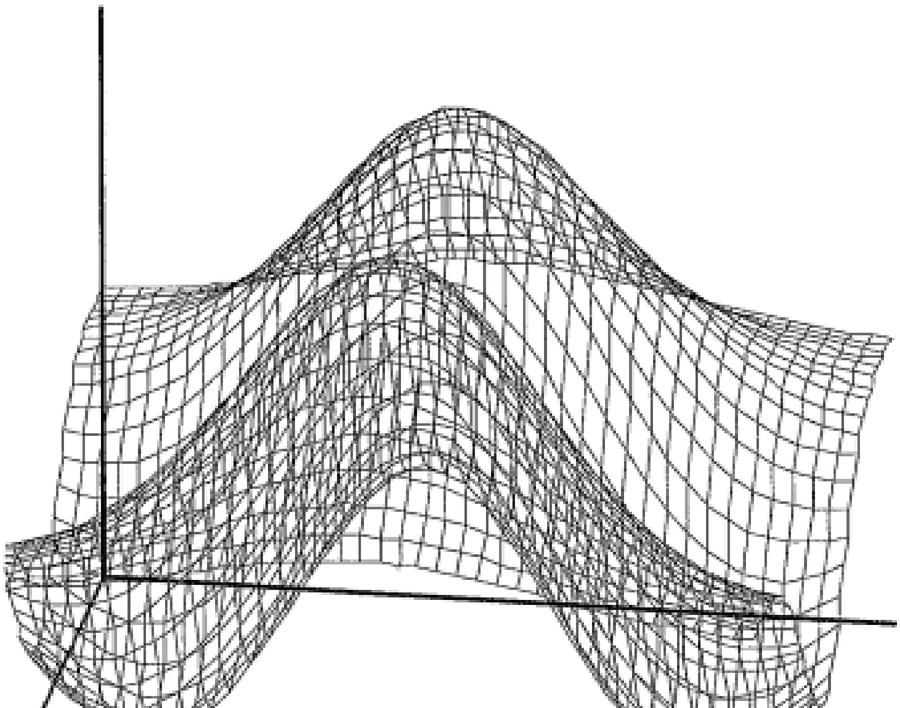
MATLAB

Yhden muuttujan funktiot

Timo Mäkelä



Tammertekniikka

7. YHDEN MUUTTUJAN FUNKTIOT

7.1 Yhtälön numeerinen ratkaiseminen

Tarkastellaan yhtälön

$$f(x)=0$$

ratkaisemista eli funktion $f(x)$ 0-kohtien määrittämistä.

MATLABilla yhtälö ratkaistaan funktiolla **fzero**, jolla on kaksi muotoa.

- Jos funktio sijaitsee tiedostossa f.m annetaan komento
  - **fzero(@f, x_0)**: ratkaisulle annetaan alkuarvo x_0
  - **fzero(@f, [a, b])**: ratkaisua etsitään väliltä [a, b].
- Jos funktio ei sijaitse tiedostossa annetaan komento
  - **fzero(inline('f(x)'), x_0)**: ratkaisulle annetaan alkuarvo x_0
  - **fzero(inline('f(x)'), [a, b])**: ratkaisua etsitään väliltä [a, b].

Edellä

- **@f** on tiedoston f.m otin (handle) ja
- **inline('f(x)')** tekee funktiosta inline-objektin.

Yleisempi yhtälö

$$f(x)=g(x)$$

voidaan saattaa muotoon yo. muotoon

$$f(x)-g(x)=0.$$

Yhtälöä

$$f(x)=g(x)$$

numeerisesti ratkaistaessa on usein syytä piirtää kuva tilanteesta. Tässä on kaksi tapaa:

- Piirretään funktion $y=f(x)-g(x)$ kuvaaja, josta selvitetään x-akselin leikkauspisteet. Tällöin on syytä piirtää samaan kuvaa 0-funktion kuvaaja.
- Piirretään funktioiden

$$y = f(x)$$
$$y = g(x)$$

kuvaajat ja määritetään kuvaajien leikkauspisteiden x-koordinaatit.

Kuvaajien leikkauspisteiden likimääräisessä määrittämisessä voi käyttää hyväksi **zoomausta**, joka suoritetaan kuvaikkunassa seuraavasti:

- Valitaan **Tools: Zoom in** ja rajataan hiirellä zoomattava alue
- Valitaan **Tools: Zoom out** ja klikkaillaan hiirellä kuvaa.

Funktion **fzero** alkuarvoa tai ratkaisuväliä muuttamalla saadaan yhtälön eri ratkaisujen likiarvot.

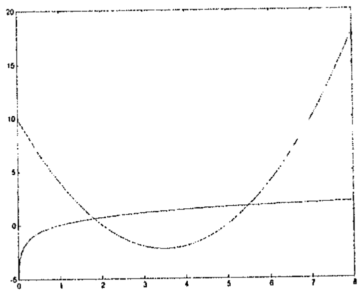
Esim. Ratkaistaan yhtälö

$$\ln x = x^2 - 7x + 10$$

Piirretään kuva.

```
>> x=0.01:.01:8;
>> f1 = log(x);
>> f2 = x.^2 -7*x +10;
>> plot(x,f1,x,f2)
```

Kokeile zoomausta!



Kuvasta nähdään, että yhtälöllä on kaksi juurta, joiden likiarvot ovat 2 ja 5,5. Määritetään nämä tarkemmin.

Tehdään ensin *funktioeditorilla* funktio

```
function y = fun1(x)
y = log(x)-(x.^2 - 7*x +10);
```

ja tallennetaan tämä nimellä fun1.m

Määrätään 0-kohdat numeerisesti:

```
>> x1 = fzero(@fun1,2)
x1 =
    1.8133
>> x2 = fzero(@fun1,5.5)
x2 =
    5.4881
```

Tarkistetaan tulokset

```
>> fun1(x1)
ans =
  -7.7716e-016
>> fun1(x2)
ans =
  -4.4409e-016
```

Luvut ovat hyvin lähellä nollaa.

TEHTÄVIÄ

Ratkaise yhtälöt

1. $\sin x = \frac{x}{3} - 0{,}2$

2. $e^{-x^2} = 2x - x^2$

3. $\frac{x}{4} - \cos x = 0$

7.2 Minimointi ja maksimointi

MATLABissa funktion $f(x)$ minimikohta välillä $[a,b]$ määritetään seuraavasti:

- Jos funktio sijaitsee tiedostossa f.m annetaan komento

  fminbnd(@f, a, b),

- Jos funktio ei sijaitse tiedostossa annetaan komento

  fminbnd(inline('f(x)'), a, b),

Koska

funktion $f(x)$ *maksimikohta on funktion* $-f(x)$ *minimikohta,*

voidaan maksimikohta välillä $[a,b]$ määritetään seuraavasti:

fminbnd(inline('-f(x)'), a, b).

Funktio **fminbnd** ei välttämättä löydä globaalia minikohtaa. On aina hyvä piirtää funktio $y = f(x)$ kuvaaja.

Funktiolla **fminbnd** on laajempi muoto:

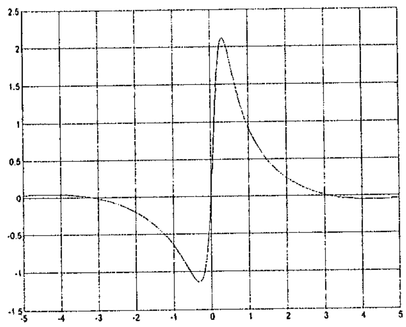
[x,fval,exitflag,output] = fminbnd(fun,a,b,asetukset).

Parametrit ovat samat kuin funktiolla **fminsearch**. Nämä parametrit onkin esitelty kappaleessa, jossa käsitellään usean muuttujan funktion lokaalin minimikohdan määrittämistä.

Esim. Määritetään funktion

$$f(x) = \frac{\sin x}{(x-0{,}1)^2 + 0{,}1}$$

suurin ja pienin arvo välillä $[-5,5]$.



Muodostetaan M-funktio

```
function y = fun2(x)
y =sin(x)./ ((x-.1).^2 + .1);
```

Määritetään minikohta.

```
>> xmin = fminbnd(@fun2,-5,5)
xmin =
   -0.3174
```

ja maksimikohta

```
>> xmax = fminbnd(inline('-fun2(x)'),-5,5)
xmax =
    0.3236
```

Funktion minimi- ja maksimiarvot ovat siten

```
>> fun2(xmin)
ans =
   -1.1381
>> fun2(xmax)
ans =
    2.1199
```

Piirretään vielä kuva tilanteesta.

```
>> x=linspace(-5,5,1000);
>> plot(x,fun2(x))
```

Lisätään kuvaan ruudukko

```
>> grid on
```

TEHTÄVIÄ

Määritä seuraavien funktioiden suurin ja pienin arvo annetulla välillä

1. $f(x)=\cos(2x)-2\sin(x)-1,\quad x\in[0,2\pi]$
2. $f(x)=\dfrac{10x}{x^2+4},\quad x\in\mathbf{R}$
3. $f(x)=\dfrac{x^2}{x^2+x+1},\quad x\in[-5,5]$

7.3 Numeerinen integrointi

Yhden muuttujan funktion integraalin numeerisen arvon laskentaan on MATLABissä käytettävissä funktiot quad, quadl ja trapz.

Integraali

$$\int_a^b f(x)dx$$

lasketaan

- funktioilla quad komennolla (vastaava funktiolle quadl)

  quad(fun, a, b)

  missä

  - fun = @f, jos funktio on tiedostossa f.m
  - fun = inline('f(x)'), jos funktio ei ole tiedostossa.

  Integraali lasketaan tarkkuudella 10^{-6}. Jos halutaan laskea tarkemmin käytetään komentoa

  quad(fun, a, b, tol),

  missä **tol** ilmoittaa toleranssin.

- funktioilla trapz komennolla

  trapz(x, y)

  missä

  - vektori **x** sisältää muuttujan x arvot välillä **a ...b**
  - vektori **y** sisältää vastaavat funktion $f(x)$ arvot.

Funktion quadl laskee tarkemmin kuin quad. Funktiota trapz voidaan käyttää myös, kun funktion arvot tiedetään vain erillisissä pisteissä. Vektorin **x** alkioiden ei tarvitse olla tasavälisesti.

Esim. Lasketaan integraali

$$\int_0^1 e^{-x^2}\,dx$$

```
>> format long
```

Funktio quad:

```
>> quad(inline('exp(-x.^2)'),0,1)
ans =
   0.74682413322961
```

Funktio trapz:

```
>> x=0:0.01:1;
>> y = exp(-x.^2);
>> trapz(x,y)
ans =
   0.74681800146797
```

Yleensä quad antaa tarkemman tuloksen.

```
>> format short
```

7.4 Numeerinen derivointi

Funktion $f(x)$ derivaatta määritellään raja-arvona

$$f'(x) = \lim_{h\to 0}\frac{f(x+h)-f(x)}{h}.$$

Täten derivaattaa voidaan approksimoida erotusosamäärällä

$$f'(x) \approx \frac{f(x+h)-f(x)}{h}, \qquad (1)$$

kun $|h|$ on pieni. Myös

$$f'(x) = \lim_{h\to 0}\frac{f(x+h)-f(x-h)}{2h},$$

joten

$$f'(x) \approx \frac{f(x+h)-f(x-h)}{2h}, \qquad (2)$$

kun $|h|$ on pieni. Tämä symmetrinen erotusosamäärä on kertaluokkaa parempi likiarvo kuin epäsymmetrinen erotusosamäärä.

Funktion $f(x)$ toiselle derivaatalle voidaan johtaa seuraava likiarvo

$$f''(x) \approx \frac{f(x+h)-2f(x)+f(x-h)}{h^2}, \qquad (3)$$

kun $|h|$ on pieni.

Jos funktion $f(x)$ arvot on taulukoita tasavälisesti vaakavektorina (tai pystyvektorina) muodossa

$$F = [f(x) \quad f(x+h) \quad \cdots \quad f(x+nh)],$$

voidaan derivaatta laskea komennolla

DF = gradient(F,h),

missä

- **F** on yo. muotoa oleva vektori

- **h** on muuttujan x muutos siirryttäessä alkiosta toiselle.

Tuloksena saadaan vektori

- **DF**, jossa on likiarvot derivaatoilla $f'(x)$.

Vektori **DF** on samaa kertalukua kuin vektori **F**. Vektorin **DF** alkiot ovat seuraavan vektorin likiarvot:

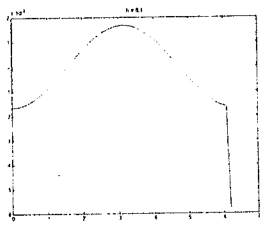
$$DF \approx [f'(x) \quad f'(x+h) \quad \cdots \quad f'(x+nh)].$$

Esim. Tutkitaan derivointia tarkastelemalla funktiota $f(x) = \sin x$, jonka derivaatta on $f'(x) = \cos x$.

Verrataan numeerista derivaattaa tarkkaan derivaattaan jaksovälillä $[0, 2\pi]$. Piirretään näiden erotuksen kuvaaja. Lasketaan kahdella askelpituudella.

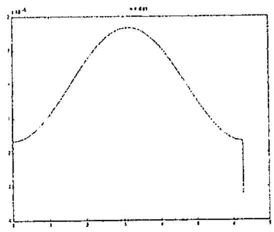
Askelpituus **h** = 0,1:

```
>> x = 0:0.1:2*pi;
>> y = sin(x);
>> dy = gradient(y,0.1);
>> dy1 = cos(x);
>> plot(x,dy-dy1)
>> title('h = 0,1')
```



Askelpituus **h** = 0,01:

```
>> x = 0:0.01:2*pi;
>> y = sin(x);
>> dy = gradient(y,0.01);
>> dy1 = cos(x);
>> plot(x,dy-dy1)
>> title('h = 0,01')
```



Havaitaan, että pienempi askelpituus **h** antaa paremman likiarvon derivaatalle.

Numeeriseen derivointiin liittyviä esimerkkejä on myös seuraavissa kappaleissa.