mlAritmetiikka

1. a) MATLABissa matriisit ja vektorit määritellään hakasulkeiden avulla. Luo tiedosto (tai vaihtoehtoisesti kirjoita komennot tulkkiin) jossa määrittelet vektorin $x2 = [2+i, 2-i, -4i]$, sekä matriisin

$$\mathbf{A} = \begin{bmatrix} i & 2i & 3 \\ 3 & 2 & -i \\ 1 & 3 & -i \end{bmatrix}.$$

Vektorin alkiot erotellaan joko välilyönnillä tai pilkulla, matriisin tai sarakevektorin rivinvaihdot tehdään puolipisteellä.

Laske tulo $y = \mathbf{A} * x2'$, missä $x2'$ on vektorin $x2$ transpoosi.

Tutki, mitä tekevät komennot `A(:,2)` ja `A(1,:)`?

b) MATLABissa kaikki operaattorit ovat oletusarvoisesti matriisioperaatioita. Usein kuitenkin halutaan käyttää *alkioittaisia* operaatioita. Tarkastellaan vektoria $x = [-1 : 0.1 : 1]$, jota hyväksikäyttäen haluttaisiin laskea arvot $x^2 + 4$. Koska määritelty x on vektori, laskutoimitukset `x^2` ja `x*x` tuottavat virheilmoituksen, ja `x*x'` ei tuota haluttua tulosta. Vastaus on MATLAB operaattori '.', joka ilmoittaa kyseessä olevan alkioittaisen toimenpiteen, esimerkiksi `x.*x`. Laske ja piirrä funktion $x^2 + 4$ kuvaaja välillä $[-1, 1]$.

Vihje: Huomaa, että ...

Ratkaisu: Tässsäpä näin ...

mlCurveFit

2. Piirrä suorat $y_1 = 0.2 - 0.5x, y_2 = 0.8 - 2x, y_3 = 0.2 + 2x, y_4 = -0.3 + 1.5x, y_5 = 0.95x$. Kirjoita nämä yhtälöt muodossa

$$\mathbf{A}[x; y] = \mathbf{b},$$

ja ratkaise syntyvä yhtälöryhmä pienimmän neliösumman mielessä käyttäen viimeviikkoista QR-hajotelmaa hyväksesi. Piirrä saamasi tulos.

Vihje: Siirrä termejä yhtälöissä niin että saat aikaan 5×2 matriisin, jolle voit suorittaa pienimmän neliösumman ratkaisun. MATLABin backslash osaa ratkaista myös pienimmän neliösumman tehtäviä. Suoran piirtämiseen et välttämättä tarvitse muuta kuin kaksi suoran pistettä.

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 polynomin 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.

Helpotettu versio: Jätetään cond pois.

Vihje: Olkoon $p(x) = a_0 + a_1 x + \ldots + a_n x^n$ etsitty polynomi.
Määrätään tuntemattomat kertoimet interpolaatioehtojen $p(x_k) = y_k, k = x_0, \ldots x_n$ avulla saatavasta lineaarisesta yhtälösysteemistä ratkaisemalla.
Kirjoita yhtälöryhmä tässä yleisessä muodossa ja tee ensin Matlab-skripti tyyliin

```
xd=...;
yd=...;
A=...;  % Yhtälöryhmän matriisi, help/doc vander
a=      % Ratkaisuna saatava kerroinvektori, help slash (a=A\...)d
x=linspace(alkup,loppup); % x-pisteet piirtämistä varten
y=polyval(...);           % Polynomin arvot x-pisteissä
...
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.

Ratkaisu:`mlCF03ratk.m` [TULEE] (Numero?)

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 backslashilla.

7. W.A Mozartin(1756-1791) sävellyksiä indeksoidaan Koechel-luvuilla, jotka ilmaisevat teosten sävellysjärjestyksen. Alla on eräitä Koechel-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 Koechel-numero on 364.

Vihje: Piirrä ensin datapisteet tasoon, ja päätä millaista menetelmää kannattaa käyttää.

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. Tutkitaan nk. Rungen 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

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

mukaan.

Vihje: Polynominen interpolaatio kannattaa tehdä MATLAB-funktiolla `polyfit`. Funktio g kannattaa määritellä funktiokahvan avulla: `g = @(x)1./(1+x`^{2}`)`$.Tasavälisiäpisteistöjäsaafunktiolla$`linspace`

10. 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 polynominen interpolointi datalle, ja ennusta väestön määrä vuonna 2010. Kuinka ennusteesi suhtautuu laskennan todelliseen tulokseen: 308,745,538 laskettua asukasta?

Vihje:

mlDifferentiaali(yhtälöt)

11. 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:

12. 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:

13. Ratkaise toisen kertaluvun alkuarvotehtävä

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

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

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

14. Eulerin menetelmä on klassinen menetelmä differentiaaliyhtälöiden ratkaisemiseksi. Menetelmän lähtökohtana on yhtälö

$$y'(t) = f(t, y(t)); y(t_0) = y_0,$$

josta koetetaan ratkaista $y(t)$ kun $t = [t_0, t_n]$. Menetelmä perustuu t:n diskretoinnille: t muunnetaan joksikin vektoriksi $T = (t_0, t_0 + h, \ldots, t_n)$, jonka jälkeen koetetaa etsiä vastaavia arvoja $y(t_0 + ih)$. Merkintöjen selvyyden vuoksi kirjoitetaan $t_0 + jh = t_j$. Eulerin menetelmässä approksimoidaan arvoa $y(t_i)$ seuraavasti:

$$y(t_i) = y(t_{i-1}) + h * f(t_{i-1}, y(t_{i-1})).$$

Ratkaise Eulerin menetelmällä differentiaaliyhtälö

$$y'(t) = y(t)\sin(t).$$

Käytä eri h:n arvoja, ja vertaa oikeaan ratkaisuun $e^{1-\cos(t)}$.

Vihje: Eulerin menetelmä MATLABissa on helpointa toteuttaa yksinkertaisena silmukkana. Olettaen, että yhtälön oikea puoli on kirjoitettu funktioon `f(t,y)`, Euler-iteraatio voidaan tehdä seuraavasti:

```
for n = 2:length(t)
    y(n) = y(n-1)+h*f(t(n-1),y(n-1));
end
```

missä t on vektori joka on ratkaisuvälin diskretointi, ja h on t:n pisteiden välinen etäisyys.

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

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

Diskretointiaskeleeksi tulee nyt

$$y_{n+1} = y_n + hk = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4).$$

a) Ratkaise edellisen tehtävän yhtälö

$$y'(t) = y(t)\sin(t); y(0) = 1.$$

käyttämällä Rungen-Kutan menetelmää. Vertaa tulosta sekä todelliseen ratkaisuun $y(t) = e^{1-cos(t)}$, että Eulerin menetelmällä saavutettuun. Mitä huomaat tarvittavasta askelkoosta?

b) Ratkaise alkuarvotehtävä

$$y'(t) = \sqrt{1 - y^2}; y(0) = 0$$

käyttämällä Rungen-Kutan menetelmää.

Vihje:

mlDiffint

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

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

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

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

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

Vihje: Muista, että alkioittaiset alkioittaiset operaatiot ilmoitetaan pisteellä.

17. [Polynomeja] HA

Vihje:

Ratkaisu: `./mlG07ratk.m`
`./html/mlG07ratk.html`

mlFunktiot

18. Välinpuolitusmenetelmä on eräs tapa löytää funktion nollakohta. Bolzanon lauseen nojalla, jos jatkuvalla funktiolla on jonkin suljetun välin $[a, b]$ päätepisteissä erimerkkiset arvot, sillä on vähintään yksi nollakohta tällä välillä. Välinpuolitusmenetelmä toimii seuraavasti:

- Laske välin $[a, b]$ keskikohta $m = \frac{b-a}{2}$.
- Laske $f(m)$. Jos $f(m) = 0$, ollaan löydetty nollakohta ja lopetetaan algoritmi.
- Jos $f(m)$:n merkki on sama kuin $f(a)$:n, voidaan tutkittavan välin vasenta päätepistetta siirtää kohtaan m, eli $a \leftarrow m$, ja palataan algoritmin kohtaan 1.
- Jos $f(m)$:n merkki on sama kuin $f(b)$:n, voidaan tutkittavan välin oikeaa päätepistetta siirtää kohtaan m, eli $b \leftarrow m$, ja siirrytään algoritmin kohtaan 1.

Tätä ideaa noudattaen, laske funktion $f(x) = e^{\sin(x^2)} e^{-x^2} \sin(x^2) - \frac{1}{2}$ välillä $[0, 1]$ sijaitseva nollakohta.

Vihje: Numeerisessa tapauksessa absoluuttisen nollan löytäminen on lähes mahdotonta – sinun tulee määrittää jokin hyväksyttävä toleranssi. Funktion arvojen samanmerkkisyyttä voidaan tutkia laskemalla näiden tulo: jos kahden luvun tulo on positiivinen, ovat ne samanmerkkisiä.

19. Sekanttimenetelmä on toinen funktion nollakohtien löytämiseen käytettävä menetelmä. Sekantti on suora joka leikkaa annettua käyrää kahdessa pisteessä. Sekanttimenetelmän idea on approksimoida annettua funktiota f välillä $[a, b]$ pisteiden $f(a)$ ja $f(b)$ välille piirretyllä suoralla. Tämän suoran, ja x-akselin välinen leikkauspiste otetaan välin uudeksi päätepisteeksi. Iteraatiokaavaksi saadaan näin

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

Sekanttimenetelmä suppenee yleensä, mutta ei aina, nopeammin kuin välinpuolitusmenetelmä.

Sekanttimenetelmää käyttäen laske funktion

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

välillä $[0, 5]$ sijaitseva nollakohta.

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

20. 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.

21. 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.

22. Kun $z = x + iy$ ja $-2 \leq x, y \leq 2$, eksponenttifunktion $z \mapsto \exp(z)$ kuvaajan voi piirtää seuraavasti:

```
t = -2:0.2:2;
[x y] = meshgrid(t,t);
z = x+i*y;
r = exp(z)
mesh(real(r));
```

Imaginääriosan saa piirrettyä komennolla `mesh(imag(r))`. Tee vastaavat graafit seuraavista kuvauksista edellämainitulla välillä.

1. $z \mapsto \log(z)$
2. $z \mapsto z^2$
3. $z \mapsto z + 1/z$

Vihje:

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

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

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

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

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

Kuinka maksimiperiaate ilmenee näiden funktioiden tapauksessa?

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

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

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

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

Vihje:

24. Näytä että jokaiselle kiinnitettylle arvolle x luku $S_n(x)$ lähestyy nollaa, kun $n \to \infty$, ja etsi $S_n(x)$:n ääriarvot x:n suhteen. Piirrä funktio $S_n(x)$ välillä $[-2, 2]$ kun $n = 2, 4, 6, 8, 10$.

Vihje: Ohjelman suorituksen voi keskeyttää kesken skriptin komennolla `pause`. Välin voi määritellä joko vektorinotaatiolla `-2:0.02:2` tai komennolla `linspace(-2,2,100)`.

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

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

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

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

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

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

mlGrafiikka

26. 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 grafiikkaruutu osiin. Tämä onnistuu `subplot`:n avulla. Kokeile näitä vaihtoehtoisia tapoja (nyt tai myöhemmin).

27. Piirrä

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

Vihje: Muista pisteet laskutoimituksissa!

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

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

Vihje: Pane merkille tällaiset grafiikan ulkoasua säätelevät lisäkomennot (jotka voidaan antaa jälkikäteen (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).

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

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

30. 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.

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

```
% % 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:

```
>> potenssipiirto
```

Tällöin tiedostossa olevat Matlab-komennot suorittuvat.

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

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

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.

32. [**3D-grafiikkaa, korkeuskäyriä**] **HA**

Olkoon

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

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

Vihje: Opiskele:
`http://math.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`

33. HA [MatOhj2012]

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 graiikkaikkuna
>> plot(x,f-exp(x))
```

34. Olkoot c ja z_0 kompleksilukuja. Tällöin rekursion

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

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

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

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

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

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

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

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

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`.

35. 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.

mlLinis

36. Piirrä suorat $y_1 = 0.2-0.5x, y_2 = 0.8-2x, y_3 = 0.2+2x, y_4 = -0.3+1.5x, y_5 = 0.95x$. Kirjoita nämä yhtälöt muodossa

$$\mathbf{A}[x; y] = \mathbf{b},$$

ja ratkaise syntyvä yhtälöryhmä pienimmän neliösumman mielessä käyttäen viimeviikkoista QR-hajotelmaa hyväksesi. Piirrä saamasi tulos.

Vihje: MATLABin backslash-ratkaisin toimii myös pienimmän neliösumman kanssa.

37. Tee matriisille

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

LU hajotelma ja ratkaise sitä käyttäen yhtälöryhmä $\mathbf{Ax} = \mathbf{b}$, kun

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

Käytä ratkaisussasi hyödyksi kolmiomatriisien rakennetta, eli älä käytä MATLABin ”backslash” ratkaisinta.

Vihje:

mlMatriisit

38. Potenssimenetelmä on eräs keino löytää magnitudiltaan isoin ominaisarvo ja -vektori. Menetelmä toimii seuraavasti:

- Valitse alkuarvaus $\mathbf{b}_0$. Ainoa vaatimus on, että tällä vektorilla on nollastapoikkeava komponentti ominaisarvon suuntaan – käytännössä kannattaa valita vektori, jonka jokainen alkio on nollasta poikkeava.
- Aseta
$$\mathbf{b}_{k+1} = \frac{\mathbf{A}\mathbf{b}_k}{||\mathbf{A}\mathbf{b}_k||}$$
- Jatka kunnes jono $(\mathbf{b}_k)$ suppenee. Ominaisarvo $\lambda = ||\mathbf{A}\mathbf{b}_k||$ ja vektori $\mathbf{x} = \mathbf{b}_k$.

Toteuta menetelmä MATLABissä, ja laske matriisin `gallery(5)` isoin ominaisarvo ja -vektori. Testaa tuloksen oikeellisuus.

Vihje:

39. Tehdään LU-hajotelma tuentaa hyväksikäyttäen. Ensimmäiseksi, yritetään ymmärtää, kuinka tuenta toimii seuraavan pseudokoodin avulla.

[Kommentoitu pois codebox-osuus]

Kirjoita vastaava MATLAB-funktio, ja ratkaise sen avulla ongelma $\mathbf{A}\mathbf{x} = \mathbf{b}$, kun

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

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

Vihje:

Hilbertin matriisi MATLABina:

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

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

- Ortogonalisoidaan:
  $\mathbf{u}_1 = \mathbf{v}_1$
  $\mathbf{u}_2 = \mathbf{v}_2 - \frac{\mathbf{u_1 \cdot v_2}}{\mathbf{u}_1 \cdot \mathbf{u}_1}\mathbf{u}_1$
  $\vdots$
  $\mathbf{u}_n = \mathbf{v}_n - \sum_{k=1}^{n-1} \frac{\mathbf{u}_k \cdot \mathbf{v_n}}{\mathbf{u}_k \cdot \mathbf{u}_k}\mathbf{u}_k$
- Normitetaan: $\mathbf{e}_i = \frac{\mathbf{u}_i}{||\mathbf{u}_i||}, i = 1 \ldots n$

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

Vinkki: Laskutoimitus $\frac{\mathbf{u}_k \cdot \mathbf{v_n}}{\mathbf{u}_k \cdot \mathbf{u}_k}\mathbf{u}_k$ vastaa toimitusta $(\mathbf{u}_k^T \mathbf{v}_n)\mathbf{u}_k$. **Lisätehtävä nopeille:** Matriisin sarakeavaruuden normalisointi ei poikkea kovin paljon QR-hajotelman tekemisestä. Jos ehdit, toteuta oma algoritmisi QR-hajotelmalle.

Vihje:

41. Matriisi $(a_{ij}) = \mathbf{A} \in \mathrm{R}^{n\times m}$ on *yläkolmiomatriisi*, jos $a_{ij} = 0$ kun $i > j$.
(a)Kirjoita MATLAB funktio, joka ratkaiseen yhtälöryhmän $\mathbf{Ax} = \mathbf{b}$, kun $\mathbf{A}$ on yläkolmiomatriisi.
(b) Generoi satunnaisia yläkolmiomatriiseja, ja tutki josko

(1) kahden yläkolmiomatriisin tulo on aina yläkolmiomatriisi,

(2) yläkolmiomatriisin käänteismatriisi on aina yläkolmio.

(3) determinantti on aina nollastapoikkeava.

Vihje: Satunnaisia matriiseje voi luoda kommennoilla `rand` ja `randn`. Näitä kertomalla saa matriiseja joiden arvot ovat millä vain halutulla välillä. Yläkolmion saa matriisista $\mathbf{A}$ komennolla `triu(A)`.

mlNonlinequ, Epälineaariset yhtälöt

42. Historiallisesti mielenkiintoinen yhtälö on

$$x^3 - 2x - 5 = 0,$$

jota Wallis-niminen matemaatikko käsitteli, kun hän ensi kertaa esitteli Newtonin menetelmää Ranskan akatemialle. [Lähde: Moler NCM]

1. Piirrä kuvaaja saadaksesi alkuarvon Newtonin menetelmälle reaalijuurta varten.
2. Määritä reaalijuuri omanewton:lla Neuvo: Polynomifunktion voit määritellä tähän tapaan, kun p on kerroinvektori:

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

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

Vihje:

Ratkaisu:`MlNl01ratk.m`[Tulee]

43. [NCM 4.15, p. 138]

Keplerin malli:

$$M = E - e\sin,\ E$$

1. Ratkaise fzero:lla
2. Sarjakehitelmä:

$$E = 2\sum_{m=1}^{\infty} \frac{1}{m} J_m(me)\sin(mM)$$

Vihje:

Ratkaisu:`MlNl02ratk.m`[Tulee]

44. [NCM 4.15, p. 138]

Vesiputken syvyys, jotta ei jäädy.

Vihje:

Ratkaisu:`MlNlxxratk.m`[Tulee]

matlabteht/mlPerusteet, Matlab-perusteita

---

45. 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.

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

47. 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 alkioiden neliöjuuri.
4. Laske vektorin alkioiden neliöt ja neliösumma.

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

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

1. Lisää vektorin x alkioiden summa vektoriin y
2. Korota vektorin x alkiot vektorin y vastinalkioiden osoittamiin potensseihin.
3. Jaa y:n jokainen alkio vektorin x vastinalkiolla.
4. Kerro x:n jokainen alkio y:n vastaavalla alkiolla ja talleta tulos muuttujaan z

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

49. Muodosta vektori x, joka koostuu alkioista:

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

Vihje: Kahdessa viimeisessä kohdassa voit selkeyttää komentamalla:
`format rational`
Paluu oletusformaattiin: `format`

50. 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              % Mik\"a vektori?
x^2               % Miksi virhe ?

sqrt(x*x')
sqrt(sum(x.^2))   % Miksi sama tulos kuin edell\"a?
norm(x)
help norm
```

Vihje:

Ratkaisu: Skripti: ./mlP06ratk.m
Diarytiedosto: ./mlP06ratk.txt

51. Määrittele matriisit

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

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

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

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

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

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

Ratkaisu:

Skripti: ./ mlP07ratk.m

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

Vihje:

Ratkaisu:

Avainsanat: Matlab perusteet, Matriisikertolasku, taulukko-operaatiot.

53. Luentotehtävä
Viite: `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 suuruisten alkulukujen lukumäärän ja summan. Aloita tiedosto näin:

```
% Selit"\a, mit"\a skripti tekee ja vaikka oma nimi, pvm. ym.
N = 100
lkm = ...
summa= ...
```

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

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

Indeksointi, length, clear

Vihje:

Ratkaisu:

Avainsanat: Matlab perusteet, skripti, m-tiedosto, alkuluvut, luentotehtävä

54. Luentoteht.

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 $w = [u_1, v_1, u_2, v_2, \ldots]$
   Vihje: Liitä vektorit allekkain ja jonouta näin saatu 2-rivinen matriisi sarakkeittain (ovelaa). (Sarakkeittain jonoutus matriisille A saadaan näin: `A(:)`.)

Vihje:

Ratkaisu:

Avainsanat: Matlab perusteet, fliplr,flipud, kaksoispiste (:), vastakkainen järjestys, matriisin jonoutus, "merge", limitys, luentotehtävä

55. Matriisin kokoaminen osista, lohkomatriisit, skriptit

Nyt jo viimeistään on syytä ottaa skriptit käyttöön. Kts. skriptiohjetta (laitetaan myös tähän) (myös help script)

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

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

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

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

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

Lopuksi voit käyttää `publish`-komentoa dokkarin aikaansaamiseksi. Totuttele tähän, ohjeita on ... (tähän viittauksia).

Vihje:

Ratkaisu:

Avainsanat: Lohkomatriisit, skriptit, m-tiedostot, diag.

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

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

Vihje:

Ratkaisu:

Avainsanat:

57. 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 (kts. yllä, teht. 6)

Vihje:

Ratkaisu:

Avainsanat: Markovin matriisit, prosessit, ominaisarvot.

Liittyy: Lineaarialgebra/ominaisarvot. Aiemmissa kurssimatskuissa, my ös Solmu-kirjoituksessa, koetehtävissä on monia tehtäviä ratkaisuineen.

58. Rekursiokaava

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

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

59. Näytä numeerisesti 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 $n \to \infty$.

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

---

matlabteht/mlTodari, Todennäköisyys, satunnaisuus, tilasto

---

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

a) kirjoita funktio joka ottaa argumenttina kokonaisluvun n ja palautaa k-pituisen otoksen jakaumasta F^n.

b) Generoi otoksia jakaumasta F^n eri arvoilla n.

c) Piirrä histogrammit eri otoksista.

d) Voidaanko histogrammien perusteella päätellä, mikä on $E[F^n]$?

e) Laske $E[F^n]$

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

61. HA [MatOhj2012, projekti] 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 $-1 < x <, 1 < y < 1$, niin ympyrään ja neliöön osuneiden tikkojen lukumäärän suhde lähenee lukua $\pi/4$, kun satunnaisheittojen lukumäärä kasvaa. Miksi? Generoi tasajakautuneita pistepareja ja laske ko. osuus.

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.

Vihje:

Ratkaisu:

mlVektori

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

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

Kuinka laskisit numeerisen Jacobin matriisin?

Vihje: Muista, että numeerisesti

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

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

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

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

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

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

kuva:

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

.

Piirrä näillä keinoin seuraavien funktioiden kuvat

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

- $f(x, y) = x^3 - 3xy$ alueessa $[-1, 1] \times [-1, 1]$.

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

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

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

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

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

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

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

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

65. 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

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

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

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