

# Luku 1

## Ohjeita ohjelmiston Scilab käyttöön

### 1.1 Ohjelmiston lataaminen

Ohjeet ohjelmiston lataamiseen Windows-koneelle.

- Mene verkko-osoitteeseen *www.scilab.org*.
- Valitse ruudun yläosassa oleva painike *Download Scilab*.
- Valitse käyttöjärjestelmäsi mukainen versio. Esimerkiksi Windows-koneella paina aliotsikon *For Windows 2003/XP/VISTA platforms* alla olevasta kohdasta *Scilab x.x.x installers for 32 bits platforms* linkkiä *Scilab x.x.x installer*.
- Paina *Tallenna (Save file)*.
- Nyt koneellesi latautuu varsinaisen ohjelmiston latausohjelma. Kun lataus on valmis, käynnistä ohjelma ja seuraa ohjeita. Latauksen pitäisi onnistua, jos joka kohtaan valitsee *next* ja lopuksi *Install*. Tämän jälkeen ohjelma on käyttövalmis.

#### 1.1.1 Merkinnät

Ohjelmiston Scilab version 5.1.1 komentorivi näyttää seuraavalta

-->

Symbolilla `//` merkitään kommentin alkamista. Kommentti on esimerkiksi komennon sanallinen selitys, joka ei vaikuta mitenkään komennon suorittamiseen. Kaikki mitä komentoriville kirjoitetaan kommenttimerkin jälkeen, jätetään huomioimatta komennon suorituksen aikana. Esimerkiksi komennossa

```
-->2 + 4 // 3+5
ans =
```

6.

lasketaan vain ensimmäinen summa, sillä jälkimmäinen on kirjoitettu kommenttimerkin jälkeen, eikä sitä siis huomioida.

Nuolinäppäimillä  $\uparrow$  ja  $\downarrow$  voi selata käytettyjä komentoja.

### 1.1.2 Lisätietoa tai apua ohjelman käyttämiseen

Lisätietoja tai apua tietyn komennon käyttämiseen saa käyttämällä komentoa *help 'komento'*.

### 1.1.3 Ohjelman sulkeminen

Ohjelma suljetaan käyttämällä komentoa *exit* tai komentoa *quit*.

### 1.1.4 Muuttujan luominen

Sijoitusoperaattorina Scilabissa on yhtäsuuruusmerkki (`=`). Luotaessa jokin muuttuja, jolle halutaan antaa jokin arvo voidaan kirjoittaa

```
muuttujan nimi = haluttu arvo,
```

jonka jälkeen *muuttujan nimeä* voidaan käyttää siihen talletetun arvon asemesta kaikissa sen jälkeen tulevissa komennoissa. Muuttujan arvoa on myös mahdollista muuttaa myöhemmissä komennoissa. Esimerkiksi seuraava komento luo muuttujan *luku*, johon talletetaan arvo 5.

```
-->luku = 5
luku =
```

5.

Huom! Pienillä ja suurilla kirjaimilla on merkitysero.

Kuten yllä olevassa esimerkissä näkyy, niin Scilab tulostaa ruudulle toimenpiteen tuloksen. Mikäli tätä tulostetta ei haluta näkyviin, voidaan komennon perään laittaa puolipiste (;). Tällöin ruudulle tulee näkyviin vain seuraava tyhjä komentorivi. Esimerkki tulostuksista:

```
-->luku = 4
```

```
luku =
```

```
4.
```

```
-->luku = 2;
```

```
-->luku
```

```
luku =
```

```
2.
```

Yhdellä komentorivillä on mahdollista luoda myös useita muuttujia kerralla. Tällöin voidaan käyttää erottimena joko pilkkua (,) tai puolipistettä (;). Jos muuttujan jälkeen tulee pilkku, niin ruudulle tulostuu muuttujan nimi ja arvo. Jos muuttujan jälkeen erottimena on puolipiste, niin muuttujan nimi ja arvo eivät tulostu ruudulle. Tulosteet voisivat näyttää esimerkiksi tältä:

```
-->a=4, b=7; c=16
```

```
a =
```

```
4.
```

```
c =
```

```
16.
```

```
-->b
```

```
b =
```

```
7.
```

### 1.1.5 Laskutoimitukset

Tavallisimmat laskutoimitukset *summa*, *erotus*, *tulo*, *osamäärä* ja *potenssiin korotus* voidaan suorittaa käyttämällä operaattoreita

+, -, \*, /, ^.

Huom! Kertomerkki on aina kirjoitettava näkyviin muuttujien ja lukujen väliin.

Useita eri operaatioita voidaan tehdä samalla komentorivillä. Tällöin on usein hyödyllistä käyttää sulkuja ( ja ) tai [ ja ] selkeyttämään, mikä operaatio halutaan suoritettavan missäkin vaiheessa. Sulkuja { ja } ei voi käyttää tähän tarkoitukseen.

```
-->1+2
```

```
ans =
```

3.

```
-->a= 3, b=1, c= a-b
```

```
a =
```

3.

```
b =
```

1.

```
c =
```

2.

```
-->d=a*c
```

```
d =
```

6.

```
-->(d/2)^2
```

```
ans =
```

9.

Jos viimeinen kohta olisi kirjoitettu ilman sulkuja, olisi saatu aivan toisenlainen tulos.

```
-->d/2^2
```

```
ans =
```

1.5

## 1.2 Kompleksiluvut

### 1.2.1 Kompleksilukujen syöttötapa

Kompleksiluvut voidaan syöttää Scilabille summamuodossa. Imaginaariyksikkö voidaan syöttää komennolla %i. Esimerkiksi

```
-->z=3+4*i, w=2+5*i
```

jolloin Scilabin tulostus näyttää tältä:

```
z =  
  
3. + 4.i  
w =  
  
2. + 5.i
```

### 1.2.2 Kompleksilukujen laskutoimitukset

Kompleksilukujen *summa*, *erotus*, *tulo* ja *osamäärä* voidaan laskea käyttämällä muuttujien välissä normaaleja operaattoreita +, -, \* ja /. Esimerkkejä laskutoimituksista:

```
-->z=5+5*i,w=-3+3*i
```

```
z =  
  
5. + 5.i  
w =  
  
- 3. + 3.i
```

```
-->z+w
```

```
ans =  
  
2. + 8.i
```

```
-->z-w
```

```
ans =  
  
8. + 2.i
```

```
-->z*w
```

```
ans =
- 30.

-->z/w
ans =
- 1.6666667i
```

Muita kompleksilukuihin liittyviä operaatiota:

**abs(z)** Laskee kompleksiluvun  $z$  itseisarvon.

**conj(z)** Palauttaa kompleksiluvun  $z$  liittoluvun.

**imag(z)** Palauttaa kompleksiluvun  $z$  imaginaariosan.

**real(z)** Palauttaa kompleksiluvun  $z$  reaaliosan.

Kompleksiluvun  $z$  argumentti (vaihekulma) voidaan laskea esimerkiksi komennolla **atan(imag(z),real(z))**.

## 1.3 Matriisit

### 1.3.1 Matriisin luominen

Scilabissa matriisi luodaan kirjoittamalla alkiot hakasulkujen väliin. Vaakarivin alkiot erotellaan pilkulla (,) ja seuraavan vaakarivin alkaminen ilmoitetaan puolipisteellä (;). Esimerkiksi matriisi

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

esitetään komentorivillä näin:

```
-->A=[1,2,3;4,5,6;7,8,9]
A =
```

```
1.    2.    3.
4.    5.    6.
7.    8.    9.
```

### 1.3.2 Vektorin luominen

Vektori luodaan vastaavalla tavalla kuin matriisi. Vektori

$$\mathbf{x} = \begin{bmatrix} 2 \\ 5 \\ -3 \end{bmatrix}$$

luodaan Scilabissa komennolla:

```
-->x=[2;5;-3];
```

Vektoria luodessa jokaisen alkion väliin tulee puolipiste (;).

Matriisista voidaan tarvittaessa poimia yksi rivi tai sarake, jolloin sitä käsitellään kuin vektoria. Esimerkiksi matriisista  $A$  saadaan poimittua toinen rivi komennolla:

```
-->A=[1,2,3;4,5,6;7,8,9]
```

```
A =
```

```
1.    2.    3.
4.    5.    6.
7.    8.    9.
```

```
-->a2=A(2,:)
```

```
a2 =
```

```
4.    5.    6.
```

Vastaavasti kolmannen sarakkeen saa poimittua komennolla:

```
-->a2=A(:,3)
```

```
a2 =
```

```
3.
6.
9.
```

### 1.3.3 Matriisien laskutoimitukset

Matriisien *summa*, *erotus* ja *tulo* voidaan laskea käyttämällä operaattoreita +, - ja \*. Jokaisessa näissä laskutoimituksissa Scilab huomioi luonnollisesti matriisien kertaluokat ja antaa virheilmoituksen, jos laskutoimitusta ei ole määritelty annetuille matriiseille.

```
-->A=[1,0;0,1]; B=[1,1;0,1];
```

```
-->A+B
ans =
```

```
2.    1.
0.    2.
```

Matriisin  $A$  transpoosista käytetään merkintää  $A^T$ , mutta Scilabissa transpoosi saadaan käyttämällä operaattoria  $'$ .

```
-->A=[1,1,1;0,1,0], B=A'
```

```
A =
```

```
1.    1.    1.
0.    1.    0.
```

```
B =
```

```
1.    0.
1.    1.
1.    0.
```

Matriisin

$$B = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

käänteismatriisi saadaan näin

```
-->B=[1,1;0,1]; inv(B)
```

```
ans =
```

```
1.   -1.
0.    1.
```

Käänteismatriisin avulla voidaan esimerkiksi ratkaista lineaarinen yhtälöryhmä  $A\mathbf{x} = \mathbf{b}$ , sillä

$$\begin{aligned} A^{-1}A\mathbf{x} &= A^{-1}\mathbf{b} \\ \Leftrightarrow I\mathbf{x} &= A^{-1}\mathbf{b} \\ \Leftrightarrow \mathbf{x} &= A^{-1}\mathbf{b}, \end{aligned}$$

jos käänteismatriisi  $A^{-1}$  on olemassa.

### Riviporrasmuoto

Pelkistettyyn riviporrasmuotoon matriisiin saa muutettua komennolla *rref*. Esimerkiksi matriisiin

$$A = \begin{bmatrix} 2 & -1 & -5 & -10 \\ -1 & 1 & 3 & 6 \\ -1 & -1 & 2 & 5 \end{bmatrix}$$

riviporrasmuoto saadaan komennolla:

```
-->A=[2, -1, -5, -10; -1, 1, 3, 6; -1, -1, 2, 5];
```

```
-->rref(A)
```

```
ans =
```

```
1.    0.    0.    2.
0.    1.    0.   -1.
0.    0.    1.    3.
```

### 1.3.4 Matriisin piirtäminen

Kaksiulotteisen kuvan saa Scilabilla tuotettua komennolla *plot(x)*, missä *x* on parametrina annettava vektori tai matriisi. Jos komennolle antaa parametrina kokonaisen matriisin, niin kuvaan piirtyy yhtä monta käyrää, kuin matriisissa on sarakkeita.

Faasitasokuva piirretään antamalla parametrina piirrettävät vektorit pilkuilla erotettuina.

Komennolla *figure* saa avattua uuden ikkunan kuvaa varten, jolloin useita kuvia voidaan tarkastella samanaikaisesti. Komennon käyttäminen estää myös uuden kuvan piirtämisen vanhan päälle.

## 1.4 Differentiaaliyhtälöt

Ohjelmistolla Scilab voidaan ratkaista muotoa

$$\frac{dx}{dt} = f(x, t), \quad x(t = t_0) = x_0$$

olevia differentiaaliyhtälöitä.

Tarkastellaan differentiaaliyhtälöä  $x'(t) = kx(t)(M - x(t))$ ,  $x(0) = x_0$ , missä  $k = 10^{-9}$ ,  $M = 10^{10}$  ja  $x(0) = 2$ .

Syötetään ensin funktio. Funktiolle annetaan nimeksi *xdot*, jossa muuttujina ovat *x* ja *t*.

```
-->//Ensin annetaan nimi funktiolle.
-->//Huom. muuttujien järjestys: ensin t, sitten x.
-->function xdot=f(t,x)
-->k=10^(-9); //Luodaan vakio k.
-->M=10^10; //Luodaan vakio M.
-->xdot(1)=k*x(1)*(M-x(1)); //Tässä on ratkaistava differentiaaliyhtälö.
-->endfunction
```

Seuraavaksi asetetaan alkuehto

```
-->x0=2; //Alkuehto.
```

ja sitten muuttujan  $t$  ratkaisujoukko. Tässä halutaan ratkaista differentiaaliyhtälö muuttujan  $t$  arvoilla  $0 \leq t \leq 5$ :

```
-->t=linspace(0,5); //Millä muuttujan t arvoilla yhtälö ratkaistaan.
```

Ratkaistaan differentiaaliyhtälö käyttämällä komentoa *ode*, jolle annetaan parametreiksi alkuehto, ajanhetki alussa, muuttujan  $t$  ratkaisujoukko ja ratkaistava funktio.

```
-->x=ode(x0,t(1),t,f); //Ratkaistaan yhtälö.
```

Lopuksi piirretään kuva ratkaisusta komennolla *plot*.

```
-->plot(t,x); //Piirretään vielä kuva ratkaisusta.
```

## 1.5 Joukko-operaatiot

Joukko luodaan ohjelmistolla Scilab kirjoittamalla hakasulkujen väliin joukon alkioit pilkulla erotettuna. Esimerkiksi joukot  $A = \{1, 2, 3, 4, 5\}$  ja  $B = \{2, 3, 5, 6, 7, 8\}$  syötetään näin:

```
-->A=[1,2,3,4,5], B=[2,3,5,6,7,8]
```

```
A =
```

```
1.    2.    3.    4.    5.
```

```
B =
```

```
2.    3.    5.    6.    7.    8.
```

Huom! Joukot syötetään ohjelmistoon Scilab kuten matriisi.

Kahden joukon unioni ( $A \cup B$ ) saadaan komennolla *union*.

```
-->union(A,B)
ans =
```

1. 2. 3. 4. 5. 6. 7. 8.

Kahden joukon leikkaus ( $A \cap B$ ) saadaan komennolla *intersect*.

```
-->intersect(A,B)
ans =
```

2. 3. 5.

Komento

```
-->setdiff(A,B)
ans =
```

1. 4.

antaa joukon  $B$  komplementin joukossa  $A$  ( $A \setminus B$ ).