

Matlab-perusteita, osa 1

Heikki Apiola

12. maaliskuuta 2012

Ohjelmahahmotelma

Matlab-
perusteita,
osa 1

Heikki Apiola

- 1. viikko: Matlab
- 2. viikko: Maple (+ annettujen Matlab tehtävien ratkaisuja)
- 3. viikko: Maple ja Matlab (lopputyöt)

Mikä on Matlab

- **Matriisilaboratorio** [Cleve Moler, Mathworks inc.]
 - Perustietorakenne: Matriisi/taulukko
- Numeerisen laskennan työskentely-ympäristö

```
>> quad(@(x) exp(-x.^2),0,1)
```

```
ans =
```

```
0.7468
```

- Suuri joukko matemaattisia ja muita funktioita, joita käyttäjä voi määritellä (ohjelmoida) lisää.
- Funktionaalinen ohjelmointikieli
- Sovellusalaakohtaisia työkalupakkeja, "toolbox".

- help
 - `>> help funnimi` antaa käyttöohjeen ja **doc**-linkin
Kokeile `>> help sin, doc sin`
- `>> doc funnimi` antaa kauniimman ja täydellisemmän tekstin.
- `>> doc + "Search tab "` \implies hakutoiminto

Alkutoimet

- Työtila, komentoikkuna
 - Matriisit ja muut tietoalkiot tallettavat muistiin **työtilaan**.
 - who, whos
- Komentoja (funktioita) sovelletaan työtilan muuttujiin.
 - Matlab tulkkaa ja palauttaa tuloksen (tai virheilmon) työtilaan.

- Työtila, komentoikkuna
 - Matriisit ja muut tietoalkiot tallettavat muistiin **työtilaan**.
 - `who`, `whos`
- Komentoja (funktioita) sovelletaan työtilan muuttujiin.
 - Matlab tulkkaa ja palauttaa tuloksen (tai virheilmon) työtilaan.

- 1 Käynnistä Matlab
- 2 Luo työhakemisto (joko File-valikko tai komento `mkdir` ^a)
- 3 Aseta nykyhakemisto työhakemistoksesi.
- 4 Luo muuttuja:
`>> x=5`
- 5 Suorita: `>> y=exp(x)`
- 6 Kokeile: `>> who`, `whos`

^a Jotkut Unix-komennot voidaan antaa Matlab-komentoikkunasta

Matriisit,vektorit, skalaarit

- Perustietorakenne: Matriisi, alkiot kompleksi(liuku)lukuja.
 - Sarakevektori: $(m,1)$ -matriisi
 - Rivivektori: $(1,n)$ -matriisi
 - Skalaari: $(1,1)$ -matriisi
 - Tyhjä: $(0,0)$ -matriisi
- Matriisin määrittely ja koko (size)

Kokeile:

```
>> A=[1 2 3 4 ;5 6 7 8; 9 10 11 12]
>> [m,n]=size(A)
>> v=-[1 2 3 4 ]
>> length(v)
>> 1:10
>> size(ans)    % ans viittaa aina edelliseen,
>>              % jota ei sijoitettu muuttujaan.
>> who
```

Matriisin osat ja kokoaminen

Jatka edellistä:

```
>> A=reshape(1:6,2,3),B=ones(2,2),C=diag(1:3)
>> [A B] % Vierekkain
>> [A;C] % Allekkain
```

A =

```
    1    3    5
    2    4    6
```

B =

```
    1    1
    1    1
```

C =

```
    1    0    0
    0    2    0
    0    0    3
```

```
>> [A B]
```

```
    1    3    5    1    1
    2    4    6    1    1
```

```
>> [A;C]
```

```
    1    3    5
    2    4    6
    1    0    0
    0    2    0
    0    0    3
```


Matriisi- ja taulukkoalgebraa

A, B matriiseja, koot yhteensopivat, c skalaari.

Matriisialgebraa

- $A + B$, $A+c$
- $A*B$ matriisitulo
- A' konjugoitu transpoosi
- $A.'$ transpoosi (ilman konjug.)
- A^p Matriisipotenssi (A neliömatr.)
- $A \setminus b$
 $Ax = b \iff x = A \setminus b$
(jos A kääntävä)

Taulukkoalgebraa

- $A + B$, $A+c$
- $A.*B$ Pisteittäinen tulo
- $A.^p$, $A.^B$ Pisteittäinen potenssi
- $A./B$, $c ./A$ Pisteittäinen jako. **Huom:**sudenkuopat!
- **Huom:** c/A on yleensä aivan muuta kuin käyttäjä tarkoittaa! Miksihän?

Skalaarifunktiot, alkiioittain toimivat

- Funktiot, jotka skalaariin sovellettuna antavat skaalaarituloksen, toimivat vektoreille ja matriiseille alkiioittain ("pisteittäin"). Näin toimivat yleensä kaikki matemaattiset funktiot.

```
>> t = [1 2 3];
```

```
>> y = exp(t)
```

on sama kuin $y = [\exp(1) \exp(2) \exp(3)]$

- Maple:ssa pitää vastaavassa tilanteessa käyttää map-operaattoria:

```
> map(f,t);
```

Versiosta 14 alkaen myös :

```
> t := [a,b,c];
```

```
> f~(t);
```

$[f(a), f(b), f(c)]$

Tässä f on **funktio** (tai määrittelemätön symboli), ei

Funktioita matriisien muodostamiseen

eye, vander, hilb, zeros, ones, rand, reshape, magic

Tehtävä: Suorita komennot

```
>> A = zeros(2,5)
>> B = ones(3)      % tai ones(3,3)
>> R = rand(3,2)
>> N = randn(3,2)
```

Miten eroavat rand ja randn? Mitä tapahtuu, jos toistat:

```
>> R = rand(3,2) (↑) komentoikkunassa tai CTR-ENTER
editorissa
```

Toista :

```
>> rand(state,0); R = rand(3,2)
```

Huvittele

komentamalla seuraavaan tyyliin:

```
>> mesh(ones(30));hold on;mesh(zeros(30));  
>> mesh(eye(30));shg  
>> mesh(hilb(30));shg  
>> surf(diag(-5:5))  
>> surf(magic(10))  
>> mesh(reshape(0:24,5,5))
```

Voit vaihdella esim. mesh → surf ja lukua 30 sekä figure, hold off, ja voit keksiä muita visualisoitavia matriiseja.

Näppää “rotate”-nuolta ja kääntele mm. tasokuvaksi.

Selvitä samalla, miten ko. matriisi rakentuu, erit.

reshape-tapaus.

Muuttujien nimet ja tyypit

- Muuttujia ei tarvitse alustaa eikä tyyppiä määritellä.
- Lukujen oletustyyppi on 64:n bitin liukuluku ("double"), n. 16 desimaalinumeroa.

```
>> 2.345
```
- Merkkien tyyppi on 16 bitin 'char'

```
>> 'a'
```
- Useimmat kohtaamamme muuttujat ovat edellisistä koostuvia matriiseja.
- Luvun muuttaminen merkiksi:

```
>> num2str(2.3)  
>> str2num(ans) % ja takaisin.
```
- Muita tyyppejä: logical, single, int-tyypit, help datatypes

Erikoismuuttujia, laskenta/näyttötarkkuus

Matlab-
perusteita,
osa 1

Heikki Apiola

`i, j, eps, realmin, realmax, NaN, Inf, ans` format

Esimerkkiskripti

Kirjoita tiedosto magicrank.m

```
% Investigate the rank of magic squares
r = zeros(1,32);
for n = 3:32
r(n) = rank(magic(n));
end
r
bar(r)
```

Suorita komentoikunassa
magicrank