

## 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 (↑) 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). Matriisiker-  
tolaskun 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 transponoimalla.
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 ko-  
mennolla `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 lasketaan 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`-komentolla. 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 (myos surf1, surfc, colorbar,...)  
>> contour(x,y,Z) % Korkeusk. piirros
```