$ normaalivektori, ja piirrä sen vektorikenttä samaan kuvaan pinnan kanssa.

**Vihje:** Aloita määrittelemällä jokin tason suorakulmio funtion `meshgrid` avulla - tämän jälkeen voit määritellä pinnan, eli matriisin $z$. Pinnan normaalivektorin saat laskettua funktiolla `surfnorm`. Piirrä ensiksi pinta komennolla `surf(x,y,z)`, sen jälkee kirjoita `hold on`, jonka jälkeen kirjota `quiver3(x,y,z,u,v,w)`, missä $u, v$ ja $w$ ovat `surfnorm`in palauttamat matriisit.

mlDiVxxx.tex
Template file
Let

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

What can you say ...

**Vaativuus:** 1 - 3+
**Tehtävän Latex-koodi:**
../mlteht/mlLinalg/mlDiVxxx.tex

**Avainsanat, keywords:** Usean muuttujan funktioiden diffintlaskentaa, functions of several variables, vektoridiffintlaskentaa Matlab:lla, vector functions

## mlFseries – Matlab-tehtäviä funktiosarjoista

mlFS001
Seuraavat $2\pi$-jaksoiset funktiot on annettu jakson pituisella välillä (ja siten koko reaaliakselilla). Hahmottele niiden kuvaajat (parin jaksovälin alueella) ja selvitä, mitkä ovat parillisia, parittomia tai ei kumpaakaan.

(a) $f(x) = x^2, \quad 0 < x < 2\pi$, (b) $f(x) = e^{-|x|}, \quad -\pi < x < \pi$,
(c) $f(x) = \begin{cases} x, & -2 < x < 2 \\ 0, & 2 < x < 2\pi - 2. \end{cases}$

mlFSxxx.tex
Template file
Let

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

What can you say ...

**Vaativuus:** 1 - 3+
**Tehtävän Latex-koodi:**
../mlteht/mlLinalg/mlFSxxx.tex

**Avainsanat, keywords:** Matlab: Funktiosarjat, Fourier-sarjat, Fourier series with Matlab

-e

## mlGrafiikka

mlGr01 – Demo

Piirrä samaan kuvaan funktioiden cos ja sin kuvaajat välillä $[-2\pi, 2\pi]$ Aloita tyyliin:

```
x=linspace(-2*pi,2*pi); y1=cos(x); y2=sin(x);
plot(...)
```

**Vihje**
Voit piirtää molemmat yhdellä `plot`-komennolla tai käyrän kerrallaan "jäädyttämällä" vanhan grafiikan `hold on`-komennolla.

Jos haluat kuvat eri grafiikkaikkunoihin, voit käyttää `figure`-komentoa. Toisinaan on kätevää jakaa grafiikkaruutu osiin. Tämä onnistuu `subplot`:n avulla. Kokeile näitä vaihtoehtoisia tapoja (nyt tai myöhemmin).

**Vaativuus:** 1
**Tehtävän Latex-koodi:**
../mlteht/mlGraphics/mlGr01.tex

**Ratkaisu:**
../mlteht/mlGraphics/ratkaisut/html/mlGr01R.html Kauniissa html-muodossa
../mlteht/mlGraphics/ratkaisut/mlGr01R.m (m-tiedosto)
**Demo:**
../mlteht/mlGraphics/demo/html/mlGr01D.html

**Avainsanat:**Matlabgrafiikka,mlGraphics

**Matlabfunktioita:** linspace, plot, hold on, figure, subplot,

---

mlGr02
Piirrä

1. $xe^{-x^2}$ välillä $[-2, 2]$

2. $1/(1+x^2)$ välillä $[-4, 4]$

**Vihje:** Muista pisteet laskutoimituksissa!

**Vaativuus:** 1-
**Tehtävän Latex-koodi:**
../mlteht/mlGraphics/mlGr02.tex

**Ratkaisu:**
../mlteht/mlGraphics/ratkaisut/html/mlGr02R.html Publish -> html-muodossa
../mlteht/mlGraphics/ratkaisut/mlGr02R.m (m-tiedosto)

**Avainsanat:**Matlabgrafiikka,mlGraphics

**Matlabfunktioita:** linspace, plot `.*` `.^` `./`

---

mlGr03
Piirrä $\sin(2x)$ sinisellä ja $cos(5x)$ punaisella, välillä $[-\pi, \pi]$ (samaan kuvaan). Merkitse vielä samaan kuvaan $\sin(2x)$:n arvot o-merkeillä x-pisteissä

$$-\pi, -\pi + h, \ldots - \pi + 2h, \ldots, \pi, \text{kun}\, h = \pi/8.$$

**Vihje:**
Pane merkille tällaiset grafiikan ulkoasua säätelevät lisäkomennot (jotka voidaan antaa jälkikäteen):
`grid on/off, hold on/off, axis, xlim, ylim, figure, subplot, shg , close all`
Tutki toiminta help:stä ja oppaista. Aloita: `help plot` (tai klikkaa: *plot*).

Suorita joitakin kokeiluja (mutta älä uuvuksiin asti tässä vaiheessa vielä).

**Vaativuus:** 1
**Tehtävän Latex-koodi:**
../mlteht/mlGraphics/mlGr03.tex

**Ratkaisu:**
../mlteht/mlGraphics/ratkaisut/html/mlGr03R.html html-muodossa
../mlteht/mlGraphics/ratkaisut/mlGr03R.m (m-tiedosto)

**Avainsanat:**Matlabgrafiikka,mlGraphics

**Matlabfunktioita:** plot, hold on, axis,shg, close all

---

mlGr04
Piirrä samaan kuvaan funktioiden $\sin k\,x$ kuvaajat välillä $[0, 2\pi]$, kun $k = 1 \ldots 5$.

a. Tee komentoikkunassa nuolinäppäintä ( $\uparrow$ ) hyödyntäen (tai editorissa työskennellen) ja `hold on`-komentoa käyttäen.

b. Kirjoita pieni `for`-silmukka.

c. Muodosta 5-sarakkeinen matriisi, jonka $k:s$ sarake on $\sin kx$, missä $x$ on "x-vektori" .

**Vihje:**
Muodosta ensin $100\times 5$-matriisi , jonka sarakkeet ovat $k\,x$, $\;k=1\ldots 5$. Kätevimmin matriisikertolaskulla `x*K`, missä $x$ on (100-pituinen) sarakevektori ja $K$ (5-pituinen) indeksi(rivi)vektori. Mieti huolellisesti, miksi!
Toinen mahdollisuus on käyttää `meshgrid`-komentoa, jonka käyttöön rutinoidutaan 3d-grafiikan yhteydessä.

**Vaativuus:** 1+
**Tehtävän Latex-koodi:**
../mlteht/mlGraphics/mlGr04.tex

**Ratkaisu:**
../mlteht/mlGraphics/ratkaisut/html/mlGr04R.html html-muodossa
../mlteht/mlGraphics/ratkaisut/mlGr04R.m (m-tiedosto)

**Avainsanat:**Matlabgrafiikka,mlGraphics

**Matlabfunktioita:** plot, for, 'ulkotulo'

---

mlGr05
Funktio `plot` sovellettuna matriisiin piirtää matriisin sarakkeet rivi-indeksin funktiona.
Varsin käyttökelpoinen muoto on `plot(x,Y)`, jossa `x` on `Y`:n sarakkeiden pituinen argumenttivektori.

Suorita seuraavat komennot:

```
x=linspace(-1,1);
V=vander(x);
plot(x,V); shg
```

Jatka tähän tapaan:

```
figure   % Avaa uusi grafiikkaikkuna.
V=fliplr(V);
W=V(:,1:10);
plot(x,W);shg
```

Selitä, mitä näissä tapahtuu.

**Vaativuus:** 1
**Tehtävän Latex-koodi:**
../mlteht/mlGraphics/mlGr05.tex

**Avainsanat:**Matlabgrafiikka,mlGraphics

**Matlabfunktioita:** plot(x,A)

---

mlGr06
Piirrä samaan kuvaan potenssit $x^1, x^2, \ldots, x^n$, missä $n$ on muuteltava parametri. Käytä m-tiedostoa (skriptiä) seuraavan ohjeen mukaisesti.
Avaa uusi m-tiedosto ( `FILE`-valikosta `open->new->script` ) ja talleta se vaikkapa nimelle potenssipiirto.m .
Tai kirjoita komentoikkunassa: `>> edit 'potenssipiirto.m'`
Aloita tiedosto jotenkin näin:

```
%%  potenssipiirto.m.
% Laatinut N.N.
close all % Grafiikkaruudun tyhjennys
n=5;      % Muuteltava parametri
...
```

Talleta ja kirjoita komentoikkunaan:

```
>> potenssipiirto
```

Tällöin tiedostossa olevat Matlab-komennot suorittuvat.

Komennot suorittuvat myös editori-ikkunasta `CTR-ENTER` :llä. (Mac:ssä yleisesti `CTR`:n sijasta `cmd`.)

(Vihreä nuoli tai `F5` toimivat myös.)
Suorita skripti muutamalla eri n:n arvolla

a. Tee `for`-silmukka ja käytä `hold on`-komentoa uuden kuvan piirtämiseksi vanhan kaveriksi.

b. Olkoon aluksi vaikka $n = 3, m = 7$, missä $m$ on $x$-vektorin pituus. Muodosta matriisit `N` ja `X`, missä `N` koostuu vakiosarakkeista $1, 2, 3$ ja `X` saadaan latomalla kolme $x$-saraketta rinnakkain. Tällöin `X.^N` on matriisi, jonka sarakkeina ovat `x`-vektorin potenssit $1, 2, 3$. Kuva saadaan nyt komennolla `plot(x,X.^N)`. (Yleisesti: `plot(x,Y)` piirtää kunkin Y-matriisin sarakkeen x:n toimiessa x-akselina, kun `x` on `Y`:n sarakkeiden pituinen vektori. (Toimii myös riveittäin, jos x on rivien pituinen.)

   Miten saadaan helpoimmin matriisit `X, N` ? Standarditapa on tämä:

   ```
   >> nind=1:3;
   >> [N,X]=meshgrid(nind,x);
   ```

   Suorita ja selvitä itsellesi.

   Tee sitten esim. 100-pituinen x-vektori ja vaihtele myös n:ää ja piirrä sileitä kuvia.
   Lopuksi voit kokeilla, miltä näyttää `mesh(nind,x,X.^N)` .

   Huom! Tällainen `meshgrid`-komennon käyttö on rutiinitoimenpide 3d-grafiikan tekemisessä, sen toimintaperiaate on mukava ymmärtää, sitä tämä yrittää palvella.

c. Helpoin tapa lienee Vandermonden matriisi `vander`. Siitäpä on eri tehtävä (05), mutta ei ole huonoa harjoitella tässäkin uudestaan.

**Vaativuus:** 2-
**Tehtävän Latex-koodi:**
../mlteht/mlGraphics/mlGr06.tex

**Ratkaisu:**
../mlteht/mlGraphics/ratkaisut/html/mlGr06R.html Publish -> html-muodossa
../mlteht/mlGraphics/ratkaisut/mlGr06R.m (m-tiedosto)

**Avainsanat:**Matlabgrafiikka,mlGraphics

**Matlabfunktioita:** plot, for, meshgrid, repmat

---

mlGr07

**Aihe:** 3D-grafiikkaa, korkeuskäyriä

Olkoon

$$f(x, y) = \sin(3\,y - x^2 + 1) + cos(2\,y^2 - 2\,x).$$

Piirrä pintakuva ja korkeuskäyräpiirros, jälkimmäinen sekä `contour` että `ezcontour`-funktioilla. Tässä on mahdollisuus kokeilla korkeuskäyrien valitsemistapoja, myös `clabel`. Ota alueeksi vaikka `[-2 2 -1 1]` .

**Vihje**
Opiskele:
http://math.aalto.fi/~apiola/matlab/opas/lyhyt/grafiikka.html#sec:3d
Matlab-help: `doc mesh, doc surf, doc contour`

---

**Vaativuus:** 1+
**Tehtävän Latex-koodi:**
../mlteht/mlGraphics/mlGr07.tex

**Ratkaisu:**
../mlteht/mlGraphics/ratkaisut/html/mlGr07R.pdf Publish -> pdf-muodossa
../mlteht/mlGraphics/ratkaisut/mlGr07R.m (m-tiedosto)

**Avainsanat:**Matlabgrafiikka,mlGraphics

**Matlabfunktioita:** plot, subplot, zoom

---

mlGr08.tex

Olkoon

$$f(x) = \left( \frac{1 + \frac{x}{24}}{1 - \frac{x}{12} + \frac{x^2}{384}} \right)^8$$

Tämä on exp-funktion rationaaliapproksimaatio, ns. Pade-approksimaatio. Piirrä kuvaaja välillä [0,4].
Piirrä samaan kuvaan exp-funktio eri värillä ja eri kuvaan erotus $f(x) - exp(x)$ Aloita vaikka: `x=linspace(0,4,200);`

**Vihje:**
Tehtävässä harjoitellaan lausekkeen muodostamista pisteittäisin laskutoimituksin.
Homma selkeytyy jakamalla pienempiin osiin, ainakin nyt tällaisiin:

```
>> x=...;
>> osoittaja=...;
>> nimittaja=...;
>> f=...;  % Huomaa: f on muuttuja (200-pituinen vektori), ei funktio.
>>         % Tässä ei siten saa kirjoittaa: f(x) = ...
```

**Opettajalle:** Tästä voisi tehdä jatkotehtävän tyyppiä: Vertaa Taylorin sarjaa ja Pade-approksimaatiota. Ja vielä: voisi vaikka opettaa, miten Pade-approksimaatioita muodostetaan.

**Vaativuus:** 1+
**Tehtävän Latex-koodi:**
../mlteht/mlGraphics/mlGr08.tex

**Ratkaisu:**
../mlteht/mlGraphics/ratkaisut/mlGr08aR.m (m-tiedosto, vihjeen mukainen)
../mlteht/mlGraphics/ratkaisut/mlGr08bR.m (m-tiedosto, määritellään funktioksi isolla itseluottamuksella ilman välivaiheita)

**Avainsanat:**Matlabgrafiikka,mlGraphics,pisteittäiset laskutoimitukset vektoreilla

**Matlabfunktioita:** plot, `.* .^ ./`

---

mlGr09
Olkoot $c$ ja $z_0$ kompleksilukuja. Tällöin rekursion

$$z_n = z_{n-1}^2 + c$$

määräämä dynaaminen systeemi tunnetaan kvadraattisena kuvauksena. Valituille luvuille $c$ ja $z_0$ ylläoleva rekursio johtaa kompleksiseen lukujonoon $z_1, z_2, z_3 \ldots$. Tätä jonoa kutsutaan $z_0$:n kiertoradaksi. Riippuen lukujen $c$ ja $z_0$ valinnasta ratojen muotoja on useita.

Annetulle kiinteälle luvulle $c$ useimmilla $z_0$ rata lähestyy ääretöntä (eli $|z_n|$ kasvaa rajatta kun $n \to \infty$.) Joillakin $c$ ja $z_0$ rata kuitenkin suppenee kohti jotain periodista silmukkaa (eli arvot kiertävät $z_0$ jollain tietyllä etäisyydellä $|z_n|$); joillakin alkuarvoilla rata on kaoottinen. Nämä alkuarvot $z_0$ ovat kuvauksen Julia-joukko.

Tässä harjoituksessa kirjoitetaan MATLAB-ohjelma, joka laskee ns. täytetyn Julia-joukon, joka koostuu niistä alkioista $z_0$ joiden radat jollain annetulla arvolla $c$ eivät kasva rajatta – tavallinen Julia-joukko on tämän joukon reuna.

On näytetty, että jos $|z_n|$ kasvaa isommaksi kuin 2 jollain arvolla $n$, rekursio kasvaa rajatta. Arvoa $n$ jolla tämä tapahtuu, kutsutaan tässä tehtävässä pisteen $z_0$ "pakonopeudeksi."

Aloita kirjoittamalla funktio `n = escapeVelocity(z0,c,N)`, jossa $N$ on jokin yläraja pakonopeuksille (erityisesti: jos $|z_n| < 2 \; \forall \; n < N$, funktion tulee palauttaa $N$. Näin vältetään ikuiset silmukat).

Luodaksesi Julia-joukon kirjoita funktio `M=julia(zMax,c,N)`. Argumentti `zMax` määrää kompleksitasosta nelikulmion $|Im(z)| < z_{max}, |Re(z)| < z_{maz}$. $c$ ja $N$ ovat samat argumentit kuin edellä, palautettava matriisi $\mathbf{M}$ koostuu määritetyn hilan pakonopeuksista.

Aloita funktion `julia` kirjoittaminen määrittelemällä $500 \times 500$ hila reaalitasossa, luo sen avulla vastaava hila $\mathbf{Z}$ kompleksitasolle, ja aja funktio `escapeVelocity` jokaiselle matriisin $\mathbf{Z}$ alkiolle.

**Vihje:** Reaaliakselin väli $[a, b]$ määritellään MATLABissa komennolla `I = linspace(a,b,n)`, missä n on haluttujen pisteiden määrä, kuten esim. 500. Hila reaalitasolle määritellään komennolla `[x y] = meshgrid(t1,t2)` , missä t1 ja t2 ovat välejä reaaliakselilta. Tästä luodaan kompleksitasoa peittävä hila komennolla `z = x+i*y`.

Kompleksiluvun modulin saa selville itseisarvofunktiolla `abs`.

**Vaativuus:** 2+
**Tehtävän Latex-koodi:**
../mlteht/mlGraphics/mlGr09.tex

**Ratkaisu:**
../mlteht/mlGraphics/ratkaisut/mlGr09R.m Funktio julia, talleta tiedostoon julia.m, aja (aluksi) 'help julia'-esimerkki

**Avainsanat:**Matlabgrafiikka,mlGraphics, iteraatio, Julia-joukko, Julia-set, dynaaminen systeemi, fraktaali, fractal

**Matlabfunktioita:** meshgrid, imagesc

---

mlGr10.tex

Kirjan Anne Greenbaun - T.P. Chartier: Numerical Methods www-sivulla:
http://academics.davidson.edu/math/chartier/Numerical/matlab.html
on m-tiedosto `plotTaylor.m`

Hae se sieltä tai ota tästä:

```
%  Plot the first four Taylor polynomials for exp(x).

x = [-3:.01:2];                     % Set x values
fx = exp(x);
p0 = ones(size(x));                 % Compute Taylor polynomials
p1 = p0 + x;
p2 = p1 + (x.^2)/2;
p3 = p2 + (x.^3)/6;

subplot(2,2,1)                      % Plot results
plot(x,fx,'--', x,p0,'-');
legend('f(x)','P_0(x)')
title('plot of P_0(x) and f(x)')
subplot(2,2,2)
plot(x,fx,'--',x,p1,'-');
legend('f(x)','P_1(x)')
title('plot of P_1(x) and f(x)')
subplot(2,2,3)
plot(x,fx,'--',x,p2,'-');
legend('f(x)','P_2(x)')
title('plot of P_2(x) and f(x)')
subplot(2,2,4)
plot(x,fx,'--',x,p3,'-');
legend('f(x)','P_3(x)')
title('plot of P_3(x) and f(x)')
```

Leikkaa/liimaa Matlab- tai Octave-komentoikkunaan ja selvitä ja omaksu kunkin komennon merkitys.
Täydennä komennot skriptiksi, joka piirtää lisäksi polynomit $P_4$ ja $P_5$. Anna `subplot`- komennot muodossa `subplot(3,j,k)`.

**Vaativuus:** 1

**Tehtävän Latex-koodi:**
../mlteht/mlGraphics/mlGr10.tex

**Avainsanat, keywords:** mlGraphics, Matlab graphics, grafiikkaa Matlab:lla, plot, subplot, legend, Taylor polynomial

mlGr11
Kirjoita MATLAB-skripti, joka laskee ja piirtää seuraavat funktiot:

a) $y = 5\cos(3\pi x)$. Laske arvo 101:ssä tasavälisessä pisteessä välillä $0 \le x \le 1$.

b) $y = \frac{1}{1+x^2}$ välillä $-5 \le x \le 5$ .

c) $y = \frac{\sin(7x)-\sin(5x)}{\cos(7x)+\cos(5x)}$. Laske arvo 200 tasavälisessä pisteessä välillä $-\frac{\pi}{2} \le x \le \frac{\pi}{2}$. Käytä `axis` komentoa asettaaksesi näytettävät akselit väleille $-2 \le x \le 2$ ja $-10 \le y \le 10$.

**Vihje:**
Jako- ja kertolaskujen tapauksessa ole tarkkana: haluatko matriisioperaation vai alkioittaisen operaation? Alkioittaiset operaatiot erotetaan matriisioperaatioista operaattorin eteen sijoitettavalla pisteellä. Esimerkiksi `.*` on alkioittainen kertolasku, `*` matriisien kertolasku.
Trigonometriset funktiot toimivat MATLABissa alkioittain, ja löytyvät loogisilla nimillä. (`cos, acos, sin` jne.)
Tasavälisiä pistejoukkoja luodaan komennolla `linspace`, tai vaihtoehtoisesti MATLABin kaksoispiste-notaatiolla. Tutustu kummankin dokumentaatioon.
Huomaa (c)-kohdassa, että nimittäjällä on nollakohtia. Piirrä osoittajan ja nimittäjän kuvaajat erikseen samaan koordinaatistoon (eri väreillä). Mieti nyt sitäkin, miksei kuvaaja "räjähdä" kaikissa nimittäjän nollakohdissa.

**Vaativuus:** 2-
**Tehtävän Latex-koodi:**
../mlteht/mlGraphics/mlGr11.tex

**Ratkaisu:**
../mlteht/mlGraphics/ratkaisut/html/mlGr11R.html Publish -> html-muodossa
../mlteht/mlGraphics/ratkaisut/mlGr11R.m (m-tiedosto)

**Avainsanat:** Matlabgrafiikka,mlGraphics

**Matlabfunktioita:** linspace, plot, axis, legend, find

---

mlGr12

Lataa tältä, alla mainitun kirjan (AG-TC) webbisivulta skripti plotfunction1.m

Skriptin ajo piirtää funktion

$$f(x) = 2\cos x - e^x, -6 \le x \le 3$$

kuvaajan sanotulla välillä ja "näkee", että funktiolla on 3 nollakohtaa ko. välillä. Lisää skriptiin `grid on` sopiviin kohtiin.

Zoomaamalla kapeammalle välille saadaan tarkempi arvio. Tämä on tehty alempaan ‘”subplot-ruutuun”.

Huomaa, että alemman ruudun zoomattu funktio näyttää melkein lineaariselta. (Tapahtuuko näin yleensä ?)

a. Muokkaa skriptiä niin, että zoomattu ruutu keskittyy haarukoimaan vasemmanpuolimmaista juurta.

b. Editoi skriptiä tutkiaksesi funktiota

$$f(x) = \frac{4x \sin\, x - 1}{2 + x^2}, -1 \leq x \leq 4.$$

Piirrä zoomattu kuva pienimmän positiivisen juuren ympärille. Määritä kuvasta juurelle likiarvo $x_0$ 3:n numeron tarkkuudella ja laske funktion arvo $f(x_0)$
**Vihje:** Arvon laskemiseksi voit joko toimia tähän tapaan:

```
>> x0= ...
>> % Editoi "copy/pastella" f:n lauseke ja suorita (esim. CTR-ENTER)
>> % Tai kelaa nuolella komentoja, tai hae "command historysta"
```

Toinen, elegantimpi tapa on määritellä f funktioksi tyyliin:
`f=@(x) 4*x./....`

**Vaativuus:** 2-
**Tehtävän Latex-koodi:**
../mlteht/mlGraphics/mlGr12.tex

**Aputiedostoja,viitteitä**
Anne Greenbaum Timothy P. Chartier. *Numerical Methods.* Princeton U.P., 2012.

(AG-TC) Anne Greenbaum - Timothy P. Chartier: Numerical Methods, Princeton U.P. 2012
Kirjan nettisivu: http://academics.davidson.edu/math/chartier/Numerical/matlab.html

**Avainsanat:**Matlabgrafiikka,mlGraphics, nollakohta, zero

**Matlabfunktioita:** plot, subplot, zoom

---

Funktion $g(x)$ kiintopiste on piste $x_0$, jolle pätee $g(x_0) = x_0$. Kiintopisteen sijaintia voi hakea piirtämällä kuvaajat $y = g(x)$ ja $y = x$ samaan kuvaan ja arvioimalla käyrien leikkauspistettä graafisesti.

Esitä arvio funktion $g(x) = \cos(x)$ kiintopisteelle graafisen tarkastelun avulla.

**Vihje:** Kaksi käyrää voidaan piirtää samaan kuvaan joko yhdellä `plot` käskyllä : `plot(x1,y1,x2,y2)`, tai vaihtoehtoisesti voidaan käyttää MATLABin `hold on` optiota:

```
plot(x1,y1);
hold on
plot(x2,y2);
hold off
```

Grafiikkaikkunan zoomaus-valinnalla (tai `zoom`-komennolla) voit tarkentaa arviota. Pitemmälle pääset valitsemalla uuden, ahtaamman välin `linspace`-komennolle ja ajamalla piirtokomenno(n/t) uudestaan (nuolinäppäimellä komentoikkunassa tai editorissa CTR-ENTER).

**Vaativuus:** 1
**Tehtävän Latex-koodi:**
../mlteht/mlGraphics/mlGr13.tex

**Avainsanat:**Matlabgrafiikka,mlGraphics, kiintopiste, fixed point

**Matlabfunktioita:** linspace, plot,zoom

---

mlGr15

Piirrä MATLABilla alla oleva kuva.

./images/sinPlot.png

**Vihje:** Selityslaatikko luodaan komennolla `legend`, akselikuvaukset komennoilla `xlabel` ja `ylabel`. $\pm$ saadaan Latex-tyyliin `\pm`. Tutki help:sta yksityiskohdat.

**Vaativuus:** 1+
**Tehtävän Latex-koodi:**
../mlteht/mlGraphics/mlGr15.tex

**Ratkaisu:**
../mlteht/mlGraphics/ratkaisut/html/mlGr15R.html
../mlteht/mlGraphics/ratkaisut/mlGr15R.m

**Avainsanat, keywords:** mlGraphics, Matlab graphics, grafiikkaa Matlab:lla

---

Tässä tehtävässä tutkitaan kuvien, matriisien ja singulaariarvojen yhteyksiä.

Matriisin $\mathbf{A} \in \mathbb{R}^{m\times n}$ singulaariarvohajotelma on

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T,$$

missä matriisi $\mathbf{S}$ on diagonaalimatriisi, ja matriisit $\mathbf{U}$ ja $\mathbf{V}$ ovat ortogonaalisia neliömatriiseja. Matriisin sisältämää informaatiota voidaan tietyssä mielessä kompressoida tiputtamalla osia singulaariarvohajotelmasta pois; on todistettavissa että (MATLABilla ilmaistuna) `U(:,1:k)*S(1:k,1:k)*V(:,1:k)'` on paras mahdollinen rank($k$)-approksimaatio matriisille $\mathbf{A}$.

Kuva voidaan ajatella $m \times n$ matriisina, missä $i, j$ alkio ilmaisee vastaavassa paikassa olevan pikselin väriarvon. Tutkitaan sitten kuinka singulaariarvoja voidaan käyttää hyväksi kuvien pakkaamisessa ja hahmontunnistuksessa.

Lue haluamasi kuva sisään MATLABin `imread` komennolla. Komento luo (yleensä, mutta hieman kuvasta riippuen), $m \times n \times 3$ matriisin. Tämä vastaa RGB-esitystä: ensimmäisessä kerroksessa on punaisen värin intensiteetit, toisessa vihreän ja kolmannessa sinisen. Muuta tämä matriisi harmaaskaalaan komennolla `rgb2gray`. Tämän jälkeen tee matriisille singulaariarvohajotelma komennolla `[u s v] = svd(P)`, missä `P` on kuvasi matriisiesitys. Tutki sitten millä $k$:n arvolla komentojono

```
>> M = u(:,1:k)*s(1:k,1:k)*v(:,1:k)';
>> image(M)
```

tuottaa havaittavia tuloksia. Pitäisi myös päteä, että kuvan isommat hahmot alkavat erottua ensin, mikä tekee singulaariarvoista huomattavan tehokkaan työkalun hahmontunnistuksessa.

**Vihje**: Kuvan ulottuvuuksien ei kannata olla kovin isoja: singulaariarvohajotelma on raskas laskettava. Jos haluat lisähaastetta, erottele kuvan värikerrokset, tee hajotelma niille erikseen, ja kokoa tulokset. Näin saat aikaan värikuvia.

**Vaativuus:** 2+
**Tehtävän Latex-koodi:**
../mlteht/mlGraphics/mlGr19.tex

**Ratkaisu:**
../mlteht/mlGraphics/ratkaisut/mlGr19R.m Kutsuttava funktio “picturesque”.
*** TEE ajoskripti ***

**Avainsanat:**Matlabgrafiikka,mlGraphics,mlLinalg,kuvankasittely,singulaariarvohajotelma, singularvaluedecomposition, SVD, data compression, tiedon pakkaus

**Matlabfunktioita:** imread, imshow,im2double, svd

---

Luo $n \times n$ matriiseja $\mathbf{A}$ jollakin sopivalla $n$ (s.o. enemmän kuin kymmenen, vähemmän kuin sata), joiden alkiot ovat muotoa

$$\mathbf{A}_{i,j} = \frac{1}{i - j + t}.$$

Piirrä matriisin $\mathbf{A}$ ominaisarvot tasoon, kun $t$ vaihtelee välillä $[-1, 1]$. Mitä havaitset? Voisivatko perättäisten ominaisarvojen *radat* esittää jotain?

`Vihje:` Mieti miten matriisin voisi määritellä ilman silmukkaa. Matriisin ominaisarvot lasketaan komennolla `eig` – huomaa, että jos matriisi on kovin iso, niin laskeminen voi kestää kauan.

**Vaativuus:** 1+
**Tehtävän Latex-koodi:**
../mlteht/mlGraphics/mlGr21.tex

../mlteht/mlGraphics/ratkaisut/mlGr21R.m Funktiotiedosto, kutsu: `»mlGr21R`

**Avainsanat:**Matlabgrafiikka,mlGraphics,mlLinalg, eigenvalues, ominaisarvot

**Matlabfunktioita:** plot, eig

---

a) Piirrä funktiot $\cos t$ ja $\sin t$ samaan kuvaan eri väreillä.

b) Piirrä toiseen kuvaan yksikköympyrä ja säännöllinen $n-$kulmio esim. arvolla $n = 10$. Järjestä sopivilla `axis`-komennoilla skaalat yhtäsuuriksi, jotta ympyrä näkyy ympyränä.

c) Piirrä yksikköympyrän kuva joillain edellä esiintyneillä lineaarikuvauksilla(tai muilla keksimilläsi). (** Vaatii tarkennuksen "edellä esiintyneisiin" **)
**Vihje:** Uusi grafiikkaikkuna: `figure`
Muistathan ympyrän luonnollisen parametriesityksen.
Ympyrän data koostuu oikeasti säännöllisesen n-kulmion nurkkapisteistä, missä esim. $n = 100$ (`linspace`:n oletus). Ympyrän kuvan piirtäminen on siten sama homma kuin edellisissä lineaarikuvaustehtävissä.

**Vaativuus:** 1+
**Tehtävän Latex-koodi:**
../mlteht/mlGraphics/mlGr23.tex

**Avainsanat:**Matlabgrafiikka,mlGraphics

**Matlabfunktioita:** plot, axis equal, axis square

---

mlGr24
Kirjoita skripti, jolla piirrät seuraavat kuvaajat.

a. $f(x) = |x - 1|, -3 \leq x \leq 3$. ( `abs`)

b. $f(x) = \sqrt{|x|}, -4 \leq x \leq 4$. (`sqrt`)

c. $f(x) = e^{-x^2}, -4 \leq x \leq 4$. (`exp`)

d. $f(x) = \frac{1}{10x^2+1}, -2 \leq x \leq 2$. (Muista piste, myös jakolaskussa.)

Kun olet ensin piirtänyt kaikki, editoi jokaiseen kappaleeseen subplot-komento siten, että saat kaikkiaan $2 \times 2$-grafiikkaruudukon, joissa kussakin asustaa yksi noista kuvista. Komenna jokaisessa `grid on` ja `shg` (show graphics).

**Vaativuus:** 1+
**Tehtävän Latex-koodi:**
../mlteht/mlGraphics/mlGr24.tex

**Ratkaisu:**
../mlteht/mlGraphics/ratkaisut/html/mlGr24R.html publish -> html
../mlteht/mlGraphics/ratkaisut/mlGr24R.m (m-tiedosto)

**Aputiedostoja,viitteitä**
AG-TC
Anne Greenbaum - Timothy P. Chartier: Numerical Methods, Princeton U.P. 2024
Kirjan nettisivu: http://academics.davidson.edu/math/chartier/Numerical/matlab.html

**Avainsanat:**Matlabgrafiikka,mlGraphics

**Matlabfunktioita:** abs,sqrt,exp, plot, subplot, grid on, shg, figure

---

## mlIntegraalimuunnos

Luvussa esiintyy tehtäviä aihepiireistä:

- Signaalinkäsittely yleisesti
- Fourier-, Laplace-, Z-muunnokset ym.

---

mlInT001

Yksikköimpulssi, yksikkönäyte (diskreetti $\delta$-funktio) määritellään näin:

$$\delta(n) = \begin{cases} 1, \text{kun} & n = 0, \\ 0, \text{kun} & n \neq 0 \end{cases}$$

Matlab:ssa voidaan muodostaa mm. näin :

```
N=7; % tms.
delta=[zeros(1,N),1,zeros(1,N)];
```

(a) Sijoita yksikkönäyte vektoriin `delta` ja piirrä näytepisteet kokonaislukuvälillä $[-N, N]$, kun $N = 7$. Jotta pisteet erottuisivat paremmin, anna y-skaalausta varten komento `ylim([-0.2,1.2])`. Myös `grid on`-komento selventää (hyvin usein).

(b) Esitä sama data eri ikkunassa (`figure`) käyttäen `stem`-funktiota.

(c) Esitä jokin toinen luonteva tapa muodostaa vektori `delta`, sijoita muuttujaan `delta1`. Tarkista sopivalla Matlab-komennolla vektorien samuus.

**Vaikeus** 1

**Tehtävän Latex-koodi:**
../mlteht/mlIntTrans/mlInT001.tex

**Ratkaisu:**
../mlteht/mlIntTrans/ratkaisut/html/mlInT001R.html
../mlteht/mlIntTrans/ratkaisut/mlInT001R.m

**Avainsanat:** mlBasic, mlIntTrans, Matlabperusteet, signaalinkäsittely, diskreetti delta-funktio
**Matlabfunktioita:** zeros,plot,stem,colon(:)

---

mlIT099
Tutkitaan kohinaisen signaalin suodattamista MATLABissa. Tutkitaan signaalia

$$f(t) = 0.3\sin(3t) + \sin(t),$$

johon lisätään synteettistä virhettä:

```
t = -pi:.05:pi;
f = 0.3*sin(3*t) + sin(t);
fh = f + rand(size(f))-.5;
```

Suodata kohinaisesta signaalista `fh` alkuperäinen esille alkuperäinen signaali mahdollisimman tarkasti käyttämällä diskreettiä Fourier-muunnosta
**Vihje** Signaalin taajuuskomponenti värähtelevät taajuuksilla 3 ja 1 $2\pi$:n aikavälillä, ja taajuudella 1 värähtelevä komponentti dominoi, sillä sen amplitudi on suurempi. Tehtävässä kannattaa siirtyä aika-tasosta taajuustasoon, eli käytännössätehdä signaalille Fourier-muunnos (`fft`), ja piirtää taajuskomponenttien itseisarvot (`abs`) näkyville, ja päätellä, mitkä kuuluvat signaaliin, ja mitkä eivät. Käytä tämän jälkeen loogista indeksointia ja käänteistä Fourier-muunnosta (`ifft`) saadaksesi esille suodatetun signaalin.

---

mlInT100.tex
Ihmispuheen akustisesta energiasta suurin osa keskittyy taajuskaistalle 0-4 kHz. Eräs yksinkertainen idea signaalin salaukseksi (sekoitukseen) on jakaa taajuskaista 0-4 kHz alikaistoihin, ja permutoida näitä kaistoja jonkin ennaltamäärätyn avaimen mukaisesti.

Esimerkiksi äänisignaalin taajuuskaista 0-4 kHz on jaettu neljään alikaistaan, kukin leveydeltään yksi kilohertsi: taajuskaista A koostuu taajuuksista 0-1000Hz, B taajuuksista 1000-2000Hz, C 2-3kHz ja D 3-4kHz. Jotta signaali on ihmiselle ymmärrettävä, kaistojen on oltava järjestettynä järjestykseen ABCD. Kuitenkin, koska signaali on salattu, kaistat ovat väärässä järjestyksessä, esimerkiksi BCDA tai CBAD.

Tässä tehtävässä murretaan näin salattu äänisignaali: lataa salattu äänisignaali `scramble.wav`, ja Fourier-muunnosta ja -käänteismuunnosta sekä loogista indeksointia hyväksikäyttämällä järjestä kilohertsin levyiset kaistat järjestykseen CBDA. Kuuntele signaali varmistuaksesi onnistumisesta.

**Vaativuus:** 3+
**Tehtävän Latex-koodi:**
../mlteht/mlIntTrans/mlInT100.tex

**Ratkaisu:**
../mlteht/mlIntTrans/ratkaisut/mlInT100R.mtxt
(Selaimelle helpompaa näyttää)
../mlteht/mlIntTrans/ratkaisut/mlInT100R.m

**Aputiedostoja,viitteitä**
../mlteht/mlIntTrans/apusrc/scramble.wav

**Avainsanat:** mlInT,Matemaattinen mallinnus Matlab:lla, IntTrans, signaalinkäsittely, Fourier-muunnos, kryptaus

**Matlabfunktioita:**

---

mlInTxxx.tex
Template file
Let

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

What can you say ...

**Vaativuus:** 1 - 3+
**Tehtävän Latex-koodi:**
../mlteht/mlLinalg/mlInTxxx.tex

**Avainsanat, keywords:** Matlab: Integraalimuunnokset, integral transforms, Laplace-muunnos, Fourier-muunnos

---

## mlLinis – Matlab-tehtäviä lineaarialgebrasta

mlLA000.tex
**Ohjetiedosto**, poimi mukaan tehtäväpaperiin tarpeen mukaan.

**Matlab-ohjeita (aluksi samat kuin mlBasic-osassa)**

- Komennon suorittama tulos tulee ruudulle ENTER-painalluksen jälkeen (kuvat erilliseen ikkunaan). Jos haluat estää tulostuksen, päätä komento puolipisteeseen. Jos myöhemmin haluat katsoa muuttujan sisällön, kirjoita sen nimi (ilman puolipistettä). Jos muuttuja on suuri matriisi, kannattaa ensin katsoa sen koko `size(A)` tai sen jotain osaa, esim. `A(1:10,1:10)`

- Edellisen komennon tulos on muuttujassa `ans`. Yleensä on suositeltavaa antaa tulokselle oma nimi tyyliin `nimi= ...`

- `format long` : Tulostetaan enemmän numeroita (n. 16). Laskutarkkuuteen tämä ei vaikuta.
  `format rational` laskee rationaaliluvuilla.
  `format short`: Paluu oletustulostukseen.

- Matriisin `A` transpoosi: `A'`

- Kokonaislukuvektori: Esim `1:10` tai `1:2:20`. Myös `linspace`. Pystyvektoriksi transponoimalla.

- A(i,j) A:n alkio (i,j).
  A(2,:) A:n 2. rivi
  A(:,3) A:n 3. sarake
  A(1:4,1:4) osamatriisi
  Matriisin osaa voi päivittää, vaikkapa:
  `A(1:4,1:4)=ones(4,4)` tai
  `A(2,:)=A(2,:)-2*A(:,1)` (Gaussin rivioperaatio).

- Matriisien liittäminen: Jos $A$:lla ja $B$:llä on yhtä monta riviä, ne voidaan liittää peräkkäin: `[A b]` (tai `[A, b]`)
  Jos yhtä monta saraketta, niin allekkain: `[A;B]`

- Laskutoimitukset tarkoittavat matriisilaskua. Siis esim.
  `A*B, A^p`

- Vektorien ja matriisien (samankokoisten) pisteittäinen eli alkioittainen laskenta tapahtuu lisäämällä eteen piste. Esim:

  `u=[1 2 3], v=[-2 -2 -2], u.*v.`
  Toinen operandi voi olla skalaari.

  Siten esim. vektorin $u$ kaikki komponentit voidaan korottaa toiseen komennolla `u.^2`

### Matlab-ohjeita lineaarialgebraan

- Lineaarisen yhtälöryhmän $Ax = b$ ratkaisu:

```
x=A\b
```

- Matriisifunktioita:

  - Käänteismatriisi *inv*
  - Rangi *rank*
  - Ominaisarvot ja -vektorit:
    *[U,V]=eig(A)*
  - Täydennä
- Täydennä
- 

**Ohjetiedoston Latex-koodi:**
../mlteht/mlLinalg/mlLA000.tex

**Avainsanat:** Ohjetiedosto, Lineaarialgebraa Matlabilla, matriisilaskentaa,mlLinalg,mlLA,Harjoitusohjeita

---

mlLA002.tex
Olkoon

$$\mathbf{A} = \begin{bmatrix} 5 & -13 & -13 & -2 & 7 \\ 18 & -4 & 30 & -1 & -12 \\ -23 & 3 & 7 & 15 & 7 \\ 9 & 36 & -1 & 14 & 16 \\ 3 & 28 & 7 & 14 & 5 \end{bmatrix}$$

ja

$$\mathbf{b} = \begin{bmatrix} -196 \\ 435 \\ 11 \\ 111 \\ 195 \end{bmatrix}.$$

Ratkaise yhtälösysteemi $Ax = b$ ja tarkista tulos matriisikertolaskulla.

**Vaativuus:** 1-
**Tehtävän Latex-koodi:**
../mlteht/mlLinalg/mlLA002.tex

**Ratkaisu:**
../mlteht/mlLinalg/ratkaisut/mlLA002R.m

**Matlabfunktioita:** Takakeno, backslash, `A\b`

**Avainsanat:**
Lineaarialgebraa Matlabilla, matriisilaskentaa,mlLinalg,mlLA Lineaarinen yhtälöryhmä, Matlabperusteet, Matlabalkeet,perusmatriisilaskenta

---

mlLA003.tex [myös Maple, Mathematica]

(a) Ratkaise lineaarinen yhtälöryhmä ja tarkista tulos kertolaskulla.

$$\begin{cases} 4x - 5y = 11 \\ 2x + y = 9 \end{cases}$$

**Vihje:** Matlab: "Matriisijako": $A\backslash b$

(b) Tiedetään, että Celsius-asteiden ja Fahrenheit-asteiden välillä on lineaarinen yhteys:

$$C = aF + b.$$

Lisäksi tiedetään, että vesi jäätyy 32 F:ssa ja -40 on sama kummassakin asteikossa. Johda kaava. Tarkoitus on kirjoittaa kertoimien a ja b määräämiseksi lineaarinen yhtälösysteemi, joka ratkaistaan Matlab:n takakenolla (\).

(c) Muodosta matriisi, jonka 1. sarake on C-asteet $-50$:sta 5:n asteen välein 100 :aan ja toinen sisältää vastaavat F-asteet.

**Vihje:** Tarkan rationaalilukukaavan saat komentamalla `format rat`. Tee m-tiedosto kommentteineen.
Huomaa, että taulukkoa ei ole mukavaa katsoa koknaisuutena, esim. 10 ekaa riviä näet näin: `taulukko(1:10,:)` (eikö vain?).
Hivelevää on myös mennä "Workspace-ikkunaan" ja kaksoisklikata taulukko-ikonia.

Kokeile sen ajamista myös pdf:ksi `publish(Fahrenheit,pdf)`-komennolla (jos skripti on `Fahrenheit.m`), kunhan ensin testailet sen kuntoon.

**Vaativuus:** 2
**Tehtävän Latex-koodi:**
../mlteht/mlLinalg/mlLA003.tex

**Ratkaisu:**
../mlteht/mlLinalg/ratkaisut/html/mlLA003R.html
../mlteht/mlLinalg/ratkaisut/mlLA003R.m

**Avainsanat:** Lineaarialgebraa Matlabilla, matriisilaskentaa,mlLinalg,mlLA

**Matlabfunktioita:**

---

mlLA004.tex Matlab/Maple/Mathematica

Tarkastellaan yhtälösysteemejä:

$$\begin{cases} 10x_1 + 7x_2 + 8x_3 + 7x_4 = 32 \\ 7x_1 + 5x_2 + 6x_3 + 5x_4 = 23 \\ 7x_1 + 6x_2 + 10x_3 + 9x_4 = 33 \\ 7x_1 + 5x_2 + 9x_3 + 10x_4 = 31 \end{cases}$$

ja

$$\begin{cases} 2x_1 + x_2 + 5x_3 + x_4 = 9 \\ x_1 + x_2 - 3x_3 - x_4 = -5 \\ 3x_1 + 6x_2 - 2x_3 + x_4 = 8 \\ 2x_1 + 2x_2 + 2x_3 - 3x_4 = 3 \end{cases}$$

(a) Ratkaise molemmat systeemit.

(b) Muuttamalla vähän yhtälön dataa (oikeaa puolta ja/tai kerroinmatriisia), voidaan tutkia systeemin herkkyyttä pienille virheille (datassa ja pyöristyksessä).

Ratkaise 1. systeemi oikean puolen vektoreilla
`[32.1, 22.9, 32.9,31.1]'` ja `[32.01, 22.99,32.99,31.01]'`
ja 2. systeemi vektoreilla
`[9.1 -5.1, 7.9, 3.1]'` ja `[9.01, -5.01, 7.99, 3.01]'` .

Mitä nämä pienet häiriöt vaikuttavat ratkaisuihin?

(c) Muuta kerroinmatriiseja lisäämällä matriisien kuhunkin alkioon pieni satunnaisluku `0.1*rand` . Ratkaise systeemit alkuperäisillä oikeilla puolilla. Mitä nämä muutokset vaikuttavat ratkaisuihin.

(d) Lineaarisen yhtälösysteemin herkkyyttä pienille virheille sanotaan *häiriöalttiudeksi* (*"ill-conditioned "*). Laske kummankin matriirin häiriöalttius.

Suhteellisen virheen suurtenemisypäyhtälö:

$$\frac{||\Delta x||}{||x||} \leq \kappa \frac{||\Delta b||}{||b||}$$

Pahimmillaan ratkaisun suhteellinen virhe voi olla luokkaa $\kappa\times$ (datan suhteellinen virhe) ($\kappa = cond(A)$)

**Vaativuus:** 2+
**Tehtävän Latex-koodi:**
../mlteht/mlLinalg/mlLA004.tex

**Ratkaisu:**
../mlteht/mlLinalg/ratkaisut/html/mlLA004R.pdf
../mlteht/mlLinalg/ratkaisut/mlLA004R.m

**Avainsanat:** Lineaarialgebraa Matlabilla, matriisilaskentaa,mlLinalg,mlLA,Numeerinen lineaarialgebra, matriisit, lineaariset yhtälöryhmät, häiriöalttius

---

mlLA005.tex
Olkoon

$$\mathbf{A} = \begin{bmatrix} 5 & 1 & 1 \\ 1 & 5 & 1 \\ 1 & 1 & 5 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 10 \\ 14 \\ 18 \end{bmatrix}.$$

Varmista ensin, että matriisi **A** on kääntyvä laskemalla $\det(A)$.
Tutki, mitä muita keinoja on matriisin ei-singulaarisuuden tarkistamiseen. Kokeile vaikka **rank**, **rref**, **lu**, **cond**, **rcond** (katso helpillä).
(Huomaa, että "oikeissa tehtävissä" tärkeämpi käsite on "lähes singulaarisuus", tähän det ei ole yleispätevä työkalu.)

Ratkaise yhtälöryhmä $\mathbf{Ax} = \mathbf{b}$ käyttämällä

a) Käänteismatriisia `inv`

b) MATLABin matriisijakoa `x = A\b`.

**Opetus:** Huomaa, että "matriisijako" on numeerisen tarkkuuden ja laskentatehon kannalta yleensä parempi tapa (mikä ei pienissä, hyvänlaatuisissa tehtävissä tule ilmi).

**Opettajalle:** Tehtävään voidaan lisätä myös A-matriisin muodostaminen diagonaaleittain **diag**-funktiolla (vaikkei mene hyödyn puolelle näin pienessä tehtävässä).

mlLA006.tex = mlPDE001.tex

Tässä tehtävässä tarvitsee vain muodostaa ja ratkaista lineaarinen yhtälöryhmä annettujen ohjeiden mukaan. Toisaalta johdattelee PDE-numeriikkaan.

*Tasapainolämpötilajakauma metallilevyssä.*
Kuva esittää metallilevyä, joka on ylä- ja alapinnoiltaan lämpöeristetty ja jonka reunojen lämpötilat on kiinnitetty. (Lämpöä virtaa vain reunojen kautta.) Tasapainolämpötilajakauma saadaan *Laplacen yhtälön* $\Delta u = 0$ ratkaisuna. Numeerinen approksimaatio voidaan laskea ns. differenssimenetelmällä: Jaetaan levy sopivilla hilaviivoilla osiin ja numeroidaan näin muodostuvat solmupisteet. Menetelmä: Kunkin hilasolmun lämpötila on naapurisolmujen lämpötilojen keskiarvo. (Johdetaan kurssin lopulla.)

Muodosta $4 \times 4-$ yhtälösysteemi solmujen $1, 2, 3, 4$ lämpötilojen likiarvoille $u_1, u_2, u_3, u_4$. Ohje: Aloitetaan solmusta 1: $u_1 = \frac{1}{4}(30 + u_2 + u_3 + 10)$. Vastaavasti muut kolme solmua.

```
          20        20
    |----- o   ----   o ------|
    |                         |
    |                         |
 10 o      o 3      o 4       o  40
    |                         |
    |                         |
 10 o      o 1      o 2       o  40
```

```
|                        |
|                        |
|----- o ------ o ------|
       30       30
```

(a) Ratkaise yhtälösysteemi ja sijoita ratkaisulämpötilat ao. hilapisteisiin.

(b) Muodosta $4 \times 4-$ matriisi, jossa on reunalämpötilat ja ratkaisemasi sisäpistelämpötilat sekä nurkissa lähinnä olevien kahden reunasolmun keskiarvot tähän tapaan: `U=[15 20 20 30;10 u3 u4 40; 10 u1 u2 40; 20 30 30 35];` Piirrä ratkaisupinnan approksimaatio: `mesh(U)` tai `surf(U)`.

(b') Ehka hiukan selkeämpää on rakentaa U-matriisi vaiheittain vaikka tähän tapaan:

```
u=u'  % Vaakavektoriksi
U=zeros(4,4)
U(1,:)=[15 20 20 30]
U(2,:)=[10 u(3:4) 40]
...
...
```

**Vaativuus:** 2
**Tehtävän Latex-koodi:**
../mlteht/mlLinalg/mlLA006.tex

**Ratkaisu:**
../mlteht/mlLinalg/ratkaisut/html/mlLA006R.pdf
../mlteht/mlLinalg/ratkaisut/mlLA006R.m

**Avainsanat:** Lämpötilamatriisi, Laplacen yhtälön diskretointi, differenssimenetelmän perustehtävä, lineaarinen yhtälöryhmä, Discretizaton of Laplace PDE, Linear system of equations

---

mlLA007.tex
Oheinen kuva esittää liikenneverkkoa. Kuhunkin solmuun A,B,C,D tulevien ja siitä lähtevien ajoneuvojen lukumäärien summa pysyy samana (solmuun ei häviä eikä siinä synny ajoneuvoja). Kadut ovat yksisuuntaisia nuolien osoittamalla tavalla.

(a) Muodosta yhtälösysteemi tuntemattomien ajoneuvomäärien $x_1, \ldots x_5$ suhteen.
(b) Määritä systeemin yleinen ratkaisu.
(c) Jos $x_4$:llä merkitty katuosuus suljetaan, niin mikä on yleinen ratkaisu?
(d) Määritä kohdan (c) tilanteessa pienin $x_1$ :n ja suurin $x_3$ :n arvo (jotta yhdensuuntaisuutta osoittavia liikennemerkkejä ei tarvitse kääntää).

Huom! Porrasmuotoon saattamisessa saat halutessasi käyttää Matlab/Octave-funktiota `rref` (kts. `help rref`).

**Avainsanat:** Liikenneverkko, lineaarinen yhtälöryhmä, (redusoitu)porrasmuoto, rref.

mlLA008.tex
*Tehtävä liittyy LU-hajotelma-algoritmin opiskeluun.*

(a) Muodosta $5 \times 5$-yksikkömatriisi I. (`help eye`)

(b) Muodosta matriisi $E_1$, jossa on vaihdettu $I$:n rivit 2 ja 5.

(c) Muodosta matriisi $E_2$, joka saadaan kertomalla $I$:n 4. rivi luvulla 4.

(d) Muodosta matriisi $E_3$, joka saadaan $I$:stä *Gaussin rivioperaatiolla*:

$$r_4 \leftarrow r_4 + 4\, r_1,$$

missä $r_i$ tarkoittaa matriisin riviä numero $i$.

(e) Muodosta matriisit $E_1^{-1},\ \ E_2^{-1},\ \ E_3^{-1}$ käyttäen komentoa `inv` ja selitä, mitä rivioperaatiota ne vastaavat.

**Vaativuus:** 2-
**Tehtävän Latex-koodi:**
../mlteht/mlLinalg/mlLA008.tex

**Avainsanat:** Lineaarialgebraa Matlabilla, matriisilaskentaa,mlLinalg,mlLA, alkeismatriisit,elementarymatrix, LU-hajotelma, Gaussin rivioperaatio, käänteismatriisi, LU-decomposition, matrixinverse

---

mlLA009.tex (KP3-ii, 2008, harj1)
Olkoon

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}$$

ja olkoot

$$E_0 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad E_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 3 \end{bmatrix}, \qquad E_2 = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Muodosta matriisitulot $E_0 A$, $E_1 A$, $AE_1$ ja $E_2 A$ ja selvitä, mitä nämä operaatiot tekevät matriisin $A$ riveille/sarakkeille.

**Vihje:**
Käsinlasku ja ajattelutehtävä, tarkistukseen voit hyödyntää Matlabin **syms**-komentoa tai voit tehdä symboliset matriisioperaatiot Maplella/Mathematicalla.

**Avainsanat:** Alkeismatriisit, LU-hajotelma, Gaussin rivioperaatio

mlLA010.tex
**Harjoituksen (KP3-II/s. 2006) ohjetta:**
*Seuraavissa tehtävissä voitaisiin johonkin johtopäätökseen päästä determinantin avulla. Näissä harjoituksissa ei kelpuuteta tällaisia ratkaisuja, vaan harjoitellaan johtopäätösten tekoa rivio-peraatioiden seurauksena.*

Osa tehtävistä on käsinlaskuun tarkoitettu, mutta niiden yhteydessä voidaan harjoitella samalla Matlab/Octave/Scilab-työskentelyä (Kts. MattieO).

Annettuna on $3 \times 3-$systeemin liitännäismatriisi

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

Muodosta rivioperaatioilla porrasmuoto "ref" — "row echelon form". Merkitse tukisarakkeet ja tukialkioiden paikat. Jatka sitten rivioperaatioita alhaalta ylöspäin päästäksesi redusoituun porrasmuotoon "rref".

**Avainsanat:** Lineaarinen yhtälöryhmä, Gaussin rivioperaatio, ref, rref, (redusoitu) porrasmuoto, row echelon form.

mlLA013.tex (** Siirrä aiheeseen mlDiffintV) **
Piirretään toisen asteen pintoja. Tätä varten tulee pinnat esittää parametrimuodossa

$$x_1 = x_1(u,v), x_2 = x_2(u,v), x_3 = x_3(u,v),$$

missä muuttujat $u$ ja $v$ saava arvoja jostain sopivasta alueesta. Esimerkiksi parametrisointi

$$\begin{cases} z_1 = r_1 \sin u \cos v, \\ z_2 = r_2 \sin u \sin v \\ z_3 = r_3 \cos u, \end{cases} \qquad \begin{cases} 0 \le u \le \pi \\ 0 \le v \le 2\pi \end{cases}$$

MATLAB ilmoittaa $\mathrm{R}^2$:n muuttujat tietyllä tavalla organisoituhin matriiseihin seuraavasti:

```
[U,V] = meshgrid(linspace(0,pi,21),linspace(0,2*pi,21));
```

Nyt ellipsoidin $r_1 = r_2 = r_3$ parametrisointi tehdään seuraavasti

```
Z1 = sin(U).*cos(V);
Z2 = sin(U).*sin(V);
Z3 = cos(U);
```

Kuvan tästä saa piirrettyä komennolla `surf(Z1,Z2,Z3)`. Kokeile miten pinta muuttuu, kun asetat kertoimiksi $r_k$ eri arvoja. Tutustu myös komentoon `axis`.

**9.** mlLi014.tex

Tarkastellaan yhtälöryhmää

$$\begin{cases} 0.0001x_1 + 2x_2 = 4 \\ x_1 + x_2 = 3 \end{cases}$$

Tarkka ratkaisu on $[\frac{2}{1.9999}, \frac{3.9997}{1.9999}]^T$, joka 5:llä numerolla esitettynä on $[1.0001, 1.9999]^T$.

(a) Ratkaise yhtälösysteemi niin, että suoritat laskut (järjestystä vaihtamatta) 3:lla merkitsevällä numerolla. (Laske laskimella, Matlabilla tms. ja pyöristä kunkin operaation jälkeen tulos 3:een numeroon.)

(b) Tee samoin kuin (a)-kohdassa, mutta vaihda yhtälöiden järjestys.

Selitä, miksi (a)-tapauksessa tulee suuri suhteellinen virhe ( 100%:n suhteellinen virhe toisessa komponentissa), kun taas (b)-tapauksessa virhe on olematon.

**Avainsanat:** Lineaarinen yhtälöryhmä, Gaussin rivioperaatio, numeerinen ratkaisu, numeerinen lineaarialgebra, pyöristysvirhe.

**10.** mlLi015.tex

Olkoon $A = \begin{bmatrix} 1 & 8 & 6 \\ -1 & -4 & 5 \\ 2 & 4 & -6 \end{bmatrix}$ ja $\mathbf{b} = \begin{bmatrix} 8 & 1 & 4 \end{bmatrix}^T$. Ratkaise yhtälö $Ax = b$ *osittaistuentaa* ("partial pivoting") käyttäen.

Olkoon $P$ permutaatiomatriisi (rivinvaihtomatriisi), joka määräytyy rivien vaihdoista. Muodosta hajotelma $PA = LU$.

Matlab:lla: `help lu`, `[L,U,P]=lu(A)` (Tämä siis vertailun vuoksi, tarkoitus on laskea käsin.)

**Avainsanat:** Lineaarinen yhtälöryhmä, Gaussin rivioperaatio, LU-hajotelma, numeerinen lineaarialgebra, (osittais)tuenta, "(partial) pivoting".

**Vihje:** Osittaistuenta tarkoittaa itseisarvoltaan suurimman tukialkion valitsemista pienen tukialkion aiheuttamien numeeristen ongelmien välttämiseksi. Matlab saattaa käyttää esim. ns. skaalattua osittaistuentaa, jolloin rivinvaihtostrategia voi olla erilainen.

**11.** mlLi017.tex

Muodosta “ylimääräytyvälle” yhtälöryhmälle

$$\begin{cases} x + 3y = 5 \\ x - y = 1 \\ x + y = 0 \end{cases}$$

normaaliyhtälöt ja ratkaise pienimmän neliösumman (PNS,LSQ) mielessä. Piirrä suorat ja ratkaisupiste tasoon.

Vastaustarkistuskeino: Huomaa, että Matlab:n yhtälösysteemin ratkaisija: $x = A\backslash b$ on niin älykäs, että se ymmärtää ylimääräytyvässä tapauksessa suorittaa PNS-ratkaisun.

**Avainsanat:** Lineaarinen yhtälöryhmä, PNS,LSQ, pienimmän neliösumman menetelmä, LU-hajotelma, numeerinen lineaarialgebra, (osittais)tuenta, “(partial) pivoting”.

**Vihje:**

**12.** mlLi018.tex

Huom: Käsinlasku täydennettynä pikku Matlab-osuudella.

Eräässä mittauksessa saatiin seuraava data:

```
xdata    1    2   3   4   5
ydata    1.8 2.7 3.4 3.8 3.9
```

Dataa mallinnetaan polynomilla $p(x) = c_1\, x + c_2\, x^2$.

(a) Muodosta PNS-tehtävän matriisi $X$ ja vektori $y$ siten, että tehtävä saadaan ylimääräytyväksi yhtälöryhmäksi $X\, c = y$.

(b) Ratkaise kerroinvektori $c$. Piirrä data ja PNS-polynomi samaan kuvaan.

**Vihje:** (b)-kohdassa saat mieluusti käyttää Matlab:ia. Tee kuitenkin vaiheittain matriikertolaskut, transpoosit ym., lopuksi toki voit tarkistaa “takakenolla”. Piirrä samaan kuvaan datapisteet ja polynomi.

Piirtäminen käy näin:

```
xd=1:5; yd=[1.8 ...]; plot(xd,yd,'x'); hold on; kertoimet=[c2 c1 0]; x=linspace(1,5); y=polyval(
plot(x,y,'r'); xlim([0 6]); grid on
```

Huomaa, että *polyval* haluaa kertoimet korkeimmasta potenssista alkaen.

Vast: $p(x) = 1.76x - 0.2x^2$

**Avainsanat:** PNS,LSQ, pienimmän neliösumman menetelmä, käyrän sovitus, curve fitting, data fitting.

mlLA019.tex

Määritä PNS-ratkaisu tehtävälle $A\,x = b$, kun $A = \begin{bmatrix} 1 & -1 \\ 1 & 4 \\ 1 & -1 \\ 1 & 4 \end{bmatrix}$ ja $b = \begin{bmatrix} -1 & 6 & 5 & 7 \end{bmatrix}^T$. Hyödynnä QR-hajotelmaa, jonka voit muodostaa seuraavilla komennoilla: (Huom: Yleensä ei lasketa rationaaliaritmetiikalla, mutta opettelussa voi olla hyödyksi.)

```
>> format rational
>> A=[...]      % Jos kirjoitat [...], olet TONTTU!
>> [Q,R]=qr(A)
```

**Vihje** Matlab muodostaa ns. täyden QR-hajotelman. Kuten huomaat, riittää ottaa Q:n kaksi ensimmäistä saraketta ja R:n 2 ensimmäistä riviä, miten nyt vain haluat. Huomaa siis, että Q on ortogonaalinen ja R on yläkolmiomatriisi.

**Avainsanat:** PNS,LSQ, pienimmän neliösumman menetelmä,QR-hajotelma.

mlLA0191.tex

**Ohjeita, ominaisarvo-oppia** (Liitettäväksi aiheen tehtäväpaperiin)

- **Ominaisarvo** on luku, se voi olla kompleksiluku, vaikka matriisi olisi reaalinen.
- **Ominaisvektori** on (reaalisen matriisin tapauksessa) $\mathbb{R}^n$:n tai $\mathbb{C}^n$:n vektori sen mukaan, onko vastaava ominaisarvo reaalinen vai kompleksinen.
- Ominais**arvo saa** aivan mainiosti **olla** 0, ominais**vektoriksi emme hyväksy nollavektoria**.
- Ominaisarvoon $\lambda$ liittyvä **ominaisavaruus** $E_\lambda$ koostuu kaikista $\lambda$:aan liittyvistä ominaisvektoreista ja lisäksi nollavektorista. Tällöin kyseessä on vektori(ali)avaruus, nimittäin matriisin $A - \lambda I$ nolla-avaruus, $N(A - \lambda I)$.
- Ominaisarvon $\lambda_j$ **algebrallinen kertaluku** $M_{\lambda_j}$ on karakteristisen polynomin $\det(A-\lambda I)$ juuren kertaluku. **Geometrinen kertaluku** $m_{\lambda_j}$ on $\dim(E_{\lambda_j})$.
  Pätee: $m_{\lambda_j} \le M_{\lambda_j}$
- Jos reaalisella matriisilla $A$ on **kompleksinen** ominaisarvo $\lambda = \alpha + i\,\beta$, niin myös $\overline{\lambda} = \alpha - i\,\beta$ on $A$ :n ominaisarvo. Jos $\mathbf{v}$ on $\lambda$ :aa vastaava ominaisvektori, niin liittolukua $\overline{\lambda}$ vastaava ominaisvektori on $\overline{\mathbf{v}}$. (Tarkoittaa vektoria, jonka koordinaatit ovat $\mathbf{v}$ :n koordinaattien liittolukuja.)
- Jos on määrättävä diagonaalimatriisin ominaisarvot ja -vektorit, niin laskentatyötä ei jää lainkaan. Älä siis suotta ryhdy veivaamaan $\det(A - \lambda I)$:n kautta. (Koko ominaisarvohomman perustavoite on saattaa lineaarikuvauksen matriisi diagonaalimuotoon. Jos se jo on, niin mitään ei tarvitse enää tehdä, kunhan osaat siitä lukea.)
- Kolmiomatriisin (ylä- tai ala-) ominaisarvot ovat diagonaalialkiot. (Siis yleistys edelliselle, tässä tapauksessa ominaisvektoreista ei voida sanoa mitään yleistä.)

- Kun pyydetään laskemaan johonkin ominaisarvoon liittyvät ominaisvektorit, on sopivaa antaa vastaukseksi ominaisavaruuden kanta. Helpoimmin se saadaan antamalla ratkaisun vapaille muuttujille vuorollaan arvot ( $1,0,0$) , $(0,1,0)$ , $(0,0,1)$ (jos kyseessä on 3-ulotteinen ominaisavaruus). Tässähän on kyse nolla-avaruuden kannan määräämistehtävästä.

- Eri ominaisarvoihin liittyvät ominaisvektorit ovat `LRT`.

- Diagonalisointi: Annettu $A$. Etsittävä, jos mahdollista, matriisit $V$ ja $D$, $V$ kääntyvä ja $D$ diagonaalimatriisi siten, että $A = VDV^{-1}$.
  Jos tehtävänä on diagonalisoida $A$, etsitään matriisit $V$ ja $D$ ja perustellaan $V$:n kääntyvyys. (Yleensä ei vaadita $V^{-1}$:n laskemista ilman eri kehoitusta, tai jatkotehtävän asettamaa tarvetta.)

- Octave/Matlab-komentoa `eig` kannattaa käyttää ainakin tarkistukseen. Muoto `[V,D]=eig(A)` antaa suoraan diagonalisointimatriisit: $V$ :n sarakkeina ominaisvektorit ja $D$ :n diagonaalilla (samassa järjestyksessä) ominaisarvot. Jos $A$ on diagonalisoituva, niin $V$ :n sarakkeet ovat LRT, jolloin voidaan muodostaa $V^{-1}$; Matlab/Octavella: `inv(V)`.

**Ohjetiedoston Latex-koodi:**
../mlteht/mlLinalg/mlLA0191.tex

**Avainsanat:** Ominaisarvo-opin perus(lasku)ohjeita

mlLA0192.tex
Määritä matriisin $A$ ominaisarvot ja muodosta kutakin ominaisarvoa vastaavalle ominaisavaruudelle jokin kanta, kun

$$A = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

Mitk"øvat algebralliset ja geometriset kertaluvut?

**Vaativuus:** 1+
**Tehtävän Latex-koodi:**
../mlteht/mlLinalg/mlLA0192.tex

**Avainsanat:** Lineaarialgebraa Matlabilla, matriisilaskentaa,mlLinalg,mlLA, Ominaisarvot, ominaisvektorit, Matlab: eigshow, eig, ominaisarvon kertaluku (alg/geom)

---

mlLA020.tex
Suorita Matlab-komento `eigshow`. Opiskele helppiteksti ja suorita joitakin kokeiluja kuljettamalla $x$-vektoria läpi koko yksikköympyrän. (Tämä vain lämmittelyksi.)

Valitse erityisesti matriisit `A=[1 3;4 2]/4`, `B=[3 1;-2 4]/4` ja
`C=[2 4;2 4]/4`. Määritä kuvan perusteella kunkin ominaisarvot ja -vektorit. Saat vektorit tarkemmin komentamalla `grid on`.

Mitä, jos kuvan perusteella ominaisarvoja/vektoreita ei näyttäisi olevan?
Määritä kuvan perusteella myös ominaisavaruuden dimensio matriisin $C$ tapauksessa.

Laske kunkin matriisin ominaisarvot ja -vektorit `eig`-komennolla. (`help eig`)

**Huom:** Jos näitä matriiseja ei sattuisi olemaan valmiina valikossa, voit ne sinne lisätä help-piruudun "View code for eigshow"-linkistä. Hae koodista kohta `mats = ...`. Siitä näet, miten matriiseja voi lisätä. Jos editoit koodia, tallenna se omaan hakemistoosi vaikkapa nimelle `ominashow`, ja sitten vaan `ominashow`.

Tee joitakin omia kokeiluja erilaisilla matriiseilla oman "ominashow":n avulla.

**Vaativuus:** 1+
**Tehtävän Latex-koodi:**
../mlteht/mlLinalg/mlLA020.tex

**Avainsanat:** Lineaarialgebraa Matlabilla, matriisilaskentaa,mlLinalg,mlLA, Ominaisarvot, ominaisvektorit, ominaisarvojen graafinen havainnollistaminen, ominaisvektorien graafinen havainnollistaminen, Matlab: eigshow, eig

---

mlLA020a.tex
**Ohjeita, ominaisarvo-oppia** (Liitettäväksi aiheen tehtäväpaperiin)

- **Ominaisarvo** on luku, se voi olla kompleksiluku, vaikka matriisi olisi reaalinen.
- **Ominaisvektori** on (reaalisen matriisin tapauksessa) $\mathbb{R}^n$:n tai $\mathbb{C}^n$:n vektori sen mukaan, onko vastaava ominaisarvo reaalinen vai kompleksinen.
- Ominais**arvo saa** aivan mainiosti **olla** 0, ominais**vektoriksi emme hyväksy nollavektoria**.
- Ominaisarvoon $\lambda$ liittyvä **ominaisavaruus** $E_\lambda$ koostuu kaikista $\lambda$:aan liittyvistä ominaisvektoreista ja lisäksi nollavektorista. Tällöin kyseessä on vektori(ali)avaruus, nimittäin matriisin $A - \lambda I$ nolla-avaruus, $N(A - \lambda I)$.
- Ominaisarvon $\lambda_j$ **algebrallinen kertaluku** $M_{\lambda_j}$ on karakteristisen polynomin $\det(A-\lambda I)$ juuren kertaluku. **Geometrinen kertaluku** $m_{\lambda_j}$ on $\dim(E_{\lambda_j})$.
  Pätee: $m_{\lambda_j} \leq M_{\lambda_j}$
- Jos reaalisella matriisilla $A$ on **kompleksinen** ominaisarvo $\lambda = \alpha + i\,\beta$, niin myös $\overline{\lambda} = \alpha - i\,\beta$ on $A$ :n ominaisarvo. Jos $\mathbf{v}$ on $\lambda$ :aa vastaava ominaisvektori, niin liittolukua $\overline{\lambda}$ vastaava ominaisvektori on $\overline{\mathbf{v}}$. (Tarkoittaa vektoria, jonka koordinaatit ovat $\mathbf{v}$ :n koordinaattien liittolukuja.)
- Jos on määrättävä diagonaalimatriisin ominaisarvot ja -vektorit, niin laskentatyötä ei jää lainkaan. Älä siis suotta ryhdy veivaamaan $\det(A - \lambda I)$:n kautta. (Koko ominaisarvohomman perustavoite on saattaa lineaarikuvauksen matriisi diagonaalimuotoon. Jos se jo on, niin mitään ei tarvitse enää tehdä, kunhan osaat siitä lukea.)

- Kolmiomatriisin (ylä- tai ala-) ominaisarvot ovat diagonaalialkiot. (Siis yleistys edelliselle, tässä tapauksessa ominaisvektoreista ei voida sanoa mitään yleistä.)

- Kun pyydetään laskemaan johonkin ominaisarvoon liittyvät ominaisvektorit, on sopivaa antaa vastaukseksi ominaisavaruuden kanta. Helpoimmin se saadaan antamalla ratkaisun vapaille muuttujille vuorollaan arvot $(1,0,0)$, $(0,1,0)$, $(0,0,1)$ (jos kyseessä on 3-ulotteinen ominaisavaruus). Tässähän on kyse nolla-avaruuden kannan määräämistehtävästä.

- Eri ominaisarvoihin liittyvät ominaisvektorit ovat `LRT`.

- Diagonalisointi: Annettu $A$. Etsittävä, jos mahdollista, matriisit $V$ ja $D$, $V$ kääntyvä ja $D$ diagonaalimatriisi siten, että $A = VDV^{-1}$.
  Jos tehtävänä on diagonalisoida $A$, etsitään matriisit $V$ ja $D$ ja perustellaan $V$:n kääntyvyys. (Yleensä ei vaadita $V^{-1}$:n laskemista ilman eri kehoitusta, tai jatkotehtävän asettamaa tarvetta.)

- Octave/Matlab-komentoa `eig` kannattaa käyttää ainakin tarkistukseen. Muoto `[V,D]=eig(A)` antaa suoraan diagonalisointimatriisit: $V$ :n sarakkeina ominaisvektorit ja $D$ :n diagonaalilla (samassa järjestyksessä) ominaisarvot. Jos $A$ on diagonalisoituva, niin $V$ :n sarakkeet ovat LRT, jolloin voidaan muodostaa $V^{-1}$; Matlab/Octavella: `inv(V)`.

**Ohjetiedoston Latex-koodi:**
../mlteht/mlLinalg/mlLA020a.tex

**Avainsanat:** Ominaisarvo-opin perus(lasku)ohjeita

mlLA021.tex (käsinlasku, Matlab sopii avuksi/opiksi)

Muodosta matriisin $A = \begin{bmatrix} 3 & -2 & 4 \\ -2 & 6 & 2 \\ 4 & 2 & 3 \end{bmatrix}$ ortogonaalinen diagonalisointi (tarkoittaa ortonormaalia).

Laskutyön vähentämiseksi annetaan (tai pyydetään oppilasta komentamaan):

```
>> eig(A)
ans =
         -2.00
          7.00
          7.00
```

**Avainsanat:** Ominaisarvot, ominaisvektorit, ortogonaalinen diagonalisointi.

**Vihje:**
Muista, että ominaisvektorit eivät automaattisesti ole yksikkövektoreita, ja useampikertaista ominaisarvoa vastaavat ominaisvektorit eivät automaattisesti ole ortogonaaliset.

Jos olet saanut samaan ominaisavaruuteen kuuluvat `LRT` ominaisvektorit $v_1$ ja $v_2$, niin ortonormaalin kannan saat
1) geometrisen ajattelun avulla: Muodosta $v_2$:n kohtisuora projektio $v_1$:llä ja vähennä se $v_2$:sta.
Tai
2) algebrallisesti: Määritä kerroin kerroin $c$ siten, että $(v_1 + c\,v_2) \perp v_1$.

---

**13.** mlLi028.tex
Olkoon

$$\mathbf{A} = \begin{bmatrix} 5 & 1 & 1 \\ 1 & 5 & 1 \\ 1 & 1 & 5 \end{bmatrix}$$

a) Laske matriisin $\mathbf{A}$ diagonalisointiin tarvittavat matriisit $P$ ja $D$.

b) Varmista, että $P$ on ortogonaalinen, ja $D$ on diagonaalinen ja diagonaalialkiot suuruusjärjestyksessä.

c) Osoita, että spektraalikaava

$$\lambda_1 \mathbf{u}_1 \mathbf{u}_1^T + \lambda_2 \mathbf{u}_2 \mathbf{u}_2^T + \lambda_3 \mathbf{u}_3 \mathbf{u}_3^T = A$$

pätee.

**Vihje:**

**14.** mlLi029.tex
Potenssimenetelmä on eräs keino löytää itseisarvoltaan suurin ominaisarvo ja vastaava ominaisvektori. Menetelmä toimii seuraavasti:

- Valitse alkuarvaus $\mathbf{b}_0$. Ainoa vaatimus on, että tällä vektorilla on nollastapoikkeava komponentti ominaisarvon suuntaan – käytännössä kannattaa valita vektori, jonka jokainen alkio on nollasta poikkeava.
- Aseta
$$\mathbf{b}_{k+1} = \frac{\mathbf{A}\mathbf{b}_k}{||\mathbf{A}\mathbf{b}_k||}$$
- Jatka kunnes jono $(\mathbf{b}_k)$ suppenee. Ominaisarvo $\lambda = ||\mathbf{A}\mathbf{b}_k||$ ja vektori $\mathbf{x} = \mathbf{b}_k$.

Toteuta menetelmä MATLABissa, ja laske matriisin `gallery(5)` suurin ominaisarvo ja vastaava ominaisvektori. Testaa tuloksen oikeellisuus.

mlLA030.tex
Ominaisarvojen laskentamenetelmiä, Power method [KRE[9]] Sec. 20.8
Sovella potenssimenetelmää (3 kierrosta) matriisiin

$$A = \begin{bmatrix} 3.5 & 2.0 \\ 2.0 & 0.5 \end{bmatrix}$$

alkuarvolla $x_0 = [1, 1]^T$. Laske *Rayleigh-osamäärät* $q$ ja virherajat.

**Ratk:** Vastaus: $q = 4, 4.493, 4.4999; |\epsilon| \leq 1.5, 0.1849, 0.0206$

**Avainsanat:** Potenssimenetelmä, ominaisarvojen laskentamenetelmät, Power method.

---

**15.** mlLi031.tex

Osoita, että jos $x$ on ominaisvektori, niin $\delta = 0$ virhekaavassa (2) Theorem 1 s. 872 (KRE$^9$, luvun 20.8, Power Method for Eigenvalues alkusivulla) .

**Vihje:** Päättelytehtävä, ohjelmistoista ei hyötyä.

**Avainsanat:** Potenssimenetelmä, ominaisarvojen laskentamenetelmät, Power method.

mlLA032.tex

(a) Päättele Gershgorinin lauseen avulla matriisin

$$A = \begin{bmatrix} 11 & 0.4 & -0.5 \\ 0.4 & 7 & a \\ -0.5 & a & 4 \end{bmatrix}$$

ominaisarvojen likiarvot ja missä rajoissa ne ovat.

(b) Millä $a$:n reaaliarvoilla nähdään suoraan, että matriisi on kääntyvä. (Tarkoitus ei ole laskea determinanttia tai ominaisarvoja, korkeintaan halutessasi tarkistukseksi ja varmistuksesi Gershgorinin pätevyydelle.)

**Vihje** Käsinlasku, jossa voit harjoitella Matlabin laskinkäyttöä.

**Avainsanat:** Gershgorinin lause , ominaisarvojen laskentamenetelmät, ominaisarvoarvio.

---

mlLA033

Gershgorinin lauseen graafinen havainnollistus.

Tee funktio G:n ympyröille.

Testaa erilaisilla matriiseilla.

**16.** mlLi040.tex

Olkoot $\mathbf{A}$ kompleksinen $n \times n$ matriisi, ja olkoot $R_i = \sum_{j \neq i} |a_{ij}|$, eli rivin alkioiden itseisarvojen summa diagonaalia lukuunottamatta. Gershgorinin kiekkolauseen väite on, että jokainen matriisin $\mathbf{A}$ ominaisarvo $\lambda_i$ sijaitsee jossakin kiekossa $D(a_{ii}, R_i)$, (kompleksitasoon piirretty kiekko, jonka keskipiste on pisteessä $a_{ii}$, ja jonka säde on $R_i$). Totea lauseen väite kokeellisesti, kun `A=10*randn(12)+ 5*randn(12)*i;`

**Vihje:** Ympyrän, jonka keskipiste on $(x, y)$ ja säde $r$ saa MATLABissa piirrettyä helposti seuraavasti:

```
x = 0.4; y= -0.34
t = 0:0.02:2*pi;
plot(x+cos(t),y+sin(t));
hold on
%Yksittäinen piste piirretään seuraavasti
plot(x,y,'r.');
```

**17.** mlLi050.tex

Gram-Schmidtin menetelmä vektorijoukon $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ ortonormalisoimiseksi toimii seuraavasti:

- Ortogonalisoidaan:
  $\mathbf{u}_1 = \mathbf{v}_1$
  $\mathbf{u}_2 = \mathbf{v}_2 - \frac{\mathbf{u_1 \cdot v_2}}{\mathbf{u_1 \cdot u_1}}\mathbf{u}_1$
  $\vdots$
  $\mathbf{u}_n = \mathbf{v}_n - \sum_{k=1}^{n-1} \frac{\mathbf{u}_k \cdot \mathbf{v_n}}{\mathbf{u}_k \cdot \mathbf{u}_k}\mathbf{u}_k$
- Normitetaan: $\mathbf{e}_i = \frac{\mathbf{u}_i}{||\mathbf{u}_i||}, i = 1 \ldots n$

Kirjoita MATLAB-funktio `B = grmsch(A)` joka hakee Gram-Schmidtin menetelmällä ortonormaalin kannan matriisin $\mathbf{A}$ sarakeavaruudelle. Testaa ortonormaalius laskemalla `B'*B`.

**Vinkki:** Laskutoimitus $\frac{\mathbf{u}_k \cdot \mathbf{v_n}}{\mathbf{u}_k \cdot \mathbf{u}_k}\mathbf{u}_k$ vastaa toimitusta $(\mathbf{u}_k^T \mathbf{v}_n)\mathbf{u}_k$. **Lisätehtävä nopeille:** Matriisin sarakeavaruuden normalisointi ei poikkea kovin paljon QR-hajotelman tekemisestä. Jos ehdit, toteuta oma algoritmisi QR-hajotelmalle.

mlLA090

Seuraava kuva esittää kymmenen sivun "internettiä".

Laske tämän verkon tärkein sivu käyttämällä PageRank-algoritmia:

- Luo verkon vierusmatriisi $A = [a_{ij}]$, missä

$$a_{ij} = \begin{cases} 1 & \text{jos sivulta j on linkki sivulle i} \\ 0 & \text{muuten} \end{cases}$$

- Laske vierusmatriisin suurin ominaisarvo, ja vastaava ominaisvektori.
- Normalisoi laskemasi suurinta ominaisarvoa vastaava ominaisvektori (jaa kaikki vektori alkiot vektorin summalla). Mikä on tämän verkon tärkein sivu.
- Piirrä verkon kuva käyttäen laatimaasi vierusmatriisia ja `gplot`-komentoa. Tutustu `gplot`in help-sivuun.

---

mlLA500.tex

Luovuttaja: Riikka Kangaslampi (Perustuu Juha-Matti Perkkiön laatimaan hahmontunnistusharjoitukseen.)

Käynnistä Matlab, ja käy lataamassa kurssin Noppa-sivulta tiedosto nimeltä `kuvatiedosto.mat`. Vaihda Matlabin työhakemistoksi se hakemisto, mistä tämä tiedosto löytyy, ja kirjoita

```
load␣kuvatiedosto
who
```

Matlabin muistiin ilmestyi $10304 \times 400$ -kokoinen matriisi KUVA, jonka kussakin sarakkeessa on pystyvektoriksi venytetty $112 \times 92$ -kokoinen harmaasävykuva. Kuvissa esiintyy 40 eri henkilöä, joista kustakin on 10 eri valokuvaa. Kuvat on järjestetty niin, että sarakkeissa

$$10(n-1)+1, ..., 10n$$

on kuvat samasta henkilöstä $n = 1, \ldots, 40$. Kuvat on saatavilla ilmaiseksi PNG-muodossa AT&T Laboratories Cambridgen webbisivulta
http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.
Sieltä löytyy myös kätevä esikatseluversio, jossa näkyy yhtä aikaa kaikki 400 kuvaa.

Luodaan funktio, jonka avulla voidaan katsella sarakkeista löytyviä kuvia. Muodosta tiedosto nimeltä `nayta.m`, ja kirjoita sinne seuraava koodinpätkä:

```
function nayta(kuva)
imagesc(reshape(kuva,112,92))
colormap gray
colorbar
axis equal tight
```

Nyt sarakkeesta $n$ löytyvä kuva saadaan näkyviin kirjoittamalla

```
figure(1)
nayta(KUVA(:,n))
```

Selaa läpi matriisista KUVA löytyviä kuvia.

Yritetään selvittää, miten hyvin tietokoneen saa tunnistamaan ihmiskasvoja. Muodostetaan ensin keskimääräinen kasvokuva $KA$, ja sen jälkeen matriisi $X$, joka sisältää kuvat, joista on poistettu tämä keskiarvokuva:

```
KA = sum(KUVA')'/400; % Laskee sarakkeiden keskiarvon
X = zeros(size(KUVA));
for n=1:400; X(:,n) = KUVA(:,n)-KA; end
```

Katso, miltä keskiarvokuva ja matriisista $X$ löytyvät kuvat näyttävät.

Seuraavaksi muodostetaan matriisin $X$ singulaariarvohajotelma komennolla

```
[U,S,V] = svd(X,0);
```

Tässä saattaa mennä hetki. Matriisin $A \in C^{m \times n}$ singulaariarvohajotelma on $A = USV^H$, missä matriisien $U \in C^{m \times n}$ ja $V \in C^{n \times n}$ sarakevektorit ovat ortonormeerattuja, ja $S \in C^{n \times n}$ on diagonaalimatriisi, jonka kaikki alkiot ovat ei-negatiivisia reaalilukuja, laskevassa järjestyksessä. Erityisesti $V$ on unitaarinen, eli $V^{-1} = V^H$. Alkuperäisen matriisin $X$ sarakkeet saadaan esiin seuraavalla käskyllä:

```
X(:,n) = U*S*V(n,:)';
```

Matriisin $X$ sarakevektorit tulee siis esitettyä lineaarikombinaatioina ortonormaaleista sarakevektoreista $U(:,n)$, joista ensimmäisiä vastaavat kertoimet ovat tyypillisesti häntäpään kertoimia suurempia. Matriisin U sarakkeet ovat tietyssä mielessä kuvakirjaston "merkittäviä piirteitä". Tarkastele näitä kirjoittamalla `nayta(U(:,n))`.

Matriisia $X$ voidaan approksimoida singulaariarvohajotelman avulla. Singulaariarvojen suhteista voidaan päätellä, kuinka monta ensimmäistä matriisin $U$ pystyriviä kannattaa ottaa mukaan. Kirjoita

```
var = (diag(S).^2);
varpart = cumsum(var)/sum(var);
plot(varpart)
```

Nyt graafista näkyy, kuinka suuri osa singulaariarvojen neliöistä on mukana, kun matriisista U otetaan mukaan annettu määrä vektoreita.

Katsotaan, miltä eripituiset approksimaatiot näyttävät. Pystyvektoria $X(:,n)$ vastaava approximaatio $Y$, jossa on mukana $k$ ensimmäistä komponenttia, saadaan näkyviin käskyillä

```
Y = U(:,1:k)*S(1:k,1:k)*V(n,1:k)';
nayta(KA + Y)
```

Kokeile, miltä kuvat näyttävät eripituisilla approksimaatioilla.

Pudotetaan seuraavaksi singulaariarvohajotelmasta 300 viimeistä komponenttia:

```
U = U(:,1:100);
C = S(1:100,1:100)*V(:,1:100)';
```

Nyt siis approksimaatio kuvalle KUVA(:,n) saadaan seuraavasti:

```
nayta(KA + U*C(:,n))
```

Huomaa, kuinka paljon datan määrä tippui, ja kuinka kunkin yksittäisen kasvon koodaamiseen riittää alkuperäisen 10304 luvun sijasta vain 100 lukua. Lisäksi olisi hyvin todennäköistä, että jonkin aivan uuden kasvokuvan koodaamiseen kelpaisi sama $10304 \times 100$ -matriisi $U$, eli jopa kirjastosta puuttuvien kasvojen koodaus onnistuisi todennäköisesti ainoastaan sadalla luvulla.

Kokeillaan seuraavaksi, miten hahmontunnistus onnistuu. Ajatellaan, että tietokone on nähnyt kustakin neljästäkymmenestä henkilöstä yhdeksän eri kuvaa, jotka virittävät aliavaruudet $W^n \in \mathbb{R}^{10304}$ kullekin henkilölle $n = 1, \ldots, 40$, ja annetaan tietokoneen laskea lopuista 40 kuvasta etäisyydet kuhunkin näistä aliavaruuksista. Pienin etäisyys vastaa siis suurella todennäköisyydellä sitä, että kyseessä on sama henkilö. Koska matriisin $U$ pystyvektorit ovat ortonormeerattuja, niin etäisyyksien laskemisen voi aivan yhtä hyvin suorittaa sataulotteisessa komponenttiavaruudessa, jossa kasvokuvia vastaavat matriisin $C$ pystyvektorit. Nyt siis approksimaatiota $U * C(:, n)$ vastaa sataulotteinen komponenttivektori $C(:, n)$, ja sisätuloille (ja sitä kautta etäisyyksille) pätee

```
(U*C(:,n))'*(U*C(:,k)) = C(:,n)'*C(:,k).
```

Muodosta tiedosto nimeltä `tunnistuskoe.m`, ja kirjoita sinne seuraava koodinpätkä:

```
P = cell(1,40);

for n=1:40
    [Q,R]=qr(C(:,(n-1)*10+1:n*10-1),0);
    P{n} = Q*Q';

    for m=1:40
W(m,n)=norm(C(:,10*m)-P{n}*C(:,10*m));
    end
end
```

Nyt siis $P\{n\}$ on projektiomatriisi sarakevektoreiden

```
C(:,(n-1)*10+1),...,C(:,n*10-1)
```

virittämään aliavaruuteen, ja $W(m, n)$ on vektorin $U * C(:, 10 * n)$ etäisyys aliavaruudesta $W_m \subset \mathbb{R}^{10304}$. Aja tämä pieni skripti kirjoittamalla tunnistuskoe, ja tutki matriisia $W$. Kuinka monen henkilön kohdalla tunnistus meni oikein?

mlLAxxx.tex
Template file
Let

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

What can you say ...

**Vaativuus:** 1 - 3+
**Tehtävän Latex-koodi:**
../mlteht/mlLinalg/mlLAxxx.tex

**Avainsanat, keywords:** Lineaarialgebraa Matlabilla, matriisilaskentaa,mlLinalg,mlLA
-e

## mlMatriisit

**18.** Potenssimenetelmä on eräs keino löytää magnitudiltaan isoin ominaisarvo ja -vektori. Menetelmä toimii seuraavasti:

- Valitse alkuarvaus $\mathbf{b}_0$. Ainoa vaatimus on, että tällä vektorilla on nollastapoikkeava komponentti ominaisarvon suuntaan – käytännössä kannattaa valita vektori, jonka jokainen alkio on nollasta poikkeava.
- Aseta
$$\mathbf{b}_{k+1} = \frac{\mathbf{A}\mathbf{b}_k}{||\mathbf{A}\mathbf{b}_k||}$$
- Jatka kunnes jono $(\mathbf{b}_k)$ suppenee. Ominaisarvo $\lambda = ||\mathbf{A}\mathbf{b}_k||$ ja vektori $\mathbf{x} = \mathbf{b}_k$.

Toteuta menetelmä MATLABissä, ja laske matriisin `gallery(5)` isoin ominaisarvo ja -vektori. Testaa tuloksen oikeellisuus.

**Vihje:**

**19.** Tehdään LU-hajotelma tuentaa hyväksikäyttäen. Ensimmäiseksi, yritetään ymmärtää, kuinka tuenta toimii seuraavan pseudokoodin avulla.

[Kommentoitu pois codebox-osuus]

Kirjoita vastaava MATLAB-funktio, ja ratkaise sen avulla ongelma $\mathbf{A}\mathbf{x} = \mathbf{b}$, kun

$$\mathbf{A} = \begin{bmatrix} 0 & 3 & 12 \\ 3 & 12 & 1 \\ -1 & -3 & 4 \end{bmatrix}, b = \begin{bmatrix} 29 \\ 21 \\ 45 \end{bmatrix}.$$

Kokeile sitten ratkaista ongelma $\mathbf{A}\mathbf{x} = \mathbf{b}$, kun $\mathbf{A}$ on $18 \times 18$ Hilbertin matriisi, ja $\mathbf{b} = \mathbf{A}[1]_{18}$,

**Vihje:**

Hilbertin matriisi MATLABina:

```
A = hilb(18);
b = A*ones(18,1);
```

**20.** Gram-Schmidtin menetelmä vektorijoukon $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ ortonormalisoimiseksi toimii seuraavasti:

- Ortogonalisoidaan:
  $\mathbf{u}_1 = \mathbf{v}_1$

  $\mathbf{u}_2 = \mathbf{v}_2 - \frac{\mathbf{u_1 \cdot v_2}}{\mathbf{u_1 \cdot u_1}}\mathbf{u}_1$

  $\vdots$

  $\mathbf{u}_n = \mathbf{v}_n - \sum_{k=1}^{n-1} \frac{\mathbf{u}_k \cdot \mathbf{v_n}}{\mathbf{u}_k \cdot \mathbf{u}_k}\mathbf{u}_k$
- Normitetaan: $\mathbf{e}_i = \frac{\mathbf{u}_i}{||\mathbf{u}_i||}, i = 1 \ldots n$

Kirjoita MATLAB-funktio `B = grmsch(A)` joka hakee Gram-Schmidtin menetelmällä ortonormaalin kannan matriisin $\mathbf{A}$ sarakeavaruudelle. Testaa ortonormaalius laskemalla `B'*B`.

**Vinkki:** Laskutoimitus $\frac{\mathbf{u}_k \cdot \mathbf{v_n}}{\mathbf{u}_k \cdot \mathbf{u}_k}\mathbf{u}_k$ vastaa toimitusta $(\mathbf{u}_k^T \mathbf{v}_n)\mathbf{u}_k$. **Lisätehtävä nopeille:** Matriisin sarakeavaruuden normalisointi ei poikkea kovin paljon QR-hajotelman tekemisestä. Jos ehdit, toteuta oma algoritmisi QR-hajotelmalle.

**Vihje:**

**21.** Matriisi $(a_{ij}) = \mathbf{A} \in \mathrm{R}^{n\times m}$ on *yläkolmiomatriisi*, jos $a_{ij} = 0$ kun $i > j$.
(a)Kirjoita MATLAB funktio, joka ratkaiseen yhtälöryhmän $\mathbf{Ax} = \mathbf{b}$, kun $\mathbf{A}$ on yläkolmiomatriisi.
(b) Generoi satunnaisia yläkolmiomatriiseja, ja tutki josko

(1) kahden yläkolmiomatriisin tulo on aina yläkolmiomatriisi,

(2) yläkolmiomatriisin käänteismatriisi on aina yläkolmio.

(3) determinantti on aina nollastapoikkeava.

**Vihje:** Satunnaisia matriiseje voi luoda kommennoilla `rand` ja `randn`. Näitä kertomalla saa matriiseja joiden arvot ovat millä vain halutulla välillä. Yläkolmion saa matriisista $\mathbf{A}$ komennolla `triu(A)`.

**22.** Olkoon

$$\mathbf{A} = \begin{bmatrix} 5 & 1 & 1 \\ 1 & 5 & 1 \\ 1 & 1 & 5 \end{bmatrix}$$

a) Laske matriisin $\mathbf{A}$ diagonalisointiin tarvittavat matriisit $P$ ja $D$.

b) Varmista, että $P$ on ortogonaalinen, ja $D$ on diagonaalinen ja diagonaalialkiot suuruusjärjestyksessä.

c) Osoita, että spektraalikaava

$$\lambda_1 \mathbf{u}_1 \mathbf{u}_1^T + \lambda_2 \mathbf{u}_2 \mathbf{u}_2^T + \lambda_3 \mathbf{u}_3 \mathbf{u}_3^T = A$$

**Vihje:**

## mlteht/mplModelling, Mallinnus

Matemaattinen mallinnus Matlab:ia käyttäen. Tässä osiossa on tehtäviä, joissa saatetaan soveltaan useamman osa-alueen työkaluja. Jotkin tehtävät saattaisivat sopia hyvin myös jonkun muun aihealueen alle.

---

mlMod001

[Hunt]s. 132 Illuminating a Room

Huoneen mitat olkoot: $10 \times 4 \times 3$ metriä, korkeus =3m.

(a) aaaaaaaaaaaaaaaa

**Vihje**:

**Vaativuus:** 1-3+
**Tehtävän Latex-koodi:**
../mlteht/mlModelling/mlModxxx.tex
**Ratkaisu:**

**Avainsanat:** mlMod,Matemaattinen mallinnus Matlab:lla, Modelling
**Matlabfunktioita:**

---

mlModxxx.tex
Template file
Let

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

What can you say ...

**Vaativuus:** 1 - 3+
**Tehtävän Latex-koodi:**
../mlteht/mlLinalg/mlModxxx.tex

**Avainsanat, keywords:** Matlab:modelling, mallinnus

---

## mlNonlinequ, Epälineaariset yhtälöt

mlNl005.tex
Tarkastellaan yhtälöä

$$\tan(0.1x) = 9.2e^{-x}.$$

(a) Osoita, että yhtälöllä on yksikäsitteinen ratkaisu avoimella välillä $(0, 2\pi)$.

(b) Piirrä kummankin puolen kuvaaja samaan kuvaan etsiäksesi sopivan alkuarvon $x_0$.

(c) Ratkaise yhtälö numeerisesti Matlab-funktiolla `fzero`

(d) Ratkaise yhtälö numeerisesti Newtonin menetelmällä.
*Opettajalle:* Voit antaa `ratkaisut/myNewton.m` - funktion valmiina käyttöön tai sisällyttää sen ohjelmoimisen tähän tai aiempaan tehtävään.

**Vihje 1**: Newtonia varten voit kokeilla derivointia symbolisesti Matlabilla tyyliin

```
>> syms x
>> diff(tan(0.1*x),x)
```

**Vihje 2**: Jos ratkaisu on muuttujassa x, saat kätevästi arvon piirroksen otsikkoon komentamalla piirtokomennon jälkeen:

```
title(['Ratkaisu = ',num2str(x)])
```

**Vaativuus** 2-

**Tehtävän Latex-koodi:**
../mlteht/mlNonlinEqu/mlNl005.tex

**Ratkaisu:**
../mlteht/mlNonlinEqu/ratkaisut/html/mlNl005R.html (Publish: m-tied.-> html)
../mlteht/mlNonlinEqu/ratkaisut/mlNl005R.m (Matlab:n m-tiedosto)

**Avainsanat:** mlNonlinequ, mlNl, Epälineaariset yhtälöt, Matlab, nonlinear equations, numerical solutions of equations, Newton's method for nonlinear equation
**Matlabfunctions:** fzero, num2str, syms, diff

---

**23.** Historiallisesti mielenkiintoinen yhtälö on

$$x^3 - 2x - 5 = 0,$$

jota Wallis-niminen matemaatikko käsitteli, kun hän ensi kertaa esitteli Newtonin menetelmää Ranskan akatemialle. [Lähde: Moler NCM]

1. Piirrä kuvaaja saadaksesi alkuarvon Newtonin menetelmälle reaalijuurta varten.
2. Määritä reaalijuuri omanewton:lla Neuvo: Polynomifunktion voit määritellä tähän tapaan, kun p on kerroinvektori:

   ```
   pf=@(x) polyval(p,x)
   ```

   Derivaatan saat polyder- (tai omapolyder)-funktiolla.
3. Yritä löytää kompleksijuuri antamalla kompleksisia alkuarvoja. (Jos löydät yhden, niin toinen on sen liittoluku.)
4. Määritä juuret roots-funktion avulla.

**Vihje:**

**Ratkaisu:**`MlNl01ratk.m`[Tulee]

Määritä funktion

$$f(x) = (x-1)\sin\left(\frac{x}{x^2+0.4x+0.1}\right)$$

kaikki nollakohdat reaaliakselilla. Perustele, että "siinä" on kaikki.
Piirrä kuva sopivassa skaalassa ja merkitse nollakohdat punaisilla rinkuloilla.

**Vaativuus** 1+

**Tehtävän Latex-koodi:**
../mlteht/mlNonlinEqu/mlNl010.tex

**Ratkaisu:**
../mlteht/mlNonlinEqu/ratkaisut/html/mlNl010R.html (Publish: m-tied.-> html)
../mlteht/mlNonlinEqu/ratkaisut/mlNl010R.m (Matlab:n m-tiedosto)

**Avainsanat:** mlNonlinequ, mlNl, Epälineaariset yhtälöt, nonlinear equations, numerical solutions of equations
**Matlabfunctions:** plot, linspace, fzero,

---

Selvitä piirtämällä sopivissa skaaloissa, kuinka monta nollakohtaa on f- funktiolla annetulla välillä.

$$f(x) = (x-1)\sin\left(\frac{12x}{x^2+0.4x+0.1}\right) \quad x \in [-4, 4]$$

Määritä pienin nollakohta välillä $[-0.2, -0.1]$ funktion `fzero` avulla.

**Vaativuus** 2-

**Vihje:** Tee Matlab-editorissa kappalejaottelu tyyliin:

```
close all
clear
%%
x=linspace(-4,-2,1000);
plot(x,f(x)); grid on;shg
N=1   % Laskuri nollakohtien lukumäärälle
%%
figure
x=linspace(-2,-1,1000);
plot(x,f(x)); grid on;shg
N=N+3  % Lasketaan kuvasta lisäys
%%
...
```

**Tehtävän Latex-koodi:**
../mlteht/mlNonlinEqu/mlNl011.tex

**Ratkaisu:**
../mlteht/mlNonlinEqu/ratkaisut/html/mlNl011R.html (Publish: m-tied.-> html)
../mlteht/mlNonlinEqu/ratkaisut/mlNl011R.m (Matlab:n m-tiedosto)

**Avainsanat:** mlNonlinequ, mlNl, Epälineaariset yhtälöt, nonlinear equations, numerical solutions of equations
**Matlabfunctions:** plot, linspace, fzero,

---

**24.** [NCM 4.15, p. 138]

Keplerin malli:

$$M = E - e\sin,\ E$$

1. Ratkaise fzero:lla
2. Sarjakehitelmä:

$$E = 2\sum_{m=1}^{\infty} \frac{1}{m} J_m(me)\sin(mM)$$

**Vihje:**

**Ratkaisu:**`MlNl02ratk.m`[Tulee]

**25.** [NCM 4.15, p. 138]

Vesiputken syvyys, jotta ei jäädy.

**Vihje:**

**Ratkaisu:**`MlNlxxratk.m`[Tulee]

mlNl030
Määritä pisteet $X_1 = (x_1, x_2)$ ja $X_2 = (x_3, x_4)$ siten, että $p_1, \ldots, p_4$ näihin $x$-pisteisiin laskettujen etäisyyksien neliöiden summa on minimi, kun

$$p_1 = (0,0), p_2 = (1.8,0), p_3 = (1.5,1), p_4 = (0.3,1.6).$$

(Tarkennus: Pisteitä $p-$ ja $X-$pisteiden yhdistysjanojen muodostama murtoviivasto on mahdollsimman lyhyt. Tarvittaneen kuva, piirrä Matlabilla)

**26.** Välinpuolitusmenetelmä on eräs tapa löytää funktion nollakohta. Bolzanon lauseen nojalla, jos jatkuvalla funktiolla on jonkin suljetun välin $[a, b]$ päätepisteissä erimerkkiset arvot, sillä on vähintään yksi nollakohta tällä välillä. Välinpuolitusmenetelmä toimii seuraavasti:

- Laske välin $[a, b]$ keskikohta $m = \frac{b-a}{2}$.
- Laske $f(m)$. Jos $f(m) = 0$, ollaan löydetty nollakohta ja lopetetaan algoritmi.
- Jos $f(m)$:n merkki on sama kuin $f(a)$:n, voidaan tutkittavan välin vasenta päätepistetta siirtää kohtaan $m$, eli $a \leftarrow m$, ja palataan algoritmin kohtaan 1.
- Jos $f(m)$:n merkki on sama kuin $f(b)$:n, voidaan tutkittavan välin oikeaa päätepistetta siirtää kohtaan $m$, eli $b \leftarrow m$, ja siirrytään algoritmin kohtaan 1.

Tätä ideaa noudattaen, laske funktion $f(x) = e^{\sin(x^2)}e^{-x^2}\sin(x^2) - \frac{1}{2}$ välillä $[0, 1]$ sijaitseva nollakohta.

**Vihje:** Numeerisessa tapauksessa absoluuttisen nollan löytäminen on lähes mahdotonta – sinun tulee määrittää jokin hyväksyttävä toleranssi. Funktion arvojen samanmerkkisyyttä voidaan tutkia laskemalla näiden tulo: jos kahden luvun tulo on positiivinen, ovat ne samanmerkkisiä.

**27.** Sekanttimenetelmä on toinen funktion nollakohtien löytämiseen käytettävä menetelmä. Sekantti on suora joka leikkaa annettua käyrää kahdessa pisteessä. Sekanttimenetelmän idea on approksimoida annettua funktiota $f$ välillä $[a, b]$ pisteiden $f(a)$ ja $f(b)$ välille piirretyllä suoralla. Tämän suoran, ja x-akselin välinen leikkauspiste otetaan välin uudeksi päätepisteeksi. Iteraatiokaavaksi saadaan näin

$$x_{n+1} = x_n - f(x_n)\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}$$

Sekanttimenetelmä suppenee yleensä, mutta ei aina, nopeammin kuin välinpuolitusmenetelmä.

Sekanttimenetelmää käyttäen laske funktion

$$g(x) = \cos(\frac{1}{2}\sqrt{x})^2\sqrt{x^3\pi}$$

välillä $[0, 5]$ sijaitseva nollakohta.

**Vihje:** Tälle funktiolle ja tälle menetelmälle on mahdollista löytää alkuarvo, jolla menetelmä ei toimi: kokeile siis useaa arvausta.

**28.** Matlab/Maple/Mathematica
H2T17/mlNl100/mplY100/mmaY100

Etsi yhtälön $x^8 - 36x^7 + 546x^6 - 4536x^5 + 22449x^4 - 67284x^3 + 118124x^2 - 109584x + 40320 = 0$ välillä $[5.5, 6.5]$ oleva juuri. Muuta $x^7$:n kerroin luvuksi $-36.001$ ja katso, mikä vaikutus sillä on juureen.

**Vihje:** Maple: `fsolve`
Matlab: `roots` Mathematica: ...

**Ratkaisu:** Ratkaisutiedostossa lisää variaatioita ja analyysiä tehtävään.
**Avainsanat:** Polynomin juuret, numeriikka, häiriöalttius, ill-conditioned

**29.** Ratkaise yhtälöryhmä

$$\begin{cases} x_1^3 x_2 - 2 &= 0 \\ \sin(x_1) - 1 &= 0 \\ x_3^2 - 3 &= 0 \end{cases}$$

käyttäen Newtonin menetelmää. Tutki suppenemista eri alkuarvoilla.

**Vihje:** Tehtävässä tarvittava Jacobin matriisi kannattaa (ehkä) tehdä spesifinä funktiona.

**30.** Maple,Matlab,[Mathematica] (H2T8)

Newtonin menetelmän askel voidaan määritellä vähäeleisesti Maplelle. Määritellään iterointifunktio:

```
> N := x -> evalf(x - f(x)/D(f)(x));
```

Iterointi tapahtuu joko for-silmukalla tai iterointioperaattorilla `N@@k`. (For silmukka lienee tehokkaampi, kun halutaan muodostaa koko iterointijono.) Ratkaise seuraavat yhtälöt Newtonin menetelmällä. Sopivat alkuarvot vaikkapa kuvan avulla.

a) $x \cos x = \sin x + 1, \quad 0 < x < 2\pi$

b) $x^2 + \sin x = 8$

**Vihje:** Matlab-tehtävässä on antoisinta tehdä Maple-Matlab-työnjako: Muodostetaan ensin iteraatiokaava Maplella symbolisessa muodossa (jätetään yllä N-kaavasta *evalf* pois) ja siirretään kaava Matlabiin (`lprint`, Matlabissa `vectorize` lisää pisteet.

Kaikkein kätevintä lienee käyttää Symbolic Toolboxia symboliseen derivointiin, jos se on käytettävissä.

**Avainsanat:** Epälineaarinen yhtälö, Newtonin menetelmä, iteraatio

mlNl09
Maple , Matlab (H2T7/2014)

Lähde: Burden-Faires

Tarkastellaan väestönkasvumallia

$$N(t) = N_0 e^{\lambda t} + \frac{v}{\lambda}(e^{\lambda t} - 1),$$

jossa otetaan huomioon biologisen lisääntymisen ohella myös maahanmuutto, jonka oletetaan tapahtuvan vakionopeudella $v$ yksilöä vuodessa (netto). Oletetaan, että tietty populaatio on alunperin $10^6$ yksilöä, 435000 yksilöä muuttaa "maahan"1. vuoden aikana ja populaatiossa on 1564000 yksilöä vuoden lopulla. Määritä luku $\lambda$ Käytä tätä $\lambda$:n arvoa ennustamaan populaation koko toisen vuoden lopussa, kun oletetaan maahanmuuttovauhdin säilyvän vakiona.

**Vihje:** Maple: `fsolve`, Matlab: `fzero`

**Vaativuus:** 2
**Tehtävän Latex-koodi:**
../mlteht/mlNonlinEqu/mlNl09.tex

**Ratkaisu:**
../mlteht/mlNonlinEqu/ratkaisut/html/mlNl09R.html (Publish: m-tied.-> html)
../mlteht/mlNonlinEqu/ratkaisut/mlNl09R.m (Matlab:n m-tiedosto)

**Avainsanat:** Epälineeriset yhtälöt, Nonlinear equations, Matlab ,mlNonlinEqu,mlNl, vaestonkasvumalli, väestonkasvumalli, population growth

**Matlabfunktioita:** fzero

---

mlNl11.tex

Newtonin menetelmä lienee kaikista funktion nollakohdan etsimiseen käytetyistä menetelmistä kuuluisin. Silloin, kun menetelmä toimii, se on todella tehokas, suppenee kvadrattisesti, ts. oikeiden numeroiden lukumäärä likipitäen kaksinkertaistuu joka iteraatiokierroksella. Toisaalta menetelmä vaatii usein varsin hyvän alkuarvauksen, ja joissakin tilanteissa ei suppene hyvälläkään alkuarvolla.

Newtonin iteraatioilla tarkasteltavilta funktioilta odotetaan jatkuvuutta ja derivoituvuutta. Newtonin iteraatiokaava on

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Kirjoita approksimatiivinen Newtonin meentelmä korvaamalla derivaatta differenssiapproksimaatiolla. Etsi funktion $f(x) = e^{\sin(x^2)} e^{-x^2} \sin(x^2) - \frac{1}{2}$ nollakohtia käyttäen Newtonin menetelmää. Käytä alkuarvoina ainakin arvoja $0.5, 12.2$ ja $2.2$. Kuten huomataan, alkuarvoilla on todella dramaattinen vaikutus siihen, kuinka ja minne menetelmä suppenee.

HUOM! Otetaan tehtävään myös symbolinen derivaatta. Hyvä NUMSYM-tehtävä.

Kokeile sitten ratkaista funktion $g(x) = x^3 - 2x + 2$ nollakohta Newtonin menetelmällä käyttäen alkuarvauksena $x_0 = 1$. Mitä tapahtuu? (vinkki: `Ctrl + C` lopettaa ikuisen luupin.)

Viimeisenä kokeile ratkaista funktion $h(x) = 1-x^2$ nollakohta Newtonin menetelmällä käyttäen alkuarvauksena $x_0 = 0$. Mitä tapahtuu?

**Vihje:**
Numeerisissa tapauksissa etsitään nollakohtaa jollain sopivalla toleranssilla. Derivaattana kannattaa käyttää määritelmän sijasta 3-pisteen sääntöä:

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h},$$

kun $h$ on pieni.

**Vaativuus:** 1+
**Tehtävän Latex-koodi:**
../mlteht/mlNonlinEqu/mlNl11.tex

**Avainsanat:** Epälineeriset yhtälöt, Nonlinear equations, Matlab ,mlNonlinEqu,mlNl, Newtonin menetelmä nollakohdalle

**Matlabfunktioita:** fzero

---

**31.** Kun $z = x + iy$ ja $-2 \leq x, y \leq 2$, eksponenttifunktion $z \mapsto \exp(z)$ kuvaajan voi piirtää seuraavasti:

```
t = -2:0.2:2;
[x y] = meshgrid(t,t);
z = x+i*y;
r = exp(z)
mesh(real(r));
```

Imaginääriosan saa piirrettyä komennolla `mesh(imag(r))`. Tee vastaavat graafit seuraavista kuvauksista edellämainitulla välillä.

1. $z \mapsto \log(z)$
2. $z \mapsto z^2$
3. $z \mapsto z + 1/z$

**Vihje:**

**32.** Määritellään

$$S_n(x) = \frac{nx}{1 + n^2x^2}$$

Näytä että jokaiselle kiinnitettylle arvolle $x$ luku $S_n(x)$ lähestyy nollaa, kun $n \to \infty$, ja etsi $S_n(x)$:n ääriarvot $x$:n suhteen. Piirrä funktio $S_n(x)$ välillä $[-2, 2]$ kun $n = 2, 4, 6, 8, 10$.

**Vihje:** Ohjelman suorituksen voi keskeyttää kesken skriptin komennolla `pause`. Välin voi määritellä joko vektorinotaatiolla `-2:0.02:2` tai komennolla `linspace(-2,2,100)`.

mlNl20.tex [Moler: NCM probl. 4.22 p. 139]

Olkoon $f(x) = \sin(\tan x) - \tan(\sin x)$.

(a) Piirrä $f$ välillä $[0, \pi]$

(b) Etsi minimiä tällä välillä, piirrä erikseen kumpikin termi ja kavenna väliä, muista `grid on` .

(c) Mitä saat käyttämällä funktiota *fminbnd*

(d) Etsi suurinta alarajaa (*infimum*) "find-tekniikalla"

(e) Määritä $\lim_{x \to \pi/2} f(x)$. Mikä on *infimum*?

**Vihje:**

**Vaativuus:** 2+
**Tehtävän Latex-koodi:**
../mlteht/mlNonlinEqu/mlNl20.tex

**Avainsanat:** Epälineeriset yhtälöt, Nonlinear equations, Matlab ,mlNonlinEqu,mlNl, Newtonin menetelmä nollakohdalle

**Matlabfunktioita:** fzero

---

mlNlxxx.tex
Template file
Let

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

What can you say ...

**Vaativuus:** 1 - 3+
**Tehtävän Latex-koodi:**
../mlteht/mlLinalg/mlNlxxx.tex

**Avainsanat, keywords:** mlNl, Matlab: non-linear equations, epälineaariset yhtälöt

---

## mlNumerics, Perusnumeriikkaa

Numeriikkaa esiintyy erikseen kunkin aihepiirin yhteydessä.

---

mlNum001

Kone-epsilon

(a) aaaaaaaaaaaaaaaa

**Vihje**:

**Vaikeus**

**Avainsanat:** mlNumerics, Basic numerical methods, perusnumeriikkaa Matlab:lla
**Matlabfunctions:** xxx

---

mlNum002

Toisen asteen yhtälön $a\,x^2 + b\,x + c = 0$ ratkaisukaava on kaikille tuttu.

(a) Käytä ratkaisukaavaa yhtälöön molempien juurien laskemiseen,
kun $a = 1, b = -10^{-8}, c = 1$.

(b) Vertaa tulosta Matlabin numeeriseen ratkaisijan `roots` antamaan.

(c) Selvitä, kumman ratkaisun tarkkuus on hyvä, olkoon se $x_1$. Laske toinen ratkaisu juurien tulon kavaa käyttämällä.

**Vihje**: Jos et muista tai halua johtaa, niin voit käyttää Matlabin symbolilaskentaominaisuuksia tai Maplea/Mathematicaa yms.
Matlab:

```
>> syms a b c x
>> yht=a*x^2+b*x+c==0
>> solve(yht,x)
>> pretty(ans)   % Ei kovin "pretty".
```

Tai

```
 >> mupad
 >> yht:=a*x^2+b*x+c=0
 >> solve(yht,x)
```

Maple:

```
 > yht := a*x^2+b*x+c = 0
 > solve(yht,x)
```

**Vaikeus** 1

**Avainsanat:** mlNumerics, Basic numerical methods, perusnumeriikkaa Matlab:lla basicMatlabbasic
**Matlabfunctions:** roots

mlNumxxx.tex
Template file
Let

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

What can you say ...

**Vaativuus:** 1 - 3+
**Tehtävän Latex-koodi:**
../mlteht/mlLinalg/mlNumxxx.tex

**Avainsanat, keywords:** Matlab:Numerics, numeriikka, numeric analysis, NA, numerical mehods, numeeriset menetelmät

## mlDifferentiaali(yhtälöt)

mlD001.tex (iv3 harj.1 av 2001)
Tätä tehtävää harjoitellaan Matlab-teknisesti loppuviikon harjoituksissa. Osattava esittää (ke 19.9.) liitutaululla käsin piirtäen periaatteessa.

Tarkastellaan diffyhtälöä $y' = y - x$ alueessa $-1 \leq x, y \leq 1$,

Piirrä $xy$-koordinaatistoon suuntakenttä ja isokliinejä käsin ja kokeile myös Matlabia laskimen roolissa. Ota hilaväliksi aluksi vaikka $h = 0.5$.

Matlab-laskussa kannattaa muodostaa matriisi, sanokaamme K, jonka alkioina ovat arvot $y_i - x_j$, $i = 1 \ldots 5, j = 1 \ldots 5$ tähän tapaan:

```
h=0.5; t=-1:h:1;x=0:h:2
for i=1:5
  for j=1:5
    K(i,j)=y(i)-x(j)
  end
end
```

Koska matriisissa rivi-indeksi i juoksee alaspäin, on helpompaa sijoittaa arvot koordinaatistoon kääntämällä matriisin sarakkeet ylösalaisin; tämä tapahtuu komennolla `flipud`. Katso siis matriisista `flipud(K)` arvot ja merkitse ne kynällä piirrokseen. Tarkista käsin (tai "skalaarilaskimella" (Matlabkin käy)) muutama alkio ainakin. (Hilan tihentäminen käy nyt helposti muuttamalla vain yllä $h$:ta.)

Myöhemmin opimme, että K-matriisin muodostaminen käy kätevämmin, tehokkaammin ja rutiininomaisemmin näin:

```
x=...;y=...  % Kuten edellä
[X,Y]=meshgrid(x,y);
K=Y-X;    % Jos esiintyy kertolaskua, potenssia ym.,
          % on varustettava pisteellä, siis
          %     .*, .^  ./  (taulukko-operaatiot)
```

Nyt meillä on kaikki data kerättynä suuntakentän piirtämistä varten. Voit katsoa skriptiä suuntak1.m alla aputiedostossa ja miettiä, mitä siinä tapahtuu. (myös: http://www.math.hut.fi/teaching/v/mat

Lopuksi voit kokeilla LAODE-funktiota `dfield8` (versio v. 2012) , jonka käyttö ei vaadi Matlabin tuntemista lainkaan. http://math.rice.edu/~dfield/

**Vaativuus** 2.

**Tehtävän Latex-koodi:**
../matlabteht/mlODE/mlODE001.tex

**Avainsanat:**MatlabODE, diffyhtalot, mlDifferentiaaliyhtälöt,suuntakenttä

**Viitteet:** `[LAODE]` Golubitzky-Dellnitz: Linear Algebra and Differential Equations using Matlab, Brooks/Cole 1999.

---

mlODE002.tex [vastaava Maple: mplteht/mplODE003.tex] (iv3/2001, harj. 1)

Millä xy-tason käyrillä on ominaisuus: Käyrän tangentin kulmakerroin jokaisessa pisteessä $(x, y)$ on $-\frac{4x}{y}$ ?

Ratkaise yhtälö muuttujien erottelulla ("separation of variables"). Piirrä suuntakenttä isokliineja apuna käyttäen käsin vaikkapa alueessa $[-2, 2] \times [-2, 2]$.

Kokeile myös LAODE-funktiota dfield8. Tässä on käytettävä ahkerasti stop-näppäintä, ratkaisu ajautuu aina ongelma-alueelle, mikäli x-akseli on mukana.

Voit myös täydentää kuvaa alussa laskemillasi ratkaisukäyrillä tyyliin:

```
x=linspace(-2,2,30);y=x; [X,Y]=meshgrid(x,y);
Z=... % muista pisteitt\"aiset laskutoimitukset.
contour(x,y,Z,1:10); shg
```

Kokeile ja selitä!

**Vihje:** Hae m-tiedosto *dfield8* sivulta `http://math.rice.edu/~dfield/` ja sijoita se Matlab-polkusi varrelle. Kirjoita Matlab-istuntoon : `dfield8`

**Vaativuus:** 2
**Tehtävän Latex-koodi:**
../mlteht/mlODE/mlODE002.tex

**Viitteet:**

`[LAODE]` Golubitzky-Dellnitz: Linear Algebra and Differential Equations using Matlab, Brooks/Cole 1999.

**Avainsanat:** mlODE,Matlabdiffyht,differentiaaliyhtaloita Matlab:lla,suuntakenttä, isokliinit

---

**33.** mlD003.tex (iv3/2001, harj. 1, teht. 3) (Jatkoa: mlD007.tex)
Lisääntymiskykyinen populaatio, jonka lisääntymistä rajoittavia tekijöitä ei ole, noudattaa yleensä likimain *Malthus'n* lakia, ts. nykyhetkellä kasvunopeus on verrannollinen populaation nykykokoon.

Muodosta ilmiölle differentiaaliyhtälömalli ja ratkaise.

Ratkaisussa esiintyy vakiot $y_0$ = populaation koko alkuhetkellä ja $k$ = "lisääntymiskykyvakio".

Täydennämme vanhaa tuttua USA:n väkilukutaulukkoa arvoilla, jossa toinen rivi ilmaisee väkiluvun miljoonissa ja ensimmäinen vuosiluvun.

```
1800    1830      1860      1890     1920     1950     1980
5.3      13        31        63       106      150      230
```

a) Määritä $k$ kahden ensimmäisen datasarakkeen avulla. Tutki, mihin vuosilukuun saakka malli antaa siedettävän tuloksen ja mistä alkaen sietämättömän.

b) Määritä $k$ ensimmäisen ja viimeisen datasarakkkeen perusteella (jolloin kyseessä ei enää ole tulevaisuuden ennustaminen), ja tutki tämän mallin käyttäytymistä muissa datapisteissä.

c) Havainnollista kuvilla, käytä tarpeen mukaan logaritmista skaalaa.

**Vihje:** Voit valita ajan 0-hetken vuosiluvuksi 1800 (miksi?). Toki ei mitenkään välttämätöntä.

**Avainsanat:**MatlabDy, diffyhtälöt, mlDifferentiaali(yhtälöt), populaationkasvumalli, *Malthus'n* laki

**Vastaavanlaisia tehtäviä:** `http://matriisi.ee.tut.fi/~piche/ode/` Pichet: TTY 1999, course 73107 (hyvä kokonaisuus):

1) Perusesim tähän kohtaan: *In 1980 the population of Altlantia was 254512; in 1995 it was 294726. What is its population in 2000.* (Vuosilukuihin voisi lisätä nyt vaikka 15.)

2) Jatkoa: mlD007.tex

mlODE004.tex [Maple-versio: ...mplODE004.tex] (iv3/2001, harj. 1, LV teht. 1-2)

Laskuvarjohyppääjän yhtälö. Oletetaan, että hyppääjän + varustuksen massa = $m$ ja ilmanvastus on verrannollinen nopeuden neliöön, olkoon verrannollisuuskerroin = $b$. Tällöin Newtonin 2. laki antaa liikeyhtälön:

$$mv' = mg - bv^2.$$

Olkoon yksinkertaisuuden vuoksi $m = 1, b = 1$ ja $g = 9.81 m/s^2$.

Piirrä suuntakenttä.

Oletetaan, että laskuvarjo aukeaa, kun $v = 10 m/s$, valitaan tämä alkuhetkeksi $t = 0$. Piirrä tämä ratkaisukäyrä suuntakenttäpiirrokseen. Yritä nähdä suuntakentästä, että kaikki ratkaisut näyttävät lähestyvän rajanopeutta $v \approx 3.13$ ja että ratkaisut ovat joko kasvavia tai pieneneviä (ja millä alkuarvoilla mitäkin, ja mitä tarkoittaa fysikaalisesti)

Määritä rajanopeus suoraan yhtälöstä.

Käytä Matlab-piirroksiin funktiota `dfield8` ja Maplessa DEtools-kirjaston DEplot-funktiota.

**Vihje** Kts. [HAM] ss. 169-170 tai `?DEplot`

```
> with(DEtools)
> with(plots)
```

Suuntakenttään:*DEplot*,
grafiikkojen yhdistämiseen: *display*.

**dfield-ohje:**
Hae m-tiedosto *dfield8* sivulta http://math.rice.edu/∼dfield/ ja sijoita se Matlab-polkusi varrelle.
Kirjoita Matlab-istuntoon : `dfield8`

**Avainsanat:** MatlabDy, MapleDy, diffyhtälöt, suuntakenttä, isokliinit, mplDifferentiaali(yhtälöt), mlDifferentiaali(yhtälöt)

**Viitteet:** `[HAM]` Heikki Apiola: Symbolista ja numeerista matematiikkaa Maple-ohjelmalla, Otatieto 588, 1998

---

mlODE005.tex (Pichet Course 73107)
Määritä diffyhtälön

$$x' = x^2 - t\,x + 4t$$

suuntakenttä ja pisteen $(x, t) = (-1, -1)$ kautta kulkeva ratkaisukäyrä graafisesti funktion *dfieldxx* avulla, missä $xx = 8$ (v. 2012).
Mikä on tämän funktion minimipisten likiarvo (vaikka 2:n tai 3:n numeron tarkkuudella).

**Vihje:**
Hae m-tiedosto *dfield8* sivulta http://math.rice.edu/∼dfield/ ja sijoita se Matlab-polkusi varrelle.

Kirjoita Matlab-istuntoon : `dfield8`
Kun sinulla on yhtälö, suuntakenttä ja ratkaisukäyrä, voit valita `Edit/Zoom in`, jolla pääset tarkentamaan minipisteen hakua (askeleen kerrallaan).
Ennen minimin hakua kannattaa poistaa piirtämäsi ratkaisukäyrät `Options/Erase solutions`, niin `Options/Keyboard input`-valinnalla voit antaa tarkan alkuarvon.

**Vaativuus:** 1+
**Tehtävän Latex-koodi:**
../mlteht/mlODE/mlODE005.tex

**Avainsanat:** mlODE,Matlabdiffyht,differentiaaliyhtaloita Matlab:lla,suuntakenttä, dfield8

**Matlabfunktioita:** ode45,dfield8(Rice University)

---

**34.** mlD006.tex
Kun Marsiin laskeutuvan luotain laukaisee laskeutumisrakettinsa, sen nopeutta kuvaa differentiaaliyhtälö

$$\frac{dV}{dt} = g_m - \frac{k}{m}V^2,$$

missä $g_m = 3.688$ on Marsin painovoimakerroin, $k = 1.2$ on aerodynaaminen vastusvoima, ja $m = 150$ on luotaimen massa kilogrammoissa. Käytä vapaan pudotuksen rajanopeutta alkuarvona (Marsissa rajanopeus on 67.056 m/s), ja ratkaise yhtälö välillä $[0 : 0.05 : 6]$. Piirrä sekä nopeuden että kiihtyvyyden kuvaajat.

**Vihje:**

mplODE017.tex, mlODE007.tex

Käyrän sovituksen yhteydessä huomasimme, että eksponentiaalinen kasvumalli, ns. *Malthus'n laki* $y' = ky$ ei toimi USA:n väestödataan pitkällä aikavälillä. Mallia voidaan tarkentaa lisäämällä sopiva kasvua rajoittava termi, tällöin johdutaan ns. logistiseen kasvulakiin:

$y' = ay - by^2$

USA:n väestödataan liityen *Verhulst* arvioi v. 1845 arvot $a = 0.03$ ja $b = 1.610^{-4}$, kun $t$ mitataan vuosissa ja väkiluku $y(t)$ miljoonissa.

**Opettajalle:** Tehtävä voidaan käsitellä ehkä luontevamminkin kokonaan erillisenä numeeristen diffyhtälöratkaisujen opetuksesta. Tällöin otetaan vain alla olevat kohdat (c) ja/tai (d).

(a) Ratkaise tehtävä ($y(0) = 5.3$) Eulerin menetelmällä käyttämällä askelpituussa $h = 10$

(b) rk4:llä käyttäen n. nelinkertaista askelta (voit kokeilla pienempiäkin)

(c) Matlabin `ode45`:llä.

(d) Laske analyyttinen ratkaisu Maplella. (Kyseessähän on *Bernoullin yhtälö*.)

Piirrä kuvia ja laske kaikissa tapauksessa ratkaisujen arvot annetuissa taulukkopisteissä. (ode45-tapauksessa onnistuu ainakin sovittamalla dataan splini funktiolla spline, joka on maailman helppokäyttöisin.)

kts. http://www.math.hut.fi/teaching/v/matlab/opas.htmlsplinit
(Nykyään (2012) ei tarvita erillistä splinisovitusta, laskentapisteet voidaan antaa suoraan ode45-funktiolle syötteenä.)

```
function [T,Y]=eulerS(f,Tspan,ya,n)
% Tämä vain kehittely- ja opettelutarkoituksessa.
% Funktio eulerV hoitaa niin skalaari- kuin vektoriversion.
%  (24.2.04, modifioitu 21.8.2010)
% Esim: y'=t+y, y(0)=1
%       f=@(t,y)t+y
%       [T,Y]=eulerS(f,[0 4],1,6), plot(T,Y,T,Y,'.r');shg
a=Tspan(1);b=Tspan(2);
h=(b-a)/n;
Y=zeros(n+1,1);T=(a:h:b)'; %Pystyvektorit yhdenmukaisesti ode45:n
Y(1)=ya;                   %  kanssa
for j=1:n
   Y(j+1)=Y(j)+h*f(T(j),Y(j));
end;
```

**Vaativuus:** 2+
**Tehtävän Latex-koodi:**
../mplteht/mplODE/mplODE017.tex

**Ratkaisu:** *** Hukassa, etsi/tee ***

**Viitteitä:**

http://math.aalto.fi/opetus/kp3-ii/06/L/L14dynumkalvot.pdf
http://www.math.hut.fi/ apiola/matlab/opas/lyhyt/esim/eulerS.m
(Listaus yllä)

**Avainsanat:** mplODE,Maplediffyht,differentiaaliyhtaloita Maplella, logistinen kasvu, numeeriset menetelmät, Eulerin menetelmä, Heunin menetelmä, Runge Kutta

**Maplefunktioita:** dsolve

**35.** mlD008.tex,mplD018.tex
Tarkastellaan yhtälöä $y' = -2\alpha(t-1)y$. Ratkaise aluksi analyyttisesti (saat käyttää Mapleakin.)

Totea kuvasta ja derivaattaehdosta yhtälön stabiilisuus/epästabiilisuusalueet. Ota kuvassa ja aina tarvittaessa vaikkapa $\alpha = 5$.
Ratkaise yhtälö sekä Eulerilla että BE:llä. Sopivia arvoja voisivat olla vaikkapa $h = 0.2$, väli: $[1, 4.5]$, $y(1) = 1$.
Vertaa kokeellisesti stabiilisuukäyttäytymistä teorian ennustamaan ja pane merkille, miten epästabiilisuus käytännössä ilmenee.
Tämä tehtävä soveltuu erityisen hyvin Maplella tehtäväksi, se on pitkälle ideoitu [HAM] sivulla 124, myös Euler ja BE ovat valmiina. (Koodit saa kurssin maple-hakemistosta.) ** Tulee aputiedostoon **

** apu puuttuu, editoi viitteet! **

**Vihje:**

**Viitteitä:**

`http://math.aalto.fi/opetus/kp3-ii/06/L/L14dynumkalvot.pdf`
`http://www.math.hut.fi/~apiola/matlab/opas/lyhyt/esim/eulerS.m`
(Listaus yllä)

**36.** Maple,Matlab (H2T10)

a) Ratkaise alkuarvotehtävä

$$y' - y = \cos x, \quad y(0) = 1$$

analyyttisesti Maplella ja numeerisesti Matlabilla. Piirrä ratkaisukäyrä.

b) Anna alkuarvoksi symboli $c$ ja piirrä ratkaisukäyräparvi sopivalla välillä, kun $c = -0.9, -0.8, \ldots, 0$.

Miltä parvi näyttää suurilla $x$ :n arvoilla. Tässä pitäisi erottua kolmenlaista käytöstä.

**Vihje:** Maple: dsolve, Matlab: ode45

**Avainsanat:** Differentiaaliyhtälö, alkuarvotehtävä, analyyttinen ratkaisu, numeerinen ratkaisu.

**37.** mplD007.tex (iv3/2001, harj. 2, AV teht. 1)
Ratkaise (AA)-tehtävä $y' - 2xy = 1, \quad y(0) = -0.5$

Tässä näyttää siltä, että (EHY):n erikoinen olisi helppo löytää, mutta huomaat pian, että luonnolliset yritteet eivät toimi. (Kyseessähän on lineaarinen, mutta ei-vakiokertoiminen yhtälö.)

Ratkaise vaan sitten kiltisti integroivan tekijän menettelyllä.

Integrointi johtaa *erf*-funktioon, Maple antaa sen suoraan, voit myös konsultoida KRE-kirjaa hakusanalla *erf*. Lausu siis ratkaisu *erf*:n avulla.

Piirrä suuntakenttäpiirros Maplen `DEtools`-pakkauksen `DEplot`-funktion avulla (kts [HAM] s. 169), voit toki käyttää myös Matlab:n dfield8-funktiota (ohje alla).

Valitse alkuarvoja $y_0$ väliltä $(-1, -0.5)$ yrittäen löytää kriittistä arvoa $y_0$, joka jakaa ratrkaisukäyrät plus tai miinus ääretöntä lähestyviin. (Tuo kriittinen ratkaisukäyrä on rajoitettu.) Käytä hyväksesi *erf*-funktion ominaisuutta $\lim_{x\to\infty} erf(x) = 1$ laskeaksesi tarkan arvon $y_0$:lle.

**Vihje:** dfield-ohje: Hae m-tiedosto *dfield8* sivulta `http://math.rice.edu/~dfield/` ja sijoita se Matlab-polkusi varrelle.
Kirjoita Matlab-istuntoon : `dfield8`

**Avainsanat:** MapleDy, diffyhtälöt,erf, mplDifferentiaali(yhtälöt)

**Viitteet:** `[KRE]` E. Kreyszig: Advanced Engineering Mathematics, Wiley
`[HAM]` Heikki Apiola: Symbolista ja numeerista matematiikkaa Maple-ohjelmalla, Otatieto 588, 1998.

**38.** a) Piirrä tasokäyrä $(x, y) = (e^{-t/20} \cos t, e^{-t/10 \sin t})$ (faasitaso).

b) Piirrä edelliset koordinaattikäyrät $x(t)$ ja $y(t)$ (aikataso).

**Vihje:** Muista `(.*)`, kun muodostat vektorilausekkeita. kokeile eri parametrivälejä. Kokeile `linspace(a,b,n)`:ssä muitakin kuin oletusarvoa $n = 100$. Käytä `axis`-komentoa. Ja kysele `help`:llä.
Uusi grafiikkaikkuna: `>> figure`, Useampia samaan kuvaan: `>> hold on`

**Avainsanat:** Differentiaaliyhtälöryhmä, aikakuva,faasikuva,grafiikka,plot

mlODE100.tex
Mallinnetaan heiluria kolmessa ulottuvuudessa. Pisteen $\mathbf{x}(t) = [x_1(t), x_2(t), x_3(t)]^T$, massa on $m$, ja siihen kohdistuu voima $\mathbf{F} = [0, 0, -gm]^T$ ($g = 9.81 m/s^2$). Pistettä rajoittaa pallon pinnan yhtälö $\Phi(x) = x_1^2 + x_2^2 + x_3^2 - 1 = 0$. Pisteen liikettä mallintaa seuraava differentiaaliyhtälö:

$$\mathbf{x}''(t) = \frac{1}{m}\left(\mathbf{F} - \frac{m\mathbf{x}'(t)^T \mathbf{H}\mathbf{x}'(t) + \nabla\Phi\mathbf{F}}{|\nabla\Phi|^2}\nabla\Phi\right)$$

Yhtälössä $\nabla\Phi$ on funktion $\Phi$ tilagradientti ($x_i$ komponenttien suhteen) ja $\mathbf{H}$ on Hessin matriisi funktiolle $\Phi$ – tässä tapauksessa diagonaalimatriisi, jonka alkiot ovat kaikki 2.

Lisäksi yhtälö tarvitsee alkuarvot $\mathbf{x}(0) = \mathbf{x}_0$ ja $\mathbf{x}'(0) = \mathbf{v}_0$.

Ratkaise yhtälö kertalukua tiputtamalla, ja käyttämällä MATLAB-ratkaisinta `ode45`. Tarkastele tulosta graafisesti: ovatko trajektorit sitä mitä odotit? Ratkaise ongelma sitten käyttämällä metodia `ode23`. Vastaavatko tulokset nyt sitä mitä odotit?

**Vaativuus:** 3+
**Tehtävän Latex-koodi:**
../mlteht/mlODE/mlODE100.tex

**Avainsanat:** mlODE,Matlabdiffyht,differentiaaliyhtaloita Matlab:lla, 3d-heiluri

**Matlabfunktioita:** ode45,ode23, odexx

---

mlODE102
Ratkaise toisen kertaluvun alkuarvotehtävä

$$y'' - 0.05y' + 0.15y = 2t; y'(0) = 0, y(0) = 0.$$

Matlabin ratkaisijoita (`ODE45, ODExx`) varten 2. kertaluvun yhtälö tulee muuntaa 1. kertaluvun systeemiksi määrittelemällä $y_1 = y, y_2 = y'$, jolloin saadaan systeemi:

$$\begin{cases} y_1' = y_2 \\ y_2' = ... \end{cases}$$

Piirrä ratkaisukäyrä ja derivaatta sekä faasitasokuva.

**Vaativuus:** 1+
**Tehtävän Latex-koodi:**
../mlteht/mlODE/mlODE102.tex

**Avainsanat:** mlODE,Matlabdiffyht,differentiaaliyhtaloita Matlab:lla, numeeriset menetelmät, faasitaso

**Matlabfunktioita:** ode45,odexx

---

**39.** Eulerin menetelmä on klassinen menetelmä differentiaaliyhtälöiden ratkaisemiseksi. Menetelmän lähtökohtana on yhtälö

$$y'(t) = f(t, y(t)); y(t_0) = y_0,$$

josta koetetaan ratkaista $y(t)$ kun $t = [t_0, t_n]$. Menetelmä perustuu $t$:n diskretoinnille: $t$ muunnetaan joksikin vektoriksi $T = (t_0, t_0 + h, \ldots, t_n)$, jonka jälkeen koetetaa etsiä vastaavia arvoja $y(t_0 + ih)$. Merkintöjen selvyyden vuoksi kirjoitetaan $t_0 + jh = t_j$. Eulerin menetelmässä approksimoidaan arvoa $y(t_i)$ seuraavasti:

$$y(t_i) = y(t_{i-1}) + h\, f(t_{i-1}, y(t_{i-1})).$$

Ratkaise Eulerin menetelmällä differentiaaliyhtälö

$$y'(t) = y(t) \sin(t).$$

Käytä eri $h$:n arvoja, ja vertaa oikeaan ratkaisuun $e^{1-\cos(t)}$.

**Vihje:** Eulerin menetelmä MATLABissa on helpointa toteuttaa yksinkertaisena silmukkana. Olettaen, että yhtälön oikea puoli on kirjoitettu funktioon `f(t,y)`, Euler-iteraatio voidaan tehdä seuraavasti:

```
for n = 2:length(t)
    y(n) = y(n-1)+h*f(t(n-1),y(n-1));
end
```

missä $t$ on vektori joka on ratkaisuvälin diskretointi, ja $h$ on $t$:n pisteiden välinen etäisyys.

**40.** Rungen-Kutan menetelmä differentiaaliyhtälöiden ratkaisemiseksi on huomattavasti kehittynyt versio Eulerin menetelmästä. Menetelmässä määritellään seuraavasti:

$$\begin{aligned}
k_1 &= f(x_n, y_n),\\
k_2 &= f(x_n + \tfrac{h}{2}, y_n + \tfrac{1}{2}hk_1),\\
k_3 &= f(x_n + \tfrac{h}{2}, y_n + \tfrac{1}{2}hk_2),\\
k_4 &= f(x_n + h, y_n + hk_3).
\end{aligned}$$

Diskretointiaskeleeksi tulee nyt

$$y_{n+1} = y_n + hk = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4).$$

a) Ratkaise edellisen tehtävän yhtälö

$$y'(t) = y(t)\sin(t); y(0) = 1.$$

käyttämällä Rungen-Kutan menetelmää. Vertaa tulosta sekä todelliseen ratkaisuun $y(t) = e^{1-cos(t)}$, että Eulerin menetelmällä saavutettuun. Mitä huomaat tarvittavasta askelkoosta?

b) Ratkaise alkuarvotehtävä

$$y'(t) = \sqrt{1-y^2}; y(0) = 0$$

käyttämällä Rungen-Kutan menetelmää.

**Vihje:**

**41.** Ratkaise differentiaaliyhtälöryhmä

$$\begin{cases}
\frac{dx}{dt} = -\sigma x + \rho y\\
\frac{dy}{dt} = \sigma x - y - xz\\
\frac{dz}{dt} = -\beta z + xy
\end{cases}$$

numeerisesti välillä $[0, 20]$, kun $\sigma = 10, \rho = 28$ ja $\beta = 8/3$. Piirrä ratkaisukäyrät samaan kuvaan, ja piirrä käyrät $x(t)$ ja $z(t)$ parametrisesti. Tämän jälkeen piirrä 3-ulotteinen parametrisoitu käyrä kaikista koordinaateista.

Onko ratkaisu rajoitettu? Suppeneeko se kohti jotain arvoa?

Kokeile muuttaa alkuarvoja, sekä parametrien arvoja. Vallitsevan teorian mukaan systeemi on *kaoottinen dynaaminen systeemi*, jonka käyttäytyminen voi muuttua merkitsevästi jo pienistä muutoksista lähtötilanteessa; itse asiassa termi perhosvaikutus keksittiin kuvaamaan juuri tämän systeemin käytöstä.

**Vihje:** Kolmiulotteinen parametrisoitu käyrä (tai pistejoukko) piirretään MATLABissa funktiolla `plot3`.

**42.** Kirjoita heiluriyhtälö $\Theta'' + \frac{g}{L}\sin(\Theta) = 0$ ensimmäisen kertaluvun systeemiksi ja samantien Matlab-funktioksi (joko inline tai m-tiedosto). Voit ottaa $g/L = 1$.

Laske ratkaisu sopivalla aikavälillä (esim. $[0, 10]$) ja kolmella erilaisella alkuarvolla, joilla saat erityyppiset ratkaisut. Käytä `ode45`-funktiota.

Piirrä ratkaisukäyrät aikatasoon ja trajektorit faasitasoon.

**43.** **Reuna-arvotehtävä tähtäysmenetelmällä** (Moler odes teht. 7.19 ss. 45–47.) Olkoon ratkaistavana reuna-arvotehtävä $y'' = y^2 - 1, \quad y(0) = 0, \quad y(1) = 1$. Tee Matlab-funktio, joka ottaa argumentikseen "alkunopeuden"0:ssa ja palauttaa vastaavan AA-tehtävän ratkaisun arvon 1:ssä, vähennä siitä vielä 1, niin sinulla on funktio, jota sopii tarjota `fzero`:lle.

**44.** Ratkaise differentiaaliyhtälö

$$\frac{\partial u(t,x)}{\partial t} = -u - 5e^{-t}\sin(5t)$$

nelikulmiossa $[0, 5] \times [-2, 2]$, alkuarvokäyrällä $u(0, x) = e^{-x^2}$.

Piirrä tuloksista kuva:

**Vihje:** Määrittele ensin sopivan harva diskretaatio välille $[-2, 2]$, ja sen jälkeen tähän diskretaatioon sopiva alkuarvokäyrä, sen jälkeen ratkaise differentiaaliyhtälö numeerisesti (`ode45`) vuoroittain kullekin alkuarvolle.

**45.** Ratkaise nk. *Rösslerin systeemi*

$$\begin{cases} y_1'(t) = -y_2(t) - y_3(t), \\ y_2'(t) = y_1(t) + a y_2(t), \\ y_3'(t) = b + y_3(t)(y_1(t) - c), \end{cases}$$

missä $a, b, c$ ovat parametreja, välillä $t \in [0, 100]$. Käytä alkuarvoja $y(0) = [1, 1, 1]^T$ ja $(a, b, c) = (0.2, 0.2, 2.5)$.

Kyseessä on kaoottinen systeemi, ts. ratkaisurata riippuu suuresti joko alkuarvoista ja parametreista. Saatuasi ratkaisun, piirrä ratkaisun kuvia alkuarvon muuttujana, ja tutki kuinka pienet muutokset aiheuttavat havaittavia eroja.

**46.** Ratkaise differentiaaliyhtälö

$$y'(t) = -\frac{1}{t^2} + 10\left(y - \frac{1}{t}\right); y(1) = 1.$$

käyttäen Backward-Euler menetelmää:

$$y_{n+1} = y_n + hf(t_{n+1,y_{n+1}}).$$

Jotta tämä onnistuisi, tulee jokaisella iteraation askeleella ratkaista yhtälö

$$y_{n+1} - hf(t_{n+1,y_{n+1}}) = y_n$$

$y_{n+1}$:n suhteen. Tämä tarkoittaa, että yleisen ratkaisimen kirjoittaminen olisi vaikeaa, mutta tässä eksplisiittisessä tapauksessa se onnistuu.

Vertaa sitten saamaasi tulosta MATLABin `ode45` funktiolla saamaasi: kuinka selität eron?

**Vihje:**

mlODExxx.tex
Template file
Let

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

What can you say ...

**Vaativuus:** 1 - 3+
**Tehtävän Latex-koodi:**
../mlteht/mlLinalg/mlODExxx.tex

**Avainsanat, keywords:** Matlab:ODE, tavalliset differentiaaliyhtälöt, differential equations, ODE45, ODExx

---

## mlPDE, Osittaisdifferentiaaliyhtälöitä Matlab:lla

mlPDE000.tex

Tähän tiedostoon kootaan laskaripaperiin sopivia pikaohjeita ja mielen virkistyksiä.

- .
- ..
- ...

mlPDE001.tex [mlLA006.tex]

Tehtävä sopii myös puhtaasti lineaaristen yhtälöryhmien harjoitteluun, viittaus: Lineaarialgebraosiossa: mlLA006

*Tasapainolämpötilajakauma metallilevyssä.*
Kuva esittää metallilevyä, joka on ylä- ja alapinnoiltaan lämpöeristetty ja jonka reunojen lämpötilat on kiinnitetty. (Lämpöä virtaa vain reunojen kautta.) Tasapainolämpötilajakauma saadaan *Laplacen yhtälön* $\Delta u = 0$ ratkaisuna. Numeerinen approksimaatio voidaan laskea ns. differenssimenetelmällä: Jaetaan levy sopivilla hilaviivoilla osiin ja numeroidaan näin muodostuvat solmupisteet. Menetelmä: Kunkin hilasolmun lämpötila on naapurisolmujen lämpötilojen keskiarvo. (Johdetaan kurssin lopulla.)

Muodosta $4 \times 4-$ yhtälösysteemi solmujen $1, 2, 3, 4$ lämpötilojen likiarvoille $u_1, u_2, u_3, u_4$. Ohje: Aloitetaan solmusta 1: $u_1 = \frac{1}{4}(30 + u_2 + u_3 + 10)$. Vastaavasti muut kolme solmua.

```
          20       20
   |----- o  ----  o ------|
   |                       |
   |                       |
10 o      o 3      o 4     o  40
   |                       |
   |                       |
10 o      o 1      o 2     o  40
   |                       |
   |                       |
   |----- o ------ o ------|
          30       30
```

(a) Ratkaise yhtälösysteemi ja sijoita ratkaisulämpötilat ao. hilapisteisiin.

(b) Muodosta $4 \times 4-$ matriisi, jossa on reunalämpötilat ja ratkaisemasi sisäpistelämpötilat sekä nurkissa lähinnä olevien kahden reunasolmun keskiarvot tähän tapaan: `U=[15 20 20 30;10 u3 u4 40; 10 u1 u2 40; 20 30 30 35];` Piirrä ratkaisupinnan approksimaatio: `mesh(U)` tai `surf(U)`.

(b') Ehka hiukan selkeämpää on rakentaa U-matriisi vaiheittain vaikka tähän tapaan:

```
u=u'  % Vaakavektoriksi
U=zeros(4,4)
U(1,:)=[15 20 20 30]
U(2,:)=[10 u(3:4) 40]
...
...
```

**Vaativuus:** 2
**Tehtävän Latex-koodi:**
../mlteht/mlPDE/mlPDE001.tex

**Ratkaisu:**
../mlteht/mlPDE/ratkaisut/html/mlPDE001R.pdf
../mlteht/mlPDE/ratkaisut/mlPDE001R.m

**Avainsanat:** Lämpötilamatriisi, Laplacen yhtälön diskretointi, differenssimenetelmän perustehtävä, lineaarinen yhtälöryhmä, Discretizaton of Laplace PDE, Linear system of equations, mlPDE,mlLA

---

mlPDE002.tex, [Maple:mplLinalg/mplLA010.tex]
Lay: Linear Algebra p. 12 probl. 35

Tehtävä sopii myös puhtaasti lineaaristen yhtälöryhmien harjoitteluun.

Tarkastellaan lämmönjohtumista ohuessa metallilevyssä. Oletetaan, että johtumista tapahtuu vain levyn suunnassa, ja levyn reunoilla on annetut (ajan suhteen) vakiolämpötilat. Levyn lämpötilat eri pisteissä asettuvat ajan kuluessa arvoihin, jotka ovat ajan suhteen vakioita, tällöin puhutaan lämpötilajakauman tasapainotilasta ("steady state"). Tehtävänä on määrittää lämpötilajakauma levyssä tasapainotilan vallitessa.

Tarkastellaan kuvan mukaista tilannetta:

```
 --- 20----20---20----
 |    |     |     |    |
10----*-----*-----*---40
 |    |     |     |    |
10----*-----*-----*----40
 |    |     |     |    |
 ----20----20----20---
```

Kuvassa näkyvät annetut vakioreunalämpötilat (reunaehdot). Tehtävänä on laskea ratkaisuapproksimaatiot *:llä merkityissä sisäsolmupisteissä käyttäen seuraavaa periaatetta: Lämpötila levyn solmupisteessä on naapurisolmujen lämpötilojen keskiarvo.

Jos indeksoidaan solmupisteiden lämpötilat vaakarivijärjestyksessä: $T_1, \ldots T_6$, voidaan ryhtyä kirjoittamaan yhtälöitä tyyliin:

$T_1 = \frac{20+10+T_4+T_2}{4}, \ldots.$

Kirjoita koko $6 \times 6$- yhtälösysteemi "standardimuodossa".
**Huom:** Tasapainotilaratkaisu saadaan ns. *Laplacen yhtälön* $\nabla^2 T = 0$ ratkaisuna. Tässä esitettyyn likimääräismenettelyyn ns. *differenssimenetelmään*

**Vrt:** Tehtävä mlPDE001 sisältää vastaavan suorituksen jatkokäsittelyohjeineen ja ratkaisuineen. Samanlaiset ohjeet sopivat tähän.
** LINKKI **

**Vaativuus:** 2
**Tehtävän Latex-koodi:**
../mlteht/mlPDE/mlPDE002.tex

**Avainsanat:** Lämpötilamatriisi, Laplacen yhtälön diskretointi, differenssimenetelmän perustehtävä, lineaarinen yhtälöryhmä, Discretizaton of Laplace PDE, Linear system of equations, mlPDE,mlLA, Osittaisdifferentiaaliyhtalo, Partial differential equation, Matlab

---

mlPDExxx.tex
Template file
Let

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

What can you say ...

**Vaativuus:** 1 - 3+
**Tehtävän Latex-koodi:**
../mlteht/mlLinalg/mlPDExxx.tex

**Avainsanat, keywords:** mlPDE, Matlab: PDE, partial differential equations, osittaisdifferentiaaliyhtälöt Matlab:lla

---

-e

---

## matlabteht/mlTodari, Todennäköisyys, satunnaisuus, tilasto

---

mlPr003.tex
Vrt. mlPr003.m

Satunnaisvirhettä sisältävää dataa voidaan Matlabissa muodostaa seuraavaan tapaan:

```
num=100;
v=0.1;
% rng(0,'twister')   Tällä voisit alustaa saunnaislukugeneraattorin
                     toistettavaan tilaan.
data=5+v*sign(1-2*rand(1,num)).*rand(1,num);
```

Selitä, mitä viimeinen rivi tekee.

(a) Piirrä datapisteet ja histogrammi, myös isommalla datalla.

(b) Vaihda viimeinen `rand` muotoon `randn`, ja tee sama kuin edellä.

(c) Kirjoita ohjelma, joka laskee keskiarvon ja hajonnan. Tee oma funkio, voit verrata valmiisiin `mean` ja `std`- funktioihin.
`for`-looppeja ei hyväksytä.

(d) Sovella funktio(i)tasi edellä muodostettuun dataan.

**Vihje:** Jos haluat tehdä toistettavan satunnaiskokeen `rand`-tyyppisillä funktioilla, voit antaa ensin komennon: `rng(0,'twister')`.
Tässä tehtävässä on havainnollisempaa olla antamatta tätä, ainakin useimmissa vertailuissa.

**Vaativuus** 2
**Avainsanat:** Matlab, todennäköisyys, probability, mlPr, satunnaisluvut, random, matlabperusteet, Matlabfunktioita:rand,randn,hist,bars

mlPr004.tex
Laskemme yksikkökolmion $T$ (virittävät pisteet $(0,0),(1,0),(0,1)$) pinta-alan tasaisesti jakautuneilla satunnaisluvuilla Monte-Carlo menetelmää mukaillen:

1. Generoi $N$ tasaisesti jakautunutta satunnaislukuparia $(x_1, x_2)$ yksikköneliöön.
2. Selvitä, kuinka moni valitsemistasi satunnaispisteistä osuu kolmion $T$ sisälle. Havainnollista tätä piirtämällä $T$:n sisälle osuvat pisteet ja $T$:n ulkopuoliset pisteet samaan kuvaan eri väreillä.
3. Approksimoi $T$:n alaa laskemalla kolmion sisälle osuneiden pisteiden osuus kaikista valituista. Kokeile menetelmän tarkkuutta eri arvoilla $N$.

**Vihje:**
Funktion `rand` generoi tasaisesti jakautuneita satunnaislukuja.
On useita keinoja tutkia, osuuko piste kolmion sisään.

1. Voit käyttää `for-` silmukkaa. Vähiten suositeltava tapa (mutta opettaa kuitenkin ohjausrakenteita, selkeästi Matlabin "väärinkäyttöä").
2. Muodosta saunnaisvektorille ehto kolmioon kuulumiselle ja käytä loogista indeksointia. Oikeaoppinen Matlab-tyyli (tehokas sekä ajatuksellisesti että suoritusajassa).
3. Funktio `inpolygon` on huomattavan monipuolinen funktio. Yleistyskelpoinen erilaisille monikulmioalueille. Kuuluu pikemminkin luokkaan "hyvä tietää" kuin tässä tarkoitettuun Matlab-perusoppiin. Mutta on mielenkiintoinen ja kokeilemisen arvoinen tässäkin yhteydessä.

**Vaativuus** 2
**Avainsanat:** Matlab, Probability, mlPr, satunnaisluvut, random, matlabperusteet, Monte Carlo, looginen indeksointi

mlPr005.tex
Monte Carlo-approksimaatio $\pi$:lle.

Piirrä kuva

```
t=linspace(0,2*pi);
x=cos(t);y=sin(t);
plot(x,y,[1 1 -1 -1 1],[-1 1 1 -1 -1]);
axis([-1.5 1.5 -1.5 1.5])
axis square
```

Heitetään tikkaa kuvan mukaiseen tauluun (tikat eivät eksy taulua ympäröivään neliön ulkopuolelle, ehkä tähän oikeasti tarvitaan "satunnaisrobotti"). Jos tikkojen osumatarkuus on satunnaismuuttuja, joka on tasajakautunut neliöllä
$-1 < x <, 1 < y < 1$, niin ympyrään ja neliöön osuneiden tikkojen lukumäärän suhde lähenee lukua $\pi/4$, kun satunnaisheittojen lukumäärä kasvaa. Miksi? Generoi tasajakautuneita pistepareja ja laske ko. osuus.

**Vihje:** Käytä loogista indeksointia, kun ehtona on `X.^2+Y.^2 <= 1`, ja laske bittivektorin ykkösten lukumäärä.

Jatkolukemista ja lisätehtäviä:

Kirjassa
*Charles van Loan:* Introduction to Scientific Computing, a Matlab approach
on hyvä tiivis selvitys aiheesta "Random processes"1.3.2 ss. 34 - 37.

**Vaativuus** 2
**Avainsanat:** Matlab, Probability, mlPr, satunnaisluvut, random, matlabperusteet, Monte Carlo, looginen indeksointi
**Matlabfunktioita:** rand

**47.** Olkoot $F^n$ satunnaismuuttuja joka kuvaa kiinteiden pisteiden lukumäärää satunnaispermutaatiossa (so. alkiot joiden paikka ei muutu permutaatiossa).

a) kirjoita funktio joka ottaa argumenttina kokonaisluvun $n$ ja palautaa $k$-pituisen otoksen jakaumasta $F^n$.

b) Generoi otoksia jakaumasta $F^n$ eri arvoilla $n$.

c) Piirrä histogrammit eri otoksista.

d) Voidaanko histogrammien perusteella päätellä, mikä on $E[F^n]$?

e) Laske $E[F^n]$

**Vihje:** Käytä hyväksesi odotusarvon lineaarisuutta laskiessasi odotusarvoa $E[F^n]$.

mlPrxxx.tex
Template file
Let

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

What can you say ...

**Vaativuus:** 1 - 3+
**Tehtävän Latex-koodi:**
../mlteht/mlLinalg/mlPrxxx.tex

**Avainsanat, keywords:** mlPr, Matlab:probability, todennäköisyyslaskentaa Matlab:lla