Numeeriset menetelmät

ALGORITMIT
MATEMATIIKASSA, MAA12

Kun esimerkiksi yhtälön tarkka ratkaisu ei ole oleellinen/tarpeellinen tai yhtälöä ei osata tarkasti ratkaista, käytetään hyödyksi *numeerisia menetelmiä*, joilla päästään *haluttuun tarkkuuteen*. Myös derivointia ja integrointia voidaan suorittaa numeerisesti. Tarkastellaan aluksi yhtälön ratkaisemiseen liittyviä numeerisia menetelmiä:

- haarukointi eli puolitusmenetelmä
- Newtonin menetelmä (Newton-Rhapson)
- Kiintopistemenetelmä ja iterointi.

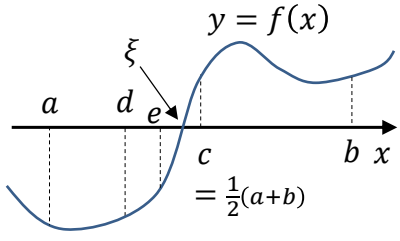
Toisinaan saatetaan edellä mainittuja menetelmiä yhdistää ja/tai tehdä ns. esitutkimusta mitä numeerista menetelmää kannattaa käyttää.

Tärkeään rooliin nousee myös käytetyn menetelmän virheen arviointi ja ***vaikutus arvoihin***, kts. Tietok.harj.1_teht3. Aluksi numeeriset arvot tarkentuvat, mutta jostakin lähtötilanteesta alkaen arvot eivät enää tarkennu, vaan ne saattavat jopa hajaantua (syy: pyöristysvirheet).
Pieni virhe lähtötilanteessa → suuri virhe loppuarvossa.

Haarukointi eli puolitusmenetelmä

Bolzanon lause sanoo, että jos f on jatkuva suljetulla välillä $[a, b]$ ja f saa välin $[a, b]$ päätepisteissä erimerkkiset arvot, niin f:llä on ainakin yksi nollakohta välillä $]a, b[$. Jos lisäksi f on aidosti monot. avoimella välillä $]a, b[$, niin f:llä on täsmälleen yksi nollakohta $\xi \in]a, b[$.
Sanotaan, että kohdat a ja b ovat nollakohdan ξ ala- ja ylälikiarvot. Kysymys kuuluu, kuinka voidaan parantaa näitä likiarvoja?

Valitaan välin keskikohta $c = \frac{b+a}{2}$ ja lasketaan funktion f arvo tässä kohdassa. Jos $f(a)f(c) < 0$ eli f:n arvot ovat erimerkkiset, niin $\xi \in]a, c[$. Muutoin joko $f(c) = 0$ (ja nollakohta on löydetty) tai $f(a)f(c) > 0$ ja $\xi \in]b, c[$.
Näin jatketaan, löydetään kohta d, kohta e ja virhe puolittuu joka kerta.



Näin suoritetun puolittamisen n. vaiheen jälkeen absoluuttinen virhe on

$$\frac{1}{2^n}(b+a)\,.$$

Esimerkki **a)** Osoita, että funktiolla $f(x) = x + \ln x$ on täsmälleen yksi nollakohta.
b) Ohjelmoi laskin haarukoimaan se kuuden desimaalin tarkkuudella.

a) Funktio on määritelty, jatkuva ja derivoituva, kun $x > 0$. Derivaatta

$$f'(x) = 1 + \frac{1}{x} > 0, \qquad \forall x > 0.$$

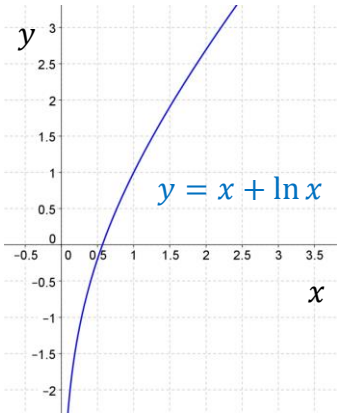
Lisäksi

$$f(0{,}5) \approx -0{,}19 < 0$$

ja

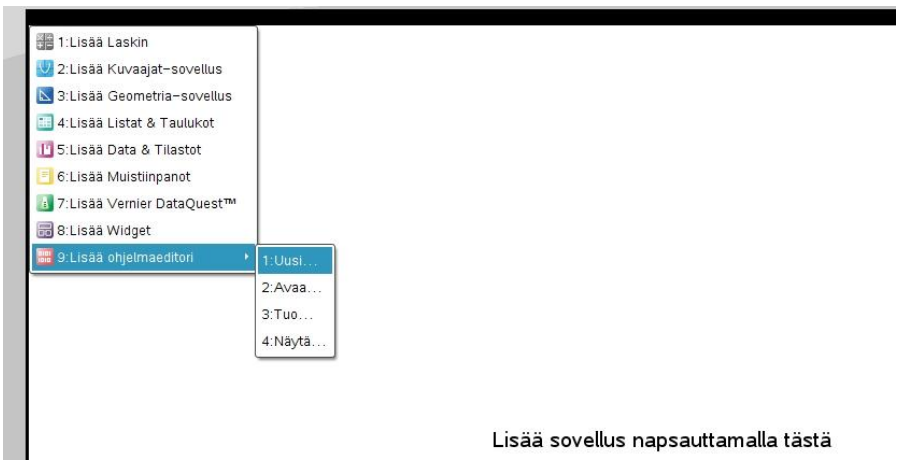
$$f(0{,}6) \approx 0{,}09 > 0,$$

joten Bolzano + aito monot. antavat täsmälleen yhden nollakohdan $\xi \in]0{,}5;\,0{,}6[$.



b) Ohjelmointitila löytyy TI-nspirestä (älkää vielä aloittako, tehdään yhdessä):

- Valitse ensin **Uusi TI-NSpire™-asiakirja → 9.Lisää ohjelmaeditori → 1: Uusi**



Seuraavanlainen ohjelmointi pitäisi saada aikaiseksi, ohjelman nimenä on HAAR (haarukointi), *TI-nspiressä hieman erilainen koodi*:

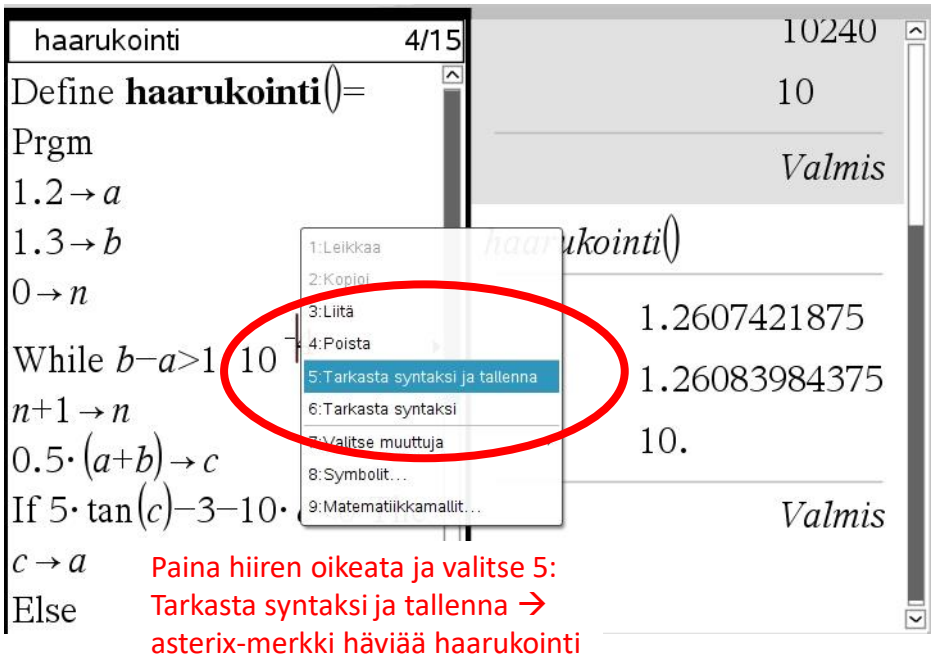
```
PROGRAM:HAAR
:0.5→A
:0.6→B
:0→N
:While B-A>1E-7
:N+1 →N
:0.5*(A+B) →C
:If C+ln C<0
:Then
:C →A
:Else
:C →B
:End
:End
:Disp A
:Disp B
:Disp N
```

Ohjelmoidaan tämä nyt yhdessä laskimiin
→ hyödynnetään ohjeita.

Ohjelmakoodin aluksi kerrotaan ala- ja ylälikiarvot sekä tallennetaan ne muistipaikkoihin A ja B. Sitten 3. rivillä perustetaan kierroslaskuri, joka nollataan. (Tämä ei ole tehtävän kannalta välttämätöntä, mutta on kiinnostavaa tietää kierrosten lukumäärä.)
Tämän jälkeen luodaan **While**-silmukka, joka pitää päättää **End** –komentoon (13. rivi), muuten tulee syntaksivirhe. Silmukka suoritetaan, niin kauan kunnes etsityn nollakohdan ylä- ja alalikiarvojen erotus saavuttaa tai alittaa arvon 10^{-7}. **While**-silmukan sisällä tapahtuu seuraavaa:

1. Viidennellä rivillä kasvatetaan laskuria yhdellä.
2. 6.nnella rivillä lasketaan ylä- ja alalikiarvojen keskiarvo ja tallennetaan se muistipaikkaan C.
3. 7.nnellä rivillä alkaa algoritmin kolmosvaihetta vastaava **If** – **Then** – **Else**–lause, joka päättyy 12.rivin **End**-komentoon. Funktio f voitaisiin ennen ohjelman kirjoittamista määritellä y1=x+lnx, jolloin **If** – lauseen ensimmäinen komento olisi y1(C)<0.

Lopuksi tulostetaan (Disp-komento) arvot A ja B sekä kierrosten lukumäärä N. (Lopuksi kannattanee tarkistaa, eli **Solve(x+lnx=0,x)** , OK).



Paina hiiren oikeata ja valitse 5: Tarkasta syntaksi ja tallenna → asterix-merkki häviää haarukointi sanan ylh.vasem. puolelta.

Haarukoinnin eli puolitus-menetelmän algoritminen kaaviokuva.