

# Harjoitus 2: Ohjelmointi (Matlab)

MS-C2107 Sovelletun matematiikan tietokonetyöt 2023



## 2. Harjoituskerta

### **Aiheet:**

- Matlabin kontrollirakenteet
- Funktiot ja komentojonotiedostot Matlabissa

### **Osaamistavoitteet:**

- Osaat laatia omia funktioita ja komentojonotiedostoja
- Hallitset tavallisimpien kontrollirakenteiden käyttämisen

## Relaatio- ja loogiset-operaattorit Matlabissa

< Pienempi kuin

<= Pienempi tai yhtä suuri kuin

> Suurempi kuin

>= Suurempi tai yhtä suuri kuin

== Yhtä suuri kuin

~= Eri suuri kuin

&& Palauttaa arvon **true** jos molemmat tarkasteltavat lausekkeet saavat arvon **true**, muussa tapauksessa palauttaa arvon **false**.

|| Palauttaa arvon **true** jos jompikumpi inputeista (tai molemmat) saavat arvon **true**, muussa tapauksessa palautetaan **false**.

**Huom!** = on sijoitusoperaattori ks. viime luento).

## Ohjelman kontrollirakenteet - if, else, elseif

- **Haarautuminen**: Ohjelmakoodissa voidaan edetä eri tavoin riippuen ohjelman tilasta.

- **if, else, elseif**

```
if (x < 0)
    disp('luku x on negatiivinen')
elseif (x > 0)
    disp('luku x on positiivinen')
else
    disp('luku x on nolla')
end
```

- Huom! **if** aloittaa ehtolausekkeen ja **end** lopettaa
- **elseif** ja **else** ovat optionaalisia

## Ohjelman kontrollirakenteet - for

- **Silmukoiden** avulla joukko komentoja voidaan ajaa useamman kerran peräkkäin.
- **for** - silmukka
  - Käytetään kun silmukan toistojen lukumäärä on ennaltamäärätty.
  - Esim. Käydään läpi kaikki matriisin rivit.

```
for muuttuja = alkuarvo : askelpituus : loppuarvo
    % Suorita komentoja
end
```

## Esimerkki silmukasta

- Tehdään 'ohjelma', joka tulostaa joka kierroksella indeksin  $i$  toisen potenssin.

```
for i=1:2:7
    i^2
end
```

```
ans =
     1
```

```
ans =
     9
```

```
ans =
    25
```

```
ans =
    49
```

## Ohjelman kontrollirakenteet - while

- `while` - silmukka
  - Käytetään kun toistojen lukumäärä ei ole ennaltamäärätty.
  - Esim. Haetaan ratkaisua iteratiivisesti kunnes tietty toleranssiehto täyttyy.

```
x=0;
while (cos(x) > 0)
    x=x+0.1;
end
x

x =

    1.6000
```

## continue, break, return

- Silmukoissa käytettäviä komentoja:
  - **continue**: Siirrytään silmukan alkuun seuraavalle iteraatiokierrokselle.
  - **break**: Siirrytään ulos silmukasta.
  - **return**: Lopetetaan koodin suoritus.

```
for x=0:0.1:pi
    if (cos(x)<=0)
        break;
    end
end
x
x =
    1.6000
```



## Silmukoita ei aina tarvitse käyttää...

- Matlab on suunniteltu vektori- ja matriisioperaatioita varten.
- Vektorisointi: `for` ja `while` -silmukoiden korvaaminen vektori- tai matriisioperaatioilla.
  - Yleensä laskennallisesti tehokkaampaa kuin silmukoiden käyttäminen.

```
% Sinifunktion arvojen  
% laskenta for-silmukassa
```

```
i = 0;  
for t = 0:.01:10  
    i = i + 1;  
    y(i) = sin(t);  
end
```

```
% Vastaava operaatio  
% vektorimuodossa
```

```
t = 0:.01:10;  
y = sin(t);
```

# Funktiot

- m-tiedosto voi sisältää komentojonon sijasta **Funktion**
- Funktio ottaa sisään argumentteja ja palauttaa arvoja.
- Funktio tallentaa käyttämänsä muuttujat omaan sisäiseen työtilaan:
  - Matlabin työtilassa olevia muuttujia ei voida suoraan käyttää funktion sisällä.
  - Funktion sisäisiä muuttujia ei voida käyttää sen ulkopuolella.

## Funktioiden syntaksi

```
function [x, y] = minunfunktio(a, b, c) % Funktion määrittelyrivi
x = a*b+c;                               % Funktion toteutus
y = x.^2
```

- Yo. esimerkissä:
  - `minunfunktio` : funktion nimi.
  - `a`, `b`, `c` : funktion input-argumentit.
  - `x`, `y` : funktion palauttavat arvot.
  - Funktio tallennetaan saman nimiseen m-tiedostoon.
  - Esim. funktio `minunfunktio()` tallennettaisiin tiedostoon `minunfunktio.m`.
  - Funktion 1. rivi alkaa avainsanalla `function`: Kertoo Matlabille, että tiedosto on funktio eikä komentojonotiedosto.

## Ohjelman kehittäminen

- Monesti ongelma kannattaa pilkkoa itsenäisiin osiin.
    - Oma funktio erikseen eri osille.
  - Pseudo-koodi-luonnos ennen varsinaista koodia.
    - Pääpaino ohjelman rakenteessa eikä syntaksissa.
    - Helppo muuttaa tämän jälkeen varsinaiseksi ohjelmaksi.
  - Ohjelmointityyli
    - Käytä selkeitä funktioiden ja muuttujien nimiä.
    - Kommentoi koodiasi.
    - Käytä sisennyksiä ja tyhjää tilaa.
- ⇒ Virheiden etsintä selkeästä koodista on helpompaa.

## Tehtävä A: Asiakasdatan käyttö

Tehtävässä harjoitellaan for-loopin ja loogisten operaattoreiden käyttöä. Käytössäsi on dataa 10000:n henkilön terveystiedoista. Tehtävänäsi on etsiä datasta tietty ihmisryhmä, joka tulee kutsua tarkastukseen.

- Lataa kurssin MyCourses-sivuilta tiedosto asiakasdata.mat, vie se Matlab-kansioosi ja tallenna se Matlabin työtilaan.
- Matriisin sarakkeista löytyy henkilön sukupuoli (1=nainen, 0=mies), syntymävuosi (1940-2000), edellinen tarkastuskäynti sekä geneettisten riskitekijöiden gen1 (asteikko 0-7), gen2 (asteikko 0-5) ja gen3 (asteikko 0-10) lukuarvot.
- ✎ Mitä tietoja kullakin sarakkeelta löytyy? Vinkki: Voit katsoa matriisin A sarakkeen  $i$  maksimiarvon komennolla `max(A(:,i))`.

## Tehtävä A: Asiakasdatan käyttö

- Etsi asiakasdatasta niitä potilaita vastaavien rivien indeksejä, joilla täyttyvät seuraavat ehdot: Nainen, ennen 1970 syntynyt, gen1 yli 5 ja gen2 yli 3 ja viimeisin tarkastus ennen vuotta 2010.
  - Voit tehdä tämän esimerkiksi käyttäen for-silmukkaa, if-kontrollirakennetta ja kalvoissa aiemmin mainittuja loogisia- ja vertailuoperaattoreita.
- 📌 Kuinka monta nämä ehdot täyttävää asiakasta löydät?

## Tehtävä B: Alkulukupalautin

Kirjoita oma Matlab-funktio, joka ottaa argumentikseen luonnollisen luvun  $N$  ja palauttaa pienimmän alkuluvun, joka on suurempi kuin  $N$ . Esimerkiksi, jos  $N = 80$ , niin funktio palauttaa alkuluvun 83. Jos luku  $N$  on alkuluku, funktio palauttaa sen. (Voit olettaa, että syöte on positiivinen kokonaisluku.)

### Ohje:

1. Aloita funktiotiedosto näin: `function alkuluku = omafunktio(N)`. Muuttuja `alkuluku` on funktion palauttama alkuluku. Funktiota kutsutaan Matlabin komentoriviltä kirjoittamalla esim. `omafunktio(80)`, jossa siis  $N$ :lle on annettu arvo 80.

## Tehtävä B: Alkulukupalautin

2. Matlabin funktiolla `isprime` voit tarkistaa onko luku alkuluku. Jos tarkastettu luku ei ole alkuluku, niin kasvatata sen arvoa yhdellä kunnes löydät seuraavan alkuluvun. Käytä ohjelmasi toteutuksessa `while`-toistorakennetta.
- ✎ Mikä on opiskelijanumerostasi (ilman kirjainta) seuraava alkuluku?



## Tehtävä C: Infektiomalli

- Tarkastellaan infektion leviämistä kuvaavaa SIR-mallia. Populaatio voidaan jakaa kolmeen ryhmään: alttiit  $s(t)$  (s=susceptible), infektoituneet  $i(t)$  (i=infected) ja toipuneet  $r(t)$  (r=recovered). Ryhmien kokojen muutoksia voidaan kuvata seuraavasti:

$$\Delta s(t) = -\alpha \cdot i(t) \cdot s(t)$$

$$\Delta i(t) = \alpha \cdot i(t) \cdot s(t) - \beta \cdot i(t)$$

$$\Delta r(t) = \beta \cdot i(t)$$

## Tehtävä C: Infektiomalli

- Ryhmien koot ajanhetkellä  $t + 1$  saadaan laskettua seuraavilla differenssiyhtälöillä:

$$s(t + 1) = s(t) + \Delta s(t)$$

$$i(t + 1) = i(t) + \Delta i(t)$$

$$r(t + 1) = r(t) + \Delta r(t)$$

- Alussa tuhannen ihmisen populaatiossa on ainoastaan yksi infektoitunut ja muut ovat alttiita:

$$s(1) = 999$$

$$i(1) = 1$$

$$r(1) = 0.$$

## Tehtävä C: Infektiomalli




1. Mallinna infektion leviämistä parametreilla  $\alpha = 0.0011$  ja  $\beta = 0.03$ .  
Vihje: kirjoita malli m-tiedostoon käyttäen silmukkarakennetta `for t=1:100`. Laske ryhmien koot kullakin ajanhetkellä yllä annetuista differenssiyhtälöistä.  
✎ Millä ajanhetkellä infektoituneiden lukumäärä saavuttaa maksiminsa, ja mikä on lukumäärä silloin? Vihje: Selvitä komentoa `help max` käyttäen, kuinka saat vektorin `i` suurimman alkion ja sen indeksin.

## Tehtävä C: Infektiomalli

2. Infektion huomataan olevan tappava. Lisää malliin kuolleiden ryhmä  $d(t)$  (d=dead), jonka koko muuttuu yhtälön

$$\Delta d(t) = \gamma \cdot i(t)$$

mukaisesti. Tutki infektion leviämistä, kun  $\gamma = 0.0013$ .

-  Mikä differenssiyhtälö kuvaa infektoituneiden lukumäärän kehittymistä, kun malliin on lisätty kuolleiden ryhmä?
-  Kuinka suuri osa populaatiosta lopulta kuolee? Vertaa kuolevien osuutta parametrien  $\beta$  ja  $\gamma$  keskinäiseen suhteeseen  $\frac{\gamma}{\beta+\gamma}$ . Mitä havaitset? Mistä havaintosi voisi johtua?
-  Liitä vastauksiisi kuva eri ryhmien kehittymisestä ajan funktiona sekä ilman kuolleiden ryhmää, että niiden kanssa (käytä komentoa `subplot`). Lisää kuvaan myös tunnisteet käyrille (`legend`) ja anna akseleille nimet.

## Kotitehtävä: Kulutus-investointimallin simulointi

- Tehtävänäsi on simuloida kansantalouden kehitystä käyttäen yksinkertaista kulutus-investointimallia. Mallissa BKT ( $Y$ ) on kulutuksen ( $C$ ) ja investointien ( $I$ ) summa:  $Y(t) = C(t) + I(t)$
- Kulutus riippuu edellisen vuoden BKT:stä kaavan  $C(t) = \alpha Y(t - 1)$  mukaisesti, missä  $\alpha$  on marginaalinen alttius kuluttaa edellisen vuoden BKT.
- Investoinnit riippuvat kulutuksesta kaavan  $I(t) = \beta(C(t) - C(t - 1)) + \gamma$  mukaisesti, missä  $\beta$  kuvaa investointien riippuvuutta kulutuksen kasvusta ja  $\gamma$  on valtion vuosittainen investointi (oletetaan vakioksi).

## Kotitehtävä: Kulutus-investointimallin simulointi

1. Kirjoita `for`-silmukkaa käyttäen scriptti, jolla simuloit kansantalouden toimintaa 50 vuoden aikavälillä ( $t = 1, 2, \dots, 50$ ) seuraavilla lähtötiloilla ja parametrien arvoilla:
  - A.  $C(0) = 1 \text{ M€}$ ,  $I(0) = 1 \text{ M€}$ ,  $\alpha = 0.89$ ,  $\beta = 0.89$  ja  $\gamma = 0.5 \text{ M€}$ .
  - B.  $C(0) = 1 \text{ M€}$ ,  $I(0) = 1 \text{ M€}$ ,  $\alpha = 0.75$ ,  $\beta = 1.4$  ja  $\gamma = 0.5 \text{ M€}$ .
  - C.  $C(0) = 0.5 \text{ M€}$ ,  $I(0) = 0.5 \text{ M€}$ ,  $\alpha = 1.0$ ,  $\beta = 1.1$  ja  $\gamma = 0.3 \text{ M€}$ .
- 🗨️ Kuvaile lyhyesti talouden kehitystä jokaisessa skenaariossa. Miten skenaario A poikkeaa muista?

## Kotitehtävä: Kulutus-investointimallin simulointi

2. Matti Matemaatikko kertoo sinulle, että BKT:lle löytyy myös analyttinen ratkaisu

$$\hat{Y}(t) = k_1 r_1^t + k_2 r_2^t + c, \quad (1)$$




$$k_1 = -1.27 + 0.98i, \quad k_2 = -1.27 - 0.98i, \quad r_1 = 0.84 - 0.29i, \\ r_2 = 0.84 + 0.29i \quad \text{ja} \quad c = 4.55$$

- Kirjoita funktio, joka ottaa argumenttina vektorin vuosia  $(t_1, \dots, t_m)$  ja palauttaa vektorin  $(\hat{Y}(t_1), \dots, \hat{Y}(t_m))$ .

Älä käytä silmukkarakenteita vaan komentoa `.^` ! Esimerkiksi syöte `2.^[1,3]` antaa tulokseksi `[2,8]`.

- Huom. Voit käyttää Matlabissa kompleksilukuja aivan samoin kuin reaalityyppisiä lukuja. Luku sijoitetaan muuttujaan esim. `a=2-3i`.

## Kotitehtävä: Kulutus-investointimallin simulointi

-  Vertaa saamaasi tulosta 1A-kohdan simuloituihin tuloksiin. Onko Matti ratkaissut tehtävän oikein? (eli päteekö  $Y = \hat{Y}$ ?). Huom! varmista, että indeksointi molemmissa tuloksissa yhtenäinen (jos teit 1A kohdan ohjeiden mukaan, niin syötä funktiollesi argumentiksi vektori 0:50).
-  Liitä vastauksiisi kuvat (3 kpl) simuloinnin tuloksista käyttäen annettuja parametrejä. Jokaisessa kuvassa tulee näkyä BKT:n, investointien ja kulutuksen kehitys 50 vuoden aikana. Lisäksi A-kohdan kuvassa tulee olla analyttisellä kaavalla ratkaistut pisteet (vain pisteet, ei viivaa). Nimeä käyrät, akselit ja laita otsikoksi oma nimesi.
-  Liitä vastauksiisi kommentoitu lähdekoodi.