

GLOBALI OPTIMOINTI
POSITRONIEMISSIONOTOMOGRFIA-KUVANTAMISEEN
LIITTYVÄSSÄ MALLINTAMISESSA

Kaisa Sederholm

Pro gradu -tutkielma
Tammikuu 2003

TURUN YLIOPISTO
MATEMATIIKAN LAITOS
20014 TURKU

TURUN YLIOPISTO
Matematiikan laitos

SEDERHOLM, KAISA: Globaali optimointi positroniemissiotomografia-
kuvantamiseen liittyvässä mallintamisessa
Pro gradu -tutkielma, 34 s., 9 liites.
Sovellettu matematiikka
Tammikuu 2003

Tutkielmani on tehty Turun valtakunnallisen PET-keskuksen toimeksiannosta. Positroniemissiotomografia (PET) on lääketieteellinen kuvantamismenetelmä. Kuvien tulkintaan liittyy mallintamista, jossa sovelletaan muun muassa pienimmän neliösumman menetelmää. Tarkoitus oli tutkia tätä neliösumman minimointiongelmää globaalina optimointitehtävänä. Työni sisältää siis sovelletun matematiikan alalta mallintamista ja optimointia. Näiden lisäksi tutkielmassa on pohdittu myös hiukan tilastotieteen menetelmien käyttöä tulosten analysoinnissa.

Tutkielmani johdannon jälkeinen luku käsittelee globaalin optimoinnin teoriaa. Siinä esitellään myös muutama testifunktio, joita myöhemmissä luvuissa on käytetty ohjelmien testaamiseen.

Kolmas luku esittelee lokeromalleilla tapahtuvaa PET-mallinnusta, ja siihen liittyviä matemaattisia ongelmia.

Neljännessä luvussa esitetään PET-keskuksella tekemästämme minimirakenteen karkeitusohjelmasta saadut tulokset. Minimilistausten ja piirtämieni kuvien perusteella todetaan, että lokaaleja minimejä ei ole todennäköisesti ole kovin paljon ja että globaali minimi näyttäisi yleensä olevan hyvin laakea.

Viides luku esittelee Törnin ja Viitasen Iterative Topographical Global Optimization (ITGO) algoritmin perusteella kirjoittamani globaalin optimointialgoritmin. Algoritmin todetaan löytävän samat globaalit minimi kuin kolmannen luvun algoritminkin. Lisäksi ITGO toimii huomattavasti paremmin, kuin PET-keskuksella tähän asti käytössä ollut Pivot algoritmi. ITGO algoritmi poistaa ongelmat tuloksen riippuvuudesta alkuasetuksista. Nämä olivat Pivot algoritmin huonoja puolia ja tekivät siitä huonon menetelmän rutiinikäytössä.

Kuudennessa luvussa pohditaan vielä tulosten analysointia tilastotieteen keinoin, ns. bootstrap tekniikkaa käyttäen. Kiinnostavinta on luottamusvälien laskeminen parametrien estimaateille.

Asiasanat: positroniemissiotomografia, mallintaminen, globaali optimointi, lokero-malli, bootstrap.

Sisältö

1	Johdanto	1
2	Globaalin optimoinnin teoriaa	2
2.1	Globaali optimointi	2
2.2	Globaali optimointitehtävä	2
2.3	Ratkaisun löytymisestä	2
2.3.1	Ratkaisun löytämistä helpottavia tekijöitä	2
2.3.2	Ratkaisuun kuluva aika	3
2.3.3	Mikä tulkitaan ratkaisuksi?	4
2.4	Menetelmistä	5
2.4.1	Menetelmien jaottelua	5
2.4.2	Menetelmien sopivuuden tutkiminen ja vertailu	5
3	Positroniemissiotomografia-kuviin liittyvä mallintaminen	9
3.1	Positroniemissiotomografia	9
3.2	Lokeromallit ja minimointiongelma	9
3.2.1	Differentiaaliyhtälöryhmän ratkaiseminen	11
3.2.2	Pienimmän neliösumman ongelma	13
3.3	Yhteenvedo	15
4	Datan kartoitus	16
4.1	Tavoitteet	16
4.2	Algoritmi	16
4.2.1	Ohjelman testaus	16
4.2.2	Minimirakenteen kartoitus	17
5	Ohjelma PET-keskuksen käyttöön	21
5.1	Tavoitteet	21
5.2	Algoritmi	21
5.3	Testaus	22
5.4	Tulokset	24

6	Estimaattien harhaisuus	27
6.1	Keinotekoisien otosjakauman konstruointi	27
6.2	Prosenttipisteiden määrittäminen	29
6.3	Luottamusvälilaskenta	30
6.4	Analyysi	30
	Kirjallisuutta	31

1 Johdanto

Sain PET-keskukselta tehtäväksi tutkia globaalien optimointimenetelmien soveltuvuutta heidän tarpeisiinsa. Aluksi tutkin jonkin aikaa geneettisiä algoritmeja, joiden käyttö päätettiin kuitenkin hylätä. Ne toimivat hyvin ongelmille joiden ratkaisualue ja lokaalien minimien määrä ovat suuria. Meidän minimointiongelmissamme ratkaisualue ei kuitenkaan ole kovin suuri eikä todennäköisesti myöskään lokaalien minimien määrä. Tutkin myös Nelder–Mead simplex -algoritmia. Lopulta kuitenkin päädyimme kokeilemaan Aimo Törnin ja Sami Viitasen Åbo Akademiassa kehittämää ITGO-algoritmia. Olin voinut työssäni myös kokeilla monia eri algoritmeja ja vertailla niitä. Tällöin olisin kuitenkin joutunut itse kirjoittamaan kaikki algoritmit c-kielelle, joten totesin että yhden ohjelman tekemisessä oli riittävästi työtä yhtä Pro gradu -tutkimusta varten.

Pro gradu -työssäni analysoin siis PET-kuvantamiseen liittyvää mallintamisongelmaa ja testaan globaalin ITGO-optimointimenetelmän toimivuutta.

Luvussa 2 käyn läpi globaalin optimoinnin teoriaan liittyviä perusasioita.

Luvussa 3 esittelen PET-kuvantamista sekä siihen liittyvää mallintamista: lokeromalleja, joilla yritetään kuvata aineiden kulkeutumista ja muuttumista elimistössä, differentiaaliyhtälöryhmän ratkaisua ja lopulta pienimmän neliösumman minimointiongelmaa jotka kumpikin liittyvät lokeromallin parametrien ratkaisemiseen.

Luvussa 4 kartoitetaan minimoitavien funktioiden lokaaleja minimejä itseni ja Vesa Oikosen kirjoittaman c-ohjelman avulla. Näiden tulosten perusteella pohditaan, minkälainen rutiinikäyttöön tulevan ohjelman tulisi olla.

Luvussa 5 esitellään ITGO-algoritmi, joka on otettu rutiinikäyttöön PET-keskuksessa. Testaan tekemääni ohjelmaa sekä testifunktioilla että todellisella datalla.

Luvussa 6 esitetään vielä eräs tapa laskea luottamusvälejä tutkittavien parametrien estimaateille. On tarkoitus, että teen tulevaisuudessa tätä menetelmää käyttävän c-ohjelman PET-keskukselle.

2 Globaalien optimoinnin teoriaa

2.1 Globaali optimointi

Globaali optimointi on optimoinnin laji, jolla on hyvin paljon käytännön sovelluksia muun muassa tekniikan, kaupan ja talouden aloilla. Siinä käsitellään monihuippuisia funktioita, ja päämääränä on löytää useasta huipusta korkein (globaali maksimi) tai syvin (globaali minimi). Tämä luku käsittelee globaaliin optimointiin liittyvää teoriaa [1] [2].

2.2 Globaali optimointitehtävä

Puhun jatkossa minimointitehtävistä, koska työni sovellusosa on minimointia. Kaikki tulokset voidaan kuitenkin helposti soveltaa myös maksimointiongelmiin.

Monet ratkaisualgoritmit olettavat tehtävien olevan *rajoittamattomia*:

Olkoon $f : (\mathbb{R}^n \rightarrow \mathbb{R})$. Jos $x^* \in \mathbb{R}^n$ toteuttaa epäyhtälön

$$f(x^*) \leq f(x), \forall x \in \mathbb{R}^n, \quad (1)$$

on se tehtävän ratkaisu eli *optimipiste*. Funktiota f kutsutaan *kohdefunktioksi*. Tehtävän sanotaan olevan *olennaisesti rajoittamaton*, jos se on seuraavan määritelmän mukainen.

Määritellään alue $S \subset \mathbb{R}^n$, joka on kompakti ja konvekksi sekä sisältää globaalien optimien sisäpisteenään. Ratkaisu

$$x^* = \min_{x \in S} f(x) \quad (2)$$

on globaali minimi *sallitussa alueessa* S .

2.3 Ratkaisun löytymisestä

2.3.1 Ratkaisun löytämistä helpottavia tekijöitä

Minimointitehtävän ratkaisemista helpottaa, jos tutkittavasta datasta tiedetään etukäteen joitakin piirteitä. Esimerkiksi tiedot lokaalien minimien määrästä auttavat globaalien minimien löytymistä. Jos minimien täsmällinen määrä tiedetään ja tutkittavasta funktiosta löydetään tämä määrä minimejä,

voidaan varmasti sanoa, että pienin näistä on globaali minimi. Käytännössä lokaalien minimien täsmällistä määrää ei juuri koskaan tiedetä etukäteen. Ongelman ratkaisemista helpottaisi myös tieto kohdefunktion arvosta globaalissa minimipisteessä.

Useissa minimointialgoritmeissa käytetään hyväksi derivaattoja. Funktion oletetaan usein olevan jatkuva ja derivoituva, mikä helpottaa ongelman ratkaisemista. Sovelluksessani näitä oletuksia on pyritty välttämään, mikä on laskenut selvästi ratkaisemiseen mahdollisesti käytettävien algoritmien määrää.

2.3.2 Ratkaisuun kuuluva aika

Soveltavan matematiikan osa-alueena globaali optimointi on kaukana eksaktista matematiikasta, ja voidaankin todistaa, että globaalin optimoinnin ongelmia on mahdoton ratkaista tarkkaan äärellisessä ajassa.

Lause 2.1. Tehtävää (1) ei voida ratkaista äärellisessä ajassa.

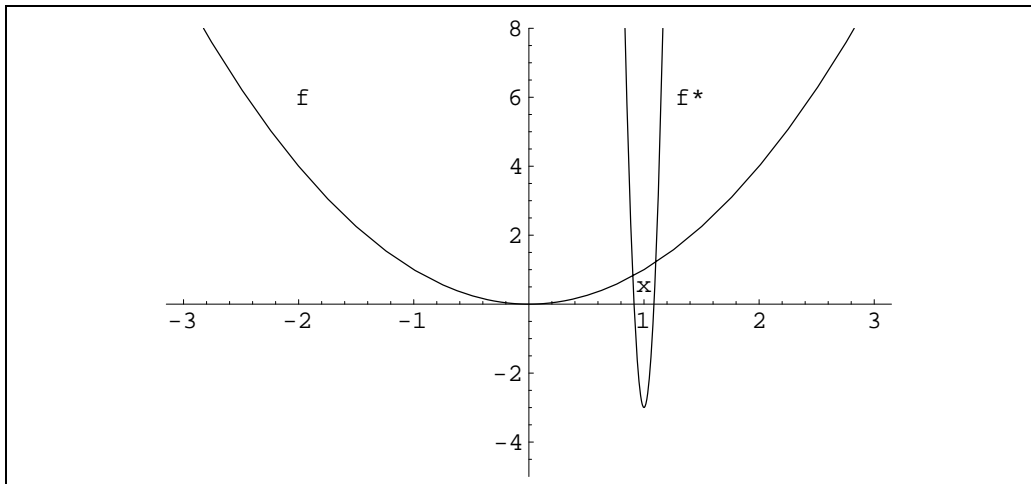
Todistus. Olkoon kohdefunktio f jokin jatkuvasti differentioituva funktio ja $\bar{x} \in \mathbb{R}^n$ mielivaltainen piste. Olkoon B jokin \bar{x} :n ympäristö. Pisteelle \bar{x} voidaan aina määrittellä funktio f^* niin, että

$$g(x) = \begin{cases} f(x) + f^*(x), & \text{kun } x \in B \\ f(x), & \text{kun } x \notin B \end{cases}$$

on jatkuvasti differentioituva, ja $\bar{x} = \min_{x \in \mathbb{R}^n} g(x)$ (kuva 2.1).

Jotta mielivaltaisesta pisteestä \bar{x} voitaisiin varmuudella todeta että se *ei* ole globaali minimi, täytyisi kohdefunktion arvo laskea jokaisessa \bar{x} :n ympäristössä vähintään yhdessä pisteessä. Koska ympäristö voidaan valita mielivaltaisen pieneksi, voidaan todeta, että tehtävä (1) ei ratkea äärellisellä määrällä funktion arvotuksia.

Luvussa 2.3.3 pohditaan sitä, miten tähän tulokseen käytännön ongelmis-
sa suhtaudutaan.



Kuva 2.1. Funktiot f ja f^* .

2.3.3 Mikä tulkitaan ratkaisuksi?

Koska globaalisissa optimointitehtävissä ei päästä tarkkaan ratkaisuun, on optimoijan tyydyttävä globaalin optimin approksimaatioon. On siis keksittävä keinotekoisia ehtoja sille, milloin tyydytään saatuun globaaln optimin estimaattiin. Ohjelmoitaessa nämä ehdot ovat käytettävän algoritmin lopeusehtoja.

Globaali optimointitehtävä (2) ratkeaa, jos löydetään esimerkiksi jomankumman seuraavan joukon alkio ($\varepsilon > 0$ on jokin pieni luku).

Olkoon S konvekksi ja kompakti joukko. Joukoista

$$A_x(\varepsilon) = \{x \in S \mid \|x - x_*\| \leq \varepsilon\} \text{ ja}$$

$$A_f(\varepsilon) = \{x \in S \mid |f(x) - f(x_*)| \leq \varepsilon\}$$

löytyneet alkioit voidaan tulkita globaaln minimipisteen approksimaatioiksi. Mitä pienempi on ε , sitä tarkempaa tulosta haetaan ja sitä tiheämmin sallittua aluetta S pitää tutkia.

2.4 Menetelmistä

2.4.1 Menetelmien jaottelua

Globaalin optimoinnin menetelmät voidaan karkeasti jakaa *deterministisiin* ja *stokastisiin* menetelmiin. Stokastiset eli tilastolliset menetelmät perustavat toimintansa sille, että kohdefunktion arvoja lasketaan satunnaisesti generoiduissa pisteissä. Tällä pyritään siihen, että sallittu alue käydään läpi suhteellisen kattavasti. Tämä on tarpeen koska yleensä globaalin minimin sijainnista ei voida etukäteen päätellä juuri mitään.

Perinteisissä yhden minimin optimointitehtävissä hyvin toimivat deterministiset menetelmät eivät sellaisenaan sovellu globaaliin optimointiin. Toimiakseen ne vaativat kohdefunktiosta tehtyjä lisäoletuksia.

Monet stokastiset menetelmät käyttävät deterministisiä menetelmiä osana algoritmia. Esimerkiksi niin sanotut *multistart-menetelmät* etsivät lokaaleja minimejä eri puolilta ratkaisuavaruutta ja *ryväsmenetelmät* pilkkovat ratkaisualuetta pienempiin osiin ja etsivät lokaaleja minimejä näiltä alueilta.

2.4.2 Menetelmien sopivuuden tutkiminen ja vertailu

Parhaan menetelmän ja sille parhaiden mahdollisten parametrien löytäminen ovat myös optimointitehtäviä. Globaalissa optimoinnissa ei yleensä voida etukäteen tietää, mikä menetelmä M parhaiten sopii mihinkin ongelmaan. Menetelmiä joudutaan käytännössä aina vertailemaan empiirisesti.

Vertailussa täytyy ottaa huomioon, mitä algoritmilta halutaan: tarkkuutta vai tehokkuutta. Tietyn menetelmän toimivuus riippuu sille annetuista parametreista (esim. askelpituus) ja valitusta lopetusehdosta. Lopputulokseen vaikuttavat myös lähtöpisteet ja algoritmin generoimat satunnaisluvut. Kun parametrien arvot pidetään vakiona, voidaan ajatella globaalin minimin f^* estimaatin \hat{f}^* olevan otos jostakin jakaumasta. Ajatellaan, että $g_M(f)$ on \hat{f}^* :n tiheysfunktio ja $G_M(f)$ jakautumafunktio. Funktiolla $g_M(f)$ voidaan olettaa olevan piikkejä lokaalien minimien kohdilla.

Olkoon $h(f)$ hyötyfunktio, joka saa arvoja väliltä $(0, 1)$ sen mukaan, miten hyödyllinen globaalin minimin estimaatti f on. Nyt tietyn menetelmän

hyödyllisyyttä voidaan mitata laskemalla Stiltjesin integraali

$$U_M = \int_{f^*}^{\infty} h(f) dG_M(f),$$

jossa $h(f) \in (0, 1)$ ja $h(f) = 1$, kun $f \in A_f(\varepsilon)$.

Kun lokaalit minimit toteuttavat epäyhtälön $f^* = f_1^* < f_2^* < f_3^* \dots < f_n^*$, voidaan olettaa että funktio $h(f)$ on vähenevä. Jos $h(f) = C$, (C on jokin vakio), niin $U_M = C, \forall M$. Siis kaikki menetelmät tulkitaan yhtä hyviksi. Jos taas

$$h(f) = \begin{cases} 1, & \text{jos } f \in A_f(\varepsilon) \\ 0, & \text{muuten,} \end{cases}$$

niin U_M voidaan tulkita globaalin minimin löytymistodennäköisyydeksi.

Empiirisesti menetelmien vertailun voi toteuttaa seuraavasti. Algoritmi toteutetaan m kertaa, joista m_1 :llä kerralla löydetään globaali minimi. Näistä voidaan laskea suhde m_1/m . Jos sama tehdään eri algoritmeilla, niin kiinnittämällä suhdeluvut (esim. $m_1/m = 0.90, \forall M$) voidaan vertailla eri algoritmien tehokkuutta.

Yksittäisen algoritmin toimivuutta voidaan testata erityisillä testifunktioilla. Näitä ovat mm. Rosenbrockin funktio, Levyn funktio, Griewankin funktio, Braninin funktio ja Shekelin funktio [3].

Esimerkki 2.1. Griewankin funktio

$$g(x) = \frac{x_1^2 + x_2^2}{4000} - \cos(x_1) \cos\left(\frac{x_2}{\sqrt{2}}\right) + 1, \quad x_1, x_2 \in [-10, 10]$$

on kehitetty nimenomaan globaalien optimointialgoritmien testaamiseen. Sillä on ääretön määrä lokaaleja minimejä avaruudessa \mathbb{R}^2 . Kuvassa 2.2 on esitetty funktion $-g$. Minimit näkyvät siis huippuina.

Esimerkki 2.2. Braninin funktio

$$b(x) = \left(1 - 2x_2 + \frac{1}{20} \sin 4\pi x_2 - x_1\right)^2 + \left(x_2 - \frac{1}{2} \sin 2\pi x_1\right)^2, \quad x_1, x_2 \in [-10, 10]$$

saavuttaa globaalin minimin $b^* = 0$ viidessä eri pisteessä. Nämä pisteet ovat

$$\begin{pmatrix} 1 & , & 0 \end{pmatrix} \\ \begin{pmatrix} 0.148696 & , & 0.402086 \end{pmatrix} \\ \begin{pmatrix} 0.402537 & , & 0.287408 \end{pmatrix} \\ \begin{pmatrix} 1.59746 & , & -0.287408 \end{pmatrix} \\ \begin{pmatrix} 1.85130 & , & -0.402086 \end{pmatrix}.$$

Kuvassa 2.3 on funktio b kuvattuna alueella $[0, 2] \times [-0.5, 0.5]$.

Esimerkki 2.3. Shekelin funktiosta

$$s(x) = - \sum_{i=1}^m \frac{1}{(x - A_i)(x - A_i)^T + c_i}, \quad x_i \in [0, 10] \quad i = 1, \dots, 4,$$

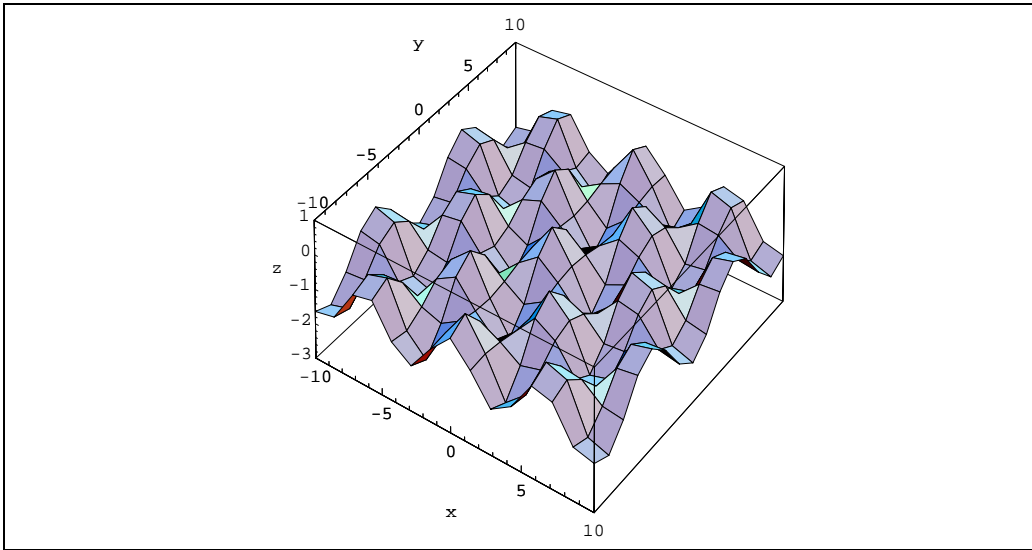
jossa

$$A = \begin{bmatrix} 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 \\ 8 & 8 & 8 & 8 \\ 6 & 6 & 6 & 6 \\ 3 & 7 & 3 & 7 \\ 2 & 9 & 2 & 9 \\ 5 & 5 & 3 & 3 \\ 8 & 1 & 8 & 1 \\ 6 & 2 & 6 & 2 \\ 7 & 3.6 & 7 & 3.6 \end{bmatrix} \text{ ja } c = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.2 \\ 0.4 \\ 0.4 \\ 0.6 \\ 0.3 \\ 0.7 \\ 0.5 \\ 0.5 \end{bmatrix},$$

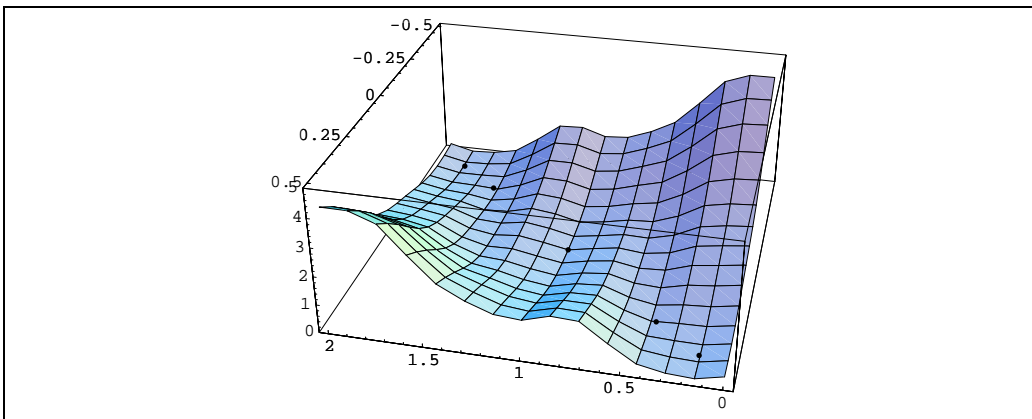
voidaan käyttää kolmea versiota käyttämällä muuttujalle m arvoja 5, 7 tai 10. Merkintä A_i viittaa matriisiin A i :nteen riviin ja merkintä c_i vastaavasti vektorin c i :nteen alkioon. Shekel 5:n, Shekel 7:n ja Shekel 10:n ratkaisut ovat

$$\begin{aligned} m = 5 & \quad s^* = -10.15319967 \quad x^* = (4.00004, 4.00013, 4.00004, 4.00013) \\ m = 7 & \quad s^* = -10.40294056 \quad x^* = (4.00057, 4.00069, 3.99949, 3.99961) \\ m = 10 & \quad s^* = -10.53640981 \quad x^* = (4.00075, 4.00059, 3.99966, 3.99951). \end{aligned}$$

Kaikki kolme ovat siis neljän muuttujan funktioita, ja niillä on lokaaleja minimejä nimiensä mukaisesti 5, 7 ja 10.



Kuva 2.2. Griewankin testifunktio.



Kuva 2.3. Braninin testifunktio.

3 Positroniemissiotomografia-kuviin liittyvä mallintaminen

3.1 Positroniemissiotomografia

Positroniemissiotomografia (PET) on funktionaalinen, kvantitatiivinen kuvantamismenetelmä, jolla tutkitaan erilaisten aineiden kulkeutumista elimistössä. Tähän käytetään niin sanottuja merkkiaineita (radiolääkeaineita), joita saadaan kun mikä tahansa biologinen molekyyli leimataan jollakin radioaktiivisella isotoopilla, joka tuottaa hajotessaan positroneja. Esimerkki leimatusta yhdisteestä on radioaktiivisella fluorilla leimattu deoksiglukoosi, jota käytetään tutkittaessa glukoosin kulkeutumista elimistössä. PET-kamera havaitsee elimistössä tapahtuvan radioaktiivisen aineen hajoamisen. Mittaamalla radioaktiivisuutta jollakin mielenkiintoalueella (region of interest, ROI) ja tekemällä hajoamiskorjaus tutkimuksen alkuhetkeen (puoliintumisajat tunnetaan) voidaan laskea merkkiaineen pitoisuus ROI-alueella (mol/ml). Mittauksia tehdään yleensä tiuhemmin tutkimuksen alussa ja harvemmin loppupäässä. Sovituksen helpottamiseksi aikapisteet (t_1, t_2, \dots, t_m) interpoloidaan joskus tasavälisiksi.

Käytännön laskuissa ei käytetä merkkiaineen pitoisuusarvoja, vaan radioaktiivisuuden määriä. Hajoamiskorjattuna aktiivisuuden määrä on suoraan verrannollinen merkkiaineen pitoisuuteen. Tällöin yksikkö on kBq/ml , jossa Bq kuvaa hajoamisten määrää sekunnissa. Merkitään mitattua radioaktiivisuutta (radioaktiivisen aineen pitoisuutta ROI-alueella) C_{PET}^ε :lla. Epsilon viittaa siihen, että mitatuissa tuloksissa on mukana kohinaa.

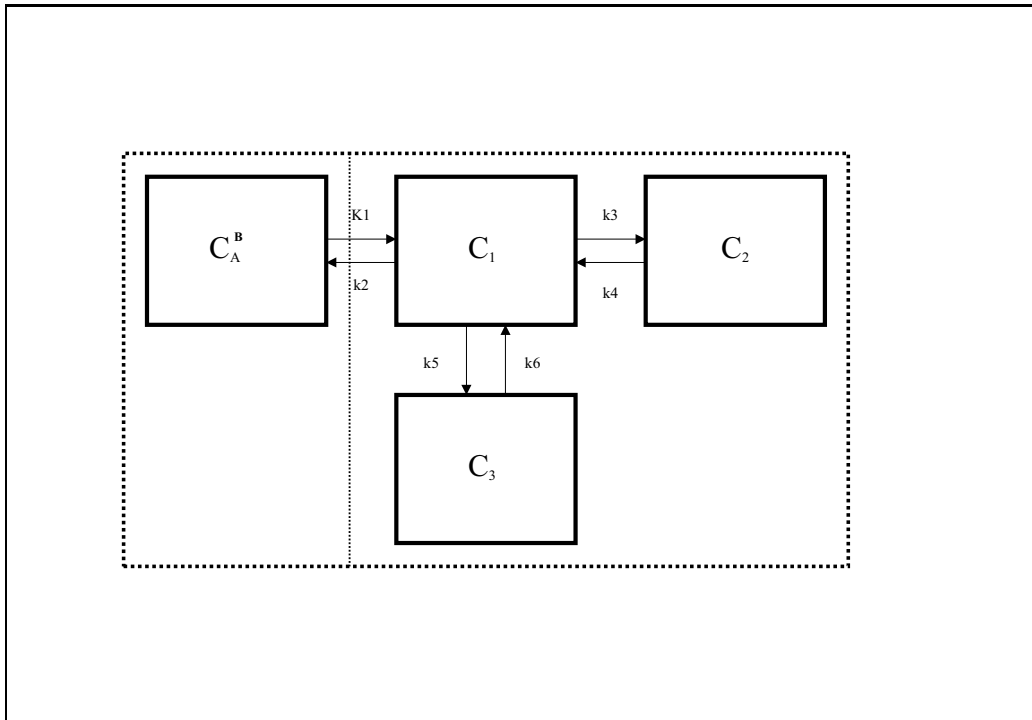
3.2 Lokeromallit ja minimointiongelma

Selvitetään nyt lähemmin, minkälaisien minimointiongelmien kanssa joudutaan tekemisiin PET-dataa analysoitaessa. PET-tutkimuksissa mitataan merkkiaineen pitoisuus veressä ja kudoksessa. Mielenkiinnon kohteena ovat aineen kulkeutuminen ja muutokset, joita kuvataan lokeromalleilla. Merkkiaineen pitoisuus veressä saadaan diskreettinä aineistona mittaamalla veri-

näytteitä ja kudoksen kokonaispitoisuuksia PET-kuvista.

Tarkastellaan lokeromalleja seuraavan esimerkin 3.1 avulla.

Esimerkki 3.1. Kuvassa 3.1 on nelilokeromalli, jossa vasemman puoleisin lokero on verilokero ja kolme muuta lokeroa ovat kudoslokeroja.



Kuva 3.1. Elimistössä tapahtuvaa aineenvaihduntaa kuvaava nelilokeromalli.

Muuttujat C_A^B ja C_1 kuvaavat vapaana olevan merkkiaineen pitoisuutta valtimoverisuonissa (arterial blood) ja kudoksessa. C_2 kuvaa mielenkiinnon kohteena oleviin reseptoreihin sitoutuneen merkkiaineen pitoisuutta. C_3 kuvaa sellaista sitoutumista kudoksen muihin rakenteisiin, jota ei pystytä syrjäyttämään (epäspesifinen sitoutuminen). Parametreja K_1, k_2, \dots, k_6 kutsutaan siirtymänopeusvakioiksi, ja ne kuvaavat aineen siirtymänopeutta lokerosta toiseen. K_1 :n yksikkönä käytetään $ml/min * ml$ tai $ml/s * ml$. Muiden siirtymänopeusvakioiden $k_2 \dots k_6$ yksikkönä käytetään $1/min$ tai $1/s$. Aineiden siirtymät ovat verrannollisia lähtölokeron kokoon, jolloin siirtymänopeusva-

kio toimii verrannollisuuskertoimena. Kaikki nopeusvakiot ovat ajan suhteen vakioita.

Yleisimmissä lokeromalleissa on kaksi–neljä lokeroa. Yksilokeromalli voi teoriassa kuvata radioaktiivisuuden hajoamista tai aineen poistumista verestä.

Merkitään esimerkin 3.1 kokonaiskudospitoisuutta seuraavasti $C_T = C_1 + C_2 + C_3$. Pitoisuuksien muutoksia kuvaa differentiaaliyhtälö (DY) -ryhmä

$$\frac{dC_1(t)}{dt} = K_1 C_A^B(t) - (k_2 + k_3 + k_5)C_1(t) + k_4 C_2(t) + k_6 C_3(t) \quad (3)$$

$$\frac{dC_2(t)}{dt} = k_3 C_1(t) - k_4 C_2(t) \quad (4)$$

$$\frac{dC_3(t)}{dt} = k_5 C_1(t) - k_6 C_3(t). \quad (5)$$

Yhtälöryhmästä saadaan ratkaistua C_1 , C_2 ja C_3 , ja näistä voidaan laskea C_T . Kun C_T :hen lisätään veressä havaitut pitoisuudet, saadaan arvo muuttujalle C_{PET} . Vertaamalla tätä PET-kuvista saatuihin todellisiin arvoihin, saadaan varsinainen minimointiongelma, jossa muuttujina ovat parametrit K_1 , k_2 , \dots , k_6 ja verivolyymi V_B . Näille pyritään etsimään arvot, joilla saadaan paras sovitus. PET-kuvista saadun datan kohinaisuus vaikeuttaa sovituksen tekemistä, ja täydellistä sovitusta ei tietenkään voida löytää.

3.2.1 Differentiaaliyhtälöryhmän ratkaiseminen

Esimerkin 3.1 differentiaaliyhtälöryhmän (yhtälöt (3)–(5)) ratkaisutapoja on monia, mutta esittelen tässä tavan, jolla ongelma tällä hetkellä ratkotaan Turun PET-keskuksessa [10].

Yhtälöryhmän yhtälöt (3)–(5) integroidaan puolittain hetkestä 0 hetkeen T, jolloin saadaan yhtälöt:

$$\begin{aligned}
C_1(T) - C_1(0) &= K_1 \int_0^T C_A^B(t) dt - (k_2 + k_3 + k_5) \int_0^T C_1(t) dt \\
&\quad + k_4 \int_0^T C_2(t) dt + k_6 \int_0^T C_3(t) dt
\end{aligned} \tag{6}$$

$$C_2(T) - C_2(0) = k_3 \int_0^T C_1(t) dt - k_4 \int_0^T C_2(t) dt \tag{7}$$

$$C_3(T) - C_3(0) = k_5 \int_0^T C_1(t) dt - k_6 \int_0^T C_3(t) dt. \tag{8}$$

Tiedetään, että pitoisuudet $C_1(t)$, $C_2(t)$ ja $C_3(t)$ ovat 0 hetkellä $t = 0$. Nä-mä voidaan sijoittaa yhtälöihin. Oletetaan myös, että aikapisteet (t_1, t_2, \dots, t_m) on interpoloitu tasavälisiksi.

Lokeroon n liittyvää integraalia voidaan approksimoida implisiittisesti Adamsin ja Moultonin toisen kertaluvun menetelmällä (trapetsimenetelmä) [8]

$$\int_0^T C_n(t) dt = \int_0^{T-\Delta t} C_n(t) dt + \frac{\Delta t}{2} C_n(T - \Delta t) + \frac{\Delta t}{2} C_n(T), \tag{9}$$

jossa integraali $\int_0^{T-\Delta t} C_n(t) dt$ lasketaan puolisuunnikasmenetelmää käyttäen. Sijoittamalla (9) yhtälöihin (7) ja (8) saadaan

$$C_2(T) = \frac{k_3 \int_0^T C_1(t) dt - k_4 \int_0^{T-\Delta t} C_2(t) dt - k_4 \frac{\Delta t}{2} C_2(T - \Delta t)}{1 + \frac{\Delta t}{2} k_4} \tag{10}$$

ja

$$C_3(T) = \frac{k_5 \int_0^T C_1(t) dt - k_6 \int_0^{T-\Delta t} C_3(t) dt - k_6 \frac{\Delta t}{2} C_3(T - \Delta t)}{1 + \frac{\Delta t}{2} k_6}. \tag{11}$$

Kun (9), (10) ja (11) sijoitetaan (6):een, saadaan $C_1(T)$:lle lauseke

$$\begin{aligned}
C_1(T) = & \frac{K_1 \int_0^T C_A^B(t) dt}{D} \\
& - \frac{\left(k_2 + \frac{k_3}{1 + \frac{\Delta t}{2} k_4} + \frac{k_5}{1 + \frac{\Delta t}{2} k_6}\right) \left(\int_0^{T-\Delta t} C_1(t) dt + \frac{\Delta t}{2} C_1(T - \Delta t)\right)}{D} \\
& + \frac{\frac{k_4}{1 + \frac{\Delta t}{2} k_4} \left(\int_0^{T-\Delta t} C_2(t) dt + \frac{\Delta t}{2} C_2(T - \Delta t)\right)}{D} \\
& + \frac{\frac{k_6}{1 + \frac{\Delta t}{2} k_6} \left(\int_0^{T-\Delta t} C_3(t) dt + \frac{\Delta t}{2} C_3(T - \Delta t)\right)}{D},
\end{aligned}$$

jossa

$$D = 1 + \frac{\Delta t}{2} \left(k_2 + \frac{k_3}{1 + \frac{\Delta t}{2} k_4} + \frac{k_5}{1 + \frac{\Delta t}{2} k_6}\right).$$

Koska pitoisuudet hetkellä 0 tiedetään (yleensä $C_1(0) = C_2(0) = C_3(0) = 0$), voidaan nyt laskea $C_1(\Delta t)$ ja tämän avulla $C_2(\Delta t)$ ja $C_3(\Delta t)$. Näin saadaan kudoslokeroiden pitoisuudet laskettua yksi aikapiste kerrallaan. Laskeamalla C_1 :n, C_2 :n ja C_3 :n arvot yhteen jokaisessa aikapisteessä, saadaan C_T :lle diskreetti estimaatti.

3.2.2 Pienimmän neliösumman ongelma

Jotta laskettuja ja mitattuja pitoisuuksia voitaisiin verrata, täytyy C_T :hen vielä lisätä pitoisuudet kudoksen sisältämässä veressä. Samalla verivolyyymi (veren tilavuusosuus kuvatusta alueesta) V_B saadaan liitettyä malliin. Yhtälössä

$$C_{PET}(t) = (1 - V_B)C_T(t) + V_B r_A C_A^B(t) + V_B (1 - r_A) C_V^B(t)$$

r_A on valtimoiden tilavuusosuus verisuonistosta V_B ja $1 - r_A$ siis laskimoiden tilavuusosuus. $C_{PET}(t)$ kuvaa kokonaispitoisuutta ROI-alueella ajan funktiona, ja sen kuvaajaa kutsutaan *aika-aktiivisuuskäyräksi*.

Pitoisuus C_V^B laskimoverisuonissa (venous blood) voidaan laskea kaavasta

$$C_V^B = C_A^B - \frac{dC_T(t)/dt}{f},$$

jos veren läpivirtaus (perfuusio) f tunnetaan.

Esimerkin 3.1 kokonaiskudospitoisuuden muutos saadaan laskemalla yhteen differentiaaliyhtälöt (3)–(5).

$$\frac{dC_T(t)}{dt} = K_1 C_A(t) - k_2 C_1(t)$$

Käytännössä f tunnetaan vain niin sanotuissa flow-tutkimuksissa. Usein tehdäänkin oletus, että pitoisuus laskimoissa on sama kuin pitoisuus valtimoissa.

Pienimmän neliösumman ongelmaan päädytään, kun laskettuja pitoisuusarvoja C_{PET} yritetään sovittaa mitattuihin arvoihin C_{PET}^ε . Jos aikapisteitä on m kappaletta, saadaan minimoitava lauseke

$$\min \sum_{i=0}^m | C_{PET}^\varepsilon(i \cdot \Delta t) - C_{PET}(i \cdot \Delta t) |^2 .$$

Mitä lähempänä nollaa neliösumman arvo on, sitä parempi sovitus saadaan. Kohinaisen datan takia täydellistä sovitusta on mahdotonta löytää, ja minimoinnissa pyritäänkin vain löytämään mahdollisimman pieni arvo neliösummalle. Jos täydellinen sovitus voitaisiin saavuttaa, se tietenkin helpottaisi minimin löytymistä, koska kohdefunktion arvo minimipisteessä olisi tunnettu (vrt. lukuun 2.3.1). Ongelma on *olennaisesti rajoittamaton, epälineaarinen* ja mahdollisesti *monihuippuinen* optimointitehtävä.

Nyt muuttujina ovat parametrit $K_1 - k_6$ ja V_B . Tehtävä on siis löytää avaruudesta \mathbb{R}^7 piste $(\hat{K}_1, \dots, \hat{k}_6, \hat{V}_B)$, joka minimoi neliösumman arvon. Muuttujille annetaan aina seuraavanlaiset rajoitukset:

$$\begin{aligned} a_1 &\leq K_1 \leq b_1 \\ a_2 &\leq k_2 \leq b_2 \\ &\vdots \\ a_6 &\leq k_6 \leq b_6 \\ a_7 &\leq V_B \leq b_7, \end{aligned}$$

joissa $a_i \leq b_i \forall i$. Yleensä $(a_i, b_i) \in [0, 1]^2$, kun $i = 1, \dots, 7$, mutta periaatteessa rajat voivat olla melkein mitä vain.

3.3 Yhteenveto

Tutkittavan asian laadusta riippuu se, mitä merkkiainetta kulloinkin käytetään ja mihin lokeromalliin päädytään. Minimoitava lauseke vaihtelee siis täysin tutkimuksesta riippuen. Minimointiongelma on yleensä työläs, koska estimaattia C_{PET} laskettaessa parametrit $K_1 \dots k_n$ ($n \in (2, 7)$) täytyy olla kiinnitetty. Joudumme siis tekemään lähtöarvauksia parametrien estimaateille ja ratkaisemaan DY-ryhmän joka kerta erikseen, jotta saadaan laskettua neliösummafunktion arvo.

4 Datan kartoitus

4.1 Tavoitteet

Lokaalien minimien kartoitusta varten tehtiin erillinen testiohjelma, jonka avulla yritettiin saada kuvaa edellisessä luvussa esitetynlaisten pienimmän neliösumman ongelmien minimirakenteista. Testidatana käytettiin 12 eri tutkimuksen mittauksia, jotka kukin sisälsivät tuloksia kahdeksalta eri aivoalueelta [11].

4.2 Algoritmi

PET-keskuksen mallintajan Vesa Oikosen kanssa yhteistyössä tehty `globsrch.c`-niminen testiohjelma (liite 1) arpoo sallitulle alueelle R pistettä tasaisen jakauman mukaan, ja käynnistää lokaalin optimointialgoritmin jokaisessa pisteessä. Jokainen löydetty minimi talletetaan, mikäli se poikkeaa edellisistä talletetuista määrättyllä tarkkuudella. Lokaalina minimointimenetelmänä `globsrch.c` käyttää Powellin algoritmia [4]. Löydetyistä lokaaleista minimeistä algoritmi valitsee parhaan ja kutsuu vielä kerran Powell-algoritmia tässä pisteessä. Ohjelmalla saadaan kattava kuva minimirakenteesta, kunhan arvottavien pisteiden lukumäärä R asetetaan riittävän suureksi.

4.2.1 Ohjelman testaus

Ohjelman toimivuutta testattiin Griewankin funktiolla (ks. luku 2.4.2). Funktion globaali minimi on origossa ja lokaaleja minimejä on tasaisesti sen ympärillä.

Tuhannella alkupisteellä testiohjelma löysi alueelta $[-10, 10]^2$ 21 minimiä. Näistä minimeistä neljä löytyi sallitun alueen kulmista. Loput 17 ovat tällä alueella olevia funktion varsinaisia minimejä. Kun alkupisteiden määrän tiputti sataan, ohjelma löysi edelleen funktion varsinaiset minimiä mutta ei kulmissa olevia minimejä (taulukko 4.1).

Ohjelma kuitenkin selvästi löytää kaikki testifunktion olennaiset lokaalit minimiä ja osaa valita näistä oikean globaalin minimin.

Taulukko 4.1. Alueelta $[-10, 10]^2$ löytyneet Griewankin testifunktion lokaalit minimit .

Powell global search (1000 points)			Powell global search (100 points)		
Found 21 minimum			Found 17 minimum		
6.280045e+00	-8.876889e+00	=> 2.958416e-02, 5.60%	-3.140023e+00	-4.438445e+00	=> 7.396040e-03, 7 %
3.140023e+00	4.438445e+00	=> 7.396040e-03, 5.00%	9.420068e+00	-4.438445e+00	=> 2.712538e-02, 5 %
6.280045e+00	8.876889e+00	=> 2.958416e-02, 5.50%	-1.016909e-08	-5.719908e-09	=> 0.000000e+00, 8 %
6.280046e+00	-3.489096e-09	=> 9.864672e-03, 8.00%	6.280046e+00	-1.407106e-08	=> 9.864672e-03, 6 %
-9.420068e+00	-4.438445e+00	=> 2.712538e-02, 5.50%	3.140023e+00	4.438445e+00	=> 7.396040e-03, 8 %
-3.140023e+00	-4.438445e+00	=> 7.396040e-03, 6.70%	3.140023e+00	-4.438445e+00	=> 7.396040e-03, 10 %
-6.280046e+00	-1.826308e-09	=> 9.864672e-03, 6.20%	5.768872e-10	8.876889e+00	=> 1.971949e-02, 6 %
-6.280045e+00	8.876889e+00	=> 2.958416e-02, 6.30%	-6.280046e+00	-6.710627e-09	=> 9.864672e-03, 6 %
3.140023e+00	-4.438445e+00	=> 7.396040e-03, 6.40%	-6.280045e+00	-8.876889e+00	=> 2.958416e-02, 5 %
-6.280045e+00	-8.876889e+00	=> 2.958416e-02, 5.80%	-6.280045e+00	8.876889e+00	=> 2.958416e-02, 5 %
-9.420068e+00	4.438445e+00	=> 2.712538e-02, 4.60%	6.280045e+00	-8.876889e+00	=> 2.958416e-02, 1 %
-9.578720e-09	8.876889e+00	=> 1.971949e-02, 4.40%	-9.420068e+00	4.438445e+00	=> 2.712538e-02, 5 %
9.420068e+00	4.438445e+00	=> 2.712538e-02, 5.30%	-3.140023e+00	4.438445e+00	=> 7.396040e-03, 7 %
1.837735e-09	-1.063502e-08	=> 0.000000e+00, 6.80%	9.420068e+00	4.438445e+00	=> 2.712538e-02, 5 %
-3.140023e+00	4.438445e+00	=> 7.396040e-03, 6.90%	6.280045e+00	8.876889e+00	=> 2.958416e-02, 5 %
9.420068e+00	-4.438445e+00	=> 2.712538e-02, 4.40%	-9.420068e+00	-4.438445e+00	=> 2.712538e-02, 4 %
2.536455e-11	-8.876889e+00	=> 1.971949e-02, 5.90%	1.748457e-10	-8.876889e+00	=> 1.971949e-02, 7 %
-1.000000e+01	-1.000000e+01	=> 1.641837e+00, 0.10%	Best fit:		
-1.000000e+01	1.000000e+01	=> 1.641837e+00, 0.20%	-1.016909e-08	-5.719908e-09	=> 0.000000e+00
1.000000e+01	1.000000e+01	=> 1.641837e+00, 0.10%			
1.000000e+01	-1.000000e+01	=> 1.641837e+00, 0.30%			
Best fit:					
1.837735e-09 -1.063502e-08 ==> 0.000000e+00					

4.2.2 Minimarakenteen kartoitus

Testidataan sovitettiin esimerkkiä 3.1 vastaavan nelilokeromallin mukaiset parametrit $K_1, k_2, k_3, k_4, k_5, k_6$ ja V_B . Parametrit k_5 ja k_6 asetettiin nolliksi, koska käytännössä epäspesifiä sitoutumista on vaikea erottaa sitoutumattoman merkkiaineen kudoskonsentraatiosta. Parametri V_B asetettiin arvoon 0.035 syystä, joka selitetään myöhemmin tässä luvussa. Sallitun alueen rajoiksi asetettiin $[0, 0.7], [0, 0.2], [0, 0.3], [0, 0.3]$ ja lähtöpisteiden määräksi 100. Tulokset tutkimusten 1 ja 5 *frontal cortex*- ja *thalamus*-aivoalueiden minimirakenteista on esitetty liitteissä 2 ja 3. Tulostuksen ensimmäinen rivi kertoo, mille aivoalueelle sovitus on tehty (*fcx=frontal cortex, tha=thalamus*), toinen rivi käytetyn lokaalin optimointimenetelmän ja kolmas rivi löydettyjen lokaalien minimien lukumäärän. Jokainen löydetty minimi on listattu kukin rivilleen niin, että ensimmäiset seitsemän saraketta ovat parametrien $K_1, k_2, k_3, k_4, k_5, k_6$ ja V_B arvot tässä pisteessä ja kahdeksas sarake neliösummafunktion sama arvo tässä pisteessä. Kaksi viimeistä saraketta kertovat vielä

lokaalin minimin löytymiskerrat ja -osuudet. Viimeisellä rivillä on parhaan sovituksen parametrien arvot eli neliösummafunktion globaali minimi.

Tutkimuksessa 1 *frontal cortex* -alueelta löytyi 43 lokaalia minimiä, *thalamus*-alueelta 36 ja tutkimuksessa 5 vastaavasti 39 ja 29 minimiä. Muissa kymmenessä tutkimuksessa lokaalien minimien lukumäärä vaihteli *frontal cortex* -alueella välillä [32, 47] ja *thalamus*-alueella välillä [28, 40]. Kuitenkin esimerkiksi *frontal cortex* -alueella lokaali optimointialgoritmi löysi kaikissa 12 tutkimuksessa globaalin minimin n. 40 – 60%:lla suorituskerroista, kun taas suurin osa muista minimeistä löytyi vain kerran. Yksinkertaisin selitys useille kerran löytyneille minimeille olisi lokaalin optimointialgoritmin ennenaikainen pysähtyminen, mutta ei myöskään voida sulkea pois mahdollisuutta, että minimoitavassa funktiossa todella on tällaisia pieniä piikkejä. Molemmilta aivoalueilta löydettiin aina myös muutama minimi, joka löytyi 2–8 kertaa. Nämä minimi ovat usein varsin lähellä globaalia minimiä.

Cerebellum-alueelle sovitettiin kaksilokeromalli, jonka sovitettavina parametreina oli K_1 ja K_1/k_2 . Näille annettiin rajat $[0, 0.7]$ ja $[0, 10]$. Parametrit k_3-k_6 asetettiin nolliksi ja V_B arvoon 0.035. Globaali minimi löytyi pisteessä (0.2935, 4.2254) (taulukko 4.2). Lisäksi algoritmi löysi yhden lokaalin minimin kerran sadasta Powellin suorituskerrosta. Tämä piste on hyvin lähellä globaalia minimiä, ja voi hyvin johtua Powellin algoritmin ennenaikaisesta pysähtymisestä.

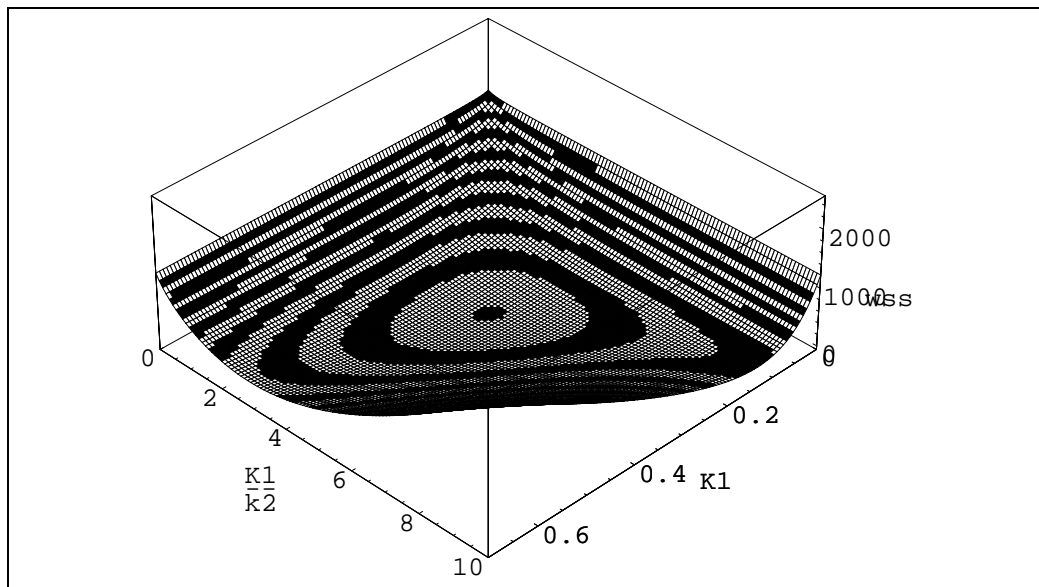
Taulukko 4.2. Tutkimuksen 1 *cerebellum*-alueen sovituksen lokaalit minimi.

```
cer avg All
Powell global search (100 points)
Found 2 minimum
 2.934681e-01 4.225429e+00 0.0e+00 0.0e+00 0.0e+00 0.0e+00 3.5e-02 => 1.446269e+01, 99.00 %
 2.913684e-01 4.232774e+00 0.0e+00 0.0e+00 0.0e+00 0.0e+00 3.5e-02 => 1.448391e+01,  1.00 %
Best fit:
 2.934681e-01 4.225429e+00 0.0e+00 0.0e+00 0.0e+00 0.0e+00 3.5e-02 ==> 1.446269e+01
```

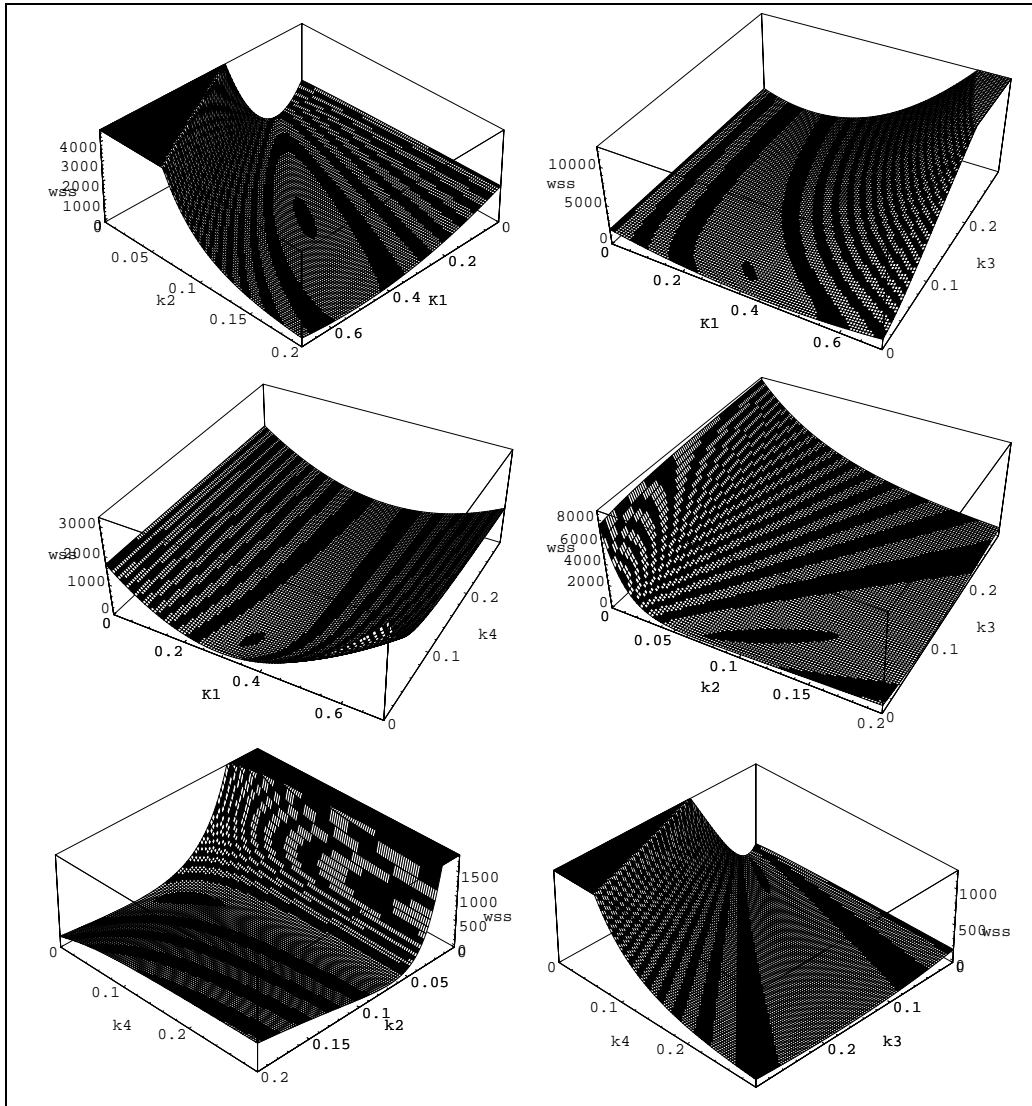
Testiohjelman olin ohjelmoinut myös mahdollisuuden tulostaa tiedostoon matriisit kohdefunktion arvoista, kun kahden parametrin arvojen an-

netaan muuttua muiden parametrien ollessa asetettuna globaaliin minimiarvoonsa. Näistä matriiseista pystyin Mathematica-ohjelmalla piirtämään kolmiulotteisia kuvia kohdefunktion käyttäytymisestä globaalin minimin ympäristössä. Tein aluksi kymmenen kolmiulotteista kuvaa, joissa aina kolme parametria oli asetettu niiden globaalissa minimissä saamiin arvoihin. Kuvista näki selvästi, ettei verivolyymin arvolla ollut juurikaan vaikutusta muiden parametrien arvoihin. Tämän huomion jälkeen testiohjelma käynnistettiin uudelleen asetuksella $V_B = 0.035$ (aivojen massan veriosuus on keskimäärin 3.5 prosenttia, mutta vaihtelee aivoalueittain). Tämän muutoksen jälkeen saatiin globaaliksi minimipisteeksi $(0.3369, 0.0979, 0.0306, 0.0327, 0, 0, 0.035)$. Kuva 4.2 esittää tämän globaalin minimin ympäristöjä kahden parametrin suhteen ja kuva 4.1 *cerebellum*-alueen sovituksen neliösummafunktiota.

Kuvista nähdään, että globaali minimi on yleensä hyvin laakea. Mitään selviä piikkejä ei ole havaittavissa. Myöskään algoritmin löytämiä lokaaleja minimejä ei kuvista voi nähdä. Joko minimit ovat siis ”hyvin pieniä kuoppia isossa kuopassa”, tai nämä lokaalit minimit eivät ole todellisia lainkaan. Minimien loivuus saattaa vaikeuttaa globaalin optimipisteen tarkkaa löytämistä ja vaikuttaa globaalin optimointimenetelmän lopetusehdon valintaan.



Kuva 4.1. Tutkimuksen 1 *cerebellum*-alueen sovituksen neliösummafunktio.



Kuva 4.2. Tutkimuksen 1 *frontal cortex* -alueen sovituksen globaalien minimien ympäristö kolmiulotteisina esityksinä.

Tuloksista voidaan nähdä, että jos suoritetaan Powellin algoritmi kerran satunnaisesti arvotussa pisteessä, niin globaali optimi löydetään korkeintaan noin 60 prosentin todennäköisyydellä. Todelliseen käyttöön otettavaksi optimointialgoritmiksi ei siis riitä aivan näin yksinkertainen toteutus.

5 Ohjelma PET-keskuksen käyttöön

5.1 Tavoitteet

Käytössä olevan globaalin optimointialgoritmi Pivot:n [12] tilalle haluttiin luotettavampi algoritmi, joka mahdollisimman varmasti löytää globaalin minimin. Koska datassa saattaa olla lokaaleja minimejä, minimointialgoritmin täytyy olla globaali. Ratkaisuavaruutta ei ole tarkoitus kartoittaa yhtä kattavasti kuin glosrch.c-ohjelmassa, joten lopullisesta ohjelmasta oletettiin tulevan tätä ohjelmaa tehokkaampi. Kuitenkin PET-keskukselle ohjelman tehokkuutta tärkeämpää on tulosten tarkkuus. Ohjelman kestolle asetetaan vain kohtuullisia rajoituksia.

Yksi Pivot-algoritmin ongelma on ollut se, että ratkaisu on riippunut rajoituksista. Samoja ongelmia on ollut myös askelpituuden ja alkuarvausten kanssa. Asetuksia on siis pitänyt muuttaa erikseen jokaiselle mallille sopivaksi. Huonoilla asetuksilla algoritmi on toiminut hitaasti tai epäluotettavasti. Nämä ominaisuudet tekevät Pivot-algoritmista huonon vaihtoehdon rutiinikäyttöön. Uuden algoritmin toivottiin poistavan näitä ongelmia.

Koska tulosten halutaan olevan toistettavissa, tehtiin lähtöpisteitä arpovasta funktiosta pseudosatunnainen. Se siis arpoo samat pisteet jokaisella ohjelman suorituskeralla, jollei siemenlukua vaihdeta.

5.2 Algoritmi

Tekemäni ohjelma perustuu Törnin ja Viitasen ITGO (iterative topographical global optimization) -algoritmiin [13] [14] [15].

Ohjelma (liite 4) arpoo pseudosatunnaisella algoritmilla N pistettä sallitulle alueelle ja muodostaa näiden pisteiden etäisyyksistä symmetrisen matriisin. Pisteiden määrittelyä olevan *topografinen minimi*, jos kohdefunktion arvo pisteessä on pienempi kuin K :ssa lähimmässä naapuripisteessä. Artikkelissa [13] päädyttiin käyttämään arvoa $K = 3$, joten teimme tämän asetuksen myös omaan ITGO-algoritmiimme.

Lukua N ei voida valita kovin suureksi, koska suuren $N \times N$ -matriisin talletus vie liikaa muistia. Tästä syystä teimme algoritmista iteratiivisen. Kun

ensimmäisestä pistejoukosta on löydetty n kappaletta topografisia minimejä, arvotaan uudelleen $N - n$ kappaletta uusia pisteitä sallitulle alueelle, lasketaan etäisyysmatriisi ja etsitään topografiset minimi joukosta, johon kuuluu edellisen iteraatiokierroksen topografiset minimi ja nyt arvotut $N - n$ pistettä. Iteraatioiden määrä asetettiin riippumaan suoraan sovitettavien parametrien lukumäärästä.

Viimeisen iteraatiokierroksen jälkeen senhetkisissä topografisissa minimipisteissä käynnistetään lokaali optimointialgoritmi (Powell). Paras löydettyistä minimeistä talletetaan. Koska minimien oletetaan olevan loivia, suoritetaan lokaali optimointialgoritmi lopuksi kerran niin, että lopetusehto vaatii hyvin suurta tarkkuutta kohdefunktion arvossa.

5.3 Testaus

Algoritmi testattiin aluksi kolmella eri testifunktiolla: Griewank, Branin ja Shekel 10 (ks. luku 2.4.2). Näistä kahdessa ensimmäisessä on kaksi muuttujaa ja viimeisessä neljä.

Taulukko 5.1. Griewankin testifunktion tuloksia.

```
The global minimum of Griewanks testfunction:
6.101970543661187e-11 -2.558655134239017e-09 ==> 0.000000000000000e+00
The global minimum of Griewanks testfunction:
-1.024864241659318e-08 -1.056061003841518e-08 ==> 0.000000000000000e+00
The global minimum of Griewanks testfunction:
-5.949090592070857e-09 -3.286708825904778e-09 ==> 0.000000000000000e+00
The global minimum of Griewanks testfunction:
-1.944552535487734e-09 -1.863497058202588e-09 ==> 0.000000000000000e+00
The global minimum of Griewanks testfunction:
7.326239530909778e-11 -1.587887236725306e-09 ==> 0.000000000000000e+00
The global minimum of Griewanks testfunction:
-6.524084433516304e-09 -1.910995216934453e-09 ==> 0.000000000000000e+00
The global minimum of Griewanks testfunction:
-5.905005247830130e-09 -1.165148165018476e-10 ==> 0.000000000000000e+00
The global minimum of Griewanks testfunction:
-1.862861293894530e-11 -1.164410608606219e-10 ==> 0.000000000000000e+00
The global minimum of Griewanks testfunction:
7.254280567687923e-11 -1.163694808683170e-10 ==> 0.000000000000000e+00
The global minimum of Griewanks testfunction:
6.059266466769578e-11 -6.375686239224425e-09 ==> 0.000000000000000e+00
```

Griewankin testifunktion globaali minimipiste on origossa, jossa funktio saa arvon 0. Ohjelma löytää näiden kymmenen otoksen perusteella globaalin minimin algoritmin lopetusehdon määräämällä tarkkuudella. Muuttujien x_1 ja x_2 arvot ovat vähintään seitsemän desimaalin tarkkuudella oikein.

Taulukko 5.2. Braninin testifunktion tuloksia.

```

The global minimum of Branins testfunction:
4.025370514345005e-01  2.874076004692598e-01 ==> 5.238529448733282e-32
The global minimum of Branins testfunction:
1.486957174391270e-01  4.020864188176334e-01 ==> 3.851859888774472e-33
The global minimum of Branins testfunction:
1.597463439202156e+00  -2.874077921117283e-01 ==> 4.691753315289879e-27
The global minimum of Branins testfunction:
1.851304602842500e+00  -4.020865977219297e-01 ==> 1.602373713730180e-31
The global minimum of Branins testfunction:
1.486957174391562e-01  4.020864188176263e-01 ==> 4.064430939712313e-27
The global minimum of Branins testfunction:
1.486957174396056e-01  4.020864188173789e-01 ==> 1.318620859811934e-24
The global minimum of Branins testfunction:
1.597463439202190e+00  -2.874077921117525e-01 ==> 1.540743955509789e-30
The global minimum of Branins testfunction:
1.851304602842493e+00  -4.020865977223350e-01 ==> 6.903146753075728e-25
The global minimum of Branins testfunction:
1.851304602842495e+00  -4.020865977219320e-01 ==> 1.602404528609290e-28
The global minimum of Branins testfunction:
1.000000168859038e+00  -1.231036522266674e-07 ==> 4.217695050845942e-27

```

Braninin testifunktion globaali minimi saavutetaan pisteissä $(1, 0)$, $(0.148696, 0.402086)$, $(0.402537, 0.287408)$, $(1.59746, -0.287408)$ ja $(1.85130, -0.402086)$, joissa funktio saa arvon 0. Pseudosatunnaisuuden takia pisteitä arpovan funktion saama siemenluku vaikuttaa siihen, minkä minimipisteen algoritmi kulloinkin löytää. Kymmenellä suorituskerralla siemenlukua vaihtamalla saadaan pisteiden jakaumaksi $(1, 3, 1, 2, 3)$. Siemenluku ”arvottiin” tällä kertaa käsin, joten sitä ei voi pitää täysin satunnaisena. Nähdään kuitenkin, että funktio löytää jokaisen minimipisteen. Jos haluttaisiin tutkia, löytyvätkö kaikki pisteet samalla todennäköisyydellä, täytyisi siemenluvun asettaminen satunnaistaa ja ohjelma suorittaa huomattavasti useampia kertoja.

Taulukko 5.3. Shekel 10 testifunktion tuloksia.

```
The global minimum of Shekel 10:
4.000746532052143e+00 4.000592935058059e+00 3.999663396888725e+00 3.999509800569268e+00 ==> -1.053640981669205e+01
The global minimum of Shekel 10:
4.000746531810031e+00 4.000592934114497e+00 3.999663398105726e+00 3.999509800813388e+00 ==> -1.053640981669205e+01
The global minimum Of Shekel 10:
4.000746531797256e+00 4.000592934162273e+00 3.999663398295875e+00 3.999509799198316e+00 ==> -1.053640981669205e+01
The global minimum Of Shekel 10:
4.000746532247097e+00 4.000592934843805e+00 3.999663398025913e+00 3.999509798668068e+00 ==> -1.053640981669205e+01
The global minimum Of Shekel 10:
4.000746531151155e+00 4.000592936482043e+00 3.999663396010907e+00 3.999509799674326e+00 ==> -1.053640981669205e+01
The global minimum Of Shekel 10:
4.000746530446309e+00 4.000592931785202e+00 3.999663395558895e+00 3.999509799122933e+00 ==> -1.053640981669205e+01
The global minimum Of Shekel 10:
4.000746532718867e+00 4.000592934137266e+00 3.999663397679462e+00 3.999509800732739e+00 ==> -1.053640981669205e+01
The global minimum Of Shekel 10:
4.000746531137708e+00 4.000592934454086e+00 3.999663396598157e+00 3.999509800120757e+00 ==> -1.053640981669205e+01
The global minimum Of Shekel 10:
4.000746531299123e+00 4.000592934449644e+00 3.999663399046817e+00 3.999509798723940e+00 ==> -1.053640981669205e+01
The global minimum Of Shekel 10:
4.000746532864300e+00 4.000592935971202e+00 3.999663397668798e+00 3.999509801871981e+00 ==> -1.053640981669205e+01
```

Shekel 10 testifunktion globaali minimipiste viiden desimaalin tarkkuudella on $(4.00075, 4.00059, 3.99966, 3.99951)$ ja funktion arvo tässä pisteessä -10.5364 . Tällä tarkkuudella algoritmi löytää globaalin minimin joka kerta.

Näille testifunktioille algoritmi toimii siis erinomaisesti. Se selviää varsin hyvin Griewankin funktion monihuippuisesta maastosta kuten myös Shekelin neljän muuttujan optimoinnista.

5.4 Tulokset

Ohjelmaa päätettiin testata tällä kertaa myös normaalin simuloidun datan sijaan todellisella datalla. Tämä on haastavaa sovitushjelmalle, koska lokeromallit eivät tietenkään käytännössä ole koskaan täysin sopivia. Sovitus voi epäonnistua mallin sopimattomuuden lisäksi myös huonon datan takia. Testidatana käytettiin samojen tutkimusten mittauksia ja niihin sovitettiin sama nelilokeromalli kuin luvussa 4.

Liitteessä 5 on tutkimuksen 1 ja 5 sovitusten tulokset. Tulostuksen ensimmäiseltä riviltä nähdään, että on käytetty *fitglob*-nimistä ohjelmaa. Tämä ohjelma lukee mitatun data tiedostoista ja kirjoittaa tulokset erilliseen tiedostoon. Se käyttää neliösummafunktion minimointiin ITGO-algoritmia, jota se kutsuu *tgo.c*-nimisestä tiedostosta. Tulostuksen toisella rivillä on ohjelman

suoritus-aika, kolmannella, neljännellä ja viidennellä rivillä data-tiedostojen nimet ja seitsemännellä rivillä kuvausjakson aika sekä aikapisteen lukumäärä. Sovituksissa saadut parametrien estimaatit on listattu niin, että kunkin aivoalueen tulokset on omalla rivillään ja aivoalueen nimi rivin alussa. Symmetrisesti sijaitsevien aivoalueiden vasemman- ja oikeanpuoleisiin lohkoihin on tehty erikseen sovitukset. Nämä erotetaan merkinnöillä dx ja sin , joista dx tarkoittaa oikeanpuoleista ja sin vasemmanpuoleista osaa. Merkinnät avg tai $.$ tarkoittavat, että sovitukset on tehty näiden alueiden mittauksista laskettuihin keskiarvoihin. ITGO-algoritmin lähtöpisteiden lukumääräksi N asetettiin 100, ja se suoritti neljä iteraatiokertaa sovitettavien parametrien lukumäärän mukaan.

Tutkimusten 1 ja 5 tulostuksista nähdään, että ITGO-algoritmi löytää *frontal cortex* - ja *thalamus*-alueilta samat globaalit minimi kuin mapmin-ohjelmakin. Näin on myös muissa kymmenessä tutkimuksessa näillä kahdella aivoalueella lukuun ottamatta epäonnistuneita sovituksia, joissa on törmätty sallitun alueen rajoihin.

ITGO-algoritmin hyviä puolia on, että vastuu lähtöpisteiden arpomisesta on ohjelmalla eikä käyttäjällä kuten Pivot-algoritmissa. Käyttäjän ei myöskään tarvitse määrittää mitään erityistä askelpituutta. Pivotia käytettäessä näiden valintojen lisäksi lopputulokseen vaikutti usein myös valitut rajat. Tämän riippuvuuden tutkimista varten sovituksia tehtiin ITGO-algoritmilli myös erilaisia sallitun alueen rajoja käyttäen (liite 6). Aluksi sovitukset tehtiin rajoilla

$$K1 \in (0, 0.7), k2 \in (0, 0.2), k3 \in (0, 0.3) \quad ja \quad k4 \in (0, 0.3).$$

Liitteen ensimmäinen tulostus on tästä sovituksesta. Siitä on poistettu aivoalueet, joilla sovitukset epäonnistui eli törmäsi rajoihin. Toinen tulostus on sovituksesta, jossa rajat olivat

$$K1 \in (0, 1), k2 \in (0, 0.2), k3 \in (0, 0.3) \quad ja \quad k4 \in (0, 0.3).$$

Kolmas tulostus on sovituksesta, jossa rajat olivat

$$K1 \in (0, 0.7), k2 \in (0, 0.5), k3 \in (0, 0.3) \quad ja \quad k4 \in (0, 0.3).$$

Neljännän tulostuksen sovituksessa rajat olivat

$$K1 \in (0, 0.7), k2 \in (0, 0.2), k3 \in (0, 1) \quad ja \quad k4 \in (0, 0.3)$$

ja viidennen tulostuksen sovituksessa

$$K1 \in (0, 0.7), k2 \in (0, 0.2), k3 \in (0, 0.3) \quad ja \quad k4 \in (0, 1).$$

Näistä huomataan, että tulokset muuttuvat hieman. Tämä voi johtua siitä, että algoritmi arpoo eri rajoilla eri lähtöpisteet. Eroja syntyy kuitenkin korkeintaan kolmannessa desimaalissa. Estimaattien ei koskaan oleteta olevan tarkkoja, joten yleensä tuloksissa esitetäänkin enintään kolme desimaalia. Tällöin voidaan sanoa, että rajoilla ei ole vaikutusta ITGO-algoritmin toimintaan PET-sovituksissa.

6 Estimaattien harhaisuus

Sen lisäksi että globaalille minimointiongelmalle löydetään ratkaisu, olisi kiinnostavaa tietää myös arvioita siitä, millä todennäköisyydellä ratkaisu on varmasti globaali minimi. Tämän todennäköisyyden laskeminen vaatii rankkoja oletuksia kohdefunktiosta. On kuitenkin olemassa tapoja tutkia ratkaisun *harhaisuutta*. Toisin kuin useimmissa tieteellisissä tutkimuksissa, PET-kuvausta ei voida toistaa sellaisenaan. Kaikki päätelmät virheestä perustuvat siis yhteen mittaustulokseen. Käytämme nyt virheanalyysiin menetelmää, jossa luodaan keinotekoisesti uusia mittaustuloksia. Näiden avulla voidaan tehdä päätelmiä estimaattien jakaumista.

6.1 Keinotekoisien otosjakauman konstruointi

Tässä esitetty virheanalyysi perustuu artikkeliin [16] sekä kirjaan [17]. Artikkeliki kertoo spektraalianalyysin käytöstä parametrien estimaattien ratkaisemisessa ja ratkaisun parantamisesta niin sanotulla bootstrap-tekniikalla (lähde-tään liikkeelle jostakin varmasta tiedosta, ja otetaan yksittäisiä askeleita, jotka perustuvat tähän varmaan tietoon). Minä kuitenkin käytän parametrien estimaattien ratkaisemiseen spektraalianalyysin sijaan luvussa 3 esitettyä lockeromallia ja luvun 5 ITGO-algoritmia, sekä sovellan bootstrap-tekniikkaa estimaattien jakaumien hahmottamiseen ja luottamusvälilaskentaan.

PET-aika-aktiivisuuskäyrien mittausvirheiden varianssin oletetaan saavan eri arvon jokaisessa aikapisteessä. Painokertoimet w_1, w_2, \dots, w_m aikapisteissä t_1, t_2, \dots, t_m määritetään niin, että ne ovat verrannollisia mittaustulosten varianssiin kussakin pisteessä. En käsittele tätä asiaa, koska suhteellisten painokertoimien approksimointia varten on jo olemassa ohjelma. Tutkittavat parametrit ratkaistaan *painotetusta* neliösummasta luvuissa 3.2.1 ja 3.2.2 esitettyyn tapaan. Merkitään ratkaisua vektorilla \hat{k} . Merkitään yksinkertaisuuden vuoksi $C_{PET}^e(t_j) = X(t_j)$. Virhe kussakin aikapisteessä voidaan laskea kaavasta

$$\xi_j = X(t_j) - C_{PET}(t_j), \quad j = 1, \dots, m,$$

jossa $X(t_j)$ on siis PET-kamerasta saatu mittaustulos hetkellä t_j , ja $C_{PET}(t_j)$

on laskettu ratkaisun \hat{k} estimaatteja käyttäen kuten luvuissa 3.2.1 ja 3.2.2.

Lasketaan painotetut virheet kaavasta

$$r_j = \xi_j/w_j, \quad j = 1, \dots, m.$$

Tämän jälkeen painotettujen virheiden avulla muodostetaan R kappaletta keinotekoisia mittaustuloksia seuraavasti:

Joukolle (r_1, r_2, \dots, r_m) suoritetaan järjestetty otanta palauttaen, jolloin saadaan uusi painotettujen virheiden joukko $(r_1^{(i)}, r_2^{(i)}, \dots, r_m^{(i)})$. Uudeksi "mittausotokseksi" saadaan

$$X_{(i)}^*(t_j) = C_{PET}(t_j) + w_j r_j^{(i)} \quad j = 1, \dots, m.$$

Tehdään sovitus uudestaan tälle otokselle $X_{(i)}^*$, jolloin saadaan uusi ratkaisuvektori $k_{(i)}^*$. Tämä tehdään R kertaa, jolloin saadaan R otosta $X_{(1)}^*, \dots, X_{(R)}^*$ ja R ratkaisuvektoria $k_{(1)}^*, \dots, k_{(R)}^*$. Ratkaisuvektorit talletetaan $R \times N$ bootstrap-jakauma-matriisiksi A^* (N on sovittavien parametrien lukumäärä). Matriisin jokaisesta sarakkeesta voidaan laskea keskiarvo tietyn parametrin saamille arvoille. Merkitään $\bar{k}^* = \frac{1}{R} \sum_{i=1}^R k_{(i)}^*$.

Olkoon mittausvirheellä ε_j jakauma F , jota ei tunneta. Havaittujen virheiden ξ_j jakauma \hat{F} tunnetaan. Vektorin k estimaatin \hat{k} :n harhaisuus saadaan lausekkeesta

$$B = E_F[\hat{k}] - k,$$

jossa $E_F[\hat{k}]$ on \hat{k} :n (virhejakauman F mukainen) odotusarvo. Voidaan ajatella, että vektorit $k_{(i)}^*$ estimoivat vastaavasti \hat{k} :ta virhejakauman \hat{F} mukaisesti. Siis koska \hat{k} on k :n harhainen estimaatti, niin $k_{(i)}^*$:t \hat{k} :n harhaisia estimaatteja. Tästä seuraa, että

$$E_{\hat{F}}[k_{(i)}^*] \neq \hat{k},$$

ja näiden erotusta voidaan käyttää harhaisuuden estimaattina

$$\hat{B} = E_{\hat{F}}[k_{(i)}^*] - \hat{k} = \bar{k}^* - \hat{k}.$$

Nyt

$$\hat{B} \sim B, \text{ kun } R \rightarrow \infty \quad \text{ja}$$

$$E_{\hat{F}}[\hat{B}] \sim E_F[\hat{B}] \sim B, \text{ kun } R \rightarrow \infty.$$

Tämän jälkeen voidaan laskea uusi harhaton estimaatti

$$k^{uus} = \hat{k} - \hat{B} = \hat{k} - (\bar{k}^* - \hat{k}).$$

Käyttämällä tässä todettuja ominaisuuksia, saadaan

$$\begin{aligned} E_F[k^{uus}] &= E_F[\hat{k} - \hat{B}] = E_F[\hat{k}] - E_F[\hat{B}] \\ &= E_F[\hat{k}] - B = E_F[\hat{k}] - (E_F[\hat{k}] - k) = k. \end{aligned}$$

Estimaatti k^{uus} on siis asympotoottisesti virheetön.

6.2 Prosenttipisteiden määrittäminen

Estimaatin \hat{k}_j ($j = 1, \dots, N$) jakauman ρ -prosenttipiste

$$pp(\hat{k}_j; \rho) = \{\alpha \mid P_F(\hat{k}_j < \alpha) = \rho\},$$

voidaan laskea jokaiselle $\rho \in [0, 1]$. Tässä P_F on virhejakauman F mukainen todennäköisyys. Prosenttipiste kuvaa siis lukua, jonka alle parametrin estimaatin saama arvo jää $\rho \times 100$ prosentin varmuudella. Koska jakaumaa F ei tunneta, joudutaan käyttämään bootstrap-otosten havaittua virhejakaumaa \hat{F} .

Oletetaan, että estimaatin \hat{k}_j bootstrap menetelmällä saatujen estimaattien jakauma, eli matriisin A^* j :s sarake, on järjestetty vektoriksi $[k_j^{*(1)}, \dots, k_j^{*(R)}]^T$, jossa alkiot ovat suuruusjärjestyksessä pienimmästä suurimpaan. Nyt bootstrap-jakauman mukainen prosenttipiste on

$$pp^*(\hat{k}_j; \rho) = \{\alpha \mid P_{\hat{F}}(\hat{k}_j < \alpha) = \rho\} = k_j^{*(X)},$$

jossa $X = [\rho \times R]$. Koska estimaatti \bar{k}^* sisältää kuitenkin harhaisuutta, täytyy erikseen laskea harhaton prosenttipiste

$$pp^*(\hat{k}_j; \rho) = \max \{0, \hat{k}_j^{*(X)} - \hat{B}_j\} = \max \{0, k_j^{*(X)} - (\bar{k}_j^* - \hat{k}_j)\}.$$

Jakauman k_j^* keskiarvo siirtyi nyt kohtaan \hat{k}_j , jolloin pp^* :tä voidaan käyttää pp :n harhattomana estimaattina. Maksimifunktio varmistaa, ettei prosenttipiste voi saada negatiivisia arvoja.

Mitä lähempää jakauman reunaa prosenttipiste halutaan laskea, sitä enemmän bootstrap-otoksia tarvitaan hyvän estimaatin saamiseksi.

Estimaatti \hat{k} itsessään sisältää harhaisuutta. Sitä korjattaessa kuviteltu jakauma \hat{k}_j siirtyy \hat{B}_j :n verran. Jakauman k_j^{uusi} harhaton prosenttipiste saadaan siis yhtälöstä

$$pp^*(k_j^{uusi}; \rho) = \max \{0, k_j^{*(X)} - 2\hat{B}_j\} = \max \{0, k_j^{*(X)} - 2(\bar{k}_j^* - \hat{k}_j)\}.$$

6.3 Luottamusvälilaskenta

Prosenttipisteiden avulla voidaan laskea estimaateille luottamusvälejä. Laskettaessa ρ prosentin luottamusväliä saadaan ylä- ja alarajoiksi

$$\begin{aligned} k_{\text{ala}} &= pp(\hat{k}_j; \frac{1-\rho}{2}) \\ k_{\text{ylä}} &= pp(\hat{k}_j; \rho + \frac{1-\rho}{2}) = pp(\hat{k}_j; \frac{1+\rho}{2}). \end{aligned}$$

Lääketieteessä yleensä käytössä olevan 95 prosentin luottamusvälin rajat saadaan siis prosenttipisteistä

$$\begin{aligned} k_{\text{ala}} &= pp(\hat{k}_j; 0.025) \\ k_{\text{ylä}} &= pp(\hat{k}_j; 0.975). \end{aligned}$$

Edellisen luvun tapaan laskettavat harhattomat prosenttipisteet $pp^*(\hat{k}_j; \rho)$ estimoivat todellisia $pp(\hat{k}_j; \rho)$ prosenttipisteitä eli estimaatin \hat{k}_j luottamusväliä.

6.4 Analyysi

Teoreettisesti globaalissa optimoinnissa ei koskaan voida äärellisessä ajassa löytää globaalia minimiä tarkasti (ks. luku 2.3.2). Luottamusvälien määrittämisestä on kuitenkin se hyöty, että voidaan jollain tavoin perustella, että löydetty globaali minimi on tietyllä tarkkuudella todennäköisesti oikein. Niiden avulla voidaan siis tehdä päätelmiä siitä, millä tarkkuudella parametreja kannattaa ja täytyy estimoida.

Kirjallisuutta

- [1] Törn A., Žilinskas A.: *Global optimization*.
Springer-Verlag, 1989.
- [2] Nemhauser G.L., Rinnooy Kan A.H.G., Todd M.J.: *OPTIMIZATION (volume 1)*.
North-Holland, 1989.
- [3] <http://www.imm.dtu.dk/~km/GlobOpt/testex/testproblems.html#01>
26.11.2002.
- [4] Press, W., Teukolsky, S., Vetterling, W., Flannery, B.: *NUMERICAL RECIPES in C - The Art of Scientific Computing, Second Edition*.
Cambridge University Press, 1992.
- [5] Hickernell F.J., Ya-xiang Y.: *A Simple Multistart Algorithm for Global Optimization*.
OR Transactions. 1 (1997), pp. 1-11.
- [6] Haataja J.: *Optimoointitehtävien ratkaiseminen*.
CSC - Tieteellinen laskenta Oy, 1995.
- [7] Haataja J., Käpyaho J., Rahola J.: *Numeeriset menetelmät*.
CSC - Tieteellinen laskenta Oy, 1993.
- [8] Haataja J., Heikonen J., Leino Y., Rahola J., Ruokolainen J., Savolainen V.: *Numeeriset menetelmät käytännössä*.
CSC - Tieteellinen laskenta Oy, 1999.
- [9] Pursiheimo Ulla.: *Optimoointialgoritmit*.
Luentomoniste, Turun Yliopisto.
- [10] Oikonen, V., Sederholm, K.: *Turku PET Centre Modelling report*.
TPCMOD0001 2002-09-03
- [11] Vilkmán, H., Kajander, J., Nägren, K., Oikonen, V., Syvälahti, E., Hietala, J.: *Measurement of extrastriatal D₂-like reseptor binding with*

- $\int^1 C/FLB 457 - a test-retest analysis.$
Springer-Verlag, 2000.
- [12] Serra, P., Stanton, A., Kais, S.: *Pivot method for global optimization.*
Department of Chemistry, Purdue University, West Lafayette, Indiana,
1996.
- [13] Törn, Aimo: *Topografical Global Optimization.*
Åbo Akademi, Reports on Computer Science & Mathematics, 1990.
- [14] Törn, A., Viitanen, S: *Topografical Global Optimization for Constrained Problems.*
Åbo Akademi, Reports on Computer Science & Mathematics, 1993.
- [15] Törn, A., Viitanen, S: Iterative Topografical Global Optimization.
in C.A. Floudas and Panos M. Pardalos: *State of art in Global optimization* 353-363,
Kluwer Academic Publishers, 1996.
- [16] Turkheimer, F., Sokoloff, L., Bertoldo, A., Lucignani, G., Reivich, M., Jaggi, J. L., Schmidt, K.: Estimation of Component and Parameter Distributions in Spectral Analysis.
in *Journal of Cerebral Blood Flow and Metabolism* 18 (1998), 1211-1222.
Lippincott Williams & Wilkins.
- [17] Bradley, E., Tibshirani, J.: *An Introduction to the Bootstrap.*
Chapman & Hall, 1993.

Pseudokoodi ohjelmasta mapmin

Tiedot muuttujien $minp$, $minf$ ja vektorien min sekä $mincounter$ arvoista palautuvat kutsuvalle ohjelmalle. Saadaan siis tiedot kaikista löydetyistä minimeistä ja niiden löytymistäajuuksista.

$locAlg()$ =lokaali optimointialgoritmi. Sitä voi kutsua pisteestä p , jolloin se palauttaa pisteen johon optimoinnissa päättyy.

$objF()$ =kohdefunktio

$loopNr := 0$

$mincounter := \bar{0}$

$minNr := 0$

$bestf := MAXREAL$

REPEAT

1. Arpoo pisteen p sallitulle alueelle

2. $p := locAlg(p)$

3. $f := objF(p)$

4. IF $f < bestf$ THEN $bestp := p$, $bestf := f$ ENDIF

5. IF $p = min(i)$ jollakin $i \in [1, minNr]$
THEN $mincounter(i) := mincounter(i) + 1$
ELSE

$minNr := minNr + 1$

$min(minNr) := p$

$mincounter(minNr) := mincounter(minNr) + 1$

ENDIF

6. $loopNr := loopNr + 1$

UNTIL $loopNr = R$

Tutkimuksen 1 lokaalit minimiit

```

fcx All
Powell global search
Found 43 minimum
3.368735e-01 9.797972e-02 3.064607e-02 3.275507e-02 0 0 3.5e-02 => 8.905570e+00, 50 time(s), (50.00 %)
3.384637e-01 1.002016e-01 3.311542e-02 3.489243e-02 0 0 3.5e-02 => 8.910833e+00, 1 time(s), (1.00 %)
3.368715e-01 9.797606e-02 3.064127e-02 3.275041e-02 0 0 3.5e-02 => 8.905570e+00, 4 time(s), (4.00 %)
3.369365e-01 9.806027e-02 3.072166e-02 3.281249e-02 0 0 3.5e-02 => 8.905578e+00, 1 time(s), (1.00 %)
3.364114e-01 9.759311e-02 3.035280e-02 3.254025e-02 0 0 3.5e-02 => 8.905911e+00, 1 time(s), (1.00 %)
3.381118e-01 9.938449e-02 3.257057e-02 3.506309e-02 0 0 3.5e-02 => 8.913723e+00, 1 time(s), (1.00 %)
3.368659e-01 9.796860e-02 3.063415e-02 3.274494e-02 0 0 3.5e-02 => 8.905570e+00, 2 time(s), (2.00 %)
3.368692e-01 9.796517e-02 3.063044e-02 3.274765e-02 0 0 3.5e-02 => 8.905571e+00, 1 time(s), (1.00 %)
3.368663e-01 9.797086e-02 3.063817e-02 3.274940e-02 0 0 3.5e-02 => 8.905570e+00, 2 time(s), (2.00 %)
3.368954e-01 9.801242e-02 3.068473e-02 3.279109e-02 0 0 3.5e-02 => 8.905572e+00, 1 time(s), (1.00 %)
3.368617e-01 9.801620e-02 3.076319e-02 3.291187e-02 0 0 3.5e-02 => 8.905617e+00, 1 time(s), (1.00 %)
3.629927e-01 1.446973e-01 8.503085e-02 5.882401e-02 0 0 3.5e-02 => 9.719523e+00, 1 time(s), (1.00 %)
3.368922e-01 9.800188e-02 3.066021e-02 3.275969e-02 0 0 3.5e-02 => 8.905571e+00, 1 time(s), (1.00 %)
3.369088e-01 9.821921e-02 3.088835e-02 3.289112e-02 0 0 3.5e-02 => 8.905868e+00, 1 time(s), (1.00 %)
3.727693e-01 1.709211e-01 1.209735e-01 6.852822e-02 0 0 3.5e-02 => 1.009345e+01, 1 time(s), (1.00 %)
3.369176e-01 9.806519e-02 3.074600e-02 3.283589e-02 0 0 3.5e-02 => 8.905582e+00, 1 time(s), (1.00 %)
3.368766e-01 9.798503e-02 3.065262e-02 3.276126e-02 0 0 3.5e-02 => 8.905570e+00, 2 time(s), (2.00 %)
3.368809e-01 9.798730e-02 3.065090e-02 3.275704e-02 0 0 3.5e-02 => 8.905570e+00, 1 time(s), (1.00 %)
3.368306e-01 9.789975e-02 3.055393e-02 3.268306e-02 0 0 3.5e-02 => 8.905581e+00, 1 time(s), (1.00 %)
3.368360e-01 9.788089e-02 3.051694e-02 3.262486e-02 0 0 3.5e-02 => 8.905593e+00, 1 time(s), (1.00 %)
3.368811e-01 9.799330e-02 3.066411e-02 3.277297e-02 0 0 3.5e-02 => 8.905571e+00, 3 time(s), (3.00 %)
3.370136e-01 9.820361e-02 3.093044e-02 3.303132e-02 0 0 3.5e-02 => 8.905645e+00, 1 time(s), (1.00 %)
3.364125e-01 9.725374e-02 2.990121e-02 3.211693e-02 0 0 3.5e-02 => 8.906301e+00, 1 time(s), (1.00 %)
3.341112e-01 9.447314e-02 2.678494e-02 2.898487e-02 0 0 3.5e-02 => 8.922111e+00, 1 time(s), (1.00 %)
3.368905e-01 9.798884e-02 3.064477e-02 3.274796e-02 0 0 3.5e-02 => 8.905571e+00, 1 time(s), (1.00 %)
3.368688e-01 9.797530e-02 3.064276e-02 3.275296e-02 0 0 3.5e-02 => 8.905570e+00, 1 time(s), (1.00 %)
3.375948e-01 9.838992e-02 3.068180e-02 3.256721e-02 0 0 3.5e-02 => 8.906993e+00, 1 time(s), (1.00 %)
3.369037e-01 9.800785e-02 3.067928e-02 3.278801e-02 0 0 3.5e-02 => 8.905573e+00, 1 time(s), (1.00 %)
3.378526e-01 9.925091e-02 3.206120e-02 3.401833e-02 0 0 3.5e-02 => 8.907436e+00, 1 time(s), (1.00 %)
3.427062e-01 1.055311e-01 3.752514e-02 3.720466e-02 0 0 3.5e-02 => 8.965426e+00, 1 time(s), (1.00 %)
3.368654e-01 9.796284e-02 3.061613e-02 3.272008e-02 0 0 3.5e-02 => 8.905572e+00, 1 time(s), (1.00 %)
3.369591e-01 9.805017e-02 3.064855e-02 3.272863e-02 0 0 3.5e-02 => 8.905632e+00, 1 time(s), (1.00 %)
2.959570e-01 5.802874e-02 3.816912e-03 4.532890e-13 0 0 3.5e-02 => 1.636446e+01, 1 time(s), (1.00 %)
2.839555e-01 6.156674e-02 6.633726e-03 6.776065e-14 0 0 3.5e-02 => 1.523294e+01, 1 time(s), (1.00 %)
3.362603e-01 9.763762e-02 3.103840e-02 3.364113e-02 0 0 3.5e-02 => 8.909675e+00, 1 time(s), (1.00 %)
3.368528e-01 9.797204e-02 3.066280e-02 3.279128e-02 0 0 3.5e-02 => 8.905575e+00, 1 time(s), (1.00 %)
3.369926e-01 9.667485e-02 2.792637e-02 2.978263e-02 0 0 3.5e-02 => 8.925582e+00, 1 time(s), (1.00 %)
2.760496e-01 5.127047e-02 2.461401e-03 2.394975e-15 0 0 3.5e-02 => 2.094684e+01, 1 time(s), (1.00 %)
2.839379e-01 6.155937e-02 6.632250e-03 4.888015e-15 0 0 3.5e-02 => 1.523807e+01, 1 time(s), (1.00 %)
3.386168e-01 9.661775e-02 2.734181e-02 2.882576e-02 0 0 3.5e-02 => 8.958948e+00, 1 time(s), (1.00 %)
3.597738e-01 1.354994e-01 7.074184e-02 5.275391e-02 0 0 3.5e-02 => 9.593611e+00, 1 time(s), (1.00 %)
3.377314e-01 9.901852e-02 3.188736e-02 3.404601e-02 0 0 3.5e-02 => 8.907556e+00, 1 time(s), (1.00 %)
3.675595e-01 1.533962e-01 9.371946e-02 6.037355e-02 0 0 3.5e-02 => 9.909312e+00, 1 time(s), (1.00 %)
Best fit:
3.368735e-01 9.797972e-02 3.064607e-02 3.275507e-02 0 0 3.5e-02 ==> 8.905570e+00

tha All
Powell global search
Found 36 minimum
4.287591e-01 1.513903e-01 1.756879e-01 3.937707e-02 0 0 3.5e-02 => 1.182637e+01, 51 time(s), (51.00 %)
4.287666e-01 1.514063e-01 1.757082e-01 3.937783e-02 0 0 3.5e-02 => 1.182637e+01, 8 time(s), (8.00 %)
4.244710e-01 1.436559e-01 1.677380e-01 3.950933e-02 0 0 3.5e-02 => 1.183482e+01, 1 time(s), (1.00 %)
3.671379e-01 5.106641e-02 2.690100e-02 1.109842e-02 0 0 3.5e-02 => 1.559759e+01, 1 time(s), (1.00 %)
3.896720e-01 2.000000e-01 3.873086e-01 5.865291e-02 0 0 3.5e-02 => 1.966129e+01, 1 time(s), (1.00 %)
4.343418e-01 1.648199e-01 1.933968e-01 4.007138e-02 0 0 3.5e-02 => 1.183250e+01, 1 time(s), (1.00 %)
4.286517e-01 1.512798e-01 1.756903e-01 3.940485e-02 0 0 3.5e-02 => 1.182639e+01, 1 time(s), (1.00 %)
4.287508e-01 1.513742e-01 1.756680e-01 3.937631e-02 0 0 3.5e-02 => 1.182637e+01, 4 time(s), (4.00 %)
4.345167e-01 1.683314e-01 2.003426e-01 4.064913e-02 0 0 3.5e-02 => 1.184117e+01, 1 time(s), (1.00 %)
4.273838e-01 1.482697e-01 1.717802e-01 3.927667e-02 0 0 3.5e-02 => 1.182693e+01, 1 time(s), (1.00 %)
4.287035e-01 1.512495e-01 1.754945e-01 3.936826e-02 0 0 3.5e-02 => 1.182638e+01, 1 time(s), (1.00 %)
4.358034e-01 1.620847e-01 1.842355e-01 3.877926e-02 0 0 3.5e-02 => 1.185362e+01, 1 time(s), (1.00 %)
4.287549e-01 1.513968e-01 1.757124e-01 3.938108e-02 0 0 3.5e-02 => 1.182637e+01, 1 time(s), (1.00 %)
4.285005e-01 1.512214e-01 1.758509e-01 3.945325e-02 0 0 3.5e-02 => 1.182650e+01, 1 time(s), (1.00 %)
4.344440e-01 1.624281e-01 1.873683e-01 3.929751e-02 0 0 3.5e-02 => 1.183664e+01, 1 time(s), (1.00 %)
4.288200e-01 1.515324e-01 1.758794e-01 3.938589e-02 0 0 3.5e-02 => 1.182638e+01, 1 time(s), (1.00 %)
4.287428e-01 1.513523e-01 1.756368e-01 3.937475e-02 0 0 3.5e-02 => 1.182637e+01, 5 time(s), (5.00 %)
4.287833e-01 1.514473e-01 1.757651e-01 3.938074e-02 0 0 3.5e-02 => 1.182637e+01, 1 time(s), (1.00 %)
4.285937e-01 1.509999e-01 1.751293e-01 3.934224e-02 0 0 3.5e-02 => 1.182638e+01, 1 time(s), (1.00 %)
4.164914e-01 1.329586e-01 1.598026e-01 4.057338e-02 0 0 3.5e-02 => 1.192414e+01, 1 time(s), (1.00 %)
4.540730e-01 2.000000e-01 2.224669e-01 3.882164e-02 0 0 3.5e-02 => 1.200004e+01, 1 time(s), (1.00 %)
4.309592e-01 1.566275e-01 1.827064e-01 3.968253e-02 0 0 3.5e-02 => 1.182739e+01, 1 time(s), (1.00 %)
4.287315e-01 1.513256e-01 1.756004e-01 3.937297e-02 0 0 3.5e-02 => 1.182637e+01, 1 time(s), (1.00 %)
4.121232e-01 1.168586e-01 1.282680e-01 3.656056e-02 0 0 3.5e-02 => 1.190176e+01, 1 time(s), (1.00 %)
4.422027e-01 1.842491e-01 2.174555e-01 4.068609e-02 0 0 3.5e-02 => 1.185693e+01, 1 time(s), (1.00 %)
4.287625e-01 1.513805e-01 1.756529e-01 3.937125e-02 0 0 3.5e-02 => 1.182637e+01, 1 time(s), (1.00 %)

```

```
4.289358e-01 1.518006e-01 1.762368e-01 3.940147e-02 0 0 3.5e-02 => 1.182638e+01, 1 time(s), (1.00 %)
4.298121e-01 1.547641e-01 1.816825e-01 3.997138e-02 0 0 3.5e-02 => 1.182923e+01, 1 time(s), (1.00 %)
4.431437e-01 2.000000e-01 2.578709e-01 4.532802e-02 0 0 3.5e-02 => 1.216690e+01, 1 time(s), (1.00 %)
4.287651e-01 1.514615e-01 1.758213e-01 3.938903e-02 0 0 3.5e-02 => 1.182638e+01, 1 time(s), (1.00 %)
4.286026e-01 1.510439e-01 1.752619e-01 3.936946e-02 0 0 3.5e-02 => 1.182638e+01, 1 time(s), (1.00 %)
4.533587e-01 2.000000e-01 2.228584e-01 3.868993e-02 0 0 3.5e-02 => 1.199103e+01, 1 time(s), (1.00 %)
4.287110e-01 1.512724e-01 1.755225e-01 3.936797e-02 0 0 3.5e-02 => 1.182638e+01, 1 time(s), (1.00 %)
4.286856e-01 1.511959e-01 1.754245e-01 3.936691e-02 0 0 3.5e-02 => 1.182638e+01, 1 time(s), (1.00 %)
4.288324e-01 1.514685e-01 1.757553e-01 3.937886e-02 0 0 3.5e-02 => 1.182638e+01, 1 time(s), (1.00 %)
4.317286e-01 1.575925e-01 1.840063e-01 3.983067e-02 0 0 3.5e-02 => 1.182916e+01, 1 time(s), (1.00 %)
Best fit:
4.287591e-01 1.513903e-01 1.756879e-01 3.937707e-02 0 0 3.5e-02 ==> 1.182637e+01
```

Tutkimuksen 2 lokaalit minimiä

```

fcx All
Powell global search
Found 39 minimum
2.685329e-01 8.763505e-02 2.253359e-02 3.193913e-02 0 0 3.5e-02 => 9.135656e-01, 58 time(s), (58.00 %)
2.685256e-01 8.762366e-02 2.252055e-02 3.192244e-02 0 0 3.5e-02 => 9.135658e-01, 2 time(s), (2.00 %)
2.685521e-01 8.769072e-02 2.270860e-02 3.232787e-02 0 0 3.5e-02 => 9.137843e-01, 1 time(s), (1.00 %)
2.685168e-01 8.761116e-02 2.250366e-02 3.189754e-02 0 0 3.5e-02 => 9.135667e-01, 1 time(s), (1.00 %)
2.685363e-01 8.763971e-02 2.253860e-02 3.194529e-02 0 0 3.5e-02 => 9.135656e-01, 3 time(s), (3.00 %)
2.685070e-01 8.761908e-02 2.254319e-02 3.197471e-02 0 0 3.5e-02 => 9.135710e-01, 1 time(s), (1.00 %)
2.685091e-01 8.761478e-02 2.251275e-02 3.190939e-02 0 0 3.5e-02 => 9.135674e-01, 1 time(s), (1.00 %)
2.298809e-01 5.894114e-02 4.777851e-03 9.458146e-15 0 0 3.5e-02 => 4.855131e+00, 2 time(s), (2.00 %)
2.713051e-01 8.472472e-02 1.496028e-02 1.687906e-02 0 0 3.5e-02 => 1.248986e+00, 1 time(s), (1.00 %)
2.685265e-01 8.762842e-02 2.252805e-02 3.193322e-02 0 0 3.5e-02 => 9.135657e-01, 1 time(s), (1.00 %)
2.324264e-01 4.911180e-02 3.670360e-15 4.998472e-01 0 0 3.5e-02 => 1.290229e+01, 1 time(s), (1.00 %)
2.685429e-01 8.764510e-02 2.252431e-02 3.190551e-02 0 0 3.5e-02 => 9.135702e-01, 1 time(s), (1.00 %)
2.685458e-01 8.766709e-02 2.257323e-02 3.198893e-02 0 0 3.5e-02 => 9.135680e-01, 1 time(s), (1.00 %)
2.689879e-01 8.850901e-02 2.554109e-02 3.767373e-02 0 0 3.5e-02 => 9.650247e-01, 1 time(s), (1.00 %)
2.728254e-01 9.507644e-02 3.065911e-02 4.014857e-02 0 0 3.5e-02 => 9.828890e-01, 1 time(s), (1.00 %)
2.685815e-01 8.762806e-02 2.255320e-02 3.203296e-02 0 0 3.5e-02 => 9.136520e-01, 1 time(s), (1.00 %)
2.686671e-01 8.756579e-02 2.247856e-02 3.198471e-02 0 0 3.5e-02 => 9.141688e-01, 1 time(s), (1.00 %)
2.685304e-01 8.764664e-02 2.256203e-02 3.198736e-02 0 0 3.5e-02 => 9.135680e-01, 1 time(s), (1.00 %)
2.685874e-01 8.774499e-02 2.264603e-02 3.206671e-02 0 0 3.5e-02 => 9.135937e-01, 1 time(s), (1.00 %)
2.674793e-01 8.640013e-02 2.154195e-02 3.098129e-02 0 0 3.5e-02 => 9.159603e-01, 1 time(s), (1.00 %)
3.074421e-01 2.000000e-01 2.028944e-01 1.005332e-01 0 0 3.5e-02 => 3.219906e+00, 1 time(s), (1.00 %)
2.685245e-01 8.762222e-02 2.251640e-02 3.191406e-02 0 0 3.5e-02 => 9.135660e-01, 1 time(s), (1.00 %)
2.685246e-01 8.762235e-02 2.251947e-02 3.192149e-02 0 0 3.5e-02 => 9.135658e-01, 1 time(s), (1.00 %)
2.615672e-01 7.843722e-02 1.372144e-02 1.898202e-02 0 0 3.5e-02 => 1.055993e+00, 1 time(s), (1.00 %)
3.052252e-01 1.545747e-01 1.024623e-01 7.280566e-02 0 0 3.5e-02 => 2.467135e+00, 1 time(s), (1.00 %)
2.685448e-01 8.764862e-02 2.254420e-02 3.194874e-02 0 0 3.5e-02 => 9.135659e-01, 1 time(s), (1.00 %)
2.685018e-01 8.758163e-02 2.247033e-02 3.186050e-02 0 0 3.5e-02 => 9.135705e-01, 1 time(s), (1.00 %)
2.685822e-01 8.771667e-02 2.254296e-02 3.184307e-02 0 0 3.5e-02 => 9.136713e-01, 1 time(s), (1.00 %)
2.685229e-01 8.760835e-02 2.249200e-02 3.187474e-02 0 0 3.5e-02 => 9.135685e-01, 1 time(s), (1.00 %)
3.083890e-01 2.000000e-01 2.060100e-01 1.028314e-01 0 0 3.5e-02 => 3.309311e+00, 1 time(s), (1.00 %)
2.685429e-01 8.764959e-02 2.254990e-02 3.195986e-02 0 0 3.5e-02 => 9.135659e-01, 1 time(s), (1.00 %)
2.707627e-01 8.564919e-02 1.665105e-02 2.029537e-02 0 0 3.5e-02 => 1.110501e+00, 1 time(s), (1.00 %)
2.829959e-01 1.013800e-01 3.364048e-02 4.165566e-02 0 0 3.5e-02 => 1.263538e+00, 1 time(s), (1.00 %)
2.686582e-01 8.792910e-02 2.282706e-02 3.224706e-02 0 0 3.5e-02 => 9.138269e-01, 1 time(s), (1.00 %)
2.685597e-01 8.766401e-02 2.255882e-02 3.196669e-02 0 0 3.5e-02 => 9.135668e-01, 1 time(s), (1.00 %)
2.684632e-01 8.756277e-02 2.248683e-02 3.189715e-02 0 0 3.5e-02 => 9.135795e-01, 1 time(s), (1.00 %)
2.685066e-01 8.757753e-02 2.245395e-02 3.182722e-02 0 0 3.5e-02 => 9.135741e-01, 1 time(s), (1.00 %)
2.685238e-01 8.759322e-02 2.246435e-02 3.182977e-02 0 0 3.5e-02 => 9.135751e-01, 1 time(s), (1.00 %)
2.685068e-01 8.760810e-02 2.251107e-02 3.191512e-02 0 0 3.5e-02 => 9.135668e-01, 1 time(s), (1.00 %)
Best fit:
2.685329e-01 8.763505e-02 2.253359e-02 3.193913e-02 0 0 3.5e-02 ==> 9.135656e-01

tha All
Powell global search
Found 29 minimum
3.183632e-01 1.037852e-01 1.064825e-01 3.160066e-02 0 0 3.5e-02 => 2.301123e+00, 63 time(s), (63.00 %)
3.183674e-01 1.037932e-01 1.064932e-01 3.160172e-02 0 0 3.5e-02 => 2.301123e+00, 4 time(s), (4.00 %)
3.183557e-01 1.037744e-01 1.064719e-01 3.160052e-02 0 0 3.5e-02 => 2.301123e+00, 3 time(s), (3.00 %)
3.192894e-01 1.054884e-01 1.089193e-01 3.192375e-02 0 0 3.5e-02 => 2.302197e+00, 1 time(s), (1.00 %)
3.470714e-01 2.000000e-01 2.480950e-01 4.106087e-02 0 0 3.5e-02 => 3.376303e+00, 1 time(s), (1.00 %)
3.204045e-01 1.075244e-01 1.109874e-01 3.187402e-02 0 0 3.5e-02 => 2.304570e+00, 1 time(s), (1.00 %)
3.184640e-01 1.039697e-01 1.067222e-01 3.162759e-02 0 0 3.5e-02 => 2.301134e+00, 1 time(s), (1.00 %)
3.183869e-01 1.038225e-01 1.065248e-01 3.160326e-02 0 0 3.5e-02 => 2.301123e+00, 2 time(s), (2.00 %)
3.183440e-01 1.037490e-01 1.064349e-01 3.159610e-02 0 0 3.5e-02 => 2.301123e+00, 1 time(s), (1.00 %)
3.183182e-01 1.037057e-01 1.064066e-01 3.160421e-02 0 0 3.5e-02 => 2.301127e+00, 1 time(s), (1.00 %)
3.183539e-01 1.037678e-01 1.064602e-01 3.159865e-02 0 0 3.5e-02 => 2.301123e+00, 4 time(s), (4.00 %)
3.183757e-01 1.038079e-01 1.065136e-01 3.160435e-02 0 0 3.5e-02 => 2.301123e+00, 1 time(s), (1.00 %)
3.487208e-01 1.652624e-01 1.802287e-01 3.663766e-02 0 0 3.5e-02 => 2.850404e+00, 1 time(s), (1.00 %)
3.189425e-01 1.049916e-01 1.081400e-01 3.176983e-02 0 0 3.5e-02 => 2.301509e+00, 1 time(s), (1.00 %)
3.184118e-01 1.038538e-01 1.065248e-01 3.158899e-02 0 0 3.5e-02 => 2.301128e+00, 1 time(s), (1.00 %)
3.183881e-01 1.038341e-01 1.065483e-01 3.160728e-02 0 0 3.5e-02 => 2.301123e+00, 1 time(s), (1.00 %)
3.179623e-01 1.033351e-01 1.061313e-01 3.161236e-02 0 0 3.5e-02 => 2.301298e+00, 1 time(s), (1.00 %)
3.191497e-01 1.049342e-01 1.076262e-01 3.162055e-02 0 0 3.5e-02 => 2.301582e+00, 1 time(s), (1.00 %)
3.408419e-01 1.495587e-01 1.588433e-01 3.410275e-02 0 0 3.5e-02 => 2.576855e+00, 1 time(s), (1.00 %)
2.677947e-01 3.571885e-02 1.240994e-02 5.906731e-14 0 0 3.5e-02 => 8.389613e+00, 1 time(s), (1.00 %)
3.151256e-01 9.653591e-02 9.748753e-02 3.121801e-02 0 0 3.5e-02 => 2.334290e+00, 1 time(s), (1.00 %)
3.183331e-01 1.037286e-01 1.064153e-01 3.159705e-02 0 0 3.5e-02 => 2.301124e+00, 1 time(s), (1.00 %)
3.183609e-01 1.037870e-01 1.064943e-01 3.160462e-02 0 0 3.5e-02 => 2.301123e+00, 1 time(s), (1.00 %)
3.183874e-01 1.037682e-01 1.063971e-01 3.157737e-02 0 0 3.5e-02 => 2.301130e+00, 1 time(s), (1.00 %)
3.187410e-01 2.000000e-01 3.312024e-01 5.020824e-02 0 0 3.5e-02 => 6.447861e+00, 1 time(s), (1.00 %)
3.183193e-01 1.037939e-01 1.065676e-01 3.162313e-02 0 0 3.5e-02 => 2.301136e+00, 1 time(s), (1.00 %)
3.183179e-01 1.036996e-01 1.063687e-01 3.158919e-02 0 0 3.5e-02 => 2.301125e+00, 1 time(s), (1.00 %)
2.367508e-01 2.598712e-02 7.825304e-03 1.482211e-14 0 0 3.5e-02 => 1.499481e+01, 1 time(s), (1.00 %)
3.183468e-01 1.036455e-01 1.062467e-01 3.157550e-02 0 0 3.5e-02 => 2.301154e+00, 1 time(s), (1.00 %)
Best fit:
3.183632e-01 1.037852e-01 1.064825e-01 3.160066e-02 0 0 3.5e-02 ==> 2.301123e+00

```

Pseudokoodi ohjelmasta tgo

locAlg()=lokaali optimointialgoritmi. Sitä voi kutsu pisteestä p , jolloin se palauttaa pisteen johon optimoinnissa päätyy.

dist()=kahden pisteen etäisyys "*dim*"-ulotteisessa avaruudessa, eli $dist(x_1, x_2) = \sqrt{(x_1^1 - x_2^1)^2 + \dots + (x_1^{dim} - x_2^{dim})^2}$

dim = parametrien lkm

K = tutkittavien naapureiden lkm ($K = 3$)

N = arvottavien pisteiden lkm

1. IF $K > N - 1$ THEN $K = N - 1$ ENDIF

2. $iterNr := dim$
 $T\bar{M} := \bar{0}$

3. FOR $m = 1, \dots, iterNr$ DO

IF $T\bar{M}(j) = 0$

THEN Arvotaan vektoriin $\bar{x}(N)$ piste kohtaan $\bar{x}(j)$.

Lasketaan kohdefunktion arvo tässä pisteessä vektorin $f(N)$ kohtaan $f(j)$.

ENDIF

Lasketaan pisteiden etäisyydet matriisiin dd .

$dd(i, j) := dist(\bar{x}(i), \bar{x}(j))$

Tutkitaan jokainen piste: onko topografinen minimi?

FOR $i = 1, \dots, N$ DO

FOR $j = 1, \dots, N$ DO

$\bar{d}(j) := dd(i, j)$ (Talletetaan etäisyysmatriisiin i :s rivi)

ENDFOR

$\bar{d}(i) := MAXREAL$ (Etäisyys itseän iso luku)

FOR $k = 1, \dots, K$ DO

$min := MAXREAL$

FOR $j = 1, \dots, N$ DO

IF $min > \bar{d}(j)$ THEN $min := \bar{d}(j)$, $minindex := j$

ENDIF

ENDFOR

IF $f(i) > f(\text{minindex})$ THEN jump to next i ENDIF
 $\bar{d}(\text{minindex}) := \text{MAXREAL}$ (ettei löydetä samaa naapurua uudestaan)

ENDFOR

Jos piste i on parempi kuin K naapurua, merkitään se topografiseen minimiin.

$TM(i) := 1$

ENDFOR

ENDFOR (tgo iteraatiot)

4. Käytetään topografisia minimejä ($TM(j) = 1$) lokaalin optimointialgoritmin lähtöpisteinä

FOR $j = 1, \dots, N$

IF $TM(j) = 1$ THEN $\bar{x}(j) := \text{locAlg}(x(j))$

ENDFOR

5. Etsitään lokaalien optimointien paras tulos ja suoritetaan lokaali optimointi vielä kerran tästä pisteestä tiukemmilla lopetusehdoilla.

ITGO:lla saadut tulokset

fitglob 1.0 (c) 2002 by Turku PET Centre
 Date: 2002-11-26 14:18:41
 Data file: ub0001.dft
 Plasma file: ub0001ap.pure.kbq
 Blood file: ub0001ap.kbq
 Data range: 69 min (N=16)

Data was not weighted.

Region			K1	k2	k3	k4	k5	k6	Vb	WSS
cau	dx	All	0.3831	0.1500	0.5000	0.0045	0	0	3.5e-02	6.7526
cau	sin	All	0.3521	0.0830	0.5000	0.0113	0	0	3.5e-02	0.1069
cer	dx	All	0.4196	0.4137	0.5000	0.1486	0	0	3.5e-02	7.7718
cer	sin	All	0.3159	0.1002	0.0165	0.0299	0	0	3.5e-02	6.3000
fcx	dx	All	0.3384	0.0993	0.0325	0.0358	0	0	3.5e-02	8.9859
fcx	sin	All	0.3353	0.0967	0.0290	0.0298	0	0	3.5e-02	8.8621
pit	.	All	0.2155	0.1214	0.0890	0.0012	0	0	3.5e-02	3.1444
put	dx	All	0.4507	0.1547	0.5000	0.0053	0	0	3.5e-02	9.2537
put	sin	All	0.4312	0.0990	0.5000	0.0059	0	0	3.5e-02	0.1295
temp	dx	All	0.3434	0.3825	0.5000	0.0538	0	0	3.5e-02	5.5757
temp	sin	All	0.3376	0.3824	0.5000	0.0509	0	0	3.5e-02	6.0764
tha	dx	All	0.4339	0.1511	0.1657	0.0374	0	0	3.5e-02	0.1183
tha	sin	All	0.4254	0.1566	0.1938	0.0418	0	0	3.5e-02	0.1245
cau	.	All	0.3669	0.1149	0.5000	0.0070	0	0	3.5e-02	7.2772
cer	avg	All	0.3239	0.0986	0.0141	0.0267	0	0	3.5e-02	6.5768
fcx	.	All	0.3369	0.0980	0.0306	0.0328	0	0	3.5e-02	8.9056
put	.	All	0.4407	0.1259	0.5000	0.0055	0	0	3.5e-02	0.1074
temp	.	All	0.3405	0.3825	0.5000	0.0523	0	0	3.5e-02	5.7534
tha	.	All	0.4288	0.1514	0.1757	0.0394	0	0	3.5e-02	0.1183

fitglob 1.0 (c) 2002 by Turku PET Centre
 Date: 2002-11-26 14:18:51
 Data file: ub0005.dft
 Plasma file: ub0005ap.pure.kbq
 Blood file: ub0005ap.kbq
 Data range: 69 min (N=16)

Data was not weighted.

Region			K1	k2	k3	k4	k5	k6	Vb	WSS
cau	dx	All	0.2607	0.0121	0.0402	0.0000	0	0	3.5e-02	2.4516
cau	sin	All	0.2842	0.0752	0.2809	0.0138	0	0	3.5e-02	0.9647
cer	dx	All	0.2776	0.0985	0.0170	0.0621	0	0	3.5e-02	1.0522
cer	sin	All	0.3451	0.3472	0.5000	0.2059	0	0	3.5e-02	1.6341
fcx	dx	All	0.2716	0.0924	0.0264	0.0367	0	0	3.5e-02	1.2255
fcx	sin	All	0.2660	0.0838	0.0197	0.0278	0	0	3.5e-02	0.7252
pit	.	All	0.2707	0.1411	0.0698	0.0088	0	0	3.5e-02	3.1343
put	dx	All	0.2903	0.0117	0.0306	0.0000	0	0	3.5e-02	1.0246
put	sin	All	0.2850	0.0135	0.0413	0.0054	0	0	3.5e-02	1.5775
temp	dx	All	0.1955	0.0506	0.0469	0.0352	0	0	3.5e-02	0.7746
temp	sin	All	0.1978	0.0495	0.0440	0.0329	0	0	3.5e-02	0.7652
tha	dx	All	0.3409	0.1307	0.1361	0.0334	0	0	3.5e-02	3.0678
tha	sin	All	0.3002	0.0860	0.0851	0.0293	0	0	3.5e-02	2.4480
cau	.	All	0.2625	0.0193	0.0836	0.0107	0	0	3.5e-02	1.4638
cer	avg	All	0.2768	0.0920	0.0062	0.0241	0	0	3.5e-02	1.1993
fcx	.	All	0.2685	0.0876	0.0225	0.0319	0	0	3.5e-02	0.9136
put	.	All	0.2869	0.0118	0.0310	0.0000	0	0	3.5e-02	1.1966
temp	.	All	0.1965	0.0498	0.0450	0.0340	0	0	3.5e-02	0.7397
tha	.	All	0.3184	0.1038	0.1065	0.0316	0	0	3.5e-02	2.3011

ITGO listaus muunnelluilla rajoilla

fitglob 1.0 (c) 2002 by Turku PET Centre
 Date: 2002-12-10 11:08:35
 Data file: ub0001.dft
 Plasma file: ub0001ap.pure.kbq
 Blood file: ub0001ap.kbq
 Data range: 69 min (N=16)

Data was not weighted.

Region			K1	k2	k3	k4	k5	k6	Vb	WSS
cer	dx	All	0.3318	0.0967	0.0114	2.1541e-02	0	0	3.5e-02	7.2254e+00
cer	sin	All	0.3158	0.1002	0.0165	3.0019e-02	0	0	3.5e-02	6.3000e+00
fcx	dx	All	0.3384	0.0993	0.0325	3.5849e-02	0	0	3.5e-02	8.9859e+00
fcx	sin	All	0.3353	0.0968	0.0290	2.9830e-02	0	0	3.5e-02	8.8621e+00
pit	.	All	0.2155	0.1214	0.0889	1.2242e-03	0	0	3.5e-02	3.1444e+00
tha	dx	All	0.4339	0.1511	0.1657	3.7412e-02	0	0	3.5e-02	1.1825e+01
tha	sin	All	0.4254	0.1567	0.1939	4.1819e-02	0	0	3.5e-02	1.2451e+01
cer	avg	All	0.3239	0.0986	0.0141	2.6683e-02	0	0	3.5e-02	6.5768e+00
fcx	.	All	0.3369	0.0980	0.0306	3.2755e-02	0	0	3.5e-02	8.9056e+00
tha	.	All	0.4288	0.1514	0.1757	3.9377e-02	0	0	3.5e-02	1.1826e+01

fitglob 1.0 (c) 2002 by Turku PET Centre
 Date: 2002-12-10 11:13:19
 Data file: ub0001.dft
 Plasma file: ub0001ap.pure.kbq
 Blood file: ub0001ap.kbq
 Data range: 69 min (N=16)

Data was not weighted.

Region			K1	k2	k3	k4	k5	k6	Vb	WSS
cer	dx	All	0.3318	0.0967	0.0114	2.1530e-02	0	0	3.5e-02	7.2254e+00
cer	sin	All	0.3158	0.1002	0.0165	3.0020e-02	0	0	3.5e-02	6.3000e+00
fcx	dx	All	0.3384	0.0993	0.0325	3.5848e-02	0	0	3.5e-02	8.9859e+00
fcx	sin	All	0.3353	0.0968	0.0290	2.9830e-02	0	0	3.5e-02	8.8621e+00
pit	.	All	0.2155	0.1214	0.0890	1.2250e-03	0	0	3.5e-02	3.1444e+00
tha	dx	All	0.4339	0.1511	0.1657	3.7412e-02	0	0	3.5e-02	1.1825e+01
tha	sin	All	0.4254	0.1567	0.1939	4.1823e-02	0	0	3.5e-02	1.2451e+01
cer	avg	All	0.3239	0.0986	0.0141	2.6679e-02	0	0	3.5e-02	6.5768e+00
fcx	.	All	0.3369	0.0980	0.0306	3.2755e-02	0	0	3.5e-02	8.9056e+00
tha	.	All	0.4288	0.1514	0.1757	3.9377e-02	0	0	3.5e-02	1.1826e+01

fitglob 1.0 (c) 2002 by Turku PET Centre
 Date: 2002-12-10 11:15:19
 Data file: ub0001.dft
 Plasma file: ub0001ap.pure.kbq
 Blood file: ub0001ap.kbq
 Data range: 69 min (N=16)

Data was not weighted.

Region			K1	k2	k3	k4	k5	k6	Vb	WSS
cer	dx	All	0.3318	0.0967	0.0114	2.1530e-02	0	0	3.5e-02	7.2254e+00
cer	sin	All	0.3158	0.1002	0.0165	3.0019e-02	0	0	3.5e-02	6.3000e+00
fcx	dx	All	0.3384	0.0993	0.0325	3.5848e-02	0	0	3.5e-02	8.9859e+00
fcx	sin	All	0.3353	0.0968	0.0290	2.9830e-02	0	0	3.5e-02	8.8621e+00
pit	.	All	0.2155	0.1214	0.0889	1.2236e-03	0	0	3.5e-02	3.1444e+00

tha	dx	All	0.4339	0.1511	0.1657	3.7412e-02	0	0	3.5e-02	1.1825e+01
tha	sin	All	0.4254	0.1567	0.1939	4.1823e-02	0	0	3.5e-02	1.2451e+01
cer	avg	All	0.3239	0.0986	0.0141	2.6683e-02	0	0	3.5e-02	6.5768e+00
fcx	.	All	0.3369	0.0980	0.0306	3.2755e-02	0	0	3.5e-02	8.9056e+00
tha	.	All	0.4288	0.1519	0.1766	3.9441e-02	0	0	3.5e-02	1.1827e+01

fitglob 1.0 (c) 2002 by Turku PET Centre

Date: 2002-12-10 11:16:07

Data file: ub0001.dft

Plasma file: ub0001ap.pure.kbq

Blood file: ub0001ap.kbq

Data range: 69 min (N=16)

Data was not weighted.

Region			K1	k2	k3	k4	k5	k6	Vb	WSS
cer	dx	All	0.3318	0.0967	0.0114	2.1529e-02	0	0	3.5e-02	7.2254e+00
cer	sin	All	0.3158	0.1002	0.0165	3.0008e-02	0	0	3.5e-02	6.3000e+00
fcx	dx	All	0.3384	0.0993	0.0325	3.5848e-02	0	0	3.5e-02	8.9859e+00
fcx	sin	All	0.3353	0.0968	0.0290	2.9833e-02	0	0	3.5e-02	8.8621e+00
pit	.	All	0.2155	0.1214	0.0890	1.2251e-03	0	0	3.5e-02	3.1444e+00
tha	dx	All	0.4339	0.1511	0.1656	3.7382e-02	0	0	3.5e-02	1.1825e+01
tha	sin	All	0.4255	0.1570	0.1946	4.1871e-02	0	0	3.5e-02	1.2451e+01
cer	avg	All	0.3239	0.0986	0.0141	2.6682e-02	0	0	3.5e-02	6.5768e+00
fcx	.	All	0.3369	0.0980	0.0306	3.2750e-02	0	0	3.5e-02	8.9056e+00
tha	.	All	0.4288	0.1514	0.1757	3.9377e-02	0	0	3.5e-02	1.1826e+01

fitglob 1.0 (c) 2002 by Turku PET Centre

Date: 2002-12-10 11:16:33

Data file: ub0001.dft

Plasma file: ub0001ap.pure.kbq

Blood file: ub0001ap.kbq

Data range: 69 min (N=16)

Data was not weighted.

Region			K1	k2	k3	k4	k5	k6	Vb	WSS
cer	dx	All	0.3318	0.0967	0.0114	2.1529e-02	0	0	3.5e-02	7.2254e+00
cer	sin	All	0.3158	0.1002	0.0165	3.0020e-02	0	0	3.5e-02	6.3000e+00
fcx	dx	All	0.3384	0.0993	0.0325	3.5848e-02	0	0	3.5e-02	8.9859e+00
fcx	sin	All	0.3353	0.0968	0.0290	2.9830e-02	0	0	3.5e-02	8.8621e+00
pit	.	All	0.2155	0.1214	0.0890	1.2250e-03	0	0	3.5e-02	3.1444e+00
tha	dx	All	0.4338	0.1509	0.1655	3.7413e-02	0	0	3.5e-02	1.1825e+01
tha	sin	All	0.4254	0.1567	0.1940	4.1823e-02	0	0	3.5e-02	1.2451e+01
cer	avg	All	0.3239	0.0986	0.0141	2.6682e-02	0	0	3.5e-02	6.5768e+00
fcx	.	All	0.3369	0.0980	0.0306	3.2755e-02	0	0	3.5e-02	8.9056e+00
tha	.	All	0.4288	0.1514	0.1757	3.9377e-02	0	0	3.5e-02	1.1826e+01