

Aalto-yliopisto
Perustieteiden korkeakoulu
Teknillisen fysiikan ja matematiikan tutkinto-ohjelma

Epälineaaristen pienimmän neliösumman
tehtävien ratkaiseminen numeerisilla
optimointimenetelmillä

kandidaatintyö
18.12.2014

Lari Pelkola

Työn saa tallentaa ja julkistaa Aalto-yliopiston avoimilla verkkosivuilla.
Muilta osin kaikki oikeudet pidätetään.

AALTO-YLIOPISTO PERUSTIETEIDEN KORKEAKOULU PL 11000, 00076 Aalto http://www.aalto.fi		KANDIDAATINTYÖN TIIVISTELMÄ	
Tekijä: Lari Pelkola			
Työn nimi: Epälineaaristen pienimmän neliösumman tehtävien ratkaiseminen numeerisilla optimointimenetelmillä			
Tutkinto-ohjelma: Teknillisen fysiikan ja matematiikan tutkinto-ohjelma			
Pääaine: Systeemitieteet		Pääaineen koodi: F3010	
Vastuupettaja(t): Prof. Harri Ehtamo			
Ohjaaja(t): Prof. Harri Ehtamo			
<p>Tiivistelmä:</p> <p>Työn tavoitteena on tutustua epälineariin pienimmän neliösumman tehtäviin. Niitä esiintyy useimmiten datansovituksessa, jossa ideana on sovittaa optimoitavien parametrien suhteen epälineaarinen malli dataan. Epälineaaristen PNS-tehtävien optimaalisia parametreja ei pystytä ratkaisemaan suljetussa muodossa, mutta ne saadaan erilaisilla numeerisilla menetelmillä.</p> <p>Työssä esitellään kolme algoritmia PNS-tehtävien ratkaisemiseen, joista yksi on yleispätevä menetelmä optimoinnissa, eli Kvasi-Newton (BFGS), sekä nimenomaan epälineaarisiin PNS-tehtäviin tarkoitetut menetelmät Gauss-Newton ja Levenberg-Marquardt. Menetelmiä testataan kolmella erilaisella esimerkillä. Tarkastelu keskittyi menetelmien vaatimiin iteraatioiden määriin, alkuarvauksen vaikutukseen niiden suppenemiseen sekä hieman menetelmien robustisuuteen.</p> <p>Esimerkeistä kävi ilmi, että menetelmien suoritus riippuu hyvin paljon tehtävästä, joten algoritmeja ei ole helppo pistää yleispätevästi paremmuusjärjestykseen. BFGS ja Levenberg-Marquardt osoittautuivat yleisesti ottaen luotettavammiksi kuin Gauss-Newton, joka taas joissain tilanteissa voi olla kuitenkin nopein. BFGS oli yleensä hitain, koska sitä ei ole modifioitu PNS-tehtäviä varten. Iteraatioiden lukumäärä oli kaikilla menetelmillä yleensä pienempi alkuarvauksen ollessa lähellä optimia.</p>			
Päivämäärä: 18.12.2014		Kieli: suomi	Sivumäärä: 23
Avainsanat: Epälineaarinen, PNS, datansovitus, parametri, numeerinen menetelmä, algoritmi, kvasi-Newton, BFGS, Gauss-Newton, Levenberg-Marquardt, iteraatio, suppeneminen, robustisuus, optimi			

Sisältö

1	Johdanto	1
2	Epälineaariset pienimmän neliösumman tehtävät	3
2.1	Pienimmän neliösumman tehtävä optimointitehtävänä	3
2.2	Ratkaisemisen lähestymistavat	4
2.2.1	Lineaarinen malli	5
2.2.2	Kvadraattinen malli	5
3	Ratkaisumenetelmät	7
3.1	Kvasi-Newton menetelmä	8
3.1.1	Kvasi-Newton menetelmät PNS-tehtävissä	10
3.2	Gauss-Newton menetelmä	11
3.3	Levenberg-Marquardt menetelmä	12
4	Menetelmien testaaminen	14
4.1	Yksinkertainen sinifunktio	15
4.2	Gaussin funktio	18
4.3	Kolmiulotteinen esimerkki	21
5	Yhteenveto	23

1 Johdanto

Optimointitehtävien ratkaiseminen on monien tieteen alojen keskeisiä ongelmia. Kun halutaan minimoida tai maksimoida jotakin, muodostetaan optimointimalli. Esimerkiksi taloudessa halutaan minimoida sijoituksen riskiä ja maksimoida tuottoa. Teollisuudessa voidaan taas olla kiinnostuneita energian kulutuksen minimoimisesta ja tuotannon maksimoimisesta. Erilaisia optimointiongelmiä löytyy siis hyvin monilta osa-alueilta, ja niitä myös ratkotaan hyvin erilaisilla menetelmillä kyseisestä optimointiluokasta riippuen.

Yleinen tehtäväluokka optimoinnissa, erityisesti tilastollisten tehtävien yhteydessä, on pienimmän neliösumman tehtävä (PNS), mikä on tämän työn aiheena. PNS-tehtävässä optimoitava kohdefunktio ilmaistaan neliöiden summana, mistä seuraa erityisiä huomioon otettavia seikkoja muihin optimointiluokkiin nähden. Tyypillinen PNS-tehtävä on matemaattisen mallin sovittaminen dataan, jossa haetaan mallin tuntemattomille parametreille optimaaliset arvot dataan nähden. Tunnetuin ja yksinkertaisin tapaus tästä on lineaarinen regressio, missä sovitetaan suora dataan. Lineaarisen regressio parametrit pystytään ratkaisemaan melko pitkälle analyttisesti. Tämä työ rajataan kuitenkin epälineaarisiin PNS-tehtäviin, joiden ratkaisemiseen tarvitaan numeerisia optimointimenetelmiä. Työssä tutustutaan näistä kolmeen: yleisestikin epälineaarisisessa optimoinnissa käytetty kvasi-Newton menetelmä, sekä nimenomaan epälineaarisiin PNS-tehtäviin tarkoitettut Gauss-Newton ja Levenberg-Marquardt menetelmät. Epälineaarinen PNS-tehtävä on siis yleinen erikoistapaus rajoittamattomasta minimointitehtävästä, jossa kohdefunktio on epälineaarinen optimoitavien parametrien suhteen.

Optimointitehtävään liittyy läheisesti kohdefunktion minimin käsite:

Määritelmä 1.1 (Paikallinen minimi). Olkoon $f : \mathbb{R}^n \mapsto \mathbb{R}$. Piste $\mathbf{x}^* \in \mathbb{R}^n$ on funktion f paikallinen minimi, jos $\exists \epsilon > 0$ siten, että

$$f(\mathbf{x}^*) \leq f(\mathbf{x}), \text{ kun } \|\mathbf{x} - \mathbf{x}^*\| < \epsilon, \forall \mathbf{x} \in \mathbb{R}^n. \quad (1.0.1)$$

□

Minimipisteen pitää toteuttaa myös seuraavat ehdot:

Määritelmä 1.2 (Välttämätön ehto paikalliselle minimille). Jos \mathbf{x}^* on funktion $f(\mathbf{x})$ paikallinen minimi, pätee

$$\nabla f(\mathbf{x}^*) = \vec{\mathbf{0}}. \quad (1.0.2)$$

□

Yllä $\vec{\mathbf{0}}$ on \mathbb{R}^n nollavektori ja $\nabla = [\frac{\partial}{\partial x_1}, \dots, \frac{\partial}{\partial x_n}]^T$ on n-komponenttinen gradienttivektori. Gradientin nollakohta takaa sen, että kyseessä on funktion lokaali ääriarvokohta tai satulapiste. Näitä kutsutaan funktion f kriittisiksi pisteiksi.

Määritelmä 1.3 (Riittävä ehto paikalliselle minimille). Oletetaan, että f on kahdesti jatkuvasti derivoituva pisteen \mathbf{x}^* ympäristössä. Jos \mathbf{x}^* on f :n kriittinen piste sekä $\nabla^2 f(\mathbf{x}^*)$ on positiivisesti definiitti, on \mathbf{x}^* funktion f paikallinen minimikohta. □

Merkintä $\nabla^2 f(\mathbf{x})$ tarkoittaa f :n toisista osittaisderivaatoista muodostuvaa symmetristä Hessian matriisia pisteessä \mathbf{x} . Hessian matriisin positiividefiniittisyys takaa sen, että kyseessä todella on minimikohta. Todistukset sivuutetaan, mutta ne löytyvät esimerkiksi lähteestä [1] sivuilta 5-6. Matriisin positiividefiniittisyyden määritelmiä ja tarkempaa tietoa löytyy lähteestä [2].

Luvussa 2 tutustutaan epälineaarisiin PNS-tehtäviin sekä ratkaisemisen lähestymistapoihin yleisellä tasolla, ja luvussa 3 itse algoritmeihin. Luvussa 4 testataan, kuinka hyvin kukin algoritmi selviää muutamasta esimerkkitehtävästä. Kaikki työssä olevat esimerkit ja kuvat on tehty Matlab-ohjelmistolla. Työkaluina on käytetty Matlabin Optimization Toolboxin solvereita `fminunc` [3], `lsqnonlin` [4] ja `lsqcurvefit` [5].

2 Epälineaariset pienimmän neliösumman tehtävät

Tässä luvussa esitellään epälineaarinen PNS-tehtävä optimointitehtävänä ja datan sovitus yleisellä tasolla. Luvussa on käytetty lähteinä pääosin kirjoja Optimointitehtävien ratkaiseminen (Juha Haataja) [6], sekä Numerical Methods for Unconstrained Optimization and Nonlinear Equations (J.E. Dennis JR ja Robert B. Schnabel) [7].

2.1 Pienimmän neliösumman tehtävä optimointitehtävänä

Epälineaarinen pienimmän neliösumman tehtävä on yleismuodoltaan seuraava:

Määritelmä 2.1 (Pienimmän neliösumman tehtävä). Etsi $\mathbf{x}^* \in \mathbb{R}^n$, joka minimoi funktion

$$f(\mathbf{x}) = \|\mathbf{r}(\mathbf{x})\|^2 = \sum_{i=1}^m (r_i(\mathbf{x}))^2, \quad (2.1.1)$$

□

missä $m \geq n$, $\|\cdot\|$ on Euklidinen normi ja $\mathbf{r}(\mathbf{x}) : \mathbb{R}^n \mapsto \mathbb{R}^m$ on epälineaarinen parametrien $\mathbf{x} \in \mathbb{R}^n$ suhteen.

Yleisimmin epälineaarisia PNS-tehtäviä esiintyy datansovituksessa, jossa pyritään sovittamaan \mathbf{x} :n suhteen epälineaarinen malli $a(t, \mathbf{x})$ datapisteisiin (t_i, y_i) , $i = 1, \dots, m$ nähden siten, että arvojen y_i ja $a(t_i, \mathbf{x})$ välisten erotusten, eli residuaalien $r_i(\mathbf{x}) = a(t_i, \mathbf{x}) - y_i$ neliösumma minimoituu. Tällöin

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) = \sum_{i=1}^m (a(t_i, \mathbf{x}) - y_i)^2. \quad (2.1.2)$$

Tehtävän 2.1.1 rakenteen vuoksi on syytä kiinnittää erityistä huomiota gradienttiin ja Hessen matriisiin, jotka poikkeavat tavallisesta rajoittamattomasta minimointitehtävästä.

Funktion $f(\mathbf{x})$ gradientti on $n \times 1$ pystyvektori:

$$\nabla f(\mathbf{x}) = 2 \sum_{i=1}^m r_i(\mathbf{x}) \nabla r_i(\mathbf{x}) = 2\mathbf{J}(\mathbf{x})^T \mathbf{r}(\mathbf{x}), \quad (2.1.3)$$

missä $\mathbf{J}(\mathbf{x})$ on residuaalivektorin $\mathbf{r}(\mathbf{x}) = (r_1(\mathbf{x}), r_2(\mathbf{x}), \dots, r_m(\mathbf{x}))^T$ Jacobin matriisi:

$$\mathbf{J}(\mathbf{x}) = \begin{pmatrix} \nabla^T r_1(\mathbf{x}) \\ \vdots \\ \nabla^T r_m(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \frac{\partial r_1(\mathbf{x})}{\partial x_1} & \dots & \frac{\partial r_1(\mathbf{x})}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial r_m(\mathbf{x})}{\partial x_1} & \dots & \frac{\partial r_m(\mathbf{x})}{\partial x_n} \end{pmatrix}. \quad (2.1.4)$$

Funktion $f(\mathbf{x})$ Hessen matriisi on

$$\mathbf{H}(\mathbf{x}) = 2 \sum_{i=1}^m (\nabla r_i(\mathbf{x}) \cdot \nabla r_i(\mathbf{x})^T + r_i(\mathbf{x}) \cdot \nabla^2 r_i(\mathbf{x})) \quad (2.1.5)$$

$$= 2\mathbf{J}(\mathbf{x})^T \mathbf{J}(\mathbf{x}) + 2\mathbf{S}(\mathbf{x}), \quad (2.1.6)$$

missä matriisi \mathbf{S}

$$\mathbf{S}(\mathbf{x}) = \sum_{i=1}^m r_i(\mathbf{x}) \nabla^2 r_i(\mathbf{x}) \quad (2.1.7)$$

kuvaa $\mathbf{H}(\mathbf{x})$:n toisen kertaluvun informaatiota eli kaarevuutta, ja osuus $\mathbf{J}(\mathbf{x})^T \mathbf{J}(\mathbf{x})$ kuvaa ensimmäisen kertaluvun informaatiota.

Epälineaarisissa PNS-tehtävissä keskeisiä termejä ovat myös pienen ja suuren residuaalin ongelmat. Pienen residuaalin tehtävä tarkoittaa, että optimipisteessä $\|f(\mathbf{x}^*)\| \approx 0$ ja $\|\mathbf{r}(\mathbf{x}_k)\| \rightarrow 0$, kun $\mathbf{x}_k \rightarrow \mathbf{x}^*$. Muuten saadaan suuren residuaalin tehtävä. Mikäli residuaalit ovat nolla, tarkoittaa se datan sovituksessa sitä, että malli $a(t_i, \mathbf{x})$ osuu tismalleen yhteen arvojen y_i kanssa jokaisessa datapisteessä.

2.2 Ratkaisemisen lähestymistavat

Tehtävän 2.1.1 ratkaisemiseen on kaksi eri lähestymistapaa, jotka esitellään tässä aluvuossa yleisellä tasolla. Lähestymistavat liittyvät toki läheisesti toisiinsa, kuten tullaan huomaamaan. Alalukujen teoria on pääosin Åke Björckin kirjasta *Numerical Methods for Least Squares Problems* [8].

2.2.1 Lineaarinen malli

Tämä lähestymistapa voidaan ajatella seuraavan epälineaaristen yhtälöryhmien ratkaisemisesta $\mathbf{r}(\mathbf{x}) = \vec{\mathbf{0}}$. Jo tästä voi päätellä, että tämä lähestymistapa toimii paremmin pienen residuaalin tehtävissä. Tästä lähtökohdasta on luonnollista approksimoida residuaalifunktiota $\mathbf{r}(\mathbf{x})$ lineaarisella mallilla.

Määritelmä 2.2 (Lineaarinen approksimaatio). Residuaalifunktion $\mathbf{r}(\mathbf{x}) : \mathbb{R}^n \mapsto \mathbb{R}^m$ lineaarinen approksimaatio $\mathbf{h}(\mathbf{x})$ pisteen \mathbf{x}_k läheisyydessä on

$$\mathbf{h}(\mathbf{x}) := \mathbf{r}(\mathbf{x}_k) + \mathbf{J}(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k). \quad (2.2.1)$$

□

Koska tavoitteena on saada residuaalifunktio mahdollisimman pieneksi ja lähelle nollaa, saadaan tästä lineaarinen pienimmän neliösumman tehtävä

$$\min_{\mathbf{x}} \|\mathbf{r}(\mathbf{x}_k) + \mathbf{J}(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k)\|^2, \quad (2.2.2)$$

joka antaa estimaatin tehtävän 2.1.1 ratkaisuksi. Tällainen lähestymistapa käyttää siis vain ensimmäinen kertaluvun derivaattoja residuaalifunktiosta $\mathbf{r}(\mathbf{x})$.

Tähän lähestymistapaan, jossa jokaisella iteraatioaskeleella ratkaistaan jokin kyseiseen menetelmään liittyvä lineaarinen PNS-tehtävä, perustuu luvussa 3 esiteltävät Gauss-Newton ja Levenberg-Marquardt menetelmät.

2.2.2 Kvadraattinen malli

Toinen lähestymistapa voidaan katsoa olevan rajoittamattoman minimoinnin erikoistapaus. Se perustuu kohdefunktion $f(\mathbf{x})$ kvadraattiseen malliin:

Määritelmä 2.3 (Kvadraattinen approksimaatio). Funktion $f : \mathbb{R}^n \mapsto \mathbb{R}$ kvadraattinen approksimaatio eli toisen kertaluvun Taylorin polynomi $g(\mathbf{x})$ pisteen \mathbf{x}_k läheisyydessä on [9]:

$$g(\mathbf{x}) := f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T(\mathbf{x} - \mathbf{x}_k) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_k)^T \mathbf{H}(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k). \quad (2.2.3)$$

□

Välttämätön ehto g :n minimille on gradientin nollakohta:

$$\nabla g(\mathbf{x}) = \nabla f(\mathbf{x}_k) + \mathbf{H}(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k) = \vec{\mathbf{0}}. \quad (2.2.4)$$

Sijoittamalla \mathbf{x} :n paikalle seuraava iteraatiopiste \mathbf{x}_{k+1} ja ratkaisemalla yhtälö 2.2.4 \mathbf{x}_{k+1} :n suhteen, saadaan perusmuodossa oleva Newtonin menetelmä:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k = \mathbf{x}_k - \mathbf{H}(\mathbf{x}_k)^{-1} \nabla f(\mathbf{x}_k). \quad (2.2.5)$$

Monissa optimointimenetelmissä hakusuunta saadaan ratkaistua muodosta $\mathbf{B}_k \mathbf{p}_k = -\nabla f(\mathbf{x}_k)$. Tärkeitä esimerkkejä ovat [10]:

$$\mathbf{B}_k = \mathbf{I} \rightarrow \text{Gradienttimenetelmä} \quad (2.2.6)$$

$$\mathbf{B}_k = \mathbf{H}(\mathbf{x}_k) \rightarrow \text{Newtonin menetelmä} \quad (2.2.7)$$

$$\mathbf{B}_k \approx \mathbf{H}(\mathbf{x}_k) \rightarrow \text{kvasi-Newton menetelmät} \quad (2.2.8)$$

Kun sijoitetaan PNS-tehtävän gradientti 2.1.3 ja Hessen matriisi 2.1.5 yhtälöön 2.2.5, saadaan Newtonin iteraatioksi

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (\mathbf{J}(\mathbf{x}_k)^T \mathbf{J}(\mathbf{x}_k) + \mathbf{S}(\mathbf{x}_k))^{-1} (\mathbf{J}(\mathbf{x}_k)^T \mathbf{r}(\mathbf{x}_k)), \quad (2.2.9)$$

missä hakusuunta \mathbf{p}_k saadaan siis yhtälöstä

$$(\mathbf{J}(\mathbf{x}_k)^T \mathbf{J}(\mathbf{x}_k) + \mathbf{S}(\mathbf{x}_k)) \mathbf{p}_k = -(\mathbf{J}(\mathbf{x}_k)^T \mathbf{r}(\mathbf{x}_k)) \quad (2.2.10)$$

Newtonin iteraatio ei ole kuitenkaan usein käytetty menetelmä PNS-tehtävissä. Tämä johtuu siitä, että Hessen matriisin toisen kertaluvun osuuden $\mathbf{S}(\mathbf{x})$ laskeminen voi olla usein liian raskasta.

Kvadraattiseen malliin perustuvat lähestymistavat käyttävät siis hyväksi ensimmäisten derivaattojen lisäksi toisen kertaluvun derivaattoja joko eksplisiittisesti tai implisiittisesti. Tähän lähestymistapaan perustuu luvussa 3 esiteltävä kvasi-Newton menetelmä, jossa Hessen matriisia arvioidaan approksimatiivisesti iteratiivisten päivitysten avulla.

3 Ratkaisumenetelmät

Iteratiivisista ratkaisumenetelmistä puhuttaessa suppeneminen (optimiin) on keskeinen käsite. Seuraavat määritelmät ovat Juha Haatajan kirjasta Optimoititehtävien ratkaiseminen [6].

Määritelmä 3.1. (Suppeneminen) Jono $\mathbf{x}_k \in \mathbb{R}^n$, $k = 1, 2, \dots$ suppenee kohti arvoa $\mathbf{x}^* \in \mathbb{R}^n$, jos

$$\lim_{k \rightarrow \infty} \|\mathbf{x}_k - \mathbf{x}^*\| = \vec{0}. \quad (3.0.11)$$

□

Iteratiivisten menetelmien ominaisuuksiin kuuluu täten myös suppenemisen kertaluku:

Määritelmä 3.2. (Lineaarinen ja neliöllinen suppeneminen) Jos on olemassa vakio $0 \leq c < 1$ ja kokonaisluku $N \geq 1$ siten, että kaikilla $k \geq N$ pätee

$$\|\mathbf{x}_{k+1} - \mathbf{x}^*\| \leq c \|\mathbf{x}_k - \mathbf{x}^*\|, \quad (3.0.12)$$

suppenee jono lineaarisesti. Tällöin siis virhe $\|\mathbf{x}_{k+1} - \mathbf{x}^*\|$ vähenee likimain vakiotekijällä jokaisella iteraatiolla. Jos taas jollakin nollaan suppenevalla jonolla c_k pätee

$$\|\mathbf{x}_{k+1} - \mathbf{x}^*\| \leq c_k \|\mathbf{x}_k - \mathbf{x}^*\|, \quad (3.0.13)$$

suppenemisen sanotaan olevan superlineaarista. Lisäksi, jos

$$\|\mathbf{x}_{k+1} - \mathbf{x}^*\| \leq c \|\mathbf{x}_k - \mathbf{x}^*\|^2, \quad (3.0.14)$$

on suppeneminen kvadraattista. □

Esimerkiksi gradienttimenetelmä suppenee yleensä lineaarisesti ja Newtonin menetelmä kvadraattisesti.

Tässä luvussa tutustutaan kolmeen algoritmiin, joita testataan luvussa 4. Kvasi-Newton menetelmät ovat yleisesti käytettyjä menetelmiä rajoittamattomassa optimoinnissa, eikä tässä esitettävää versiota ole modifioitu juuri PNS-tehtäviin. Gauss-Newton ja Levenberg-Marquardt menetelmät taas ovat tarkoitettu nimenomaan epälineaarisiin PNS-tehtäviin. Lähteinä on käytetty pääosin kirjoja [6] [11] (Juha Haataja) ja [7] (J.E. Dennis JR ja Robert B. Schnabel).

3.1 Kvasi-Newton menetelmä

Kvasi-Newton menetelmät eli sekanttimenetelmät perustuvat kvadraattisen mallin lähestymistapaan, ja ovat epälineaarisisessa optimoinnissa ehkä eniten käytettyjä menetelmiä. Ne sopivat tilanteisiin, joissa kohdefunktion Hessen matriisi on vaikea tai raskas laskea. Kuten kappaleen 2.1 lopussa todettiin, tämä usein pätee epälineaariin PNS-tehtäviin. Kvasi-Newton menetelmien modifikaatiot ovat tärkeitä eritoten silloin, kun kohdefunktio on vahvasti epälineaarinen tai kun residuaalit ovat suuria, eli kun $\|\mathbf{S}(\mathbf{x})\|$ on suuri. Ideana on, että Hessen matriisille muodostetaan arvio \mathbf{B}_k , jota ei lasketa vain yhdessä iteraatiopisteessä \mathbf{x}_k , vaan sitä arvioidaan vähitellen iteraatiivisten päivitysten avulla käyttäen hyväksi vain aikaisemmillä iteraatioilla laskettuja gradienttivektoreita. Hessen matriisin arvioon päivitystavalle on monia eri tapoja, mutta tässä esitellään vain yksi yleisemmistä, eli BFGS-sekanttimenetelmä (Broyden-Fletcher-Goldfarb-Shanno). Alaluvussa esitettävä teoria löytyy lähteestä [11].

Uuden iteraatioaskeleen suunta \mathbf{p}_k saadaan jo kappaleessa 2.1 todetusta yhtälöstä

$$\mathbf{B}_k \mathbf{p}_k = -\nabla f(\mathbf{x}_k). \quad (3.1.1)$$

Iteraatioaskeleen pituus valitaan viivahauulla (katso Juha Haataja [6] ja [11]). Perusidea on, että lasketaan ensin hakusuunta \mathbf{p}_k , jonka jälkeen kuljetaan kyseiseen suuntaan kunnes löydetään likimääräinen minikohta. Viivahaku on siis yksiulotteinen optimointitehtävä, jossa askelpituudeksi valitaan

$$\lambda_k = \arg \min_{\lambda > 0} f(\mathbf{p}_k + \lambda \mathbf{p}_k). \quad (3.1.2)$$

Näin voidaan saada algoritmin suppeneminen varmemmaksi ja estää liian pitkät askeleet.

Hessen matriisin arviota päivitetään iteratiivisesti kvasi-Newton menetelmää riippuvalla matriisilla \mathbf{Q}_k :

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \mathbf{Q}_k. \quad (3.1.3)$$

Tarkastellaan seuraavaksi matriisin \mathbf{B}_k päivitykseen liittyviä ehtoja. Kohdefunktion $f(\mathbf{x})$:n oletetaan olevan kahdesti jatkuvasti derivoituva. Integraalilaskennan väliarvolauseesta saadaan

$$\nabla f(\mathbf{x}_{k+1}) = \nabla f(\mathbf{x}_k) + \left(\int_0^1 \nabla^2 f(\mathbf{x}_k + t(\mathbf{x}_{k+1} - \mathbf{x}_k)) dt \right) (\mathbf{x}_{k+1} - \mathbf{x}_k). \quad (3.1.4)$$

Kun merkitään $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$, $\mathbf{y}_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)$ ja $\mathbf{z}(t) = \mathbf{x}_k + t\mathbf{s}_k$, saadaan

$$\left(\int_0^1 \nabla^2 f(\mathbf{z}(t)) dt \right) \mathbf{s}_k = \mathbf{y}_k, \quad (3.1.5)$$

missä integraalissa laskettava matriisi on ikään kuin keskimääräinen Hessen matriisi janalla $\mathbf{z}(t) \in [\mathbf{x}_k, \mathbf{x}_{k+1}]$. Gradientin muutos saadaan siis, kun otettu askel \mathbf{s}_k kerrotaan tällä matriisilla. Tämän tarkastelun perusteella matriisi \mathbf{B}_{k+1} valitaan siten, että se toteuttaa sekanttiyhtälön

$$\mathbf{B}_{k+1}(\mathbf{x}_{k+1} - \mathbf{x}_k) = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k). \quad (3.1.6)$$

\mathbf{B}_{k+1} :n tulisi yleensä olla myös symmetrinen ja positiivisesti definiitti algoritmin tehokkuuden ja stabiiliuden varmistamiseksi [7]. BFGS-sekanttimenetelmän tapauksessa positiividefiniittisyyden säilymisen takaa ehto $\mathbf{y}_k^T \mathbf{s}_k > 0$ (todistus esimerkiksi lähteessä [7] sivuilla 199-201).

Osoitetaan vielä, että ehto $\mathbf{y}_k^T \mathbf{s}_k > 0$ pysyy voimassa iteraation kuluessa:

Olkoon \mathbf{B}_k symmetrinen ja positiividefiniitti matriisi. Iteraation alussa voidaan valita esimerkiksi $\mathbf{B}_1 = \mathbf{I}$. Lasketaan hakusuunta \mathbf{p}_k kaavasta 3.1.1, joka on funktion vähenemissuunta \mathbf{B}_k :n positiividefiniittisyyden perusteella. Hakuaskeleen pituus löydetään viivahaualla $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{p}_k$, $\lambda_k > 0$. Tällöin pätee

$$\mathbf{y}_k^T \mathbf{s}_k = \lambda_k \left(\nabla f(\mathbf{x}_{k+1})^T \mathbf{p}_k - \nabla f(\mathbf{x}_k)^T \mathbf{p}_k \right). \quad (3.1.7)$$

Koska \mathbf{p}_k on funktion vähenemissuunta, on $-\nabla f(\mathbf{x}_k)^T \mathbf{p}_k$ positiivinen. Lisäksi $\nabla f(\mathbf{x}_{k+1})^T \mathbf{p}_k$ saadaan mielivaltaisen pieneksi tarkalla viivahaualla, joten positiividefiniittisyyden säilymisen takaava ehto $\mathbf{y}_k^T \mathbf{s}_k > 0$ pysyy voimassa. Algoritmissa 1 esitellään BFGS-menetelmä pseudokoodina [11].

BFGS-menetelmä suppenee yleensä vain superlineaarisesti, mutta matriisien \mathbf{B}_k positiividefiniittisyyden ansiosta menetelmä on hyvin määritelty ja luotettava, jota voi käyttää moniin erilaisiin optimointitehtäviin.

Algorithm 1 BFGS-sekanttimenetelmä

Valitaan alkuarvaus \mathbf{x}_1 ja \mathbf{B}_1 (esim $\mathbf{B}_1 = \mathbf{I}$) ja $\epsilon > 0$

$k = 1$

repeat

Haetaan suunta \mathbf{p}_k : $\mathbf{B}_k \mathbf{p}_k = -\nabla f(\mathbf{x}_k)$

Viivahaku: $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{p}_k$

$\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$

$\mathbf{y}_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)$

$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k} - \frac{\mathbf{B}_k \mathbf{s}_k \mathbf{s}_k^T \mathbf{B}_k}{\mathbf{s}_k^T \mathbf{B}_k \mathbf{s}_k}$

$k = k + 1$

until $\|\nabla f(\mathbf{x}_k)\| < \epsilon$

3.1.1 Kvasi-Newton menetelmät PNS-tehtävissä

Vaikkakin usein käytetty, standardimuotoinen BFGS-menetelmä ei kuitenkaan ole paras kvasi-Newton menetelmään perustuva vaihtoehto epälineaarisiiin PNS-tehtäviin, koska se ei ota huomioon Hessen matriisiin $\mathbf{H}(\mathbf{x}) = 2\mathbf{J}(\mathbf{x})^T \mathbf{J}(\mathbf{x}) + 2\mathbf{S}(\mathbf{x})$ erityistä rakennetta, vaan approksimoi suoraan koko Hessen matriisia.

Tämän takia onkin kehitetty hybridimetodeja, jotka voivat vaihdella BFGS-menetelmän ja Gauss-Newton -tyyppisen menetelmän välillä. Vaihtoehtoisesti voidaan implementoida kvasi-Newton menetelmä, joka ottaa huomioon Hessen matriisin erityisrakenteen siten, että ensimmäisen kertaluvun käyttökelpoinen termi $\mathbf{J}(\mathbf{x})^T \mathbf{J}(\mathbf{x})$ otetaan sellaisenaan, ja vain toisen kertaluvun termiä $\mathbf{S}(\mathbf{x})$ arvioidaan iteratiivisesti sekanttimenetelmien tapaan. Tämän tyyppisen kvasi-Newton menetelmän periaate on siis seuraava:

$$\begin{aligned} \mathbf{B}_k &= 2\mathbf{J}(\mathbf{x}_k)^T \mathbf{J}(\mathbf{x}_k) + 2\mathbf{A}_k \\ \rightarrow \mathbf{B}_{k+1} &= 2\mathbf{J}(\mathbf{x}_{k+1})^T \mathbf{J}(\mathbf{x}_{k+1}) + 2\mathbf{A}_{k+1}, \end{aligned}$$

missä \mathbf{A}_k ja \mathbf{A}_{k+1} ovat sekanttiapproksimaatioita matriiseille \mathbf{S}_k ja \mathbf{S}_{k+1} . Tämänkaltaiset menetelmät ovat robustimpia, mutta myös monimutkaisempia kuin lineaariseen malliin perustuvat menetelmät. Lisää tietoa löytyy esimerkiksi Weijun Zhoun ja Xiaojun Chenin artikkelista Global Convergence of a New Hybrid Gauss-Newton Structured BFGS Method for Nonlinear Least Squares Problems (2010) [12] sekä lähteistä [7] ja [8].

3.2 Gauss-Newton menetelmä

Gauss-Newton menetelmä on yksinkertaisin vaihtoehto PNS-tehtävien ratkaisemiseen, joka perustuu lineaarisen mallin lähestymistapaan. Gauss-Newton iteraatioaskel $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k$ on lineaarisen PNS-tehtävän 2.2.2 ratkaisu:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \left(\mathbf{J}(\mathbf{x}_k)^T \mathbf{J}(\mathbf{x}_k) \right)^{-1} \mathbf{J}(\mathbf{x}_k) \mathbf{r}(\mathbf{x}_k). \quad (3.2.1)$$

Hakusuunta \mathbf{p}_k saadaan ratkaistua siis jokaisella iteraatioaskeleella yhtälöstä

$$\mathbf{J}(\mathbf{x}_k)^T \mathbf{J}(\mathbf{x}_k) \mathbf{p}_k = -\mathbf{J}(\mathbf{x}_k) \mathbf{r}(\mathbf{x}_k). \quad (3.2.2)$$

Tästä huomaakin, että Gauss-Newton eroaa kvadraattiseen malliin perustuvasta Newtonin menetelmästä vain siten, että Hessen matriisin toisen kertaluvun informaatio $\mathbf{S}(\mathbf{x})$ jätetään huomioitta. Tämän perusteella menetelmän toimivuus riippuu pitkälti siitä, kuinka merkittävä osuus termillä $\mathbf{S}(\mathbf{x})$ on Hessen matriisissa $\mathbf{H}(\mathbf{x}) = 2\mathbf{J}(\mathbf{x})^T \mathbf{J}(\mathbf{x}) + 2\mathbf{S}(\mathbf{x})$.

Mikäli kohdefunktio on vain lievästi epälineaarinen tai residuaalit ovat pieniä optimin lähellä, eli $\|\mathbf{S}(\mathbf{x}^*)\|$ on pieni, menetelmä suppenee lineaarisesti. Jos $\|\mathbf{S}(\mathbf{x}^*)\| = 0$, Gauss-Newton suppenee jopa kvadraattisesti kuten Newtonin menetelmä [7]. Tämän takia menetelmä soveltuu myös epälineaaristen yhtälöryhmien ratkaisemiseen. Jos taas $\|\mathbf{S}(\mathbf{x}^*)\|$ on suuri, menetelmä suppenee hitaasti tai saattaa jopa hajaantua. Mitä suurempi $\mathbf{S}(\mathbf{x}^*)$ on verrattuna osuuteen $\mathbf{J}(\mathbf{x}^*)^T \mathbf{J}(\mathbf{x}^*)$, sitä huonommin Gauss-Newton menetelmä todennäköisesti toimii. Siksi menetelmä tarvitsee usein hyvän alkuarvauksen.

Toinen huono puoli menetelmässä on, että mikäli Jacobin matriisi ei ole täysiasteinen (eli jos sen kaikki sarakkeet eivät ole lineaarisesti riippumattomia), tai jos Jacobin matriisi on häiriöaltis, ei iteraatioaskel ole hyvin määritelty. Tämän voi kiertää tekemällä Jacobin matriisille esimerkiksi singulaariarvohajotelma (katso Juha Haatajan kirja [6]), jolloin ratkaisu löytyy vaikka Jacobin matriisi olisi singulaarinen tai häiriöaltis. Hajotelman laskeminen vaatii toisaalta melko paljon työtä.

Jos $\mathbf{J}(\mathbf{x}_k)$ kuitenkin on täysiasteinen, on iteraatioaskel aina laskusuuntaan, mutta saattaa olla liian pitkä, jolloin kohdefunktion arvo ei välttämättä pienene jokaisella iteraatiolla. Tämän perusteella Gauss-Newton menetelmää voi parantaa esimerkiksi viivahaulla, jolloin iteraatioaskel näyttää seuraavalta:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \lambda_k \left(\mathbf{J}(\mathbf{x}_k)^T \mathbf{J}(\mathbf{x}_k) \right)^{-1} \mathbf{J}(\mathbf{x}_k) \mathbf{r}(\mathbf{x}_k). \quad (3.2.3)$$

Viivahausta huolimatta iteraatioaskel ei ole hyvin määritelty mikäli $\mathbf{J}(\mathbf{x}_k)$ ei ole täysiasteinen.

Gauss-Newton algoritmi pseudokoodina (ilman viivahakua) [13] esitetään algoritmossa 2.

Algorithm 2 Gauss-Newton menetelmä

Valitaan alkuarvaus \mathbf{x}_1 ja $\epsilon > 0$

$k = 1$

repeat

Haetaan suunta \mathbf{p}_k : $\mathbf{J}(\mathbf{x}_k)^T \mathbf{J}(\mathbf{x}_k) \mathbf{p}_k = -\mathbf{J}(\mathbf{x}_k)^T \mathbf{r}(\mathbf{x}_k)$

$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k$

$k = k + 1$

until $\|\mathbf{J}(\mathbf{x}_{k+1})^T \mathbf{r}(\mathbf{x}_{k+1})\| < \epsilon$

Kuten melkein kaikissa optimointimenetelmissä, myös Gauss-Newtonista löytyy erilaisia modifikaatioita. Näistä löytyy lisää tietoa esimerkiksi lähteestä [13].

Toinen tapa saada Gauss-Newton -menetelmän suppeneminen varmemmaksi on ns. luottamusalueen käyttö, johon perustuu seuraavassa tarkasteltava Levenberg-Marquardt menetelmä.

3.3 Levenberg-Marquardt menetelmä

Levenberg-Marquardt menetelmä on suosituimpia menetelmiä epälineaarisissa PNS-tehtävissä, joka perustuu myös lineaarisen mallin lähestymistapaan. Suurena etuna Gauss-Newtoniin nähden on, että iteraatioaskel on hyvin määritelty, vaikka $\mathbf{J}(\mathbf{x}_k)$ ei olisikaan täysiasteinen tai olisi häiriöaltis. Se perustuu ns. luottamusalueen käyttöön, jossa ratkaistaan seuraava PNS-tehtävä (vertaa yhtälöön 2.2.2)

$$\min_{\mathbf{p}_k} \left(\|\mathbf{J}(\mathbf{x}_k) \mathbf{p}_k + \mathbf{r}(\mathbf{x}_k)\|^2 + \mu_k \|\mathbf{p}_k\|^2 \right), \quad (3.3.1)$$

missä parametri $\mu_k \geq 0$ rajoittaa \mathbf{p}_k :n kokoa, ja varmistaa hakusuunnan kertoimen positiividefiniittisyyden, jolloin \mathbf{p}_k on hyvin määritelty laskusuunta.

Tarkempi kuvaus menetelmän luottamusalueeseen perustuvasta luonteesta löytyy lähteistä [7], [8] ja [14]. Yleisempi kuvaus luottamusalumenetelmistä löytyy lähteestä [6].

Suppenemisominaisuuksiltaan Levenberg-Marquardt on melko samanlainen kuin Gauss-Newton. Jos kohdefunktio on vain lievästi epälineaarinen tai alkuarvaus on tarpeeksi hyvä, Levenberg-Marquardt saattaa olla hieman hitaampi, koska luottamusalue voi vaikuttaa hidastavasti suppenemiseen. Toisaalta se on myös robustimpi, eli ratkaisu löytyy todennäköisemmin vaikka alkuarvaus olisi kauempana optimista. Levenberg-Marquardt selvää myös suuren residuaalin tehtävistä paremmin kuin Gauss-Newton, vaikka saattaa-kin supeta hitaasti.

Tehtävän 3.3.1 ratkaisuna saadaan Levenberg-Marquardt iteraatioaskel

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \left(\mathbf{J}(\mathbf{x}_k)^T \mathbf{J}(\mathbf{x}_k) + \mu_k \mathbf{I} \right)^{-1} \mathbf{J}(\mathbf{x}_k)^T \mathbf{r}(\mathbf{x}_k). \quad (3.3.2)$$

Yhtäpitävästi \mathbf{p}_k saadaan siis jokaisella iteraatioaskeleella yhtälöstä

$$\left(\mathbf{J}(\mathbf{x}_k)^T \mathbf{J}(\mathbf{x}_k) + \mu_k \mathbf{I} \right) \mathbf{p}_k = -\mathbf{J}(\mathbf{x}_k)^T \mathbf{r}(\mathbf{x}_k). \quad (3.3.3)$$

Joskus suppenemista voidaan saada nopeutettua korvaamalla identiteettimatriisi diagonaalimatriisilla $\text{diag}(\mathbf{J}(\mathbf{x}_k)^T \mathbf{J}(\mathbf{x}_k))$, joka ikään kuin skaalaa gradientin komponentit kaarevuuden mukaan, jolloin voidaan välttää hidastuminen suuntiin, joissa gradientti on pieni. Tällöin hakusuunta ratkaistaan siis yhtälöstä

$$\left(\mathbf{J}(\mathbf{x}_k)^T \mathbf{J}(\mathbf{x}_k) + \mu_k \text{diag}(\mathbf{J}(\mathbf{x}_k)^T \mathbf{J}(\mathbf{x}_k)) \right) \mathbf{p}_k = -\mathbf{J}(\mathbf{x}_k)^T \mathbf{r}(\mathbf{x}_k). \quad (3.3.4)$$

Parametrin μ_k :n määrittämiseen ja päivittämiseen on monia implementaatioita, joista muutamia löytyy esimerkiksi lähteistä [15] ja [16]. Jos μ_k on suuri, on Levenberg-Marquardtin askel lähellä jyrkimmän laskun suuntaa eli negatiivista gradienttia. Tällöin myös askeleen pituus on pieni. Jos μ_k on pieni, on menetelmä lähellä Gauss-Newton menetelmää. Näin on yleensä etenkin lähellä optimia. Levenberg-Marquardt hakusuunta on siis jotain Gauss-Newton suunnan ja negatiivisen gradientin suunnan väliltä.

Eräs tapa μ_k :n päivittämiseen on adaptiivinen menettely, jossa μ_k :n arvoa kasvatetaan tarvittaessa iteraatio iteraatiolta, mutta muuten pienennetään.

Tähän perustuva Levenberg-Marquardt algoritmi on esitetty pseudokoodina algoritmissa 3 [6].

Algorithm 3 Levenberg-Marquardt menetelmä

```

Valitaan alkuarvaus  $\mathbf{x}_1$  ja  $\epsilon > 0$ 
Asetetaan  $\mu_1 = 0.01$  ja  $v = 10$  sekä  $k = 1$ 
repeat
  Asetetaan  $\mu_k = \mu_k/v$ 
  repeat
    Ratkaistaan hakusuunta  $\mathbf{p}_k$  yhtälöstä 3.3.3 tai 3.3.4
     $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k$ 
    if  $f(\mathbf{x}_{k+1}) > f(\mathbf{x}_k)$  then
       $\mu_k = v\mu_k$ 
    end if
  until  $f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$ 
   $\mu_{k+1} = \mu_k$  ja  $k = k + 1$ 
until  $\|\mathbf{J}(\mathbf{x}_{k+1})^T f(\mathbf{x}_{k+1})\| < \epsilon$ 

```

4 Menetelmien testaaminen

Tässä luvussa kokeillaan algoritmeja muutamalla esimerkkitehtävällä. Gauss-Newton menetelmä on ohjelmoitu itse, koska sitä ei matlabin nykyversioista enää löydy. BFGS-menetelmä löytyy matlabin solverista fminunc [3], ja Levenberg-Marquardt taas solveista lsqnonlin [4] ja lsqcurvefit [5]. Algoritmeissa 1 ja 3 on kuvattu menetelmien perusideat yksinkertaisena pseudokoodina. Matlabin sisäiset toteutukset ovat kuitenkin todennäköisesti monimutkaisempia. Tehtävien ratkaisussa taulukoidaan iteraatioiden määrät kullekin algoritmille mikä vaadittiin globaalin optimin saavuttamiseen, mikäli ne sinne pääsivät.

Taulukoita varten käytetään seuraavia notaatioita:

GN = Gauss-Newton ilman viivahakua

GN_v = Gauss-Newton viivahaulla

LM = Levenberg-Marquardt, jossa hakusuunta ratkaistaan yhtälöstä 3.3.3

LM_d = Levenberg-Marquardt, jossa hakusuunta ratkaistaan yhtälöstä 3.3.4

BFGS = Broyden-Fletcher-Goldfarb-Shanno sekanttimenetelmä

* = hajaantuu

p = suppenee, mutta ei globaaliin minimiin

4.1 Yksinkertainen sinifunktio

Aloitetaan yksinkertaisella tehtävällä, jonka tarkoitus on havainnollistaa epälineaarisen PNS-tehtävän idea käytännössä. Oletetaan, että haluamme sovittaa 4 datapistettä

Taulukko 1: datapisteet

(t_1, y_1)	(t_2, y_2)	(t_3, y_3)	(t_4, y_4)
$(-2, -2)$	$(0, 0)$	$(2, 2)$	$(4, -1.5)$

yksinkertaiseen sinifunktioon

$$a(t, \mathbf{x}) = 2 \sin(x_1 t + x_2). \quad (4.1.1)$$

Optimoitavana on siis 2 parametria: $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$, x_1 = taajuus ja x_2 = vaihe-siirto. Nyt $r_i(\mathbf{x}) = 2 \sin(x_1 t_i + x_2) - y_i$, $i = 1, \dots, 4$, jolloin pienimmän neliösumman tehtäväksi saadaan yleismuodossa 2.1.1:

$$\min_{\mathbf{x} \in \mathbb{R}^2} f(\mathbf{x}) = \sum_{i=1}^4 (2 \sin(x_1 t_i + x_2) - y_i)^2. \quad (4.1.2)$$

Residuaalivektorin $\mathbf{r}(\mathbf{x})$ Jacobin matriisi $\mathbf{J}(\mathbf{x}) \in \mathbb{R}^{4 \times 2}$ on

$$\mathbf{J}(\mathbf{x}) = \begin{pmatrix} 2t_1 \cos(x_1 t_1 + x_2) & 2 \cos(x_1 t_1 + x_2) \\ 2t_2 \cos(x_1 t_2 + x_2) & 2 \cos(x_1 t_2 + x_2) \\ 2t_3 \cos(x_1 t_3 + x_2) & 2 \cos(x_1 t_3 + x_2) \\ 2t_4 \cos(x_1 t_4 + x_2) & 2 \cos(x_1 t_4 + x_2) \end{pmatrix}. \quad (4.1.3)$$

ja gradientti $\nabla f(\mathbf{x}) \in \mathbb{R}^{2 \times 1}$ on

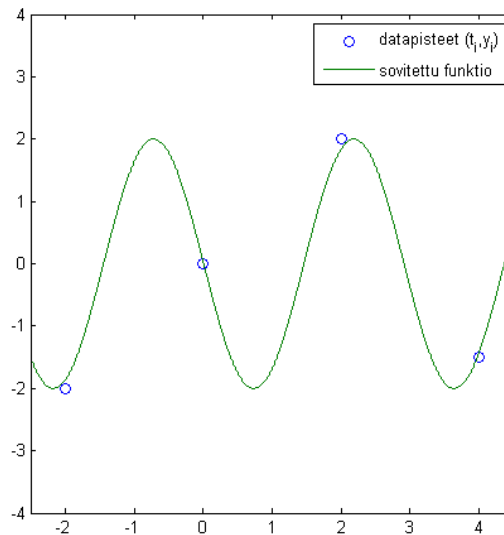
$$\begin{aligned}\nabla f(\mathbf{x}) &= 2\mathbf{J}(\mathbf{x})^T \mathbf{r}(\mathbf{x}) \\ &= 2 \begin{pmatrix} \sum_{i=1}^4 2t_i \cos(x_1 t_i + x_2) r_i(\mathbf{x}) \\ \sum_{i=1}^4 2 \cos(x_1 t_i + x_2) r_i(\mathbf{x}) \end{pmatrix}.\end{aligned}$$

Ratkaistaan tehtävä kullakin luvun 3 algoritmilla käyttäen alkuarvausta $\mathbf{x}_0 = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$. Tehtävä on niin yksinkertainen, että jokainen algoritmi selvisi tehtävästä helposti. Optimaaliseksi parametreiksi saatiin $\mathbf{x}^* = \begin{pmatrix} 2.16 \\ 3.12 \end{pmatrix}$, jolloin kohdefunktion arvo on $f(\mathbf{x}^*) = 0.0514$. Kyseessä on siis pienen residuaalin tehtävä.

Iteraatioiden määrät näkyvät taulukossa 2 ja optimaalinen sovitus kuvassa 1.

Taulukko 2: Iteraatioiden määrä

	BFGS	LM	LM _d	GN	GN _v
Iter. lkm	9	7	7	10	6



Kuva 1: Sinifunktio dataan sovitettuna

Näin yksinkertaisessa ja pienessä tehtävässä ei näe vielä suuria eroja menetelmien suorituksissa, sillä residuaalit ovat lähellä nollaa.

Näytetään seuraavaksi, minkälainen vaikutus suuremmilla residuaaleilla on tehtävään. Vaihdetaan mittauspiste y_3 :n tilalle selvästi muusta datasta poikkeava arvo; $y_3 = 6$, ja tehdään sama sovitus uudestaan samalla alkuarvauksella. Tällä on vaikutusta sekä optimiin että kohdefunktion arvoon, koska nyt $r_3(\mathbf{x})$ on suurempi. Nyt optimiksi saadaan $\mathbf{x}^* = \begin{pmatrix} 2.19 \\ 3.27 \end{pmatrix}$, ja kohdefunktion arvo $f(\mathbf{x}^*) = 16.67$ eli selvästi suurempi. Iteraatioiden määrät näkyvät taulukossa 3.

Taulukko 3: Iteraatioiden määrä

	BFGS	LM	LM _d	GN	GN _v
Iter. lkm	9	12	12	*	18

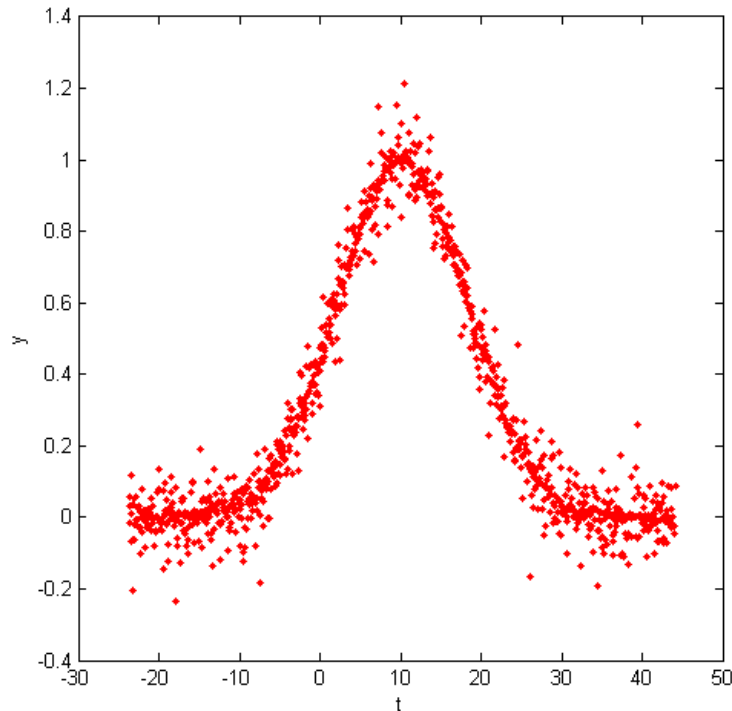
Tästä huomaa, että residuaalien kasvaessa suuremmiksi Gauss-Newton menetelmä hidastuu, ja tässä tapauksessa jopa hajaantuu mikäli ei käytetä viivahakua. Robustimmat menetelmät Levenberg-Marquardt ja kvasi-Newton ratkaisivat tehtävän edelleen helposti, vaikkakin myös Levenberg-Marquardt hidastui hieman.

Jos datapisteiden oletetaan olevan tarkkoja, ei suuren residuaalin tapauksessa malli 4.1.1 tietenkään enää kuvaa dataa kovinkaan hyvin. Tällöin sovituksista voidaan parantaa muokkaamalla mallia esimerkiksi siten, että otetaan kolmanneksi optimoitavaksi parametriksi amplitudi, mikä oli tässä esimerkissä vakio 2. Joskus data voi kuitenkin olla häiriöistä, mikä tarkoittaa että mittauspisteiden joukossa voi olla suuriresiduaalisia pisteitä, vaikka sovitettavassa mallissa ei olisi mitään vikaa. Tämäntyyppisissä tehtävissä jokaiseen mittauspisteeseen voidaan liittää tietty epävarmuus eli painokerroin. Jos tiedetään, että jokin suuriresiduaalinen piste on esimerkiksi mittausvirhe tai että mittauspiste on epätarkka, se voidaan joko jättää huomioitta tai antaa sille pieni painokerroin. Painotettu pienimmän neliösumman tehtävä ei kuitenkaan kuulu tämän työhön aiheisiin, joten näissä esimerkeissä jokaisella mittauspisteellä on sama painokerroin, eli 1. Työn intressien kannalta on muutenkin mielenkiintoisempaa pitää suuren residuaalin pisteet mukana optimoinnissa.

Tämän esimerkin tarkoitus oli lähinnä esitellä epälineaarinen PNS-tehtävä käytännön tasolla, ja havainnollistaa residuaalien suuruuden vaikutusta. Seuraavat esimerkit ovat todenmukaisempia ja hieman monimutkaisempia.

4.2 Gaussin funktio

Kuvassa 2 näkyy tuhannen pisteen testidata $(t_i, y_i), i = 1, \dots, 1000$ joka löytyi Bucknellin yliopiston sivuilta [17].



Kuva 2: Datapisteet, joihin sovitetaan Gaussin funktio

Tähän dataan on tarkoitus sovittaa Gaussin funktio:

$$a(t, \mathbf{x}) = x_1 e^{\frac{-(t-x_2)^2}{(2x_3^2)}} + x_4 \quad (4.2.1)$$

Optimoitavana on siis neljä parametria. x_1 on amplitudi, x_2 on käyrän huipun keskipisteen kohta, x_3 kontrolloi käyrän leveyttä ja x_4 on termi, jota funktion arvot asymptoottisesti lähestyvät huipulta poispäin tultaessa.

Residuaali kohdassa t_i on

$$r_i(\mathbf{x}) = a(t_i, \mathbf{x}) - y_i = x_1 e^{\frac{-(t_i-x_2)^2}{(2x_3^2)}} + x_4 - y_i, \quad (4.2.2)$$

jolloin pienimmän neliösumman tehtäväksi saadaan kaavan 2.1.1 mukaan

$$\min_{\mathbf{x} \in \mathbb{R}^4} = \sum_{i=1}^{1000} (x_1 e^{\frac{-(t_i - x_2)^2}{(2x_3^2)}} + x_4 - y_i)^2. \quad (4.2.3)$$

Gradientti ja Jacobin matriisi lasketaan kuten edellä, kaavojen 2.1.3 ja 2.1.4 mukaan.

Kuvasta 2 pystyy päättämään melko hyvät alkuarvaukset parametreille. Amplitudi näyttäisi olevan noin 1, huipun keskikohta suunnilleen 10, leveys huipulta noin 10 ja vakiotermi 0. Kukin algoritmi pystyi ratkaisemaan tehtävän tällä hyvällä alkuarvauksella. Tällöin parametrien optimaalisiksi arvoiksi tulivat

$$\mathbf{x}^* = \begin{pmatrix} 1,006 \\ 10,2375 \\ 8,0472 \\ -0,0064 \end{pmatrix} \quad (4.2.4)$$

jolloin kohdefunktion arvo eli residuaalien neliösumma on $f(\mathbf{x}^*) = 3,4065$. Kunkin algoritmin iteraatioiden määrät näkyvät taulukossa 4, ja dataan sovitettu neliösumman minimoiva käyrä on kuvassa 3.

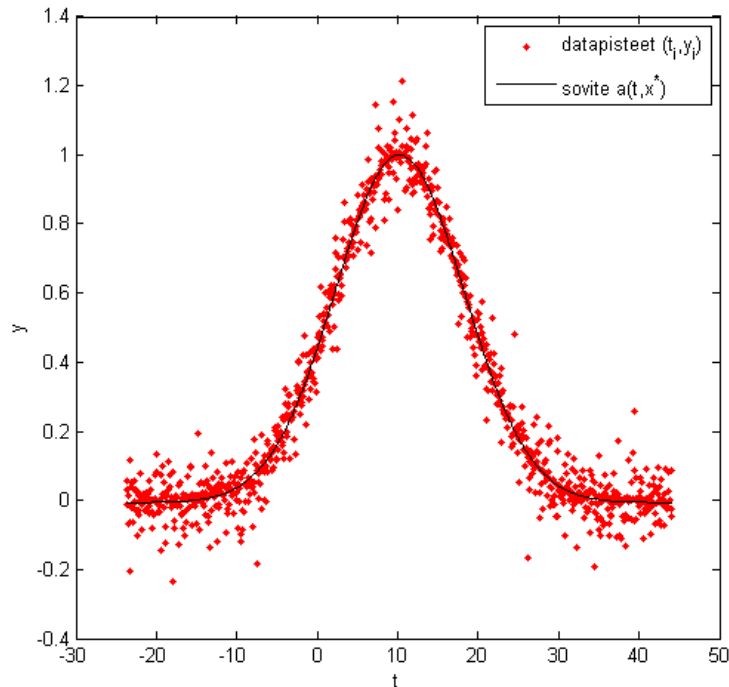
Taulukko 4: Iteraatioiden määrä hyvällä alkuarvauksella $\mathbf{x}_0 = (1, 10, 10, 0)^T$

	BFGS	LM	LM _d	GN	GN _v
Iter. lkm	25	4	4	8	8

Kuvan 3 perusteella Gaussin funktio kuvaa hyvin testidataa. Residuaalit ovat tehtävässä edelleen melko pieniä, mutta kohdefunktio arvo ei kuitenkaan ihan lähelle nollaa pääse. Tässä tapauksessa Gauss-Newton on jo hitaampi kuin Levenberg-Marquardt.

Kokeillaan seuraavaksi tehtävän ratkaisemista täysin surkealla alkuarvauksella. Ei ole tietenkään viisasta valita huonoa alkuarvausta, jos sen pystyy kuvasta päättämään kuten tässä tapauksessa, mutta aina näin ei ole. Tulokset huonolla alkuarvauksella ovat taulukossa 5.

Tästä tulee jo paremmin ilmi Levenberg-Marquardt robustisuus. Se löysi optimin edelleen vain 21 iteraatiolla huonosta alkuarvauksesta huolimatta. Tämän perusteella on jo helppo ymmärtää, miksi se on paljon käytetty menetelmä epälineaarisissa PNS-tehtävissä. Tässä tapauksessa diagonaa-



Kuva 3: Gaussin funktio sovitettuna dataan

Taulukko 5: Iteraatioiden määrä huonolla alkuarvauksella $\mathbf{x}_0 = (100, -20, 40, -20)^T$

	BFGS	LM	LM _d	GN	GN _v
Iter. lkm	17,p	21	27	*	*

limatriisin $diag(\mathbf{J}(\mathbf{x}_k)^T \mathbf{J}(\mathbf{x}_k))$ käyttäminen identiteettimatriisin tilalla jopa hieman hidasti iteraatiota, jolloin globaali optimi löytyi 27 iteraatiolla. Gauss-Newton hajaantui sekä viivahaulla että ilman. Tämä saattaa johtua siitä, että Jacobin matriisi voi olla häiriöaltis kaukana optimista, koska eksponenttifunktion arvot muuttuvat paljon pienistäkin parametrien muutoksista. BFGS-menetelmä taas väitti löytäneensä jonkun paikallisen minimin, jossa parametrien arvot olivat kuitenkin kaukana globaalista optimista ja sovitus ei vastannut ollenkaan datapisteitä. Tästä jo ehkä huomaa, ettei BFGS ole tarkoitettu nimenomaan PNS-tehtäviin, vaikka hyvällä alkuarvauksella pystyikin ratkaisemaan tehtävän melko vaivattomasti.

4.3 Kolmiulotteinen esimerkki

Otetaan viimeiseksi vielä haastava esimerkki, mikä osoittautui vaikeaksi tehtäväksi kaikille menetelmille. Tämän esimerkin testidata on peräisin yhdysvaltalaisen National Institute of Standards and Technology (NIST) tehtäväärkistosta [18]. 128 pisteen mittausdata kuvaa eristeen läpilyöntivahvuutta kilovolteissa (y) ajan (t_1) sekä lämpötilan (t_2) funktiona. Sovitettava malli on seuraavanlainen:

$$a(t_1, t_2, \mathbf{x}) = x_1 + x_2 t_1 e^{-x_3 t_2} \quad (4.3.1)$$

Malli halutaan sovittaa y :n logaritmin suhteen, jolloin pienimmän neliösumman tehtäväksi saadaan:

$$\min_{\mathbf{x} \in \mathbb{R}^3} f(\mathbf{x}) = \sum_{i=1}^{128} (x_1 + x_2 t_{1i} e^{-x_3 t_{2i}} - \log(y_i)) \quad (4.3.2)$$

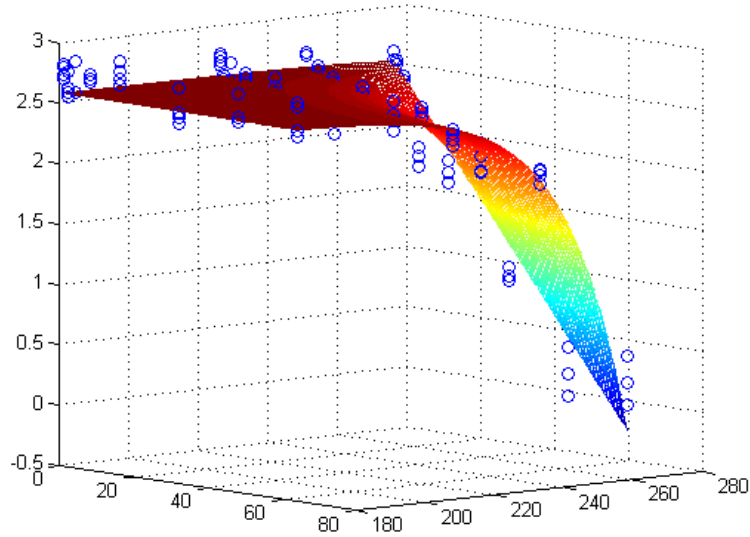
Optimoitavana on siis kolme parametria $\mathbf{x} = (x_1, x_2, x_3)^T$. Gradientti ja Jacobin matriisi lasketaan niin ikään kuten edellä.

Tehtävän globaali optimi on $\mathbf{x}^* = (2.5907, 5.618 \cdot 10^{-9}, -5.770 \cdot 10^{-2})^T$ ja kohdefunktion arvo $f(\mathbf{x}^*) = 3.798$. NIST:n sivuilla oli myös annettu muutama alkuarvaus tehtävän ratkaisemiseen. Kokeillaan ensin alkuarvauksella, joka on melko lähellä optimia. Tulokset ovat taulukossa 6 ja sovitettu malli kuvassa 4.

Taulukko 6: $\mathbf{x}_0 = (2.5, 5 \cdot 10^{-9}, -5 \cdot 10^{-2})^T$

	BFGS	LM	LM _d	GN	GN _v
Iter. lkm	0,p	95	32	*	41

Kohdefunktion arvo on todella herkkä parametrien muutokselle. Vaikka alkuarvaus on lähellä optimia, on kohdefunktion arvo silti jopa $f(\mathbf{x}_0) = 56.923$. Jacobin matriisikin on lähellä singulaarista, joten Gauss-Newton hajaantuu ilman viivahakua. Viivahaun kanssa se kuitenkin löysi globaalin optimin jopa 41 iteraatiolla. LM vaati 95 iteraatiota, kun taas LM_d ainoastaan 32. BFGS-menetelmä taas ei edes päässyt alkupisteestä liikkeelle. Matlab ilmoitti, että menetelmä ei pysty pienentämään kohdefunktion arvoa kyseiseen hakusuuntaan. Hessen matriisin arvio \mathbf{B}_1 on alustettu identiteettimatriisiksi, jolloin menetelmän ensimmäinen askel on negatiivisen gradientin suuntaan, mikä saattaa olla huono suunta tässä tapauksessa.



Kuva 4: Sovitettu malli

Kokeillaan lopuksi vielä huonommalla alkuarvauksella. Tulokset ovat taulukossa 7.

Taulukko 7: $\mathbf{x}_0 = (2, 10^{-4}, -10^{-2})^T$

	BFGS	LM	LM _d	GN	GN _v
Iter. lkm	132	229,p	162	*	40

Nyt BFGS -menetelmäkin ratkaisi tehtävän, vaatien 132 iteraatiota. LM iteroi 229 kertaa, mutta ei päätenyt globaaliin optimiin. LM_d vaati myös jopa 162 iteraatiota, mutta löysi kuitenkin globaalin optimin. Tässä tehtävässä hakusuunnan 3.3.4 käyttäminen siis paransi Levenberg-Marquardt menetelmän suoritusta huomattavasti. Gauss-Newton hajaantui jälleen, mutta viiva-haun avulla saavutti optimin ainoastaan 40 iteraatiolla, eli jopa yksi iteraatio vähemmän kuin paremmalla alkuarvauksella.

5 Yhteenveto

Työn tavoitteena oli tutustua epälineaarisiin pienimmän neliösumman tehtäviin, esitellä kolme algoritmia niiden ratkaisemiseen, ja testata algoritmeja muutamalla esimerkkitehtävällä.

Kvasi-Newton BFGS-menetelmä osoittautui varsin luotettavaksi menetelmäksi yleisesti epälineaarisisissa optimoinnissa. PNS-tehtäviin se ei kuitenkaan ole paras vaihtoehto, koska se approksimoi suoraan koko Hessen matriisia, eikä ota huomioon PNS-tehtävän erityistä rakennetta. Tämän takia onkin kehitetty erilaisia modifikaatioita kvasi-Newton menetelmistä, mitkä soveltuvat paremmin juuri PNS-tehtäviin. Myös BFGS-menetelmä ratkaisi suhteellisen hyvin työssä esitetyt esimerkkitehtävät, vaikka ei ensisijainen valinta PNS-tehtäviin olekaan.

Gauss-Newton menetelmä osoittautui toimivan parhaiten, kun alkuarvaus on hyvä ja residuaalit pieniä. Menetelmä ei ole kuitenkaan yhtä luotettava kuin kvasi-Newton ja Levenberg-Marquardt, koska esimerkiksi häiriöalttiit tai vahvasti epälineaariset tehtävät tuottavat sille ongelmia, kuten esimerkeissä huomattiin. Sekin kertoo jo jotain, että Matlabin nykyversioissa ei ole enää valmista rutiinia Gauss-Newton menetelmälle. Gauss-Newton voidaan saada kuitenkin luotettavammaksi viivahaulla.

Levenberg-Marquardt menetelmä on näistä kolmesta algoritmista ehkä luotettavin epälineaarisisissa PNS-tehtävissä. Vaikka sekään ei huomioi toisen kertaluvun derivaattoja Hessen matriisin approksimoinnissa, on menetelmä silti robustimpi kuin Gauss-Newton, sillä Levenberg-Marquardt käyttää luotamusalueeseen liittyvää logiikkaa suppenemisen varmistamiseksi.

Kolmen erilaisen esimerkin myötä kuitenkin osoittautui, että yleispäteviä johtopäätöksiä menetelmien suorituksista on vaikea tehdä yksittäisten esimerkkien perusteella. Tehtävien rakenne ja alkuarvaus vaikuttavat algoritmien suoritukseen lähes ainutlaatuisella tavalla, kuten näistäkin esimerkeistä tuli ilmi; yksikään menetelmä ei ollu varsinaisesti paras jokaisessa tilanteessa. Teoriaosuudessa esitetyistä asioista voi kuitenkin saada jonkinlaisen ohjenuoran menetelmän valitsemiseen, jos on tarvetta ratkaista epälineaarinen pienimmän neliösumman tehtävä.

Viitteet

- [1] C.T. Kelley, North Carolina State University *Iterative Methods for Optimization* http://www.siam.org/books/textbooks/fr18_book.pdf, viitattu 4.8.2014
- [2] T. Eirola. *MS-C1340 Lineaarialgebra ja Differentiaaliyhtälöt* kurssin materiaalit, Lineaarialgebran perusteet sivut 54-57. <https://noppa.aalto.fi/noppa/kurssi/ms-c1340/materiaali>, viitattu 4.8.2014
- [3] Mathworks, Documentation center *Optimization toolbox, fminunc* <http://www.mathworks.se/help/optim/ug/fminunc.html>, viitattu 4.8.2014
- [4] Mathworks, Documentation center *Optimization toolbox, lsqnonlin* <http://www.mathworks.se/help/optim/ug/lsgnonlin.html>, viitattu 4.8.2014
- [5] Mathworks, Documentation center *Optimization toolbox, lsqcurvefit* <http://www.mathworks.se/help/optim/ug/lsgcurvefit.html>, viitattu 4.8.2014
- [6] J. Haataja *Optimointitehtävien ratkaiseminen*, CSC - Tieteellinen laskenta Oy, 1995, 2.painos sivut 40-41, 127-133, 185-202
- [7] J.E. Dennis, JR. Robert B. Schnabel *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice Hall Series in Computational Mathematics, 1983. Luku 9: Secant Methods for Unconstrained Minimization, sivut 194-212. Luku 10: Nonlinear least squares, sivut 218-238.
- [8] Å. Björck *Numerical Methods for Least Squares Problems*, Society for Industrial and Applied Mathematics, 1996. sivut 339-351.
- [9] H. Ehtamo *MS-C2105 Optimoinnin perusteet* kurssin luentomateriaali, luento 10. Luennoitu 6.2.2014. <https://noppa.aalto.fi/noppa/kurssi/ms-c2105/luennot>, viitattu 11.8.2014
- [10] J. Eriksson, Ph.D. Thesis, 1996. *Optimization and Regularization of Nonlinear Least Squares Problems*. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.49.845&rep=rep1&type=pdf>, viitattu 11.8.2014
- [11] J. Haataja *Suurten optimointitehtävien ratkaiseminen*, CSC - Tieteellinen laskenta Oy, 1994 sivut 25-39.

- [12] Weijun Zhou, Xiaojun Chen, 4/2010 *Global Convergence of a New Hybrid Gauss-Newton Structured BFGS Method for Nonlinear Least Squares Problems* <http://www.polyu.edu.hk/ama/staff/xjchen/SIOPT-2010.4.10.pdf>, viitattu 15.8.2014
- [13] S. Gratton, A.S. Lawless, N.K. Nichols, 9/2004. *Approximate Gauss-Newton methods for nonlinear least squares problems*, Numerical analysis report. The University of Reading, Department of Mathematics <http://www.henley.ac.uk/web/FILES/math/09-04.pdf>, viitattu 18.8.2014
- [14] Alfonso Croeze, Lindsey Pittmann, Winnie Reynolds *Solving Nonlinear Least-squares problems with the Gauss-Newton and Levenberg-Marquardt Methods* <https://www.math.lsu.edu/system/files/MunozGroup1%20-%20Paper.pdf> 20.8.2014
- [15] H. P. Gavin, 9/2013 *The Levenberg-Marquardt method for nonlinear least squares curve-fitting problems*, Duke University, Department of Civil and Environmental Engineering <http://people.duke.edu/~hpgavin/ce281/lm.pdf>, viitattu 20.8.2014
- [16] *Optimization and Data Fitting – Nonlinear Least-Squares Problems* <http://www.imm.dtu.dk/~pcha/LSDF/NonlinDataFit.pdf>, viitattu 20.8.2014
- [17] Bucknell university, Department of Physics and Astronomy, Models and fitting http://www.eg.bucknell.edu/physics/ASTR201/IDLTutorial/tutorial_05.html, viitattu 22.8.2014
- [18] National Institute of Standards and Technology, StRD Nonlinear Least Squares Regression Datasets. <http://www.itl.nist.gov/div898/strd/nls/data/nelson.shtml>, viitattu 26.8.2014