

Realistic Source Models of ERP Data

John E. Richards

September, 2005; modified May, 2006.

(Note: Some of the information in this document is copied from journal articles published by John E. Richards or grant applications, and may not be identified as such in the paper. This material has copyright privileges by the author and should not be copied from this paper. The author claims copyright privileges on all parts of this paper. Some material is quoted verbatim from other sources and that material is indicated with quotes and references).

The purpose of this document is to detail the development of realistic ECD source models of ERP data. The electroencephalogram (EEG) consists of a varying electrical potential recorded on the scalp. The source of the EEG potential are thought to be changes in post-synaptic electrical potentials occurring during cortical activity. These source are identified with cortical source analysis (brain electrical source analysis, equivalent current dipole analysis; Huizenga & Molenaar, 1994; Scherg, 1990, 1992; Scherg & Picton, 1991). “Equivalent Current Dipole” (ECD) analysis hypothesizes a cortical dipole or set of dipoles, calculates the forward solution, compares the forward solution with the empirical data, and adjusts the dipole(s) until there is a satisfactory (or minimized) fit between the forward solution and the empirical data. The forward model is typically estimated with a “three sphere” model consisting of the scalp, skull, and brain. However, this model is a poor representation of the head electrical currents, and realistic models of the head as the basis for the ECD sources is needed. This document details an approach to that analysis.

The realistic source modeling cannot be done easily. The default parameter for computer programs are a 1-shell uniform sphere or a multi-shell sphere (3: brain, skull, scalp; 4: brain, CSF, skull, scalp). Models using realistic parameters (BEM; FEM) are hard to do and their computer implementation (e.g., BESA, EMSE) are slow and hard to do. I have created a C++ program for doing the forward model calculations, and a realistic source model C++ program that is flexible, works in batch mode, and will do several types of models.

The following is an overview of the use of ECD models for the identification of cortical sources of EEG and ERP using realistic current source models.

Scalp Electrical Activity and Cortical Source Analysis

The electroencephalogram (EEG) consists of varying electrical potential recorded on the scalp. The EEG potential changes are caused by post-synaptic electrical potentials occurring during cortical activity. Thus, the EEG changes are a direct reflection of neural electrical activity. The EEG by itself (and event-related EEG, “event-related-potential”, ERP) is not a good indicator of where in the cortex the signal is generated. However, quantitative tools have been developed to attempt to identify the source of the cortical activity, i.e., cortical source analysis. Figures in the accompanying ppt file show the EGI electrode system, the recording equipment and other EEG indices. One criticism of EEG techniques is that the spatial resolution of the cortical source localization techniques is only about 5 cm, and this resolution is poor (see spatial vs temporal resolution in ppt file, and 5 cm resolution on MRI).

There are several types of cortical source analysis. In the current model I will be emphasizing “equivalent current dipole” (ECD) analysis. This analysis hypothesizes a cortical dipole with

location and current moment. This dipole then is used to generate a current map on the scalp that would be consistent with the location/moment of the dipole, compare the computed current map with the empirical current, and adjust the parameters of the dipole to provide a close fit between the computed and empirical current maps. Other cortical source methods use extended areas of activation, other computation methods, and so for.

Equivalent current dipole (ECD) analysis uses several steps. First, an electrical dipole is hypothesized to exist in the head. Electrical dipoles have a location vector (coronal, saggital, axial) and moment vector (coronal, saggital, axial). The location vector describes the location of the dipole inside the conducting medium. The moment vector is a direction vector whose direction indicates the direction of current flow and whose amplitude indicates the current level. The initial values of the location and moment vectors are set to some a priori location (seeded) or randomly placed in the head. In early models the source locations were typically assumed to occur in a spherical shell for computational simplicity (see next paragraph), but recent quantitative models allow the dipole locations to exist in realistic head locations or be limited to cortical regions or to single locations.

The second step in ECD analysis is to calculate a current map on the scalp. This is done by calculating a current for each recording electrode. There is a specific formula for the conduction of current from a dipole to an electrode, and includes the distance from the dipole to the electrode, the moment amplitude and its angle to the electrode, and the conductivity (resistivity) of the medium between the dipole and the electrode. The current map on the entire scalp is called the “forward model”. In the earliest use of ECD analysis, forward models were done with a single sphere because of the simplicity of the formula for current conduction in a spherical medium (i.e., distance, uniformity, non-anisotropy). However, the spherical model(s) can only approximate the current on a real head, so the inverse problem for the spherical models is exacerbated. More recent models have used multiple spherical shells (3-shell model, 4-shell model) or realistic head models for individual participants. Figures in the ppt file show the algorithm for the ECD analysis along with a 1-sphere shell model, a 3-sphere model, and a realistic shell model.

One can distinguish two parts to the “forward model”. The first part is to identify the source locations. The second part is to identify the conductivity ($1 / \text{resistance}$) for the model. The “source locations” (or “source volume”) is either a location in one of the spheres of the shell model, or realistic locations in the head. The shell models can make a uniform shell for the outside of the head (radius of 1, or radius of average radius of electrodes). The one-shell source volume restricts the source to anywhere inside the spherical shell. Figures in the ppt file show one-shell models. The multi-shell model restricts the source to anywhere inside the inner shell (brain). These models are typically chosen because only the electrode positions are required. Figures in the ppt file show multi-shell models. The realistic source locations identify the sources as being in specific places in the head, i.e., gray matter, specific anatomical locations, or other. These models require anatomical MRIs and head segmenting. Figures in the ppt file show realistic models for gray matter, eyes, and 3fr files for these. The issue of identifying source locations is covered in more detail in a later section.

Second, the conductivity of the model must be specified for the forward model. The shell models use an assumption of uniform media from the source to the electrode, and assign a value for conductivity based on that media’s known conductance. For one-shell models, this is typically the value for gray matter (i.e., the shell is the brain), but may be some other value (perhaps based on the average from a realistic conductivity model). For multi-shell models, the resistance is based

on the known values for the shells, i.e., gray matter, skull, and scalp for the three-shell model, and added CSF for the four-shell model. Dipole source models that include a shell source volume and a shell conductance model are easy to calculate and the analytic values are based on the known properties of current flowing through spherical volumes. The conductivity values for the shells may be substituted with the conductivity of each element in the path and assigns the path a conductivity based on the proportional volumes of each element. “Boundary element models” (BEM) use simplified versions of the inner structure, identifying only the boundaries of the brain, CSF, skull, and scalp and calculating the values based on these boundaries. “Finite element models” (FEM) use detailed three-dimension wireframes with conductivity values for each type of material in each element. The construct a forward model of transfer coefficients based on the conductivity of each tetrahedral element in the wireframe that takes into account the volume conductor as a whole. These models are difficult to use because they are difficult to calculate, and cannot be done online in a reasonable manner. Thus, offline calculations of the forward model must be done for each element of the realistic source model.

The third step of ECD analysis is to compare the computed current map with the empirical current and adjust the parameters of the dipole to provide a close fit between the computed and empirical current maps. The comparison between the forward model and the empirical current is done with typical multidimensional fit indices (correlation; chi-squared; RMS difference; residual). All of these fit indices use the electrodes as p dependent variables. Chi-squared, RMS difference, and residual use a sum-of-squared differences between the forward model and empirical current, which may be normalized by the empirical current variance (i.e., chi-squared, residual). The ppt file shows the algorithm steps in figural form. Additionally, the successive steps for an ECD analysis of eye movement ICA is shown, and the residual map is also shown. The quantitative methods used to adjust the values for the dipole location and moment vectors are discussed in a later section of the paper.

The final step of ECD analysis is to accept the minimized model and output the model to appropriate data files or viewing programs. The minimization routines have an intrinsic comparison device that compares the fit index on successive parameter changes and stops when a small value difference between parameters is found. The final set of parameters is the best or acceptable estimable parameters for the model. At this point the forward model and empirical data may be compared with the fit indices (correlation, chi-squared, residual).

Do Realistic Models Make a Difference?

Do realistic models result in differences? Probably. Some studies are cited in the Slotnick chapter in the Handy book reviewing the difference found with the different types of dipole models. It is concluded that some advantage accrues from the very simplest models, but the BEM models fare as well as FEM models in this approach. I have done some tests in EMSE using 1-shell, 3-shell (source, forward model), 3-shell forward model with realistic sources. There was a marked advantage in these tests for the 3-shell forward model with realistic sources. I also have done extensive comparison of 1-shell, 4-shell, and realistic source models (both resistance models and source volume models) and conclude that the realistic models do show an advantage over the other types of models (see Model Comparison). A number of studies have been done comparing shell, BEM, and FEM models (e.g., Ary, Klein, & Fender, 1981; Awada, Jackson, Williams, Wilton, Baumann, & Papanicolaou, 1997; Benar & Gotman, 2002; Buchner, Waberski, Fuchs, Wischmann, Wagner, & Drenckhahn, 1995; Haueisen, Tuch, Ramon, Schimpf,

Wedeen, George, & Belliveau, 2002; Huiskamp, Vroeijsstijn, van Dijk, Wieneke, van Huffelen, 1999; Kavanagh, Darcey, Lehmann, & Fender, 1978; Marin, Guerin, Baillet, Garnero, & Meunier, 1998; Meijs, Weier, Peters, & van Oosterom, 1989; Roth, Balish, Gorbach, & Sato, 1993; Thevenet, Bertrand, Perrin, Dumont, & Pernier, 1991; Vanrumste, Van Hoey, Van de Walle, D'Have, Lemahieu, & Boon, 2000; Yvert, Bertrand, Echallier, & Pernier, 1995; Yvert, Bertrand, Echallier, & Pernier, 1996; Yvert, Bertrand, Thevenet, Echallier, & Pernier, 1997).

The use of realistic source volumes may be especially important for infants. The accurate and realistic identification of head topography may be extremely important for infant participants. "Interpretation of the underlying physiology and pathophysiology is complicated by the presence of the fontanels and sutures in the skull. The anterior and posterior fontanels are present at the bregma and the lambda. The skull is not present within the fontanels. The brain below the fontanels is protected by the dura, which is thicker within the opening. The fontanels and sutures are squeezed together during delivery, but become larger during the first months and then eventually close. The anterior fontanel may be large enough to admit an adult's thumb. The posterior fontanel is closed within the first 2 month. The anterior fontanel stays open as long as 16 months or, in abnormal cases, even for several years. The sutures can be quite wide near the fontanels. The mean width of the coronal and lambdoidal sutures at their midpositions is 3–4 mm for infants between 0 and 60 days after birth (Eramie and Ringertz, 1976). The sutures may stay unfused for several years (Hansman, 1966). EEG signals may be profoundly affected by the fontanels and sutures, which are effectively skull defects, since they represent areas of high conductivity relative to the skull. Volume currents might be affected by these openings in the skull. Thus, it is necessary to evaluate the effect of these skull openings on EEG in order to accurately infer physiological functions in human infants." (Fleming, Wang, Caprihan, Eiselt, Haeisen, & Okada, 2005).

"A significant advance in the use of the cortical source analysis in infant participants will be made only by obtaining MRIs for individual participants who also are tested in the experimental procedures. The best source analysis technique uses anatomical data from individual participants (i.e. anatomical MRI) and perhaps combines the ERP with other measures of functional brain activity (PET, MRI; e.g. Martinez *et al.*, 1999; also cf. Richards, 2003b with Richards, 2002). Obtaining anatomical MRIs for infant participants would be helpful and would be reasonable, if somewhat impractical (Patterson, Badridze, Flax, Liu & Benasich, 2004). This information could be used to constrain the ECD solutions with reasonable anatomical areas, estimate scalp, skull, CSF, and cortical thickness and topography for the numerical aspects of the ECD analysis, and identify the cortical areas of the ECD locations in the MRI of the participant rather than translating infant locations to Talairach locations. Such an enterprise would be helpful to the scientific community by developing a 'Talairach coordinate system' relevant to infant participants, perhaps over a range of ages. The possibility of relating anatomical MRI and ERP to other measures of functional brain activity (PET, fMRI) will be extremely difficult to apply to infant participants." (Richards, 2005)

"In comparison, EEG signals are significantly distorted by skull defects that are unique to the human neonates. The fontanels are present at the midline junctions of the bregma and lambda. The skull is not present within the fontanels; instead the brain is protected by a thick dura filling the windows. They are small during the delivery, but become larger in the first several months, up to about 3-4 cm along the coronal suture, and then eventually they close. The anterior fontanel may be large enough to admit an adult's thumb. The unclosed sutures can be quite wide near the

fontanels. The mean width of the coronal and lambdoidal sutures at their midpositions is 3-4 mm for infants between 0 and 60 days after birth. In abnormal cases such as hydrocephalous, the sutures may change its width with the development of the disease and become as wide as 10 mm or more. Moreover, the sutures do not close for many years. The earliest age for complete closing of the sagittal and coronal sutures are 6 and 11 years, respectively. Thus, EEG signals may be profoundly affected by the fontanels and sutures since they create paths of low conductivity for the volume currents in the brain and funnels the currents through these openings in the skull.” (Section from http://www.tristantech.com/pdf/BabySQUID_description_v6.pdf)

Identifying the brain structures that have resistances and generate dipoles

The calculation of the forward model requires the identification of source locations in the head. The shell models that use a spherical shell for both the dipole resistance values and the dipole sources restrict the source to anywhere inside the spherical shells and assume a uniform resistance value in the medium for the entire shell. The realistic source locations identify the sources as being in specific places in the head, i.e., gray matter, specific anatomical locations, or other. These models require anatomical MRIs and head segmenting.

The first task in the forward solution is to identify the brain structures that may generate dipoles and those that have resistance. This is done by segmenting the MRI into its various components. In other documents I have discussed using the MR Viewer to segment the head into scalp, skull, CSF, eyes, eye muscle, gray matter, white matter. The gray matter consists of neuron bodies that are known to be electrical current sources, i.e., can have dipoles. In addition, the eyes may have dipoles to explain eye movement electrical activity recorded on the scalp. Each of the head components segmented above also have conductivity (1/resistance) that has been characterized by others.

The segmented head structures are represented in MRViewer as “Region” (rgn) files. These region files can be used as the basis for FEM (3-d) wireframe models, 3fr. These wireframe models have the coordinates of individual elements (coronal sagittal axial) and the indices for tetrahedral (4 coordinates, conductivity, and FEM-type). The wireframes are made with “decimation” indices, of about 4 (for 64-voxel tetrahedral) and 12, 16, 20, or 32 on in order to get about 90K coordinates and 200K tetrahedral. (e.g., S0006 2/8 had 2M FEM; 3/12 had 1M FEM; 4/12 had 404K FEM; 4/16 and 4/20 didn't work; 4/32 had 87k Coordinates and 400k FEM).

There are two types of 3fr files that may be useful here. First, a whole head 3fr would consist of the segments for the entire head. This would include segmenting of gray matter, white matter, CSF, skull, scalp, eyes, eye muscles (around eyes), nasal and throat cavity, and muscles below skull. This whole head 3fr would be used to generate realistic FEM conductivities. It also would be used as a general model for identifying all portions of the head. Second, individual 3fr files might consist of specific areas of the brain (or head—e.g., eyes) that would be used as source locations in a dipole analysis. A typical 3fr for this would be the gray matter, or eyes, or cortical tissue (gray matter + eyes). Other 3fr would be specific areas of the brain identified by the Talairach daemon locations (“Gray matter in anterior cingulate.3fr”, “Extrastriate occipital cortex.3fr”, middle temporal gyrus.3fr). These specific areas would be used as the coordinates for a source volume model where the dipole sources would be allowed only in the designated area. Figures in the ppt file show some models for eye movement analysis, as well as source volume models for the whole head.

Once the brain structures have been generated in the whole head model, two types of forward models may be generated. First, a forward model can be generated from the “four-sphere resistance”, “realistic source volume” model. This calculates the paths through the four spheres for each element of the source volume of interest. This forward model is done with a C++ computer program. This program finds any eye element (FEM type 10) or gray matter (FEM type 4) and computes the conductivity from the source to each electrode. This is done by calculating in a four-shell sphere the distance from the source to each electrode, and the portion of that distance occupied by the gray matter, CSF, skull, and scalp. The length of the vector from the dipole to the electrode is calculated for each of the four spheres and the combined conductivity value is saved. This results in a file of source conductance values, with each vector consisting of 0 to 9 sum volumes for the FEM types, the index of the source in the CLabTetra array, and the electrode number (0 to 127). The resulting source conductances and the FEM model are used later in the analysis of the ECDs for a “four sphere conductance” “realistic sources” model. The realistic source model for the four-sphere resistance model could be computed online, but the calculations take a small but finite period of time and would add substantially to computing time for the source estimation. Another estimate of conductivity that improves upon the shell conductivities is to trace a path along spherical coordinates from the source to the electrode, finding any FEM tetrahedral that have an intersection with this path, and keeping the sum of all volumes for those FEM tetrahedral type for the source-electrode combination. This conductivity may be substituted for the shell conductivities, but still uses the assumptions for the forward model based on the spherical head model. This results in a file of source conductance values, with each vector consisting of volumes for the FEM types, the index of the source in the CLabTetra array, and the electrode number (0 to 127). The resulting source conductances and the FEM model are used later in the analysis of the ECDs. This computer program is very time-consuming. For example, using two “screamers”, and three very fast computers, about 2000 sources per hour can be done (e.g., 30K to 50K sources, about 20 hours). I also have used 20 desktop computers and three very fast computers and done the computations in about 6 hours (about 5000 sources per hour; massive parallel computing using separate workstations). Figures in the ppt file show the results from one such analysis, with the current paths from the dipole to the electrode represented, and the head medium types indicated by coloring the FEM elements in the path.

The second type of forward model is for the BEM or FEM models. For example, the FEM models calculate a set of coefficients for each tetrahedral that represent the influence on that element on the electrical potential at the electrodes. These coefficients take into account the conductivity of any direct paths between the tetrahedral element and any electrode, indirect paths based on local influences, and anisotropies in the conducting volume. These models should be ideal for the unique conductivity and topographical characteristics of the infant skull, and may partially account for the current leakage at fontanelles and unsutured skull junctions. Currently these forward models may be calculated with the BESA or EMSE analysis programs.

Restricting Dipole Locations

The “forward model” or “forward solution” is tractable given a specific set of model ideas, especially for realistic head models. The “inverse model” or “inverse solution” in ECD analysis has a serious (but not intractable) difficulty. The “inverse problem” in cortical source analysis is the impossibility of finding a unique and unambiguous equivalent current dipole. This is because more than one dipole configuration will provide comparable fit-index levels, because adding

extra dipoles always should improve fit, and because the minimization techniques have local minima that may not result in the best fit. The ECD analysis is sensitive to the initial estimates of dipole location and moment, and the number of dipoles chosen for the analysis. The “equivalent current dipole” model itself may be ill-conceived, especially with spherical shell models, resulting in ephemeral results that are not replicable. The inverse problem is serious, but not overwhelming.

There are several approaches to the inverse problem in ECD analysis. First, an accurate and scientifically reasonable ECD solution is aided by advance knowledge of the empirical data. For example, reasonable cortical locations may be found from prior studies in the scientific literature from MRI, fMRI, PET, invasive animal studies, ERP-cortical source studies, and so forth. Given advance knowledge of the possible solutions, the initial location of the hypothesized dipole might be “seeded” with this advance knowledge. Such seeded locations could come from comparable (or paired-parallel) fMRI studies of the same individuals. Using a realistic source volume with individual MRIs greatly enhances the chances of finding an appropriate seed location (or locations) for the dipole model. An additional restriction may be placed upon the model by fixing the initial location of the dipole from advance knowledge and estimating only the moments of the dipoles to get fit estimates. ECD models with acceptable fit index level are evidence of successful model testing. Accurate and scientifically reasonable ECD solutions remove the inherent instability from the inverse estimation procedure and make this an a priori testing procedure.

Second, characteristics about the ERP data may be inherent in a prior selection process. For example, I (Richards, 2003, 2005; Reynolds & Richards, 2005) use independent component analysis to find component loading patterns represent coherent spatial-temporal activity in the EEG. The components are clustered according to activity and topographical similarity, and the locations of the clusters may be obvious from the topographical characteristics or the experimental-related activity (eye movements; post-target-onset extrastriate activity). The ICA analysis also may result in components that have coherent single-dipole sources which are correctly recovered with ECD analysis (Richards, 2004).

An example of this is given in the ppt file. A recent use of ICA has applied it to single-trial EEG data, extracted components, and estimated equivalent current dipoles from the resulting component weights (DeLorme, Makeig, Fabre-Thorpe & Sejnowski, 2002; Jung *et al.*, 2000b, 2001b; Reynolds & Richards, 2005; Richards, 2002, 2004, 2005). An independent component analysis (ICA) was done following the procedures outlined by Makeig, Sejnowski, and their associates (DeLorme *et al.*, 2002; Jung *et al.*, 2001a; Makeig, Bell, Jung & Sejnowski, 1996; Makeig, Jung, Bell, Ghahremani & Sejnowski, 1997; also see Richards, 2004a). Independent component analysis is done on the segmented raw ERP data (not averaged data). This results in component loadings and component activations. The component loadings represent coherent patterns of spatial location that were related in the temporal ERP data. The component weights resulting from the ICA represent the topographical information in the EEG and are similar to a set of topographic scalp maps (Johnson, de Haan, Oliver, Smith, Hatzakis, Tucker & Csibra, 2001; Jung, Makeig, Humphries, Lee, McKeown, Iragui & Sejnowski, 2000a; Jung, Makeig, Lee, McKeown, Brown, Bell & Sejnowski, 2000b; Jung, Makeig, McKeown, Bell, Lee & Sejnowski, 2001a; Jung, Makeig, Westerfield, Townsend, Courchesne & Sejnowski, 2001b; Richards, 2002, 2004a). The activations represent the activity of the ERP filtered by the component loadings. Figures in the ppt file show a single ICA component with loadings around

the eyes, in a single individual and as a cluster of multiple individuals. The activations are also shown around the eye movement, for all subjects, individual participants, and for a single participant. The activity at the onset of the eye movement indicates that the set of electrodes with high loadings in the ICA components (plotted on topographical map) were coherently active during the period immediately following the eye movement.

Third, “regions of interest” may be specified in cortical source models. These regions of interest may come from parallel MRI-neuroimaging sources (fMRI study), or may simply be relevant anatomical locations from the MRI. These regions might be a specific Brodmann, gyrus, or anatomical area, whose location in the MRI space is specified by Talairach stereotaxic coordinates. The dipoles search is then restricted to the region of interest by using realistic source volume models and defined 3fr files. The “regions of interest” may be used with ECD models that restrict the location of the ECDs to the selected source volume. Alternatively, regional dipole models could estimate electrical potential changes throughout the selected region for the test fit (i.e., smaller moments, same moment direction, multiple locations in the region of interest).

One region of interest approach that is particularly appealing is the “Talairach Regions” approach (Richards, 2002). The Talairach space is a mapping of Brodmann areas and anatomical locations (gyri, nuclei) onto the brain of an individual person. This provides a stereotaxic atlas of locations in the brain that can be transformed back and forth from the Talairach space to individual participant MRIs. The Talairach Daemon has taken the Talairach designations and assigned individual coordinates into lobe, tissue, anatomy (gyrus, nuclei), Brodmann areas, and hemisphere (Fox & Uecker, 2005; Lancaster, Summerlin, Rainey, Freitas, & Fox, 1997; Lancaster, Woldorff, Parsons, Liotti, Freitas, Rainey, Kochunov, Nickerson, Mikiten, Fox, Freitas, Rainey, Kochunov, Nickerson, Mikiten, & Fox, 2000). One can read all of the coordinates from the Talairach Daemon for a particular area (e.g., anterior cingulate cortex; or Brodmann area 24) and warp the Talairach coordinates into a participant’s MRI coordinates, and then create a 3fr wireframe file with the coordinates in that participant’s MRI space. This wireframe file may then be used as a “source volume” or “region of interest” to place dipoles in cortical source analysis.

Tissue Conductivity

The analysis of cortical sources of scalp-recorded electrical potential requires accurate knowledge of the conductance of the media in the head (Awada, Jackson, Baumann, Williams, Winton, Fink, & Prasky, 1998; Haueisen, Ramon, Eiselt, Brauer, & Nowak, 1997; Stok, 1987). This might be especially true for infant participants, for two reasons. First, the conductivity of the scalp and skull may be extremely different in infants and adults. The skull is less dense and thinner. The density of the skull is the primary factor in determining its conductivity, so that infant skulls will be far less resistive than adults. The thickness of the skull also will influence conductance, but this can be handled by spherical or realistic models with different levels of thickness. However, there are changes in skull thickness that will affect skull resistivity. “The conductivity of the skull is also expected to change with age and across individuals. The skull thickness increases rapidly within the first three years of age from about 2 mm at birth for term newborns to about 5 mm at 3 years of age for boys, then the growth slows down. The skull thickness reaches a mean of 8.3 mm and 9.5 mm for 25 year-old women and men, respectively. These changes in skull thickness are associated with thickening of the dense, poorly-conducting

inner and outer bony tables of the skull relative to the spongy middle layer containing blood and with a decrease in effective conductivity of the skull with age. Also important is the variability in skull thickness. The 10th-90th percentiles are 2.4-4.6 mm and 3.0-4.9 mm for 1-1/2 year old girls and boys. That is, the range is 50-60% of the means. At the age of 25, the 10th-90th percentile range is 40-50% of the means for women and men.”

(http://www.tristantech.com/pdf/BabySQUID_description_v6.pdf).

There have been several studies that have measured conductivity of the media in the head. Many of these studies used autopsy samples and there are several compilations of these values (Malmivuo & Plonsey, 1995; Gabriel, Gabriel, & Corthoput, 1996; Gabriel, Lau, & Gabriel, 1996a, 1996b; Baumann, Wonzy, Kelly, & Meno, 1997; Burger & van Milaan, 1943; Foster & Schwan, 1989; Geddes & Baker, 1967). Tables 1 to 3 show three of these for several elements of the human body. An interesting new technique is “electrical impedance tomography”, which places an electrical current in the head and measures the impedance at other places on the head is being used to estimate conductivity of the total head space or of specific tissues (Akhtari, Bryant, Mamelak, Flynn, Heller, Shih, Mandelkern, Matlachov, Ranken, Best, DiMauro, Lee, & Sutherling, 2002; Goncalves, de Munck, & Verunt, 2001; Oostendorp, Delbeke, & Stegeman, 2000; van Burik & Peters, 2000). These values may be used in cortical source analysis. Table 4 shows the values used by the EMSE computer program (Source Signal, Inc). Table 5 shows a compilation of these values. The conductivity values for human tissue has also become an interesting topic because the specific absorption rate of tissue, dependent on conductivity and electromagnetic frequency, is important in determining the parameters of MRI neuroimaging.

Comparing forward and empirical data: Minimization, fitting, optimization

The third step of ECD analysis is to compare the computed current map with the empirical current and adjust the parameters of the dipole to provide a close fit between the computed and empirical current maps. The comparison between the forward model and the empirical current is done with typical multidimensional fit indices (correlation; chi-squared; RMS difference; residual). All of these fit indices use the electrodes as p dependent variables. Chi-squared, RMS difference, and residual use a sum-of-squared differences between the forward model and empirical current, which may be normalized by the empirical current variance (i.e., chi-squared, residual). One of these fit indices is chosen and used in minimization techniques to change the values of the location vector and moment vector (dipole “free”), or just the moment vector (“moment free”) until the fit between the forward model and empirical current is minimized.

The adjustment of the parameters is done with minimization techniques. Minimization techniques take the free (to-be-estimated) parameters of the dipole and use iterative techniques to change the free dipole parameters until the fit index value between the model and empirical data is minimized. “Numerical Recipes in C” (Press, Flannery, Teukolsky, & Vetterling, 1992; Press, Teukolsky, Vetterling, & Flannery, B.P., 2002) has several multidimensional minimization techniques (Chapter 10, <http://www.library.cornell.edu/nr/cbookcpdf.html>).

The EMSE computer program (Source Signal Analysis, Inc) uses the Nelder-Mead simplex method (Nelder & Mead, 1965; Press et al., 1992, 2003, chapter 10.4, <http://www.library.cornell.edu/nr/bookcpdf/c10-4.pdf>). This method is easy to implement and runs quickly, particularly when restrictions can be made about the parameter modifications (e.g., possible range of values, restricted range of values). “The method requires only function evaluations, not derivatives. It is not very efficient in terms of the number of function evaluations

that it requires. Powell's method (§10.5) is almost surely faster in all likely applications. However, the downhill simplex method may frequently be the *best* method to use if the figure of merit is "get something working quickly" for a problem whose computational burden is small. The method has a geometrical naturalness about it which makes it delightful to describe or work through." (Press et al., 1992, 2003, pg 408).

There are several other minimization procedures, of which several are implemented in the current model. Three methods described in Numerical Recipes are the direction set methods ("Powell", Powell, 1964; Press et al., 1992, 2003, chapter 10.5, <http://www.library.cornell.edu/nr/bookcpdf/c10-6.pdf>); the conjugate gradient methods ("Fletcher-Reeves", Fletcher & Reeves, 1964; "Polak-Ribiere", Polak & Ribiere, 1969, and Polak, 1971; Press et al., 1992, 2003, chapter 10.6, <http://www.library.cornell.edu/nr/bookcpdf/c10-7.pdf>); and variable metric methods ("Broyden-Fletcher-Goldfarb-Shanno" variant of the Davidon-Fletcher-Powell minimization, Broyden 1970a, 1970b; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970; Press et al., 1992, 2003, chapter 10.7, <http://www.library.cornell.edu/nr/bookcpdf/c10-7.pdf>). These methods consist of a parameter vector, \underline{P} , and a direction vector, \underline{n} . The procedures find a scalar value, $\underline{\lambda}$, that minimizes the function, $(\underline{P} + \underline{\lambda} * \underline{n})$, and replaces \underline{P} with $\underline{P} + \underline{\lambda} * \underline{n}$, and \underline{n} with $\underline{\lambda} * \underline{n}$, and then repeats the process. This is done iteratively until there is no longer a change (or very small change) in the value of the function between two successive passes. This is done most quickly if in addition to the direction of the change, we can change the value of \underline{P} and \underline{n} by a size of change. This can be accomplished by estimating the gradient of change with the partial derivative of the parameters in \underline{P} . The Broyden-Fletcher-Goldfarb-Shanno, Fletcher-Reeves, and Polak-Ribiere use first partial derivatives for the gradient calculation, whereas the Powell method does the optimization without gradients. It DOES make a difference which algorithm is chosen. The Powell and Broyden-Fletcher-Goldfarb-Shanno minimizations find the smallest minima. The Broyden-Fletcher-Goldfarb-Shanno procedure is a one-step memoryless quasi-Newton method, and is very accurate in its minimization. With numerical derivatives needed, the former is much faster than the latter, but the latter may be more precise. The current program uses a numerical estimation technique for the derivative (Ridders, 1982; Press et al., 1992, 2003, chapter 5.7, <http://www.library.cornell.edu/nr/bookcpdf/c5-7.pdf>). The partial derivatives are calculated by finding the first derivative for a single parameter while hold all others constant, and repeating this for each parameter. The Fletcher-Reeves, Polak-Ribiere, and Nelder-Mead simplex methods are not as accurate.

The gradient methods come from the Newton algorithm for computing the minimization functions. Newton's method(s) require computation of the *gradient vector* and the *Hessian matrix*. This method finds the minima of a function by using the derivative of the function to determine how fast the function is changing for the current parameter values, and change the parameters by an amount proportional to the gradient (first partial derivative). The Hessian matrix is a matrix (n X n) of second partial derivatives for the function. It may be used to identify the next gradient, and the rate of change of the minimization can be used for stopping points. Computation of the Hessian can be a time-consuming, or may not have analytic procedures for computing, or may not exist, or may not be positive-definite (needs inverse). The quasi-Newton methods (Broyden-Fletcher-Goldfarb-Shanno, Davidon-Fletcher-Powell) use the Hessian matrix but start with an identity matrix and build it gradually over the course of the operations. Each column of the Hessian can be approximated by taking the difference between two instances of

the gradient vector evaluated at two nearby points. For sparse Hessians, we can often approximate many columns of the Hessian with a single gradient evaluation by choosing the evaluation points judiciously. The quasi-Newton methods are generally the fastest and most accurate, though they have the issue of computing the gradient (next paragraph). The conjugate gradient methods (Fletcher-Reeves, Polak-Ribiere) use the Newton logic, but use a computational device that eliminates the need to compute the Hessian matrix. Thus they are fast, but may not be accurate if the direction vector goes awry in the minimization. A discussion of these issues is presented in chapters 9 (Newton-Raphson) and 10 (quasi-Newton) of Numerical Recipes in C (Press et al., 1992, 2002).

There is one issue specific to ECD analysis for the minimization techniques. The acceptable changes for the free (to-be-estimated) parameters must be limited. This could be the source of the dipole (1-shell, 3-shell, 4-shell, realistic) or the moment size (normed, unnormed). Each step in the procedure that estimates a new direction vector (and new parameter vector) must test the changed parameters to insure they are in the bounds of the source volume (location vector) or are normed (moment vector). The non-gradient methods (Powell, Nelder-Mead) and the gradient methods (Fletcher-Reeves, Polak-Ribiere, Davidon-Fletcher-Powell, Broyden-Fletcher-Goldfarb-Shanno) are approximately equivalent in the primary line search routines for this. However, the gradient methods need to estimate the gradient at each step of the line search minimization. The partial derivative of a specific variable with respect to each other variable must be tested against the limits each time a new parameter is estimated. The partial derivative for a single parameter (e.g., "location-sagittal") may require several limits' tests, and a dipole with location and moment free would have to do this for each of the six free parameters. For ECD models that only estimate moments this is relatively easy (only normed models adjusted), and for ECD models who have locations with spherical locations this is relatively easy (adjust based on spherical shell with analytic computation). The increased limit testing is compensated by fewer minimization steps in gradient models. For realistic source models that have free locations, there could be a search of an entire FEM space to find a suitable new parameter location, which could be several thousand locations. The fewer minimization steps in gradient models will not compensate for realistic source models with many locations. One minimization strategy in this respect is to use a less accurate non-gradient method (Powell, Nelder-Mead) followed by the gradient method (Fletcher-Reeves, Polak-Ribiere, Davidon-Fletcher-Powell, Broyden-Fletcher-Goldfarb-Shanno). This would be necessary only for a realistic source model with a large number of source locations. However, an inaccurate non-gradient method (e.g., Nelder-Mead) might lead to a location that was a poor local minima that could not be corrected by the gradient method, or, the non-gradient method might be as accurate (e.g., Powell) and the gradient method would be superfluous.

The current computer program implements the Broyden-Fletcher-Goldfarb-Shanno, Powell, Fletcher-Reeves, Polak-Ribiere, Nelder-Mead minimization methods. These are optional parameters in the computer program. One could do more programming (i.e., do successive methods) if desired for these. However, my testing has shown that the Broyden-Fletcher-Goldfarb-Shanno and Powell methods find good fits, usually nearly identical, and nearly identical parameters. The Polak-Ribiere, Fletcher-Reeves, and Nelder-Mead methods do not work as well (fit index), find different locations than the other methods (and than EMSE's Nelder-Mead), and often the Fletcher-Reeves does not fully terminate. I implemented the Nelder-Mead simplex method, and it was sufficient for simple models (e.g., one free dipole, or moment-

free models), but was difficult to adapt to the parameters for models with two or more dipoles. I have previously implemented the Nelder-Mead simplex method for minimization (Richards & Anderson, 2004), but found it to be inefficient and less accurate in comparison to the Broyden-Fletcher-Goldfarb-Shanno quasi-Newton variable-metric method. The non-gradient Powell method alone is superior both in speed and accuracy to the Nelder-Mead method followed by a gradient method.

Using 1-shell, 3-shell, source-shell, and realistic models in EMSE Data Editor

The Data Editor module of the Source Signal computer program may be used to do ECD models. This is done by loading a time series and electrodes, setting the dipoles in a ECD GUI, and doing the estimation. The estimation methods are a “1-shell” where the forward model is done from the source dipole to the electrode with a single shell radius, the “3-shell” (default) where conductivity is done with an outer scalp shell, a skull shell ($0.92 * \text{scalp}$), and a brain shell ($0.87 * \text{scalp}$).

The dipole sources can occur anywhere in the inner shell. A preferable (and not too cumbersome) method is to restrict (or expand) the locations of the dipoles with a “Source Volume”. A source volume is a wireframe created in MR Viewer that represents the places where dipoles can occur. This will restrict dipoles to that location rather than the inner shell, or expand dipoles to beyond the inner shell. Two source volumes that should be made are the “Gray Matter” source volume for typical dipole analysis, and the “Gray Matter Eyes” source volume for when the eye movements will be analyzed.

A demonstration of the shell can be done with “Make Spheres Region Point Files for three-shell model” which will take the electrode file of a subject and calculate the radius, and create the three-shell wireframes for MR Viewer. The shells may then be viewed in MR Viewer.

I have tested this with eye movement ICA component loadings from an eye movement component. Figures in the ppt file show the analysis of the ICA component loadings for the eye movement component cluster described above. The figures show the ECD locations for various models for the eye movement cluster. The locations were seeded with an initial location based on the Talairach coordinates for that participant for the eyes. The 1- and 3-shell models did ok, but the source volume models were superior in location and fit for almost every participant. The result also suggest that the accuracy of the procedure is closer to 1 cm rather than 5 cm, which is probably adequate for identifying psychophysiologicaly relevant cortical areas (fMRI is probably only that relevant due to differences in participant brains, smearing due to averaging and morphing, and unreliability of source locations in fMRI).

The Data Editor module must first be used to general forward model files for source volume analysis. This may be done with a source volume file and the 1-shell, 3-shell, or FEM conductivities. Thus, we might have a 3-shell conductivity analysis but with sources restricted to gray matter, or a FEM conductivity analysis and with sources restricted to gray matter. These combinations should proved to be superior to the regular 3-shell model.

Forward model calculations program

I have done a c++ program that computes the forward model parameters for use in my realistic current source analysis program.

One forward model that should be created is one for the “four-sphere resistance”, “realistic source volume” model. This calculates the paths through the four spheres for each element of the

source volume of interest. This forward model is done with a C++ computer program. This program finds any eye element (FEM type 10) or gray matter (FEM type 4) and computes the conductivity from the source to each electrode. This is done by calculating in a four-shell sphere the distance from the source to each electrode, and the portion of that distance occupied by the gray matter, CSF, skull, and scalp. The length of the vector from the dipole to the electrode is calculated for each of the four spheres and the combined conductivity value is saved. This results in a file of source conductance values, with each vector consisting of sum volumes for the FEM types, the index of the source in the CLabTetra array, and the electrode number (0 to 127). The resulting source conductances and the FEM model are used later in the analysis of the ECDs for a “four sphere conductance” “realistic sources” model. The realistic source model for the four-sphere resistance model could be computed online, but the calculations take a small but finite period of time and would add substantially to computing time for the source estimation.

A second type of forward model calculation is done. This program finds any eye element (FEM type 10) or gray matter (FEM type 4) and computes the conductivity from the source to each electrode. This is done by tracing a path along spherical coordinates from the source to the electrode, finding any FEM tetrahedral that have an intersection with this path, and keeping the sum of all volumes for those FEM tetrahedral type for the source-electrode combination. This results in a file of source conductance values, with each vector consisting of sum volumes for the FEM types, the index of the source in the CLabTetra array, and the electrode number (0 to 127). These conductivity values may be used in the shell calculation models in place of the shell estimates of conductivity. The resulting source conductances and the FEM model are used later in the analysis of the ECDs. This computer program is time-consuming. For example, using two “screamers (dual Pentium 3.25Ghz, 4GM)”, and three very fast computers (dual Pentium 2.5 GHz, 1GM), about 2000 sources per hour can be done (e.g., 30K to 50K sources, about 20 hours). I also have used 20 desktop computers and three very fast computers and done the computations is about 6 hours (about 5000 sources per hour; massive parallel computing using separate workstations). A typical head model 3fr file may have 30K to 50K electrical source coordinates and 400K tetrahedral. (e.g., S0006 2/8 had 2M FEM; 3/12 had 1M FEM; 4/12 had 404K FEM; 4/16 and 4/20 didn't work; 4/32 had 87k Coordinates and 400k FEM). Figures in the ppt file show the results from one such analysis, with the current paths from the dipole to the electrode represented, and the head medium types indicated by coloring the FEM elements in the path.

The computer program does not compute a full forward model for the BEM or FEM models. These forward models take into account the current flow through the whole head volume. Currently these forward models may be calculated with the BESA or EMSE analysis programs. Though at present not implemented in the computer program, the computer program could use the transfer coefficients from the BESA or EMSE forward models when computing the effect of a dipole at a tetrahedral element on the electrodes (the forward model scalp map).

The forward model program needs several files. First, the subject electrode placement file (S####.elp) and fiducials file (S#### Fiducials.elp) need to be created and are in the subjects “root directory” (“\Laboratory Data\MagneticResonanceImage\S#####”). Second, a file of conductivity values with the appropriate values for each tissue type (S#### Conductivity Values.txt). Third, the 3fr file with the FEM coordinates and FEM tetra (four coordinates, tissue type) needs to exist (S##### Whole Head.3fr). The last three files are in the “Realistic Sources” folder of the subjects root directory. The first step of the program is to read in the electrode

placement and fiducials files. Second, the program reads in the 3fr file and creates a binary file with the coordinates and a binary file with the tetra, which are later read in if they exist. From these the program calculates the midpoints of the tetra and the volume of the tetra. The output from the program creates the source conductance file for the separate electrodes/tissue types (S#### Source Conductance.bin). This is later combined with specific conductivity values to create total conductivity from the source to each electrode for the realistic model (S#### RealisticResistance Conductance.bin) and the four sphere model (S#### FourSphere Conductance.bin). The latter two files could have different values set for the tissue conductivity in order to test the effects of conductivity, or for different ages.

When the program starts, it asks for a subject / file number. The subject number must be greater than 1000, and a subject information file located in the Subject Information folder, but the subject numbers are < 1000 in the MRI folder. The file indicates to the program what to analyze:

- 0: Calculate the realistic source model
- 1: Ditto, but print out the coordinates to text files for region, vmi, wfr
- 2, 3: calculate the four sphere model (2 is calculate, 3 is calculation and print feedback)
- 4: just put four sphere resistance to FEM
- 50:, 51: put the coordinates listed with case:1 into a 3fr and analyze file for display
- 52: find any bin files in the tempdirectory, created by case 0 or case 1, and add to total
- 53: get the sums of the conductance files of the 10 FEM types
- 54: old tetra model (not used)
- 55: utility, create all CSD in a directory
- 56: compare the foursphere and realistic values

The calculation for the realistic conductivity calculations will be described. First, the program can run multiple threads (4 threads in screamer; 3 threads in fast computer; 21 thread in desktop; i.e., have run 20 desktops, 3 fast computers, 1 screamer, for about 50 threads). The threads can access a shared directory ("Shared Memory") and use file semaphores to communicate. Only one thread can access the file system at a time (SourceConductance.sem); the first thread of the first program creates a "FirstProgram.sem" that if deleted will stop all threads; after the threads are done the program will wait for another sem, either "ExitProgram.sem" to exit all programs, or "S####.sem" to work on participant number ####; each thread will claim a source coordinate with a thread (Source#####.txt) and will not work on binary files already created (Realistic Source Files\S##### Source Conductance.bin). Second, the first thread in the first program checks for bin files in the Shared Memory directory and combines all of them with the S#### Realistic Conductance.bin file in the Realistic Sources folder. Third, each thread uses the following algorithm for tracing the paths of the files:

- 1—check for the next available coordinate to be calculated, in FEM
- 2—make sure it has not been calculated (in FEM, in txt sem, in bin)
- 3—check if bin exists in shared directory and remove it
- 4—restore the original mri fem to the copy
- 5—subtract the midpoint of the current coordinate from each midpoint
this will make the points of the fem-copy relative to the midpoint
- 6—Restore the electrode copy from the originals
- 7—for each of the electrodes:
- 8—set all the source coordinate sums of the tissue types to zero.
- 9—subtract the midpoint of the current coordinate from the electrode (see step 5)

- 10—for each of the tetra in the copy
- 11—if the tetra and electrode have opposite radius, don't use (in opposite directions)
- 12—calculate a location with tetra radius, in direction (phi, theta) of electrode
- 13—if this location is within the radius of the current coordinate tetra, then use it
- 14—add the volume of the tetra to the source coordinate sums of tissue types.

Repeat steps 10 to 14 for each tetra
Repeat steps 7 to 14 for each electrode
15—save the source coordinates into a binary file
Repeat 1 to 15 for each source coordinate in the FEM

Fourth, the results of the algorithm creates a binary file with 128 electrode * 10 tissue types data containing the summed volume for each tissue type in the path from the source coordinate to the electrode. These are combined into a single file, with the offset of the tissue types stored in the CLabTetra object for each of the source coordinates. These binary files can be condensed by multiplying each volume with the associated conductivity value, and summing the volume-conductivity by the proportion of the volume occupied by that tissue type in the path. There are then 128 electrode values for each source conductance tetra save in "S##### Realistic Conductance.bin" in the Realistic Sources directory. Finally, the four sphere resistance values are calculated for each source coordinate tetra and saved in "S##### FourSphere Conductance.bin".

Realistic current source analysis program

I have done a c++ program that reads the forward models, electrodes, electrical data, and dipoles and does estimates of dipole location and amplitude based on the realistic forward model. This program can do the following models: 1-shell forward model, with 1-shell source, 4-shell source, or realistic source; 4-shell forward model, with 4-shell source, or realistic source; realistic conductance forward model, with realistic source; and in the future will do FEM forward model, with realistic source. The program accepts input from a control file and does GUI-guided batch analysis of the models, or complete batch processing. The program is described in greater detail in the following. The program is named "Realistic Current Source Analysis GUI" or "Realistic Current Source Analysis MonitorWindow".

The "GUI" version of the model presents a tabbed window with three elements: control commands, dipole editing, and analysis. The control commands tab will create and edit control files that have elements like the type of model, the subject numbers for MRI, ELP, AVG, or cluster files, and other information about the analysis separate from the dipole information. A single control commands tab is given at the program opening, and at least one must always exist, but new control commands tabs may be created from this tab. The dipole editing tabs will edit EMSE-style ECD files. There are up to six dipoles possible (could be expanded), sections for the estimate type, the resistance and source volume models, and normed or raw dipole moments. Each dipole has these elements, since the computer program is not restricted to a single model for all dipoles but uses separate specified parameters for each dipole. A single dipole editing tab is given a program opening but new dipole editing tabs may be created from this tab. The analysis tabs are generated by the control commands tab(s) and produce the results of the batch-style analysis. The analysis object takes as its input a control commands file and perform the batch commands. The PPT file has several screen shots of the GUI version, including the opening two tabs, analysis tab, several tabs open ("Finished" are finished analysis tabs).

A “MonitorWindow” version of the program exists that is similar to the CMonitorWindow object in the JERLab DLL procedures. This program opens a window with a single tab that is an analysis tab like the GUI version. In fact, the C++ programming for the MonitorWindow version is identical to that of the GUI version. The program takes as its input a control file in the command line or the “ControlRealisticCurrentSourceAnalysis.txt” in the D:\Temp directory. The main purpose of the MonitorWindow program is to run batch analysis files of several control files from a CMD program. This allows the analysis of multiple control files without GUI intervention. The ppt file has a screen shot of the MonitorWindow version.

The control command files have a specific order of commands. These are generated/edited by the control command tab of the GUI program, but are txt files that may be done by hand or SAS programs. The program takes as input a control file that specifies the following:

//Model type

//Subject number: Number of fem and source conductance

// electrodes, fem and source conductance from this subject

//AVG file: file with electrical data (or wild-card designation), single participant, or cluster

//a series of:

ECD model

Coordinates

ECD output

//ModelType: There are currently several model types. Each model represents a manner in which I do analysis of ICA data. They are SUBJECTAVGONEMRI, SUBJECTAVGSUBJECTMRI, CLUSTERAVGSUBJECTMRI, and CLUSTERAVGONEMRI with submodels CLUSTERAVGONEMRIAVGELP and CLUSTERAVGONEMRISUBJECTELP.

The model types consist of two or three parts, the AVG file with the electrical data, and the source of the MRI, and if relevant, the source of the ELP file for the analysis. The model type also designates the minimization procedure (BFGS, POWELL, FLETCHERREEVES, POLAKRIBIERE). The model types are read by the analysis programs and separate subroutines are done depending on the control statement.

AVG file designation: The SUBJECTAVG will take electrical data from a file that is listed. This file may be a single file, or a series of files in a single folder using wild-card notations. This file assumes that the subject elicited electrical data is in a file with subject number, and ELP and fiducial files are available for that subject in a fixed directory. The CLUSTERAVG designation will take electrical data from a clustered ICA file. The file is the control file for the Cluster PCA program and has a list of subject numbers and slice numbers that are individual slices in a ICA cluster. These subjects AVG files are in a companion folder (..\..\AVG Files) and the ELP files if relevant are in another companion folder (..\..\ELP Files).

MRI file designation: The ONEMRI or SUBJECTMRI refers to the source of the MRI that is used for the analysis. The ONEMRI indicates that a single MRI will be used for all participants. This MRI and all of its associated files (e.g., four sphere and realistic conductance values) will be loaded a single time and all participants will use this MRI. This should be used when a single MRI is available. The SUBJECTMRI is for doing analyses where each participant has an MRI.

In this case, the subject MRI is not loaded until the electrical analysis begins. For the CLUSTERAVG file type, the AVG file has information about the participant number AND the MRI number (or, separate folder has MRI-participant link) and each MRI is loaded for each AVG file that is analyzed.

ELP file designation: For ONEMRI, one can use the ELP file from the MRI, or append “AVGELP” and designate a single ELP file to be used for the analyses (i.e., average ELP file for an entire experiment”, or append “SUBJECTELP” where for each subject a individual ELP file is loaded. For SUBJECTELP subject ELP and fiducial files, when loaded, will be used to modify the vertices of the single MRI to put them at a similar radius as the subject, and so the models (e.g., shell) will be relative to the ELP of the individual subject with electrical data. The AVGELP and SUBJECTELP designations of the ELP file do not make sense for the SUBJECTMRI model type, since the MRI has its own relevant ELP file.

Minimization method: The designates the minimization procedure (BFGS, POWELL, FLETCHERREEVES, POLAKRIBIERE).

The ModelType parameter can be extended as needed. For example, on type of analysis could use the “estimatedipole()” method to do foil type dipoles, another with sequential dipole testing, etc. This will be extended for specific projects or experiments as needed.

//Subject number: This is a subject number for a MRI located in the Laboratory Data \ MagneticResonanceImage folder. This MRI should have the whole head.3fr files, ELP and fiducial files, four sphere and realistic conductance data, tetra and vertices, etc. The subject number is given only for the ONEMRI type analyses. For SUBJECTMRI models, the ERP participant number in the AVG file is linked with a MRI subject number and that MRI should have all the constituent files in the MRI folder.

//AVG file: The SUBJECTAVG command requires the input of a EMSE-type avg file with electrical data. This is a file with electrical data (or ICA component loadings; or PCA component loadings). This file will have a number of data slices, each data slice having “nelectrode” number of data. The number of electrodes in the AVG file should match the number of electrodes in the MRI. This file may be a single file, or use wild-card (*) notation for a series of files in a single folder. The FileFind procedure is used to go through all files that meet the criteria of the file name. Either the single designated file, or the FileFind files, will be tested with each of the ECD analysis instructions.

//AVG file with ICA clusters: The CLUSTERAVG command requires the input of a cluster file. This file has participant numbers and slice numbers for that participant. The AVG file for the participant is location in a fixed directory from the cluster file, and subject electrode and fiducial files, or subject-MRI files must be indexed. The program will read through each cluster participant and apply the ECD analysis instructions to each participant in the cluster file.

//A series of: The next lines in the control file are the ECD model, Coordinates, and ECD output.

ECD model: The ECD model is a single ECD file in the EMSE format. This has the number of electrodes, a designation for the source model and the conductance model, starting dipole locations and moments, and state (fixed, free, momentfree) information about each dipole. Note that in EMSE the model type is the same for all dipoles. In this program one can have mixed models (e.g., a “foil” location that accounts for stray variance anywhere in a 1-shell space, and source volume / realistic forward models for the real electrodes of interest). The information

about radius, sphereorigin, and fit parameters are read in but not used in the analysis. The radius and sphereorigin are replaced by the programs determination of radius (from the MRI fiducials) and calculation of sphereorigin for spherical shell models. The slicestart and slicestop values for the first dipole are used to control what slices in the electrical data will be used.

There are four types of specified information for each dipole. Each type of information may differ for each dipole. First, the estimation model type is specified. This can be inactive, free, fixed, or moments (see EMSE descriptions). Second, the resistance model may be specified. This may be 1-shell spherical model, 4-shell spherical model, realistic conductivity model, FEM model. Third, the source model may be specified. For the 1-shell spherical resistance model, this may be the 1-shell, 4-shell, or realistic source volume models. For the 4-shell spherical resistance model, this may be the 4-shell or realistic source volume models. For the realistic resistance model or FEM model, this may only be the realistic source volume model. Fourth, the moment may be normed to 1.0, or unnormed (raw). Normed moments have the effect of using a fixed current amplitude for any dipole and emphasizing moment direction and dipole location as the major parameters for the model. The unnormed moment solution allows current level to vary also, which then plays a role in dipole estimation and provides better fitting models.

There is an equivocal relation between the ECD model files for this program and the EMSE ECD program. The EMSE program and the current program will read each other's model files. However, the EMSE program requires that all dipoles have the same resistance and source volume models, uses only normed models, and has single parameters for several other models. EMSE also only allows 1-shell resistance / 1 shell source volume, 3-shell resistance / 3-shell source volume, 3-shell resistance / realistic source volume, and FEM resistance / realistic source volume models. The latter two methods also require creating and loading source volume files and forward model files. The EMSE ECD program will automatically create and save 32 dipoles regardless of the number of dipoles, where the GUI editing program will read only up to 6 dipoles, and save dipoles only up to the first non-active dipole with all zero location/moment parameters. So it is ok to read the dipole files in one program created by another program, but these should not be saved / overwritten because these parameters may change. Thus, the GUI program may be done to estimate the dipoles and the EMSE program read them in and do further testing.

Coordinates: The coordinates for the source volume, if any, are designated. The coordinates are used only for realistic source volume models in the dipole source analysis. First, the "continue" keyword here will use the prior coordinates. Second, a designated 3fr file name will use the coordinates in a wireframe file—i.e., use for designating specific anatomical locations (eyes, anterior cingulate). Third, key words including "GRAYMATTER", "EYES", or "WHITEMATTER" will select the coordinates to be found in the current MRI with that type of matter (these words may be concatenated in any order).

ECD output: The file name for a file for the output is given. This should be a STEM only, since multiple files are output with the designation. For example, at each iteration of the "slicestart to slicestop", an ECD will be calculated, a file name will be constructed (e.g., stem + subject number + slice nnnn.ecd) and the ECDs will be output. The final electrical model will use this stem (e.g., stem + subject + model.avg) and a final ELP file with all the dipole locations will be output (e.g., stem + subject + dipoles.elp).

The EMSE programs (MRViewer, Data Editor) may be used to display the results of the program. The preferable visual display for the dipoles is the MRViewer program. This may be a single dipole file for an analysis, or the ELP file that has all the dipole locations. The Data Editor ECD program also can be used to display the file, but it is harder and could save over the GUI format ECDs. It should be used primarily for testing or to see if the EMSE algorithms work similarly. Also output from the file are AVG files with model data and residual data. Each individual participant will have an AVG file for any of its slices (1 slice for CLUSTERAVG, potential multiple slices for SUBJECTAVG). The model data will have 0's for any non-estimated slices and the model otherwise. Additionally, a residual AVG file will be created for each modeled data.

Model Testing

I have tested the computer program and various models for the program. The data that was used came from a study of prosaccades and antisaccades. The raw ERP segmented data was analyzed with independent component analysis, and clusters were created on the basis of the topographical similarity of the cluster component loadings and the cluster component activations. The ppt file shows the ICA cluster for an eye movement component that had positive and negative loadings on the outer edges of the eyes and activations that occurred primarily at the onset of the eye movement identified in the EOG saccade. This cluster represents ICAs obtained from about 15 participants who participated in about 35 testing sessions.

Several parameters were tested with factorial combinations of the following variables: Model type (one moment in center; two moments in eye; one free dipole seeded at center location; two free dipoles seeded at eye), resistance and source (1s-1s, 1s-3s, 1s-real, 3s-3s, 3s-real, real-real), and norming of the moment (normed, raw). The MRI and realistic model for a single participant was used for the realistic source and realistic resistance parameters. In one type of model, the ELP file from the subject was used. In a second type of model, the ELP file from the individual participant based upon that participant's MRI was used. The latter type of model (single MRI, but subject-based ELP) would be a typical strategy when testing infant participants with a single MRI from a representative participant. Six types of optimization methods were used (BFGS, Fletcher-Reeves, Nelder-Mead, Polak-Ribiere, Powell, and a Powell-BFGS sequential). These combinations produced about 3000 ECD tests for each optimization method, about 18,000 in all. The dependent variables for this model test were 1) the fit between the model and the empirical data, and 2) the speed of the fit (number of analyses per minute, with a min value of 1 per min).

There were several expected results. First, the raw dipole moment vector resulted in better fits than did the normed moment vector, but expectedly took longer to calculate. Use the individual participant ELP file resulted in better fits but took longer to calculate. The moment free model with two dipoles, or the two active dipole models were a better fit than the one-dipole or one-moment models. However, the two free and one-free models took substantially longer to calculate than the moment models.

Using realistic models improved the model fit slightly. There was a slight increase in fit from the 1s-1s to the 1s-3s to the 1s-real, and the 3-s models were a better fit than the 1-s models. However, the model with realistic sources and realistic resistance was a significant poorer fit than any of the models. Thus, at least for this participant, the realistic model was not a source of great improvement in the fit and took substantially longer to calculate. The spherical models (1s-

1s, 1s-3s, 3s-3s) were substantially faster than the realistic source models (1s-real, 3s-real, real-real).

There was a marked difference between optimization techniques. The Powell method resulted in the best fit, followed slightly by BFGS model, the Fletcher-Reeves and Polak-Ribiere generally resulted in similar but poorer fits, and the Nelder-Mead method resulted in the poorest fits. Often, the Powell and the BFGS methods resulted in equal fits and identical parameter estimates. The BFGS model was always the fastest optimization, even when doing the real resistance / real source models. This is surprising, given the need for frequent limits checking in the real source models during the derivative calculations. Apparently the use of the derivative method (similar to Fletcher-Reeves and Polak-Ribiere) coupled with the adaptive estimation of the Hessian matrix compensated for the increased fit testing of this method relative to the Powell method. The Powell method often was closest in speed to the BFGS model, whereas the Fletcher-Reeves and Polak-Ribiere models were the slowest and had the poorest fits. One reason for the poor fits was the lack of final model fit for the Fletcher-Reeves and Polak-Ribiere. The optimization methods are designed to be a “one-pass” procedure, but unless fine-tuned specifically for the parameters, multiple passes will improve the model fit. These two procedures often reached the maximum number of iterations set by default in the Numerical Recipes program, and the new dipoles were fit again. After the first optimization the minimum was close, and the fit sequence then became very slow, and was arbitrarily ended after five passes through the procedure. The Powell procedure rarely required more than one pass and the BFGS procedure usually required less than the five maximum passes, whereas the other two procedures sometimes required multiple passes especially for the realistic models. A combined strategy using the Powell method followed by the BFGS methods resulted in a fit as good as the Powell method, and was as fast as the BFGS method. This occurred because the Powell fit often had to be repeated, but the Powell fit immediately followed by the BFGS procedure usually resulted in a single pass procedure. The Nelder-Mead method resulted in the poorest fits. The simplest models (e.g., one free dipole, or moment-free dipoles) were slightly poorer fits than the Fletcher-Reeves and Polak-Ribiere models, but the two free dipole models had very poor fits. The Nelder-Mead model was one of the fastest models for the simple models for which it provided a good fit (e.g., one-free or moment-free models, and resistance or source volume shell models), but was the slowest for the models in which it provided a poor fit (two free dipoles, or realistic source or resistance models).

A mention should be made about the relative speed of these techniques vis-à-vis the GUI and MonitorWindow programs. The beginning of both programs loads the MRI 3fr realistic model parameters, taking several seconds, whereas the fastest optimization methods do about 100 optimizations per minute and the slowest about 10. If the GUI program is being used for a single dipole fit, the front-loading time is much more significant for total time than the fit time. If the GUI program or the MonitorWindow program is being used for multiple ECD models, the front-load time becomes more important and the fit-time then may contribute substantially to the overall time. Other models may have severe front-loading issues and would have to be done in the batch mode (i.e., using realistic source and resistance models from individual MRI 3fr models).

There were several interactions among the factors that are important to consider. For example, for the 1-s and 3-s resistance models the four models (one-free, one-moment, two-free, two-moments) had similar fit levels, whereas for the realistic resistance model the two-free and two-moment models had a better fit than the other two models. The fit for the realistic resistance

models and the two-free / two-moment was better than the fit for the other models. A similar pattern was true for the realistic source models, and the best fits for the two-dipole models occurred for the realistic resistance / realistic source models. The raw dipole estimation was much better for the real models than the normed dipole, and the realistic resistance and realistic source models generally showed a slightly better fit for the raw dipole models than the spherical models—though not substantially. Since the two-dipole models are a better representation of the eye movement ICA, the realistic resistance model was more sensitive to a correct model use and resulted in a better fit when it was used with a correct model. Similarly, the BFGS and Powell model were about the same speed for the 1s and 3s models and both were faster than the other optimization methods, but for the realistic resistance model and the realistic source model(s) the BFGS procedure was faster than all other methods, and the other three methods did not differ. These interactions show that although overall the spherical resistance and source models provided better fits between the modeled and empirical data, this may have been occurring for poorly specified models (one dipole models) or for poorly estimated dependent variables (normed dipole moments). For the realistic resistance and realistic source models, when examining the raw dipole moments for the reasonable dipole model, the fits were equivalent to or better than the spherical models. An interesting conclusion to this is that a better fit can be achieved with a poor dipole model using less sophisticated methods than with more sophisticated methods, but a good fit (and among the best fits) for a good dipole model comes from the realistic models.

The optimization methods were compared to the EMSE computer program. This tests the external validity of the testing routines and the mathematical suitability of the current program. Not all models can be compared, since the EMSE program does only 1s-1s, 3s-3s, 3s-realistic models. The following comparison was made. A single participant with MRI and full source volume and source realistic resistance model was used. This participant had two testing sessions and the eye movement component from the ICA was analyzed. The 1-shell (resistance, source) EMSE and optimization models, the 3-shell (resistance, source) EMSE models, the 4-shell (resistance, source) optimization models, the “source volume” “forward model” from EMSE (3-shell resistance, realistic source), and the optimization 4-shell resistance and realistic source models were compared for one moment dipole, two moment dipoles, one free dipole, and two free dipoles models. The comparison between EMSE and the other routines primarily validated the optimization routines, but there were differences. First, the EMSE program uses a normed moment vector. The comparison between the EMSE and the other programs on the normed vector showed approximately equivalent model fits for the EMSE, BFGS and Powell methods, and slightly better fits for the EMSE than the Fletcher-Reeves and Polak-Ribiere. The EMSE program uses the Nelder-Mead simplex method, but the results from the current implementation were much poorer fits than the EMSE program. This suggests that EMSE has made modifications of the Nelder-Mead method to match the characteristics of the data (i.e., initial parameter range, parameter adjustment ranges, occasional “bump” of location / moment vector to avoid local minima). Second, the comparison between the EMSE models, which had normed moment vectors, and the other optimization methods with non-normed moment vectors, showed a clear superiority for fit for the non-normed vectors. If the moment-free models were used, the non-normed moment vectors all showed a slight improvement over the normed EMSE models—i.e., the same location vector but moment vectors that are unnormed. For the location-free models, there was a substantial improvement in model fit. This indicates that the location vector and the moment vector interact. The normed moments have the effect of using a fixed current

amplitude for any dipole and emphasizing moment direction and dipole location as the major parameters for the model. The unnormed moment solution allows current level to vary also, so that location, moment direction, and moment amplitude play a role in dipole estimation and provides better fitting models. Third, the “Source Volume” choice for the Forward Model in the settings tab of the EMSE computer program demands that a source volume 3fr file is done and a source volume forward model fwd file is done. Then, the model uses (I presume) a three-shell resistance model and a realistic source model. This model in the EMSE program performed poorly. In every EMSE model the fit was worse than the corresponding 1-shell or 3-shell source/resistance model. This was true even when the same location vector was used (moment-free models), and only the moments were estimated. I tested a model that found the moments with the 1- or 3-shell models, fixed the dipoles, and then estimated the fit with the source volume model—the latter was a poorer fit. Conversely, finding the moment free model fit with the source volume model, fixing the dipoles, and using the 1-shell or 3-shell models resulted in a better fit. I may be using the EMSE computer program incorrectly, and this behavior did not hold true for other participants (see prior section). I suspect that the EMSE program has its routines optimized for the 3-shell model (3-shell resistance and source). I found the “source volume” model difficult to use (segmenting of MRI, 3fr, and FWD files) and I suspect that it has not gotten much real-user use. The other optimization models showed different results. The more realistic models were a better fit than the 1-shell and 4-shell models, and thus they were a substantially better fit than then source volume forward model in the EMSE program. There is additional advantage to using realistic source and resistance models, so the best fits from the optimization routines, including the EMSE computer program, come from the realistic models estimated with the Powell or BFGS optimization procedures. I also compared the location vectors for the EMSE and the other models. For all models of this particular subject, the EMSE models left the location in approximately the starting point—i.e., the initial starting point was a good one, based on theoretical grounds, and the optimization found a close minima. Most of the optimization programs did similarly. Almost none of the methods put the location vector in exactly the same location. Some of the optimization procedures moved the location vector to odd locations, particularly the two free dipole models, which might put the two dipoles both in the same eye, or move one dipole to a non-eye location. In general, however, very similar locations were found with the EMSE and the other methods and these locations were in reasonable locations.

The current testing protocol will be done several times and on other ERP or ICA components to insure these results generalize. There are two additional testing procedures that should be done. compared to the EMSE computer program to determine if similar results occur. First, the current test report used a single MRI for the realistic model. This is a decent representation of a typical EEG lab where a single representative MRI might be available, or an average MRI from another source, and MRIs of individual participants cannot be done (i.e., infant participants). However, the addition of specific MRIs for individual participants should improve the results of the realistic source/resistance models. Second, this needs to be tested with infant participants and with other parameters for the conductivity values.

Table 1. Resistivity values for various tissues

Tissue	$\rho[\Omega\text{m}]$	Remarks	Reference
Brain	2.2	gray matter	Rush and Driscoll, 1969
	6.8	white matter	Barber and Brown, 1984
	5.8	average	"
Cerebrospinal fluid	0.7		Barber and Brown, 1984
Blood	1.6	<i>Hct</i> = 45	Geddes and Sadler, 1973
Plasma	0.7		Barber and Brown, 1984
Heart muscle	2.5	longitudinal	Rush, Abildskov, and McFee, 1963
	5.6	transverse	
Skeletal muscle	1.9	longitudinal	Epstein and Foster, 1982
	13.2	transverse	
Liver	7		Rush, Abildskov, and McFee, 1963
Lung	11.2		Schwan and Kay, 1956
	21.7		Rush, Abildskov, and McFee, 1963
Fat	25		Geddes and Baker, 1967
Bone	177		Rush and Driscoll, 1969
	15	longitudinal	Saha and Williams, 1992
	158	circumferential	
	215	radial (at 100 kHz)	

Malmivuo J, Plonsey R, Bioelectromagnetism, Oxford University Press, New York: 1995.
 Available online at: <http://butler.cc.tut.fi/~malmivuo/bem/bembook/>

Table 2: Estimates of the conductivity (S/m) of body tissues below 100 Hz at body temperature.

Tissue	From Duck 1990	This study
Bone -Cancellous		0.07
Bone -Marrow		0.05
Cerebro Spinal Fluid	1.81	2.0
Cornea		0.4
Fat		0.04
Lens		0.25
Muscle	0.4	0.35
Blood	0.68	0.7
Bone -Cortical	0.02	0.02
Cerebellum		0.1
Dura		0.5
White matter	0.1	0.06
Grey Matter	0.3	0.1
Skin -Wet		0.1

<http://www.brooks.af.mil/AFRL/HED/hedr/reports/dielectric/Report/Report.html>

COMPILATION OF THE DIELECTRIC PROPERTIES OF BODY TISSUES AT RF AND MICROWAVE FREQUENCIES

[Camelia Gabriel, PhD.](#)

Sami Gabriel, MSc.

Below 100 Hz the impedance of biological material is mostly resistive. The contribution of the capacitive component is of the order of 10 % in most cases. The literature surveyed in this study shows that there are wide variations in the conductivity values obtained for the same tissue in various studies. The contribution of the tissue permittivity to body current is well within the uncertainty associated with the corresponding tissue conductivity. Therefore, in practice, the estimation of induced current in tissue is based on such conductivity values. Table 1 gives an estimate for conductivity in S/m of the main body tissues below 100 Hz from this study mitigated by literature values. The values tabulated by Duck (1990) are also shown for comparison . Average values were used where appropriate.

Table 3: Dielectric properties of human tissue and electrode material: conductivity σ [S/m] for 8.25 MHz, 42.6 MHz and 64.0 MHz.

Tissue	σ [S/m]
bone marrow	0.120 / 0.149 / 0.154
fat	0.051 / 0.064 / 0.066
bone	0.041 / 0.056 / 0.060
white matter	0.149 / 0.261 / 0.292
grey matter	0.271 / 0.466 / 0.511
skin	0.172 / 0.386 / 0.436
eye	0.789 / 0.863 / 0.883
muscle	0.610 / 0.672 / 0.688
blood	1.083 / 1.184 / 1.207
CSF	2.002 / 2.035 / 2.066
nerve tissue	0.215 / 0.290 / 0.312
Lens	1.020 / 0.270 / 0.286
optical nerve	0.215 / 0.290 / 0.312

<http://www-ibt.etec.uni-karlsruhe.de/people/mag/frames/papers/EMC99-MD/node3.html>

Table 4. Resistance values used by EMSE computer program.

Tissue	σ [S/m]
fat	0.045
bone	0.0132
white matter	0.2
grey matter	0.33
skin	0.35
eye	0.5
muscle	0.35
CSF	1.79
Dura	0.33

Table 5. Conductivity values for tables 1 to 4. σ [S/m]

Tissue	Table 1	Table 2	Table 2	Table 3	Table 4
Gray Matter	0.45	0.3	0.1	0.271 / 0.466 / 0.511	0.33
White Matter	0.14	0.1	0.06	0.149 / 0.261 / 0.292	0.2
CSF	1.2	1.81	2.0	2.002 / 2.035 / 2.066	1.79
Fat	.04		.04	0.051 / 0.064 / 0.066	0.045
Bone	0.005			0.041 / 0.056 / 0.060	0.0132
Muscle		0.4	0.35	0.610 / 0.672 / 0.688	0.35
Dura			0.5		0.33
Eye				0.789 / 0.863 / 0.883	0.5
Skin				0.172 / 0.386 / 0.436	0.35

Realistic current source analysis program procedures

I have included the .h file headers for the program and the CLabEEGAnalysis objects, and annotated it with brief descriptions. (October, 2005).

```
// AnalysisDlg.h : header file
//
/////////////////////////////////////////////////////////////////
// CAnalysisDlg dialog

class CAnalysisDlg : public CMonitorWindowDlg
//class CAnalysisDlg : public CDialog
{
// Construction
public:
    CAnalysisDlg(CWnd* pParent = NULL); // standard constructor
    CTabCtrl * ctc; //this saves the calling procedures tab control window
    CButton * button; //action button for closing the window
    DWORD m_tabnumber; //nSel of this tab to send "Finished" message
    CString commandline; //passes the command line from the calling program
    CString overallmodel; //control file overall model type from the control file
    DWORD m_subject; //subj number in control file
    CLabFEM mrfem; //fem object for mri
    CLabFEM subjectfem; //..for current subject. This is actually used in the calcs.
    CLabElectrodes mrielectrodes; //electrodes for mri
    CLabElectrodes mriducials; //ditto
    CLabElectrodes subjectelectrodes; //electrodes for current subject, used in calcs
    CLabElectrodes sphericelectrodes; //spherical electrodes for current subject
    CLabElectrodes subjectfiducials; //ditto
    CLabElectrodes dipolelocations; //dipole locations for output to ELP file
    CLabDipoles dipoles; //original dipoles
    CLabDipoles workingdipoles; //temp for estimating, used in calcs
    CLabElectrical empiricalldata; //empirical electrical data
    CLabElectrical modeldata; //working dipoles model data
    CLabElectrical csddata; //current source density
    CLabElectrical modeloutputdata; //for clusters, model output data
    CString subjectstring; //string with subject number
    CString electricaldatafile; //file with electrical data, subject or cluster AVG
    CString dipoleinputfile,dipoleoutputfile; //input and output for ECD files
    CString coordinatefile; //file with coordinates for this analysis
    CString avgelpfile; //avg elp file if used
    PJER_LDOUBLE empiricalltempptr,modeltempptr; //temp ptr to location in data
    ofstream logfile,tempfile,minsearch; //files for output
    ifstream controlfile; //control file for analysis

    JER_LDOUBLE avgresistance; //avg resistance in four-sphere model
    //these are the average distances from ctr for gm, scalp, for the current subject fiducials
```

```

JER_LDOUBLE distancegm,distancescalp,thicknessscalp;
PJER_LDOUBLE femtypeconductance[20]; //max 20 conductance values for fem types
DWORD np; //no of minimization parameters
JER_LDOUBLE p_values[100]; //up to 100 free parameters
JER_LDOUBLE **xi; //for powell, allocated at beginning of program
JER_LDOUBLE **hessin; //for bfgs, allocated at beginning of program
DWORD p_types[100]; //type of parameters 0:moments 1:locations
DWORD p_dipoleno[100]; //no of dipole for this parameter

//procedures for types of analysis, based on avg and mri file
void clusteravgonemri();
void clusteravgsubjectmri();
void subjectavgonemri();
void subjectavgsubjectmri();

//procedures for analysis
void doanalysisstartthread(); //start do analysis as thread
void doanalysis(); //runs the analysis

//the model data can be calculated for specific slice, for parameters, etc.
void calculatemodeldata(); //calculate model data and residuals, using tempptrs
void calculatemodeldata(const DWORD &i); //calculate the model data for specific slice
void calculatemodeldata(PJER_LDOUBLE); //use the parameters
void calculatemodeldata(const DWORD &i,PJER_LDOUBLE); //specific slice /parameters

//this is the main routine for estimating the parameters
void estimatedipoles(const DWORD &i); //estimated dipoles for specific sample
void estimatedipoles(); //get for tempptr
JER_LDOUBLE comparedistribution(const DWORD &i); //compare at specific slice
JER_LDOUBLE comparedistribution(); //compare using temp ptrs
void computefinalparameters(const DWORD &i); //compute for specific slice
void computefinalparameters(); //compute and put into workingdipoles
void findclosestipoles(PJER_LDOUBLE); //put p in dipoles, find dipoles, restore p
void findclosestipole(DWORD,PJER_LDOUBLE); //find a single close dipole
void checkcoordinates(); //check and do coordinates
void estimatefreemoments(const DWORD &); //find first free moments
void estimatefreemoments(); //find first free moments, for temp ptrs

//minimization procedures, found in optimization.cpp
void getfirstpartialderivative(DWORD,JER_LDOUBLE*,JER_LDOUBLE*);

//these are for bfgsminimum, see reference in text and Numerical Recipes
//these are adopted closely from Numerical Recipes
void jerlnsrch(
    DWORD n,JER_LDOUBLE xold[],JER_LDOUBLE fold,JER_LDOUBLE g[],
    JER_LDOUBLE p[],JER_LDOUBLE x[],JER_LDOUBLE.&f,

```

```

        JER_LDOUBLE stpmax);
void lnsrch(
    const DWORD &,PJER_LDOUBLE,const JER_LDOUBLE &,
    PJER_LDOUBLE,PJER_LDOUBLE,PJER_LDOUBLE,
    JER_LDOUBLE &,const JER_LDOUBLE &);
void getBFGSminimum();
void dfpmin(PJER_LDOUBLE,const DWORD &);

//these are for frpr minimum
void frprmn(PJER_LDOUBLE p,const DWORD &n); //with linmin
void dfrprmn(PJER_LDOUBLE p,const DWORD &n); //with dlinmin
void fletcherreeves(PJER_LDOUBLE,const DWORD &n); //frprmn with F-R deviant
void polakribiere(PJER_LDOUBLE,const DWORD &n); //frprmn with P-R deviant

//this is powell minimum
void powell(PJER_LDOUBLE p, JER_LDOUBLE **xi, const DWORD &n);

//various subroutines for minimums
void linmin(PJER_LDOUBLE p,PJER_LDOUBLE xi, const DWORD &n,
JER_LDOUBLE &fret);
void dlinmin(PJER_LDOUBLE p,PJER_LDOUBLE xi, const DWORD &n,
JER_LDOUBLE &fret); //uses derivative
void mnbrak(JER_LDOUBLE &ax,JER_LDOUBLE &bx,JER_LDOUBLE
&cx,JER_LDOUBLE &fa,JER_LDOUBLE &fb,JER_LDOUBLE &fc);
JER_LDOUBLE brent(JER_LDOUBLE ax, JER_LDOUBLE bx, JER_LDOUBLE cx,
JER_LDOUBLE tol, JER_LDOUBLE &xmin);
JER_LDOUBLE dbrent(JER_LDOUBLE ax, JER_LDOUBLE bx, JER_LDOUBLE cx,
JER_LDOUBLE tol, JER_LDOUBLE &xmin);
JER_LDOUBLE f1dim(JER_LDOUBLE x);
JER_LDOUBLE df1dim(JER_LDOUBLE x);

int ncom; //np for external
PJER_LDOUBLE pcom,xicom; //for external

CString CLabString(const DWORD &);
CString CLabString(const JER_LDOUBLE &);

```

```

////////////////////////////////////
// CControlDlg dialog

class CControlDlg : public CDialog
{
// Construction
public:
    CControlDlg(CWnd* pParent = NULL); // standard constructor

    CString controlfilename;

//*****
//procedures
void resetcontroleditboxes(); //set control file edit boxes to blank
void readcontrolfile();      //after get controlfilename, read it
void readdipolefile();       //after get dipolefilename, read it

afx_msg void OnChangeSlicestart(); //change the slice start text box
afx_msg void OnChangeSlicestop(); //change the slice stop text box
afx_msg void OnControlfilenew();  //create a new control commands tab
afx_msg void OnControlfileopen(); //open an existing control file
afx_msg void OnControlfilesave(); //save the control information, write file
afx_msg void OnControlfilesaveas(); //save as
afx_msg void OnControldatafilebrowse(); //browse the control data
afx_msg void OnDesignatedelpfilebrowse(); //browse designated elep file
afx_msg void OnEcdinputfilebrowse(); //browse ECD input file
afx_msg void OnCoordinatefilenamebrowse(); //browse coordinate file
afx_msg void OnEcdoutputstembrowse(); //browse ECD output file
afx_msg void OnAddtolist(); //add ecd, coordinates, output to listbox
afx_msg void OnRemovefirstlist(); //remove first triplet from listbox
afx_msg void OnRemovelastlist(); //remove last triplet from listbox
afx_msg void OnReplaceselectd(); //replace selected triplet from listbox
afx_msg void OnRemoveselectd(); //remove selected triplet from listbox
afx_msg void OnGetselectd(); //get selected triplet from listbox

```

```

////////////////////////////////////
// CDipoleDlg dialog

class CDipoleDlg : public CDialog
{
// Construction
public:
    CDipoleDlg(CWnd* pParent = NULL); // standard constructor

// Dialog Data
   //{{AFX_DATA(CDipoleDlg)
    enum { IDD = IDD_DIPOLEDIALOG };
        // NOTE: the ClassWizard will add data members here
    }}AFX_DATA

//members
    CLabDipoles dipole;
    CString dipolefilename;

//*****
//procedures
    void resetbuttons(); //reset all buttons to 0
    void setbuttons(); //set buttons from dipole
    void reseteditboxes(); //set edit boxes to blank
    void seteditboxes(); //set boxes from dipoles
    void readdipolefile(); //after get dipolefilename, read it

    afx_msg void Onchangelocation(UINT); //change any location / moment edit box
    afx_msg void Onresistance(UINT); //change any resistance button
    afx_msg void Onsource(UINT); //change any source button
    afx_msg void Onmodel(UINT); //change any model button
    afx_msg void Onmomenttype(UINT); //change any moment-norm button
    afx_msg void OnFilenew(); //create a new dipole edit tab
    afx_msg void OnFileopen(); //open existing dipole
    afx_msg void OnFilesave(); //save current dipole to file
    afx_msg void OnFilesaveas(); //save as
    afx_msg void OnChangeSlicestart(); //change slice start edit box
    afx_msg void OnChangeSlicestop(); //change slice stop edit box

```

```
//
//LabEEGAnalysis.h definitions for the eeg analysis functins
//
//CLabLocation: locations in rectangular or spherical space
//CLabElectrodes: recording electrodes in EMSE ELP file
//CLabElectrical: EEG data in avg file
//CLabTetra: tetrahedral object for FEM model
//CLabFEM: Finite Element Model
//CLabDipoles: Dipole files and elements, for EMSE ECE file
//
////////////////////////////////////
//these are the definitions for the fem type conductance locations in array
const DWORD WHITEMATTER=3;
const DWORD GRAYMATTER=4;
const DWORD CSF=5;
const DWORD SKULL=6;
const DWORD BONE=SKULL;
const DWORD SKIN=7;
const DWORD SCALP=SKIN;
const DWORD MUSCLE=8;
const DWORD FAT=9;
const DWORD EYES=10;
const DWORD NASALCAVITY=11;
```

```

//CLabLocation: locations in rectangular or spherical space
//The locations are kept both in rectangular and spherical coordinates, allowing
// the easy switch from one to another. Object initialize, initialize, copy,
// simple math, distance and angle functions are included.
//Dipoles, FEM midpoints, electrodes have location objects
class CLabLocation : public CLabObject
{
public:
    DLLEXPORTEXT CLabLocation();
    DLLEXPORTEXT virtual ~CLabLocation();
    DLLEXPORTEXT CLabLocation(const JER_LDOUBLE &,const JER_LDOUBLE &,const
JER_LDOUBLE &);
    DLLEXPORTEXT CLabLocation(const CLabLocation &);

public:
    //*****
    //members
    JER_LDOUBLE coronal,saggital,axial;           //rectangle coordinates
    JER_LDOUBLE radius,phi,theta;                //spherical coordinates

    //*****
    //assignment methods
    DLLEXPORTEXT void setRectangle(const JER_LDOUBLE &,const JER_LDOUBLE &,const
JER_LDOUBLE &);
    DLLEXPORTEXT void setSphere(const JER_LDOUBLE &,const JER_LDOUBLE &,const
JER_LDOUBLE &);
    DLLEXPORTEXT void setSphere();           //set sphere from rectangle coordinates
    DLLEXPORTEXT void setRectangle();       //set rectangle from sphere coordinates
    DLLEXPORTEXT void copy(const CLabLocation &); //copy members

    //*****
    //simple math methods
    DLLEXPORTEXT void subtract(const CLabLocation &); //thislocation - location
    DLLEXPORTEXT void add(const CLabLocation &);     //thislocation + location
    DLLEXPORTEXT void add(const JER_LDOUBLE &);     //scalar addition (or -1.* for subtract)
    DLLEXPORTEXT void multiply(const JER_LDOUBLE &); //scalar multiply (or 1/ for divide)
    DLLEXPORTEXT void normalize();                  //make radius to 1.0
    DLLEXPORTEXT void normalize(const JER_LDOUBLE &); //make radius to specific size

    //*****
    //simple methods for distance and angle
    DLLEXPORTEXT JER_LDOUBLE distance();           //distance from origin
    DLLEXPORTEXT JER_LDOUBLE distance(const CLabLocation &); //distance from location
    DLLEXPORTEXT JER_LDOUBLE distance(const JER_LDOUBLE &,const JER_LDOUBLE
&,const JER_LDOUBLE &); //distance from location

```

```
    DLLEXPORTE distance(const CLabLocation &,const CLabLocation &);
//distance two arbitray locations
    DLLEXPORTE distance(          //ditto
        const JER_LDOUBLE &,const JER_LDOUBLE &,const JER_LDOUBLE &,
        const JER_LDOUBLE &,const JER_LDOUBLE &,const JER_LDOUBLE &);
    DLLEXPORTE angle(const CLabLocation &);
    DLLEXPORTE angle(const JER_LDOUBLE &,const JER_LDOUBLE
&,const JER_LDOUBLE &);
    DLLEXPORTE angle(const CLabLocation &,const CLabLocation &);
    DLLEXPORTE angle(
        const JER_LDOUBLE &,const JER_LDOUBLE &,const JER_LDOUBLE &,
        const JER_LDOUBLE &,const JER_LDOUBLE &,const JER_LDOUBLE &);

};
```

```

//CLabElectrodes: recording electrodes in EMSE ELP file
//This has the electrodes locations and other information about electrode files,
// including fiducials, mm for fiducial files. The file has object initialize,
// initialize, resize, copy, read and write methods, and spherical electrodes

class CLabElectrodes : public CLabObject
{
public:
    DLLEXPORTEXT void initialize();
    DLLEXPORTEXT virtual ~CLabElectrodes();
    DLLEXPORTEXT CLabElectrodes(const CLabElectrodes &);
    //*****
    //members
    DWORD subject;           //subject number of file
    DWORD nelectrodes;       //number of electrodes
    JER_LDOUBLE headmm1,headmm2,headmm3,headmm4,headmm5,headmm6; //head
measurements for fiducial file
    CLabLocation nasion,rpa,lpa; //fiducials
    CString filename;        //filename for electrodes
    CLabLocation * locations; //locations for electrodes
    //*****
    //methods
    DLLEXPORTEXT void initialize();           //zero members, NULL for ptrs
    DLLEXPORTEXT void copy(const CLabElectrodes &); //copy object
    DLLEXPORTEXT void resize(const DWORD &); //resize and copy current
    DLLEXPORTEXT void read();                //read from filename
    DLLEXPORTEXT void read(const CString &); //read from CString
    DLLEXPORTEXT void read(CLabFile &);      //read from CLabFile
    DLLEXPORTEXT void read(const DWORD &); //use default MRI with subject number
    DLLEXPORTEXT void readfiducials();       //read fiducials from filename
    DLLEXPORTEXT void readfiducials(const CString &); //read fiducials from CString
    DLLEXPORTEXT void readfiducials(CLabFile &); //read fiducials from CLabFile
    DLLEXPORTEXT void readfiducials(const DWORD &); //use default MRI with subject number
    DLLEXPORTEXT void write();               //write from filename
    DLLEXPORTEXT void write(const CString &); //write from CString
    DLLEXPORTEXT void write(CLabFile &);     //write from CLabFile
    DLLEXPORTEXT void write(const DWORD &); //use default MRI with subject number
    DLLEXPORTEXT void writefiducials();      //write fiducials from filename
    DLLEXPORTEXT void writefiducials(const CString &); //write fiducials from CString
    DLLEXPORTEXT void writefiducials(CLabFile &); //write fiducials from CLabFile
    DLLEXPORTEXT void writefiducials(const DWORD &); //use default MRI subject number
    DLLEXPORTEXT void makesphere(const CLabLocation &); //make uniform size sphere
    DLLEXPORTEXT void makesphere(const JER_LDOUBLE &,const CLabLocation &);
    DLLEXPORTEXT void makesphere(const JER_LDOUBLE &); //make sphere size, origin set
};

```

```
//CLabElectrical: EEG data in avg file
//This has electrical data from AVG files, and other information from those
// files like subject number, no of electrodes and samples, etc. The file has
// object initialize, initialize, resize, copy, read and write methods, and CSD methods
```

```
class CLabElectrical : public CLabObject
{
public:
    DLLEXPORTEXT CLabElectrical();
    DLLEXPORTEXT virtual ~CLabElectrical();
    DLLEXPORTEXT CLabElectrical(const CLabElectrical &);

public:
    /*******
    //members
    DWORD subject;           //subject number of file if exists
    DWORD trialnumber;      //trial number if exists
    DWORD nelectrodes;      //number of electrodes
    DWORD nsamples;         //number of samples in file
    DWORD samplelength;     //length of samples
    CString filename;       //filename for electrical data
    PJER_LDOUBLE data;      //data, [0 to 127] X nsamples
    ofstream logfile;       //output to disk

    /*******
    //methods
    DLLEXPORTEXT void initialize(); //zero members, NULL for ptrs
    DLLEXPORTEXT void copy(const CLabElectrical &); //copy object
    DLLEXPORTEXT void resize(const DWORD &,const DWORD &); //resize and copy current
    DLLEXPORTEXT void read();           //read from filename
    DLLEXPORTEXT void read(const CString &); //read from CString
    DLLEXPORTEXT void read(CLabFile &); //read from CLabFile
    DLLEXPORTEXT void write();          //write from filename
    DLLEXPORTEXT void write(const CString &); //write from CString
    DLLEXPORTEXT void write(CLabFile &); //write from CLabFile

    //CSD calculation
    JER_LDOUBLE gx(const DWORD &,const DWORD &,const JER_LDOUBLE &);
    JER_LDOUBLE hx(const DWORD &,const DWORD &,const JER_LDOUBLE &);
    void calcCoefficients(const DWORD &,const DWORD &);
    PJER_LDOUBLE csdcoeffs,G,GINV,Z; //temp arrays for CSD maps
    CMatrixArithmetic * ma; //temp matrix object
    DLLEXPORTEXT void currentsourcedensity(const CLabElectrodes &); //calculate to CSD
    CLabElectrodes sphereelectrodes; //spherical electrodes
};
```

```

//CLabDipoles: Dipole files and elements, for EMSE ECD file
//This has dipole information from ECD files, and other information from those
// files like subject number, no of dipoles, and information on each dipole. The file has
// object initialize, initialize, resize, copy, read and write methods. The object also has
// methods for parameterizing a dipole into a vector (and back) and for transforming
// dipoles according to electrode arrays and sphere origins.

class CLabDipoles : public CLabObject
{
public:
    DLLExport CLabDipoles();
    DLLExport virtual ~CLabDipoles();
    DLLExport CLabDipoles(const CLabDipoles &);

public:
    /*******
    //members
    DWORD ndipoles;           //no of dipoles in file
    DWORD count,on,fixed,momentfree,state;           //unused header information
    JER_LDOUBLE radius;       //head radius (with dipole, but applies to all)
    CLabLocation sphereorigin; //sphere origin (with dipole, but applies to all)
    JER_LDOUBLE correlation,chisquare,probability; //fit (with dipole, but applies to all)
    DWORD slicestart,slicestop; //slice number in EMSE file
    JER_LDOUBLE rms;          //root-mean-square fit index
    JER_LDOUBLE residual;     //residual fit index
    CLabLocation *locations;  //dipole location
    CLabLocation *moments;    //dipole moment
    PDWORD parameters;        //parameter for each dipole (JER state)
    PDWORD states;            //state of dipole (EMSE state)
    PDWORD femindices;        //index of dipole in fem
    PDWORD models;            //type of model (EMSE: same, but can be different)
    PDWORD resistancemodels;  //model of resistance type
    PDWORD sourcemodels;      //model of shell/source type
    PDWORD momenttypes;       //moment raw or normed
    CString filename;         //filename for electrical data

    /*******
    //methods
    DLLExport void initialize();           //zero members, NULL for ptrs
    DLLExport void copy(const CLabDipoles &); //copy object
    DLLExport void resize(const DWORD &); //resize and copy current
    DLLExport void read();                 //read from filename
    DLLExport void read(const CString &); //read from CString
    DLLExport void read(CLabFile &);      //read from CLabFile
    DLLExport void write();                //write from filename

```

```
DLLEExport void write(const CString &);           //write from CString
DLLEExport void write(CLabFile &);              //write from CLabFile

//*****
//parameterize the dipoles into a vector
DLLEExport void setparametersfromdipoles(DWORD &, PJER_LDOUBLE,
PDWORD,PDWORD);
DLLEExport void setdipolesfromparameters(PJER_LDOUBLE);

//*****
//change the dipoles from one electrode space to another
DLLEExport void transformlocations(const CLabElectrodes &,const CLabElectrodes &);
DLLEExport void transformlocationsshellmodels(const CLabElectrodes &,const
CLabElectrodes &);
DLLEExport void transformlocationsshellmodels();

};
```

```

//CLabTetra: tetrahedral object for FEM model
//defines for this only, no methods
class CLabTetra {
public:
    //*****
    //members
    DWORD corner1,corner2,corner3,corner4; //four rectangular coordinates of tetra
    DWORD type; //type of data
    DWORD realisticresistanceindex; //index in resistance file for realistic model
    DWORD foursphereresistanceindex; //index for four sphere model
};

```

```

//CLabFEM: Finite Element Model
//This is for the finite element model from the 3fr files. It has object initialize, initialize,
// copy, resize, and several read/write functions. The source types can be set for the elements
// to be sources in a FEM model (coordinate file in estimation program), and the elements
// can be searched to find a source element close to a given location (find closest dipole).
class CLabFEM : public CLabObject
{
public:
    DLLEXPORTECLABFEM();
    DLLEXPORTECLABFEM virtual ~CLabFEM();
    DLLEXPORTECLABFEM(const CLabFEM &);

```

```

public:
    //*****
    //members
    DWORD nvertices,ntetra; //no of vertices and tetra
    DWORD nfourspheresources,nrealisticsources; //no of sources
    DWORD subject; //subject of this fem
    CString femfilename; //filename for 3fr, tetra, vertices
    CString realisticfilename; //filename for realistic resistance values
    CString fourspherefilename; //filename for four sphere resistance values
    CString femtypeconductancefilename; //filename for fem type conductance values
    CLabLocation * vertices; //vertex coordinates
    CLabTetra * tetra; //tetra information
    CLabLocation * midpoints; //midpoint of tetra
    PDWORD sources; //0:unused 1:source on
    PJER_LDOUBLE volumes; //volume of each tetra
    PJER_LDOUBLE realisticresistancevalues; //values for realistic resistance
    PJER_LDOUBLE foursphereresistancevalues; //..for foursphere model
    PJER_LDOUBLE femtypeconductance; //conductance values for each fem type

    //*****
    //methods

```

```

DLLEExport void initialize();           //zero members, NULL for ptrs
DLLEExport void copy(const CLabFEM &); //copy object
DLLEExport void read();                 //read from filename
DLLEExport void read(const CString &); //read from CString
DLLEExport void read(CLabFile &);      //read from CLabFile
DLLEExport void read(const DWORD &);   //use default MRI with subject number
DLLEExport void readrealistic();       //read realistic from filename
DLLEExport void readrealistic(const CString &); //read realistic from CString
DLLEExport void readrealistic(CLabFile &); //read realistic from CLabFile
DLLEExport void readrealistic(const DWORD &); //use default MRI with subject number
DLLEExport void readfoursphere();      //read foursphere from filename
DLLEExport void readfoursphere(const CString &); //read foursphere from CString
DLLEExport void readfoursphere(CLabFile &); //read foursphere from CLabFile
DLLEExport void readfoursphere(const DWORD &); //use default subject number
DLLEExport void readfemtypeconductance(); //fem type conductance values
DLLEExport void readfemtypeconductance(const CString &); //conductance values
DLLEExport void readfemtypeconductance(CLabFile &); //conductance values
DLLEExport void readfemtypeconductance(const DWORD &); //conductance values
DLLEExport void write();                //write from filename
DLLEExport void write(const CString &); //write from CString
DLLEExport void write(CLabFile &);     //write from CLabFile
DLLEExport void write(const DWORD &); //use default MRI with subject number
DLLEExport void writerealistic();       //write realistic from filename
DLLEExport void writerealistic(const CString &); //write realistic from CString
DLLEExport void writerealistic(CLabFile &); //write realistic from CLabFile
DLLEExport void writerealistic(const DWORD &); //use default MRI with subject number
DLLEExport void writefoursphere();      //write foursphere from filename
DLLEExport void writefoursphere(const CString &); //write foursphere from CString
DLLEExport void writefoursphere(CLabFile &); //write foursphere from CLabFile
DLLEExport void writefoursphere(const DWORD &); //use default subject number

//*****
//misc procedures
DLLEExport void calculatemidpoints();   //calc midpoints from tetra, vertices
DLLEExport void calculatevolumes();     //calculate volume of each tetra
DLLEExport void transformvertices(const CLabElectrodes &,const CLabElectrodes &);
//change vertices
DLLEExport void clearsources();         //clear and allocate sources
DLLEExport void setsources(const DWORD &); //set source if equal to type
DLLEExport void setsources(const CLabFEM &); //set sources from FEM coordinates
DLLEExport void setvertices(const DWORD &,PJER_LDOUBLE); //set vertices from
array

//*****
//find closest midpoint to location
DLLEExport DWORD findclosesttetra(const CLabLocation &);

```

```
    DLLEXPORTEWORD findclosesttetra(const JER_LDOUBLE &,const JER_LDOUBLE
&,const JER_LDOUBLE &);
    DLLEXPORTEWORD findclosesttetra(const DWORD &,const CLabLocation &); //of
certain type
    DLLEXPORTEWORD findclosesttetra(const DWORD &m,const JER_LDOUBLE
&,const JER_LDOUBLE &,const JER_LDOUBLE &);
    //find closest midpoint to location with source set to on
    DLLEXPORTEWORD findclosesttetrasource(const CLabLocation &);
    DLLEXPORTEWORD findclosesttetrasource(const JER_LDOUBLE &,const
JER_LDOUBLE &,const JER_LDOUBLE &);
    DLLEXPORTEWORD findclosesttetrasource(const DWORD &,const CLabLocation &);
//of certain type
    DLLEXPORTEWORD findclosesttetrasource(const DWORD &,const JER_LDOUBLE
&,const JER_LDOUBLE &,const JER_LDOUBLE &);

};
```

Some Files: ##### is a 4 digit string representing the subject number

[\\JERServer\Laboratory Data\MagneticResonanceImage\S#####](#) folder

S#####.vmi	MRI volume file
S#####.elp	EGI sensornet electrodes
S#####.wfr	Scalp wireframe
S##### Fiducials.elp	Has fiducials for subject model
S#####.wks	Workspace
S##### Cortex.wfr	GM and white matter, "cortex-type" wfr
S##### Brain.wfr	GM, WM, Ventricles, "cortex-type" wfr
S##### Scalp.rgn	Scalp region
S##### Skull.rgn	Skull
S##### Eye Socket.rgn	both eyes
S##### Eye Muscle.rgn	eye muscles and bone socket
S##### Gray Matter.rgn	gray matter
S##### White Matter.rgn	white matter
S##### CSF.rgn	ventricular and sub-arachnoid space
S##### Cortex.rgn	Gray matter and white matter
S##### Brain.rgn	Gray matter, white matter, ventricles
S##### Outside Head.rgn	Area outside of head on MRI
S##### Brain Shell.rgn	Brain shell for 3-shell model
S##### Skull Shell.rgn	Skull shell for 3-shell model
S##### Scalp Shell.rgn	Scalp shell for 3-shell model
S##### Whole Head.3fr	Tetrahedra wireframe model for whole head
S##### Gray Matter Source Volume.3fr	Source model for ECD analysis
S##### Gray Matter Eyes Source Volume.3fr	Ditto
S##### Shell Gray Matter.fwd	Forward model with 3-shell and Gray Matter
S##### Shell Gray Matter Eyes.fwd	Forward model with 3-shell, gray matter and eyes
S##### Whole Head-Gray Matter.fwd	Forward model with whole head FEM, gm
S##### Whole Head-Gray Matter Eyes.fwd	FWD with whole head FEM, gm and eyes

[\\JERServer\Laboratory Data\MagneticResonanceImage\S#####\Realistic Sources](#) folder

S##### Whole Head Vertices.bin	Vertices for the .3fr file
S##### Whole Head Tetra.bin	Tetra, including four corners, type, four-sphere and realistic indices
S##### Source Conductance.bin	Has the 10 fem types, 128 electrodes, for each gray matter and eye FEM elements
S##### Realistic Conductance.bin	Derived from Source Conductance With n-electrodes * GM and eyes sources
S##### FourSphere Conductance.bin	Has four-sphere, realistic sources forward model

S#### Conductivity Values.txt

Listing of conductance values for each FEM type.

References

- Akhtari, M., Bryant, H.C., Mamelak, A.N., Flynn, E.R., Heller, L., Shih, J.J., Mandelkern, M., Matlachov, A., Ranken, D.M., Best, E.D., DiMauro, M.A., Lee, R.R., Sutherling, W.W. (2002). Conductivities of three layer live human skull. *Brain Topography*, **14**, 151–167.
- Ary, J.P., Klein, S.A., Fender, D.H. (1981). Location of sources of evoked scalp potentials: corrections for skull and scalp thicknesses. *IEEE Transactions on Biomedical Engineering*, **28**, 447– 452.
- Awada, K.A., Jackson, D.R., Williams, J.T., Wilton, D.R., Baumann, S.B., Papanicolaou, A.C. (1997). Computational aspects of finite element modeling in EEG source localization. *IEEE Transactions on Biomedical Engineering*, **44**, 736 –752.
- Baumann, S.B., Wonzy, D.R., Kelly, S.K., & Meno, F.M. (1997) The electrical conductivity of human cerebrospinal fluid at body temperature. *IEEE Transactions on Biomedical Engineering*, **43**, 220–223.
- Benar, C.G., & Gotman, J. (2002). Modeling of post-surgical brain and skull defects in the EEG inverse problem with the boundary element method. *Clinical Neurophysiology*, **113**, 48 –56.
- Broyden, C. G. (1970a). The convergence of a class of double-rank minimization algorithms 1. general considerations. *Journal of the Inst of Mathematical Applications*, **6**, 76-90.
- Broyden, C. G. (1970b). The convergence of a class of double-rank minimization algorithms 2. the new algorithm. *Journal of the Inst of Mathematical Applications*, **6**, 222-231.
- Buchner, H., Waberski, T.D., Fuchs, M., Wischmann, H.A., Wagner, M., & Drenckhahn, R. (1995). Comparison of realistically shaped boundary- element and spherical head models in source localization of early somatosensory evoked potentials. *Brain Topography*, **8**, 137–143.
- Burger, H.C., van Milaan, J.B. (1943). Measurement of the specific resistance of the human body to direct current. *Acta Med. Scand.*, **114**, 584–607.
- DeLorme, A., Makeig, S., Fabre-Thorpe, M., & Sejnowski, T. (2002). From single-trial EEG to brain area dynamics. *Neurocomputing*, **44–46**, 1057–1064.
- Eramie, U., & Ringertz, H. (1976). Normal width of cranial sutures in the neonate and infant. *Acta Radiol Diagn*, **17**, 565–572.
- Flemming, L., Wang, Y., Caprihan, A., Eiselt, M., Haueisen, J., & Okada, Y. (2005). Evaluation of the distortion of EEG signals caused by a hole in the skull mimicking the fontanel in the skull of human neonates. *Clinical Neurophysiology*, **116**, 1141-1152.
- Fletcher, R. & Reeves, C. (1964). Function minimization by conjugate gradients. *Computer Journal*, **7**, 149-154.
- Fletcher, R. (1970). *A new approach to variable metric algorithms*. Comput. J. 13, pages 317---322.
- Foster, K.R., & Schwan, H.P. (1989). Dielectric properties of tissues and biological materials: A critical review. *Crit. Rev. Biomed. Eng.*, **17**, 25–104.
- Fox, M. & Uecker, A. (2005). Talairach daemon client. University of Texas Health Sciences Center, San Antonio, Tx. <http://ric.uthscsa.edu/projects/talairachdaemon.html>

- Gabriel, C., Gabriel, S., & Corthout, E. (1996) The dielectric properties of biological tissues: I. Literature survey. *Phys. Med. Biol.*, **41**, 2231-2249.
- Gabriel, S., Lau, R.W., Gabriel, C. (1996). The dielectric properties of biological tissues: III. Parametric models for the dielectric spectrum of tissues. *Phys. Med. Biol.*, **41**, 2271-2293.
- Gabriel, S., Lau, R.W., Gabriel, C. (1996). The dielectric properties of biological tissues: II. Measurements in the frequency range 10 Hz to 20 GHz. *Phys. Med. Biol.*, **41**, 2251-2269.
- Geddes, L.A., & Baker, L.E. (1967). The specific resistance of biological materials: A compendium of data for the biomedical engineer and physiologist. *Med. Biol. Eng.*, **5**, 271-293.
- Goldfarb, D. (1970). A family of variable metric methods derived by variational means. *Mathematical Computing*, **24**, 23-26.
- Gonçalves, S., de Munck, J.C., Verunt, J.P.A. (2001). In vivo measurement of the skull conductivity: EIT versus SEP/SEF. *Technol Health Care*, **9**, 171-172.
- Hansman C.F. (1966). Growth of interorbital distance and skull thickness as observed in roentgenographic measurements. *Radiology*, **86**, 87-96.
- Hauelsen, J., Ramon, C., Eiselt, M., Brauer, H., & Nowak, H. (1997). Influence of tissue resistivities on neuromagnetic fields and electric potentials studied with a finite element model of the head. *IEEE Transactions on Biomedical Engineering*, **44**, no. 8, 727-735.
- Hauelsen, J., Tuch, D.S., Ramon, C., Schimpf, P.H., Wedeen, V.J., George, J.S., & Belliveau, J.W. (2002). The influence of brain tissue anisotropy on human EEG and MEG. *Neuroimage*, **15**, 159-166.
- Huiskamp, G., Vroeijsstijn, M., van Dijk, R., Wieneke, G., & van Huffelen, A.C. (1999). The need for correct realistic geometry in the inverse EEG problem. *IEEE Transactions on Biomedical Engineering*, **46**, 1281-1287.
- Huizenga, H.M., & Molenaar, P.C.M. (1994). Estimating and testing the sources of evoked potentials in the brain. *Multivariate Behavioral Research*, **29**, 237-262.
- Johnson, M.H., de Haan, M., Oliver, A., Smith, W., Hatzakis, H., Tucker, L.A., & Csibra, G. (2001). Recording and analyzing high-density event-related potentials with infants using the Geodesic Sensor Net. *Developmental Neuropsychology*, **19**, 295-323.
- Jung, T.P., Makeig, S., Westerfield, M., Townsend, J., Courchesne, E., & Sejnowski, T.J. (2001b). Analysis and visualization of single-trial event-related potentials. *Human Brain Mapping*, **14**, 166-185.
- Jung, T.P., Makeig, S., Humphries, C., Lee, T.W., McKeown, M.J., Iragui, V., & Sejnowski, T.J. (2000a). Removing electroencephalographic artifacts by blind source separation. *Psychophysiology*, **37**, 163-178.
- Jung, T.-P., Makeig, S., Lee, T.-W., McKeown, M.J., Brown, G., Bell, A.J., & Sejnowski, T.J. (2000b). Independent component analysis of biomedical signals. In P. Pajunen & Karhunen, K. *Proceedings of the 2nd International Workshop on Independent Component Analysis and Source Separation* (pp. 633-644). Helsinki, Finland.
- Jung, T.P., Makeig, S., McKeown, M.J., Bell, A.J., Lee, T.W., & Sejnowski, T. (2001a). Imaging brain dynamics using independent component analysis. *Proceedings of the IEEE*, **89**, 1107-

1122.

Kavanagh, R.N., Darcey, T.M., Lehmann, D., & Fender, D.H. (1978). Evaluation of methods for three-dimensional localization of electrical sources in the human brain. *IEEE Transactions on Biomedical Engineering*, **25**, 421– 429.

Lancaster, J.L., Summerlin, J.L., Rainey, L., Freitas, C.S., & Fox, P.T. (1997). The Talairach Daemon, a database server for Talairach Atlas Labels. *Neuroimage*, **5(4)**, S633.

Lancaster, J.L., Woldorff, M.G., Parsons, L.M., Liotti, M., Freitas, C.S., Rainey, L., Kochunov, P.V., Nickerson, D., Mikiten, S.A., & Fox, P.T. Automated Talairach Atlas labels for functional brain mapping. *Human Brain Mapping*, **10**, 120-131, 2000.

Lee, T.W., Girolami, M., & Sejnowski, T.J. (1999). Independent component analysis using an extended infomax algorithm for mixed subgaussian and supergaussian sources. *Neural Computing*, **11**, 417– 441.

Makeig, S., Bell, A.H., Jung, T.P., & Sejnowski, T.J. (1996). Independent component analysis of electroencephalographic data. *Advances in Neural Information Processing Systems*, **8**, 145 – 151.

Makeig, S., Jung, T.P., Bell, A.H., Ghahremani, D., & Sejnowski, T.J. (1997). Blind separation of auditory event-related brain responses into independent components. *Proceedings of the National Academy of Sciences*, **94**, 10979– 10984.

Malmivuo J., & Plonsey R.(1995). *Bioelectromagnetism*, Oxford University Press, New York.

Marin, G., Guerin, C., Baillet, S., Garnero, L., & Meunier, G. (1998): Influence of skull anisotropy for the forward and inverse problem in EEG: simulation studies using FEM on realistic head models. *Human Brain Mapping*, **6**, 250 –269.

Martinez, A., Anillo-Vento, L., Sereno, M.I., Frank, L.R., Buxton, R.B., Dubowitz, D.J., Wong, E.C., Hinrichs, H., Heinze, H.J., & Hillyard, S.A. (1999). Involvement of striate and extrastriate visual cortical areas in spatial attention. *Nature Neuroscience*, **2**, 364 –369.

Meijs, J.W., Weier, O.W., Peters, M.J., & van Oosterom, A. (1989). On the numerical accuracy of the boundary element method. *IEEE Transactions on Biomedical Engineering*, **36**, 1038 – 1049.

Nelder, J.A., & Mead, R. (1965). A simplex method for function minimization. *Computer Journal*, **7**, 308–313.

Oostendorp, T.F., Delbeke, J., & Stegeman, D.F. (2000). The conductivity of the human skull: results of in vivo and in vitro measurements. *IEEE Transactions on Biomedical Engineering*, **47**, 1487–1492.

Oostenveld, R., & Oostendorp, T.F. (2002). Validating the boundary element method for forward and inverse EEG computations in the presence of a hole in the skull. *Human Brain Mapping* **17**, 179-192.

Patterson, S.J., Badridze, N., Flax, J.F., Liu, W., & Benasich, A.A. (2004). A method for structural MRI scanning of non-sedated infants. Paper presented at the International Conference on Infancy Studies, Chicago, IL.

Perrin, F., Bertrand, O., & Pernier, J. (1987). Scalp current density mapping: value and

- estimation from brain data. *IEEE Transactions on Biomedical Engineering*, **34**, 283–288.
- Perrin, F., Pernier, J., Bertrand, O., & Echallier, J.F. (1989). Spherical splines for scalp potential and current density mapping. *Electroencephalography and Clinical Neurophysiology*, **72**, 184–187.
- Polak, E. & Ribière, G. (1969). Note sur la convergence de directions conjuguées. *Rev. Francaise Informat Recherche Operationelle*, **3e Année 16**, 35-43.
- Polak, E. (1971). *Computational methods in optimization*. New York: Academic Press.
- Powell, M.J.D. (1964). An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Computer Journal*, **7**, 155-162.
- Press, W.H., Flannery, B.P., Teukolsky, S.A., & Vetterling, W.T. (1992). Numerical recipes in C: The art of scientific computing (Second edition). Cambridge Press.
- Press, W.H., Teukolsky, S.A., Vetterling, W.T., & Flannery, B.P., (2002). Numerical recipes in C++: The art of scientific computing (Second edition). Cambridge Press.
- Reynolds, J.D., & Richards, J.E. (2005). Familiarization, attention, and recognition memory in infancy: An ERP and cortical source localization study. *Developmental Psychology*, **41**, 598-615.
- Richards, J.E. (2003). Cortical sources of event-related-potentials in the prosaccade and antisaccade task. *Psychophysiology*, **40**, 878-894.
- Richards, J.E. (2004). Recovering dipole sources from scalp-recorded event-related-potentials using component analysis: Principal component analysis and independent component analysis. *International Journal of Psychophysiology*, **54**, 201-220
- Richards, J.E. (2005). Localizing cortical sources of event-related potentials in infants' covert orienting. *Developmental Science*, **8**, 255–278.
- Richards, J.E., & Anderson, D.R. (2004). Attentional inertia in children's extended looking at television. *Advances in Child Development and Behavior*, **32**, 163-212
- Richards, J.E., & Hunter, S.K. (2002). Testing neural models of the development of infant visual attention. *Developmental Psychobiology*, **40**, 226-236.
- Ridders, C.J.F. (1982) Technical Note: Accurate Computation of $F'(x)$ and $F(x)F'(x)$. *Advances in Engineering Software*, **4 (2)**, 75–76.
- Roth, B.J., Balish, M., Gorbach, A., & Sato, S. (1993). How well does a three-sphere model predict positions of dipoles in a realistically shaped head? *Electroencephalography and Clinical Neurophysiology*, **87**, 175–184.
- Scherg, M. (1990). Fundamentals of dipole source potential analysis. In F. Grandon, M. Hoke & G.L. Romani (Eds.), *Auditory evoked magnetic fields and potentials* (Vol. 6, pp. 40– 69). Basel: Karger.
- Scherg, M. (1992). Functional imaging and localization of electromagnetic brain activity. *Brain Topography*, **5**, 103– 111.
- Scherg, M., & Picton, T.W. (1991). Separation and identification of event-related potential components by brain electrical source analysis. In C.H.M. Brunia, G. Mulder & M.N. Verbaten

- (Eds.), *Event-related brain research* (pp. 24–37). Amsterdam: Elsevier Science Publishers.
- Shanno, D.F. (1970). Conditioning of quasi-Newton methods for function minimization. *Mathematical Computing*, **24**, 647–656.
- Stok, C.J. (1987). The influence of model parameters on EEG/MEG single dipole source estimation. *IEEE Transactions on Biomedical Engineering*, **34**, 289–296.
- Talairach, J., & Tournoux, P. (1988). *Co-planar stereotaxic atlas of the human brain*. New York: Thieme Medical Publishers.
- Thevenet, M., Bertrand, O., Perrin, F., Dumont, T., & Pernier, J. (1991). The finite element method for a realistic head model of electrical brain activities: preliminary results. *Clin Phys Physiol Meas*, **12** (Suppl), 89–94.
- van Burik, M.J., & Peters, M.J. (2000). Estimation of the electric conductivity from scalp measurements: feasibility and application to source localization. *Clin Neurophysiol*, **111**, 1514–1521.
- Vanrumste, B., Van Hoey, G., Van de Walle, R., D’Have, M., Lemahieu, I., & Boon, P. (2000). Dipole location errors in electroencephalogram source analysis due to volume conductor model errors. *Med Biol Eng Comput*, **38**, 528–534.
- Yvert, B., Bertrand, O., Echallier, J.F., & Pernier, J. (1995). Improved forward EEG calculations using local mesh refinement of realistic head geometries. *Electroencephalography and Clinical Neurophysiology*, **95**, 381–392.
- Yvert, B., Bertrand, O., Echallier, J.F., & Pernier, J. (1996). Improved dipole localization using local mesh refinement of realistic head geometries: an EEG simulation study. *Electroencephalography and Clinical Neurophysiology*, **99**, 79–89.
- Yvert, B., Bertrand, O., Thevenet, M., Echallier, J.F., & Pernier, J. (1997). A systematic evaluation of the spherical model accuracy in EEG dipole localization. *Electroencephalography and Clinical Neurophysiology*, **102**, 452–459.