Aalto-yliopisto, Matematiikan ja Systeemianalyysin laitos **Apiola/Kuortti**

Mat-1.C Matemaattiset ohjelmistot

Harjoitus 1 11-14.3.2013

Kokoelmassa on “tavallisia” ja **DOKU**-tehtäviä.

- Doku-tehtävistä saa valita kaksi, jotka lähetetään Juhalle ma 18.3.13 klo 13.00 mennessä. (juha.kuortti@aalto.fi) . Arvostelu 1–3 p.
- Muut tehdyt, vähintään 4 kpl (ylimääräiset dokutehtävät mukaanlukien) merkitään ma 18.3 rastilistalle. Silloin keskustellaan niistä.
- Kurssin loppupuolella annetaan pieniä “projektitehtäviä”, joista valitaan yksi. (Minimiläpipääsyyn ei ihan välttämättä tarvita.)

Minimiläpipääsyyn: Dokut (2/harj.) + muut (4/harj)+läsnäolo.
Arvostelu normaalisti 1–5 (tarkoitus on, että kaikki pääsevät).

Poikkeustilanteisiin, kuten sairastumiseen ym. suhtaudutaan myötämielellä.

Kaikkien tehtävien, ja aivan erityisesti Doku- ja projektitehtävien dokumentointiin käytetään Matlabin “publish”-tyyliä. Tarkoittaa, että tehdään m-tiedosto tyyliin

```
%% Otsikko ja nimi
% Selostusta
%
koodia
%% Uusi kappale (otsikko tai ei)
%
```

Suoritetaan kapple kerrallaan ja lopuksi kokonaan. Julkaise itse “publish”-työkalulla, mutta lähetä Juhalle vain m-tiedosto. (Harjoitellaan yhdessä.)

---

1. mlP0001.tex

Olkoon `x = [3 1 5 7 9 2 6]`. Selvitä itsellesi, mitä seuraavat komennot tekevät:

```
a. x(3)
b. x(1:7)
c. x(1:end)
d. x(1:end-1)
e. x(6:-2:1)
f. x([1 6 2 1 1])
g. sum(x)
```

2. mlP0002.tex

Olkoon `A = [ 2 4 1 ; 6 7 2 ; 3 5 9]`.
Millä Matlab-komennoilla suoritat seuraavat toimet:

a. Korvaa A:n 1. rivi vektorilla, joka koostuu kakkosista

b. Muuta A:n 2 viimeistä riviä vastaluvuikseen.

c. Laske A:n sarakesummat

d. Laske A:n rivisummat

e. Laske kunkin sarakkeen “keskiarvon standardivirhe”, jolla tarkoitetaan ko. lukujen standardipoikkemaa (help std) jaettuna lukujen lukumäärän neliöjuurella.

Käytä yleiselle matriisille toimivia parametrejä, kuten `end, [m,n]=size(A)`.

3. mlP0005.tex

USA:n väestömäärälle on seuraava ennustemalli:

$$P(t) = \frac{197273000}{1 + e^{-0.0313(t-1913.25)}},$$

missä aika t on vuosia. Piirrä $P(t)$ välillä 1790 . . . 2000. Mitä malli ennustaa vuosille 2010 ja 2020 ?

Käsittele ensin lausekkeena tavalliseen tapaan:

```
>> t=...    ;
>> P=...    ;
>> plot(t,P)
>> t=2010
>> P =   % Huomaa: P -rivi on laskettava uudestaan (tietenkin)
```

Määrittele sitten P funktioksi tyyliin: `P=@(t) lauseke(t)`. Suorita piirto uudestaan, kokeile myös `fplot`-piirtofunktiota.

Funktion arvon laskenta on nyt erityisen helppoa.

4. mlG11.tex

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 $-2 \le x \le 2$. Oletko tyytyväinen kuvaan? Käytä `axis` komentoa asettaaksesi näytettävät akselit väleille $-2 \le x \le 2$ ja $-10 \le x \le 10$. (Tai pelkästään `ylim`.)

d) Piirrä nimittäjän kuva eri ikkunaan `figure`. Miksei räjähdä muiden nollakohtien lähistöllä? Piirrä samaan kuvaan myös osoittaja eri värillä tähän tapaan:
`plot(x,oso,'b',x,nimi,'r')`. Kirjoita vielä "legenda":
`legend('Osoittaja','Nimittaja')` (Ääkköset voi olla ongelmallisia joskus, edelleen!) ja `grid on; shg`.
Valaistunet. (PS. Toki osaat tyyppiä $\sin\alpha = -\cos\beta$ olevan yhtälön ratkaista trigonometriaa käyttäen, tässä on kyse Matlab-työvälineiden opettelusta.)

e) "Vapaaehtoinen" lisä: Etsi nimittäjän mahd. nollakohdat (tai lähes nollakohdat) Itse asiassa tällä diskretoinnilla luokkaa: `ind=find(abs(nimi)<0.1)`. Muodosta taulukko:
`[oso(ind)' nimi(ind)']`

Vihje: Jako- ja kertolaskujen tapauksessa ole tarkkana: haluatko matriisioperaation (et kai) vai alkioittaisen operaation? Alkioittaiset operaatiot erotetaan matriisioperaatioista operaattorin eteen sijoitettavalla pisteellä. Esimerkiksi `.*` on alkioittainen kertolasku, `*` matriisien kertolasku.

KOMPA: `1/x` tarkoittaa matriisijakolaskua eli lineaarisen yhtälöryhmän ratkaisua, jossa kerroinmatriisina on vektori x. Onneksi johtaa virheeseen. Täytyy siis muistaa `1./x`, vaikka kaikissa muissa tapauksissa (kuten esim.`2*x`) skalaarilla operoiminen käy ilmankin pistettä.

Trigonometriset funktiot toimivat MATLABissa alkioittain, ja löytyvät (lähes) tavanomaisilla nimillä. (`cos, acos, sin ..., help elfun`)

Tasavälisiä pistejoukkoja luodaan komennolla `linspace`, tai vaihtoehtoisesti MATLABin kaksoispiste-notaatiolla.

5. mlP007.tex

Määrittele matriisit

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

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.

6. mlG007.tex

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`,

Tee itsellesi selväksi `meshgrid`.

Piirtoon: `mesh, surf, surfc, surfl ..., contour, colorbar,...`

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

Laske myös A^{-1} ja kerro sillä b. (Yleisesti takakenon käyttö yhtälöryhmän ratkaisussa on suositeltavampaa (vaikka MIT-materiaalissa toisin neuvotaan).

8. mlLi0001.tex

Vektorien ja matriisien normit. Näitä tarvitaan mm. lineaarisen yhtälösysteemin häiriöalttiuden mittareina.

Vektorin $\vec{v} \in \mathbb{R}^n$ "ykkösnormi" (ns. Manhattan-normi) on

$$||\vec{v}||_1 = \sum_{k=1}^{n} |v_k|$$

Tähän liittyvä matriisin A normi on

$$||A||_1 = \max_{j=1...n} \sum_{i=1}^{n} |a_{ij}|$$

Kirjoita (nautiskellen) Matlab-lauseke, jolla lasket matriisin A ns. ykkösnormin. (siis matriisin alkioiden itseisarvojen muodostaman matriisin sarakesummien maksimin.) Vastaavasti ääretön-normi saadaan rivisummien avulla. No kirjoita ja testaa vaikka satunnaislukumatriiseilla ja vertaa valmiin `norm`-funktion antamiin.

Huom! uusissa Matlab-versioissa summaukseen voidaan liittää indeksisuunta, vanhemmissa edetään aina sarakkeita pitkin, jolloin rivisuunta on tehtävä transponoimalla. (Tämä uudistus oli välttämätön, kun useampiulotteiset taulukot tulivat mukaan.)

Vihje: Miksi käytetään "kulmikkaita" vektorinormeja, eikä euklidista? Euklidisen normin tapauksessa laskenta on raskaampaa, joudutaan ns. singulaariarvoihin. Näissä helppo suora kaava (jopa helppo johtaa).

9. mlLi0002.tex

Neliömatriisin A "häiriöluku","kuntoluku" "condition number"on
$\kappa(A) = cond(A) = ||A||||A^{-1}||$, missä $||A||$ on jokin matriisinormi. Tutki Vandermonden matriisien häiriöalttiutta (1-normin suhteen) tyyliin:

```
n=2;
x=-n:n;
V=vander(x), VI=inv(V)
cond1=norm(V,1)*norm(VI,1)
```

Kasvata n:ää ja aja uudelleen.

Mitä havaitset. Jos haluat, voit rakentaa pienen for-silmukan tähän tapaan:

```
N=5;
for n=1:N
        x=-n:n;
        V=vander(x); VI=inv(V);
        size(V)
        cond1=norm(V,1)*norm(VI,1);
end;
```

Huomaat, että format long on tarpeen.

Matlab:ssa on funktioita: cond, condest, rcond häiriöluvun estimointiin. (Määritelmän mukaan laskeminen on suurilla matriiseilla raskasta.) Nyrkkisääntö: Lineaarisen yhtälöäsysteemin $Ax = b$ ratkaisussa on varauduttava siihen, että tuloksen suhteellisesta tarkkudesta katoaa cond(A):n suuruusluokan verran merkitseviä numeroita.

Vihje: Vrt. `http://math.aalto.fi/~apiola/matlab/opas/lyhyt/tehtavia2.html`
Opiskele Matlab

10. mlLi003.tex

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

$$C = aF + b.$$

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

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

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

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

11. mlP021.tex

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

Testaa ainakin näille:

a) `A=1:10`

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

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

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

12. mlP021a.tex

Suorita Matlab-komennot:

```
>> x=linspace(-2*pi,2*pi);
>> y=sin(x);
>> z=(y>=0).*y;
>> plot(x,z);shg
```

a. Selitä, mikä on Matlab-ajatus havainnollistamalla vaikka 10:n pituisella x-vektorilla.

b. Muodosta tästä puoliaaltotasasuunnatusta siniaallosta sellainen, joka edellisen nollaväleillä noudattaa funktiota $\frac{1}{2}|\sin x|$.

c. Piirrä hienommin diskretoiden (normaalisti sataan osaan).

13. mlP015.tex

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

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

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

n	$x(n)$	virhe
0	1	$\sqrt{a} - 1$
1	$x(1)$	$\sqrt{a} - x(1)$
$\vdots$	$\vdots$	$\vdots$
N	$x(N)$	$\sqrt{a} - x(N)$

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

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

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

Taulukon voit rakentaa vaikka liittämällä 3 pystyvektoria vierekkäin tai miksei suoraan for-luupissa. Edellinen tapa on ehkä selkeämpi. Kannattaa alustaa (pysty)vektorit `x=ones(N,1); virhe=zeros(N,1);` (Matlab ei sitä vaadi, mutta tehokkaampaa ja selkeämpää.) Numerointivektoria ei suotta ajeta for-luupissa, vaan ...

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

Huom: Indeksointi alkaa 1:stä

14. **DOKU** mlT005.tex

Monte Carlo-approksimaatio π:lle.

Piirrä yksikköneliö ja sen sisään reunoja sivuava ympyrä. Neliön (yleensä monikulmion) piirtoa varten voit kirjoittaa nurkkapisteiden x-koordinaatit vektoriin x ja vastaavat y-koordinaatit vektoriin y. (Muista, että loppupisteeksi pitää ottaa alkupiste uudestaan, jotta kuvion viimeinenkin särmä piirtyy.)

Heitetään tikkaa kuvan mukaiseen tauluun (tikat eivät eksy taulua ympäröivään neliön ulkopuolelle, ehkä tähän oikeasti tarvitaan "satunnaisrobotti"). Jos tikkojen osumatarkkuus 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.

Piirrä tikkatauluun osuneet pisteet jollain värillä ja ulkopuolelle jääneet jollain toisella.

Tee työstä doku (tottakai), julkaise pdf-tiedostona.

Vihje: Avainidea: (Anteeksi, jos riistin keksimisen ilon!) `X.^2+Y.^2 <= 1` ja bittivektorin ykkösten lukumäärän laskeminen

15. DOKU mlLi004a.tex

Hilbertin matriisi koostuu alkioista $H(i,j) = \frac{1}{i+j-1}$. Se esiintyy mm. PNS-sovitustehtävissä. Luepa Matlabin luojan, *Cleve Moler'n* hiljattain kirjoittama blogikirjoitus aiheesta:
`http://blogs.mathworks.com/cleve/2013/02/02/hilbert-matrices/`

(Sivumennen sanoen "shakkilaudan" voi tehdä vähemmällä teorialla näinkin:)

```
for i=1:8
   for j=1:8, A(i,j)=(-1)^(i+j);end
end; A
imagesc(A); axis image; axis off; colormap(copper)
```

(Ethän voi olla kokeilematta !)

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

Muodosta 3×3- Hilbertin matriisi H.
Tarkastellaan systeemiä $Hx = b$ kahdella b-vektorilla:

$b_1 = [0.95; 0.67; 0.52]$ ja b_2, missä $b_2(3) = b_1(3) + 0.01$.

a) Ratkaise molemmat systeemit $Ax = b_k, k = 1, 2$

b) Laske b-vektorien ja x-ratkaisujen normien erotukset $||\Delta b||$ ja $||\Delta x||$ ja suhteelliset virheet jakamalla $||x||$:llä ja vastaavasti $||b||$:llä.

c) Lineaarisen yhtälösysteemin herkkyyttä pienille virheille sanotaan *häiriöalttiudeksi* (*"ill-conditioned "*). Laske kummankin matriisin häiriöalttius sekä määritelmän mukaan että `cond`-funktiolla.

   Suhteellisen virheen suurtenemisypäyhtälö:

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

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

   Vertaa epäyhtälön molempia puolia, onko suuruusluokka kohdallaan?

d) Kerro muutamalla virkkeellä, mitä opit Molerin blogista ja komennoista `doc hilb, doc invhilb`.

Vihje: `hilb, invhilb`

16. mlCF02.tex

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: polyfit, polyval .

Sinun on tiedettävä, mikä on polynomin asteluku.

Tarkistus: Kulkeeko polynomi kaikkien datapisteiden kautta.

17. **DOKU** mlCF03.tex

Kirjoita funktio, jonka otsikko ja "help-kommentit" voisivat olla:

```
function [kertoimet,condi]=vandinterp(xdata,ydata)
% Funktio laskee vektoreihin xdata ja ydata liittyvän
% interpolaatiopolynomin kertoimet Vandermonden
% systeemin ratkaisulla ja palauttaa myos cond-luvun.
%  Esim:
%  xdata=0:5; ydata=xdata.*sin(xdata);
%  c=vandinterp(xdata,ydata);
%   tai
%  [c,kunto]=vandinterp(xdata,ydata);
```

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

Vihje: Olkoon $p(x) = a_0 + a_1 x + \ldots + a_n x^n$ etsitty polynomi.
Määrätään tuntemattomat kertoimet interpolaatioehtojen $p(x_k) = y_k, k = x_0, \ldots x_n$ avulla saatavasta lineaarisesta yhtälösysteemistä ratkaisemalla.
Huomaa, että Matlabissa polynomi esitetään korkeimman asteisesta alkaen.
Huomaat tarvitsevasi "Vandermonden matriisia", siispä: `help vander`

Ohjelman kehitysvaiheessa voit tehdä ensin Matlab-skripti esim. kommenttiesimerkkidataa käyttäen tyyliin

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

Tarkistus: Kulkeeko polynomi kaikkien datapisteiden kautta.

Kun skripti toimii, tee sen pohjalta pyydetty funktio (ellet jo tehnyt sitä suoraan). Sovella funktiotasi myös edelliseen interpolaatiotehtävään.

18. 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^o

   Kirjoita oletukset skriptiisi.

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

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

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

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

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

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

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

19. mlDi0001.tex (a) Määritä funktion $f(x) = x^4 - 7x^3 + 11x^2 + 7x - 10$ nollakohdat, ja ääriarvot käyttäen funktioita `roots` ja `polyder`. Piirrä kuvaaja.

(b) Määritä funktion maksimi- ja mimimipisteet ja arvot “raakaa voimaa” käyttämällä soveltaen Matlabin tehokkaita vektori- ja datanpoimistyökaluja.

(c) Määritä samalla periaatteella f:n nollakohdat.

Vihje:

`polyval, polyder, roots`

(b) Tarkoitus on oikeastaan muuttaa kuvasta katsominen kvantitatiiviseksi tähän tyyliin:

```
>> x=linspace(a,b,N);
>> y=polyval(...);
>> ymax=...  % suurin y-vektorin arvo
>> maxind=find(...)  % Etsi suurimman arvon indeksi.
>>   % Indeksoi silla x-vektori.
```

Kokeile toki myös funktiota `fminsearch`

(c)

```
>> xpist=linspace(a,b,N); % Sopiva vali ja N.
>> find(...)   % Etsi eka merkinvaihtopiste.
>>          % silla valilla on 0-kohta.
>>          % toista nuolinappaimella tai CTR-ENTER
```

Tässä tehdään välin puolittamisen sijasta välin jakaminen esim. 100:aan tai 1000:een osaan. Huomaa, että pitkillä vektoreilla operointi on Matlabilla tehokasta.

20. DOKU mlG023.tex

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.

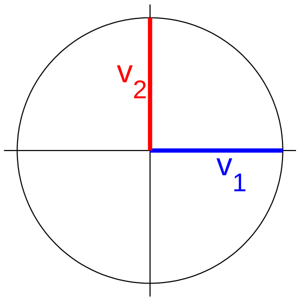
Vihje: Uusi grafiikkaikkuna: `figure`
Muistathan ympyrän luonnollisen parametriesityksen.
Ympyrän data koostuu oikeasti säännöllisesen n-kulmion nurkkapisteistä, missä esim. $n = 100$

c) Piirrä ympyrä parametrimuodossa (edellinen tehtävä tarpeeksi isolla n), ja tallenna koordinaatit matriisimuodossa `coords = [x;y]`. Tämän jälkeen piirrä ympyrä, ja lisää skriptiin seuraavat komennot

```
hold on
% Piirretään akselit.
plot([-1.1 1.1],[0 0],'k','linewidth',1)
plot([0 0],[-1.1 1.1],'k','linewidth',1)
%
plot([0,1],[0 0],'b','linewidth',4)
plot([0,0],[0 1],'r','linewidth',4)
p1 = text(.5,-.2,'v_1','fontsize',30);
set(p1,'color',[0 0 1])
p2 = text(-.25,.5,'v_2','fontsize',30);
set(p2,'color',[1 0 0])
% Axis asetukset
```

```
axis([-1.1 1.1 -1.1 1.1])
axis equal
axis off
```

Tuloksen pitäisi olla jotain tällaista:



Jatketaan skriptin editoimista. Määritä matriisi

$$\mathbf{A} = \begin{bmatrix} \frac{7}{6} & -\frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} \end{bmatrix}.$$

a) Laske matriisin $\mathbf{A}$ ominaisarvot ja piirrä ominaisvektorit samaan kuvaan (hold on).

b) Operoi matriisiin `coords` (= ympyrän koordinaattipisteet) matriisilla $\mathbf{A}$ ja piirrä myös transformoidun ympyrän kuva samaan kuvaan.

   Lisäys ma 11.3. annettuun:

c) Olkoot ominaisarvot λ_1, λ_2 ja vastaavat ominaisvektorit v_1, v_2. Piirrä vektorit $\lambda_k v_k, k = 1, 2$ ja $-Av_k, \quad k = 1, 2$ samaan kuvaan (Miinus, jotta erottuvat paremmin.)

d) Aja skripti uudelleen niin, että muutat matriisin symmetriseksi, esim näin:
   `A=rand(2,2); A=A'*A`.
   (Skriptiin voit yleensäkin kerätä kommenttimerkkirivejä, joihin laitat eri matriiseja, sitten vain annat palaa uudestaan, helppoa!)
   Vertaa kuvia, miten symmetrinen eroaa?

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

22. mlCF04.tex

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:

23. mlNL08.tex

Newtonin menetelmä on tehokas (muttei "robusti") menetelmä etsiä funktioiden nollakokohtia. Suppeneminen on huikean nopeaa (kvadraattista), kun alkupiste on riittävän hyvä, ja eräät yleiset ehdot täyttyvät. Se perustuu iteraatioon

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

missä f on funktio, jonka nollakohtaa etsitään, ja x_0 on jokin alkuarvaus funktion nollakohdaksi.

Kirjoita skripti, jossa ratkaiset seuraavat yhtälöt

a) $x \cos x = \sin x + 1, \quad 0 < x < 2\pi$

b) $x^2 + \sin x = 8$

c) Kirjoita funktiot MATLABin `f=@(x) lauseke(x)` notaatiolla, ja ratkaise ne myös käyttämällä MATLAB-funktiota `fzero`.

Vihje: Vaikka esiintyvät derivaatat eivät ole vaikeita, voidaan niiden laskemiseen käyttää MATLABia (tai Maplea toki myös). Kokeile seuraavaa:

```
syms x
diff(x*cos(x),x) % Aivan sama kuin Maple-syntaksi.
```

24. DOKU mlNl09.tex
Maple , Matlab

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`

25. DOKU mlCF017.tex
Oletetaan, että meille on annettu dataa muodossa $(x_k, y_k).k = 1 \ldots m$, johon muodustuu kaksi murtopisteen erottamaa lineaarista suuntausta. Esimerkiksi

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

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

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

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

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

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

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

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

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

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

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

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

Murtopisteet, sovitus

26. **DOKU** mlCF13.tex/mplCF13.tex

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 ennusteesi suhtautuu laskennan todelliseen tulokseen: 308,745,538 laskettua asukasta?

Sovita myö eriasteisia PNS-polynomeja, vrt. Matlab Censusgui, lue Molerista: `http://www.mathworks.se/`
Comp. with Matlab, interpolation

Vihje: