

## LUENTO 3 26.9.

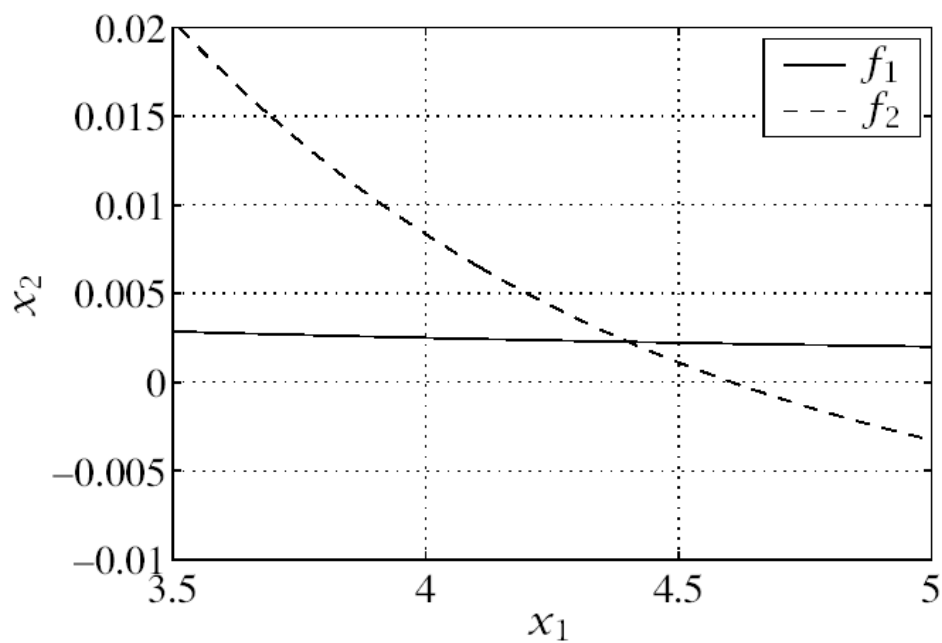
[Haa02 ] luku 6

### Epälineaariset yhtälöt ja yhtälöryhmät

**Esimerkki:** etsi ratkaisu yhtälöparille

$$\begin{cases} f_1(\mathbf{x}) = 100x_1x_2 - 1 = 0, \\ f_2(\mathbf{x}) = e^{-x_1} + e^{-x_2} - 1.01 = 0. \end{cases}$$

### Graafinen ratkaisu

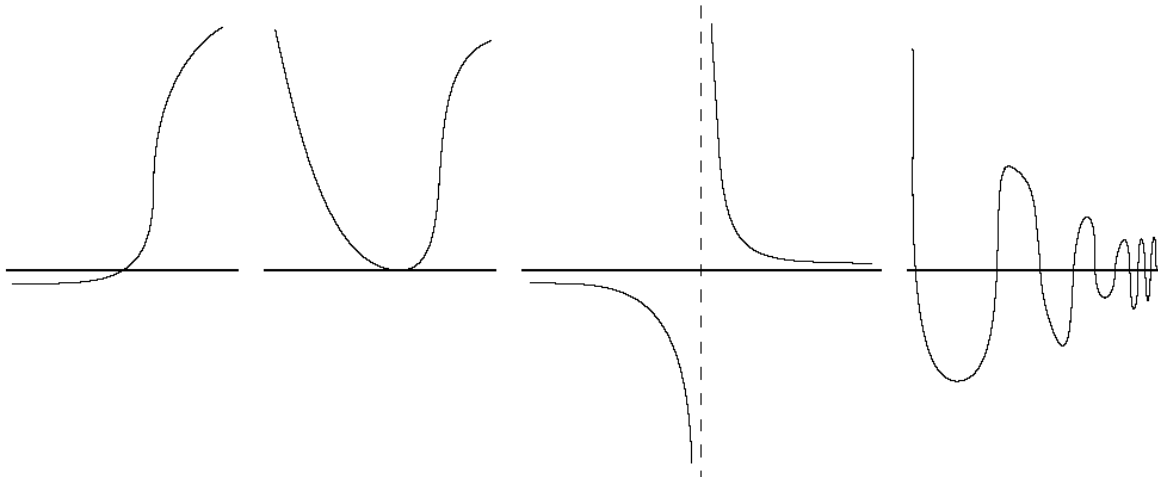


**Kuva 6.1:** Funktioiden nollakäyrät  $f_1(\mathbf{x}) = 0$  ja  $f_2(\mathbf{x}) = 0$ .

$$\mathbf{x}^* = (4.40061, 0.00227)^T$$

### Käytännön näkökohtia:

- jos funktio on ”hyvin käyttäytyvä” välillä  $[a,b]$  ja  $f(a)$ ,  $f(b)$  ovat erimerkkiset, on välillä ainakin yksi nollakohta
- funktion käyttäytyminen nollakohdan lähellä voi aiheuttaa numeerisia ongelmia (konetarkkuus)
- singulariteetti ei ole nollakohta!
- singulaarinen funktio (esim.  $f(x) = 1/(x-a)$ ) vaihtaa merkkiä singulariteetin  $x=a$  ympärillä → algoritmiongelma
- ei yleispätevää menetelmää epälineaarisen yhtälön tai yhtälöryhmän ratkaisemiseksi
- tällä kurssilla tutustumme puolituslukuun ja Newtonin menetelmään



**Kuva 6.2:** Erilaisia mahdollisuuksia skalaarifunktion käyttäytymiselle. Jatkuvalle ja välin päätepisteissä erimerkkisellä funktiolla on ainakin yksi nollakohta. Toisaalta funktion ei tarvitse vaihtaa merkkiä nollakohdan ympäristössä. Lisäksi singulariteetit voivat aiheuttaa ongelmia numeerisille algoritmeille.

$$e^x + 5 = 0$$

$$e^{-x} - x = 0$$

$$x^2 - 4 \sin x = 0$$

$$x^3 + 2x^2 - 5x - 6 = 0$$

$$\cos x = 0$$

ei ratkaisua

yksi ratkaisu

kaksi ratkaisua

kolme ratkaisua

äärettömän monta ratkaisua.

## Puolitushaku

1. Etsi pisteet  $a_1, b_1 : a_1 < b_1$  siten, että arvot  $f(a_1)$  ja  $f(b_1)$  ovat erimerkkiset. Funktion  $f$  jatkuvuuden perusteella väliltä  $[a_1, b_1]$  löytyy nollakohta.
2. Laske funktion arvo välin  $[a_k, b_k]$  puolivälissä  $m = (a_k + b_k)/2$ . Jos arvot  $f(m)$  ja  $f(a_k)$  ovat erimerkkiset, tutkimme seuraavaksi väliä  $[a_k, m]$ , muuten väliä  $[m, b_k]$ .
3. Lopeta iterointi, kun saavutat riittävän tarkkuuden.

■ **Esimerkki 6.3.1** Ratkaisemme esimerkin 6.1.2 yhtälön (6.4) käyttäen puolitushakua. Tätä varten muutamme yhtälön muotoon

$$f(\lambda) = x_0 e^\lambda + \frac{c}{\lambda} (e^\lambda - 1) - x_1 = 0.$$

Tehtävänä on hakea funktion  $f(\lambda)$  nollakohta. Annamme vakioille seuraavat arvot:  $x_0 = 0$ ,  $x_1 = 10$  ja  $c = 1$ . Koska  $f(3) < 0$  ja  $f(4) > 0$  ja funktio  $f$  on jatkuva, löytyy väliltä  $[3, 4]$  nollakohta.

Kun teemme 10 puolitushaun iteraatiota, saamme seuraavat tulokset:

| $k$ | $m$    | $f(m)$  |
|-----|--------|---------|
| 1   | 3.5000 | -0.8242 |
| 2   | 3.7500 | 1.0723  |
| 3   | 3.6250 | 0.0758  |
| 4   | 3.5625 | -0.3856 |
| 5   | 3.5938 | -0.1579 |
| 6   | 3.6094 | -0.0418 |
| 7   | 3.6172 | 0.0168  |
| 8   | 3.6133 | -0.0125 |
| 9   | 3.6152 | 0.0021  |
| 10  | 3.6143 | -0.0052 |

Siten löysimme nollakohdalle arvion  $\lambda^* \approx 3.61$ .

## Newtonin menetelmä (myös Newton-Raphson)

- perustuu Taylorin sarjakehitelmän antamaan arvioon funktion arvolle pisteen  $x$  ympäristössä (oletus:  $x$  riittävän lähellä nollakohtaa)
  - $f(x+\delta) = f(x) + (df(x)/dx)\delta + (1/2)(d^2f(x)/dx^2)\delta^2 + O(\delta^3)$
  - asetetaan  $f(x+\delta) = 0$  ja otetaan huomioon vain 1 kl. termi
- hakuaskel  $\delta \approx -f(x) / (df(x)/dx)$
- tarvitaan hyvä alkuarvaus  $x_1$  !

alla olevassa algoritmissa  $d = \delta$

### Algoritmi 6.3.1 (Newtonin menetelmä)

hae nollakohdan  $x^*$  sijainnille rajat:  $x^* \in [a, b]$

$$x_1 = (a + b) / 2$$

$$k = 1$$

**repeat**

$$d = \frac{f(x_k)}{f'(x_k)}$$

$$x_{k+1} = x_k - d$$

**if**  $x_{k+1} \notin [a, b]$

virhe: hypättiin ulos hakuvälistä

**exit**

**end**

$$k = k + 1$$

**until**  $|d| < \epsilon$

---

■ **Esimerkki 6.3.3** Ratkaisemme esimerkin 6.3.1 tehtävän  $f(\lambda) = 0$  käyttäen Newtonin menetelmää. Jos alkuarvauksena on  $\lambda = 3.5$ , saamme tulokseksi seuraavat iteraatiopisteet:

| $k$ | $\lambda_k$ | $f(\lambda_k)$ |
|-----|-------------|----------------|
| 1   | 3.5000      | -0.8242        |
| 2   | 3.6205      | 0.0417         |
| 3   | 3.6150      | 0.0001         |
| 4   | 3.6150      | 0.0000         |

Siten Newtonin menetelmä suppeni neljässä iteraatiossa ja saimme nollakohdalle arvion  $\lambda^* \approx 3.6150$ .

---



---

■ **Esimerkki 6.3.2** Newtonin menetelmää voi käyttää neliöjuuren tai kuutiojuuren laskemiseen. Kokeilemme laskea arvon  $x = \sqrt[3]{5}$  eli ratkaisemme funktion  $f(x) = x^3 - 5$  nollakohdan. Saamme derivaataksi  $f'(x) = 3x^2$  ja iteraatioksi

$$x_{k+1} = x_k - \frac{x_k^3 - 5}{3x_k^2}.$$

Alkuarvauksella  $x_1 = 1$  saamme iteraatiopisteet

| $k$ | $x_k$  | $f(x_k)$ |
|-----|--------|----------|
| 1   | 1.0000 | -4.0000  |
| 2   | 2.3333 | 7.7037   |
| 3   | 1.8617 | 1.4523   |
| 4   | 1.7220 | 0.1062   |
| 5   | 1.7101 | 0.0007   |
| 6   | 1.7100 | 0.0000   |

Siten saamme arvion  $\sqrt[3]{5} \approx 1.7100$ .

---

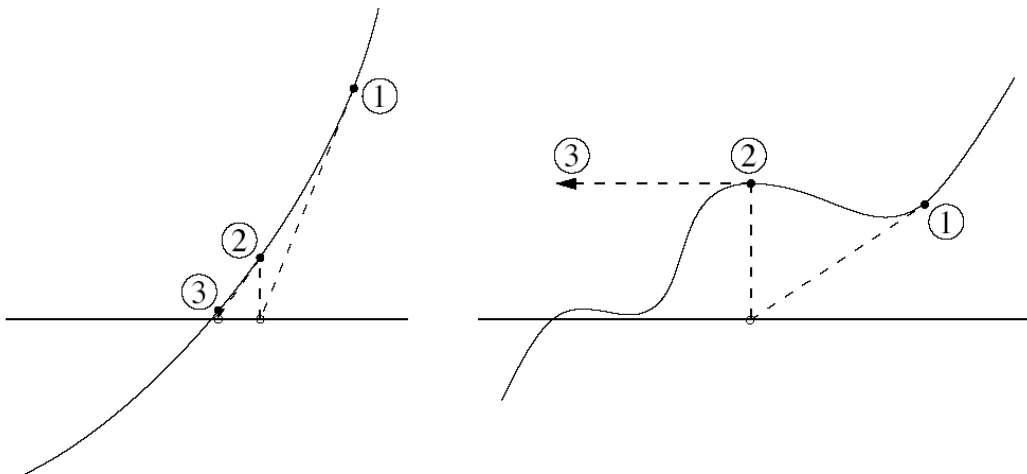
**Derivaatan laskeminen pisteessä x:**

- analyttisesti (jos tunnetaan)
- differenssilikiarvo

$$df(x_k)/dx = (f(x_k + h) - f(x_k)) / h$$

**Newtonin algoritmi voi ”karata käsistä” (kuva 6.4. o.p.)**

**Ratkaisu kannattaa tarkistaa esim. käyttämällä puolituslakua**

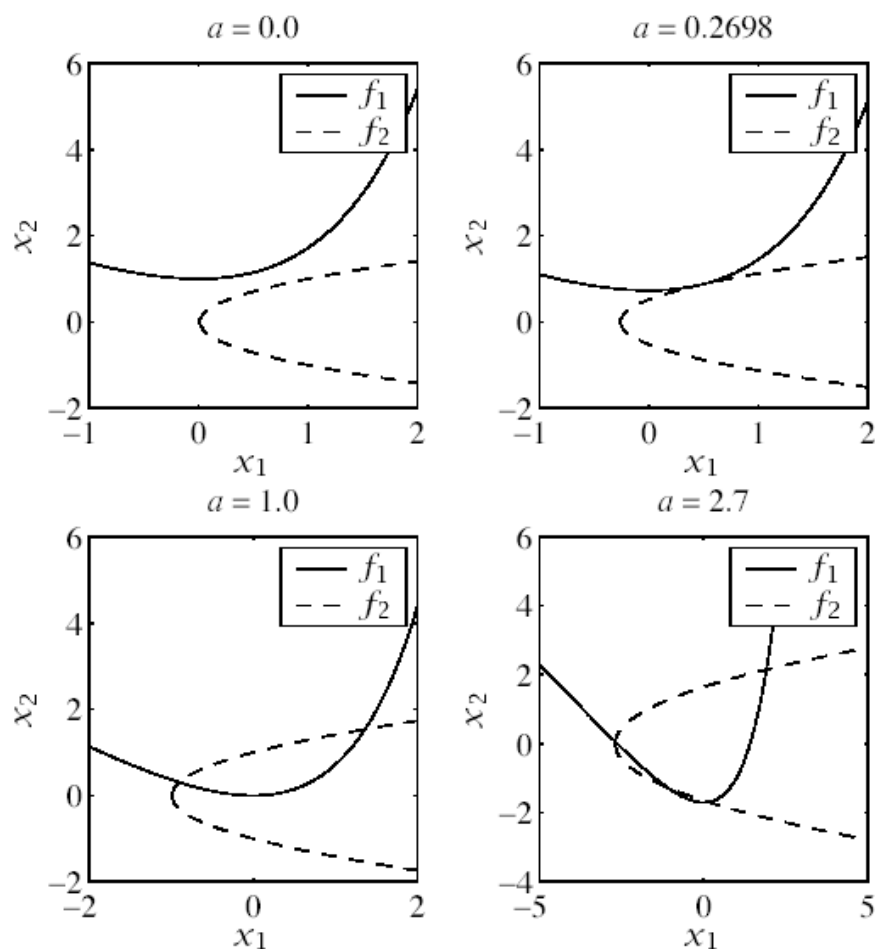


**Kuva 6.4:** Newtonin menetelmän toiminta kahdessa eri tapauksessa. Vasemmalla algoritmi suppenee hyvin nopeasti. Oikealla törmätään derivaatan nollakohtaan ja joudutaan kauas funktion nollakohdasta.

■ **Esimerkki 6.4.1** Seuraavassa on yksinkertainen esimerkki epälineaarista yhtälöryhmästä:

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} e^{x_1} - x_1 - x_2 - a \\ x_1 - x_2^2 + a \end{pmatrix} = \mathbf{0}. \quad (6.7)$$

Yhtälöryhmällä on 0, 1, 2, 3 tai 4 ratkaisua riippuen vakion  $a$  arvosta. Funktioiden nollakäyrät vakion  $a$  neljällä eri arvolla on esitetty kuvassa 6.7.



**Kuva 6.7:** Yhtälöryhmän (6.7) nollakäyrät muutamalla vakion  $a$  eri arvolla.

## Newtonin menetelmä epälineaariselle yhtälöryhmälle

### Taylorin sarja

$$\mathbf{f}(\mathbf{x} + \mathbf{s}) \approx \mathbf{f}(\mathbf{x}) + J(\mathbf{x}) \mathbf{s}.$$

Tässä vektorin  $\mathbf{s}$  kertoimena on Jacobin matriisi  $J_{ij} = \partial f_i / \partial x_j$ . Asettamalla  $\mathbf{f}(\mathbf{x} + \mathbf{s}) = \mathbf{0}$  saamme ratkaistavaksi yhtälöryhmän

$$J(\mathbf{x}) \mathbf{s} = -\mathbf{f}(\mathbf{x}). \quad (6.8)$$

(lineaarinen yhtälöryhmä vektorille  $\mathbf{s}$ )

### Algoritmi 6.4.1 (Newtonin menetelmä)

tee alkuarvaus  $\mathbf{x}^1 \in \mathbb{R}^n$

$k = 1$

**repeat**

ratkaise askel  $\mathbf{s}^k$  yhtälöryhmästä  $J(\mathbf{x}^k) \mathbf{s}^k = -\mathbf{f}(\mathbf{x}^k)$

$\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{s}^k$

$k = k + 1$

**until**  $\|\mathbf{s}^k\| < \epsilon$  tai  $k > k_{\max}$

■ **Esimerkki 6.4.2** Ratkaisemme Newtonin menetelmällä yhtälöryhmän

$$\begin{cases} e^{x_1} - x_1 - x_2 - 1 = 0, \\ x_1 - x_2^2 + 1 = 0. \end{cases}$$

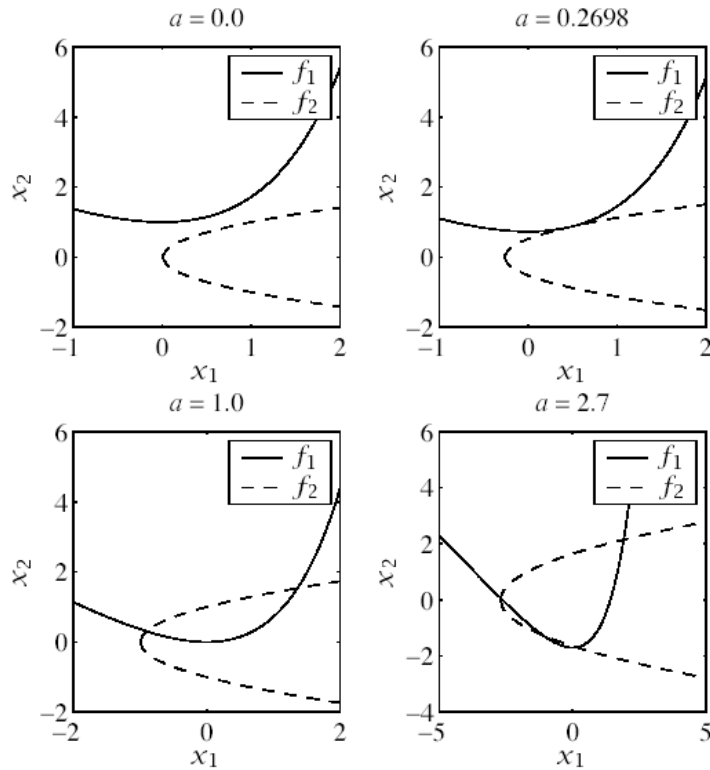
Tehtävän Jacobin matriisi on

$$J(\mathbf{x}) = \begin{pmatrix} e^{x_1} - 1 & -1 \\ 1 & -2x_2 \end{pmatrix}.$$

Alkuarvauksella  $\mathbf{x}^1 = (1, 1)^T$  saamme seuraavat iteraatit:

| $k$ | $x_1^k$ | $x_2^k$ | $f_1(\mathbf{x}^k)$    | $f_2(\mathbf{x}^k)$     |
|-----|---------|---------|------------------------|-------------------------|
| 1   | 1.0000  | 1.0000  | -0.2817                | 1.0000                  |
| 2   | 1.6417  | 1.8208  | 0.7012                 | -0.6738                 |
| 3   | 1.4138  | 1.5732  | 0.1244                 | -0.0613                 |
| 4   | 1.3623  | 1.5374  | 0.0054                 | -0.0013                 |
| 5   | 1.3600  | 1.5362  | $0.9800 \cdot 10^{-5}$ | $-0.1316 \cdot 10^{-5}$ |

Siten iteraatio suppeni viidellä askeleella arvoon  $\mathbf{x}^* \approx (1.360, 1.536)^T$ . Tehtävällä on myös toinen ratkaisu  $\mathbf{x}^* \approx (-0.904, 0.309)^T$ , joka löytyy käyttämällä tämän ratkaisupisteen lähellä olevaa alkuarvausta.



**Kuva 6.7:** Yhtälöryhmän (6.7) nollakäyrät muutamalla vakion  $a$  eri arvolla.

## **Yhteenveto**

- puolitushaku on yksinkertainen ja varma menetelmä, tarvitaan järkevä ratkaisun haun aloitusväli  $[a,b]$
- Newtonin menetelmä yleensä tehokkaampi, tarvitaan hyvä alkuarvaus  $x^1$
- ratkaisu kannattaa varmistaa useammalla kuin yhdellä menetelmällä
- käytössä myös MATLABin optimization toolbox:  
    fsolve ('f',x) (Levenberg-Marquardt tai Gauss-Newton), ks.  
    [Haa02] sivu 355-356
- graafinen tarkastelu auttaa aloitusvälin tai alkuarvauksen etsimisessä ja ratkaisun tarkistamisessa
- yhtälöiden takana olevan fysiikan ymmärtäminen voi auttaa esim. ratkaisujen lukumäärää mietittäessä