

-e

## matlabteht/mlPerusteet, Matlab-perusteita

---

### 1. 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=[2 4 3;0 1 -1;3 5 7]`. 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 transpoimalla.
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)=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:  
`[A b]`  
(tai `[A, b]`). Jos yhtä monta saraketta, niin allekkain: `[A;B]`

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:  $u.(2*ones(size(u)))$ , joka tietysti toimii.)
12. Piirtämistä varten muodostetaan  $x$ -vektori, joka edustaa diskretoitua  $x$ -akselia ja laske-  
 taan 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  $^$  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`-komennolla. 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

## 2. 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 komendoista.

### 3. mlP002.tex

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

### 4. 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 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`

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

.

### 5. mlP004.tex

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  $x$  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:** Matlabille  $0^0 = 1$  (eikä NaN)

## 6. mlP005.tex

Muodosta vektori  $x$ , joka koostuu alkioista:

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

**Vihje:** Kahdessa viimeisessä kohdassa voit selkeyttää komentamalla:

`format rational`

Paluu oletusformaattiin: `format`

## 7. 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 edellä?
norm(x)
help norm
```

**Vihje:**

## 8. 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 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.

## 9. 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`   `B*A`   `A^2`   `A.^2`   `B^2`   `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.

## 10. 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

## 11. 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 (ovela). (Sarakkeittain jonoutus matriisille  $A$  saadaan näin: `A(:)`.)

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

## 12. 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.

Olko  $Y_{n \times k}$  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.

### 13. mlP012.tex

Taikaneliön saa komennolla **magic(n)**. Muodosta muutamalla pienehkällä  $n$ :n arvolla matriisin  $M = \text{magic}(n)$  rivisummat, sarakesummat, lävistäjäsomma ja sivulävistäjäsomma.

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

### 14. 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. Aiemmissa kurssimatskuissa, myös Solmu-kirjoituksessa, koetehtävissä on monia tehtäviä ratkaisuihin.

**15.** 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

**16.** mlP015.tex

Kun Newtonin menetelmää (opetetaan mahd. myöhemmin) sovelletaan yhtälöön  $x^2 - a = 0$ , saadaan iteraatiojono

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

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` -luoppia. Se toimii syntaksilla

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

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**

**17.** mlP016.tex

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

```
    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

### 18. 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  $\pi$  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!

### 19. 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  $v$  heittohetskellä on  $4m/s$

iv Pallon etenemisvektorin suunta  $\theta$  on  $45^\circ$

Kirjoita oletukset skriptiisi.

2. Luo vektori  $\mathbf{t}$ , 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  $\mathbf{x}$  ja  $\mathbf{y}$ .

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.

**Vihje:**

**20.** 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”

**21.** 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,...1000` ja sitten vain `sum`. Jälkimmäisessä voit laskea muutamalla, toinen toistaan suuremmalla arvolla. (Numeerisesti et tietenkään voi summata äärettömyyksiin.)

**22.** 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)`