Estimation of
a smoothing parameter
of a spherical spline
interpolation

By O.R.E.Margarita

M.Sc. thesis
Carried out from September 1996 to June 1997
Under supervision of:
dr.ir.P.Cluitmans
ir.M.Quist

The department of Electrical Engineering of the Eindhoven University of Technology accepts no responsibility for the contents of M.Sc. theses or reports on practical training periods.
# Contents

Abstract........................................................................................................................................iii

Contents........................................................................................................................................iv

1. General Introduction....................................................................................................................1
   1.1 Introduction.............................................................................................................................1
   1.2 The electro-encephalogram.....................................................................................................1
   1.3 Brain mapping.........................................................................................................................2
   1.4 The electrodes positions.........................................................................................................3
   1.5 Overview of the problem.........................................................................................................5

2. Interpolation..................................................................................................................................7
   2.1 Introduction.............................................................................................................................7
   2.2 Spherical Spline Interpolation (SSI)......................................................................................8
   2.3 Scalp Current Density estimations.........................................................................................11
   2.4 A new method for the calculation of the Spherical Spline Interpolation...............................12
   2.5 Spherical Spline Interpolation with smoothing.....................................................................14
   2.6 Accuracy of the interpolation.................................................................................................15
       2.6.1 Residual variance.............................................................................................................15
       2.6.2 Evaluation methods.........................................................................................................15

3. Smoothing the spline......................................................................................................................18
   3.1 Introduction.............................................................................................................................18
   3.2 Choosing the smoothing parameter.......................................................................................19
       3.2.1 Numerical search method..............................................................................................21
       3.2.2 Golden Section with Fibonacci ratios.............................................................................24
   3.3 The algorithm.........................................................................................................................24

4. The extended brain mapping program..........................................................................................26
   4.1 Introduction.............................................................................................................................26
   4.2 Calculations............................................................................................................................27
   4.3 The output...............................................................................................................................27
## Contents

5. **Measurements and results**
   - 5.1 Introduction ........................................... 33
   - 5.2.1 Accuracy estimation without smoothing ........ 33
   - 5.2.2 Accuracy estimation with smoothing ........... 35
   - 5.3 Time calculations ...................................... 40

6. **Conclusions and recommendations** ........................................... 41

**Reference** ................................................................. 43

Appendix A ................................................................. 45

Appendix B ................................................................. 46
1. General Introduction

1.1 Introduction

The most vulnerable and complex organ of the human being is the brain. It is not only the control centre of the human body, but it also enables the body to communicate and interact with the outside world. The human brain makes it possible for us to talk with people and recognize people, animals and other things like buildings, trees and cars. To do this, the brain has at its disposal information from all kind of senses like tactile sense, hearing, olfactory and vision. The signals of these senses come together in the human brain, which classifies and reacts on them.

A lot of research has been done in the past years about the way the human brain interact with its surroundings. The research in this field can be divided in three different types, psychology, psychophysiology and neurobiology [10]. Psychology considers the human brain as a black box and tries to find the relations between the incoming and outcoming signals. Psychophysiology also considers these relations but it focuses on the brain itself. For example how the brain reacts on a certain stimulus or what role the brain plays on the reaction of the human body in some specific way. Neurobiology goes more into the brain itself. To achieve this kind of information, measurements have to be done on the reactions in the brain with few disturbances as possible. There are several ways to measure the brain activity and one of these ways is the use of a electro-encephalogram (EEG).

1.2 The electro-encephalogram

An EEG is a measurement of the electrical brain activity and is recorded with the use of electrodes that are placed on the scalp. The signals are stochastic and are not very appropriate for research on the source location of the signals. A better way to get a good view out the measurements is to make a topographic brain map. On the basis of the measured potentials on the different electrode positions a potential distribution is computed over the whole scalp. This is a topographic representation of the brain potentials on one particular time sample where the spatial distribution of the data are seen. The images gives an idea of the active and less active regions of the brain. So a brain map can be used for a more accurate analysis for source localisation of brain activity.

The main issue of topographic brain map is to give a better view for the analysis of the measurements and also a proper calculation of the location of the source. Although a brain map is based on measurements, these measurements are done only at the electrode positions and the brain map is reconstructed from these measured values. There are several techniques for the computation of a brain map, e.g., K-nearest Neighbours interpolation, spherical spline interpolation, thin-plate interpolation and polynome interpolation. In this thesis only the Spherical Spline Interpolation (SSI) is used, because from former research it appears that SSI is one of the best and easier for computation purpose for brain mapping ([1], [5], [6]). This thesis is a follow up of the work “Brain mapping with EEG signals “by E.Giele [6].
The Medical Electrical Engineering group of the Eindhoven University of Technology in corporation with the Psychonomics section of the Tilburg University has done a lot of research on brain activity. This research is done to obtain more knowledge how the brain of the human being behaves. One can think about controlling someone for epilepsy or brain diseases, the reaction of a person on some stimulus or someone who has sleep disorder and also for monitoring the brain activity of a patient under anaesthesia. That is why the Medical Electrical Engineering group of the Eindhoven University of Technology and the psychonomics section of the Tilburg University has decided to make a brain map program. In the work done by Giele [6], the brain map program is described. In this thesis the old brain map program will be extended, but the conditions are the same as for the old one.

These conditions are:

1) The extended program that is written can work with the data acquired at the Tilburg University (in European Data Format, EDF).

2) The program should be able to run on a normal DOS personal computer and if it is possible to produce multiple brain maps after each other so the change in time can be visualized.

1.3 Brain Mapping

Brain mapping is a visualisation of the brain activity. The first one who had made a brain map was the Japanese scientist Motokawa. He used multiple electrodes and wrote the average values on a map of the scalp, and after that he drew equipotential lines on the map and in this way he created the first brain map. Brain maps are usually used in physiological and in neuropsyphological research on the functioning of the brain, e.g., to check if someone has epilepsy. Today modern techniques make it possible to use large quantities of electrodes (from 23 up to 128) and modern computers can calculate these maps in a few seconds. The new methods for drawing brain maps are still based on the same EEG signals. The only difference between now and the earliest maps are how the signals are recorded and processed.

The classical recording in the earliest maps goes as follows, while multiple pens move up and down as a function of the potential on an electrode, the paper itself is pulled underneath them in a direction perpendicular to the pen movement and thus multiple signals are drawn which indicates the voltage as function of time. A disadvantage of this way of processing is that the data (potentials) cannot be analysed in spatial aspect.

The main point of attention with this type of recording is the wave form of the signals. The new type of recording for drawing maps are possible by the arrival of modern electronics and computers. These recordings are not written on paper, but the measurements are immediately digitized and stored on a digital medium (hard disk or tape). This is not the only advantage regarding to the earliest maps, but also the accuracy improves and the most important advantage is that it becomes easier to prepare the measurement for other ways of analysis. One of these ways of analysis is the making of a brain map.
The potential distribution on the scalp, the Scalp Potential (SP) increases the view in the spatial aspects of the EEG, but it also has its imperfections. The brain as well as the scalp are good conductors but the skull is a good insulator. This will cause the electrical field created by the brain and measured on the scalp to be smeared and because of this smearing effect the local character is lost. The EEG and the potential distribution are also dependent of the reference electrode during the measurement. Especially the last argument has leaded to the introduction of another reference free presentation of the data, the Scalp Current source Density (SCD) [11, 13]. The SCD shows where the current sources and current sinks are on the scalp. Another additional advantage of SCD is that it suffers less of smearing effect.

The main disadvantage of Scalp Current source Density is the less reliable reproduction of the activities of the deeper located sources. For this is SP distribution a better solution. For a reliable reproduction of topographic maps both SP and SCD are needed. The SCD is mathematically the second derivative of the potential distribution, thus to have a SCD reproduction the second derivative of the potential distribution need to exist. Practically this means that the interpolated potential map (the SP) must be continuous.

1.4 The electrodes positions

To measure the brain activity (brain potentials) the electrodes have to be placed on the head. The questions that raised are ‘Can the electrodes be put in a regular way?’ and ‘How many electrodes are needed?’ The measurements are local, this means that the only values that are known are those at the positions of the electrodes. A solution to this problem is to improve the resolution (spatial accuracy of the measurement) by increasing the number of electrodes. The main disadvantages of this method are the increasing cost of the equipment and that an electrode has physical dimensions and needs space to be placed on the head. Another problem is the time that it will take to place all the electrodes on the head. A solution for the resolution is interpolation. But interpolation has also his imperfection. Interpolation can never be able to reconstruct the real situation other then by coincidence. Interpolation is not able to construct data, but only to guess the real value at a given position.

The positions of the electrodes are not described in centimetres or millimetres but they are described in angles (from a spherical model of the head). The names of these angles are latitude and azimuth. Latitude is the angle seen from the top view with the nose as 0 degrees and azimuth the angle seen from the top of the head as 0 degrees, see figure 1.1. There is an international standard for the electrode positions [1], is called the 10-20 system. The head is regarded as a sphere for this international standard.
Brain mapping

Introduction

Figure 1.1: Definition of latitude and azimuth.

Figure 1.2: Electrode configuration for EEG recordings (23 electrodes).

Table 1.1: Description of the electrodes positions.

<table>
<thead>
<tr>
<th>Name</th>
<th>Azimuth (θ)</th>
<th>Latitude (φ)</th>
<th>Name</th>
<th>Azimuth (θ)</th>
<th>Latitude (φ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fp1</td>
<td>72.00</td>
<td>18.00</td>
<td>T5an</td>
<td>72.00</td>
<td>114.00</td>
</tr>
<tr>
<td>Fp2</td>
<td>72.00</td>
<td>342.00</td>
<td>TP3</td>
<td>55.08</td>
<td>111.82</td>
</tr>
<tr>
<td>F7</td>
<td>72.00</td>
<td>54.00</td>
<td>TP4</td>
<td>55.08</td>
<td>248.18</td>
</tr>
<tr>
<td>F8</td>
<td>72.00</td>
<td>306.00</td>
<td>T6an</td>
<td>72.00</td>
<td>114.00</td>
</tr>
<tr>
<td>T3</td>
<td>72.00</td>
<td>90.00</td>
<td>T5po</td>
<td>72.00</td>
<td>138.00</td>
</tr>
<tr>
<td>T5an</td>
<td>72.00</td>
<td>138.00</td>
<td>P3</td>
<td>47.66</td>
<td>139.12</td>
</tr>
<tr>
<td>TP3</td>
<td>55.08</td>
<td>220.88</td>
<td>Pz</td>
<td>36.00</td>
<td>180.00</td>
</tr>
<tr>
<td>TP4</td>
<td>55.08</td>
<td>248.18</td>
<td>P4</td>
<td>47.66</td>
<td>220.88</td>
</tr>
<tr>
<td>T6po</td>
<td>72.00</td>
<td>222.00</td>
<td>O1</td>
<td>72.00</td>
<td>162.00</td>
</tr>
<tr>
<td>O2</td>
<td>72.00</td>
<td>198.00</td>
<td>O2</td>
<td>72.00</td>
<td>198.00</td>
</tr>
</tbody>
</table>

T4  | 72.00       | 270.00       |
Of course the number of electrodes can be increased. Some experiments use 63 electrodes which are mounted in a kind of bathing cap. This is an easy way to connect all the electrodes to the scalp. Another type of measurement is with an electrode array. An electrode array consists of electrodes which are placed very close together in a matrix form. This type is used when the approximate location of the effect to be measured is known and scientists are interested in the detailed distribution of the scalp potential. This last type shows only a small piece of the brain activity and not the entire brain.

The choice of how many electrodes will be used in an experiment, depends on the purpose of the research. In [15] 32 electrodes have been proved to be sufficient for general mapping (mapping over the entire scalp).

Now that the problems of how many electrodes is needed and how the electrodes will be placed are solved, rest only the problem of how the image projection of the scalp potential and the source current density will look. There are three ways for visualisation.

- First method is the use of **equipotential lines**. Dotted lines are usually negative and solid lines are positive.

- Second is the use of **grey scale image**. The higher the value (potential μV or current μA) the lighter the grey and the lower the value the darker the grey.

- And third is the **use of colours**. Here it depends on the users to define the colours.

In this thesis, colour projection as used in [6] is applied, because with colour the extremes are better visible. The colours are, blue is for negative, green is for positive, black is neutral (zero) and the background of the images are red.

Another difficulty by image projection is the projection of the head on a paper or screen. Because of the curvature of the head, the sides will be compressed in a fairly small area. The use of radial projection instead of perpendicular projection is a solution to this problem. In radial projection the distance to the centre is linear with the angle of the point to be projected, instead of the sinus of the angle. See [6], page 13 and 14 for more explanation.

### 1.5 Overview of the problem

In the psychophysiological research spherical spline functions are used either for topographic mapping of the surface potentials, or the estimations of the Scalp Current Density (SCD) as described in [13]. Advantages of this technique are an improved reliability of the estimates at the borders of the electrodes montage as compared to the other methods (e.g. Hjorth derivations) and a relatively short computation time. In [6] a program is made for the plotting of the surface potential (Scalp potential SP). This program has his limitations, for example, the program can only plot Scalp Potential and the order of the spline function was two (m=2),
thus no Scalp Current Density (SCD) can be plot. Scalp Current Density is the second derivative of the potential distribution (SP) and the order of the spline function must be greater than one \( m > 1 \). In chapter two more of this subject will be discussed.

A problem by topographic mapping of SP or SCD is that the data acquired by the measurements contain noise (the potentials that was measured by the electrodes). Using this one can get the wrong information of the plotting. First step to take is to level the noise out of the measure data as much as possible. A solution is smoothing. By using spherical spline interpolation it is possible to smooth the data (by including a smoothing parameter \( \lambda \) in the spherical spline interpolation). Smoothing increases the signal-to-noise ratio by smearing the noise. In this thesis a smoothing parameter \( \lambda \) and order of the spline \( m \) will be estimated using the “leave-one-out-method” as described by G.Wahba [16,17]. The square root average of the errors in the estimations on each omitted positions is minimized as a function of \( \lambda \) and \( m \).

In the extended program “Brainmap” as well as SCD and SP can be plot by different order \( m \) (\( m \) must be greater than one) and also with the opportunity to include a smoothing parameter \( \lambda \).
2. **Interpolation**

2.1 **Introduction**

Interpolation is a mathematical technique to recreate data on points where no measurements has been done, based on the measurements that have been done in the places surrounding them. The calculated values by interpolation is a function of the surrounding measure values and the distances to these measurements. These calculated values are not real values, but only a carefully guess on what should the real value should be on that particular place.

Interpolation is used in brain mapping because of the limited number of the electrodes where the measurements had been taken place. Brain map is a picture of the entire scalp so the potentials around the electrodes need to be calculated. Accuracy is a must in brain mapping because of the dealing of human life and that one cannot make mistake or get the wrong information from the pictures. This means that the values calculated by the interpolation should approach the real value as good as possible. From a brain map picture one can get crucial information, for example where the maxima and minima are situated and with this information one can determine the direction and place of the dipoles. The potentials measured on the scalp are generated by the brain. And these sources located at different positions in the brain, can be modelled as the dipoles.

There are many interpolation techniques but in brain mapping three of them are usually used,

* K-Nearest Neighbours interpolation.
* Polynome interpolation.
* Spline interpolation ( is divided in Thin-plate interpolation and Spherical Spline interpolation).

From table 2.1 on page 8 one can conclude that spherical spline is the best technique for brain mapping, because SCD can be easily computed from SP and also the smoothening of the data can be easily performed with spherical spline interpolation. It will be explained later on in chapter three. Another advantage of spherical spline interpolation is that extremes are free from the electrodes.

The disadvantage of spherical spline interpolation is the long calculation time. Despite of this problem, current developments on computer hardware are so fast that they will catch up with the computational speed needed, if that would be necessary, on a short term.
Table 2.1: Advantage and disadvantage of the several interpolation techniques.

<table>
<thead>
<tr>
<th>Interpolation technique</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
</table>
| K-Nearest Neighbours    | - Short calculation time  
                          - generally used                                                          | - bad results when the among of electrodes is small  
                          - discontinuity  
                          - extremes bound with the electrode positions                                |
| Polynome                | - no trouble with discontinuity  
                          - extremes do not bound with the electrodes  
                          - good results with little electrodes  
                          - short calculation time                                                      | - possible oscillation problem  
                          - measure values are not exact on electrode positions                         |
| Thin-plate spline       | - better for flat planes                                                   | - sometimes discontinuities                                                   |
| Spherical spline        | - no discontinuities  
                          - extremes not necessarily on the electrodes  
                          - no oscillation problems  
                          - easy to calculate SCD from SP  
                          - interpolation based on entire scalp instead of on a small area  
                          - good for sphere area (like the head form)                                 | - long computational times                                                   |

Finally a good and fast interpolation algorithm has to be written including the calculation of the smoothing parameter $\lambda$ that will result in a faster program because if the program is too slow, it’s usefulness is limited because it would take to much time to calculate a large amount of brain maps. Speed and accuracy are a must in the extended program.

2.2 Spherical Spline Interpolation (SSI).

The Spherical Spline theory is developed from a meteorological point of view [16,17], in that time scientists were interested in a smooth function to describe the 500 millibar height (the height above the sea level at which the pressure is 500 millibars). For that purpose a large number of weather stations were distributed around the world. To describe this in a function the scientist G. Wahba has decided to use periodic splines on a circle and to generalize thin plate surface splines to spherical splines, this because of the spherical form of the earth. At the end of the 80’s the scientist J. Perrin have applied this technique to interpolate scalp potentials.
The mathematical formulas of the spherical spline look quite complicated. But first we shall describe how the spherical spline interpolation works. Let $z_i$ be the potential value measured at the $i^{th}$ electrode whose spherical projection will be denoted by $E_i$. The value at point $E$ of the spherical spline $U$ which interpolates the $z_i$'s at the $E_i$'s is given by:

$$U(E) = c_0 + \sum_{i=1}^{N} c_i g(\cos(E, E_i))$$

where the $c_i$'s are the solutions of:

$$\begin{cases}
G.C + c_0.t = Z \\
t^T.C = 0
\end{cases}$$

with $t = [1,1,\ldots,1]^T$, $C = [c_0,c_1,\ldots,c_{Ns}]^T$

$Z = [z(E_1),z(E_2),\ldots,z(E_{Ns})]$, $G = (g_{ij}) = G(E_i, E_j) = (g(\cos(E_i, E_j)))$

$G$ is symmetrical since $\cos(a,b)$ is equal to $\cos(b,a)$

$\cos(E_i,E_j)$ denotes the cosine of the angle by the sphere’s centre point $O$, $E_i$ and $E_j$.

The function $g(x)$ is defined as the sum of the following series:

$$g(x) = \frac{1}{4\pi} \sum_{n=1}^{\infty} \frac{2n+1}{n^n(n+1)^m} P_n(x)$$

The interpolation is calculated as follows: First the formulas in 2.2 are solved. The resulting vector $c$ and the value $d$ are then filled in 2.1 to calculate the potential at each point. The set of equations 2.2 is as follows,

$$\begin{pmatrix}
g_{11} & g_{12} & \cdots & g_{1Ns} \\
g_{21} & g_{22} & \cdots & g_{2Ns} \\
\vdots & \vdots & \ddots & \vdots \\
g & g & \cdots & g
\end{pmatrix}
\begin{pmatrix}
c_1 \\
c_2 \\
\vdots \\
c_{Ns}
\end{pmatrix}
+ 
\begin{pmatrix}
c_0 \\
c_0 \\
\vdots \\
c_0
\end{pmatrix}
= 
\begin{pmatrix}
z_1 \\
z_2 \\
\vdots \\
z_{Ns}
\end{pmatrix}$$

$$(1 \ 1 \ \cdots \ 1) \begin{pmatrix}
c_1 \\
c_2 \\
\vdots \\
c_{Ns}
\end{pmatrix} = 0$$
, with \( g_{ij} = g(\cos(E_i, E_j)) \)

If these two matrices are combined with each other we get one matrix,

\[
\begin{pmatrix}
  g_{11} & g_{12} & \cdots & g_{1Ns} \\
  g_{21} & g_{22} & \cdots & g_{2Ns} \\
  \vdots & \vdots & \ddots & \vdots \\
  g_{Ns1} & g_{Ns2} & \cdots & g_{NsNs} \\
  1 & 1 & \cdots & 1
\end{pmatrix}
\begin{pmatrix}
  c_1 \\
  c_2 \\
  \vdots \\
  c_{Ns} \\
  0
\end{pmatrix}
= 
\begin{pmatrix}
  z_1 \\
  z_2 \\
  \vdots \\
  z_{Ns} \\
  0
\end{pmatrix}
\]

This will give the following set of linear equations, which is easy to solve with the use of a computer, for example with LU-decompositions.

\[
\begin{pmatrix}
  g_{11} & g_{12} & \cdots & g_{1Ns} \\
  g_{21} & g_{22} & \cdots & g_{2Ns} \\
  \vdots & \vdots & \ddots & \vdots \\
  g_{Ns1} & g_{Ns2} & \cdots & g_{NsNs} \\
  1 & 1 & \cdots & 1
\end{pmatrix}
\begin{pmatrix}
  z_1 \\
  z_2 \\
  \vdots \\
  z_{Ns} \\
  0
\end{pmatrix}
\]

The \( m \) in 2.3 is the order of the spherical spline interpolation. Depending on the constant \( m \) chosen (it must be an integer greater than 1) one obtains a set of functions \( V \) with \( x \) between -1 and 1. Besides, it should be noticed than the Legendre polynomial can be easily put in a formula and can be computed by using the recurrence relation:

\[(n+1)P_{n+1}(x)=(2n+1)xP_n(x)-nP_{n-1}(x),\]

with \( P_0 = 1 \) and \( P_1 = x \).

**Note**

The formula in 2.3 is only useful for \( m < 3 \), with \( m = 3 \) the picture (brain map) that one get is hazy. In paragraph 2.4 a new formula will be given that is valid as well as for scalp potentials and the estimation of scalp current density for all \( m \geq 2 \).

With formulas 2.1, 2.2, 2.3 and 2.5 the scalp potentials can be calculated where no measurement has taken place. The last step is to write a suitable algorithm that can calculate fast and make maps of scalp potentials or scalp current density.
2.3 Scalp Current Density estimations

In chapter one a method is mentioned for the representation of EEG-data in another form, the Scalp Current Density (SCD). SCD has advantages above SP. SCD is reference independent and its peaks and through are sharper than those of the scalp potential (SP) and also SCD suffers less from smearing effect. SCD mapping appears to be a valuable tool to spatially split smeared SP distribution due to simultaneously active generators. The SCD map may be computed from any sufficiently smooth (continuous) mathematical SP map.

The Scalp Current source Density \( I \) is defined as the radial component of the gradient of the current density \( J \) and has as unit \( \text{A/m}^3 \). With a conducting coefficient \( \sigma \) of the neural tissue and a electric field \( \xi \), \( J \) can be write as [12]:

\[
J = \sigma \xi = -\sigma \nabla U
\]

For the Scalp Current Density \( I \) is as follows:

\[
I = \left( \frac{\partial J_i}{\partial e_i} + \frac{\partial J_j}{\partial e_j} \right) = -\sigma \left( \frac{\partial^2 U}{\partial e_i^2} + \frac{\partial^2 U}{\partial e_j^2} \right)
\]

One of the advantages of spherical spline interpolation is that the Scalp Current Density can very easily be calculated, because with spherical spline method a distribution can be calculated without discontinuities and scalp current density is the second derivative of the scalp potential. Thus with the formulas of spherical splines interpolation the scalp current density can be calculated. The only thing to do is to adjust the formulas. The calculation is solved in a number of sequences steps:

The 2-dimensional spherical laplacian of the Legendre polynomials is the multiplication of the same Legendre polynomial:

\[
\nabla^2 P_n = \frac{-n(n+1)}{R^2} P_n
\]

The expression of the current density \( I(E) \), which is proportional to the second dimensional spherical laplacian of the potential then becomes quite simple:

\[
I^\sigma(E) = \frac{\sigma}{R^2} \sum_{i=1}^{N} c_i h(\cos(E, E_i)) \tag{2.7}
\]

\( R \) is the radius of the head and \( \sigma \) is the conducting coefficient.

with

\[
h(x) = \frac{1}{4\pi} \sum_{n=1}^{\infty} \frac{2n+1}{n^{n-1}(n+1)^{n-1}} P_n(x) \tag{2.8}
\]
The coefficients $c_i$'s in 2.7 is the same as in 2.1. So if the coefficients $c_i$'s in 2.1 are solved they can so be used in 2.7. The total calculation time is the same for scalp potential and scalp current density [2]. The functions $g(x)$ en $h(x)$ have the same form and are easy to be put in a table, because $x$ represents the cosines of the angle between the electrode position and the position where the potential is going to be predicted. The cosine function has his values between -1 and 1. If one can write a program for the calculation of scalp potentials, it will be easy to calculate the estimation of the current (SCD) in that same program.

2.4 A new method for the calculation of the Spherical Spline Interpolation

The formulas for spherical spline interpolation (formula 2.3 and 2.8) given in the previous paragraphs are only applicable for order less than three ($m < 3$). Another formula will be given in this paragraph for spherical spline interpolation for all $m \geq 2$.

The problem of formulas 2.3 and 2.8 is that there is not a closed form expression available for larger $m$ ($m > 3$). A closed form expression will be obtained when we can approximate the function, for example for $m=2$ formula 2.3 becomes as follows,

$$g_2(x) = \sum_{n=1}^{\infty} \frac{2n+1}{n^3(n+1)^2} P_n(x)$$

leaving the Legendre polynomial $P_n(x)$ out, the formula becomes,

$$\frac{2n+1}{n^3(n+1)^2}$$

is equal to:

$$\frac{1}{n^2} - \frac{1}{(n+1)^2}$$

This function is the result of the following integration

$$\frac{1}{n^2} - \frac{1}{(n+1)^2} = \int_0^1 \log x (1 - \frac{1}{x}) x^n \, dx \quad n=1, 2, \ldots$$

The Legendre polynomial is also achieved from a summoning of integration,
\[ \sum_{n=0}^{\infty} h^n P_n(x) = (1 - hx + h^2)^{-1/2} - 1, \quad -1 < x < 1 \]  

formula 2.10 with formula 2.11 gives

\[ g_n(x) = \int_0^1 \log \left( \frac{1}{h} \right) \left( \frac{1}{\sqrt{1 - 2hx + h^2}} - 1 \right) dh \]  

This is the approximation of \( g(x) \) with \( m = 2 \). For \( m > 2 \) there is not an approximation available, that is why the formula 2.3 cannot be used for order greater than \( m > 2 \). This is also valid for the function \( h(x) \) in formula 2.8, because it has the same form (\( h(x) \) is one order lower than \( g(x) \)).

In [17] G. Wahba has presented another formula that is almost the same with \( g(x) \) that is applicable for all \( m \geq 2 \). The formula is as follow,

\[ R_m(x) = \frac{1}{2\pi} \sum_{n=1}^{\infty} \frac{1}{(n+1)(n+2)\ldots(n+2m-1)} P_n(x) \]  

with \( x = \cos(P, P) \) and \( m \) the order of the spline function

The formula 3.5 has a closed form expression, because it can be approximated for all \( m \geq 2 \), for more details see [17] page 13 and 14.

The spherical spline interpolation become now:

\[ U^p(P) = c_0 + \sum_{i=1}^{N} c_i r(\cos(P, P_i)) \]  

For the solutions of \( c_i \)'s the following matrix has to be solved,

\[ \begin{cases} R.C + c_0.t = Z \\ t^T.C = 0 \end{cases} \]  

The \( h(x) \) function for the calculation of the SCD will also change, because SCD is the second derivative of the SP (derivation of a Legendre polynomial is a multiplication of the Legendre polynomial itself, see paragraph 2.3).

The new \( h(x) \) is,

\[ h_n(x) = \frac{1}{2\pi} \sum_{n=0}^{\infty} \frac{n(n+1)}{(n+1)(n+2)\ldots(n+2m-1)} P_n(x) \]
Brain mapping

Interpolation

formula 3.8 is also equal to

\[
 h_n(x) = \frac{1}{2\pi} \sum_{n=0}^{\infty} \frac{n}{(n+2)(n+3)\ldots(n+2m-1)} P_n(x) \quad 2.17
\]

with formula 2.17 computation time can be gained (one multiplication less)

The spherical spline for SCD is now:

\[
 I^\sigma(P) = \frac{\sigma}{R^2} \sum_{i=1}^{N} c_i h(\cos(P, P_i)) \quad 2.18
\]

with \( \sigma = 0.45 \) Sie/m (\( \sigma \) conducting coefficient) and \( R=0.09 \) m (\( R \) is the radius of the head). The \( c_i \)'s are the same \( c_i \)'s that are calculated for SP spline interpolation (formula 2.15).

These are the formulas that will be used for spherical spline interpolation in the extended "Brainmap" program for the calculation of SP and SCD. These formulas are different from the formulas that was used in [6].

2.5 Spherical Spline Interpolation with smoothing

Besides spline interpolation there is also spline smoothing. Spline smoothing is when a smoothing parameter is added at the formula of spherical spline (see equation 2.19). The use of a smoothing parameter is useful when the data contains noise. The measured data that will be used in this thesis contain noise, so it is useful to implement an algorithm for smoothing of the measure values.

The Spherical Spline Interpolation formula with smoothing is:

\[
 \begin{align*}
 (R + \lambda)C + c_0 & = Z \\
 \ell^T C & = 0
 \end{align*} \quad 2.19
\]

\( \lambda \) = the smoothing parameter

The rest of the Spherical Spline Interpolation formula remains the same. With the use of a smoothing parameter one allows a certain deviation is tolerated to the sphere surface. The calculated line (or surface) will be forced not to go through all measured values. This is relevant when the measured values contain noise. The smoothing parameter \( \lambda \) depends on the measured data and the order of the spline function. There are statistical methods for the estimation of the smoothing parameter \( \lambda \).

In the next chapter the subject of smoothing parameter will be discussed and how the smoothing parameter \( \lambda \) will be estimated.
2.6 Accuracy of the interpolation

There are several statistical ways to calculate the accuracy of an interpolation. In this thesis the accuracy will be calculated with,

- Residual variance
- Evaluation methods (cross validation)

2.6.1 Residual variance

The residual variance gives the error between measured value and predicted value. It is a kind of measure of how good the predicted value fits in the measured value.

The residual variance is obtained from,

\[ \text{var}(v) = \frac{1}{T} \sum_{i=1}^{T} (U^p(t) - U^m(t))^2 \] 2.20

with \( U^p \) is the predicted value (interpolated value), \( U^m \) the measured value and \( T \) the number of samples.

2.6.2 Evaluation methods

The method of evaluation involved cross-validation where the measured potential at each electrode is predicted by interpolation by leaving out every electrodes once.

The error in the interpolation is based upon the difference between the predicted and the measured potentials at each electrode. For electrode \( i \), sample index \( t \), and scalp potential \( U(i,t) \), the residual potential is \( U^r(i,t) \), is defined as:

\[ U^r(i,t) = U^p(i,t) - U^m(i,t) \] 2.21

where \( U^p \) is the interpolated potential and \( U^m \) is the measured potential. The complete variance of the measured potentials is obtained from:

\[ \sigma_m^2 = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} U^m(i,t)^2 \] 2.22

the variance over the predicted potential is
Brain mapping

Interpolation

\[ \sigma_p^2 = \frac{1}{N_s T} \sum_{i=1}^{N_e} \sum_{t=1}^{T} U^p(i,t)^2 \]  \hspace{1cm} 2.23

and the complete residual variance is,

\[ \sigma_r^2 = \frac{1}{N_s T} \sum_{i=1}^{N_e} \sum_{t=1}^{T} U^r(i,t)^2 \]  \hspace{1cm} 2.24

where \( T \) is the number of samples and \( N_s \) the number of electrodes. Equation 2.24 gives the averaged residual variance over all electrodes and equation 2.20 gives the averaged residual variance of one electrode.

The covariance between the predicted (interpolated) and the measured potentials is calculated as follows,

\[ \sigma_{pu}^2 = \frac{1}{N_s T} \sum_{i=1}^{N_e} \sum_{t=1}^{T} (U^p(i,t).U^m(i,t)) \]  \hspace{1cm} 2.25

and the covariance between the residual and the measured potential is,

\[ \sigma_{rm}^2 = \frac{1}{N_s T} \sum_{i=1}^{N_e} \sum_{t=1}^{T} (U^r(i,t).U^m(i,t)) \]  \hspace{1cm} 2.26

The normalized residual variance \( nr_v \) is obtained from:

\[ nr_v = \frac{\sigma_r^2}{\sigma_m^2} \]  \hspace{1cm} 2.27

The value of \( nr_v \) for a perfectly accurate interpolation would be zero, because it includes the error and the variance of the measured potential. \( nr_v \) is a measure of inaccuracy of the interpolator.

The precision of the interpolator is measured by the prediction covariance (\( cv \)) and is defined as:

\[ cv = \frac{\sigma_{pm}^2}{\sqrt{\sigma_p^2 \sigma_m^2}} \]  \hspace{1cm} 2.28

\( cv \) is a measure of precision because it describes how closely the signal form predicted by the interpolation technique follows the measured signal form. The value of \( cv \) for a perfectly precise interpolator would be one.

The residual covariance \( rc_v \) is a measure of the correlation between the residual signal and the measured signal. \( rc_v \) is calculated as follow,
Brain mapping

\[ r_c v = \frac{\sigma_{m}^2}{\sqrt{\sigma_m^2 \sigma_m^2}} \]  

\( r_c v \) measures whether the interpolation technique over or underestimates the measured signal. A negative value of \( r_c v \) indicates a consistent underestimation while a positive value indicates an overestimation. The value of \( r_c v \) for a good interpolation would be zero.
3. **Smoothing the spline function**

3.1 Introduction

Data acquired during measurements contain most of the time noise, so a method must be designed to level the noise as much as possible out of the measured data. There are several ways to achieve this, for example by using a filter (low pass filter) or by smoothing the data. Smoothing is used when the data are already stored on disk or other media, while a filter is used before the data are stored. In this thesis only smoothing will be treated, because the data are already stored on disk. With smoothing a certain deviation is tolerated to the measured values so the spline is not forced to go trough all the measured points. The splines in figure 3.1 and 3.2 show the difference between a smoothed and an unsmoothed spline. The order \( m \) of the spline function represents the stiffness in spherical shell and the smoothing parameter \( \lambda \) represents the amount of smoothing.

![Figure 3.1: Unsmoothed spherical spline approximation. Vertical axis: arbitrary units](image)

![Figure 3.2: Smoothed spherical spline approximation. Vertical axis: arbitrary units](image)
Before smoothing can be applied, the parameter $\lambda$ has to be calculated. There are statistical methods for the estimation of the smoothing parameter $\lambda$, for example with "general cross validation (GCV) or ordinary cross validation (OCV)". The goal of GCV and OCV is to minimize the error between measured signal and interpolated signal by using a smoothing parameter $\lambda$ during the calculation of the interpolated signal. In the next paragraphs more details will be given for the estimation of the smoothing parameter $\lambda$ and as well an algorithm will be presented.

### 3.2 Choosing the smoothing parameter

The goal of smoothing is to minimize the difference between the interpolated (predicted) value and the real (measured) value. To do that the smoothing parameter $\lambda$ must be known. The optimal value of the smoothing parameter $\lambda$ depends on the EEG data. The estimation of this value takes place with the EEG data. Suppose the measured data $\{z_i\}$ can be written as $z_i = u(P_i) + \varepsilon_i$, where $u$ is an arbitrary continuous linear function and $\varepsilon_i$ is a measurement error. The function $u$ must be able to minimize [18]:

A) $J_\lambda(u) \text{ subject to } u(P_i) \approx z_i$

or

B) $N^{-1} \sum_{i=1}^{N} (u_i - z_i)^2 + \lambda J_\lambda(u)$  \hspace{1cm} 3.1

$N$ is the number of measure points and $m = 2, 3, \ldots$

where

$$J_\lambda(u) = \int \sum_{m=0}^{M} \left( \frac{\partial^m u}{\partial x_1^r \partial x_2^{m-r}} \right)^2 dx_1 dx_2$$  \hspace{1cm} 3.2

$m = 2, 3, \ldots$

The problem with the function in 3.2 is that,

1) it is difficult to compute.
2) the domain of the integration is not known.

G.Wahba described a function that minimizes the equation 3.1 and 3.2 combined with each other. She calls it "thin plate spline". Later on she has worked on the formula of "thin plate spline" and she introduced the spherical spline. The main purpose of the spherical spline interpolation was to describe the weather conditions at a height of 500 millibar above the sea level. The 500 millibar is measured (with error) at a large number $n$ of weather stations distributed around the world. It is desired to find a smooth function $u = u(\theta, \Phi)$ defined on the
surface of the earth ($\theta = \text{latitude}, \Phi = \text{longitude}$) which is an estimate of the 500 millibar height at position ($\theta, \Phi$). The formula of the spherical spline interpolation $u_{n,m,\lambda}$ is a minimizer of the equation 3.1 and 3.2 combined with each other. The function $u_{n,m,\lambda}$ is the same as equation 2.14 in chapter two, namely

$$U_{n,m,\lambda} = c_0 + \sum_{i=1}^{N_c} c_i r(\cos(E_i,E_j))$$  \hspace{1cm} 3.3

Only the calculation of the $c_i$'s are different because of the use of the $\lambda$ parameter in the matrix. The calculation of the $c_i$'s are as follow,

$$\begin{bmatrix} (R + \lambda I).C + T.c_0 = Z \\ T'.C = 0 \end{bmatrix}$$ 3.4

with $Z$ the measured potentials, $T$ a vector of ones, $R$ is a matrix with the cosines of the angle values of the positions between the electrodes and the parameter $\lambda$ must be between zero and one ($0 \leq \lambda < 1$). For the calculation of the $R$ matrix and for further explanation of the spherical spline interpolation see paragraphs 2.4 and 2.5.

The value $\lambda$ is used as a tuning parameter to minimize the function,

$$\frac{1}{N_s} \sum_{k=1}^{N_c} (U^k_{n,m,\lambda} - z_k)^2$$ 3.5

with $z_k$ as the measured value, $u_{n,m,\lambda}$ as the predicted (interpolated) value and $N_s$ is the number of the electrodes.

Equation 3.5 is called the residual (in some literature also residual variance). This function represents the error between the measured value and the predicted value. It is difficult to calculate the parameter $\lambda$ and $m$ from equation 3.5. It is easy to choose the $\lambda$ and the order $m$ is assumed to be known. Using the parameter $\lambda$ as a tuning parameter in equation 3.5 we get a costfunction. The costfunction can have two basic form, namely

$$\begin{array}{ll}
\text{Figure A} & \text{Figure B} \\
\text{Figure 3.3} & \text{Costfunction.}
\end{array}$$
- **Figure A** looks like a parabolic function. The function begins to descend by increasing the $\lambda$ till the point $\lambda_{\text{min}}$ and after the point $\lambda_{\text{min}}$ the function ascends with increasing $\lambda$. The case of figure A can occur when the difference between measured value and predicted value is big.

- **Figure B** is an ascending function. By increasing the parameter $\lambda$ the costfunction will also increase. Figure B can happen when the predicted value is almost the same as the measured value (error is almost zero).

From figure A we can see that by using $\lambda_{\text{min}}$ the spherical spline approximation can be optimized, because at $\lambda_{\text{min}}$ the costfunction shows a minimum. Using a $\lambda$ value in the case of figure B can only deteriorate the spherical spline approximation, because by increasing the $\lambda$ value the error between measured and predicted value will also increased. The best results from the interpolations is when no smoothing parameter $\lambda$ is used ($\lambda = 0$).

The next step is to find a searching algorithm that can find the minimum in the residual function. There are several techniques for finding the minimum in a function. In the next paragraph more theory will be given on this subject.

### 3.2.1 Numerical search method

Searching for the minimum in function $F(x)$ is the same as finding the square root in the first derivative of function $F(x)$ ($F'(x) = 0$). Sometimes the derivative is too complex to find. Then an algorithm need to be written that can find the minimum without the calculation of the first derivative. There are several methods for searching for the minimum, some of which are [4]

- Parabolic Interpolation
- Brent’s method
- Golden Section

Parabolic Interpolation and Brent’s method are better for smooth functions [4]. If the function is nicely parabolic near to the minimum then with use of a parabola fitted through three points in the function can approach in a single leap to the minimum, or at least very near to it. A Golden Section search is more designed to handle, in effect, the worst possible case of function minimization. Golden Section can be used for one-dimensional minimization (minimize a function of one variable) without calculation of the derivative.

In the case of the costfunction calculated with the residual function we do not have a nice parabolic and smooth function, so it is better to use the Golden Section search method.
Golden search works as the same as bisection search (interval halving method). Bisection is a numerical method for finding the root of a function.

![Figure 3.4 Finding root with bisection](image)

If $F$ is continuous in the interval $[a,b]$ and if $F(a) \times F(b) < 0$ than $F$ has at least one root in $[a,b]$. With bisection the root $\alpha$ can be found easily. Let

$$m = \frac{a + b}{2}$$

then there are three possibilities:

- $F(m) = 0$; The root $\alpha$ is found
- $F(a) \times F(m) < 0$; Then $b = m$ and repeat the process
- $F(a) \times F(m) > 0$; Then $a = m$ and repeat the process

To find the root in a function, the function needs to be bracketed in a pair of points, say $a$ and $b$, when the function has opposite signs at those two points. For finding the minimum the function also needs to be bracketed in a pair of points $[\theta_1, \theta_2]$. One has to be sure that the minimum lies between the interval $[\theta_1, \theta_2]$. Suppose that there is a minimum at $\theta^*$, then is $\theta_1 < \theta^* < \theta_2$ and $f(\theta_1) > f(\theta^*) < f(\theta_2)$.

The analogue of bisection is to choose a new point $x$, either between $a$ and $m$ or between $m$ and $b$. The value of $f(x)$ need to be evaluated. Golden Section algorithm calculates two value beside the values given for the interval. With these two value ($x$ and $y$) the function $f(x)$ and $f(y)$ is calculated. Depend of values of $f(x)$ and $f(y)$ a choice will be made.

Calculation of the new points and choosing of the minimum with Golden Section is a form of reducing the uncertainty surrounding the location of the minimum by reducing the interval in which the minimum is located. This reduction can be done by the Golden Section algorithm. With each iteration the algorithm reduces the interval by affixed value $k$, with

$$k = \frac{-1 + \sqrt{5}}{2} = 0.618033988.$$  

This number $k$ satisfies the relation $1/k = 1 + k$, which once important in ancient Greek architecture. The Golden Section method proceeds as follows.
Brain mapping  

Calculation of the smoothing parameter

Select an interval \([\theta_1, \theta_2]\) with \(\theta_1 < \theta^* < \theta_2\). Two new points \(\theta_3\) and \(\theta_4\) will be calculated, with \(0.5 < k < 1\):

\[
\begin{align*}
\theta_3 &= k\theta_1 + (1-k)\theta_2 \\
\theta_4 &= k\theta_2 + (1-k)\theta_1
\end{align*}
\]

The numbers \(k\) and \((1-k)\) are called the golden ratios. Now, compare the values of \(F(\theta_3)\) and \(F(\theta_4)\).

- If \(F(\theta_3) > F(\theta_4)\): \(\theta_2 = \theta_3\); \(\theta_1 = \theta_3\); \(\theta_3 = \theta_4\) and \(\theta_4\) is calculated according to 3.7.
- If \(F(\theta_3) < F(\theta_4)\): \(\theta_1 = \theta_3\); \(\theta_2 = \theta_4\); \(\theta_4 = \theta_3\) and \(\theta_3\) is calculated according to 3.6.

Once an appropriate initial interval has been found, the Golden Section method always converges to the minimum. This iterative procedure is continued until the interval \([\theta_1, \theta_2]\) has become less than a predefined small value \(\epsilon\).

Figure 3.5 is used for an explanation. Suppose an interval is given between \([1,2]\) (look figure 3.5). Point 3 and 4 will be calculated with the formulas given in 3.6 and 3.7. Suppose \(F(3) < F(4)\), then the interval is reduced to \([1,4]\) and new point \(4^*\) is at place 3 and another point three will be calculated according to formula 3.6. Suppose the new point \(3^*\) is at point 5 and \(F(3^*) > F(4^*)\). Then new interval become now between \([5,4]\) and new point \(3^{**}\) is at point 3 and a new point \(4^{**}\) is calculated with the formula given in 3.7. This will go on till the distance between the two utmost values \([1,2]\) is less than a given tolerant value.

![Figure 3.5](image-url)

Figure 3.5: Successive bracketing of a minimum. The minimum is originally bracket by points \([1,2]\). The 3 and 4 will be calculated according to the formulas given in 3.6 and 3.7. The function will evaluate \(F(3)\) and \(F(4)\). Suppose \(F(3) < F(4)\) then the interval will be reduced till \([1,4]\). The new point 4 is shifted to point 3. And a new point \(3^*\) is calculated and the position of the new point \(3^*\) is at point 5. And so the comparison will go on till the interval will be reduced to a small value.
3.2.2 Golden Section with Fibonacci ratios

Instead of using the golden ratio values, the values of the Fibonacci sequence can also be used to calculate the reduction $k$ factor of the Golden Section method. The convergence speed of the Golden Section method can be improved if the reduction factor $k$ iteratively is depend. If this dependency is selected according to the Fibonacci sequence an improvement can be made in the interval reduction. The Fibonacci sequence is defined by

$$a_i = a_{i-1} + a_{i-2}$$  

with $a_0 = a_1 = 1$. This sequence looks like 1, 1, 2, 3, 5, 8, 13, 21, .... The reduction factor $k_i$ in iteration $i$ becomes,

$$k_i = \frac{a_i}{a_{i+1}}$$

For large values of $k_i$, this ratio becomes the same as the golden section number $k_\infty = 0.618033988...$. The reduction factor $k_i$ must begin at $0.666667$ (2/3) because the factor has to be between $0.5 < k < 1$. In formula,

$$k_2 = \frac{a_2}{a_3} = \frac{2}{3}$$

3.3 The algorithm

The algorithm for the smoothing parameter $\lambda$ is based on the Golden Section method. With the formula in 3.5 $\lambda_{\text{min}}$ will be estimated. For the estimation of $\lambda$ on the basis of the data, splines are computed on the basis of $N_s \cdot J$ electrodes, while the omitted electrode delivers a test value. The estimation error is defined as the difference between the predicted and the measured potential on each omitted electrode position. Minimizing the function given in formula 3.5 will provide an optimal $\lambda$.

First the interval for $\lambda$-values is defined within which the searching will take place. Definition of the interval is needed for the Golden Section method. Suppose the interval is defined between $[\lambda_1, \lambda_2]$. The algorithm for the estimation of the smoothing parameter $\lambda$ will be explained on basis of the following steps,

1) All measured values are read at the user defined time

2) With a given interval $[\lambda_1, \lambda_2]$ two more $\lambda$'s ($\lambda_3$ and $\lambda_4$) are calculated according to the formulas given in 3.6 and 3.7
3) These two \( \lambda \)'s (\( \lambda_3 \) and \( \lambda_4 \)) are used in the function given in 3.5 to calculate the residual, say \( R(\lambda_3) \) and \( R(\lambda_4) \). The residual function represents the average estimated error over the electrodes. Each electrode will be omitted once to calculate the error between the predicted value and the measured value. This will be done with the two \( \lambda \)'s, \( \lambda_3 \) and \( \lambda_4 \) in the spherical spline interpolation formula given in 3.3 and 3.4.

According to the assumption of the Golden Section method, if \( R(\lambda_3) < R(\lambda_4) \) then the interval will be adjusted and a new \( \lambda_3 \) will be calculated according to (3.6). The new reduced interval will be \([\lambda_1, \lambda_4]\) (\( \lambda_1 = \lambda_2; \lambda_2 = \lambda_4; \lambda_4 = \lambda_3 \) and \( \lambda_3 \) is calculated). An advantage of the Golden Section method is that only the new point needed to be calculated and the function value of the new point. The rest can be shifted into each other. This is easy to put in an iterative procedure.

For \( R(\lambda_3) \geq R(\lambda_4) \) the same procedure can be done. But now the new interval becomes \([\lambda_3, \lambda_2]\) and point \( \lambda_4 \) and the function value of point \( \lambda_4, R(\lambda_4) \) need to be calculated and the rest of the points can be shifted into each other.

4) Step 3 will repeat until the difference between the two utmost points of the interval is smaller than a predefined tolerance value, \( \delta \). When the difference is smaller than \( \delta \) \( | \lambda_2 - \lambda_1 | < \delta \) step 3 will finish, the stop criterion.

5) After step 4, if \( R(\lambda_3) < R(\lambda_4) \) then is \( \lambda_{\min} = \lambda_3 \) else \( \lambda_{\min} = \lambda_4 \).

This algorithm converges always to a minimum. Even if the costfunction is ascending this algorithm will give \( \lambda_{\min} \) equal to zero (\( \lambda_{\min} = 0 \)). The algorithm will be tested with golden ratios and also with ratios calculated with Fibonacci sequence.

In the program the interval will be defined between \([0, 0.1]\). This is based form former research done in [2] and also during testing of the program with data acquired at the University of Tilburg. The smoothing parameter \( \lambda \) will decrease by increasing the order \( m \) of the spline function.
4. **The extended brain mapping program**

4.1 Introduction

The extended brain mapping program has the same structure as the old one (the program in [6]). The difference between the extended brain mapping program and the one described in [6] is that the extended brain mapping program has more possibilities. The possibilities of the extended program are,

- Plotting of Scalp Potential (SP) maps.
- Plotting of Scalp Current Density (SCD) maps.
- The spherical spline interpolation can be calculated with order $m$ ranging from two to five ($2 \leq m \leq 5$).
- Possibility to estimate a smoothing parameter $\lambda$ and including it to the spherical spline interpolation.

The demands for the extended program remain the same as it was for the former brain mapping program described in [6]. The program is able to do the following things,

* Read electrode positions from an electrode file and data from an EEG recording file in European Data Format (EDF).
* Calculate the brain maps (SP or SCD) with spherical spline interpolation.
* Possibility to estimate a smoothing parameter $\lambda$ if the user wants.
* That the user can browse through the recording (only on a brain map).
* Varying the browsing steps, this allowing the user to browse more quickly through the entire file.
* Display brain maps in colours, where the colours indicate the electrical potential or current on each place on the scalp.
* Ability to switch the electrodes on or off by clicking on the keyboard.
* Ability to switch an indication of the contours of the potential or current. (a stepped map, so one can see the deeper and higher regions in potentials or current field distribution).

The description of the headers of an EDF-file can be found in appendix A.

The program is written in **Microsoft Visual C++**, edition 3.1. At this moment only a DOS version is available, but in the future the extended brain mapping program can run also under Windows environment.
4.2 Calculations

A great part of the extended brain mapping program consists of calculations of the spline functions. The way the calculations are programmed can cause delay in the program for example if there is a lot of multiplication or lot of cosine and sines functions (standard functions in math.h). The program is made with less multiplication as possible.

One of the most time consuming calculations is the calculation of the \( r_m(p, p_i) \) and \( h_m(p, p_i) \) function, see formula 2.13 and 2.16 in chapter two. To increase the speed according to [8] is by making of two tables in which \( r(p, p_i) \) and \( h(p, p_i) \) are precalculated for each \( p \) and \( p_i \). The tables contain the results of the \( r \) and \( h \) equation as given in equation 2.13 and 2.16 for all possible angles between \( p \) and \( p_i \).

Calculation of the smoothing parameter can also be optimalized in time. From former researches we can see that the minimum lambda is most of the time less than 0.1 for all order \( m \), so the interval that is given for the golden search routine can be between 0 and 0.1 \([0, 0.1]\), if the optimal \( \lambda \) lies outside the interval the program will give an error message). The larger the interval is the more time it needs to find the minimum.

Solving of the matrix equation 2.5 is also part of the spherical spline. This matrix equation has to be solved for each measurement. In [8] a way is described to increase the speed of the solving of the matrix. The matrix is filled with values that only depend on the electrode positions. The matrix is solved to the measured values. If the matrix could be prepared to be more easily solvable this could increase the speed. This is possible with the so called LU decomposition. The LU decomposition decomposes a matrix in two triangular matrices, namely

- \( L \) inferior triangular matrix,
- \( U \) superior triangular matrix.

These two matrices are faster to solve then the original matrix. The advantage of decomposition is that it is independent from the measured values, so one decomposition is enough for an entire set of measurements, as long as the same electrodes set-up is used.

4.3 The output

The output of the program is a plot of a map on a specified time that the user want. The screen plays an important role because the output of the program will appear on the screen with all explanations of the keys of the keyboard next to the brain map picture. Interpolation was the main subject for the creation of the brain map picture.

The image is displayed in 256 colours with 253 for the brain map. The colours for the brain map are light blue to dark blue, black and light green to dark green. Dark blue is for the lowest value in the brain map plot and dark green is for highest value in the brain map plot.
A bar will appear on the screen with the colours that are displayed in brain map picture with the value of the highest and lowest value on that moment. The values will change if the user change the time. See figures 4.1 and 4.2 for examples of scalp potential and scalp current density maps with and without smoothing factor. The background of the screen is red and light red is used to display the electrodes positions on the scalp and grey is used as the text colour.

A small screen is introduced for a brain map image in corner of the screen for search purpose, figure 4.3. This small image can be calculated very fast. With this time is gained when the user want to make a plot on a specify time. For the difference between stepped and smooth transient picture see figure 4.4.
Brain mapping

The extended brain mapping program

Figure 4.1a: Unsmoothed Scalp Potential field ($\lambda = 0$)

Figure 4.1b: Smoothed Scalp Potential field ($\lambda = 6.27 \times 10^{-6}$)
Figure 4.2a: Unsmoothed Scalp Current Density field ($\lambda = 0$)

Figure 4.2b: Smoothed Scalp Current Density field ($\lambda = 6.27 \times 10^{-6}$)
Figure 4.3: Scalp Potential field with small screen (smooth colour gradient)

Figure 4.4: Scalp Potential field with stepped colour gradient and without electrodes indication
5. Measurements and results

5.1 Introduction

The usefulness of the brain mapping program is based on accuracy and speed of the program. The accuracy of the brain mapping program is cannot be determined from the visual output of Scalp Potentials field or Scalp Current Density estimation fields. Therefore we need a measure for the accuracy of the estimated fields. In paragraph 5.2 we present results of several accuracy measures.

The speed of program is determined by time measurements on two different processors. A 80486 DX running on 66 Mhz and a processor running on 75 Mhz which are presented in paragraph 5.3.

5.2 Accuracy estimation

The accuracy of the estimated fields is determined by the goodness of fit between predicted values based on the spline and corresponding measured values. Predicted potentials can be calculated of any position on the head. Since measured potentials are only known at the electrode positions we have to obtain the predicted values here as well. This implies that the observed electrode must be left out for the computation of the spline so the potential at this position can be predicted by interpolation. The program is tested for two different types of data, namely single trial data and grand averaged data often test persons. The data are obtained by a measurement set-up of 23 electrodes at a sample rate of 128 Hz.

5.2.1 Accuracy estimation without smoothing

A (spherical) spline without smoothing acts as an interpolator. There are several statistical ways to calculate the accuracy of an interpolation. We will test the accuracy with the following methods,

- **Residual variance**
- **Evaluations** (cross validation),
  * normalized residual $nrv$
  * prediction covariance $cv$
  * residual covariance $rcv$
Each of these measures is described in paragraph 2.6. In figure 5.1 and 5.2 the residual variance (equation 2.20) of nine electrodes are shown which are representative for the complete set of electrodes. The interpolated values are calculated for different order $m$ of the spline. For the position of the nine electrodes see figure 1.2 in paragraph 1.4.

Figure 5.1: Residual variances of the averaged data for different order $m$.

Figure 5.2: Residual variances of the single trial data for different order $m$.

From figures 5.1 and 5.2 we see that residual variances depend on the recording and the order of the spline function. The residual variances of the single trial data are a lot higher then those of the averaged data. With order $m = 2$ we get the smallest residual variance values for the single trial data. With averaged data it is difficult to say at which order $m$ we get the smallest residual variance values.

Another method to analyse the goodness-of-fit is using the evaluations theory, see paragraph 2.6.2. In tables 5.1 and 5.2 the measures $nrv$, $cv$ and $rcv$ are given for different orders $m$ and in figures 5.3 and 5.4 the graphs of these results are shown.

<table>
<thead>
<tr>
<th>$m$</th>
<th>$nrv$</th>
<th>$cv$</th>
<th>$rcv$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.062892</td>
<td>0.969114</td>
<td>0.418454</td>
</tr>
<tr>
<td>3</td>
<td>0.051427</td>
<td>0.974019</td>
<td>0.277042</td>
</tr>
<tr>
<td>4</td>
<td>0.054363</td>
<td>0.972456</td>
<td>0.208918</td>
</tr>
<tr>
<td>5</td>
<td>0.060103</td>
<td>0.969637</td>
<td>0.175527</td>
</tr>
</tbody>
</table>

Table 5.1: Results of the calculations with the averaged data.

Figure 5.3: Graph of the $nrv$, $cv$ and $rcv$. 
Brain mapping

| m  | nr
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>cv</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.283696</td>
<td>0.849036</td>
</tr>
<tr>
<td>3</td>
<td>0.375333</td>
<td>0.809968</td>
</tr>
<tr>
<td>4</td>
<td>0.479846</td>
<td>0.771340</td>
</tr>
<tr>
<td>5</td>
<td>0.586268</td>
<td>0.737111</td>
</tr>
</tbody>
</table>

Table 5.2: Results of the calculations with the single trial data

From the results shown in tables 5.1 and 5.2 and figure 5.3 and 5.4 it can be seen that for the averaged data file the unsmoothed spherical spline gives better results than for single trial data. For averaged data is the prediction covariance almost one for all orders ($cv > 0.95$), the normalized residual ($rcv$) is smaller than 0.1 for all orders and the overestimation value (residual covariance) is also small compared to single trial data. For single trial data we get smaller values for the $cv$ and the value of the $rcv$ is also higher compared to averaged data. With order $m$ equal to three the interpolation is best for the averaged data, because the inaccuracy $nr = 0.051427$ is small compared to the rest and also the prediction covariance $cv = 0.974019$ is the nearest to one compared to the other values calculated with different orders. Only the overestimation $rcv$ is larger compared to the other $rcv$. The best results for the single trial data are obtained with order two. In this case the highest $cv$ is reached ($cv = 0.849036$) and the inaccuracy $nr$ is the smallest.

### 5.2.2 Accuracy estimation with smoothing

Accuracy estimations are done on the same two data files but now with a smoothed spherical spline. The smoothing parameter $\lambda$ is estimated with the residual equation 3.5 given in paragraph 3.2. The parameter $\lambda$ depends on the measured data and the order $m$. The residual variance of a smoothed spline is compared with an unsmoothed ($\lambda = 0$) spline and the results are shown in figure 5.5 and figure 5.6. This is done for several values for the orders $m$ and on averaged data and single trial data. The results of the averaged data are in figure 5.5 and those of the single trial data are in figure 5.6.
The smoothing parameter $\lambda$ has almost no influence on the residual variance for averaged data, except by higher order $m$ values ($m \geq 5$). At $m = 5$ is the residual variance reduced a little bit.

For the values of $\lambda$'s used for the estimation of the smoothed splines at different order $m$ see table 5.3. The values of the $\lambda$'s was estimated with the program.

<table>
<thead>
<tr>
<th>$m$</th>
<th>$\lambda$</th>
<th>nrv</th>
<th>cv</th>
<th>rcv</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.0066915</td>
<td>0.113495</td>
<td>0.943863</td>
<td>0.50615</td>
</tr>
<tr>
<td>3</td>
<td>4.90579.10^{-2}</td>
<td>0.064071</td>
<td>0.968214</td>
<td>0.395727</td>
</tr>
<tr>
<td>4</td>
<td>1.13842.10^{-2}</td>
<td>0.050225</td>
<td>0.972609</td>
<td>0.264311</td>
</tr>
<tr>
<td>5</td>
<td>9.25609.10^{-10}</td>
<td>0.049826</td>
<td>0.984776</td>
<td>0.239762</td>
</tr>
</tbody>
</table>

Table 5.3: Evaluation results of the averaged data

From the results of the evaluation methods we see that the use of a smoothing parameter did not contribute to an improvement of the spline function when applied to averaged data. The $cv$ values becomes smaller compared to when no smoothing parameter is used.
From the results of the residual variance in figure 5.7 we can conclude that the smoothing parameter $\lambda$ has more influence on the residual variance. For each order $m$, the residual variance is reduced. At the order five, the residual variance for all nine electrodes is reduced with 50%.

From the results in figure 5.5 and 5.7 we can conclude that the extended brain mapping program works better on single trial data files than on averaged data files when smoothing is included. For the values of the $\lambda$'s parameter used in the estimation of the smoothed spline, see table 5.4.

<table>
<thead>
<tr>
<th>$m$</th>
<th>$\lambda$</th>
<th>$nrv$</th>
<th>$cv$</th>
<th>$rev$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.00172209</td>
<td>0.260905</td>
<td>0.860222</td>
<td>0.458889</td>
</tr>
<tr>
<td>3</td>
<td>5.4406.10^-5</td>
<td>0.258128</td>
<td>0.862531</td>
<td>0.426386</td>
</tr>
<tr>
<td>4</td>
<td>1.26253.10^-6</td>
<td>0.243563</td>
<td>0.870234</td>
<td>0.439743</td>
</tr>
<tr>
<td>5</td>
<td>1.02652.10^-8</td>
<td>0.241949</td>
<td>0.877387</td>
<td>0.426334</td>
</tr>
</tbody>
</table>

Table 5.4: Evaluations results on the single trial data

The evaluation methods show also an improvement when a smoothing parameter $\lambda$ is used. The prediction covariance $cv$ calculated for $m = 2, 3, 4$ and $5$ shows an increasing compared to unsmoothed splines (table 5.5).
A better way to see that the use of a smoothing parameter improves the estimation is by showing the measured signal, the unsmoothed and the smoothed signal. In figure 5.9 and figure 5.10 the measured and predicted signal of electrode Cz taken on the single trial data are shown.

Figure 5.9: The measured (Mea) and unsmooth ($\lambda = 0$) predicted (Pred) signal of electrode Cz calculated with order $m = 5$ (single trial data)

Figure 5.10: The measured and smooth ($\lambda = 1.02652 \times 10^{-4}$) predicted signal of electrode Cz calculated with order $m = 5$ (single trial data)
When the signal is smoothed the interpolated signal is almost the same as the measured signal.

The same is done on averaged data with order \( m = 5 \). In figure 5.11 and 5.12 the graphs are shown.

![Figure 5.10: The measured and unsmooth \((\lambda = 0)\) predicted signal of electrode Cz calculated with order \( m = 5 \) (averaged data)](image)

![Figure 5.12: The measured and smooth \((\lambda = 9.256.10^{-10})\) predicted signal of electrode Cz calculated with order \( m = 5 \) (averaged data)](image)

From the graphs shown in figures 5.10 and 5.12 we can conclude that there is not an improvement by using a smoothing parameter. Sometimes the smoothing parameter has deteriorated the predicted signal. The error between measured and predicted signal becomes larger compared to when no smoothing parameter was used. The cause of this is that the smoothing parameter is calculated from the averaged error over all the electrodes. If the error between the interpolated and measured signal is small or even zero on one electrode and for the rest of the electrodes we have a large error, than can the error on that one electrode be blown up by the smoothing parameter. And for the rest of the electrodes can the error be decreased. Smoothing minimizes the averaged residual variance over all electrodes. However for an individual electrode the residual variance may be higher than without smoothing.
5.3 Time calculations

The parts of the extended brain mapping program that are very time consuming are the calculation of the spline functions, estimation of the smoothing parameter $\lambda$ and the graphical output. Time measurements are done using a built in function that returns the clock time of the processor. The measurements are done with an electrode set-up of 23 electrodes and a data file which contain a recording of six seconds at a sample rate of 128 Hz. In table 5.5 we can see the results of these measurements.

<table>
<thead>
<tr>
<th>Type processor</th>
<th>40486 DX 66Mhz</th>
<th>Pentium 75Mhz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reading inputs</td>
<td>0.06 seconds</td>
<td>&lt;&lt; 0.01 seconds</td>
</tr>
<tr>
<td>Initialisation</td>
<td>2.47 seconds</td>
<td>1.87 seconds</td>
</tr>
<tr>
<td>Estimation of $\lambda$</td>
<td>55.64 seconds</td>
<td>24.82 seconds</td>
</tr>
<tr>
<td>Creating matrix and LU decomposing</td>
<td>0.05 seconds</td>
<td>&lt;&lt; 0.01 seconds</td>
</tr>
<tr>
<td>Put legend on screen</td>
<td>0.17 seconds</td>
<td>&lt;&lt; 0.01 seconds</td>
</tr>
<tr>
<td>Read measure values, calculate spline and put image on screen</td>
<td>5.16 seconds</td>
<td>2.04 seconds</td>
</tr>
<tr>
<td><strong>Total time of 1 image complete run</strong></td>
<td><strong>63.49 seconds</strong></td>
<td><strong>28.29 seconds</strong></td>
</tr>
</tbody>
</table>

Table 5.5: Time measurements on the extended brain map program.

The initialisation part consists of reading of the electrode file and the header of the data file. Creating matrix and LU decomposing consists of creating the matrix with the angles values between the electrodes and decomposing of the matrix in a $L$ and $U$ matrix. A complete run consists of all parts mentioned above in table 5.5. The smoothing parameter $\lambda$ is estimated with the residual formula 3.5 given in paragraph 3.2.
6. Conclusions and recommendations

The program “Brainmap” is now extended with new possibilities. Beside Scalp Potentials maps (SP), the user can plot Scalp Current Density estimation maps (SCD) and include a smoothing parameter $\lambda$ for the spline. The structure of the program and the output images are the same as the old program described in [6].

Estimation of the smoothing parameter
- From the time measurements done on the program (Paragraph 5.2) we may conclude that the most consuming part of the program is the estimation of the smoothing parameter $\lambda$. The cause of this delay is that the program needs to go through all the measurement values of every electrode in the given time-window. For every electrode the program calculates the error between measured and interpolated value. From the averaged error a smoothing parameter is estimated. If the time window is large it will cost a lot of time to estimate the smoothing parameter.

Calculation of SCD and SP maps
- Calculation of SCD maps is the same as the calculation of SP maps. Both make use of spherical splines interpolation and both can be smoothed. The time to calculate SCD maps is almost the same as SP maps.

Effect of smoothing
- Smoothing has larger effect on single trials then on averaged data. The error between interpolated and measured signals is smaller on averaged data than on single trials data for all orders $m$. Taking averaged of the data is also a manner to smooth the data. If we look at the signal of electrode Cz of the averaged data (figure 5.10), it is more smoother than the signal of the same electrode of the single trial data (figure 5.9).

Smoothing and estimation errors
- Smoothing results in smaller estimation errors for splines with orders 3, 4 and 5. For order $m$ equal to 2 is the reduction of the residual variance smaller compare to higher orders and the error is not reduced for all electrodes.

Order $m$ and estimation errors
- The higher the order $m$ the smaller the error between the interpolated and measured signal will be on averaged data. On single trials data the error will increase when the order $m$ increases.
Dependence on smoothing parameter $\lambda$

- The smoothing parameter $\lambda$ is dependent on the data and the order $m$ of the spline functions. The estimated smoothing parameter $\lambda$ changes in value when the order $m$ changes and the sample number or time remains the same.

Recommendations

Time window
- It is recommended to do research on where the time-window will be defined. Now the user can define randomly the time-window where the smoothing parameter $\lambda$ is to be estimated. There are periods in the recording where there is a lot of activity and where there is not so much activity. It can happen that the user defines a time-window where the estimated smoothing parameter $\lambda$ has no influence on the interpolation. It is also possible that with the same data file on another time (sample number) the program estimates a smoothing parameter that reduces the residual variance. There are techniques of how and where the time-window can be defined [18].

Numerical minimalization
- The number of steps to find the smoothing parameter $\lambda$ corresponding to the minimum point of the cost function is dependent of the initialisation (the first given interval, see paragraph 3.). A large interval gives a large number of steps.

Computer
- A computer with a Pentium processor is recommended to prevent inconvenient working time for research purposes or to analyse patients in a hospital or laboratory.


Reference

   *Topografische afbeeldingen van hersenpotentialen.*

   *Performances of surface Laplacian estimators: A study of simulated and real scalp potential distributions.*
   Brain Topography, Volume 8, Number 1, 1995: pag. 35-45

   *Artificially high coherences result from using spherical spline computation of scalp current density.*

[4] Bosch P.P.J van den, Klauw A.C van der
   *Modeling, Identification and Simulation of Dynamical Systems*
   CRC Press, 1994

[5] Broek J van den
   *Brain Mapping II*
   M. Sc. thesis Eindhoven University of Technology, The Netherlands, 1992

   *Brain mapping with EEG signals.*

   *Scalp potential and current density mapping with an enhanced spherical spline interpolation.*
   Medical progress through technology, 1994, pag. 23-30

   *A simple format for exchange of digitized polygraphic recordings.*
   Electroencephalography and clinical neurophysiology, 1992, pag. 391-393

    *Brain Mapping of EEG and Evoked Potentials.*
    Springer-Verslag 1989

    *Scalp current density: concept and properties.*
    Electroencephalography and clinical Neurophysiology, 1988, Volume 69: pag. 385-389
<table>
<thead>
<tr>
<th>Reference</th>
<th>Author(s)</th>
<th>Title</th>
<th>Journal</th>
<th>Volume/Number/Pages</th>
</tr>
</thead>
</table>
Appendix A

European Data Format (EDF)

HEADER RECORD

<table>
<thead>
<tr>
<th>Field Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 ascii</td>
<td>version of this data format (0)</td>
</tr>
<tr>
<td>80 ascii</td>
<td>local patient identification</td>
</tr>
<tr>
<td>80 ascii</td>
<td>local record identification</td>
</tr>
<tr>
<td>8 ascii</td>
<td>start date of recording (dd.mm.yy)</td>
</tr>
<tr>
<td>8 ascii</td>
<td>start time of recording (hh.mm.ss)</td>
</tr>
<tr>
<td>8 ascii</td>
<td>number of bytes in header record</td>
</tr>
<tr>
<td>44 ascii</td>
<td>reserved (used by the Tilburg University for local variables)</td>
</tr>
<tr>
<td>8 ascii</td>
<td>number of data records (-1 if unknown)</td>
</tr>
<tr>
<td>8 ascii</td>
<td>duration of a data record in seconds</td>
</tr>
<tr>
<td>4 ascii</td>
<td>number of signals (ns) in data record</td>
</tr>
<tr>
<td>ns * 16 ascii</td>
<td>ns * label (e.g. EEG FpzCz or Body temp)</td>
</tr>
<tr>
<td>ns * 80 ascii</td>
<td>ns * transducer type (e.g. AgAgCl electrode)</td>
</tr>
<tr>
<td>ns * 8 ascii</td>
<td>ns * physical dimension (e.g. µV or °C)</td>
</tr>
<tr>
<td>ns * 8 ascii</td>
<td>ns * physical minimum (e.g. -500 or 34)</td>
</tr>
<tr>
<td>ns * 8 ascii</td>
<td>ns * physical maximum (e.g. 500 or 40)</td>
</tr>
<tr>
<td>ns * 8 ascii</td>
<td>ns * digital minimum (e.g. -2048)</td>
</tr>
<tr>
<td>ns * 8 ascii</td>
<td>ns * digital maximum (e.g. 2047)</td>
</tr>
<tr>
<td>ns * 80 ascii</td>
<td>ns * prefiltering (e.g. HP:0.1Hz LP:75Hz)</td>
</tr>
<tr>
<td>ns * 8 ascii</td>
<td>ns * nr of samples in each data record</td>
</tr>
<tr>
<td>ns * 32 ascii</td>
<td>ns * reserved (used by the Tilburg University for local variables)</td>
</tr>
</tbody>
</table>

DATA RECORD

<table>
<thead>
<tr>
<th>Field Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nr of samples[1] * integer</td>
<td>first signal in the data record</td>
</tr>
<tr>
<td>nr of samples[2] * integer</td>
<td>second signal</td>
</tr>
<tr>
<td>....</td>
<td>....</td>
</tr>
<tr>
<td>nr of samples[ns] * integer</td>
<td>last signal</td>
</tr>
</tbody>
</table>

Detailed digital format of the header record (upper block, ascii's only) and of each subsequent data record (lower block, 2-byte integer only). Each of the ns signals is characterized separately in the header.

ASCII, American Standard for Information Interchange. This code declares the binary values of the main characters of the alphabet, digits and punctuation marks.
Appendix B

User’s Reference

The user’s reference is a guidance through the extended brain mapping program. It helps the users working with the program, for example how the program start, what are the inputs, etc.

The main program is **BRAINMAP.EXE**. It creates images of the brain activity (Scalp Potentials or Scalp Current Density maps) measured on the scalp. To run the program it needs an input file named **BRAINDAT.INI**. The syntax is:

```
brainmap braindat.ini.
```

**BRAINDAT.INI**

The file BRAINDAT.INI contains the name of the electrode file and data file, order value m, time sample window that will be used for the calculation of the smoothing factor, answer to question “Smoothed or unsmoothed spline?”

Y is for a smoothed spline and N for unsmoothed spline. Further the user has to specify what kind of map he wants to see “SP or SCD”.

The user can adjust the braindat.ini file in a windows environment. An example of the braindat.ini file is as follow,

```
electrodefile = name.elc
datafile = name.rec
time_window = 0 2000
ss_order = 2
question_smooth = Y
question_map = SP
```

**Electrode positions file**

The brain mapping program needs an electrode position file to know how the electrodes are placed on the scalp. The electrode file is a very simple file in ASCII\(^1\) format. The electrode position file contains the number of the electrodes at the first row, names of the electrodes, latitude angle in radial (first value) and azimuth angle (second value) also in radial. At the bottom of the file is space for comments. The comments are preceded with a *.

\(^{1}\)ASCII, American Standard Code for Information Interchange
Measure data file
The measured data file contains the digitally stored measure values of the EEG signal. The measured values had been taken placed over a time period on a specified sample frequency. The format of the data file is in European Data Format (EDF). A data file of this format consists of two part, a header and a data part. The header contains the information about the patient, like name of the patients, recording date, time recording and information about the data.
The data part contains the measured potentials. See Appendix A for an example of an EDF.

Time window
The calculation of the smoothing parameter is achieved in a time window declare by the user. The user must give the start time and the stop time [in seconds]. The increment is given by:

Time increment = \frac{1}{f_{\text{sample}}}

Spline order
The user must define the order \( m \) of the spline. The order \( m \) is ranging from 2 to 5.

Smoothing question
The user can give if he wants to estimate a smoothing parameter. The letter \( Y \) is for a smoothed spline and \( N \) for unsmoothed spline.
Map_question

The user must also specify what kind of map he wants to see. He can choose between Scalp potential map and Source Current Density map. SP is for Scalp potential and SCD is for Scalp Current Density.

The output of the program is an image of scalp potential or scalp current density. This will be a circular image. Underneath is a legend that indicates the potential or current that the different colours represents [\(\mu A/m^2\) or \(\mu V\)]. At the right side the user can see the name of the type of map, the time at which the image is taken, as the time step and the value of the smoothing parameter \(\lambda\). Further there is also a small key reference. The key reference remains the same as in [6].

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>→</td>
<td>Increase the time at which recording is viewed with the amount indicated by the timestep. As soon as this key is pressed, a small brain map appears to increase the speed with which can be searched in the data file.</td>
</tr>
<tr>
<td>←</td>
<td>Decrease the time at which recording is viewed with the amount indicated by the timestep. As this key is pressed, a small brain map appears to increase the speed with which can be searched in the data file.</td>
</tr>
<tr>
<td>↑</td>
<td>Increase the timestep with a factor 10. The default timestep is 1 second. The maximum timestep is 100 seconds</td>
</tr>
<tr>
<td>↓</td>
<td>Decrease the timestep with a factor 10. The minimum timestep is 1/100 of a second</td>
</tr>
<tr>
<td>PgUp</td>
<td>Use a small image. This image is placed in the upper left corner over the large brain map. The time that is indicated by the program refers to the time in the recording of the small image</td>
</tr>
<tr>
<td>PgDn</td>
<td>Create a large brain map at the same time in the recording as the small brain map is made.</td>
</tr>
<tr>
<td>T</td>
<td>This will set the timestep to one sample. The time will increase with one sample. timestep = 1/ sample frequency</td>
</tr>
<tr>
<td>Home</td>
<td>Go to the first sample in the data file (time = 0)</td>
</tr>
<tr>
<td>End</td>
<td>Go to the last sample in the data file (time = last time)</td>
</tr>
<tr>
<td>F1</td>
<td>This switch the indication of the electrode positions in the brain map off.</td>
</tr>
<tr>
<td>F2</td>
<td>This switch the indication of the electrode positions in the brain map on.</td>
</tr>
<tr>
<td>F3</td>
<td>This set the image colours to a smooth transient of light blue to black to light green. This is the most accurate version of the brain map</td>
</tr>
<tr>
<td>F4</td>
<td>This set the image colours to a stepped transient of light blue to black to light green. There will be 18 colours which will enhance the contours of the maxima and minima.</td>
</tr>
<tr>
<td>Esc</td>
<td>Ends the program and returns to DOS</td>
</tr>
</tbody>
</table>
Error messages

Most of the error messages remains the same as in [6].

*Error locating data memory (200) and error locating image memory (200)*
This error indicates an error on memory management in the program.

*Error opening ini file (10)*
Error with the braindat.ini file, probably caused by incorrect name.

*No electrode file is read in ini file (10)*
No electrode file was given in the braindat.ini file.

*Error opening electrode file (10)*
Error with the electrode file. This probably caused by incorrect file name.

*Error opening data file (10)*
Error with the data file. This probably caused by incorrect file name.

*No initialization for time window is read in ini file (10)*
No time windows was given in the braindat.ini file.

*No order for spherical spline interpolation is read in the ini file (10)*
The order is not given in the braindat.ini file.

*Input order m is not valid (10)*
The order m must lie between two and five (2 ≤ m ≤ 5).

*No correct answer to the question_smoothing (10)*
Only the letter Y or N is usable by this question.

*No correct answer to the question_map (10)*
SP or SCD are the only answer to this question.

*Order m must be greater than 2 for SCD (10)*
The order m must be greater than two for the calculation of SCD maps.

*No floating point unit*
Program is unable to locate the mathematical coprocessor. This is needed for the calculations.

*Not possible to open video mode. Make sure VESA drivers are loaded (1)*
The program has problem with the screen. The computer card of the computer should be VESA compatible or VESA drivers should be loaded.
The names of the EEG channels do not match. This can also be caused by caps/non-caps differences (11)

The names of the EEG channels in the electrode file don’t match with those in data file.

The number of EEG channels does not match (11)

The numbers of EEG channels in the electrode file does not match with the number of electrode in the data file.

Programmer’s reference

In this reference several subroutines for the BRAINMAP.EXE are described. Some of the subroutines remains the same as it was in [6]. The brain map program consists of three part, the main program BRAINMAP.CPP and two header files containing subroutines and class definitions, namely BRAINMAP.H and GRAPHOUT.H. In brainmap.h the initialisation and the calculation takes places. Initialisation is definition of the classes, reading of the electrode positions file, reading of the data file and calculation of the table r_m(p,p_j). From the information about electrode position the L and U matrix will be calculated. In graphout.h the matrix is solved and the spline is also calculated based on the data that was read at the moment. The image calculation happens also in this header file. The calculation of each pixel in an image size makes use of the of the spherical spline.

BRAINMAP.H

Class definitions

class subheader
Classes for the definitions of a structure for the channel information. The class contains,
- Number of channel in data file.
- Calibration gain factor value.
- Calibration offset for the values.
- Sample frequency of the channel.
- The number of samples of the channel.
- Starting position of the data of the channel in the data file.

This class is used to define an array that has a depth that equals the number of electrodes. The data in this array is only accessible by the subroutines readheader, readvalues, samplefreq and calculation_estimated_value.

class mainheader
This class is for the general data information. It contains,
- Size of the header of the data file.
- Number of the data records.
- Recording duration.
- Number of signals.

These data are only accessible by the subroutines readheader and sampleduration.
**class electrodes**
This class contains information on the electrodes. It contains
- Name electrodes.
  - Latitude, Azimuth and cosines of azimuth of the electrodes.
  - $x$ and $y$ co-ordinate of the electrode.
This array has a depth equals to the amount of electrodes. The data in this array is only accessible by the subroutines `electrode_position`, `angle_one_omitted_electrode`, `readheader`, `create_imageSCD`, `create_imageSP` and `calculation_estimated_value`.

**class qtable**
This class is for the table $g(x)$. It contains precalculated values. The array has a depth of 2001. The values in the table is only accessible by the subroutines $g_x$, $g$, `create_imageSP` and `calculation_estimated_value`.

**class htable**
This class is for the table $h(x)$. It contains precalculated values. The array has a depth of 2001. The values in the table is only accessible by the subroutines $h_x$, $h$ and `create_imageSCD`.

The following pages describes in alphabetical order the subroutines used in the program “brainmap”

**angle**
**Description**
calculates the angle between two electrodes and the middle of the sphere

```
#include "brainmap.h"

double angle(int electrodenum1, int electrodenum2)
```

<table>
<thead>
<tr>
<th>electrodenum1</th>
<th>number of electrode</th>
</tr>
</thead>
<tbody>
<tr>
<td>electrodenum2</td>
<td>number of electrode</td>
</tr>
</tbody>
</table>

**Return**
angle value

**angle_one_omitted_electrode**
**Description**
calculates the angle between two electrode and the middle of the sphere while one electrode is omitted

```
#include "brainmap.h"

double angle_one_omitted_electrode(int electrodenum1, int electrodenum2, int omitted_electrode )
```

<table>
<thead>
<tr>
<th>electrodenum1</th>
<th>number of electrode one</th>
</tr>
</thead>
<tbody>
<tr>
<td>electrodenum2</td>
<td>number of electrode two</td>
</tr>
<tr>
<td>omitted_electrode</td>
<td>number of the electrode that is omitted</td>
</tr>
</tbody>
</table>
Return angle value

calculation_estimated_value
Description estimates the value of the measured potential of an omitted electrode

```
#include "brainmap.h"

void calculation_estimated_value(double *measure values, int
omitted_electrode, double lambda, double result[1])

measure_values array containing the measured potential values
omitted_electrode number of the omitted electrode
lambda smoothing parameter \( \lambda \)
result[1] array containing the estimated value of the omitted electrode
```

costfunction
Description calculates the values of the costfunction (equation 3.5)

```
brainmap.cpp

double costfunction(double *measure values, double lambda)

measure_values array containing the measured potential values
lambda smoothing parameter \( \lambda \)
```

Return costfunction value corresponding to the lambda

create_imageSCD
Description calculates the spline based on the measured values.

```
#include "graphout.h"

void create_imageSCD(unsigned char *sized_image, double
*measured_values, int picture_size, double extreme[2])

sized_image size image on the screen
measured_values array containing the measured potential values
picture_size size of the circular image (SCD map)
extreme[2] array containing the highest and lowest value in the SCD brain map
see also create_imageSP

create_imageSP
Description Calculates the spline based on the measured values.

```c
#include "graphout.h"

void create_imageSP(unsigned char *sized_image, double *measured_values, int picture_size, double extreme[2])
```

- sized_image: size image on the screen
- measured_values: array containing the measured potential values
- picture_size: size of the circular image (SP map)
- extreme[2]: array containing the highest and lowest value in the SP brain map

create_L_u
Description creates the L and U matrix based on the electrode positions

```c
#include "brainmap.h"

void create_L_u(double lambda)
```

- lambda: smoothing parameter λ

current_graph
Description Draws the scalp current density map

```c
brainmap.cpp

void current_graph(double *measure_values, int picture_size, int electrodes, int map_indication)
```

- measure_values: array containing measured potential values
- picture_size: size of the circular image (map)
- electrodes: indication for displaying of electrodes
- map_indication: indication what kind of map (SP/SCD)

see also voltage_graph
display_electrodes
Description displays the electrodes on the map

```c
#include "graphout.h"

void display_electrodes(int picturesize)
picturesize size of the circular image (map)
```

double2string
Description converts a double to a string

```c
#include "graphout.h"

inline void double2string(double number, char *string)

number value to be converted
string the converted value in a string
```

see also dtoa
dtoa
Description converts a double to a string

```c
#include "graphout.h"

void dtoa(double floattobeconverted, char *returnvalue)

floattobeconverted value to be converted
returnvalue the converted value in a string
```
electrodename
Description reads the name of the given electrode

```c
#include "brainmap.h"

cchar* electrodename(int number_electrode)
number_electrode number of the given electrode

Return name of the given electrode
```
**electrode_position**

**Description**
reads the latitude and azimuth value of the given electrode number

```c
#include "brainmap.h"

void electrode_position(double &latitude, double &azimuth, int number)
```

number number of an electrode

**Return**
Latitude and azimuth value of the given electrode number

**findmin**

**Description**
finds minimum in a function with golden section (golden ratios) method

brainmap.cpp

```c
double findmin(double *measure_values, double ax, double cx)
```

measure_values array with the measured potential values
ax first value of the interval [ax, cx]

```c
cx last value of the interval [ax, cx]
```

**Return**
input value (λ) corresponding to the minimum point of the function

see also **findmin_with_fib**

**findmin_with_fib**
The same as **findmin** but now with Fibonacci values in the Golden section.

**gx**

**Description**
calculates equation 2.13 and put it in an array

```c
#include "brainmap.h"

void gx(int ss_order)
```

```c
ss_order order value of the spline
```

**Remark**
this subroutine is called in the beginning of the program. The subroutine `double g(double input_value)` returns the value of `gx` (-1 ≤ input_value ≤ 1)

see also **hx**
hx
Description calculates equation 2.16 and put it in an array

#include "brainmap.h"

void hx(int ss_order)

ss_order order value of the spline

Remark this subroutine is called in the beginning of the program. The subroutine \texttt{double h(double input\_value)} returns the value of \texttt{hx} \((-1 \leq \text{input\_value} \leq 1)\)

\begin{verbatim}
init\_electrode\_positions
Description Initializes the electrode information file.

#include "brainmap.h"

void init\_electrode\_positions(char filename[])

filename[] file containing information about the electrode file
\end{verbatim}

\begin{verbatim}
readheader
Description reads the header and subheader of the data file.

#include "brainmap.h"

void readheader(FILE *datafile, subheader *channelinfo)

datafile handle of the data file
channelinfo pointer to the array that will contain the electrode information
\end{verbatim}

\begin{verbatim}
readvalues
Description reads the data of all electrodes at the given time

#include "brainmap.h"

void readvalues(double *measure\_values, double time, FILE *datafile, subheader *channelinfo)

measure\_values pointer where the measured values should be stored
time time in the recording
\end{verbatim}
datafile
channelinfo
handle of the data file
pointer to a array containing the channel information

sampleduration
Description reads sample duration in the data file

#include “brainmap.h”

long unsigned int sampleduration(void)

subroutine accepts no inputs

Return sample duration of the data

samplefrequency
Description reads sample frequency in the data file

#include “brainmap.h”

double samplefrequency(subheader *channel_data)

channel_data pointer to the array containing channel information

Return the sample frequency of the data file

set_palet_blue_green_red
Description sets the pallet.

#include “graphout.h”

void set_palet_blue_green_red(int colorres)

colorres colour resolution.

Remark Colour resolution determines the step in which the colour changes. If colorres = 1 than the colours will change smoothly.
**voltage_graph**

**Description**

draws the scalp potential maps

brainmap.cpp

```c
void voltage_graph(double *measure_values, int picture_size,
int electrodes, int map_indication)
```

- `measure_values`: array containing measured potential values
- `picture_size`: size of the circular image (map)
- `electrodes`: indication for displaying of electrodes
- `map_indication`: indication what kind of map (SP/SCD)