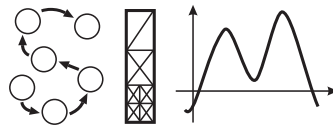


**FRIEDRICH-ALEXANDER-UNIVERSITÄT ERLANGEN-NÜRNBERG**  
INSTITUT FÜR INFORMATIK (MATHEMATISCHE MASCHINEN UND DATENVERARBEITUNG)

**Lehrstuhl für Informatik 10 (Systemsimulation)**



**Dipollokalisierung bei Epilepsiepatienten**

Daniel Gerlach

Studienarbeit

# **Dipollokalisierung bei Epilepsiepatienten**

Daniel Gerlach

Studienarbeit

Aufgabensteller: Prof. Dr. U. Rüdé

Betreuer: Dipl.-Math. M. Mohr

Bearbeitungszeitraum: 20.10.2001 – 31.1.2002

**Erklärung:**

Ich versichere, daß ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und daß die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Erlangen, den 13. März 2002

.....

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Der Nelder-Mead Algorithmus</b>	<b>4</b>
2.1	Theorie zum Nelder-Mead Algorithmus . . . . .	4
2.2	Die Implementierung . . . . .	10
2.3	Anwendung des Nelder-Mead Algorithmus . . . . .	13
<b>3</b>	<b>Dipollokalisierung</b>	<b>19</b>
3.1	Hintergrund und Theorie . . . . .	19
3.2	Deviation Scan und Dipollokalisierung . . . . .	25
3.3	Anwendung auf 2D Testfälle . . . . .	29
3.4	Verhalten bei wirklichen Meßwerten . . . . .	39
3.5	Einfluß von Störungen auf die Dipollokalisierung . . . . .	47
<b>4</b>	<b>Zusammenfassung</b>	<b>54</b>
<b>5</b>	<b>Ausblick</b>	<b>56</b>

# Kapitel 1

## Einleitung

Bei Patienten mit fokaler Epilepsie gibt es erkrankte Gehirnteile, die zeitweise aktiv werden und somit zu epileptischen Anfällen führen. Um den Patienten von seiner Krankheit zu befreien, muß der besagte Gehirnteil operativ entfernt werden. Dies setzt eine genaue Kenntnis der Position voraus. Da sich das Gewebe des erkrankten Gehirnteils nicht vom restlichen unterscheidet, somit also nicht durch klassische bildgebende Verfahren wie MRI und CT gefunden werden kann, bedient man sich der numerischen Simulationen basierend auf den Maxwell's Gleichungen.

Der erkrankte Teil des Gehirns verursacht während eines epileptischen Anfalls charakteristische bioelektrische Felder, die durch ein Elektroencephalogramm (EEG) aufgenommen werden können. Diese Spannungsmesswerte auf der Kopfoberfläche des Patienten stellen den Ausgangspunkt der numerischen Simulation dar. Basierend auf diesen Meßdaten soll nun auf die Quelle zurückgeschlossen werden. Mathematisch stellt dies ein inverses Problem dar, da nicht, wie im klassischen Sinne, durch Lösen eines elliptischen Randwertproblems Potentialwerte auf der Kopfoberfläche berechnet werden. Die Lösung des inversen Problems wird dadurch erschwert, daß die EEG-Messwerte zum einen durch das Meßverfahren selbst und zum anderen durch andere Gehirnaktivitäten mit Fehlern behaftet sind.

Für eine numerische Simulation muß das Problem durch ein geeignetes Modell beschrieben werden. Die erkrankte Gehirnstelle wird durch einen Dipol ersetzt, der durch seinen Ort, seine Orientierung und Stärke definiert ist. Im Fall der fokalen Epilepsie beschränkt sich die erkrankte Gehirnstelle auf einen einzelnen, begrenzten

Bereich, so daß im Modell nur ein einziger Dipol vorhanden ist. Aus Magnetresonanzenaufnahmen werden verschiedenen Kopf-Compartments segmentiert (Luft, Kopfhaut, Knochen, Gehirnmasse und ventrikuläres System) und damit ein Voxelmodell des Kopfs erstellt. Für einen definierten Dipol lassen sich somit die Potentialwerte an den Elektroden, definiert an der Kopfoberfläche, durch Lösen eines sogenannten Vorwärtsproblems berechnen. Ein solches Vorwärtsproblem besteht aus der Lösung eines klassischen elliptischen Randwertproblems auf dem 3D Kopfmodell.

Ein Standardverfahren zur Lösung des inversen EEG-Problems ist die Dipollokalisierung. Hierbei sollen die freien Parameter des Dipols (Position, Orientierung und Stärke) so gefunden werden, so daß sich die damit ergebenden Meßwerte den wahren möglichst gut annähern. Für diese Optimierungsaufgabe ist in dieser Arbeit der Nelder-Mead Algorithmus [1] verwendet worden. Dieses Verfahren basiert auf einer geometrischen Figur, dem Simplex, das seine Form nach vorgegebene Regeln verändert und sich um ein lokales Minimum der gegebenen Funktion iterativ zusammenzieht. Durch die Berechnung verschiedener Testpunkte wird in jeder Iteration ein verbessertes Simplex gesucht, in dem Sinne, daß sich die Funktionswerte an den Eckpunkten verringert haben. Vorteile des Nelder-Mead Algorithmus sind, daß er keine Ableitungen der Funktion verwendet und im Allgemeinen nur wenige Funktionsauswertungen für ein neues Simplex benötigt. Im Fall der Dipollokalisierung minimiert der Nelder-Mead Algorithmus die normierte Differenz aus den EEG-Meßwerten und sogenannten virtuellen Meßwerten. Diese werden aus der Lösung eines Vorwärtsproblems für eine virtuelle Dipolposition gewonnen. Damit sind für das inverse Problem einige Vorwärtsprobleme zu lösen. Suchen wir den Dipol nach dem obigen Verfahren, so ist das Problem überbestimmt. Deshalb muß zur Berechnung der Dipolorientierung und -stärke für einen gegebenen Ort zusätzlich ein Least Squares Problem gelöst werden.

In der vorliegenden Studienarbeit ist zuerst der Nelder-Mead Algorithmus implementiert worden. Die Theorie des Algorithmus und Hinweise zur Implementierung sind in Kapitel 2.1 und 2.2 beschrieben. Danach ist der Code an einfachen Funktionen kontrolliert und getestet worden. Am Beispiel der Rosenbrock Funktion wird dies in Kapitel 2.3 dargestellt. In Kapitel 3.1 wird der Hintergrund und die mathematische Beschreibung der Dipollokalisierung gegeben. Basierend auf einem weiteren

Standardverfahren, dem Deviation Scan, wird die Methode der Dipollokalisierung erklärt (Kap. 3.2). Anschließend wird die Dipolsuche an 2D Testdaten (Kap. 3.3) und an wirklichen EEG-Meßwerten (Kap. 3.4) untersucht. In Kapitel 3.5 wird der Einfluß von Meßstörungen analysiert. Abschließend wird eine Zusammenfassung und mögliche Ausblicke gegeben.

# Kapitel 2

## Der Nelder-Mead Algorithmus

### 2.1 Theorie zum Nelder-Mead Algorithmus

Seit seiner Veröffentlichung 1965 [1] ist der *Nelder-Mead Algorithmus* einer der bekanntesten Verfahren zur Lösung von nichtlinearen Optimierungsproblemen. Die Nelder-Mead Methode zählt zu der Klasse der Direkten Suchverfahren, da zur Lösung des Problems keine Ableitungen benötigt werden. Als Subklasse der Direkten Suchverfahren gehört der Algorithmus zu denjenigen Methoden, die in jeder Iteration ein feststehendes *Simplex* besitzen. Diese geometrische Figur stellt im  $n$ -dimensionalen Raum die konvexe Hülle von  $n + 1$  Eckpunkten dar. Mit Hilfe dieses Simplex minimiert der Nelder-Mead Algorithmus den Skalarwert einer nichtlineare Funktion. Dazu werden die Eckpunkte des Simplex von Iteration zu Iteration verändert, und passen sich somit den Konturen der vorgegebenen Funktion an, um sich dem Minimum zu nähern.

Obwohl der Nelder-Mead Algorithmus so weit verbreitet ist und schon seit 1965 angewandt wird, ist die Konvergenz der Methode bis heute nicht theoretisch nachgewiesen. Sogar für einfache Funktionen kann nicht vorhergesagt werden, ob das Verfahren immer zu einem Minimalwert konvergiert. In verschiedenen Veröffentlichungen sind Funktionen vorgestellt worden, bei denen sich das Simplex um einen nicht-minimalen Wert zusammenzieht (z.B. [4] und [5]). Dieses Verhalten wird bis heute nicht verstanden. Lagarias et al. [2] zeigten wie schwer es ist, theoretisch die



Konvergenz des Verfahrens zu belegen. Sie wendeten den Nelder-Mead Algorithmus auf streng konvexe Funktionen an und erreichten Konvergenzergebnisse für die erste und zweite Dimension. Aber auch sie kommen zu dem Schluß, daß bis jetzt für keine Funktion einer Dimension größer als eins bewiesen ist, daß der original Nelder-Mead Algorithmus immer zu einem Minimalwert konvergiert.

Trotz des Fehlens eines theoretischen Konvergenznachweises ist das Nelder-Mead Verfahren weit verbreitet im Chemieingenieurwesen, der Chemie und der Medizin. Folgende Gründe können dafür gegeben werden: In vielen Anwendungen sind die Funktionswertberechnungen sehr aufwendig und teuer oder Ableitungen können nicht berechnet werden. Der Vorteil des Nelder-Mead Algorithmus liegt nun darin, daß er in der Regel nur eine geringe Anzahl von Funktionswertberechnungen und keine Ableitungen benötigt, um ein verbessertes Simplex zu erzielen. Im Gegensatz dazu brauchen andere Suchverfahren (z.B. "pattern search" Methoden)  $n$  Funktionswertberechnungen für die Bildung des nächsten Simplex. Eine weitere Eigenschaft ist, daß sich der Nelder-Mead Algorithmus schon nach wenigen Iterationen dem Minimalwert stark angenähert hat. Oft reichen solche Näherungswerte aus. Außerdem ist das Verfahren sehr beliebt, weil es robust gegenüber Unstetigkeiten ist, weil es leicht zu programmieren und jeder Schritt verständlich ist.

Das Problem kann wie folgt geschrieben werden:

$$\min_{\nu \in \mathbb{R}^n} f(\nu). \quad (2.1)$$

Eine Lösung von 2.1 mit dem Nelder-Mead Algorithmus läuft auf folgende Weise ab: Gestartet wird mit einem Simplex bestehend aus den Eckpunkten  $x_1, x_2, \dots, x_{n+1}$ , wobei  $n$  die Dimension des Problems ist. Für alle Eckpunkte wird das Funktional berechnet und nach der Größe des Funktionswerts sortiert, so daß  $x_1$  den kleinsten Funktionswert besitzt. Mit anderen Worten bedeutet dies, daß  $x_1$  bester und  $x_{n+1}$  schlechtester Eckpunkt ist. Ausgehend von diesem Simplex werden verschiedene Testpunkte mit ihrem Funktionswert berechnet und durch Vergleiche mit den alten Eckpunkten ein neues Simplex gebildet. Das neue Simplex muß sich im Vergleich zum alten in dem Sinne verbessert haben, daß sich der Nelder-Mead Algorithmus durch eine Folge von Iterationen dem Minimum annähert.

Im Nelder-Mead Algorithmus stehen vier mögliche Schritte zur Verfügung, um

eine Verbesserung des Simplex zu erzielen. *Reflektion*, *Expansion*, *innere* und *äußere Kontraktion* versuchen, den schlechtesten Eckpunkt  $x_{n+1}$ , zu verbessern. Gelingt dies nicht, so wird eine *Schrumpfung* um den besten Punkt  $x_1$  durchgeführt, bei dem alle Eckpunkte außer dem Besten neu berechnet werden. In Abbildung 2.1 bis 2.3 sind alle fünf Schritte für den zwei dimensionalen Raum dargestellt. Nach jeder Iteration steht also entweder ein neuer Eckpunkt als Verbesserung des vorher Schlechtesten zur Verfügung oder es werden durch eine Schrumpfung  $n$  neue Punkte berechnet, die zusammen mit dem vorher Besten das neue Simplex darstellen. Daraus ergibt sich, daß nach jeder Iteration entweder ein neuer Punkt in die Liste der Funktionwerte eingeordnet (Iteration ohne Schrumpfung) oder alle Punkte neu sortiert werden müssen (Iteration mit Schrumpfung). Für beide Fälle gibt es Regeln wie die Eckpunkte neu geordnet werden müssen:

**Ordnungsregel für Iteration ohne Schrumpfung.** Falls keine Schrumpfung durchgeführt worden ist, wird der schlechteste Punkt  $x_{n+1}^{(k)}$  durch den neuen Punkt  $v$  ersetzt, wobei  $k$  die jeweilige Iteration bezeichnet. Dieser neue Punkt nimmt nun in der geordneten Liste der Eckpunkte Position  $j$  an, wobei

$$j = \min_{1 \leq l \leq n+1} \{l \mid f(v^{(k)}) < f(x_l^{(k)})\}. \quad (2.2)$$

**Ordnungsregel für Iteration mit Schrumpfung.** Ist eine Schrumpfung durchgeführt worden, so bleibt nur der beste Punkt  $x_1^{(k)}$  im neuen Simplex erhalten. Alle anderen Eckpunkte werden neu berechnet. Die einzige Regel zur Neuordnen der Eckpunkte für den Fall, daß  $x_1^{(k)}$  und einer oder mehrere der neuen Punkte den gleichen Funktionswert haben, ist: Falls

$$\min\{f(v_2^{(k)}), \dots, f(v_{n+1}^{(k)})\} = f(x_1^{(k)}), \quad (2.3)$$

dann muß der beste Punkt des alten Simplex gleich dem neuen Besten sein, d.h.  $x_1^{(k+1)} = x_1^{(k)}$ . Neben dieser Einschränkung kann jede Regel zur Neuordnung angewandt werden.

Damit jeder Schritt des Nelder-Mead Algorithmus definiert ist, müssen vier Parameter festgelegt werden: Reflektion ( $\rho$ ), Expansion ( $\chi$ ), Kontraktion ( $\gamma$ ) und Schrumpfung ( $\sigma$ ). Wie im original Nelder-Mead Algorithmus beschrieben, sollten

die Parameter folgende Bedingungen erfüllen:

$$\rho > 0, \chi > 1, \chi > \rho, 0 < \gamma < 1 \quad \text{und} \quad 0 < \sigma < 1. \quad (2.4)$$

Im *original* Nelder-Mead Algorithmus sind die Parameter definiert als

$$\rho = 1, \chi = 2, \gamma = 0.5 \quad \text{und} \quad \sigma = 0.5. \quad (2.5)$$

Eine Iteration des Nelder-Mead Algorithmus besteht aus folgenden Schritten:

1. **Ordnen.** Ordne die  $n + 1$  Eckpunkte so, daß  $f(x_1) \leq f(x_2) \leq \dots \leq f(x_{n+1})$ .
2. **Reflektion.** Berechne den *Reflektionspunkt*  $x_r$  aus

$$x_r = \bar{x} + \rho(\bar{x} - x_{n+1}) = (1 + \rho)\bar{x} - \rho x_{n+1}, \quad (2.6)$$

wobei  $\bar{x} = \sum_{i=1}^n x_i/n$  als Schwerpunkt der  $n$  besten Punkte definiert ist (alle Eckpunkte außer  $x_{n+1}$ ). Berechne  $f_r = f(x_r)$ .

Wenn  $f_1 \leq f_r < f_n$ , dann verwende Reflektionspunkt  $x_r$  und beende die Iteration.

3. **Expansion.** Wenn  $f_r < f_1$ , berechne den *Expansionspunkt*  $x_e$ ,

$$x_e = \bar{x} + \chi(x_r - \bar{x}) = \bar{x} + \rho\chi(\bar{x} - x_{n+1}) = (1 + \rho\chi)\bar{x} - \rho\chi x_{n+1}, \quad (2.7)$$

und berechne  $f_e = f(x_e)$ .

Wenn  $f_e < f_r$ , dann verwende  $x_e$  und beende die Iteration; ansonsten (wenn  $f_e \geq f_r$ ), verwende  $x_r$  und beende die Iteration.

4. **Kontraktion.** Wenn  $f_r \geq f_n$ , dann führe eine *Kontraktion* zwischen  $\bar{x}$  und dem besseren Punkt von  $x_{n+1}$  und  $x_r$  aus.

- (a) **Äußere.** Wenn  $f_n \leq f_r < f_{n+1}$  (d.h.,  $x_r$  ist besser als  $x_{n+1}$ ), dann führe eine *äußere Kontraktion* durch: berechne

$$x_c = \bar{x} + \gamma(x_r - \bar{x}) = \bar{x} + \gamma\rho(\bar{x} - x_{n+1}) = (1 + \rho\gamma)\bar{x} - \rho\gamma x_{n+1}, \quad (2.8)$$

und berechne  $f_c = f(x_c)$ .

Wenn  $f_c \leq f_r$ , dann verwende  $x_c$  und beende die Iteration; ansonsten, gehe zu Schritt 5 (führe eine Schrumpfung durch).

(b) **Innere.** Wenn  $f_r \geq f_{n+1}$ , führe eine *innere Kontraktion* durch: berechne

$$x_{cc} = \bar{x} - \gamma(\bar{x} - x_{n+1}) = (1 - \gamma)\bar{x} + \gamma x_{n+1}, \quad (2.9)$$

und berechne  $f_{cc} = f(x_{cc})$ .

Wenn  $f_{cc} < f_{n+1}$ , dann verwende  $x_{cc}$  und beende die Iteration; ansonsten, gehe zu Schritt 5 (führe eine Schrumpfung durch).

5. **Schrumpfung.** Berechne  $f$  an den  $n$  Punkten  $v_i = x_1 + \sigma(x_i - x_1)$ ,  $i = 2, \dots, n+1$ . Die (ungeordneten) Eckpunkte des Simplex der nächsten Iteration besteht aus  $x_1, v_2, \dots, v_{n+1}$ .

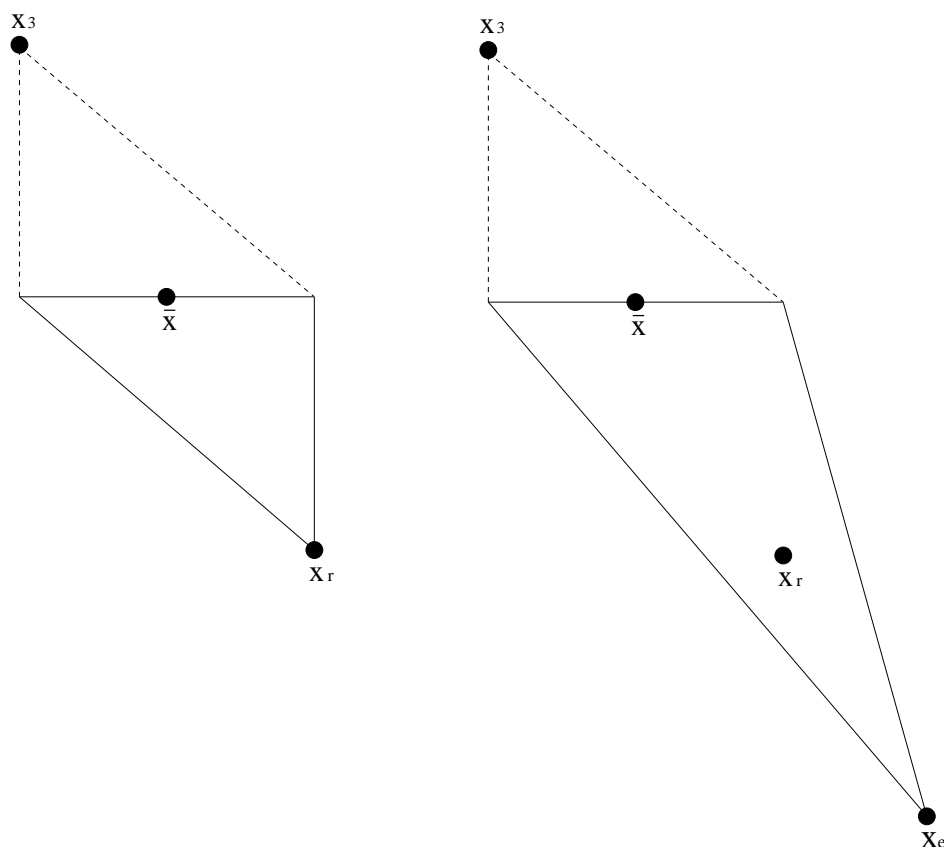


Abbildung 2.1: Veränderung des Simplex durch Reflektion und Expansion im 2D Fall. Gestrichelte Linien sind die alten und durchgezogene Linien die neuen Dreiecke.

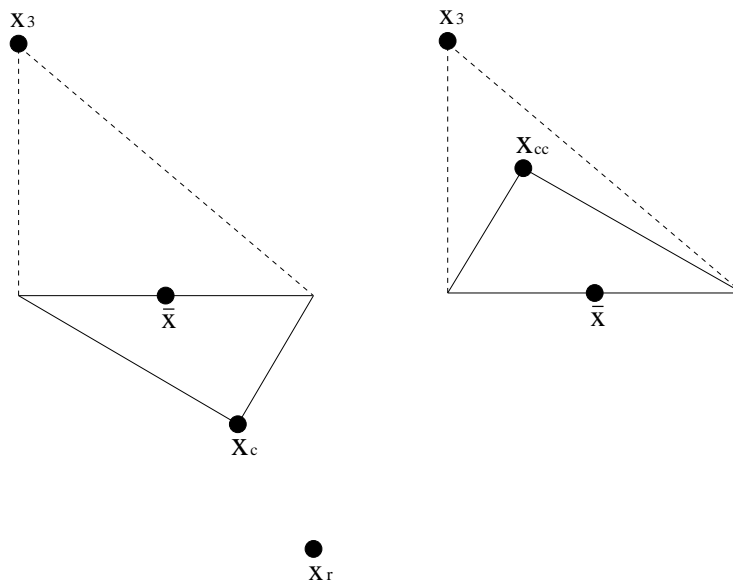


Abbildung 2.2: Veränderung des Simplex durch äußere und innere Kontraktion im 2D Fall. Gestrichelte Linien sind die alten und durchgezogene Linien die neuen Dreiecke.

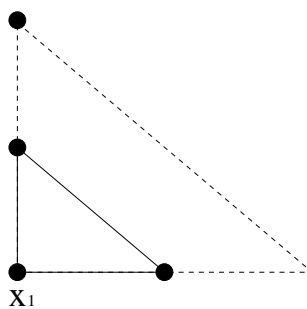


Abbildung 2.3: Veränderung des Simplex durch Schrumpfung im 2D Fall. Gestrichelte Linien sind die alten und durchgezogene Linien die neuen Dreiecke.

## 2.2 Die Implementierung

Neben dem ursprünglichen Nelder-Mead Algorithmus [1] gibt es heute eine Vielzahl von abgeänderten Versionen. Der hier implementierte Algorithmus entspricht dem original Nelder-Mead Verfahren wie er bei Lagarias et al. [2] beschrieben und in Kapitel 2.1 vorgestellt ist. Bei der Implementierung des Nelder-Mead Algorithmus ist, neben seiner fehlerfreien Funktionsweise, besonders darauf geachtet worden, daß der Aufbau des Algorithmus möglichst universell gehalten wurde. Somit kann der Nelder-Mead Algorithmus problemlos auch auf andere Optimierungen angewendet werden. Um die eigentlichen Schritte des Verfahrens gemäß dem Pseudocode vom Anwender abzukapseln, besteht das Programm aus

- dem Hauptprogramm, in dem das Startsimplex und Parameter (Abbruchkriterium, Abbruchtoleranz und die Nelder-Mead Parameter 2.5) festgelegt und der Lösungsalgorithmus gestartet werden,
- der eigentlichen Nelder-Mead Routine, in der sich das Simplex iterativ mit Hilfe der Nelder-Mead Schritt (Reflektion, Expansion, Kontraktion und Schrumpfung) dem Funktionsminimum annähert,
- einer Funktion, die den Funktionswert eines Eckpunkts berechnet und
- einer Loggingfunktion, mit der der Anwender in jeder Iteration von ihm bestimmte Variablen herausschreiben lassen kann.

Das implementierte Interface und die verwendeten Strukturen der Lösungsroutine des Nelder-Mead Algorithmus sind nachfolgend dargestellt:

**Interface:**

```
int nm_algorithm( int dim,  
                  int max_iterations,  
                  int tolerance_criteria,  
                  double tolerance,  
                  point *simplex,  
                  point *opt_point,  
                  int opt_point_is,
```

```

nm_parameter *nm_para,
void (*function)(point *, problem_data *),
problem_data *prob_data,
void (*logging)(int ,int ,point *,int *ind, problem_data *)

```

**Strukturen:**

```

typedef struct { double *coord;
                double fval; } point;
typedef struct { double rho;
                double chi;
                double gamma;
                double sigma; } nm_parameter;

```

Die Übergabeparameter des Interface sind:

- die Dimension des Problems
- ein Maximalwert für die Anzahl von Iterationen
- ein Integer, um zwischen verschiedenen Abbruchkriterien auswählen zu können
- der Toleranzwert für das gewählte Abbruchkriterium
- ein Zeiger auf einen Vektor von Eckpunkten des Typs *point*, die das aktuelle Simplex bilden; die Struktur *point* besteht aus  $n$  Koordinaten und einem Funktionswert, wobei  $n$  die Dimension bezeichnet
- ein einzelner *point*, in dem der optimale Punkt, das Ergebnis gespeichert wird
- einen Integer, mit dem man die Lage des optimalen Punktes bestimmen kann; als optimaler Punkt kann gewählt werden zwischen dem Eckpunkt mit dem kleinsten Funktionswert oder dem Schwerpunkt des Simplex, jeweils für die Figur der letzten Iteration
- eine Struktur, die die vier Parameter der Nelder-Mead Schritte beinhaltet
- ein Zeiger auf eine vom Benutzer definierte Funktion, die den zu minimierenden Funktionswert für einen Eckpunkt berechnet; übergeben wird der Funktion der

*point*, für den der Funktionswert berechnet werden soll, und weitere notwendige Daten (*problem\_data*); hängt die Funktion nur von den Koordinaten ab, so brauchen keine *problem\_data* übergeben zu werden.

- die spezifischen Problemdaten, die auch an die Nelder-Mead Routine selbst übergeben werden
- ein Zeiger auf eine Loggingfunktion, mit der der Benutzer bei jeder Iteration Daten herausschreiben lassen kann; die Übergabeparameter dieser Funktion sind die Dimension, der aktuelle Iterationszähler, das gesamte Simplex mit Koordinaten und Funktionswert, einen Integervektor, um die Reihenfolge der nach der Größe geordneten Eckpunkte festzuhalten, und die Problemdaten

Der Integervektor `ind` wird aus folgendem Grund verwendet: Nach jeder Iteration müssen die Eckpunkte neu nach der Größe ihres Funktionswerts geordnet werden. Bei Problemen mit sehr großer Dimension kann dies aufwendig werden, da z.B. nach einer Schrumpfung alle  $n + 1$  Eckpunkte mit ihren  $n$  Koordinaten und dem Funktionswert durch Kopieren neu sortiert werden müssen. Um dies zu vermeiden, haben wir einen Indexvektor definiert, der aus  $n$  Integern besteht. Zum Sortieren wird nur die Position des Eckpunkts im Simplex an die richtige Stelle im Indexvektor geschrieben.

Wie gesagt, müssen nach einer Schrumpfung alle  $n + 1$  Punkte neu sortiert werden. Dafür wurde eine direkte Sortiermethode verwendet ([6]), die auf folgende Weise abläuft: Als Beispiel soll die Reihe von Integern

109 75 200 25 38 19 150 11 20

der Größe nach sortiert werden. Wir durchsuchen nun die Folge und vertauschen den kleinsten Integer (11) mit dem ersten Element (109). Der erste Integer der Folge befindet sich nun an der richtigen Position, so daß wir uns nur noch um die übrigen Elemente kümmern müssen beginnend mit 75.

11 75 200 25 38 19 150 109 20

Aus dieser Menge suchen wir nun wieder das kleinste Element (19) heraus und vertauschen es mit dem zweiten Wert.

11 19 200 25 38 75 150 109 20



Nun fahren wir mit den Elementen ab 200 fort usw. Somit sind am Ende die Interger so sortiert, daß sie mit dem kleinsten Wert beginnen. Angewandt auf unser Problem erfüllt der Algorithmus die nötigen Anforderungen, da er auch die Ordnungsregel für Iterationen mit Schrumpfung (Gleichung 2.3) berücksichtigt. Da der beste alte Punkt  $x_1^{(k)}$  bei der Neuordnung des Simplexvektors das erste Element in der Folge ist, bleibt er auch an dieser Position, falls einer oder mehrere “geschrumpfte” Punkte den gleichen Wert haben sollten.

Als Abbruchkriterium für den Nelder-Mead Algorithmus ist zum einen eine maximale Anzahl von Iterationen festgelegt. Zweitens kann zwischen den drei folgenden Kriterien ausgewählt werden, um den Algorithmus zu terminieren:

1. Der kleinste Funktionswert ist kleiner als die gegebene Toleranz.
2. Der Abstand zwischen den Simplexschwerpunkten von zwei aufeinanderfolgenden Iterationen ist kleiner als die Toleranz.
3. Der größte Abstand zwischen dem Simplexschwerpunkt und einem Eckpunkt ist kleiner als die Toleranz.

Das erste Kriterium setzt eine Kenntnis vom kleinsten Funktionswert voraus. Die beiden anderen orientieren sich nur an der Veränderung zwischen zwei Iterationen und der Größe des Simplex. Damit soll verhindert werden, daß sich das Simplex zu stark um einen Punkt zusammenzieht.

## 2.3 Anwendung des Nelder-Mead Algorithmus

Um zuerst einen Eindruck von der Veränderung des Simplex einer Nelder-Mead Suche vermitteln zu können, sind in Abb. 2.4 die ersten 9 Dreiecke eines Minimierungsproblems dargestellt. Als Funktional dient die konvexe Funktion

$$f(x, y) = \sqrt{x^2 + y^2}, \quad (2.10)$$

deren Konturlinien aus Kreisen um das globale Minimum  $f(0, 0) = 0$  bestehen. In der Abbildung 2.4 läßt sich nun anschaulich der Ablauf einer Nelder-Mead Optimierung mit Hilfe von Abbildung 2.1 bis 2.3 und dem Pseudocode von Kapitel 2.1 nachvollziehen. Die verschiedenen Schritte sind:

1. Expansion
2. Äußere Kontraktion
3. Äußere Kontraktion
4. Reflektion
5. Innere Kontraktion
6. Innere Kontraktion
7. Innere Kontraktion
8. Innere Kontraktion

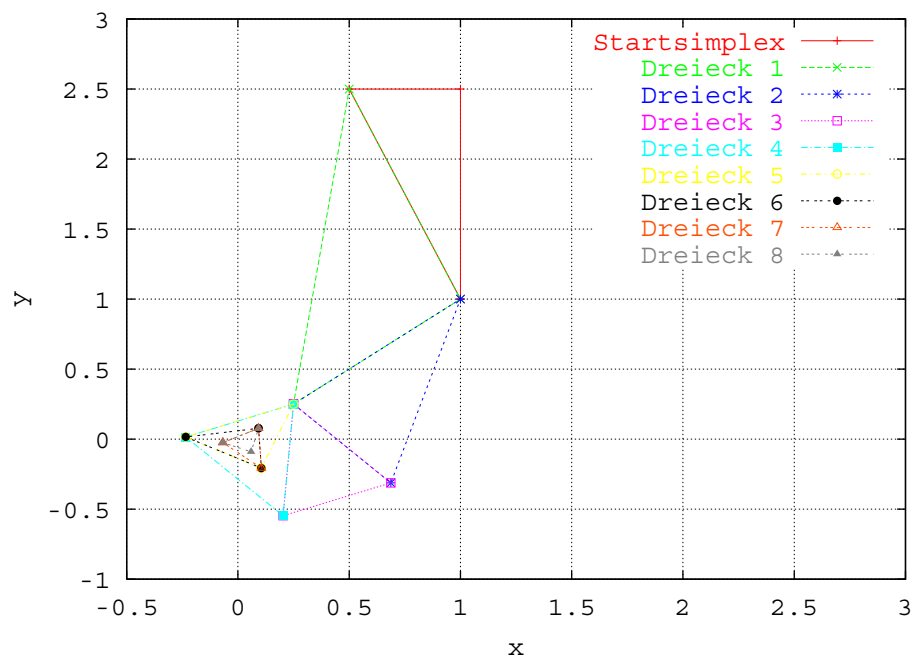


Abbildung 2.4: Die ersten neun Dreieck der Minimalwertsuche von Gleichung (2.10).

Bevor der Nelder-Mead Algorithmus zur Dipollokalisierung eingesetzt worden ist, haben wir ihn an verschiedenen Funktionen auf seine Korrektheit getestet. Im folgenden soll am Beispiel der Rosenbrock Funktion die Suche des Funktionsminimums mit der Nelder-Mead Methode gezeigt werden.

Die Rosenbrock Funktion in 2D ist wie folgt definiert

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2 \quad (2.11)$$

und besitzt ihr globales Minimum bei  $f(1, 1) = 0$ . Diese Funktion stellt eine klassische Testfunktion für Suchalgorithmen dar, was bei der Betrachtung ihrer Konturlinien in Abbildung 2.5 leicht nachvollziehbar ist. Das Minimum liegt in einem

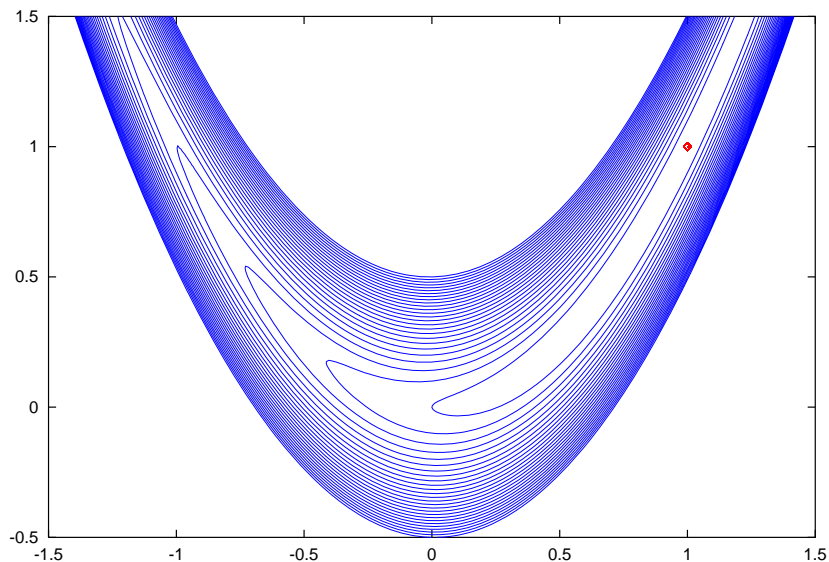


Abbildung 2.5: Konturlinien der Rosenbrock Funktion. Das Minimum befindet sich bei  $(1, 1)$  mit dem Funktionswert 0. Z.B ist der Funktionswert bei  $f(0, 0) = 1$  und bei  $f(1.5, 0.5) = 306.5$ .

schmalen langgezogenen Tal, dessen Wände auf beiden Seiten sehr steil ansteigen. Da die Funktion in 2D definiert ist ( $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ ), entspricht das Simplex einem Dreieck. Mit Hilfe der Loggingfunktion haben wir am Ende jeder Iteration die Koordinaten und den kleinsten Funktionswert des Simplex herausgeschrieben und dargestellt (Abb. 2.6 und 2.7). Der Suchalgorithmus bewegt sich, entlang des größten Gradienten, in das schmale Tal hinein und kommt dann nur noch sehr langsam mit überwiegend Reflektionen und Expansionen in Richtung des Minimums voran. Am Verlauf der Dreiecke erkennt man deutlich den Charakter der Konturlinien. In dem dargestellten Beispiel sind 94 Iterationen durchgeführt worden. Als Abbruchkriterium ist hier gefordert worden, daß der Funktionswert kleiner als 0.001 sein muß, was nur aufgrund der Kenntnis des Minimalwerts sinnvoll war.

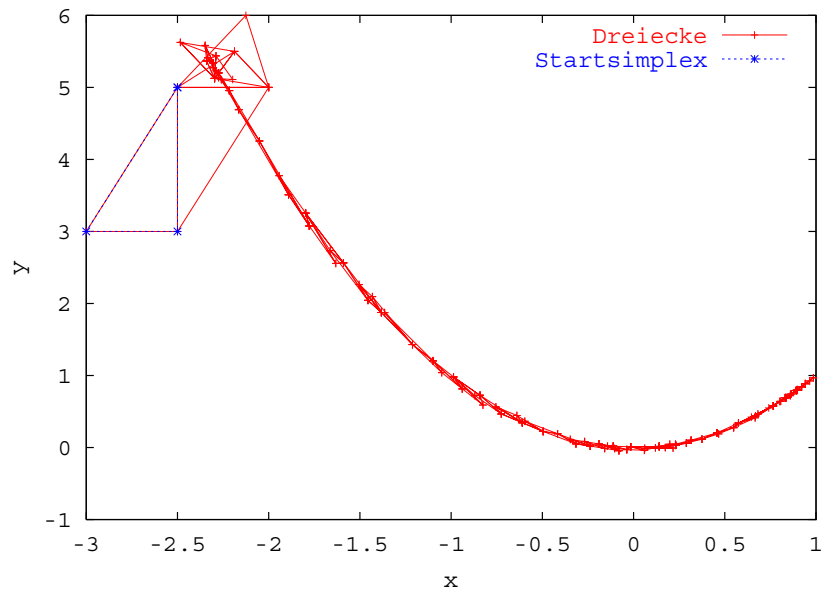


Abbildung 2.6: Veränderung der Dreiecke während der Minimierung.

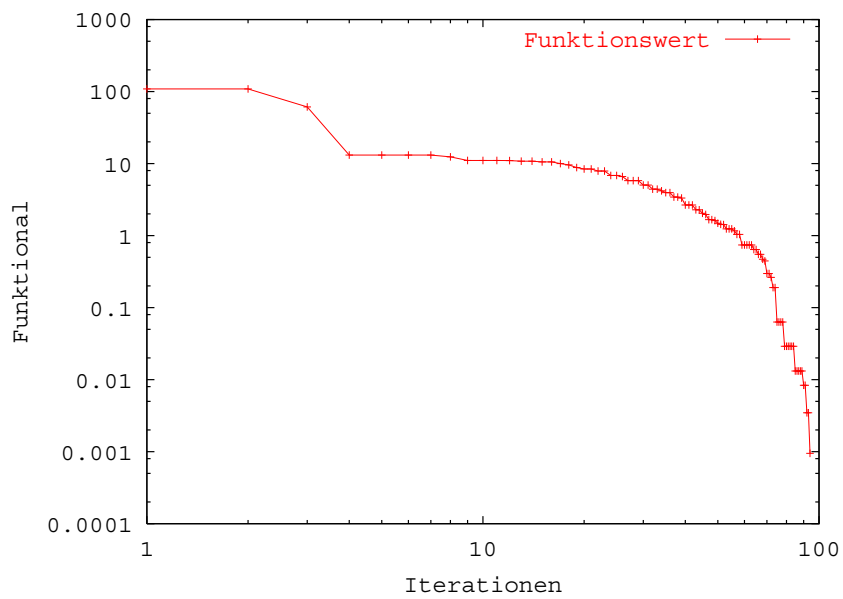


Abbildung 2.7: Verlauf des kleinsten Funktionswerts pro Iteration.

In der Darstellung des kleinsten Funktionswerts pro Iteration (Abb. 2.7) kann man beobachten, daß der Funktionwert entweder gleich bleibt oder kleiner wird. Bei genauerer Betrachtung des Pseudocodes des Nelder-Mead Algorithmus (Kap. 2.1) versteht man, daß der kleinste Funktionswert entweder durch Reflektion, Expansion oder Kontraktion verbessert wird oder gleich bleibt, was bedeutet, daß ein anderer Eckpunkt verbessert wird oder durch eine Schrumpfung alle Punkte außer dem Besten neu berechnet werden. Der kleinste Funktionswert fällt also monoton. Durch die logarithmische Auftragung ist nicht sofort erkennbar, daß sich der Funktionwert zu Beginn sehr stark verbessert und dann nur noch sehr langsam gegen das Minimum konvergiert, was schon am Verlauf der Dreiecke sichtbar wurde. Die Ergebnisse zeigen deutlich, daß der Nelder-Mead Algorithmus das Minimum der Rosenbrock Funktion zuverlässig gefunden hat, was sich auch bei der Verwendung von andere Startsimplices ergab.

Untersucht man die Veränderungen des Simplex, wenn es sich schon sehr um das Minimum zusammengezogen hat, so stellt man fest, daß immer noch verschiedene Nelder-Mead Schritte durchgeführt werden. Der Algorithmus fährt sich also nicht auf einen Schritt fest. Im Folgenden soll veranschaulicht werden, mit welchen Nelder-Mead Schritten sich das Simplex hauptsächlich vorwärtsbewegt. Dafür sind jeweils 100 Nelder-Mead Iterationen mit der Rosenbrock Funktion (2.11), der Kreisfunktion (2.10), einer ungestörten und gestörten 2D Dipollokalisierung durchgeführt worden. Die beiden letzten Anwendungen des Nelder-Mead Algorithmus werden später beschrieben. Die Anzahl der durchgeführten Nelder-Mead Schritte, durch die das nächste Simplex definiert worden ist, sind:

N.-M. Schritt	Rosenbrock	Kreis	Dipol 1	Dip. 1 gr. 2, $SNR = 0.1$
Reflektion	40	11	22	40
Expansion	22	1	4	8
äußere Kontrak.	8	19	7	3
innere Kontrak.	30	69	57	47
Schrumpfung	0	0	10	2

Man sieht, daß sich die Anteile der Nelder-Mead Schritte für alle Funktionen unterscheiden. Für die ersten beiden Fälle liegt dies an der Definition der Startsimplices.

Im Fall der Rosenbrock Funktion liegt das Startsimplex in großer Entfernung zum Minimum und muß sich erst mit vielen Reflektionen und Expansionen im Tal weiterbewegen, bevor es sich mit Kontraktionen zusammenzieht. Bei der Kreisfunktion ist das Minimum schnell gefunden, wie Abbildung 2.4 zeigt. Danach werden überwiegend Kontraktionen durchgeführt und es wird eine sehr viel größere Genauigkeit erreicht. Bei den beiden Dipollokalisierungen sind die Startsimplices relativ klein gewählt worden, so daß sich das Simplex wie bei der Rosenbrock Funktion zuerst mit Reflektionen und Expansionen nähert. Außerdem kommen auch Schrumpfun-gen vor. Dies weist darauf hin, daß das Funktional dort komplizierter ist, da die Schrumpfung eigentlich nur eine Notlösung ist.

# Kapitel 3

## Dipollokalisierung

### 3.1 Hintergrund und Theorie

Der implementierte und getestete Nelder-Mead Algorithmus soll nun zur Auffindung von erkrankten Stellen im Gehirn durch numerische Simulationen praktische Anwendung finden. Im Fall eines Patienten mit fokaler Epilepsie wird der erkrankte Teil des Gehirns zeitweise aktiv und es kommt zum epileptischen Anfall. Um diesen Teil des Gehirns operativ entfernen zu können, muß er genau lokalisiert werden. Die Aufgabe, die Gehirnstelle mit Hilfe von numerischen Simulationen zu finden, ergibt sich, weil das erkrankte Gehirngewebe nicht durch seine Zusammensetzung vom restlichen Gehirn unterschieden werden kann und somit der Lokalisation durch klassische bildgebende Verfahren wie MRI und CT unzugänglich bleibt. Um dieses Problem zu umgehen, bedient man sich des Elektroencephalogramms (EEG). Hierbei mißt man die von der besagten Gehirnstelle ausgesendeten bioelektrische Felder, die zu Spannungsdifferenzen an den Elektroden führen. Die Messung läuft folgendermaßen ab: Die Gehirnaktivitäten des Patienten werden mit Hilfe eines Langzeit-EEG durch Elektroden am Kopf aufgenommen. Ein Neurologe sucht dann charakteristische Abschnitte aus den Daten heraus, die einen epileptischen Anfall zeigen. Solch ein Ausschnitt (Epoche) besteht aus einer Reihe von Zeitsamples, von denen wir im Folgenden eines für die Lokalisation des Dipols verwendet werden. Nun steht ein Satz von  $n$  Meßwerten zur Verfügung, wobei  $n$  die Anzahl der Elektroden am Kopf ist.

Ausgehend von diesen Meßwerten soll eine Quelle rekonstruiert werden, die die Potentialwerten auf der Kopfoberfläche verursacht. Solch ein Problem wird als *inverses Problem* bezeichnet, weil man nicht, wie im klassischen Sinne, von einer Quelle und Randbedingungen startet, sondern von den Meßwerten auf die Quelle zurückschließen will. Die Lösung des Problems wird durch Meßfehler stark erschwert. Hierbei sind zwei Fehlerquellen zu unterscheiden: Zum einen beinhalten die Meßwerte Fehler, die von dem Verfahren bzw. der Apparatur verursacht werden. Zum anderen werden bei der Messung auch immer Hintergrundaktivitäten anderer Gehirnteile aufgenommen.

Zur numerischen Simulationen des Lokalisationsproblems muß der Kopf mit der erkrankten Stelle durch ein geeignetes Modell beschrieben werden (siehe auch [3]): Neuronale Zentren, wie die besagte Stelle, bestehen im Allgemeinen aus mehreren tausend Neuronen, die gleichzeitig elektrisch aktiv sind. Es ist nun üblich, solch ein Zentrum als Dipol zu modellieren. Das einfachste Modell besteht aus einem einzelnen Dipol, was voraussetzt, daß die die Meßwerte verursachende Quelle sehr stark und auf einen kleinen Bereich konzentriert ist. Dies ist im Fall der fokalen Epilepsie erfüllt. Der Dipol ist charakterisiert durch seine Position  $r$ , die Orientierung  $d$  und die Stärke, die durch die Länge des Orientierungsvektors codiert werden kann. Ziel ist also das Finden von solchen Dipolparametern, die die Meßwerte möglichst gut annähern. Zur Suche des Dipols soll also der folgend Ausdruck minimiert werden:

$$R = \frac{\|m - m_v(r, d)\|_2}{\|m\|_2}, \quad (3.1)$$

wobei  $R$  die Relative Residuumsenergie ist,  $m$  die gemessenen Potentiale sind,  $m_v(r, d)$  die virtuellen Meßwerte für einen Dipol mit Parametern  $(r, d)$  sind und  $\|\cdot\|_2$  die euklidische Norm ist. Somit ist zur Lösung des inversen Problems für jeden gegebenen Dipol ein sogenanntes Vorwärtsproblem zu lösen. Ein Vorwärtsproblem besteht aus der Berechnung von Potentialen, d.h. von virtuellen Meßwerten, an den  $n$  Elektroden für einen gegebenen Dipol durch Lösen eines elliptischen Randwertproblems (siehe dazu [3]). Die virtuellen Meßwerte  $m_v$  lassen sich berechnen durch

$$m_v(r, d) = \hat{L}(r) \cdot d. \quad (3.2)$$

Hier ist  $\hat{L} \in \mathbb{R}^{n \times 3}$  die Leadfield Matrix für den drei-dimensionalen Fall. Sie bildet



die Orientierung auf die Potentiale an den Elektroden ab. Die Leadfield Matrix kann aufgespalten werden in

$$\hat{L}(p) = N \cdot L(r). \quad (3.3)$$

Das Produkt der Matrix  $L(r) \in \mathbb{R}^{(n-1) \times 3}$  und  $d$  bildet die Dipolorientierung auf Potentialdifferenzen zwischen einem repräsentativen Satz  $\pi$  von Elektrodenpaaren ab. Für die Berechnung müssen also aus den  $n$  Elektroden  $(n-1)$  unabhängige Elektrodenpaare  $\pi$  gebildet werden.  $N \in \mathbb{R}^{n \times (n-1)}$  ist eine Referenzmatrix, die die Spannungsdifferenzen zwischen den Elektrodenpaaren auf Potentialwerte an den Elektroden umrechnet.

Stellen wir uns nun zwei Elektroden  $A$  und  $B$  um einen Kopf angeordnet vor, dann besteht die Vor-Leadfield Matrix  $L$  nur aus einer Zeile mit den drei Einträgen

$$V_{AB}(r) = [V_{AB}^x(r), V_{AB}^y(r), V_{AB}^z(r)], \quad (3.4)$$

wobei z.B.  $V_{AB}^x(r)$  die Potentialdifferenz zwischen den beiden Elektroden ist, die ein in  $x$ -Richtung ausgerichteter Einheitsdipol am gegebenen Ort  $r$  bewirken würde. Es zeigt sich, daß die Berechnung der Leadfield Matrix für einen gegebenen Punkt  $r$  sehr aufwendig ist, da für jede Richtungskomponente ein Vorwärtsproblem gelöst werden muß. Die Aufstellung der Leadfield Matrix kann aber durch die Verwendung des Reziprozitätstheorems von Helmholtz stark vereinfacht werden. Stellen wir uns nochmals das Modell des Kopfs mit zwei Elektroden  $A$  und  $B$  vor. Ein Dipol befindet sich im Kopf mit einer bestimmten Position und Orientierung und wird im Modell durch eine Klammer ersetzt. Fließt nun ein Strom  $I_{Klammer}$  durch die Klammer, so erzeugt er eine Potentialdifferenz  $V_{AB}$  zwischen den Elektroden  $A$  und  $B$ . Im umgekehrten Falle bewirkt ein Strom  $I_{AB}$  zwischen den Elektroden eine Spannung, bzw. Potentialdifferenz,  $V_{Klammer}$  zwischen den Enden der Klammer. Das Reziprozitätstheorem von Helmholtz stellt die folgende Beziehung zwischen den Größen auf:

$$\frac{I_{AB}}{V_{Klammer}} = \frac{I_{Klammer}}{V_{AB}}. \quad (3.5)$$

Um die Potentialdifferenz  $V_{Klammer}$  zu erhalten, wird in einem Vorbereitungsschritt das Potentialfeld zwischen den beiden Elektroden  $A$  und  $B$  durch Lösung eines Vorwärtsproblems berechnet. Dabei fließt der Strom  $I_{AB}$  von der einen Elektrode

(Quelle) zur zweiten (Senke). Die Potentialdifferenz  $V_{Klammer}^x$ , als Beispiel, wird nun als Differenz der Potentiale an den Enden der Klammer berechnet, wobei die Klammer die Länge  $2h$  besitzt, wenn  $h$  die Gitterweite in die jeweilige Richtung ist:

$$V_{Klammer}^x = \tilde{V}_{AB}(x + h, y, z) - \tilde{V}_{AB}(x - h, y, z). \quad (3.6)$$

Hier bezeichnet  $\tilde{V}$  die Potentialwerte, die aus dem Potentialfeld des Vorwärtsproblems entnommen werden, entweder direkt, wenn  $(x, y, z)$  ein Gitterpunkt ist, oder durch Interpolation aus den Nachbarpunkten.

Nimmt man  $I_{Klammer}^x = \frac{1A}{2h}$  an, so daß sich mit dem Hebelarm  $2h$  ein Dipol mit Einheitsmoment ergibt, kann  $V_{AB}^x$  für die  $x$ -Komponente berechnet werden durch

$$V_{AB}^x = \frac{I_{Klammer}^x}{I_{AB}} V_{Klammer}^x. \quad (3.7)$$

$I_{AB}$  kann frei gewählt werden, da sich eine proportionale Spannung  $V_{Klammer}$  ergeben würde und somit nach Gleichung 3.7 keinen Einfluß hat. Gleiches kann für die beiden anderen Komponenten  $V_{AB}^y$  und  $V_{AB}^z$  gemacht werden. Der große Vorteil dieser Methode ist, daß nur einmalig für jedes Elektrodenpaar ein Vorwärtsproblem gelöst werden muß. Stehen die Potentialfelder der  $(n - 1)$  Elektrodenpaare zur Verfügung, kann die Leadfield Matrix für beliebig viele Dipolpositionen leicht berechnet werden. Dadurch wird der Rechenaufwand, besonders bei 3D-Problemen, stark verringert. Würden man die  $(n - 1)$  Potentialfelder nicht in einem Vorbereitungsschritt berechnen, so müßte man für jede Dipolposition drei Vorwärtsprobleme lösen, nämlich für das Potential an den Elektroden jeweils für einen Einheitsdipol in  $x$ ,  $y$  und  $z$ -Richtung. Bei der Beschreibung von Verfahren zur Dipollokalisierung im nächsten Abschnitt wird sich zeigen, daß bei einer Dipolsuche die virtuellen Meßwerte für sehr viele Dipolpositionen benötigt werden, was deutlich für die Anwendung des Reziprozitätstheorems spricht.

Mit der Aufstellung der Leadfield Matrix können die virtuellen Meßwerte jedoch noch nicht berechnet werden, da die optimale Orientierung des Dipols für einen gegebenen Ort noch nicht bekannt ist, mit denen die wirklichen Meßwerte möglichst gut erfüllt werden. Außerdem muß noch die Referenzmatrix  $N$  aufgestellt werden. Starten wir nun nochmal von Gleichung 3.2 und 3.3 und nehmen  $\varphi = (\varphi_1, \varphi_2, \dots, \varphi_n)$  als einen Vektor von Potentialwerten an den  $n$  Elektroden an. Zusätzlich bilden wir

einen Vektor  $\varphi^\pi$  mit der Länge  $(n - 1)$ , der die Potentialdifferenzen zwischen den Elektrodenpaaren beinhaltet. Eine Beziehung zwischen beiden Vektoren kann durch die Transfermatrix  $D$  durch

$$D \cdot \varphi = \varphi^\pi \quad (3.8)$$

hergestellt werden, wobei  $D \in \mathbb{R}^{(n-1) \times n}$ . Um nun  $\varphi$  aus 3.8 berechnen zu können, muß  $D$  invertiert werden. Dies ist aber nicht möglich, da  $D \in \mathbb{R}^{(n-1) \times n}$  nicht quadratisch ist. Dieses Problem läßt sich beheben, indem man eine zusätzliche Bedingung einführt und somit  $D$  auf eine quadratische Form bringt. Als Zusatzbedingung fordern wir, daß die Summe aller Meßwerte null sein muß:

$$\sum_{k=1}^n \varphi_k = 0. \quad (3.9)$$

Daraus ergibt sich:

$$\hat{D} \cdot \varphi = \begin{pmatrix} \varphi^\pi \\ 0 \end{pmatrix}, \quad (3.10)$$

wobei  $\hat{D}$  um  $(1 \times n)$  Einsen erweitert wurde:

$$\hat{D} = \begin{pmatrix} D \\ \mathbf{1}_{(1 \times n)} \end{pmatrix}. \quad (3.11)$$

$\hat{D}$  ist nichtsingulär und deshalb invertierbar. Dies ergibt sich, weil  $\pi$  ein repräsentativer Satz von Elektrodenpaaren ist. Die Meßwerte an den Elektroden lassen sich also bestimmen durch

$$\varphi = \hat{D}^{-1} \cdot \begin{pmatrix} \varphi^\pi \\ 0 \end{pmatrix}. \quad (3.12)$$

Für die Referenzmatrix  $N$  bedeutet das, daß sie nach der Invertierung durch eine Transformationsmatrix  $\tilde{E}$  auf die gewünschte Form  $N \in \mathbb{R}^{n \times (n-1)}$  gebracht wird.

$$N = \hat{D}^{-1} \tilde{E} = \begin{pmatrix} D \\ \mathbf{1}_{(1 \times n)} \end{pmatrix}^{-1} \cdot \begin{pmatrix} \mathbf{1}^{(n-1) \times (n-1)} \\ 0 \end{pmatrix}. \quad (3.13)$$

D.h. also, daß  $D$  zuerst um eine Zeile zu einer invertierbaren Matrix vergrößert wird und nach der Invertierung durch die Transformationsmatrix  $\tilde{E}$  wieder um die Zusatzzeile verkleinert wird.

Die virtuellen Meßwerte können also geschrieben werden als

$$m_v = \hat{L} \cdot d = N \cdot L(r) \cdot d = \hat{D}^{-1} \cdot \tilde{E} \cdot L(r) \cdot d. \quad (3.14)$$

Setzen wir diesen Ausdruck in 3.1 ein, so erhalten wir

$$R = \| m - \hat{D}^{-1} \cdot L(r) \cdot d \|_2 \quad (3.15)$$

und durch Umformen

$$R = \| \hat{D}^{-1}(\hat{D} \cdot m - \tilde{E} \cdot L(r) \cdot d) \|_2. \quad (3.16)$$

Die optimale Orientierung  $d$  ist durch die Lösung des gewichteten Least Squares Problems

$$d = \arg \min_{r \in \mathbb{R}^3} \| m - m_v(r) \|_2 \quad (3.17)$$

$$= \arg \min_{r \in \mathbb{R}^3} \| \hat{D}^{-1}(\hat{D} \cdot m - \tilde{E} \cdot L(r) \cdot d) \|_2 \quad (3.18)$$

gegeben. Die Orientierung und die Stärke des Dipols  $d$  soll also so gefunden werden, daß das Relative Residuum (3.16) möglichst klein wird. Glücklicherweise bietet die Lapack Bibliothek eine vorgefertigte Funktion an, die zur Lösung verwendet werden kann. Die Funktion heißt DGGGLM und dient zur Lösung von sogenannten allgemeinen Gauß-Markov linearen Modellproblemen mit der Form

$$\min_x \| y \|_2 \quad \text{mit } d = Ax + By. \quad (3.19)$$

Ist nun  $B$  invertierbar, so ist das Problem 3.19 äquivalent zu dem gewichteten linearen Least Squares Problem

$$\min_x \| B^{-1}(d - Ax) \|_2. \quad (3.20)$$

Übertragen auf unser Problem bedeutet dies, daß

$$B \equiv \hat{D}, \quad \dim B = n \times n \quad (3.21)$$

$$A \equiv \tilde{E} \cdot L(r), \quad \dim A = n \times 3 \quad (3.22)$$

$$d \equiv \hat{D} \cdot m, \quad \dim d = n \times 1. \quad (3.23)$$

Der Lapackfunktion werden also die Matrizen 3.21 bis 3.23 übergeben und man erhält als Ergebnis die optimale Orientierung  $d$  und die Relative Residuumsenergie, wobei die Länge von  $d$  die Stärke des Dipols angibt.

Das Inverse Problem soll also für einen gegebenen Ort auf folgende Weise gelöst werden: Zuerst müssen die Potentialfelder für einen repräsentativen Satz von Elektrodenpaaren berechnet werden (Vorbereitungsschritt). Die Leadfield Matrix läßt sich nun mit Hilfe des Reziprozitätstheorems leicht aufstellen. Der Lapackfunktion werden die Matrizen in geeigneter Form übergeben. Als Ergebnis erhält man die optimale Orientierung und die Stärke des Dipols für den Ort. Zusätzlich steht die Relative Residuumsenergie zur Verfügung, mit der abgeschätzt werden kann, wie gut ein Dipol an diesem Ort die wirklichen Meßwerte erfüllt. Verfahren zur Optimierung des Ortes werden im nächsten Kapitel beschrieben.

## 3.2 Deviation Scan und Dipollokalisierung

Aufbauend auf der im vorherigen Abschnitt dargestellten Theorie sollen nun Verfahren zur Lösung des inversen EEG-Problems beschrieben werden. Als Standardverfahren sollen der Deviation Scan und die Methode der Dipollokalisierung erklärt werden, wobei im Rahmen der Studienarbeit die Methode der Dipollokalisierung basierend auf dem Deviation Scan implementiert wurde.

Für beide Methoden muß das Rechengebiet (Kopf) in eine endliche Anzahl von Zelle diskretisiert werden, um die obige Matrixgleichung 3.18 anwenden zu können. Die Zellen sind derart definiert, daß die Potentialwerte an den Zellzentren berechnet worden sind. Das 3D-Modell des Kopfs wird durch Magnetresonanzaufnahmen (MRI-Scans) erstellt. Somit steht ein diskretisiertes Kopfmodell zur Verfügung, daß durch Segmentierung in fünf verschiedene Anteile (Compartments) aufgeteilt ist. Abbildung 3.1 zeigt einen axialen Schnitt durch einen Kopfdatensatz. Die Compartments sind von außen: Luft, Kopfhaut, Schädel, Gehirnmasse und ventrikuläres System. Für die Berechnung der Potentialfelder der Elektrodenpaare (Vorwärtsprobleme) werden die Leitfähigkeiten der einzelnen Compartments benötigt. Hierbei wird die Vereinfachung getroffen, daß jedes Compartment eine konstante und isotrope Leitfähigkeit besitzt. Bis zu diesem Punkt verlaufen beide Verfahren, Deviation

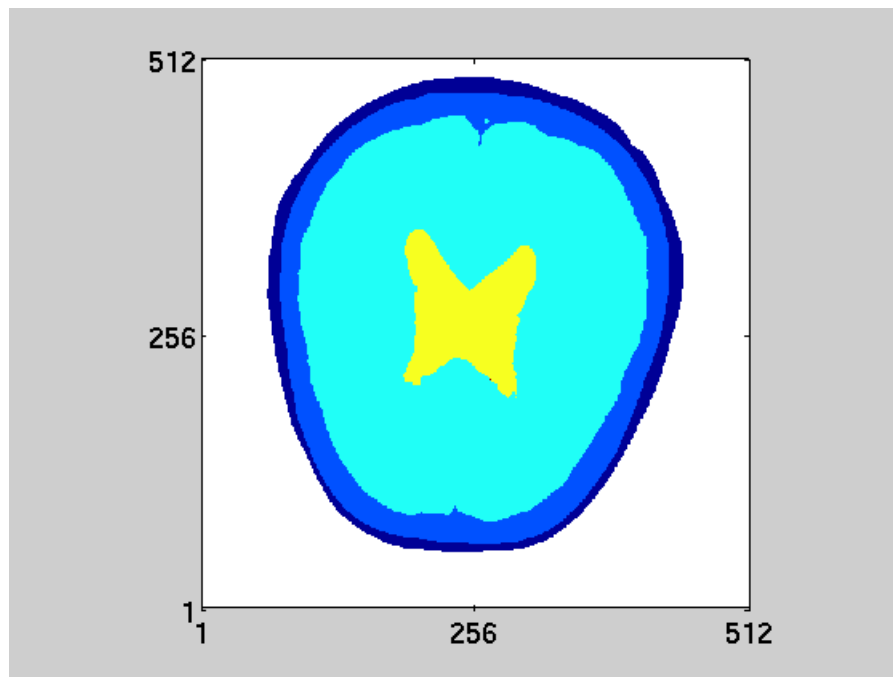


Abbildung 3.1: Compartments am Beispiel eines axialen Schnittes durch die Kopfdaten bestehend aus Luft, Kopfhaut, Schädel, Gehirnmasse und ventrikuläres System (von außen).

Scan und Dipollokalisierung, gleichermaßen.

Zuerst soll nun kurz der **Deviation Scan** beschrieben werden: Da sich die erkrankte Stelle des Epilepsiepatienten nur im Gehirn befinden kann, beschränkt sich die Suche auf diesen Anteil. Für jedes Gehirnvoxel wird beim Deviation Scan die optimale Orientierung des Dipols für das Zellzentrum berechnet. Dies erfolgt wie im vorherigen Abschnitt 3.1 erklärt: Im Vorbereitungsschritt wird für jedes unabhängige Elektrodenpaar das Potentialfeld berechnet, wobei ein Strom von der einen Elektrode (Quelle) einfließt und aus der anderen (Senke) herausfließt. Als Beispiel ist die Potentialverteilung eines Elektrodenpaars des gleichen Schnittes wie in Abbildung 3.1 dargestellt (Abb. 3.2). Eine genaue Beschreibung dieses Datensatzes erfolgt in Kapitel 3.3. Für die Koordinaten des jeweiligen Voxels wird die Leadfield Matrix aus den Potentialwerten der Nachbarzellen aufgestellt. Durch den Gebrauch der Laplace-Funktion wird das gewichtete Least Squares Problem gelöst und wir erhalten

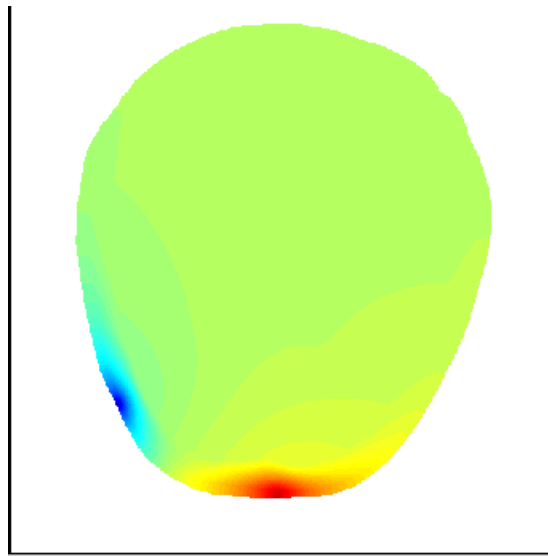


Abbildung 3.2: Potentialverteilung des ersten Elektrodenpaars (Vorbereitungsschritt).

die optimale Orientierung und die Relative Residuumsenergie für einen Dipol im Zellzentrum. Die Relative Residuumsenergie ist ein Maß dafür, wie gut der optimale Dipol an diesem Ort die wirklichen Meßwerte annähert. Ist für jedes Gehirnvoxel das Least Squares Problem gelöst, so kann man die Position die Dipols durch Suchen des kleinsten relativen Residuums lokalisieren. Es muß also für eine sehr große Anzahl von Zellen das Least Squares Problem gelöst werden und die Lokalisation des Dipols beschränkt sich auf die diskreten Punkte der Zellzentren. Ein Vorteil des Deviation Scans ist, daß mehrere Dipole gefunden werden können, falls nicht nur eine einzelne sehr dominante Energiequelle vorliegt. Außerdem ist es möglich, aus den Voxeldaten des Deviation Scans Konturfelder zu erstellen, die eine Vorstellung des Funktionals geben. Beispiele von Konturfeldern werden im nächsten Abschnitt gezeigt.

Im Rahmen der Studienarbeit ist aufbauend auf dem Deviation Scan ein weiteres Standardverfahren, die Methode der **Dipollokalisation**, implementiert worden. Im Gegensatz zum Deviation Scan werden nicht alle Gehirnvoxel untersucht, sondern man nähert sich mit Hilfe eines Suchalgorithmus dem Dipol. Wie schon oben erklärt,

sind die freien Parameter des inversen Problems die Position und Orientierung des Dipols. Die Position des Dipols soll mit dem Nelder-Mead Algorithmus und die Orientierung durch Lösen des Least Squares Problems optimiert werden. Die Suche sieht nun so aus, daß wir mit einem Startsimplex beginnen und der Nelder-Mead Algorithmus sich durch Veränderung seines Simplex dem Minimum iterativ nähern wird. Um nun entscheiden zu können, ob ein neu berechneter Testeckpunkt eine Verbesserung des Simplex darstellt, wird jedesmal das relative Residuum berechnet. Das relative Residuum stellt also den zu minimierenden Funktionswert dar. Daraus folgt, daß die Leadfield Matrix für diesen Punkt aufgestellt und das gewichtete Least Squares Problem gelöst werden muß. Das Simplex wird sich nun so lange um das Minimum zusammenziehen, bis ein Abbruchkriterium die Dipolsuche terminiert.

Es müssen folgende Punkte beachtet werden: Zur Aufstellung der Leadfield Matrix werden Potentialwerte aus den Vorwärtsproblemen benötigt. Da das Nelder-Mead Verfahren seine Simplexpunkte auf einem kontinuierlichem Gebiet berechnet, müssen die Potentialwerte trilinear interpoliert werden.

Wie bei dem Deviation Scan sollte sich die Suche auf die Gehirnzellen beschränken, da sich nur dort der Dipol befinden kann. Um dies zu gewährleisten, steht ein "Strafterm" zur Verfügung. Dieser wird aktiv, wenn der Funktionswert an einer Stelle berechnet werden soll, die außerhalb des Gehirns liegt. Im Code ist dies so umgesetzt worden, daß die Routine zur Funktionswertberechnung einen sehr großen Wert zurück gibt, so daß der Algorithmus zu einem anderen Nelder-Mead Teilschritt gezwungen wird und innerhalb des Gebiets bleibt. Tests mit dem in Abbildung 3.1 gezeigten Kopfschnitt haben jedoch gezeigt, daß der Nelder-Mead Algorithmus an dem ventrikulären System "hängen" bleiben kann, wenn sich der Dipol auf der anderen Seite befindet. Um dies zu umgehen, wird der Dipol in den Compartments des Gehirns und des ventrikulären Systems gesucht. Um einen Eindruck vom Verlauf des Funktionals auf diesen beiden Compartments zu bekommen, sind Konturlinien im nächsten Kapitel dargestellt. Testrechnungen haben gezeigt, daß das Rechengebiet auf Gehirn und ventrikuläres System erweitert werden kann, so daß das Minimum von jedem Startsimplex gefunden werden kann.

In Kapitel 2.2 sind die Strukturen des implementierten Nelder-Mead Algorithmus erklärt worden. Die Struktur *problem\_data* ist nicht näher beschrieben worden, da bei



einfachen Funktionen nur die Koordinaten zur Berechnung des Funktionals verwendet werden. Im Fall der Dipollokalisierung ist dies jedoch anders, da die Berechnung des relativen Residuums viele Informationen wie die Anzahl der Elektroden, die Potentialfelder, der Compartmenttyp jeder Zelle, die Leadfield Matrix usw. benötigt. Diese Informationen werden der Routine zur Funktionswertberechnung durch die Struktur *problem\_data* übergeben.

In nächsten Kapitel wird die Methode der Dipollokalisierung auf 2D- und 3D-Testfälle angewandt, bei denen die “wahren” Meßwerte an den Elektroden durch numerische Simulationen erstellt wurden. Das Verhalten der Dipollokalisierung bei wirklichen, fehlerbehafteten Meßwerten wird in Kapitel 3.4 und 3.5 vorgestellt und analysiert.

### 3.3 Anwendung auf 2D Testfälle

Nachdem im letzten Kapitel die Theorie der Dipollokalisierung beschrieben worden ist, soll nun die Anwendung auf 2D und 3D Testdaten vorgestellt werden. Später werden die Ergebnisse der Dipolsuche mit wirklichen Meßwerten dargestellt. Der große Unterschied zwischen den hier verwendeten Testmeßwerten, die durch eine Simulation erstellt worden sind, und den wirklichen Meßwerten ist, daß den Testmeßwerten eine einzelne Quelle zugrunde liegt und sie bis auf Rechenfehler usw. völlig störungsfrei sind. Bei einer realen EEG-Messung werden immer Meßstörungen vorhanden sein.

Zuerst soll die Dipolsuche für den zweidimensionalen Fall beschrieben werden. Um ein möglichst gutes Verständnis der Dipollokalisierung zu erreichen, soll zusätzlich die Erstellung der Testmeßdaten erklärt werden. Wir beginnen mit dem axialen Schnitt aus einem Kopfdatensatz, der schon in Abbildung 3.1 gezeigt worden ist. Die Auflösung des Querschnitts beträgt  $512 \times 512$ . Jedes Voxel des Datensatzes besitzt einen Marker, der die Zugehörigkeit des Voxels zu einem Compartmenttyp markiert. Die Marker sind (0) = Luft, (1) = Kopfhaut, (2) = Knochen, (3) = Gehirnmasse und (4) = ventrikuläres System. Es werden sieben Elektroden um den Kopfquerschnitt definiert. Die Koordinaten der Elektroden sind:

Elektrode	x	y
1	250	54
2	100	140
3	67	340
4	167	476
5	336	477
6	449	326
7	395	145

Es wird nun von den sieben Elektroden ein repräsentativer Satz von Paaren benötigt. Von den verschiedenen Möglichkeiten sind die folgenden sechs Paare verwendet worden:

Quelle	Senke
1	2
1	3
1	4
1	5
1	6
1	7

Für jedes dieser Elektrodenpaare muß ein Vorwärtsproblem gelöst werden. Dafür wird eine Elektrode als Quelle und die andere als Senke definiert, und durch numerisches Lösen dieses elliptische Randwertproblems werden die Potentialfelder im Kopf berechnet. Das Potentialfeld für das erste Paar ist in Abbildung 3.2 dargestellt.

Um die Funktionsfähigkeit des Programms zur Dipollokalisierung zu kontrollieren, wird ein Dipol an einer selbst definierten Stelle positioniert und mit vorgegebener Orientierung und Stärke werden Meßwerte an den Elektroden berechnet, die in Wirklichkeit die Ergebnisse der EEG-Messung sind. Basierend auf diesen Meßwerten muß das Dipollokalisationsprogramm die Position des Dipols wiederfinden. Für das 2D Problem sind zwei Dipole "versteckt" worden bei (380, 400) (Dipol 1) und (194, 249) (Dipol 2). Um die Potentiale an den Elektroden zu berechnen, wird folgendermaßen vorgegangen: Zuerst muß für jede Dipolposition die Leadfield Matrix  $L \in \mathbb{R}^{7 \times 2}$  aufgestellt werden. Um die Testmeßwerte möglichst unabhängig von der späteren

Dipolsuche zu erstellen, werden die Potentialwerte an den Elektroden nicht mit Hilfe des Reziprozitätstheorems berechnet, sondern jeder Eintrag der Leadfiel Matrix wird durch eine eigene Simulation erhalten. Der Dipol ist hierbei modelliert durch eine Klammer der Länge  $2h$ , wobei  $h$  die Maschenweite in die jeweilige Richtung ist. Jede Klammer (Dipol) besteht also aus einem Quelle-Senke Paar in die jeweilige Richtung.

		Dipol 1	Dipol 2
in $x$ -Richtung	Quelle	(379, 400)	(194, 250)
	Senke	(381, 400)	(196, 250)
in $y$ -Richtung	Quelle	(380, 399)	(195, 249)
	Senke	(380, 401)	(195, 251)

Die Testmeßwerte an den Elektroden (Spannungen  $m_t$ ) lassen sich analog zu Gleichung 3.2 berechnen mit der Formel

$$m_t = L(r) \cdot d(\alpha, m), \quad (3.24)$$

wobei  $r$  die Position des Dipols und  $d$  die Orientierungsvektor bestehend aus dem Winkel  $\alpha$  und der Stärke  $m$  sind.  $d$  ist definiert durch

$$d(\alpha, m) = m \cdot \begin{pmatrix} \cos(\alpha) \\ \sin(\alpha) \end{pmatrix}. \quad (3.25)$$

Für die beiden Dipole sind folgende Parameter festgelegt

Dipol	$\alpha$	$m$
1	45°	2.0
2	195°	3.5

Wir erhalten als Testmeßwerte an den Elektroden

Spannungen an den Elektroden		
Elektrode	Dipol 1	Dipol 2
1	2.517395124593209e-03	-2.630231455671983e-05
2	2.470847222773952e-03	-1.072641501190064e-02
3	2.155109907879404e-03	-8.818207652066146e-03
4	6.130555417841188e-04	1.531145236788156e-03
5	-7.809859245434758e-03	5.433486934821624e-03
6	-2.245727763185625e-03	6.992195294972303e-03
7	2.299179211589708e-03	5.614097511941502e-03

Somit stehen alle nötigen Daten zur Verfügung, um ausgehend von den Meßwerten die Dipole mit Hilfe der Dipollokalisation zu suchen. Da sich der Dipol nur in dem Gehirncompartment befinden kann, haben wir die Suche am Anfang nur auf dieses Gebiet beschränkt. Besonders bei der Lokalisation des Dipols 2 haben sich dadurch Probleme ergeben. Dieser Dipol liegt direkt links neben dem ventrikulären System und ist somit bei einer Suche, die direkt auf der anderen Seite der Ventrikel startet, nicht erreichbar. Der Nelder-Mead Algorithmus bleibt an dem ventrikulären System hängen. Um diese Problem besser verstehen zu können, haben wir für jeden Dipol einen Deviation Scan durchgeführt. Stellen wir nun in jedem Voxel den errechneten Funktionswert, die Relative Residuumsenergie, farblich dar, so erhalten wir Konturlinien der Funktionals. Die Verteilungen der Relativen Residuumsenergie sind nachfolgend dargestellt für den Dipol 1 (gesamter Kopf und Gehirn, Abb. 3.1) und Dipol 2 (wieder gesamter Kopf und Gehirn, Abb. 3.2) jeweils mit und ohne Konturlinien. Untersucht man die Bilder des gesamten Kopfs genauer, erkennt man, daß das Funktional am Rand falsche Werte annimmt. Der Grund liegt in der Aufstellung der Leadfield Matrix, da dort die Spannung mit Hilfe der Nachbarpunkte errechnet wird und diese außerhalb des Kopfs fehlen. Ein großer Einfluß der Ventrikel auf die Potentialverteilung kann erwartet werden, da die Leitfähigkeit der Ventrikel achtmal höher ist als die des Gehirns. Anhand der Verteilungen des Funktionals kann man sehen, daß die Konturlinien stetig über das Gebiet der Ventrikel verlaufen und das Funktional dort keinen auffälligen Verlauf hat. Somit sollte die Dipollokalisation mit dem Nelder-Mead Algorithmus auf dem Gebieten des Gehirns erweitert durch das

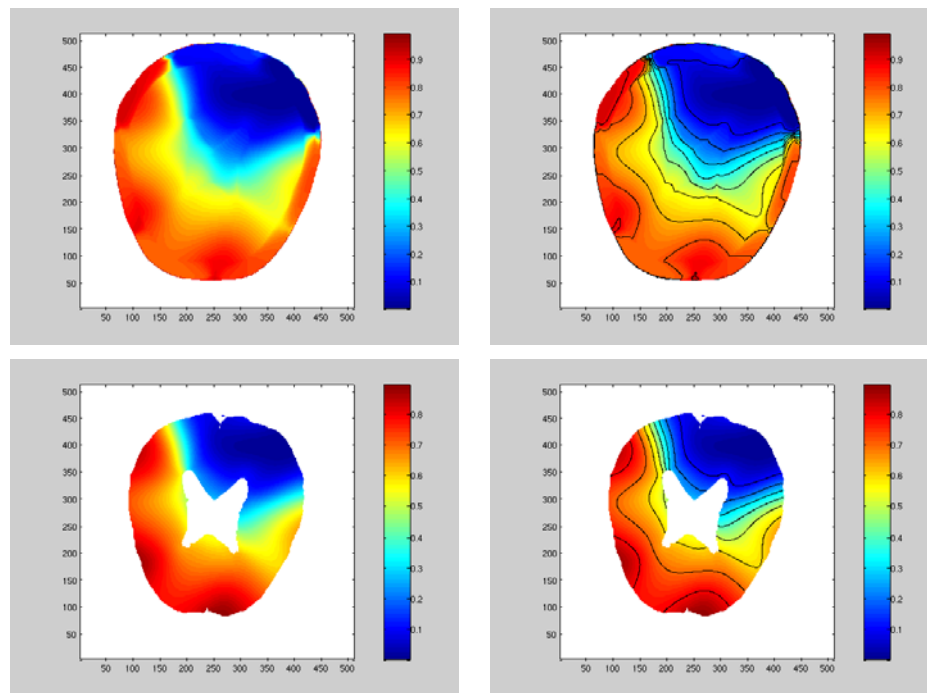


Tabelle 3.1: Verteilung der Relativen Residuumsenergie des gesamten Kopfs (oben) und des Gehirns (unten) für die 1. Meßwerte.

ventrikuläre System problemlos funktionieren und folglich jeder Dipol im Gehirn von jedem Startsimplex aus gefunden werden. Tests mit verschiedenen Startdreiecken haben gezeigt, daß sich der Suchalgorithmus fehlerfrei durch das ventrikuläre System bewegt, falls dies nötig ist.

Für jeden der beiden Dipole sind Dipollokalisierungen von vier verschiedenen Simplexes gestartet worden, die auf dem Querschnitt des Kopfs, wie in Abbildung 3.3 dargestellt, positioniert worden sind. Als Kantenlänge des 2D Querschnitts ergibt sich 20 cm, da es sich bei den  $512 \times 512$  Voxeln um quadratische Zellen der Kantenlänge 0.0390625 cm handelt. Wie oben beschrieben sind zwei Dipole im Gehirn-Compartment definiert worden (Dipol 1 bei (380, 400) und Dipol 2 bei (195, 250)). Da diese Testdaten in Matlab erstellt worden sind und da dort, im Gegensatz zur Kopfdiskretisierung, von eins ab gezählt wird, suchen wir die beiden folgenden Dipole: Dipol 1 bei (379, 399) und Dipol 2 bei (194, 249). Als Ergebnis der Dipollokalisierung

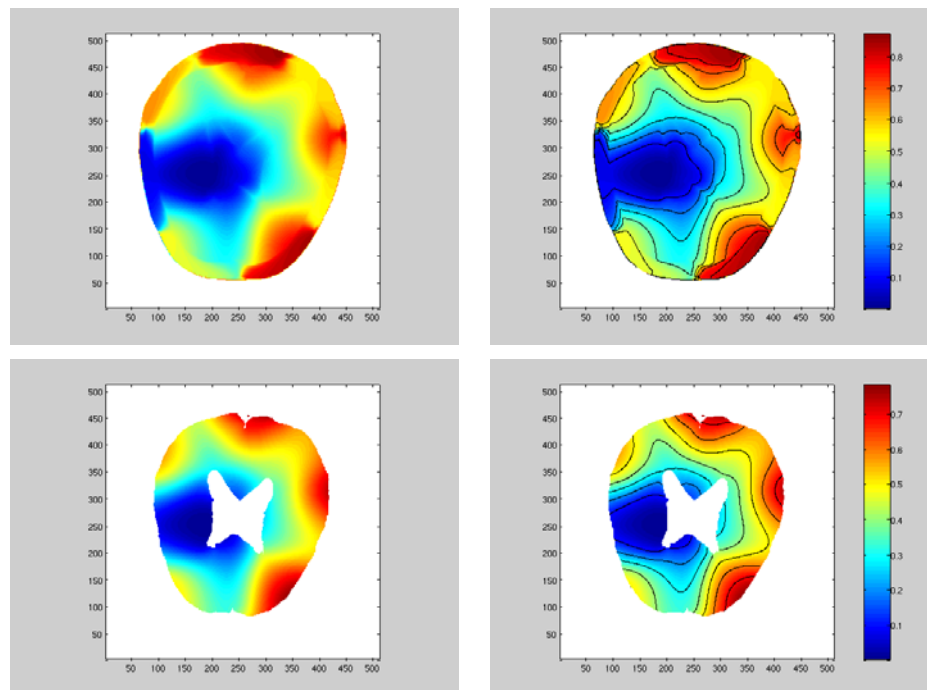


Tabelle 3.2: Verteilung der Relativen Residuumsenergie des gesamten Kopfs (oben) und des Gehirns (unten) für die 2. Meßwerte.

tionen sind in Abbildung 3.4 und 3.5 zu jedem Dipol der Verlauf des Funktionswerts (Relative Residuumsenergie), der Abstand des kleinsten Funktionswerts zum Ziel (Dipol), der maximale Abstand von einem Eckpunkt des Simplex zum Simplexschwerpunkt und die Bewegung des Simplexschwerpunkts zwischen zwei aufeinanderfolgenden Iterationen für jedes der vier Startdreiecke pro Iteration dargestellt. Außer dem “Abstand zum Ziel” entsprechen die anderen Abbildungen dem Verlauf der drei möglichen Abbruchkriterien. Als Abbruchkriterium für die Testfälle ist gefordert worden, daß der kleinste Funktionswert (Relative Residuumsenergie) des aktuellen Simplex kleiner als  $10^{-10}$  sein muß, was man am Verlauf des Funktionswerts in Abbildung 3.4 gut sehen kann. Als maximale Anzahl von Iterationen ist 100 gewählt worden. Der Deviation Scan findet als Minimum der beiden Funktionale die richtigen Voxel ((397, 399) und (194,249) für Dipol 1 und 2) und gibt an diesen beiden Positionen die Funktionswerte  $3.96e-10$  und  $1.48e-9$  an.

Vergleicht man die Ergebnisse der Dipollokalisierungen von Dipol 1 und 2 (Abb.

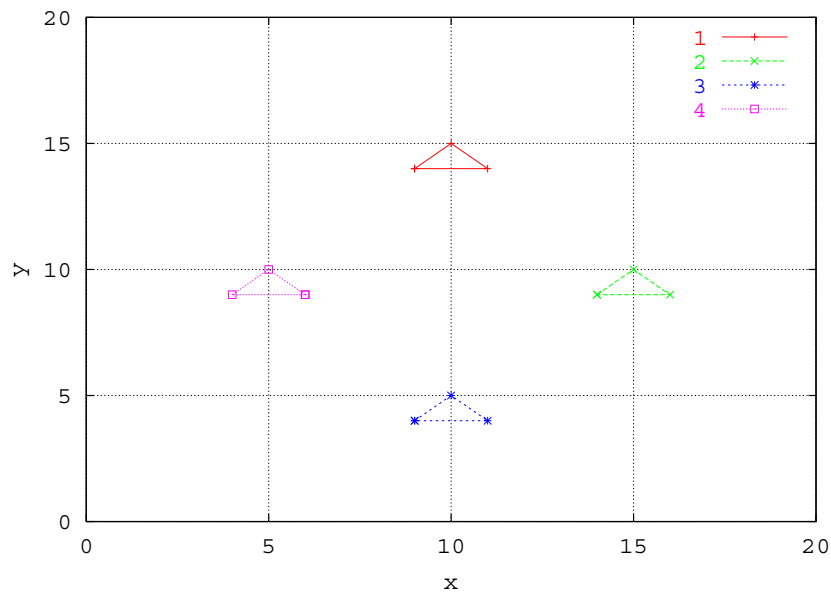


Abbildung 3.3: Anordnung der Startsimplices für die Dipollokalisierung im 2D Kopfquerschnitt.

3.4 und 3.5), so kann man erkennen, daß Dipol 1 mit größerer Genauigkeit gefunden wird als Dipol 2. Schon nach ungefähr 40 Iterationen wird bei den vier Suchen von Dipol 2 keine Verbesserung mehr von Funktionswerten und Abständen zum Dipol erzielt. Der Verläufe des maximalen Abstands zum Schwerpunkt und die Veränderung des Schwerpunkts weisen darauf hin, daß sich die Simplices stark um einen Punkt zusammenziehen, dieser jedoch nicht die exakte Position des Dipols ist. Folglich verringern sich Funktionswerte und Abstände zum Dipol nicht weiter. Die Dipollokalisierungen werden nach 100 Iterationen abgebrochen, da die Funktionswerte die Toleranz nicht unterschreiten. Im Gegensatz dazu sieht man in Abbildung 3.4, daß die Suchen nach Dipol 1 nach 30-50 Iterationen beendet werden, da das Toleranzkriterium erfüllt wird. Bis an diese Stelle ziehen sich die Dreiecke um den richtigen Punkt zusammen, da der Abstand zum Dipol noch kleiner zu werden scheint. Wie man in allen acht Dipollokalisierungen erkennen kann, nähert sich das Simplex mindestens bis auf 1/100 mm der Position des Dipols an. Vergleicht man die Ergebnisse mit dem jeweiligen Deviation Scan, erkennt man, daß das Verfahren der

Dipollokalisierung den Dipol mit größerer Genauigkeit findet. Die erreichten Werte der Relativen Residuumsenergie von der Größenordnung  $1e-10$  scheinen eine Grenze für die beiden Verfahren zu sein, da außer bei Dipol 1 keine größere Genauigkeit erzielt werden konnte. Vielleicht zeigt sich hier der Einfluß von Rechenfehlern auf die mögliche Genauigkeit der Verfahren.

Das Verfahren der Dipollokalisierung wurde auch an einem künstlich erstellten 3D Kopfmodell getestet. Der Kopf hat hier eine quadratische Form und besteht aus  $21 \times 21 \times 21$  Voxeln. Der Dipol hat die Position  $(10,10,10)$  und die von ihm verursachten Potentiale werden von 24 Elektroden beschrieben, die um den Quadratschädel definiert sind. Auch in diesem Testfall konnte der Nelder-Mead Algorithmus den Dipol genau lokalisieren.



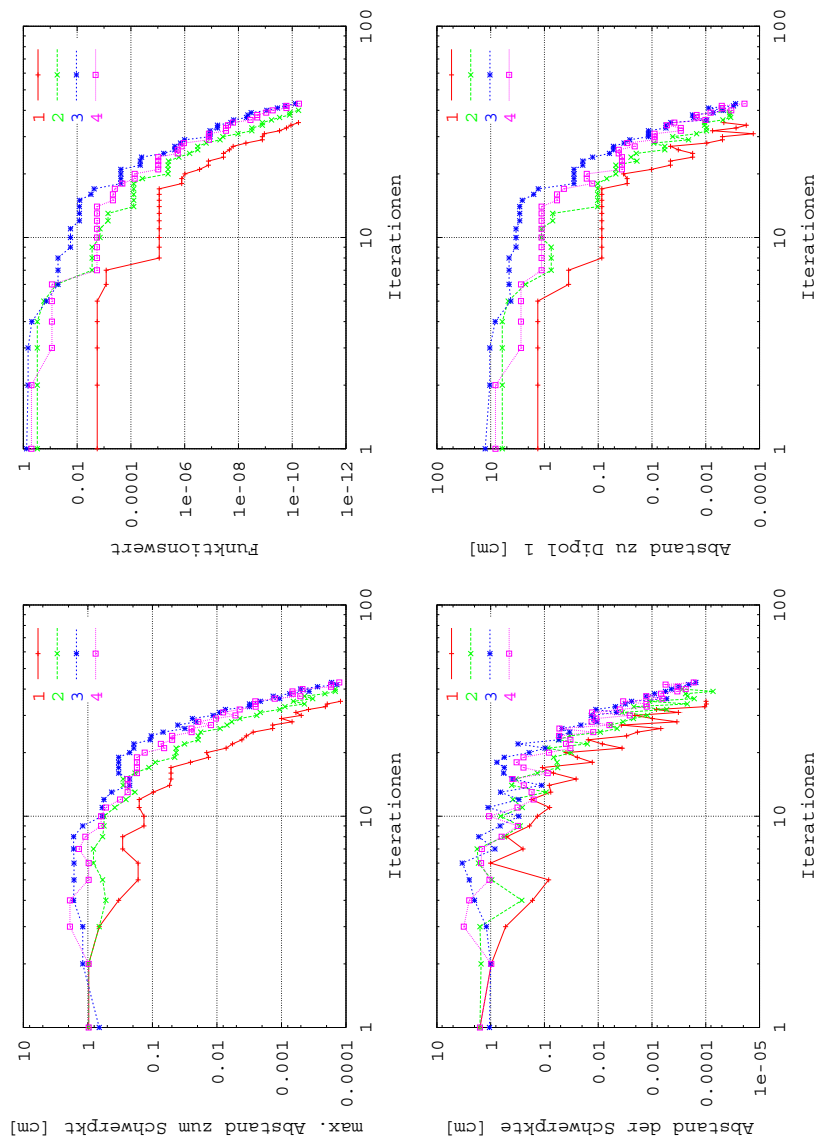


Abbildung 3.4: Ergebnisse der Dipollokalisierung von **Dipol 1**. Dargestellt sind der Verlauf des Funktionswerts, der Abstand des Punkts des kleinsten Funktionswerts zum Ziel, der max. Abstand von einem Eckpunkt zum Simplexschwerpunkt und die Bewegung des Schwerpunkts zwischen zwei Iterationen für jedes der vier Startdreiecke. Als Abbruchkriterium ist gefordert worden, daß der kleinste Funktionswert  $< 10^{-10}$ . Die maximale Anzahl von Iterationen ist 100.

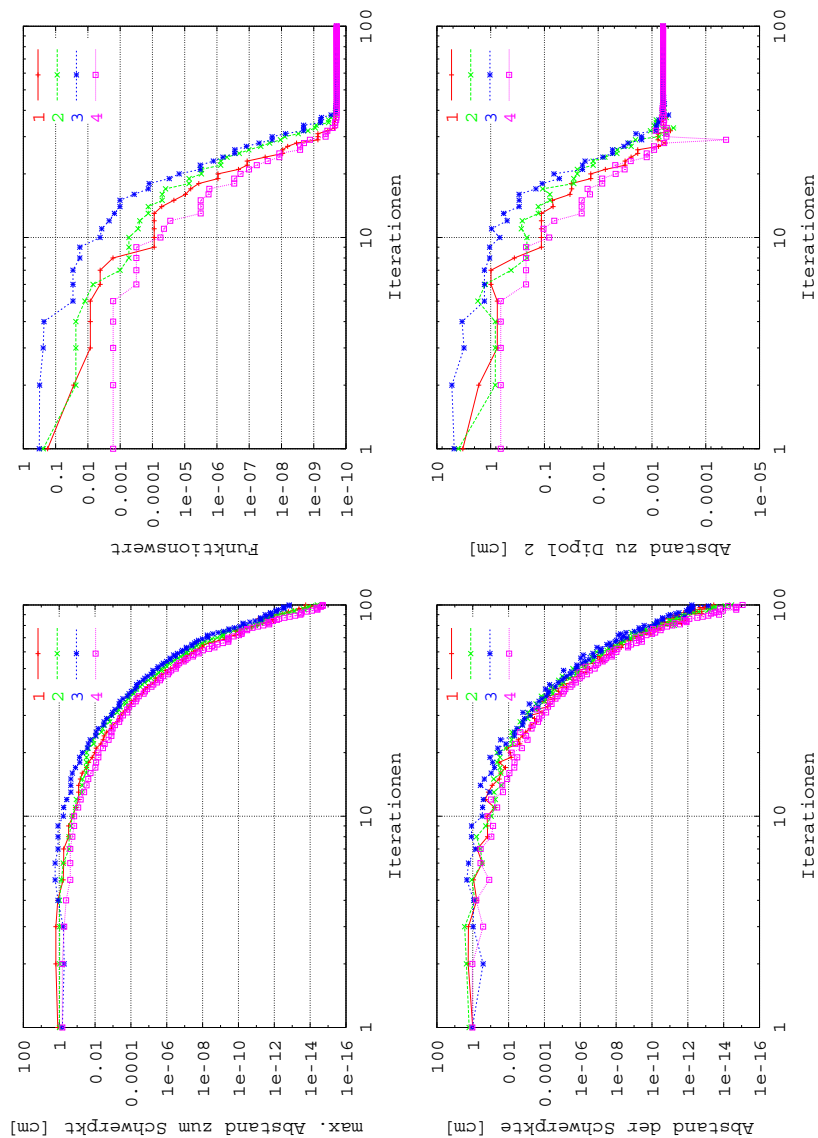


Abbildung 3.5: Ergebnisse der Dipollokalisierung von **Dipol 2**. Dargestellt sind der Verlauf des Funktionswerts, der Abstand des Punkts des kleinsten Funktionswerts zum Ziel, der max. Abstand von einem Eckpunkt zum Simplexschwerpunkt und die Bewegung des Schwerpunkts zwischen zwei Iterationen für jedes der vier Startdreiecke. Als Abbruchkriterium ist gefordert worden, daß der kleinste Funktionswert  $< 10^{-10}$ . Die maximale Anzahl von Iterationen ist 100.

### 3.4 Verhalten bei wirklichen Meßwerten

Im folgenden soll das Programm zur Dipollokalisierung auf wirkliche EEG-Meßwerte angewandt werden. Das 3D Modell des Kopfs ist durch eine Magnetresonanzaufnahme erstellt worden und besteht aus  $129 \times 129 \times 129$  Voxeln. Die verschiedenen Compartments dieses Datensatzes sind wieder mit unterschiedlichen Markern gekennzeichnet. Die Kantenlänge eines Voxels ist 0.00196 m, was eine Kantenlänge des gesamten Kopfmodells von ca. 25.3 cm ergibt. Bei der EEG-Messung sind 27 Elektroden am Kopf des Patienten befestigt worden. Aus der Langzeit-EEG-Messung wird eine Epoche herausgenommen, die charakteristisch ist für einen kleinen epileptischen Anfall. Diese Epoche besteht nun aus 180 Zeitsamples, von denen eines zur Dipollokalisierung verwendet wird.

Als Beispiel steht im Folgenden der Kopfdatensatz des Patienten pat790II aus der Uniklinik in Gent zur Verfügung. Dieser Datensatz ist in vier Compartments eingeteilt: Luft, Haut, Knochen und Gehirn. Es gibt also kein ventrikuläres System wie im Fall des 2D Datensatzes. Der Ausgangspunkt der Dipollokalisierung sind 27 Spannungen eines Zeitsamples der EEG-Messung (Tabelle 3.3). Das Ergebnis des Deviation Scans für diese Meßwerte ist eine Relative Residuumsenergie von 0.067 im Zentrum der Zelle (39, 54, 62). Ausgehend von den folgenden fünf Simplices ist der Dipol gesucht worden:

Startsimplex	1	2	3	4	5
$x_1$	0.08	0.09	0.07	0.06	0.07
$y_1$	0.13	0.12	0.10	0.07	0.08
$z_1$	0.17	0.13	0.14	0.14	0.13
$x_2$	0.12	0.08	0.11	0.12	0.10
$y_2$	0.12	0.09	0.14	0.14	0.11
$z_2$	0.13	0.16	0.15	0.13	0.15
$x_3$	0.13	0.11	0.12	0.07	0.11
$y_3$	0.13	0.09	0.09	0.06	0.07
$z_3$	0.17	0.17	0.17	0.17	0.17
$x_4$	0.14	0.15	0.13	0.14	0.14
$y_4$	0.10	0.11	0.13	0.08	0.14
$z_4$	0.16	0.11	0.17	0.16	0.16

Elektrode	Spannung
1	2.3120999999999999e+01
2	3.3948999999999998e+00
3	3.2878999999999998e+01
4	4.6330999999999998e+00
5	4.9286999999999999e+01
6	3.7518999999999998e+01
7	1.2491000000000000e+01
8	1.9027999999999999e+01
9	-1.8559999999999999e+01
10	-2.7559999999999999e+01
11	-1.9920000000000002e+01
12	-3.6590000000000003e+01
13	-2.1170000000000002e+01
14	-1.6870000000000001e+01
15	-2.4760000000000002e+01
16	-1.8120000000000001e+01
17	-2.1219999999999999e+01
18	-3.3159999999999997e+01
19	-2.0420000000000002e+01
20	4.4420000000000002e+01
21	-1.6590000000000000e+01
22	-4.6289999999999996e+00
23	3.9843000000000004e+01
24	-5.7359999999999998e+00
25	4.3902999999999999e+01
26	-1.8789999999999999e+01
27	3.9748999999999999e+00

Tabelle 3.3: EEG-Meßwerte an den 27 Elektroden.

Als Abbruchkriterium sind entweder 200 Iterationen oder ein Minimalwert des relativen Residuums von 0.05 festgelegt worden. Dieser Wert ist so gewählt, daß die Dipolsuche so genau wie möglich verläuft. Die Ergebnisse sind in Abbildung 3.6 dargestellt. Aufgetragen sind wieder der Funktionswert (Relative Residuumsenergie), der maximale Abstand von einem Eckpunkt zum Simplexschwerpunkt und die Veränderung des Simplexschwerpunkts pro Iteration für alle fünf Startdreiecke. Zusätzlich werden in Tabelle 3.4 die Ergebnisse der fünf Dipollokalisierungen mit denen des Deviation Scans verglichen. In Tabelle 3.4 bedeutet RRE die Relative Residuumsenergie,  $x$ ,  $y$  und  $z$  die Koordinaten des kleinsten Funktionswerts,  $i$ ,  $j$

und  $k$  die Zellkoordinaten des kleinsten Funktionswerts und o. D. M. das optimale Dipolmoment. Zuletzt sind die benötigten Funktionsauswertungen verglichen.

Wie man in dem mittleren und unteren Bild von Abbildung 3.6 sehen kann, ziehen sich die Simplices der fünf Dipolsuchen alle um einen Punkt stark zusammen. Der Verlauf des Funktionswerts und Tabelle 3.4 zeigen, daß alle fünf Dipolsuchen den gleichen Punkt als Dipolposition lokalisieren. Allerdings erkennt man auch, daß die Relative Residuumsenergie nicht kleiner als 0.06495 wird, somit also der Dipol nicht mit sehr großer Genauigkeit gefunden werden konnte. Verglichen mit den 2D Dipollokalisationen mit Testmeßwerten (Kapitel 3.3) sind die Ergebnisse sehr viel schlechter. Der Grund dafür liegt darin, daß den EEG-Meßwerten Hintergrundaktivitäten anderer Gehirnteile und Fehler durch das Meßverfahren selbst überlagert sind. Im Gegensatz dazu sind die 2D Testmeßwerte durch einen einzelnen, ungestörten Dipol erstellt worden.

Vergleicht man die Ergebnisse der Dipollokalisationen mit denen des Deviation Scans, erkennt man keinen sehr großen Genauigkeitsunterschied zwischen den beiden Verfahren. Das relative Residuum im Fall der Dipollokalisation ist etwas geringer als beim Deviation Scan. Der Dipol wird in der gleichen Zelle wie beim Deviation Scan gefunden, nur nicht direkt im Zentrum. Es scheint, daß das Verfahren zur Dipollokalisation die kleinste Relative Residuumsenergie exakt gefunden hat, diese aber durch Störungen verschoben ist. Anscheinend kann das Modell die wahre Dipolposition mit gestörte Meßdaten nicht exakt bestimmen. Bemerkenswert ist allerdings die Unterschiede in der Anzahl der Funktionsauswertungen. Für die gewählten Startsimplices benötigt das Verfahren der Dipollokalisation ca. 0.3% der Funktionsauswertungen des Deviation Scans, die sich aus der Anzahl der Gehirnvoxel ergeben. Bei gleicher Genauigkeit der beiden Methoden wäre der Unterschied noch gravierender.

Bei der Anwendung der Dipollokalisation auf einen zweiten Satz von EEG-Meßwerten zeigte sich der Einfluß von Meßstörungen noch deutlicher. Der Deviation Scan ergibt als kleinstes relatives Residuum einen Wert von 0.527 im Zentrum der Zelle (56, 71, 54). Für die Dipollokalisation sind wieder die gleichen fünf Startsimplices und Abbruchkriterien wie im vorherigen Beispiel benutzt worden. In Abbildung 3.7 sieht man deutlich, daß der Nelder-Mead Algorithmus bei diesem Satz von Meßwerten nur in zwei Fällen ein Relatives Residuum kleiner als im Deviation Scan

findet. Wie man in Tabelle 3.5 erkennen kann, unterscheiden sich die errechneten Orte aller fünf Dipole. Das Verfahren zur Dipollokalisierung erzielt also für diese Meßwerte kein zuverlässiges Ergebnis. Der Grund hierfür ist die Größenordnung des Relativen Residuums. Der Minimalwert aus dem Deviation Scan und den Dipollokalisierungen ist 0.507. Nimmt man jetzt an, daß die Parameter dieses Dipols optimal sind, so heißt das, daß nur 50% der Meßwerte den exakten Dipol beschreiben. Die Dipolsuchen werden also durch Meßstörungen von anderen Quellen “angezogen”, so daß sich die Simplices der verschiedenen Suchen um verschiedene lokale Minima zusammenziehen. Schlußfolgernd heißt das, daß man aus den zur Verfügung stehenden Meßwertsätzen einer Epoche diejenigen heraussuchen muß, bei denen die kleinste Residuumsenergie möglichst klein ist. Die Größenordnung dieses Werts steht natürlich nur nach der Durchführung von Dipollokalisierungen oder einem Deviation Scan zur Verfügung. Außerdem ist klar geworden, daß zur Dipolsuche mit dem Verfahren der Dipollokalisierung immer mit mehreren unterschiedlichen Simplices gestartet werden muß, falls wirkliche EEG-Meßwerte verwendet werden. Lokalisieren alle Suchen den Dipol an der gleichen Stelle, wie im ersten Beispiel, so kann davon ausgegangen werden, daß das Minimum wirklich dort ist.

Im nächsten Kapitel soll auf diese Problematik näher eingegangen werden. Dort werden auf dem 2D-Kopfquerschnitt (Kapitel 3.3) Meßdaten getestet, die gestört sind oder bei denen zwei verschieden starke Quellen zur Meßwernerstellung definiert wurden.

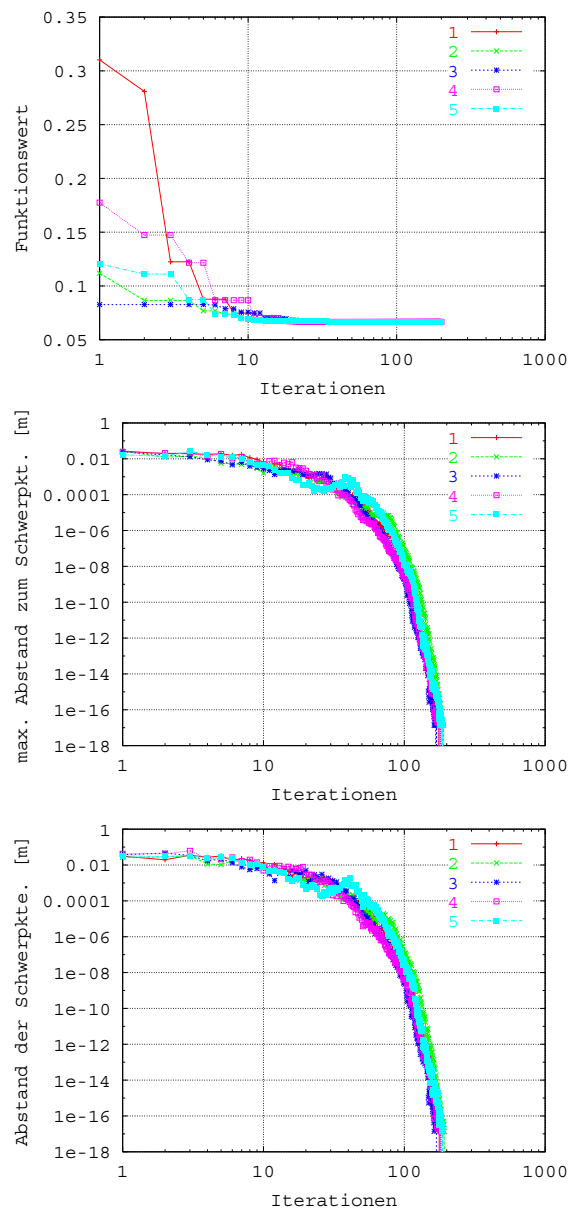


Abbildung 3.6: Ergebnisse der Dipollokalisierung mit wirklichen Meßwerten. Dargestellt sind der Verlauf des Funktionswerts, der max. Abstand von einem Eckpunkt zum Simplexschwerpunkt und die Bewegung des Schwerpunkts zwischen zwei Iterationen für jedes der fünf Startdreiecke. Als Abbruchkriterium ist gefordert worden, daß der kleinste Funktionswert  $< 0.05$ . Die maximale Anzahl von Iterationen ist 200.

Startsimpl.	RRE	$x$ [m]	$y$ [m]	$z$ [m]	i	j	k	o. D. M.	Fkt.ausw.
1	6.649573e-02	0.077103	0.106167	0.122147	39	54	62	26.397036	504
2	6.649573e-02	0.077103	0.106167	0.122147	39	54	62	26.397036	450
3	6.649552e-02	0.077028	0.106095	0.122849	39	54	62	26.321490	522
4	6.649573e-02	0.077103	0.106167	0.122147	39	54	62	26.397036	471
5	6.649552e-02	0.077028	0.106095	0.122849	39	54	62	26.321490	454
Dev. Scan	6.7e-02	0.07742	0.10682	0.1225	39	54	62	26.36644	157898

Tabelle 3.4: Ergebnisse der Dipollokalisierung von fünf verschiedenen Startsimplices. RRE ist die Relative Residuumsenergie und o. D. M. ist das optimale Dipolmoment. Zum Vergleich sind die Ergebnisse des Deviation Scans hinzugefügt.



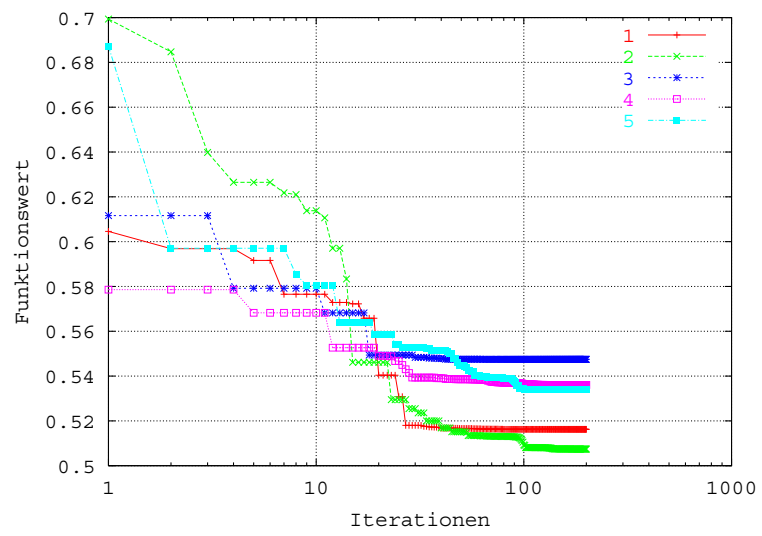


Abbildung 3.7: Verläufe der Relativen Residuumsenergie des zweiten Datensatzes.

Startsimpl.	RRE	$x$ [m]	$y$ [m]	$z$ [m]	$i$	$j$	$k$	o. D. M.	Fkt.ausw.
1	5.162269e-01	0.101920	0.145040	0.117600	52	73	59	6.198085	226
2	5.074399e-01	0.109760	0.143077	0.111720	56	72	56	6.252005	242
3	5.474956e-01	0.077891	0.164702	0.121520	39	84	61	8.571300	250
4	5.360543e-01	0.110702	0.148960	0.121520	56	75	62	4.552417	264
5	5.339134e-01	0.103861	0.154094	0.123480	52	78	63	5.612834	283
Dev. Scan	0.527	0.11074	0.14014	0.10682	56	71	54	9.010506	157898

Tabelle 3.5: Ergebnisse der Dipollokalisierung von fünf verschiedenen Startsimplices des zweiten Meßwertsatzes. Definitionen wie in Tabelle 3.4.

### 3.5 Einfluß von Störungen auf die Dipollokalisation

Im letzten Kapitel ist der große Einfluß von Meßstörungen auf die Genauigkeit der Dipollokalisationen gezeigt worden. Im Folgenden sollen weitere Tests zu diesem Problem vorgestellt werden. Dafür verwenden wir wieder den Querschnitt des Kopfs aus Kapitel 3.3. Dort sind zwei ungestörte Dipole (Dipol 1 bei (379, 399) und Dipol 2 bei (194, 249)) definiert worden und sollten durch das Verfahren der Dipollokalisation gesucht werden. Den Meßwerten beider Dipole werden nun Störungen überlagert und zwar auf folgende Weise: Erstens wird zu jedem der sieben Spannungsmeßwerten eine Störung addiert, wobei das Maß der Störung  $SNR$  (Signal to Noise Ratio) definiert ist durch das Verhältnis von Störung  $\delta$  zu den Meßwerten  $m$ :

$$SNR = \frac{\|\delta\|_2}{\|m\|_2} = \frac{\|m - m_\delta\|_2}{\|m\|_2}. \quad (3.26)$$

Testfälle für Störungsgrößen  $SNR = 0.01$  und  $SNR = 0.1$  stehen für beide Dipole zur Verfügung. Zusätzlich sind Testdaten erstellt worden, bei denen beide Dipole, jedoch mit unterschiedlicher Stärke, aktiv sind. Die Dipolsuche nach z.B. Dipol 1 ist durch den schwächeren Dipol 2 abgelenkt worden. Das Verhältnis der beiden Dipolstärken ist immer so festgelegt worden, daß der stärkere Dipol das gleiche Dipolmoment wie in Kapitel 3.3 hat, also ein Moment von 2 für Dipol 1 und ein Moment von 3.5 für Dipol 2, und der Stördipol ein Zehntel so groß ist. Somit können die Ergebnisse mit den exakten Werten verglichen werden. In weiteren Fällen sind diesen Meßwerten zusätzlich die Störungen  $SNR = 0.01$  und  $SNR = 0.1$  überlagert worden. Zum Beispiel bedeutet der Testfall Dipol 1 größer Dipol 2 mit  $SNR = 0.1$ , daß Dipol 1 ein Moment von zwei und Dipol 2 eines von 0.2 hat und den sich daraus ergebenden Meßwerten eine Störung von  $SNR = 0.1$  überlagert ist. Insgesamt sind also für jeden der beiden Dipole zusätzlich zum ungestörten Fall fünf weitere Fälle untersucht worden.

Für jeden dieser 12 Fälle sind die Dipollokalisationen von vier verschiedenen Simplexes (Abbildung 3.3) gestartet worden. Als Abbruchkriterium ist wie im Kapitel 3.3 gefordert worden, daß das kleinste relative Residuum kleiner als  $10^{-10}$  sein muß oder eine maximale Anzahl von Iterationen von 100 erreicht ist. Die Ergebnisse sind

in Tabelle 3.6 für Dipol 1 und in 3.7 für Dipol 2 dargestellt. Bei der Definition des Testfalls ist “größer” durch gr. abgekürzt worden, so daß z. B. der Fall 1 gr. 2 bedeutet, daß Dipol 1 größer als Dipol 2 ist. Gezeigt werden Daten für diejenige Dipolsuche der vier Startsimplices, die das kleinste relative Residuum erreicht. Unterschiede in den Ergebnissen treten nur bei den Dipollokalisierungen von Dipol 1 auf. In solchen Fällen sind jedoch die Unterschiede zwischen den Ergebnissen der vier Startsimplices sehr gering. Wirkliche Ausreißer sind nicht vorgekommen. Dargestellt sind die Relative Residuumsenergie RRE, die Koordinaten  $x$  und  $y$  des lokalisierten Dipols, der sich daraus ergebende Abstand zur exakten Dipolposition und das errechnete optimale Dipolmoment (Stärke). Zum Vergleich sind für jeweils den stärkeren der beide Dipole die exakten Daten hinzugefügt.

Beide Testdipole sind am Rand des Gehirns definiert, was die Suche in jedem Fall erschwert. Für Dipol 2 haben wir in Kapitel 3.3 gesehen, daß wir das direkt danebenliegende ventrikuläre System mit in die Rechnung einbeziehen können, um die Dipolsuche aus allen Richtungen überhaupt zu ermöglichen. Dipol 1 befindet sich rechts oben direkt am Schädel. Bei den Suchen dieses Dipols kommt es sehr oft vor, daß ein Testpunkt außerhalb des Gehirns ausprobiert wird. Der Nelder-Mead Algorithmus ist so programmiert, daß bei der Funktionswertberechnung in diesem Fall ein sehr großer Wert zurückgegeben wird, damit dieser Punkt nicht verwendet wird. Die Folge ist: Liegt der Reflektionspunkt außerhalb des Gehirns, so wird als nächstes immer eine innere Kontraktion durchgeführt. Befindet sich der Expansionspunkt außerhalb, so wird in jedem Falle der Reflektionspunkt verwendet. Dieser Einfluß auf die Bewegung des Simplex hat anscheinend keine sehr starke Auswirkung auf das Ergebnis, da keine Regelmäßigkeit gefunden werden konnte, daß die Ergebnisse von Dipol 1 schlechter sind als die von Dipol 2. Als einziger Unterschied läßt sich erkennen, daß nicht alle Suchen nach Dipol 1 den gleichen Ort lokalisieren im Gegensatz zum zweiten Fall. Wie gesagt sind jedoch die Unterschiede zwischen den vier Startsimplices sehr klein. Eine Vergrößerung des Nelder-Mead Gebiets auf Gehirn, ventrikuläres System und Schädel haben keine Verbesserung der Ergebnisse von Dipol 1 verursacht.

Die Folge der Meßstörungen ist klar ersichtlich. Jede Störung der Meßwerte wirkt sich negativ auf das Ergebnis der Dipollokalisierung aus. Die Suche des Dipols wird

also mit steigender Störung in der Regel schlechter. Dies ist gut bei den Suchen von Dipol 2, Tabelle 3.7, erkennbar, bei denen sich die Relative Residuumsenergie und der Abstand zum Zieldipol mit steigender Störung vergrößern. Die Lokalisationen von Dipol 1 ergeben teilweise widersprüchliche Ergebnisse im Fall eines zweiten Dipols, was eventuell auf das oben beschriebene Problem zurückgeführt werden kann, daß dieser Dipol direkt am Schädel positioniert ist. So verringert sich z.B. sogar das relative Residuum im Fall von Dipol 1 größer 2 und  $SNR = 0$  und  $SNR = 0.01$  etwas, obwohl die Störung der Meßdaten zunimmt. In allen Testfällen hat sich gezeigt, daß eine Vergrößerung der Störung zu einer Verschlechterung der Vorhersage des optimalen Dipolmoments führt. Außerdem sieht man bei beiden Lokalisation, d.h. von Dipole 1 und 2, daß sich der zusätzlicher Dipol schlechter auf die Suche auswirkt als die Überlagerungen vom Rauschen der Stärke  $SNR = 0.01$  oder  $SNR = 0.1$ .

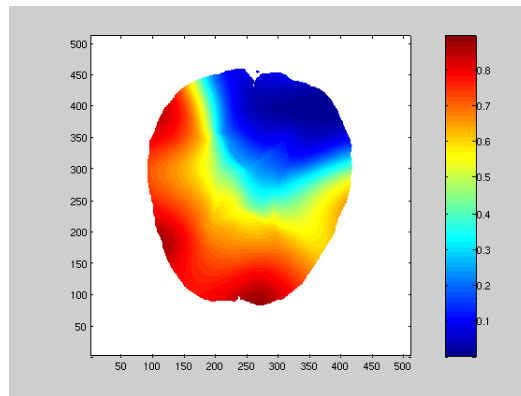
Vergleicht man jedoch das Ergebnis der 12 Dipollokalisierungen mit der jeweiligen Verteilung des Funktional, d.h. mit dem Deviation Scan, so sieht man, daß das Verfahren zur Dipollokalisierung fehlerfrei arbeitet. Denn auch der Deviation Scan gibt als Ort des kleinsten relativen Residuums nicht die exakte Dipolposition an, sondern einen anderen Punkt in der Nähe. Das Verfahren der Dipollokalisierung findet ein etwas geringeres relatives Residuum als der Deviation Scan in der selben Zelle oder einer Nachbarzelle. D.h. also, daß der Nelder-Mead Algorithmus das Minimum des Funktional zuverlässig lokalisiert, diese Position jedoch nicht mit der wahren Position des Dipols übereinstimmt. Die überlagerten Störungen bzw. der zweite Dipol bewirken, daß sich das Minimum des Funktional verschiebt und an "Tiefe" verliert. Untersuchungen der Größe des Simplex (2. und 3. Abbruchkriterium) bestätigen dieses Ergebnis. Die Simplices ziehen sich sehr stark zusammen, ohne daß sich die Residuumsenergie verringert. Dieses Verhalten haben wir schon bei den wirklichen Meßwerten in Abbildung 3.6 gesehen. Im Folgenden ist die Veränderung des Funktional durch Störungen an zwei Beispielen dargestellt. Gezeigt wird jeweils zuerst der ungestörte, danach der gestörte Fall und zuletzt die Differenz aus beiden, d.h. der zusätzliche Fehler.

Testfall	RRE	von Start- simplex	$x$ [cm]	$y$ [cm]	Abstand zu Dipol 1 [cm]	o. D. M.
1	5.616738e-11	4	14.824331	15.605623	1.906971e-04	2.000022e+00
1, SNR=0.01	4.207889e-05	alle	14.808048	15.595570	1.896044e-02	2.005670e+00
1, SNR=0.1	3.879516e-03	alle	14.673744	15.511592	1.773571e-01	2.065078e+00
1 gr. 2	5.142607e-03	alle	15.273437	15.529308	4.556273e-01	6.201919e-01
1 gr. 2, SNR=0.01	4.800732e-03	1, 4	15.273437	15.529699	4.555641e-01	6.148956e-01
1 gr. 2, SNR=0.1	5.477058e-03	1, 2, 4	15.273437	15.533290	4.549806e-01	5.676182e-01
exakt	0		14.8242187	15.6054687	0	2.0

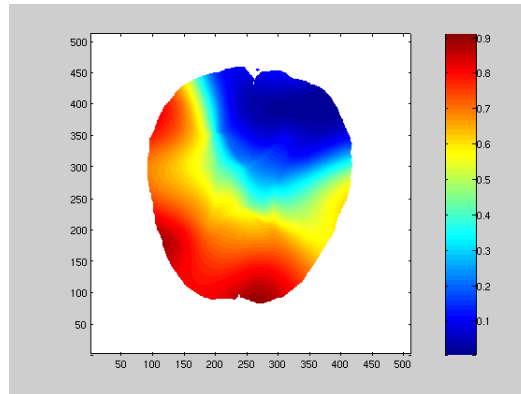
Tabelle 3.6: Ergebnisse der Dipollokalisierungen von **Dipol 1** mit verschiedenen Störungen.

Testfall	RRE	von Start- simplex	$x$ [cm]	$y$ [cm]	Abstand zu Dipol 2 [cm]	o. D. M.
2	1.935415e-10	alle	7.597060	9.746278	6.238535e-04	3.500221e+00
2, SNR=0.01	2.503530e-05	alle	7.597991	9.702030	4.406457e-02	3.470865e+00
2, SNR=0.1	2.755056e-03	alle	7.725362	9.252734	5.096201e-01	7.853980e+00
2 gr. 1	3.484255e-03	alle	7.715818	9.314690	4.472933e-01	4.688089e+00
2 gr. 1, SNR=0.01	3.516407e-03	alle	7.736361	9.267719	4.980778e-01	8.947793e+00
2 gr. 1, SNR=0.1	6.129476e-03	alle	7.761032	9.021910	7.423839e-01	2.176605e+01
exakt	0		7.5976562	9.7460937	0	3.5

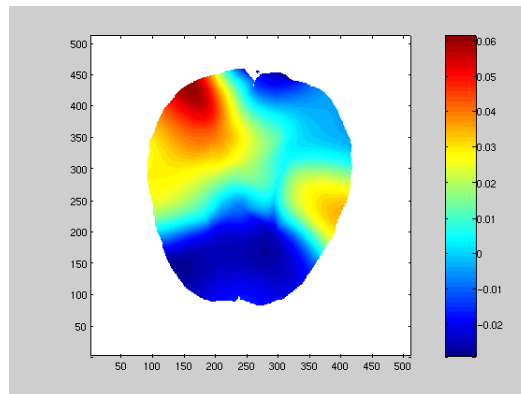
Tabelle 3.7: Ergebnisse der Dipollokalisation von **Dipol 2** mit verschiedenen Störungen.



Verteilung der Relative Residuumsenergie von Dipol 1.

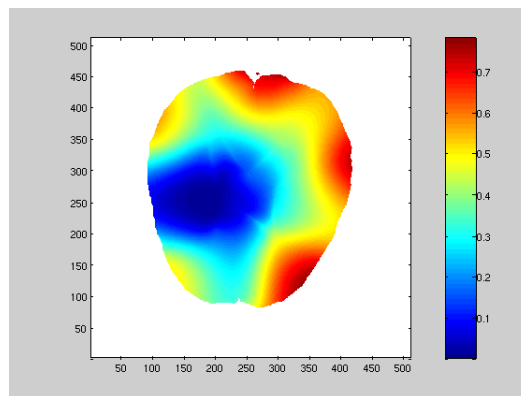


Verteilung der Relative Residuumsenergie von Dipol 1,  $SNR = 0.1$ .

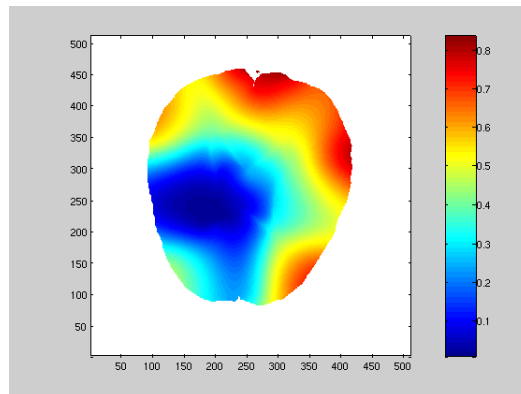


Differenz der Relativen Residuumsenergie von Dipol 1 und Dipol 1,  $SNR = 0.1$ .

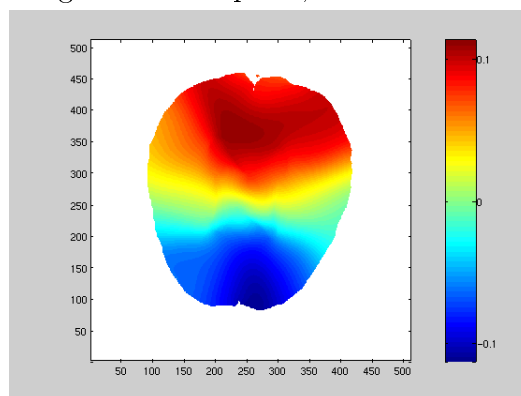




Verteilung der Relative Residuumsenergie von Dipol 2.



Verteilung der Relative Residuumsenergie von Dipol 2  
größer als Dipol 1,  $SNR = 0.1$ .



Differenz der Relativen Residuumsenergie von Dipol 2  
und Dipol 2 größer als Dipol 1,  $SNR = 0.1$ .

# Kapitel 4

## Zusammenfassung

In der vorliegenden Studienarbeit ist das Verfahren der Dipollokalisierung implementiert und an unterschiedlichen Meßdaten getestet worden. Die daraus gewonnenen Ergebnisse sind sorgfältig analysiert worden. Zuerst ist der für das Verfahren nötige Nelder-Mead Optimierungsalgorithmus programmiert und an einfachen Funktionen geprüft worden. Aufbauend auf dem schon vorhandenen Deviation Scan, einem weiteren Verfahren zur Dipolsuche, ist die Methode der Dipollokalisierung implementiert worden. Die Anwendung des Codes auf 2D Testmeßwerte und wirkliche EEG-Meßwerte und die genaue Analyse der Ergebnisse haben die Möglichkeiten und Grenzen des Verfahrens aufgezeigt. Die gewonnenen Kenntnisse können wie folgt zusammengefaßt werden:

- Der Nelder-Mead Algorithmus ist ein geeignetes Werkzeug, um in Verbindung mit dem Verfahren der Dipollokalisierung das Minimum des Funktionals zu lokalisieren.
- Mit ungestörten Meßwerten, die also durch numerische Simulationen erstellt worden sind, kann mit dem Verfahren der Dipollokalisierung der optimale Dipol sehr genau gefunden werden. Große Vorteile ergeben sich verglichen mit dem Deviation Scan, da der Dipol genauer lokalisiert werden kann, wenn wir den sehr wahrscheinlichen Fall voraussetzen, daß der Dipol nicht im Zentrum einer Zelle liegt. Außerdem muß eine viel geringere Anzahl von Vorwärtsproblemen gelöst werden.

- Der Dipol kann nur mit einer sehr viel schlechteren Genauigkeit gefunden werden, wenn wirkliche EEG-Meßwerte verwendet werden. Den Signalen der erkrankte Gehirnstelle sind Störungen anderer Quellen und des Meßverfahrens selbst überlagert, so daß das Minimum des Funktionals stark anwächst und sich verschiebt. Die Folge ist, daß bei gestörten Meßdaten das Simplex den wahren Dipol nur ungenau finden kann und sich um das verschobene Minimum zusammenzieht.
- Genauere Untersuchungen des Einflusses der Störungen haben ergeben, daß der Nelder-Mead Algorithmus auch bei gestörten Meßdaten fehlerfrei arbeitet, das Minimum jedoch wegen der Störungen nicht mit der wahren Dipolposition übereinstimmt. In der Regel verschlechtern sich die Lokalisationsergebnisse bei einer Vergrößerung der Störung.
- Die gleichen Vorteile gegenüber dem Deviation Scan wie bei ungestörten Meßwerten ergeben sich auch hier. Wird ein “guter” Satz von Meßwerten benutzt, so lokalisiert das Verfahren der Dipollokalisierung den Dipol genauer, weil die Suchen nicht nur an diskreten Punkten erfolgt. Die Anzahl der Funktionsauswertungen ist mehr als zwei Potenzen geringer als die des Deviation Scans.
- Um den Dipol bei gestörten Meßwerten mit dem Verfahren der Dipollokalisierung möglichst genau finden zu können, sollten folgende Hinweise beachtet werden:
  - Zuerst sollte der Satz von Meßwerten von einigen Dipollokalisierungen getestet werden. Damit kann abgeschätzt werden, wie groß das relative Residuum ist und wie das Abbruchkriterium gewählt werden muß. Ist die Relative Residuumsenergie zu groß, so sollte ein anderer Satz von Meßwerten dieser Epoche mit kleinerem Minimum verwendet werden.
  - Es sollten immer mehrere Dipolsuchen durchgeführt werden, die von unterschiedlichen Simplexes starten. Ergeben alle Suchen die gleiche Position und Orientierung des Dipols, ist das Minimum gefunden.

# Kapitel 5

## Ausblick

In Kapitel 3.1 ist das Vorgehen der EEG-Messung bei Epilepsiepatienten beschrieben worden. Der Datensatz einer charakteristischen EEG-Messung (Epoche) besteht aus 180 Zeitschritten, wobei ein Zeitschritt einen Satz von Meßwerten beinhaltet. Für die Dipollokalisierung und den Deviation Scan wird immer nur ein Zeitsample zur Suche des Dipols verwendet. Nun stellt sich die Aufgabe, nicht nur für einen einzigen Satz von Meßwerten die optimale Position, Orientierung und Stärke des Dipols zu finden, sondern im gesamten Datensatz der 180 Zeitsamples den optimalen Dipol zu lokalisieren. Gemäß der Definitionen aus Kapitel 3.1 läßt sich Gleichung 3.2 umschreiben zu

$$M_v(r, d) = \hat{L}(r) \cdot (d_1, d_2, \dots, d_s), \quad (5.1)$$

wobei jetzt  $M_v \in \mathbb{R}^{n \times s}$  die virtuellen Meßwerte ( $n \equiv$  Anzahl der Elektroden) und  $(d_1, d_2, \dots, d_s)$  die Dipolorientierungen der  $s$  Zeitsamples sind. Die Dipolorientierung eines Samples ist definiert durch

$$d_i = m_i \cdot \begin{pmatrix} e_1(\varphi, \vartheta) \\ e_2(\varphi, \vartheta) \\ e_3(\varphi, \vartheta) \end{pmatrix}, \quad (5.2)$$

wobei Stärke  $m_i$  mit dem Orientierungsvektor multipliziert wird. Der Orientierungsvektor hat die Länge eins und ist im drei dimensionalen Fall abhängig von  $\varphi$  und  $\vartheta$ .

Gleichung 5.1 kann nun umgeschrieben werden zu

$$M_v = \hat{L}(r) \cdot \begin{pmatrix} e_1(\varphi, \vartheta) \\ e_2(\varphi, \vartheta) \\ e_3(\varphi, \vartheta) \end{pmatrix} \cdot (m_1, m_2, \dots, m_s) = h(r, \varphi, \vartheta) \cdot m^T. \quad (5.3)$$

Das zu minimierende Funktional nimmt nun folgende Form an:

$$R = \|M - M_v\| = \|M - h(r, \varphi, \vartheta) \cdot m^T\|_F, \quad (5.4)$$

wobei  $M \in \mathbb{R}^{n \times s}$  die wirklichen Meßwerte aller Zeitsamples sind und  $\|\cdot\|_F$  die Frobeniusnorm ist. Um den Vektor  $h \in \mathbb{R}^{n \times 1}$  zu bestimmen, kann der Nelder-Mead Algorithmus verwendet werden. Diesmal werden nicht nur die drei Koordinaten des Dipols optimiert, sondern zusätzlich die Orientierung bestehend aus den zwei Winkeln  $\varphi$  und  $\vartheta$ . Der Vektor der Dipolstärken  $m^T$  wird durch

$$m^T = h^\dagger \cdot M \quad (5.5)$$

berechnet, wobei  $h^\dagger$  die More-Penrose Pseudoinverse ist. Der Vektor  $h^\dagger$  kann einfach bestimmt werden durch

$$h^\dagger = h^T \cdot (h^T \cdot h)^{-1}, \quad (5.6)$$

da  $h^T \cdot h$  ein Skalar und ungleich Null ist. Somit erhält  $m^T$  die gewünschte Form  $m^T \in \mathbb{R}^{1 \times s}$ .

# Literaturverzeichnis

- [1] J. A. Nelder und R. Mead, A simplex method for function minimization, *Computer Journal*, 7:308-313, 1965.
- [2] J. C. Lagarias, J. A. Reeds, M. H. Wright, und P. E. Wright, Convergence properties of the Nelder-Mead simplex algorithm in low dimensions, *SIAM J. Optim.*, 9:112-147, 1998.
- [3] M. Mohr, Comparison of solvers for a Bioelectric Field Problem, *Lehrstuhlbericht 01-2, Lehrstuhl für Informatik 10 (Systemsimulation), Friedrich-Alexander-Universität Erlangen-Nürnberg*, 2001.
- [4] D. J. Wood, An Interactive Approach for Solving Multi-objective Optimization Problems, *Ph.D. thesis, Rice University, Houston, Texas*, 1985.
- [5] K. I. M. McKinnon, Convergence of the Nelder-Mead simplex method to a nonstationary point, *SIAM J. Optim.*, 9:148-158, 1998.
- [6] L. Ammeraal, Programmdesign und Algorithmen in C, *Hanser Verlag, München*, 1994.