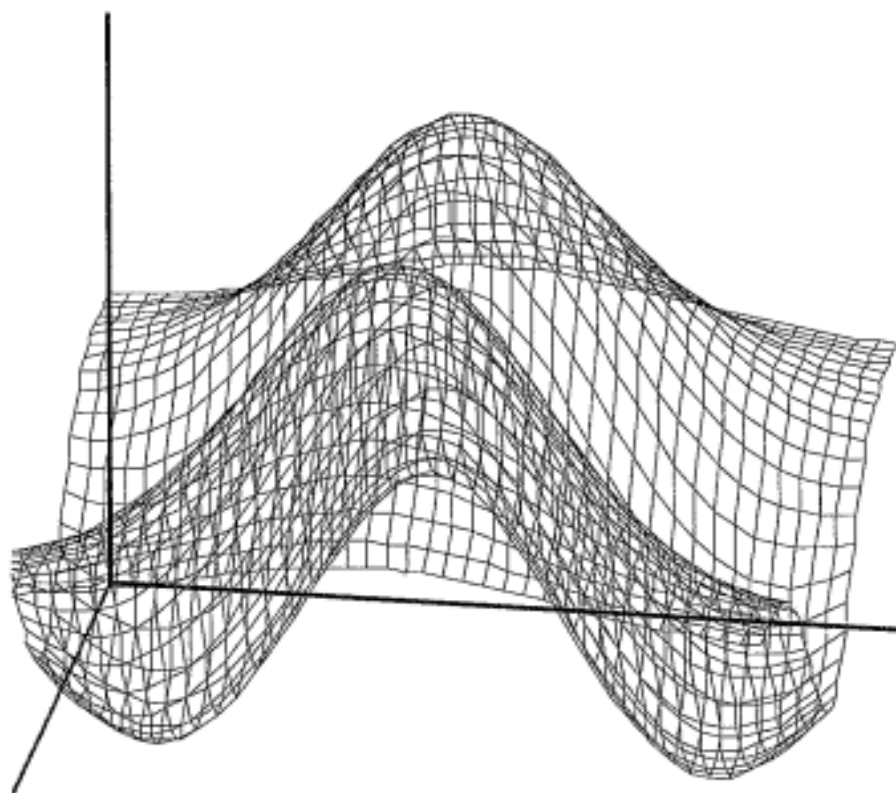


MATLAB

m-Tiedostot

Timo Mäkelä



6. M-TIEDOSTOT

Editorilla voi luoda **M-tiedostoja**, joiden tarkenteena on **.m**. M-tiedostot voivat olla *skriptejä* tai *funktioita*. Seuraava taulukko sisältää vertailun näiden välillä.

Skripti	Funktio
Ei sisäänmenoa eikä ulostuloa.	Voi olla sisäänmenoparametreja ja ulostuloparametreja.
Muuttujat työtilassa.	Muuttujat ovat funktion sisäisiä.
Käytetään automatisoimaan laskenta- askeleita, joita joutuu useasti suorittamaan.	Käytetään laajentamaan MATLABia omiin sovelluksiin.

Editori käynnistetään työkalurivin painikkeilla tai **File**-valikon komennoilla **New: M-file** tai **Open**. M-tiedoston sisällön voi tulostaa komennolla **type**.

Ohjelmointia tarkastellaan tarkemmin myöhemmin; tässä otetaan esille vain joitain perusasioita.

Yksinkertaisen funktion rakenne on seuraava:

```
function y = funktion_nimi(x)
Komentoja
y = funktion_määrävä_lauseke;
```

Edellisessä funktiossa on yksi sisäänmenoparametri x ja yksi ulostuloparametri y .

Skripti koostuu vain jonosta MATLAB-komentoja.

Esim. Määritellään funktio `heron(a,b,c)`, jolla on kolme parametria:

```
function A = heron(a,b,c)
% HERON Kolmion pinta-ala
% HERON(a,b,c) laskee kolmion pinta-alan, kun kolmion sivujen
% pituudet ovat a, b, c.

p = (a+b+c)/2;          % Kolmion piirin puolikas
A = sqrt(p*(p-a)*(p-b)*(p-c));
```

Funktioita käytetään seuraavasti:

```
>> heron(3,4,5)
ans =
     6
```

Skripteissä ja funktioissa voidaan tiedon tulostamiseen ja lukemiseen käyttää mm. seuraavia komentoja:

- **disp(A)**: Tulostaa matriisin A. Jos A on merkkijono, niin merkkijono tulostetaan.
- **n = input(text)**: Tulostaa näytölle merkkijonon text ja jää odottamaan käyttäjän syöttöä, joka tallennetaan muuttujaan n.

Esim. Seuraavassa on toteutettu satunnaiskävelyn simulointi skriptinä ja funktiona. Tee ohjelmat editorilla ja tallenna tiedostoihin. Tyhjennä MATLABin työtila Workspace komennolla **Edit: Clear Workspace** ennen skriptin ja funktion ajoa. Mitä eroa on työtilassa ajon jälkeen? Aja ohjelmia eri askelten määrillä.

Skripti: Tallenna skripti tiedostoon **Satunnaiskavely.m**. Voit ajaa skriptin käskyllä **Satunnaiskavely**.

```
% SATUNNAISKÄVELY
% Hiukkanen lähtee liikkeelle origosta ja liikkuu satunnaiseen
% suuntaan yhden yksikön pituisen matkan joka askeleella.

disp('Anna askelten lukumäärä')
n = input('n = ');

fii = 2*pi*rand(n,1);          % Satunnaisliikkeen suuntakulmat
x = cumsum(cos(fii));          % Satunnaisliikkeet x-akselin
y = cumsum(sin(fii));          % ja y-akselin suunnassa

clf;                            % Grafiikkaikkunan tyhjennys
plot(x,y);                       % Kulun piirto
axis equal;                       % Samat skaalat x- ja y- akselille

hold on;                          % Säilytetään kuva
plot(x(1),y(1),'ro',x(n),y(n),'ro'); % Merkitään alku ja loppu

axs = axis;                       % Haetaan akselien rajat
scale = axs(2)-axs(1);            % x-akselin pituus

text(x(1)+scale/30, y(1), 'Start'); % Kirjoitetaan Start alkuun ja
text(x(n)+scale/30, y(n), 'Finish'); % Finish loppuun.

hold off;                          % Säilytys pois päältä

xlabel('x'); ylabel('y');          % Nimetään akselit
title('Satunnaiskulku');           % Otsikko
```

Funktio: Tallenna funktio tiedostoon **RandomWalk.m**. Voit ajaa funktion käskyllä **RandomWalk(n)**, missä n on askelten lukumäärä.

```
function RandomWalk(n)
% RANDOMWALK Satunnaiskävely
% RandomWalk(n). Luku n ilmoittaa askelten lukumäärän.
% Hiukkanen lähtee liikkeelle origosta ja liikkuu n askelta
% satunnaiseen suuntaan. Askeleen pituus on yksi.

fii = 2*pi*rand(n,1);           % Satunnaisliikkeen suuntakulmat
x = cumsum(cos(fii));           % Satunnaisliikkeet x-akselin
y = cumsum(sin(fii));           % ja y-askelin suunnassa

Loppu sama kuin skriptissä.
```