

-e

mlCurveFit

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

Hermiten interpolaatio: Interpolaatioehdoissa esiintyy myös derivaattoja.

Opettajalle: *Tehtävän alkuperäinen tarkoitus on demonstroida luontevaa, nautittavaa **Maple**-työskentelyä. Sopii myös oikein hyvin **Matlab**-tehtäväksi. Matlabissa on valmiina polyderfunktio, jonka ohjelmointi on sinänsä myös oikein sopiva pikku harjoitustehtävä. Tässä tehtävässä ei ole tarvetta/syytä käyttää *symbolic toolboxia*.*

Määritä 4. asteen polynomi p , joka toteuttaa ehdot:

$$p(0) = p'(0) = 1, p(1) = p'(1) = p''(1) = 2.$$

(a) Käsittele polynomi lausekkeena.

Tarkista tulos sopivasti `subs`-komennoilla ja piirrä kuva/kuvia polynomista ja derivaatoista.

(b) Käsittele polynomi funktiona.

Huom: 5 ehtoa ja 5 tuntematonta kerrointa \implies 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 “Vandermondin neliömatriisi” on aina ei-singulaarinen).

Tässä opetellaan erityisesti Maplen kätevää ratkaisutekniikkaa.

Vihje: Kirjoita polynomi lausekkeeksi tyyliin:

```
p:=a*x^4+b*x^3 + . . . . ,  
missä  $a, b, \dots, e$  ovat määrättävät kertoimet.
```

Derivaatta: `diff`

Arvojen ($x=0, x=1$) sijoittaminen p :n lausekkeeseen: `subs`

Yhtälön ratkaiseminen: `solve`

Kaikista saat tietoa näin `?diff, ...`

2. Oletetaan, että meille on annettu dataa muodossa $(x_k, y_k), k = 1 \dots m$, 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

$$y = k_1x + b_1, x < s$$

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

$$y = k_2x + b_2, x > s$$

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 indeksöintiä: esimerkiksi valitaan kaikki vektorin pisteet, jotka ovat pienempiä kuin 5.

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

3.

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(doc) polyfit, polyval` .

Sinun on tiedettävä, mikä on polynomien asteluku.

Tarkistus: Kulkeeko polynomi kaikkien datapisteiden kautta.

4. 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.

Vihje: Olkoon $p(x) = a_0 + a_1x + \dots + a_nx^n$ etsitty polynomi.

Määritetään tuntemattomat kertoimet interpolaatioehtojen $p(x_k) = y_k, k = 0, \dots, x_n$ avulla saatavasta lineaarisesta yhtälösystemistä ratkaisemalla.

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(...);          % Polynomien 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 tehtäväkoelman interpolaatiotehtävään.

5. 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 $[-1, 1]$

1. Piirrä datapisteet.
2. Muodosta interpolaatiopolynomi ja piirrä samaan kuvaan.
3. Muodosta asteita 2,3,4 olevat PNS-polynomit ja piirrä samaan kuvaan
4. Sovita vielä splini ja piirrä samaan.
5. 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`

Vihje:

Ratkaisu: mlCF04ratk.m [Tulee, numero?]

6.

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

b) Kirjoita suorat yhtälö $A[x; y] = b$ PNS-mielessä, ja piirrä ratkaisu samaan kuvaan suorien kanssa.

Vihje: Samaan kuvaan piirtäminen onnistuu komennolla `hold on`. PNS-ratkaisu tehdään MATLABin `backslash`illa.

7. `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.

8.

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 muotoa $(x-c_1)^2+(y-c_2)^2=r^2$. 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 muotoa

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

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))`.

9. Maple tai Matlab

Tutkitaan nk. Runge'n ilmiötä. Laske funktion $g(x) = 1/(1+x^2)$ arvoja tasaisin välein väliltä $[-5, 5]$, ja tee näihin pisteisiin perustuva polynominen 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

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

mukaan.

Vihje: Polynominen interpolaatio kannattaa tehdä MATLAB-funktiolla `polyfit`. Funktio g kannattaa määrittellä funktiokahvan avulla: `g = @(x)1./(1+x.^2)`. Tasavälisiä pisteistä saa funktiolla `linspace`

Sopii aivan yhtä hyvin Maplelle.

10. H2T14.tex/mlCF13.tex/mplCF13.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

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

Sovita myös eriasteisia PNS-polynomeja, vrt. Matlab Censugui, lue Molerista: <http://www.mathworks.se/Comp.with/Matlab/interpolation>

Vihje:

11. 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 ratkaisutiedoissa.

(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 (Lagrange'n) interpolaatiokaavan virhetermin avulla interpolaatiovirheen yläraja y_0 välillä ja vertaa todelliseen.

Lause Olkoot x_0, x_1, \dots, 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 tulotermin 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 matriisijattelua, muistiahan ei nykyisin tarvitse säästellä.)

Avainsanat: Interpolaatio, käyrän sovitus, interpolaatiovirhe, Lagrange

12. 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 menetelmää.

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.

13. mlCF20.tex

Pienimmän neliösumman sovitus, PNS, LSQ

Lineaarialgebra-osioissa on aiheesta myös joitakin perustehtäviä.

Tähän tiedostoon kootaan laskaripaperiin sopivia pikaohjeita ja mielen virkistyksiä.

- .
- ..
- ...

14. 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.

-e

mlDifferentiaali(yhtälöt)

15. 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 $y_i - x_j$, $i = 1 \dots 5, j = 1 \dots 5$ 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
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 piirroksen. Tarkista käsin (tai “skalaa-tilaskimella” (Matlabkin käy)) muutama alkio ainakin. (Hilan tihentäminen käy nyt helposti muuttamalla vain yllä h :ta.)

Myöhemmin opimme, että K -matriisin muodostaminen käy kätevämmiin, tehokkaampiin ja rutiinimaisempiin 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/>)

Lopuksi voit kokeilla LAODE-funktiota `dfield8` (versio v. 2012) , jonka käyttö ei vaadi Matlabin tuntemista lainkaan. <http://math.rice.edu/~dfield/>

Avainsanat: `matlabDiffyhtälöt`, `mlbDifferentiaali`,suuntakenttä

Viitteet: [LAODE] Golubitzky-Dellnitz: Linear Algebra and Differential Equations using Matlab, Brooks/Cole 1999.

16. `mlD002.tex` [vastaava Maple: `...mplD003.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`

Avainsanat: `MatlabDy`, `diffyhtälöt`, suuntakenttä, isokliinit, `mlDifferentiaali`(yhtälöt)

Viitteet:

[LAODE] Golubitzky-Dellnitz: Linear Algebra and Differential Equations using Matlab, Brooks/Cole 1999.

17. m1D003.tex (iv3/2001, harj. 1, teht. 3) (Jatkoa: m1D007.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 datasarakkeen perusteella (jolloin kyseessä ei enää ole tulevaisuuden ennustaminen), ja tutki tämän mallin käyttäytymistä muissa datapisteissä.

c) Havainnollista kuvilla, käytä tarpeen mukaan logaritmistä skaalaa.

Vihje: Voit valita ajan 0-hetken vuosiluvuksi 1800 (miksi?). Toki ei mitenkään välttämätöntä.

Avainsanat: MatlabDy, diffyhtälöt, m1Differentiaali(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 Atlantia was 254512; in 1995 it was 294726. What is its population in 2000.* (Vuosilukuihin voisi lisätä nyt vaikka 15.)

2) Jatkoa: m1D007.tex

18. m1D004.tex [Maple-versio: ...m1D004.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 $g = 9.81m/s^2$.

Piirrä suuntakenttä.

Oletetaan, että laskuvarjo aukeaa, kun $v = 10m/s$, valitaan tämä alkuhetkeksi $t = 0$. Piirrä tämä ratkaisukäyrä suuntakenttäpiirroksen. 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

19. mLD005.tex (Pichet Course 73107)

Määritä diffyhtälön

$$x' = x^2 - tx + 4t$$

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 minimipisteen hakua (askeleen kerrallaan).

Ennen minimin hakua kannattaa poistaa piirtämäsi ratkaisukäyrät `Options/Erase solutions`, niin `Options/Keyboard input`-valinnalla voit antaa tarkan alkuarvon.

Avainsanat: MatlabDy, diffyhtälöt, mlDifferentiaaliyhtälöt, suuntakenttä, `dfield8`

20. 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:**21.** mplD017.tex, mld007.tex

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 johdetaan 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 luontavammin 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 askelpituus $h = 10$

(b) rk4:llä käyttäen n. nelinkertaista askelta (voit kokeilla pienempiäkin)

(c) Matlabin ode45:llä.

(d) Laske analyyttinen ratkaisu Maplella (kyseessä 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/opus.html#splinit>

(Nykyään (2012) ei tarvita erillistä splinisovitusta, laskentapistet voidaan antaa suoraan ode45-funktiolle syötteenä.)

Vihje:

```
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;
```

Viitteitä:

<http://math.aalto.fi/opetus/kp3-ii/06/L/L14dynamkalvot.pdf>

<http://www.math.hut.fi/~apiola/matlab/opus/lyhyt/esim/eulerS.m>

(Listaus yllä)

22. 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 stabiilisuuskäyttäytymistä 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/L14dynamkalvot.pdf>

<http://www.math.hut.fi/~apiola/matlab/opas/lyhyt/esim/eulerS.m>

(Listaus yllä)

23.

Maple, Matlab (H2T10)

a) Ratkaise alkuarvotehtävä

$$y' - y = \cos x, \quad y(0) = 1$$

analyttisesti Maplella ja numeerisesti Matlabilla. Piirrä ratkaisukäyrä.

b) Anna alkuarvoksi symboli c ja piirrä ratkaisukäyräparvi sopivalla välillä, kun $c = -0.9, -0.8, \dots, 0$.

Miltä parvi näyttää suurilla x :n arvoilla. Tässä pitäisi erottua kolmenlaista käytöstä.

Vihje: Maple: dsolve, Matlab: ode45

Avainsanat: Differentiaaliyhtälö, alkuarvotehtävä, analyttinen ratkaisu, numeerinen ratkaisu.

24. mplD007.tex (iv3/2001, harj. 2, AV teht. 1)

Ratkaise (AA)-tehtävä $y' - 2xy = 1$, $y(0) = -0.5$

Tässä näyttää siltä, että (EHY):n erikoinen olisi helppo löytää, mutta huomaat pian, että luonnolliset yritteet eivät toimi. (Kyseessä 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`-funktioita (ohje alla).

Valitse alkuarvoja y_0 väliltä $(-1, -0.5)$ yrittäen löytää kriittistä arvoa y_0 , joka jakaa ratkaisukäyrät plus tai miinus ääretöntä lähestyviin. (Tuo kriittinen ratkaisukäyrä on rajoitettu.) Käytä hyväksesi *erf*-funktion ominaisuutta $\lim_{x \rightarrow \infty} erf(x) = 1$ 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.

25.

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`

26. Ratkaise toisen kertaluvun alkuarvot tehtävä

$$y''(t) - 0.05y'(t) + 0.15y(t) = 2t; y'(0) = 0, y(0) = 0.$$

Vihje: Voi olla avuksi määritellä $y_1 = y, y_2 = y'$, jolloin toisen kertaluvun tehtävä muuttuu ensimmäisen kertaluvun systeemiksi

$$\begin{cases} y_1' = y_2 \\ y_2' = -ay_2 - by_1 + 2t \end{cases} .$$

27. 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 $T = (t_0, t_0 + h, \dots, t_n)$, jonka jälkeen koetetaan etsiä vastaavia arvoja $y(t_0 + ih)$. Merkintöjen selvyuden 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 diskreetointi, ja h on t :n pisteiden välinen etäisyys.

28. Rungen-Kutan menetelmä differentiaaliyhtälöiden ratkaisemiseksi on huomattavasti kehittynyt versio Eulerin menetelmästä. Menetelmässä määritellään seuraavasti:

$$\begin{aligned}k_1 &= f(x_n, y_n), \\k_2 &= f(x_n + \frac{h}{2}, y_n + \frac{1}{2}hk_1), \\k_3 &= f(x_n + \frac{h}{2}, y_n + \frac{1}{2}hk_2), \\k_4 &= f(x_n + h, y_n + hk_3).\end{aligned}$$

Diskreetointiaskeleeksi 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:

29. 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 *kaottinen dynaaminen systeemi*, jonka käyttäytyminen voi muuttua merkittävästi jo pienistä muutoksista lähtötilanteesta; itse asiassa termi perhosvaikutus keksittiin kuvaamaan juuri tämän systeemin käytöstä.

Vihje: Kolmiulotteinen parametrisoitu käyrä (tai pistejoukko) piirretään MATLABissa funktiolla `plot3`.

30.

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.

31. Reuna-arvotettava tähtäysmenetelmällä (Moler odes teht. 7.19 ss. 45–47.) Olkoon ratkaistavana reuna-arvotettava $y'' = y^2 - 1$, $y(0) = 0$, $y(1) = 1$. 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.

32. 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.

33. 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 merkittävästi jo pienistä muutoksista lähtötilanteessa; itse asiassa termi perhosvaikutus keksittiin kuvaamaan juuri tämän systeemin käytöstä.

Vihje: Kolmiulotteinen parametrisoitu käyrä (tai pistejoukko) piirretään MATLABissa funktiolla `plot3`.

34. Ratkaise nk. *Rösslerin systeemi*

$$\begin{cases} y_1'(t) = -y_2(t) - y_3(t), \\ y_2'(t) = y_1(t) + ay_2(t), \\ y_3'(t) = b + y_3(t)(y_1(t) - c), \end{cases}$$

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.

35. Tässä tehtävässä mallinnetaan lyijyn kulkeutumista elimistöön ja sieltä pois. Ratkaistavaksi tulee epähomogeeninen differentiaaliyhtälöryhmä.

Taustaa. Lyijyä kulkeutuu ihmiselimistöön hengityksen, ruoan ja juoman kautta. Keuhkoista ja vatsalaukusta lyijy kulkeutuu nopeasti verenkierron mukana maksaan ja munuaisiin, ja imeytyy edelleen hitaasti muihin pehmytkudoksiin ja hyvin hitaasti luustoon. Se kulkeutuu elimistöstä pois pääasiassa virtsan mukana, mutta myös hikoilun, hiusten ja kynsien kautta (ks. kuva):

Mallinamme ihmiselimistöä eri *osastoina* (veri, pehmytkudokset, luusto). Funktio $x_i(t)$ ($i = 1, 2, 3$) merkitsee lyijypitoisuutta osastossa i hetkellä t . Pitoisuuden yksikkö on mikrogramma ja ajan vuorokausi. Oletamme, että **lyijyn virtausnopeus osastosta i osastoon j on suoraan verrannollinen pitoisuuteen $x_i(t)$ kertoimella $a_{ji} > 0$** . Ulkomaailmalle on oma osastonsa, jotta mallista saadaan mielekäs, mutta emme ole kiinnostuneita ulkomaailman lyijypitoisuuksista. Lisäksi oletamme, että **lyijyn virtaus ulkomaailmasta elimistöön tapahtuu vakionopeudella L** – tämä ilmenee yhtälöryhmän oikeana puolena kuten pakottava voima massa-jousi-systeemeissä eikä siis liity ulkomaailman osastoon.

1. Matriisimuotoilu. Edellisen sivun mallista saadaan epähomogeeninen differentiaaliyhtälöryhmä

$$\mathbf{x}'(t) = A\mathbf{x}(t) + \mathbf{b}, \tag{1}$$

missä

$$A = \begin{bmatrix} -(a_{01} + a_{21} + a_{31}) & a_{12} & a_{13} \\ a_{21} & -(a_{02} + a_{12}) & 0 \\ a_{31} & 0 & -a_{13} \end{bmatrix}$$

ja $\mathbf{b} = (L, 0, 0)$.

- Selitä edellisen sivun mallin perusteella miksi matriisi A ja vektori \mathbf{b} ovat kuten yllä.
- Miksi matriisi A on aina kääntyvä? (Ks. Matlab-tiedosto.)
- Etsi differentiaaliyhtälöryhmälle (1) erityisratkaisu $\mathbf{x}_e(t)$.

2. Käytännön esimerkki. Tutkijat mittasivat pitkän aikavälin lyijypitoisuuksia vapaaehtoisien, terveen kaupunkilaismiehen (Los Angeles) veressä, luustossa ja kudoksissa. Mittausten pohjalta saatiin $L = 49,3$ mikrogrammaa / päivä ja seuraava taulukko (yksikkönä 1/päivä):

$$\begin{array}{lll} a_{21} = 0,011, & a_{12} = 0,012 & \text{(verestä kudoksiin ja takaisin)} \\ a_{31} = 0,0039, & a_{13} = 0,000035 & \text{(verestä luihin ja takaisin)} \\ a_{01} = 0,021, & a_{02} = 0,016 & \text{(poistuma verestä ja kudoksista)} \end{array}$$

- Syötä ylläolevat arvot matriisiin A .
- Laske uudelleen kohdan 1 yksittäisratkaisu \mathbf{x}_e .
- Etsi homogeeniselle differentiaaliyhtälöryhmälle $\mathbf{x}'(t) = A\mathbf{x}(t)$ yleinen ratkaisu $\mathbf{x}_h(t)$.
- Mikä on epähomogeenisen differentiaaliyhtälöryhmän (1) yleinen ratkaisu?
- Ratkaise (1) alkuarvolla $\mathbf{x}(0) = 0$, ts. määritä kehon lyijypitoisuudet ajan funktiona tapauksessa, jossa henkilö on juuri muuttanut Los Angelesiin keho vailla lyijyä.

36. 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:

-e

mlDiffint

37. Muista, että funktion f derivaatta pisteessä x_0 määritellään seuraavasti:

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

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

Kokeile laskea funktion $f(x) = \sin(x)$ ja vertaa samaasi 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.

38.

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 quad-alkuiset Matlab-funktiot.

Luokittelu:

`mplteht/mplDiffint/mplxx.tex`, `matlabteht/mlDiffint/mlDixx.tex`

`mmateht/maDiffint/maDi100`

Avainsanat:

Symbolinen integrointi, numeerinen integrointi, funktiot, lausekkeet

39. `mmaDi104/mplDi11/mlDi11`

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

Vihje: `arcsin` on Mathematicassa `ArcSin`, Mapleissa `arcsin` ja Matlabissa `asin`.

Käytä symboliohjelmissa perinteistä "diffistekniikkaa" kuvan kanssa, Matlab:ssa raakaa "numeronmurskausta" tyyliin: `linspace`, `plot`, `zoom`, uusi `linspace` kapeammalla välillä, `find`, ...

-e

mlFunktiot

40. Jos f on analyyttinen, sille pätee Cauchyn integraalikaava, josta muuttujanvaihdolla saadaan

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

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 $f : U \rightarrow \mathbb{C}$ 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 maksimiperiaatteen 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:

41. On esitetty, että jatkuva funktio $f : [0, \infty) \rightarrow [0, \infty)$, 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:

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

$n = \dots, -2, -1, 0, 1, 2, \dots$ 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.

-e

mlGrafikka

42. 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 pitämällä vanhan kuvan `hold`-komennon avulla.

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

43. Piirrä

1. xe^{-x^2} välillä $[-2, 2]$
2. $1/(1+x^2)$ välillä $[-4, 4]$

Vihje: Muista pisteet laskutoimituksissa!

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

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

Vihje: Pane merkille tällaiset grafiikan ulkoasua säätelevät lisäkomennot (jotka voidaan antaa jälkikäteen (paitsi `hold on` pitää antaa ajoissa)):

`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).

45. Piirrä samaan kuvaan funktioiden $\sin kx$ kuvaajat välillä $[0, 2\pi]$, kun $k = 1 \dots 5$.

1. Tee nuolinäppäintä (\uparrow) komentoeditoinnissa hyödyntäen ja `hold on`- komentoa käyttäen.

2. Kirjoita pieni `for`-silmukka.

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

Vihje: Muodosta ensin 100×5 -matriisi, jonka sarakkeet ovat kx , $k = 1 \dots 5$. 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.

46. Matriisiin sovellettuna `plot`-funktio piirtää kunkin matriisin sarakkeen. Varsin käyttökelpoinen muoto on `plot(x,A)`, jossa x on A :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.

47. Piirrä samaan kuvaan potenssit x, x^2, \dots, 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:

```
% % Piirret\ "a\ "an potenssifunktioita.
% Tiedosto: potenssipiirto.m.
% Laatinut Vilja Varis 1.1.2012 % HUOM! ellet muuta tätä, saat 0 pistettä!
close all % Grafiikkaruudun tyhjennys
n=5;      % Muuteltava parametri
...
```

Talleta ja kirjoita komentoikkunaan:

```
>> potenssiipiirto
```

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

Vihje:

1. Tee `for`-silmukka ja käytä `hold on`-komentoa uuden kuvan piirtämiseksi vanhan kaveriksi.
2. 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.

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

48.

[3D-grafiikkaa, korkeuskäyriä] HA

Olkoon

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

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.tkk.fi/~apiola/matlab/opas/lyhyt/grafiikka.html#sec:3d>

Matlab-help: `doc mesh`, `doc surf`, `doc contour`

Ratkaisu: `./mlG07ratk.m`

`./html/mlG07ratk.html`

<http://math.tkk.fi/~apiola/matlab/opas/lyhyt/ratkaisuja/html/H3teht4.html>

49.

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) = ...
```

Ratkaisu:

```
>> x=linspace(0,4,200);
>> osoittaja=1+x/24; % Skalaarilla jaossa ei tarvia pistettä.
>> % (Jos skalaari jaettaisiin vektorilla, niin toki tarvittaisiin.)
>> nimittaja=1-x/12+x.^2/384;
>> f=(osoittaja./nimittaja).^8;
>> plot(x,f)
>> hold on
>> plot(x,exp(x),'r')
>> figure % uusi graikkaikkuna
>> plot(x,f-exp(x))
```

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

50. 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 $z_1, z_2, z_3 \dots$. 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 $n \rightarrow \infty$.) 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 500×500 hila realitasossa, luo sen avulla vastaava hila \mathbf{Z} kompleksitasolle, ja aja funktio `escapeVelocity` jokaiselle matriisin \mathbf{Z} alkioille.

Vihje: Realiakselin 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`.

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

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

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

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

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, ja päättäkumpaa kannattaa tässä tilanteessa käyttää.

52. Funktion $g(x)$ kiintopiste on piste x_0 , jolle pätee $g(x_0) = x_0$. Valistuneen arvauksen kiintopisteen sijainnista voi piirtämällä kuvaajat $y = g(x)$ ja $y = x$ samaan kuvaan, ja arvioimalla käyrien leikkauspistettä graafisesti. Käyttämällä tätä tekniikkaa, arvioi funktion $g(x) = \cos(x)$ kiintopisteen sijaintia.

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

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


53. Piirrä MATLABilla alla oleva kuva.

Vihje: Selityslaatikko luodaan komennolla `legend`, akselikuvaukset komennoilla `xlabel` ja `ylabel`.

54. 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 tiettyssä mielessä kompressoida tiputtamalla osia singulaariarvohajotelmasta pois; on todistettavissa että (MATLABilla ilmaistuna) $\mathbf{U}(:, 1:k) * \mathbf{S}(1:k, 1:k) * \mathbf{V}(:, 1:k)'$ on paras mahdollinen $\text{rank}(k)$ -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 \mathbf{P} on kuvasi matriiesitys. 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.

55. 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.

56. HT

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).

Vihje: Uusi grafiikkaikkuna: `figure`

Muistathan ympyrän luonnollisen parametriesityksen.

Ympyrän data koostuu oikeasti säännöllisen n -kulmion nurkkapisteistä, missä esim. $n = 100$ (`linspace:n` oletus). Ympyrän kuvan piirtäminen on siten sama homma kuin edellisissä lineaarikuvaustehtävissä.

57. Matlab ja Maple (tee molemmilla).

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

Piirrä pintakuva ja korkeuskäyräpiirros.

Ota alueeksi vaikka `[-2 2 -1 1]`.

Vihje: Tutustu samalla Matlabin `meshgrid:n` toimintaan.

Korkeusarvomatriisi Z tehdään kahden muuttujan funktiolle tähän tapaan:

```
>> x=linspace(a,b,m); y=linspace(c,d,n); % m ja n luokkaa 30.
>> [X,Y]=meshgrid(x,y);
>> Z=f(X,Y);
```

(Kokeile periaatetta pienillä, hiukan erikokoisilla matriiseilla X, Y .)

Tässä funktion f on toimittava pisteittäisin operaatioin. Jos vaikka $f(x, y) = x^2 - y^2$, kirjoitettaisiin:
`Z=X.^2 - Y.^2;`

Pintoihin `mesh(x,y,Z)`, `surf(x,y,Z)`, ... Kokeile myös `colorbar` yms.

Matlabilla korkeuskäyriin `contour`, voit myös kokeilla `ezcontour`-funktiota. Mahdollisuus on kokeilla myös korkeuskäyrien valitsemistapoja, `clabel`.

Älä diskretoi liian hienoksi. `Linspace`ssa 100 on ihan liikaa, n . luokkaa 30 olkoon lähtökohta.

Maple: Helpompaa, mutta tulos ei aivan niin loistava kuin Matlabissa. (Osin tosin varsin hienoa tämäkin, ja "context sensitive").

```
> with(plots):
> plot3d(f(x,y),x=a..b,y=c..d);
> contour(f(x,y),x=a..b,y=c..d); # Tarkista!
```

mlIntegraalimuunnos

58. 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))-0.5;
```

Suodata kohinaisesta signaalista `fh` alkuperäinen esille alkuperäinen signaali mahdollisimman tarkasti käyttämällä diskreettiä Fourier-muunnosta

Vihje: Signaalin taajuuskomponentit 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 käytännössä tehdä signaalille Fourier-muunnos (`fft`), ja piirtää taajuuskomponenttien magnitudit (`abs`) näkyville, ja päätellä, mitkä kuuluvat signaaliin, ja mitkä eivät. Tämän jälkeen käytä loogista indeksointia ja käänteistä Fourier-muunnosta (`ifft`) saadaksesi esille suodatettu signaali.

-e

mlKompleksianalyysi

59. mlK001.tex

Ensiapuhjeita

- 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!

Tämä ei siis ole tehtävä, alla vain tämän ohjeen Latex-koodi.

60. mlK002.tex

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

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

Opiskele aihetta "kompleksilukujen piirtäminen":

math.aalto.fi/~apiola/matlab/opus/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).

61. mlK003.tex

Olkoon z :lla edellisen tehtävän arvo. 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 Moivre)?

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: `.../opus/ei-niin-lyhyt/osa2.html#luku4` matriisi- ja taulukko-operaatioesimerkit ja `...opus/lyhyt/perusteet.html#sec:matriisilaskenta`

62. mlK004.tex

Muodosta vektori, jossa on luvut $w_{n,k} = \sqrt[n]{1}$, $k = 0 \dots n - 1$. 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 myöhempisiin tarpeisiin)
%
n=10    % Tätä voit vaihdella.
k=0:n-1 % kokonaislukuvektori.
w=...   % w-vektori, jossa nuo n:nnet juuret.
```

Skripti, jossa on Matlab-komentoja (ja selityksiä %-merkin takana), suoritetaan Matlabissa komentamalla `h1teht3` tai leikkaus/liimaus- menetelmällä. Suosittelemme jälkimmästä myös siksi, että lienee yksinkertaisinta sijoittaa tärkeimmät komentojen tulokset samaan tiedostoon, jolloin sitä ei enää voi suorittaa skriptinä (ellei halua nähdä virheilmoitussumaa).

63. mlK005.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) $0 \leq y \leq \pi/2$. (Mieti, kuuluuko 0 mukaan kuvaan.)

(c) $\pi < y \leq 3\pi$.

(d) $\ln 3 < x < \ln 5$.

64. mlK006.tex

Avaa

<http://www.math.hut.fi/teaching/k3/03/L/CA1.html>

ja sieltä kohta `demoexp`. Voit käyttää 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.

65. mlK007.tex

(Puhdas käsinlasku)

Kompleksiluvulla $e^{i\alpha}$ kertominen suorittaa kierron kulman α 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).

66. mlK008.tex

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 kissaparkkaa. Päättä ei tarvitse pyörittää, ympyrä ei pyöritetäessä miksiäkään 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 mieleemme ja jätämme seuraavaan kertaan.

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

67. mlK009.tex

Matlab on ennenkaikkea matriisikieli. Jos kompleksiaritmetiikka sujuu kätevästi, niin samoin on laita matriisilaskujen. Katsotaan siksi vielä, miten edellinen hoidettaisiin matriisioperaatioin. (Vrt. teht. 4) Näin saadaan malli myös yleisemmille lineaarikuvauksille.

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
z =

    2.0000 + 3.0000i

>> plot(z)
>> plot(z,'*')
>> axis([0 4 0 4])
>> hold on
>> plot([0 z])
>> abs(z)

ans =
    3.6056
>> angle(z)
ans =
    0.9828
>> atan(3/2)
```



```
ans =  
    0.9828
```

-e

mLlinis – Matlab-tehtäviä lineaarialgebrasta

68. mLli001.tex

Ohjetiedosto, poimi mukaan tehtäväpaperiin tarpeen mukaan.

Matlab-ohjeita

- Komennon suorittama tulos tulee ruudulle ENTER-painalluksen jälkeen (kuvat erilliseen ikkunaan). Jos haluat estää tulostuksen, pääätä 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 $u.^2$

Avainsanat:

Ohjetiedosto, Harjoitusohjeita, Lineaarialgebra, Matlabperusteet,

69. mlLi002.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 yhtälösystemi $Ax = b$ ja tarkista tulos matriisikertolaskulla.

Avainsanat:

Lineaarinen yhtälöryhmä, Matlabperusteet, Matlabalkeet, perusmatriisilaskenta

70. mlLi003.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": $A \setminus b$

- 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 määrittämiseksi lineaarinen yhtälösystemi, joka ratkaistaan Matlab:n takakenolla (\setminus).

- 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 testaillet sen kuntoon.

71. mlLi004.tex Matlab/Maple/Mathematica

Tarkastellaan yhtälösystemejä:

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

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 yhtälösystemin herkkyyttä pienille virheille sanotaan *häiriöalttiudeksi* (“*ill-conditioned*”). Laske kummankin matriirin häiriöalttius.

Suhteellisen virheen suurtenemisyhtälö:

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

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

Luokittelu:

mplteht/mplLinis/mplLixx.tex, matlabteht/mlLinis/mlLixx.tex

Avainsanat:

Numeerinen lineaarialgebra, matriisit, lineaariset yhtälöryhmät, häiriöalttius

72. mlLi005.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 \mathbf{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).

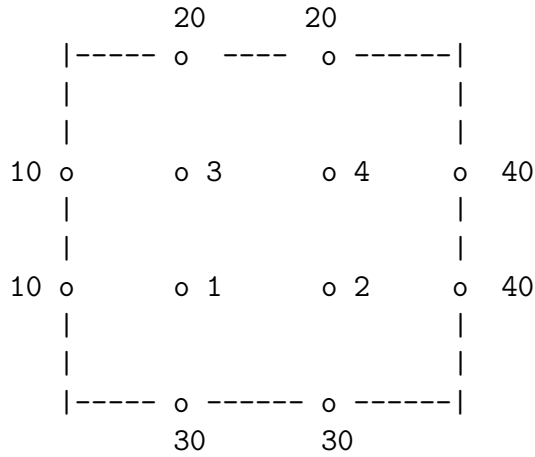
Opettajalle: Tehtävään voidaan lisätä myös A -matriisin muodostaminen diagonaaleittain **diag**-funktiolla (vaikkei mene hyödyn puolelle näin pienessä tehtävässä).

73. mlLi006.tex

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 4×4 - yhtälösystemi 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.



- (a) Ratkaise yhtälösystemi ja sijoita ratkaisulämpötilat ao. hilapisteisiin.
- (b) Muodosta 4×4 - matriisi, jossa on reunalämpötilat ja ratkaisemasi sisäpistelämpötilat sekä nurkissa lähinnä olevien kahden reunasolmun keskiarvot tähän tapaan:
 $U = [5 \ 20 \ 20 \ 30; 10 \ u_3 \ u_4 \ 40; 10 \ u_1 \ u_2 \ 40; 20 \ 30 \ 30 \ 35]$; Piirrä ratkaisupinnan approksimaatio: `mesh(U)` tai `surf(U)`.

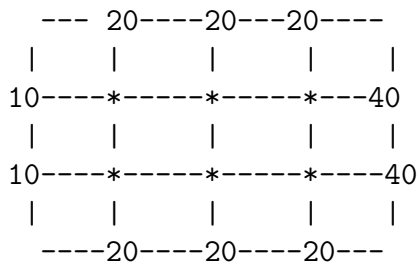
Opettajalle: Hilakokoa voidaan kasvattaa. Jos annetaan tehtäväksi ilman kerroinmatriisin struktuurihavaintoa, 4×4 lienee suurin ajateltavissa oleva (16×16 yhtälösystemi), joka sekkin saattaa jo hipoa joidenkin kärsivällisysrajaa.

Avainsanat: Lämpötilamatriisi, Laplacen yhtälön diskretointi, differenssimenetelmän alkeellisin perustehtävä, lineaarinen yhtälöryhmä.

74. mlLi006a.tex, [Maple: mplLinis/ mplLi010.tex]
 (Kynä-paperitehtävä)

Tarkastellaan lämmönjohtumista ohuessa metallilevyssä. Oletetaan, että johtumista tapahtuu vain levyn suunnassa, ja levyn reunoilla on annetut (ajan suhteen) vakioämpötilat. 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: (Klikkaa oikealla olevaa pdf-linkkiä, niin kuva näkyy kunnolla.)



Kuvassa näkyvät annetut vakio reunälämpötilat (reunaehdot). Tehtävänä on laskea ratkaisuaap-
proksimaatiot $*$:llä merkityissä sisäsolmupisteissä käyttäen seuraavaa periaatetta: Lämpötila
levyn solmupisteessä on naapurisolmujen lämpötilojen keskiarvo.

Jos indeksoidaan solmupisteiden lämpötilat vaakarivijärjestyksessä: T_1, \dots, T_6 , voidaan ryhtyä
kirjoittamaan yhtälöitä tyyliin:

$$T_1 = \frac{20+10+T_4+T_2}{4}, \dots$$

Kirjoita koko 6×6 - yhtälösystemi "standardimuodossa".

Huom: Tasapainotilaratkaisu saadaan ns. *Laplacen yhtälön* $\nabla^2 T = 0$ ratkaisuna. Tässä esitet-
tyyn likimääräismentetelyyn ns. *differenssimenetelmään*

Ratkaisua pyydetään seuraavassa tehtävässä.

75. mLi006b.tex, [Maple: mplLinis/ mplLi011.tex]

Ratkaise edellisen tehtävän yhtälösystemi Maplea (tai Matlabia) käyttäen. (Tässä Maple-
ohjeet.) Muodosta sitten edellisen tehtävän kuvan mukainen 4×5 matriisi, jossa on annetut
reunalämpötilat sekä lasketut sisälämpötilat oikeilla kohdillaan. Ota nurkkapisteiden lämpöti-
loiksi kahden naapurisolmun lämpötilojen keskiarvo. Piirrä kuva, pyörittele hiirellä.

Vihje: Maplevihje

(Matlabissa et tarvitse vihjettä, vaan teet suoraan todella "matlabmaisesti".)

Tehtävässä riittää käytellä LinearAlgebra-kirjaston funktiota `LinearSolve`.

Ratkaisuvektorin muokkaaminen matriisiksi onnistuu mukavasti, kun leikkaat/liimaat alla olevan funktiomää-
rityksen Maple-työarkillasi. (Suorita leikkaus pdf-tehtävätiedostosta.)

```
Reshape:=(vek,m,n)->Matrix(linalg[matrix](m,n,convert(vek,list)));
```

Funktio on tehty vastaamaan Matlabin funktion `reshape` käytöstä siinä tapauksessa, jossa vektori muutetaan
annetun kokoiseksi matriisiksi.

Lämpötilamatriisin rakentelu kannattaa hoidella (Matlabinomaiseen) tyyliin:

```
Tsisa:=Reshape(T,2,3); # vektorissa T on ratkaisulämpötilat.  
Tiso:=Matrix(4,5,0);  
vaaka:=<15|20|20|20|30>;  
pysty:=...;  
Tiso[2..3,2..4]:=Tsisa;  
...
```

Piirtäminen komennolla `matrixplot` (muista `with(plots):`)

```
matrixplot(Tiso,axes=boxed);
```

Pyörittele kuvaa hiirellä.

Huom: Sanomattakin on selvää, että tehtävä sopii erikoisen hyvin Matlabille. Tässä pikem-
minkin näytetään, että Maplen LinearAlgebra-työkaluilla voidaan matkia Matlab-työtapaa ja
päästä lähelle samaa käsittelymukavuutta.

Lisätehtävä: Tee ratkaisu Matlabilla!

Palataan asiaan perusteellisemmin Matlab-tehtävien yhteydessä, jolloin käsitellään lähemmin
differenssimenetelmää.

76. mlLi007.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.

Kuvan saat mukaan tehtävääsi, kunhan kopioit

polku/img/liik.eps-tiedoston omaan img-hakemistoosi. Tässä:

polku=<http://www.math.aalto.fi/opetus/Mattie/MattieT/matlabteht/mlLinis/>

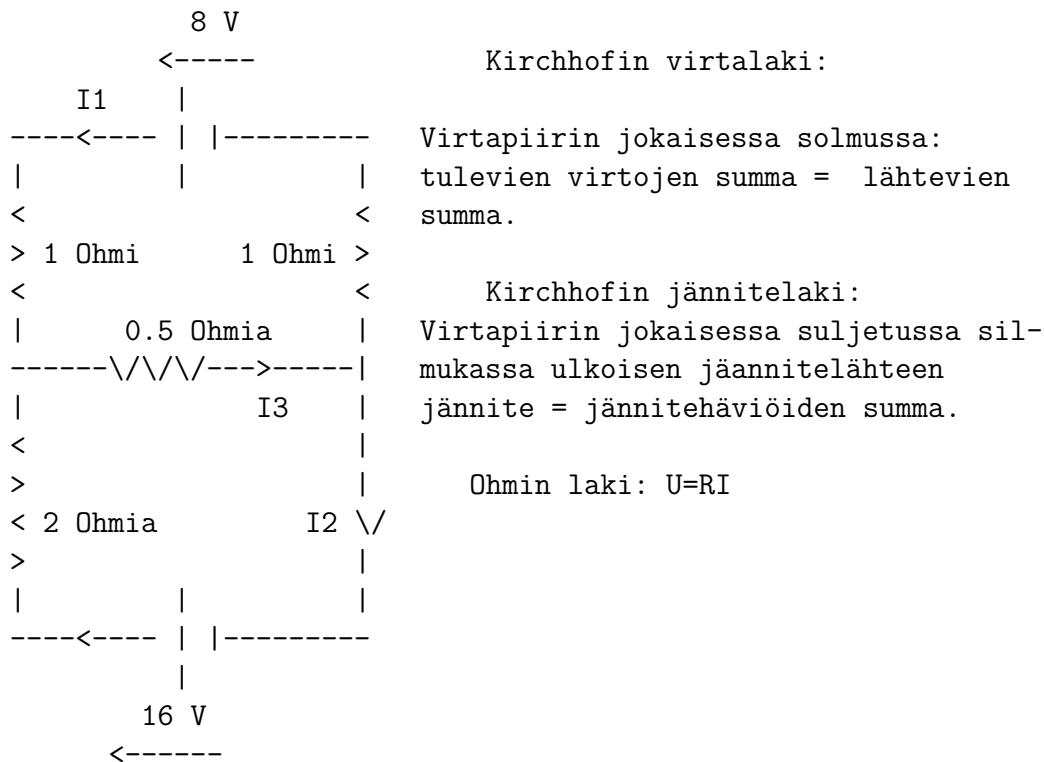
- (a) Muodosta yhtälösystemi tuntemattomien ajoneuvomäärien x_1, \dots, x_5 suhteen.
- (b) Määritä systeemin yleinen ratkaisu.
- (c) Jos x_4 :llä merkitty katuosuus suljetaan, niin mikä on yleinen ratkaisu?
- (d) Määritä kohdan (c) tilanteessa pienin x_1 :n ja suurin x_3 :n arvo (jotta yhdensuuntaisuutta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

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

Avainsanat: Liikenneverkko, lineaarinen yhtälöryhmä, (reduoitu)porrasmuoto, rref.

77. mlLi007a.tex

Muodosta lineaarinen yhtälösystemi alla olevalle virtapiirille ja ratkaise virrat.



Vast.: $I_1 = 2, I_2 = 6$

Avainsanat: Sähköpiiri, tasavirtapiiri, lineaarinen yhtälöryhmä, (reduoitu)porrasmuoto, rref.

78. mlLi008.tex

1. Muodosta 5×5 -yksikkömatriisi I . (**help eye**)
2. Muodosta matriisi E_1 , jossa on vaihdettu I :n rivit 2 ja 5.
3. Muodosta matriisi E_2 , joka saadaan kertomalla I :n 4. rivi luvulla 4.
4. Muodosta matriisi E_3 , joka saadaan I :stä *Gaussin rivioperaatiolla*:

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

missä r_i tarkoittaa matriisin riviä numero i .

5. Muodosta matriisit E_1^{-1} , E_2^{-1} , E_3^{-1} käyttäen komentoa `inv` ja selitä, mitä rivioperaatioita ne vastaavat.

Avainsanat: Alkeismatriisit, LU-hajotelma, Gaussin rivioperaatio, käänteismatriisi.

79. mlLi009.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

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

Muodosta matriisitulot E_0A , E_1A , AE_1 ja E_2A ja selvitä, mitä nämä operaatiot tekevät matriisin A riveille/sarakkeille.

Avainsanat: Alkeismatriisit, LU-hajotelma, Gaussin rivioperaatio

Vihje: Käsineläskä ja ajattelutehtävä, tarkistukseen voit hyödyntää Matlabin `syms`-komentoa tai voit tehdä symboliset matriisioperaatiot Maplalla/Mathematicalla.

80. mlLi010.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 rivioperaatioiden seurauksena.

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

Annettuna on 3×3 -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.

81. mlLi013.tex

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

$$\begin{cases} z_1 = r_1 \sin u \cos v, \\ z_2 = r_2 \sin u \sin v \\ z_3 = r_3 \cos u, \end{cases} \quad \begin{cases} 0 \leq u \leq \pi \\ 0 \leq v \leq 2\pi \end{cases}$$

MATLAB ilmoittaa 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`.

82. 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 yhtälösystemi niin, että suoritat laskut (järjestystä vaihtamatta) 3:lla merkittävällä 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.

83. mlLi015.tex

Olkoon $A = \begin{bmatrix} 1 & 8 & 6 \\ -1 & -4 & 5 \\ 2 & 4 & -6 \end{bmatrix}$ ja $b = [8 \ 1 \ 4]^T$. Ratkaise yhtälö $Ax = b$ osittaistuenta

(“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 tukiakion valitsemista pienen tukiakion aiheuttamien numeeristen ongelmien välttämiseksi. Matlab saattaa käyttää esim. ns. skaalattua osittaistuenta, jolloin rivinvaihtostrategia voi olla erilainen.

84. 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 yhtälösystemin 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:

85. 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 $Xc = y$.

(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(kertoimet,x); 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.

86. mlLi019.tex

Määritä PNS-ratkaisu tehtävälle $Ax = b$, 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.

87. mlLi020.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 ominaisvaruuden 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.

Avainsanat: Ominaisarvot, ominaisvektorit, ominaisarvojen graafinen havainnollistaminen, ominaisvektorien graafinen havainnollistaminen, Matlab: eigshow, eig

88. mlLi020a.tex

Ohjeita, ominaisarvo-oppia (Liitettäväksi aiheen tehtäväpaperiin)

- **Ominaisarvo** on luku, se voi olla kompleksiluku, vaikka matriisi olisi reaallinen.
- **Ominaisvektori** on (reaalisen matriisin tapauksessa) \mathbb{R}^n :n tai \mathbb{C}^n :n vektori sen mukaan, onko vastaava ominaisarvo reaallinen vai kompleksinen.
- Ominaisarvo saa aivan mainiosti olla 0, ominaisvektoriksi emme hyväksy nollavektoria.
- Ominaisarvoon λ liittyvä **ominaisavaruus** E_λ koostuu kaikista λ :aan liittyvistä ominaisvektoreista ja lisäksi nollavektorista. Tällöin kyseessä on vektori(al)avaruus, nimittäin matriisin $A - \lambda I$ nolla-avaruus, $N(A - \lambda I)$.
- Ominaisarvon λ_j **algebraalinen kertaluku** M_{λ_j} on karakteristisen polynomin $\det(A - \lambda I)$ juuren kertaluku. **Geometrisen kertaluku** m_{λ_j} on $\dim(E_{\lambda_j})$.
Pätee: $m_{\lambda_j} \leq M_{\lambda_j}$
- Jos reaalilla matriisilla A on **kompleksinen** ominaisarvo $\lambda = \alpha + i\beta$, niin myös $\bar{\lambda} = \alpha - i\beta$ on A :n ominaisarvo. Jos \mathbf{v} on λ :aa vastaava ominaisvektori, niin liittolukua $\bar{\lambda}$ vastaava ominaisvektori on $\bar{\mathbf{v}}$. (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). Tässä on kyse nolla-avaruuden kannan määräämistehävistä.
- 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)`.

Avainsanat: Ominaisarvo-opin perus(lasku)ohjeita

89. mlLi021.tex (käsinlasku, Matlab sopii avuksi/opiksi)

Muodosta matriisiin $A = \begin{bmatrix} 3 & -2 & 4 \\ -2 & 6 & 2 \\ 4 & 2 & 3 \end{bmatrix}$ 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 ominaisarvuuteen 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 $(v_1 + c v_2) \perp v_1$.

90. 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:

91. mLi029.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 nolasta 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.

92. mLi030.tex

Ominaisarvojen laskentamenetelmiä, Power method [KRE⁹] 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.

Ratkaisu: Vastaus: $q = 4, 4.493, 4.4999$; $|\epsilon| \leq 1.5, 0.1849, 0.0206$

Avainsanat: Potenssimenetelmä, ominaisarvojen laskentamenetelmät, Power method.

93. mLi031.tex

Osoita, että jos x on ominaisvektori, niin $\delta = 0$ virhekaavassa (2) Theorem 1 s. 872 (KRE⁹, luvun 20.8, Power Method for Eigenvalues alkusivulla) .

Vihje: Päätelytehtävä, ohjelmistoista ei hyötyä.

Avainsanat: Potenssimenetelmä, ominaisarvojen laskentamenetelmät, Power method.

94. mLi032.tex

(a) Päättele Gershgorinin lauseen avulla matriisiin

$$A = \begin{bmatrix} 11 & 0.4 & -0.5 \\ 0.4 & 7 & a \\ -0.5 & a & 4 \end{bmatrix}$$

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 varmistukseksi Gershgorinin pätevyydelle.)

Vihje: Käsineläskä, jossa voit harjoitella Matlabin laskinkäyttöä.

Avainsanat: Gershgorinin lause, ominaisarvojen laskentamenetelmät, ominaisarvoarvio.

95. 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 λ_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 $\mathbf{A} = 10 \cdot \text{randn}(12) + 5 \cdot \text{randn}(12) \cdot \mathbf{i}$;

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.');
```

96. mlLi050.tex

Gram-Schmidtin menetelmä vektorijoukon $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$ ortonormalisoimiseksi toimii seuraavasti:

- Ortogonalisoidaan:

$$\mathbf{u}_1 = \mathbf{v}_1$$

$$\mathbf{u}_2 = \mathbf{v}_2 - \frac{\mathbf{u}_1 \cdot \mathbf{v}_2}{\mathbf{u}_1 \cdot \mathbf{u}_1} \mathbf{u}_1$$

⋮

$$\mathbf{u}_n = \mathbf{v}_n - \sum_{k=1}^{n-1} \frac{\mathbf{u}_k \cdot \mathbf{v}_n}{\mathbf{u}_k \cdot \mathbf{u}_k} \mathbf{u}_k$$

- Normitetaan: $\mathbf{e}_i = \frac{\mathbf{u}_i}{\|\mathbf{u}_i\|}, i = 1 \dots n$

Kirjoita MATLAB-funktio $\mathbf{B} = \text{grmsch}(\mathbf{A})$ joka hakee Gram-Schmidtin menetelmällä ortonormaalin kannan matriisiin \mathbf{A} sarakeavaruudelle. Testaa ortonormaalius laskemalla $\mathbf{B}' \cdot \mathbf{B}$.

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.

97. mlLi090.tex

Seuraava kuva esittää kymmenen sivun ”internetiä”.

Laske tämän verkon tärkein sivu käyttämällä PageRank-algoritmia:

- Luo verkon vierusmatriisi $A = [a_{ij}]$, missä

$$a_{ij} = \begin{cases} 1 & \text{jos sivulta } j \text{ on linkki sivulle } i \\ 0 & \text{muuten} \end{cases}$$

- Laske vierusmatriisin suurin ominaisarvo, ja vastaava ominaisvektori.
- Normalisoi laskemasi suurinta ominaisarvoa vastaava ominaisvektori (jaa kaikki vektori alkioit vektorin summalla). Mikä on tämän verkon tärkein sivu.
- Piirrä verkon kuva käyttäen laatimaasi vierusmatriisia ja `gplot`-komentoa. Tutustu `gplot`in help-sivuun.

-e

mlMatriisit

98. 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 nolasta 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)` isoin ominaisarvo ja -vektori. Testaa tuloksen oikeellisuus.

Vihje:

99. 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

$$\mathbf{A} = \begin{bmatrix} 0 & 3 & 12 \\ 3 & 12 & 1 \\ -1 & -3 & 4 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 29 \\ 21 \\ 45 \end{bmatrix}.$$

Kokeile sitten ratkaista ongelma $\mathbf{A}\mathbf{x} = \mathbf{b}$, kun \mathbf{A} on 18×18 Hilbertin matriisi, ja $\mathbf{b} = \mathbf{A}[1]_{18}$,

Vihje:

Hilbertin matriisi MATLABina:

```
A = hilb(18);
b = A*ones(18,1);
```


100. Gram-Schmidtin menetelmä vektorijoukon $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$ ortonormalisoimiseksi toimii seuraavasti:

- Ortogonalisoidaan:

$$\mathbf{u}_1 = \mathbf{v}_1$$

$$\mathbf{u}_2 = \mathbf{v}_2 - \frac{\mathbf{u}_1 \cdot \mathbf{v}_2}{\mathbf{u}_1 \cdot \mathbf{u}_1} \mathbf{u}_1$$

⋮

$$\mathbf{u}_n = \mathbf{v}_n - \sum_{k=1}^{n-1} \frac{\mathbf{u}_k \cdot \mathbf{v}_n}{\mathbf{u}_k \cdot \mathbf{u}_k} \mathbf{u}_k$$

- Normitetaan: $\mathbf{e}_i = \frac{\mathbf{u}_i}{\|\mathbf{u}_i\|}, i = 1 \dots n$

Kirjoita MATLAB-funktio $\mathbf{B} = \text{grmsch}(\mathbf{A})$ joka hakee Gram-Schmidtin menetelmällä ortonormaalin kannan matriisiin \mathbf{A} sarakeavaruudelle. Testaa ortonormaalius laskemalla $\mathbf{B}' * \mathbf{B}$.

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:

101. Matriisi $(a_{ij}) = \mathbf{A} \in \mathbb{R}^{n \times m}$ on *yläkolmiomatriisi*, jos $a_{ij} = 0$ kun $i > j$.

(a) Kirjoita MATLAB funktio, joka ratkaisee yhtälöryhmän $\mathbf{A}\mathbf{x} = \mathbf{b}$, 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 matriiseja 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)`.

102.

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

$$\lambda_1 \mathbf{u}_1 \mathbf{u}_1^T + \lambda_2 \mathbf{u}_2 \mathbf{u}_2^T + \lambda_3 \mathbf{u}_3 \mathbf{u}_3^T = \mathbf{A}$$

Vihje:

-e

mNonlineu, Epälineaariset yhtälöt

103.

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 määrittellä tähän tapaan, kun p on kerroinvektori:

$$pf=@(x) polyval(p,x)$$

Derivaatan saat polyder- (tai omapolyder)-funktioilla.

3. Yritä löytää kompleksijuuri antamalla kompleksisia alkuarvoja. (Jos löydät yhden, niin toinen on sen liittoluku.)
4. Määritä juuret roots-funktion avulla.

Vihje:

Ratkaisu: M1N101ratk.m[Tulee]

104. Määritä funktion

$$f(x) = 12(x - 1) \sin \frac{x}{x^2 + 0.4x + 0.1} \quad x \in [-4, 4]$$

kaikki nollakohdat (33 kpl).

105.

[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: M1N102ratk.m[Tulee]

106.

[NCM 4.15, p. 138]

Vesiputken syvyys, jotta ei jäädy.

Vihje:

Ratkaisu: M1N1xxratk.m[Tulee]

107. 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 päätepistettä 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 päätepistettä 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.

108. 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

$$g(x) = \cos\left(\frac{1}{2}\sqrt{x}\right)^2 \sqrt{x^3\pi}$$

välillä $[0, 5]$ sijaitseva nollakohta.

Vihje: Tälle funktiolle ja tälle menetelmälle on mahdollista löytää alkuarvo, jolla menetelmä ei toimi: kokeile siis useaa arvausta.

109. Matlab/Maple/Mathematica
H2T17/mlN1100/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: Polynomien juuret, numeriikka, häiriöalttius, ill-conditioned

110. 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.

111. Maple,Matlab,[Mathematica] (H2T8)

Newtonin menetelmän askel voidaan määritellä vähäeleisesti Maplelle. Määritellään iterointifunktio:

$$> N := x \rightarrow \text{evalf}(x - f(x)/D(f)(x));$$

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 kätevinä lienee käyttää Symbolic Toolboxia symboliseen derivointiin, jos se on käytettävissä.

Avainsanat: Epälineaarinen yhtälö, Newtonin menetelmä, iteraatio

112. Maple, Matlab (H2T9)

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 λ Käytä tätä λ :n arvoa ennustamaan populaation koko toisen vuoden lopussa, kun oletetaan maahanmuuttovauhdin säilyvän vakiona.

Vihje: Maple: `fsolve`, Matlab: `fzero`

Avainsanat: Epälineaarinen yhtälö, väestönkasvumalli, epälineaarinen yhtälö, väestönkasvumalli.

113. Newtonin menetelmä lienee kaikista funktion juuren etsimiseen käytetyistä menetelmistä kuuluisin. Toisin kuin aikaisemmin esittelemämme menetelmät, se ei edellytä tietoa juuren sijainnista, mutta toisaalta se ei aina välttämättä suppene kohti juurta.

Newtonin iteraatioilla tarkasteltavilta funktioilta odotetaan jatkuvuutta ja derivoituvuutta. Newtonin iteraatiokaava on

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Käyttäen aikaisemmin kirjoittamaasi numeerista derivointikaavaa, ja 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.

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.

114. 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 edellämämainitulla välillä.

1. $z \mapsto \log(z)$

2. $z \mapsto z^2$
3. $z \mapsto z + 1/z$

Vihje:

115. Määritellään

$$S_n(x) = \frac{nx}{1 + n^2x^2}$$

Näytä että jokaiselle kiinnitettylle arvolle x luku $S_n(x)$ lähestyy nollaa, kun $n \rightarrow \infty$, 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)`.

-e

Osittaisdifferentiaaliyhtälöt

116. mlPDE018.tex

Määritä sivuiltaan eristetyn sauvan $L = 10$ lämpötila $t = 2$ sekunnin kuluttua alkuhetkestä käyttämällä eksplisiittistä menetelmää numeeriseen approksimointiin. Valitse $h = 1$ ja $k = 0.5$. Alkulämpötila: $f(x) = x - 0.1x^2$ ja sauvan päät pidetään "jäissä". Tehtävä on siis:

$$u_t = u_{xx} \quad (\text{Olkoon lämpöyhtälön vakio} = 1.)$$

$$u(0, t) = 0, u(L, t) = 0 \quad (\text{Reunaehdot})$$

$$u(x, 0) = f(x) \quad (\text{Alkuehto}).$$

Piirrä alkulämpötilafunktion $f(x)$ ja ratkaisufunktiota $u(x, 2)$ approksimoivan lämpötilamurtoviivan kuvaaja.

Vihje:

117. mlPDE020.tex

Neliöalueessa diskretoidun Laplacen yhtälön matriisi on muotoa:

$$A = \begin{bmatrix} B & I & O & \dots & O \\ I & B & I & O & \dots \\ O & \ddots & \ddots & \ddots & O \\ \vdots & & & & \\ O & \dots & I & B & I \\ O & \dots & \dots & I & B \end{bmatrix}, \text{ missä } B = \begin{bmatrix} -4 & 1 & 0 & \dots & 0 \\ 1 & -4 & 1 & 0 & \dots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & & & \\ 0 & \dots & 1 & -4 & 1 \\ 0 & \dots & \dots & 1 & -4 \end{bmatrix}$$

ja I on yksikkömatriisi sekä O nollamatriisi.

Jos solmuja on reunat mukaan lukien $n \times n$ kpl., niin sisäsolmuja on $(n-2)^2$ kpl. Tällöin B, I, O ovat $(n-2) \times (n-2)$ ja A on $(n-2)^2 \times (n-2)^2$ matriisi. A koostuu siis kolmesta lohkonauhasta, diagonaalilohkoina B ja ylä- ja alanauhahlohkoina I .

Tällaisen matriisin muodostamiseen on monta tapaa. Kaikkein helpoin on valmiiksi ohjelmoitu MATLAB-funktio `delsq`. Koska matriisit ovat yleensä suuria, on syytä käsitellä niitä *harvoina*

(engl. *sparse*), ts. nolliä ei talleteta, ainoastaan nollostä poikkeavat indekseineen. Niinpä `delsq` rakentaa matriisista automaattisesti harvan. Jos haluat katsoa sitä tai sen osaa täytenä, käytä komentoa `full`. Kokeile:

```
help delsq
help numgrid
G=numgrid('S',10) % 'S' viittaa alueeseen "Square". Muitakin
A=delsq(G)         % on Matlabin numgrid-repertuaarissa.
full(A)
ans(1:10,1:10)
```

(Yhtä helppoa nelioalueen tapauksessa on käyttää omatekoista `lapm`-funktiota `matlab`-hakemistossa.)

Lopussa muita tapoja:

Kokeile näitä komentoja muutellen sopivasti parametreja. Suorita myös komento `help sparsfun`, saadaksesi katsauksen tärkeään osaan `MATLAB`:ia: harvojen matriisien käsittely-funktioihin.

Ratkaise tehtävä AV 3 käyttäen 50×50 -hilaa reunat mukaan lukien, siis $48^2 \times 48^2$ -matriisia A . Tulosvektori kannattaa muotoilla 48×48 -matriisiksi komennolla `U=reshape(u,48,48)`, jolloin sen voi heti visualisoida komennolla `surf(U)`. Teepä se.

Esitys taululla: Kerro kokeiluistasi ja havainnoista, mitä opit harvoista matriiseista ja miten lopun suorit ja mitä tunnelmia kuva herätti. (Jos joku kohta, kuten kuva ei onnistunut, voit ihmetellä ääneen, miksi.)

Vihje:

-e

matlabteht/mlPerusteet, Matlab-perusteita

118. mlP001.tex

Matlab-pikaohje

1. Komennon suorittama tulos tulee ruudulle ENTER-painalluksen jälkeen (kuvat erilliseen ikkunaan). Jos haluat estää tulostuksen, päättää 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.
2. Edellisen komennon tulos on muuttujassa `ans`. Yleensä on suositeltavaa antaa tulokselle oma nimi tyyliin `nimi= ...`.
3. Nuoliylös-näppäimellä (\uparrow) voi selata aikaisempia komentoja. Käytä ahkerasti komentoja `help`, `doc`.

4. `format long` : Tulostetaan enemmän numeroita (n. 16). Laskutarkkuuteen tämä ei vaikuta.
`format rational` laskee rationaaliluvuilla.
`format short`: Paluu oletustulostukseen.
5. Matriisi saadaan aikaan tyyliin: $A = \begin{bmatrix} 2 & 4 & 3 \\ 0 & 1 & -1 \\ 3 & 5 & 7 \end{bmatrix}$. Vektori saadaan näin: $v = [1 \ 2 \ 3]$. Pystyvektorissa käytetään erottimena puolipistettä (tietysti, vrt. matriisi A yllä). Matriisikertolaskun merkki on `*` (Edellistä virkettä ei voi päättää pisteeseen!)
6. Matriisin A transpoosi: A' (reaalisessa tapauksessa).
7. Kokonaislukuvektori: Esim `1:10` tai `1:2:20`. Myös `linspace`. Pystyvektoriksi transponoimalla.
8. $A(i, j)$ 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:
 $A(1:4, 1:4) = \text{ones}(4, 4)$ tai
 $A(2, :) = A(2, :) - 2 * A(:, 1)$ (Gaussin rivioperaatio).
9. Matriisien liittäminen: Jos A:lla ja B:llä on yhtä monta riviä, ne voidaan liittää peräkkäin:
 $\begin{bmatrix} A & b \end{bmatrix}$
(tai $\begin{bmatrix} A & b \end{bmatrix}$). Jos yhtä monta saraketta, niin allekkain: $\begin{bmatrix} A \\ B \end{bmatrix}$
10. Laskutoimitukset tarkoittavat matriisilaskua. Siis esim.
 $A * B$, A^p (jälkimmäinen mahdollinen vain neliömatriisille)
11. 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: $u.(2 * \text{ones}(\text{size}(u)))$, joka tietysti toimii.)
12. Piirtämistä varten muodostetaan x -vektori, joka edustaa diskretoitua x -akselia ja lasketaan a.o. funktion arvo vektoriin y .
Piirto: `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 \wedge on varattu matriisilaskutoimituksille, on "pisteittäin" operoitaessa lisättävä piste (.) a.o. 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`

13. 3d-piirto: Pintojen ja korkeuskäyrien piirtämiseksi tarvitaan korkeusarvojen matriisi xy -tason pistehilan päällä. Se aikaansaadaan helpoimmin (ja rutiininomaisesti) `meshgrid`-komentilla. 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 (myös surf1, surfc, colorbar,...)
>> contour(x,y,Z) % Korkeusk. piirros
```

Avainsanat: Matlab perusteet, harjoitus-pikaohje

119. mlP001a.tex

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)
```

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

120. mlP002.tex

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

121. mlP003.tex

Olkoon $x=[2 \ 5 \ 1 \ 6 \ 7 \ 4 \ 3 \ 2 \ 1 \ 11]$.

1. Lisää jokaiseen alkioon luku 12
2. Lisää 3 parittomien indeksien osoittamiin alkioihin.
3. Laske vektorin alkoiden neliöjuuri.
4. Laske vektorin alkoiden neliöt ja neliösumma.

Vihje: Helppo tapa vektorin v indeksivektorin muodostamiseen:

```
ind = 1:length(v)
```

Miten siis parittomat indeksit?

Summaus sujuu helposti: `help sum`

Avainsanat: Matlab-alkeet, vektorien muodostus,vektorioperaatiot, indeksointi, kaksoispiste(:)

.

122. mlP004.tex

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

1. Lisää vektorin x alkioden summa vektoriin y
2. Korota vektorin x alkioit vektorin y vastinalkioiden osoittamiin potensseihin.
3. Jaa y :n jokainen alkio vektorin x vastinalkiolla.
4. Kerro x :n jokainen alkio y :n vastaavalla alkioilla 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)

123. mlP005.tex

Muodosta vektori x , joka koostuu alkioista:

1. $2, 4, 6, 8, \dots, 20$
2. $10, 8, 6, 4, 2, 0, -2, -4, \dots, -10$
3. $1, 1/2, 1/3, 1/4, 1/5, \dots, 1/10$
4. $0, 1/2, 1/3, 1/4, 1/5, \dots, 1/10$

Vihje: Kahdessa viimeisessä kohdassa voit selkeyttää komentamalla:

```
format rational
```

Paluu oletusformaattiin: `format`

124. mlP006.tex

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)
help norm
```

Vihje:

125. mlP007.tex
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 lohkonan tyyliin:

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

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

Avainsanat: Matlab perusteet, Matriisikertolasku, taulukko-operaatiot.

126. mlP008.tex

Edellisen tehtävän lyhennetty versio.

Määrittele matriisit

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

Kokeile ja selitä:

$$A*B \quad B*A \quad A^2 \quad A.^2 \quad B^2 \quad B.^2$$

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.

Avainsanat: Matlab perusteet, Matriisikertolasku, taulukko-operaatiot.

127. mlP009.tex

Vrt. opas:

<http://math.tkk.fi/~apiola/matlab/opas/mini/vektgraf.html#alkulukuskripti>

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

Aloita tiedosto näin:

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

Apu: help primes (tai doc primes)
help sum .

Myöhemmin: opitaan tekemään funktio-m-tiedostoja, jolloin N voidaan antaa parametrina omalle "alkulukuja-funktiolle.

Vihje:

Avainsanat: mlPerusteet, Matlab perusteet, skripti, m-tiedosto, alkuluvut, sum

128. mlP010.tex

1. Miten kääntäisit vektorin v alkiot vastakkaiseen järjestykseen kaksoispisteen $(:)$ avulla?
2. Entä matriisin A sarakkeet, vastaavasti rivit?
Huom: Näihin on myös valmiit funktiot: `fliplr`, `flipud` "LeftRight, UpDown"
3. Miten limität ("merge") kaksi samanpituista vektoria u ja v ?
Tarkoitus on siis muodostaa vektori $w = [u_1, v_1, u_2, v_2, \dots]$

Vihje: Liitä vektorit allekkain ja jonouta näin saatu 2-rivinen matriisi sarakkeittain (ovelaa). (Sarakkeittain jonoutus matriisille A saadaan näin: `A(:)`.)

Avainsanat: mlPerusteet, Matlab perusteet, `fliplr`, `flipud`, kaksoispiste $(:)$, käänteinen järjestys, kaanteinen järjestys, matriisin jonoutus, "merge", limitys

129. Matriisin kokoaminen osista, lohkomatriisit, skriptit

Nyt jo viimeistään on syytä ottaa skriptit käyttöön. Kts. skriptiohjetta (laitetaan myös tähän) (myös `help script`) (Itse asiassa skriptillä kannattaa aloittaa koko Matlab harrastus.)

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 muuttaa.

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

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. Totuttele tähän, ohjeita on ... (tähän viittauksia).

Avainsanat: mlPerusteet, matlabperusteet, Lohkomatriisit, skriptit, m-tiedostot, `diag`.

130. mlP012.tex

Taikaneliön saa komennolla `magic(n)`. Muodosta muutamalla pienehköllä n :n arvolla matriisin $M = \text{magic}(n)$ 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.

Avainsanat: mlPerusteet, matlabperusteet, taikanelio, magic, rivisummat, sarakesummat,diag

131. mlP013.tex

Tässä vähän keskeneräistä muotoilua, mutta sopii “taikateemaan”. Tarkennellaan, kun ehditään ...

Taikaneliöillä voisi demonstroida Markovin prosesseja, ehkä jätetään ominaisarvolaskujen yhteyteen. (vrt. <http://math.tkk.fi/~apiola/matlab/opas/lyhyt/tehtavia1.html> teht. 6)

Tässä on joku, liittyy johonkin...

<http://math.tkk.fi/opetus/v/matlab/opas/osa2.html#luku251>

Matlab-opas/elo/touko2010-sivuilla tarkempi tehtäväseloste.

Vihje:

Ratkaisu:

Avainsanat: Markovin matriisit, prosessit, ominaisarvot.

Liittyy: Lineaarialgebra/ominaisarvot. Aiemmissä kurssimatskuissa, myös Solmu-kirjoituksessa, koetehtävissä on monia tehtäviä ratkaisuihin.

132. mlP014.tex, mplP014.tex

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.

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

Vihje:

a)

Maple

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

Matlab:

```
>> x=...  
>> f=...  
>> plot
```

b)

Maple

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

Matlab

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

Ratkaisu: Ratkaisu:

```
mplPerusteet/mplP014R.mw ja .pdf  
mlPerusteet/mlP014R.m ja .pdf
```

Luokittelu:

mplteht/mplPerusteet/mplP014.tex, matlabteht/mlPerusteet/mlP014.tex

Avainsanat:

Mapleperusteet, funktiot, lausekkeet, Matlabperusteet

133. Rekursiokaava

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

suppenee kohti lukua \sqrt{a} . Kirjoita MATLAB-skripti, jolla voit tarkastella tätä suppenemista, kun $a = 5$. Tee tulostus muodossa

n	$x(n)$	Error
0	1	
\vdots	\vdots	
6	...	

Vihje: Suppenemisen tutkimisessa kannattaa käyttää `for`-luoppia. Yksinkertaisimmillaan se toimii syntaksilla `for k = alkuarvo:loppuarvo`, jolloin muuttuja `k` saa arvokseen jokaisen kokonaisluvun välillä `[alkuarvo,loppuarvo]`. Tulostamiseen kannattaa käyttää funktiota `disp`.

134. mlP016.tex

Avaa Matlabin FILE-valikosta uusi m-tiedosto ja valitse "skripti". Talleta nimelle `cosinplot.m`.

Kirjoita/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: `cosinplot`

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.

Ratkaisu: Ei tarvita.

Avainsanat: Skripti, komentotiedosto, kuva, piirto

135. mlP017.tex

Vahvista numeerisesti uskoasi matemaattiseen totuuteen 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 π kun $n \rightarrow \infty$.

Vihje: a) Käytä `for`-luuppia, jossa kasvatat iteraatioiden ylärajaa suureksi. Suppeneminen on suhteellisen nopeaa, joten ylärajan ei tarvitse olla kovin iso.

b) Voit suorittaa tehtävän myös (ja mieluiten) vektoroidusti muodostamalla jonot kaksoispiste (`:`) - operaattorilla, aritmetiikan pisteittäin (`.`) ja soveltamalla `sum`-funktioita tai vielä paremmin `cumsum`:ia, jolla saat koko osasummien jonon.

Kirjoita skriptiksi, jossa voit vaihdella parametria `n`, tottakai!

136. mlP018.tex

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 θ on 45°

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 .

- 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.
- 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:

137. mlP019.tex

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 $y = x$.

Ratkaisu: mlP019R.m

(pdf puuttuu, tuskin tarpeen)

Avainsanat: Matlabperusteet, ramppifunktio, paloittain määrittely, `zeros`, `max`, “vielä elegantimpi”

138. mlP020.tex

(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 $1, 2, \dots, 1000$ ja sitten vain `sum`. Jälkimmäisessä voit laskea muutamalla, toinen toistaan suuremmalla arvolla. (Numeerisesti et tietenkään voi summata äärettömyyksiin.)

139. mlP021.tex

Esitä yhden rivin Matlab-komento, jolla saat selville vektorin tai matriisin niiden alkioden 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)`

-e

matlabteht/mlTodari, Todennäköisyys, satunnaisuus, tilasto

140. mlT004.tex

Laskemme yksikkökolmion T (virittävät pisteet $(0, 0)$, $(1, 0)$, $(0, 1)$) pinta-alan tasaisesti jakautuneilla satunnaisluvulla Monte-Carlo menetelmää mukailleen:

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 kirjoittaa tarkistuksen silmukkaan, ja tehdä päätöksen kontrollirakenteilla. 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`, joka 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.

Pisteitä piirretään `plot`-komennolla optioita hyväksikäyttäen: esimerkiksi `plot([3 2],[4 1],'.r')` piirtää pisteet $(3, 4)$, $(2, 1)$ punaisina pisteinä.

Avainsanat: mlTodari, mlPerusteet, matlabperusteet, Monte Carlo, looginen ineksinti, satunnaisluvut

141. mlT005.tex

(Osa kaavoista epäselviä html:ssä, katso pdf-tehtäviä!)

Monte Carlo-approksimaatio π :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

$-1 < x < 1, 1 < y < 1$, niin ympyrään ja neliöön osuneiden tikkojen lukumäärän suhde lähenee lukua $\pi/4$, kun satunnaisheittojen lukumäärä kasvaa. Miksi? Generoi tasajakautuneita pistepareja ja laske ko. osuus.

Alla on vihjettä pikku esitystä varten. Toisaalta tehtävä ei kaipaa mitään lisäopiskelua, tai vihjeitä. Kenties ehto $X.^2+Y.^2 \leq 1$ ja bittivektorin ykkösten lukumäärän laskeminen vähemmän Matlabia osaaville. (Vrt. edellinen kolmiotehtävä mlT004.tex.)

Vihje: Kirjassa C.vanL on hyvä tiivis selvitys aiheesta "Random processes"1.3.2 ss. 34 - 37. Tästä aiheesta voisi tehdä pienen harjoitustyön.

Esiintyviä Matlab-funktioita:

```
http://www.mathworks.com/access/helpdesk/help/techdoc/ref/hist.shtml
http://www.mathworks.com/access/helpdesk/help/techdoc/ref/rand.shtml
http://www.mathworks.com/access/helpdesk/help/techdoc/ref/randn.shtml.
```

Satunnaisprosesseihin ja tähän tehtävään (Monte Carlo simulaatio π :n laskemiseksi) on CV-sivulla selkeät skriptit:

<http://www.cs.cornell.edu/cv/Books/SCMV/Mfiles/chap1.htm#Dice>

Dice ja Darts ala C. van Loan. Opiskele, kokeile ja esittele.

Avainsanat: mlTodari, mlPerusteet, matlabperusteet, Monte Carlo, looginen ineksinti, satunnaisluvut

142.

Olkoot F^n satunnaismuuttuja joka kuvaa kiinteiden pisteiden lukumäärää satunnaispermutaatiossa (so. alkiot joiden paikka ei muutu permutaatiossa).

- kirjoita funktio joka ottaa argumenttina kokonaisluvun n ja palauttaa k -pituisen otoksen jakaumasta F^n .
- Generoi otoksia jakaumasta F^n eri arvoilla n .
- Piirrä histogrammit eri otoksista.
- Voidaanko histogrammien perusteella päätellä, mikä on $E[F^n]$?
- Laske $E[F^n]$

Vihje: Käytä hyväksesi odotusarvon lineaarisuutta laskiessasi odotusarvoa $E[F^n]$.

-e

mlVektori

Vektorimuuttujan funktioita ja niiden diffint-laskentaa, toisin sanoin: usean muuttujan funktioiden diffinttiä

143. 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:

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

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 alkiokoittaiset 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 funktioiden tasa-arvokäyrät.

144. 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 päällekkäin komentamalla:

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

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

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

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

Kuinka laskisit numeerisen Jacobin matriisin?

Vihje: Muista, että numeerisesti

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

146.

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:

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

147. 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 π .

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 $\int_0^3 \int_2^3 xy \, dx \, dy = \text{dblquad}(@(\mathbf{x}, \mathbf{y}) \mathbf{x}.*\mathbf{y}, 2, 3, 0, 1)$. Jälleen kerran, tutki dokumentaatiota `function_handle`.

148. 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 komennolla `meshgrid` ja `mesh`.

149. 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`.

150. 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 määrittellä pinnan, eli matriisin z . Pinnan normaalivektorin saat laskettua funktiolla `surfnorm`. Piirrä ensiksi pinta komennolla `surf(x, y, z)`, sen jälkeen kirjoita `hold on`, jonka jälkeen kirjoita `quiver3(x, y, z, u, v, w)`, missä u, v ja w ovat `surfnorm`in palauttamien matriisien.