

Table of Contents

INTRODUCTION.....	1
STARTING THE PROGRAM	3
IMPORT MEG DATA.....	4
.....4D Neuroimaging binary data, auxiliary files:	6
.....4D Neuroimaging export utility.....	7
.....Generic Data Import, auxiliary files:	8
.....File Descriptions:.....	10
.....MEG binary data file	10
.....Generic Data File	10
.....gen_header.txt.....	11
.....gen_sen_loc.txt	11
.....gen_hs_file.....	11
IMPORT MRI DATA.....	13
.....Importing Volumetric MRI Data	13
.....Importing MRI Slice Sequence Data	14
.....Creating 3D Head from MRI data	16
.....MEG/MRI Coregistration	21
.....Coregistration steps.....	22
.....What can go wrong	25
DATA PLOTS.....	26
.....Channel overlay plots	28
.....Contour Plots	29
.....Sensor layout plots.....	29
.....Vertical layout plots.....	29
ADAPTIVE NOISE FILTERING.....	31
CORTICAL MODEL.....	32
.....Steps required to create:.....	33
.....Defining the outer limit surface	32
.....Specifying MRI amplitudes of cortical gray matter	36
.....Distributing source points in gray matter	37

FORWARD MODEL (GAIN MATRIX CALCULATIONS)	38
CORTICAL SOURCE IMAGING	41
.....Module Overview:	41
.....Source Imaging Graphical Interface:	42
.....Imaging Techniques:	
.....SCD METRIC (single equivalent dipole goodness of fit metric)	44
.....2DII (Two Dimensional Inverse Imaging)	45
.....Multi-Resolution FOCUSS	51
.....Data Types available for cortical source imaging:	54
.....LOADING A SOLUTION	57
.....SOLUTION VIEWER	58
APPENDIX:	
.....Data Structures:	63
.....Meg/EEG Data	63
.....Source Model	64
.....Forward Model	65
.....Solution	66
.....Technical References	67
.....Matlab Path	69
.....Generic data file formats	70
.....Path for subject data files	73
.....Required files to import data and MRI	73

Acknowledgement: Research supported by NIH/NINDS Grant RO1-NS30914 72

Introduction

MEG Tools is a package of Matlab m-files with a mouse driven graphical interface for MEG data visualization and integration of MEG and MRI data for cortical source imaging. MEG data can be displayed in a variety of formats including contour plots and vertically spaced time series data including EEG (if acquired). To facilitate cortical source imaging, small segments of data can be selected and saved as separate data files. The available cortical imaging methods are single current dipole, Two Dimensional Inverse Imaging (2DII) and Multiresolution FOCUSS (M-FOCUSS). Also, special data transformations are available that can be applied to minimize the influence of noise and enhance imaging of on/off source responses.

The 2DII and M-FOCUSS techniques utilize an initialization structure which is recursively modified to be a least squares match to the MEG data. During each iterative step, a source basis of eight composite source structures is constructed by a sequential decomposition of the full cortical model. Compared to a single value decomposition (SVD) of the Forward gain matrix, source solutions based on this multiresolution source basis are overdetermined and relatively insensitive to noise in the data and magnetic fields from sources outside the region of interest, (such as heart). Also, 2DII and M-FOCUSS source solutions are generated without calculating a matrix inverse and avoid problems associated with excessively weighting low amplitude eigenvalue components of the SVD of the Gain matrix.

MRI data is integrated into these MEG imaging techniques by generating a cortical model with the 3000 sources distributed to match the location of gray matter identified in the subject's MRI. In addition, the initialization structure used in 2DII and M-FOCUSS can be used to introduce prior knowledge of cortical source activity that may be derived from fMRI, etc. Finally, this program includes 3D visualization displays for the source solutions superimposed on the MRI slice data and head model. Full sequences of solution images can be combined into movies for display in this program or imported as image.bmp files into other imaging applications.

A rudimentary EXPORT utility for the commercially available software, ASA, is available as a menu option. This and other export formats will be developed in the future. However, the data, forward model, source model and solution source structures created by this program are easily accessible from the Matlab command line or can be read from the file on disk. Automated export to other imaging software packages will eventually be developed.

This program was originally developed for 4D neuroimaging derived data with the data import capabilities expanded to include other systems. However, these additional import capabilities have not been extensively tested and the author of the program should be contacted for help if problems are encountered.

The MRI import utility can read a volumetric data file or slice sequence data. This utility includes a module to construct the 3D head and coregister the MRI and MEG coordinate systems. In addition, for MEG studies without a corresponding MRI, any other MRI can be used with the x, y, and z dimensions scaled separately such that the MRI data closely matches the digitized head shape of the subject. Major landmark structure should match closely after this rescaling. In addition, data from multiple individuals would be rescaled to a common head size and displayed on the same cortex.

To avoid chaos and facilitate program improvements, the m-files of this program may be altered by users of this program for their own benefit provided these alterations are not distributed to others without the author's (this program) permission. However, components of the program are extensively interrelated. Therefore, code changes may create unintended changes in program function. Some m-file connection diagrams have been included and more will be available in the future.

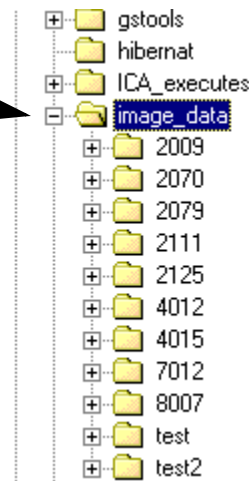
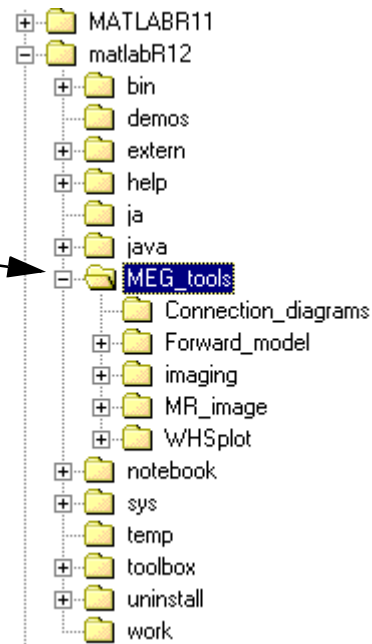
Installation

The MEG IMAGING TOOLS for MATLAB consist of MATLAB script.m files organized in 4 directories:

MEG_tools

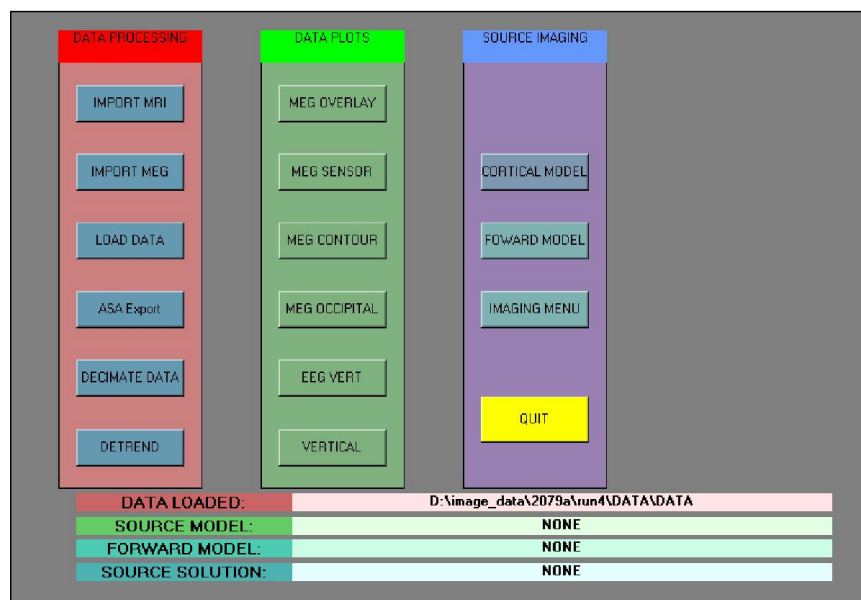
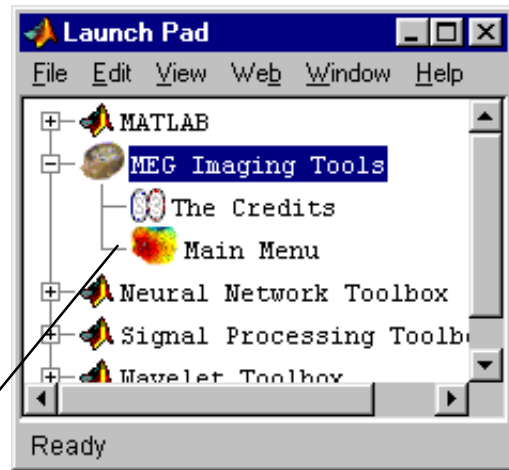
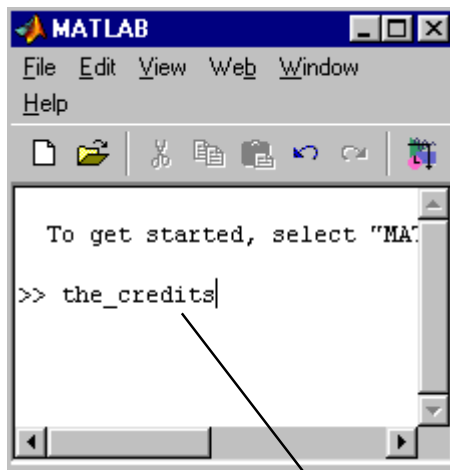
- 1.) Forward_model
 - 2.) imaging
 - 3.) MR_image
 - 4.) WHSplot
- Connection_diagrams

- Install the **MEG_tools** directory into the matlab directory , (MATLABR11 or matlabR12). Next, update the MATLAB path to include the MEG_tools directory and all the sub-directories, (See APPENDIX MATLAB PATH).
- Create a master directory to contain the subject data directories. The name of this directory must be “**image_data**”. Only one image_data directory can exist and it cannot be a subdirectory. However, it can be on any hard drive, such as (C:\image_data or D:\image_data).
- Set WINDOWS SCREEN RESOLUTION to 800 x 600 pixels to avoid hidden and misplaced control buttons and edit boxes. The program will work at other resolutions. However, the graphical interface will not always be correct.



STARTING THE PROGRAM

- Start MATLAB
- Type, whstart2 followed by <ENTER> at the MATLAB prompt.
- Alternatively, Type , the_credits <ENTER> . Then the START button on the credits window.
- (For MATLAB 6) After MEG_tools has been installed and the PATH to these files set. Restart MATLAB, The MEG_tools will be an entry on the LAUNCH PAD MENU as shown below. This will create MATLAB windows 1 and 2. Window 1 will be blank and is reserved for menu display by the program. Window 2 is the main imaging MENU :



Import MEG Data

This module imports MEG data and MEG coil array parameters and locations relative to the subject's head coordinate system.

Types of data imported:

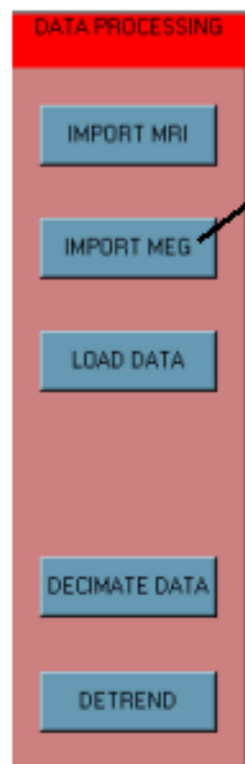
- 4D Neuroimaging binary data from SUN SPARC Ultra 60 workstation

- Generic binary data files created by other MEG systems or analysis software.

- Generic ascii data created by other MEG systems or analysis software.

Data files and Sensor array coordinate files are described.

4D Neuroimaging utilities for automatic creation of data and sensor array descriptor files are provided.



Import Data Parameters

MRI Machine Format: IEEE big-endian

Type of Data: Binary Data File

☒ 4D binary data

File Header Size (Bytes): 0

IMPORT QUIT

Default data import menu:
The 4D binary data checkbox is selected.

Import Data Parameters

MRI Machine Format: IEEE big-endian

Binary Data Format: Floating, 32 bit

Type of Data: Binary Data File

☐ 4D binary data

File Header Size (Bytes): 0

IMPORT QUIT

Generic binary data import:
After deselecting the 4D import option. The menu items are selected to conform to the data.

Import Data Parameters

Type of Data: ASCII Data File

Data Matrix ROWS are: Data channels

File Header Size (Bytes): 0

IMPORT QUIT

ASCII data file type menu:

The menu changes to allow the description of the data matrix.

Choices are:

Matrix rows = Data channels

Matrix rows = Time slices

IMPORT:

The file selection menu appears. Use this menu to select the MEG data file, (Not the header, sen_loc, or hsfile).



Import MEG

This utility is used to import data into the MEG_tools program. The actual data file can be one of the following:

1. 4D neuroimaging binary data (this is default import). This import utility is presently configured to import 4D neuroimaging formatted data as created on SUN ultra60 workstations. These data are easily transferred to the PC via ftp protocol.
2. Binary MEG/EEG data acquired with any other MEG system. (You must know the data format). These must be stored as a sequence of data time slices rather than as sequential channels of data. The type of MEG system is specified in a secondary configuration file, (gen_sen_loc.txt).
3. Data matrix in ASCII format with either rows or columns corresponding to the sequential time increments. This format should only be used for small time segments of data such as evoked response data.

Auxiliary Data Description Files

Three additional MEG data description files are required. This program includes a set of utilities to create these files for 4D neuroimaging data. However, for data imported from other MEG systems, a different set of generic datafile formats has been developed.

4D Neuroimaging binary data, auxiliary files:

1. header.txt (channels id, location in time slice, and conversion factor to Tesla units).
2. sen_loc.txt (location and orientation of the sensors, electrodes in the subject coordinate system.)
3. hsfile.txt (location of nasion, left and right pre-auricular points and hundreds/ thousands of points used to describe the full extent of the head geometry.)

The header, sen_loc, and hsfile are created by 4D helper functions implemented on the SUN ultra 60 workstation. We have automated the generation of these files by creating an executable UNIX script file, “get_info”. After installing this file, the 4D Neuroimaging software creates a drop down menu option that causes the header, sen_loc, and hsfile files to be produced for the “posted” data file on the SUN workstation. These files and the 4D binary data file are transferred to the PC via ftp protocol. (SEE 4D utilities folder README.TXT for installing and using this UNIX utility). This file is included with the MEG_tools installation files. (Included with the MEG TOOLS for Matlab program are the **4D software utilities** that contain the “Get_MEG_tools_info” file and instructions to install it in the 4D Neuroimaging software.)

Specifics of 4D utilities implemented in the “get_info” application:

4D helper function >>> output file

convert >>> header

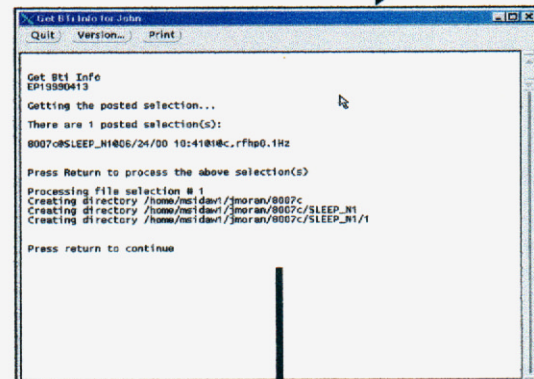
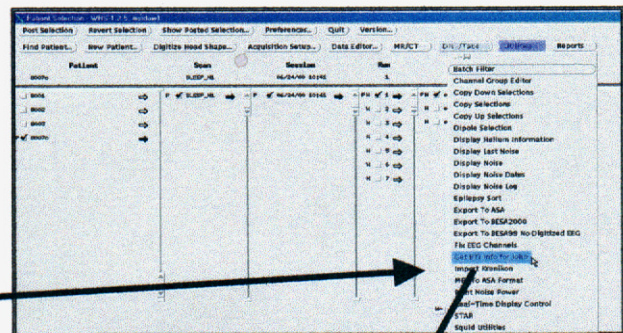
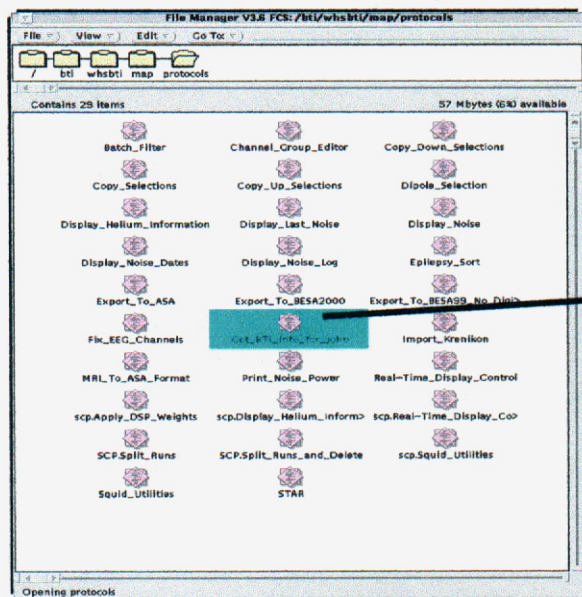
print_hs_file >>> hsfile

config_printer >>> sen_loc

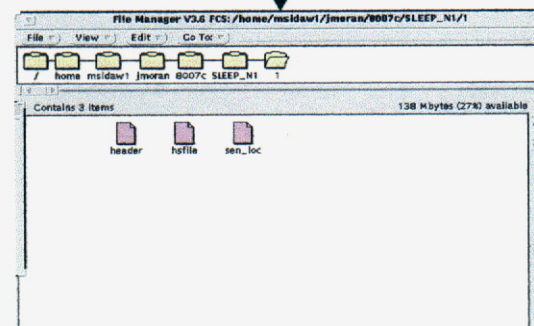
The output files are ASCII files that can be examined with WORDPAD or NOTEPAD etc. See examples of these files in the DEMO DATA that can be downloaded with the MEG TOOLS for Matlab software.

4D Neuroimaging export utility

The “Get_MEG_Tools_info” utility supplied in 4D_menu_utilities must be installed in the protocols directory of the 4D Neuroimaging software. This creates a “get_info” option on the utilities tab. Activating this option creates the header, hsfile, and sen_loc files for all posted files. For each of the posted files the helper files are placed in a subject/scan/run directory that is created by the “get_info” utility. These are ASCII files are easily transferred to the PC via ftp. Also, a link to the corresponding binary data file can be copied to this location. Therefore, during ftp transfer of the subject/scan/run directory, all the necessary files are transferred together. Prior to use, you must create a directory, /home/msidaw1/all_subjects_info or modify line 11 of the Get_MEG_Tools_info file with a text editor to a different directory location. This file must be made executable, use the `chmod +x Get_MEG_Tools_info` command to accomplish this task.



After installing the 4D “get_info” utility in the protocols subdirectory a menu option is added to the patient selection main menu> utilities of the 4D analysis application running on the UNIX workstation. This utility creates, in the jmoran directory, the appropriate subject/scan/run directories plus the hs_file, header, and sen_loc files for all posted data files. These 3 files plus the binary data file, MRI data and coregistration matrix are copied to the PC using ftp data transfer.



Rather than utilizing the 4D software menu option described above, the following 4Dneuroimaging helper functions can be implemented individually to create the desired files (Consult 4D Neuroimaging on the specific unix shell command line information required).

As previously described, the UNIX executable script to execute these functions as part of the 4D imaging software > utilities menu is included on in the installation files in sub-directory, 4D_menu_utilities. To avoid problems, 4Dneuroimaging should be contacted to include this as a utility option in their software on your system. In our implementation of this utility script, the header, sen_loc, and hsfile are created for each "posted" data file and put in the appropriate subdirectory, (/home/msidaw1/jmoran/subject_id/scan/run/), that is created as necessary on the UNIX workstation running the 4D Neuroimaging software.

Note: If multiple files for the same **run** are processed then the files for previously processed runs are overwritten. A solution to this problem is to move the 3 files to a separate runfile directory before processing each data file.

Generic Data Import, auxiliary files:

- 1) **gen_header.txt** (slightly different from the header.txt. This format should not be used with 4D data.)
- 2) **gen_sen_loc.txt** (contains the same information as sen_loc.txt plus identification of sensor type, (magnetometer, or axial or planer gradiometer), plus individual coil weight factor to be used in forward solutions.) It is important to see further description and examples in the appendix.
- 3) **gen_hsfile.txt** (the same information as in hsfile.txt. It is important to include a large number of head digitalization points. These digitalization points are an important part of the coregistration of MEG to the MRI data in this program and can be used in the calculation of the best fitting local sphere in forward model calculations. If no digitalization points are available, the nasion, left and right PA points should be reused as digitalization points to avoid program failure of the MEG/MRI coregistration and head model creation module of this program.

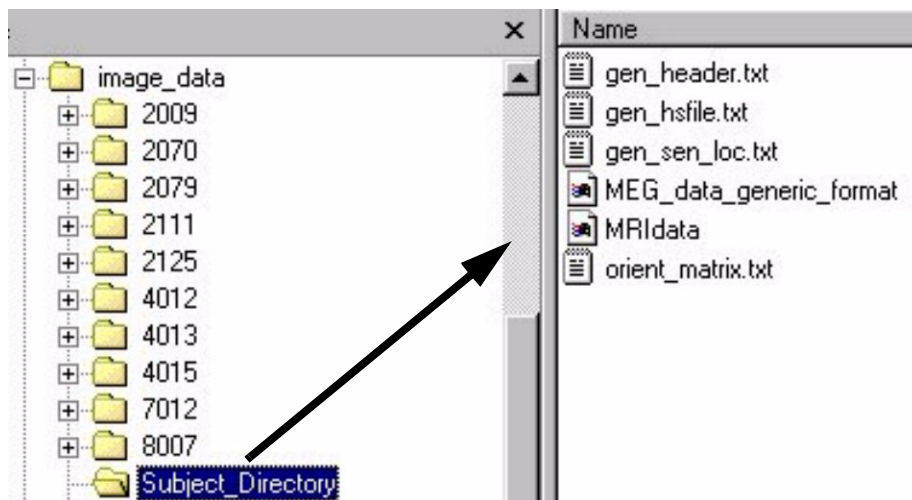
The gen_header, gen_sen_loc, and gen_hsfile must be supplied by the user. The exact format for these files is described in the APPENDIX, GENERIC FILE FORMATS and examples are supplied in the DEMO DATA that can be downloaded with this program. These formats allow MEG data from other MEG systems to be imported for imaging.

For multiple data sets, each data file and the corresponding header and sen_loc files may be put in a sub-directory such as:

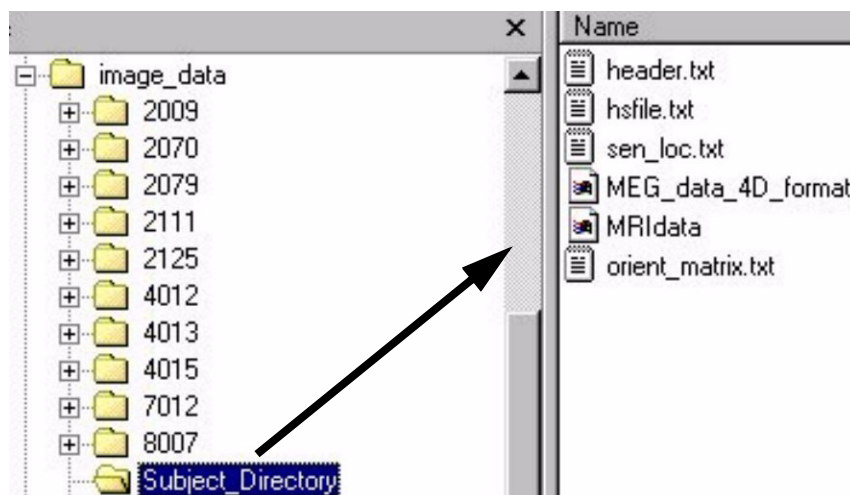
```
<drive>:\image_data\Subject_Directory\study_1\the_binary_data1
<drive>:\image_data\Subject_Directory\study_1\header.txt
<drive>:\image_data\Subject_Directory\study_2\sen_loc.txt
<drive>:\image_data\Subject_Directory\study_2\the_binary_data2
<drive>:\image_data\Subject_Directory\study_2\header.txt
<drive>:\image_data\Subject_Directory\study_2\sen_loc.txt
<drive>:\image_data\Subject_Directory\hsfile
<drive>:\image_data\Subject_Directory\MRI_binary_file
<drive>:\image_data\Subject_Directory\orient_matrix.txt
```


Note that the head coordinate points and head shape file, (hsfile), applies to all data sets and is in the main subject directory.

(BELOW): The location of the required data files prior to importing a generic (binary or ASCII) data file. If more than one data is imported, each of these MEG data files and the corresponding header, and sen_loc file should be placed in a separate subdirectory. The hsfile contains coordinate and head shape points common to all data files, and should be left in the main subject directory.



(BELOW): The location of the required data files to import 4D data.



FILE DISCRIPTIONS

MEG binary data file

This is the aquired data file or subsequent processed data file generated by the 4D software. This file should be placed in the DATA sub-directory of the subject directory prior to import. Note: The binary data file and the other three files must be located in the subject directory as previously discribed prior to selecting IMPORT DATA from the MEG TOOLS for Matlab control pannel.

During file import, data files, (Data1_1,Data1_2, ... , Data1_15, ...), are created. Each contains an EPOCH data structure which contains up to 10 seconds of data from the complete time sequence of data. After importing data, the data is not load for viewing. The LOAD DATA button is used to activate a file open menu to load the data. Any of the Data files may be selected to load all of the data. This segmentation is used to facilitate rapid processing of the data without requiring massive RAM memory. During plotting and other processing these files are reassembled as required.

Generic Data File

To facilitate the use of our MEG cortical source imaging programs we have developed a generic import format with the following features and requirements.

The time series data file: After pressing the IMPORT DATA button a menu will appear which will allow the user of the program to select ASCII or binary.

If ASCII is chosen, then the choice of TIME = row increments or TIME = column increments is presented. The ASCII data file must not contain letters. When viewed with a text editor it must appear as a matrix with each column of numbers separated from the next by one or more blank spaces.

If the choice is binary then the binary data format must be selected on the panel menu:

local machine format

IEEE floating point with little-endian byte ordering.

IEEE floating point with big-endian byte ordering

VAX D floating point and VAX ordering

VAX G floating point and VAX ordering

Cray floating point with big-endian byte ordering

IEEE floating point with little-endian byte ordering and 64 bit long data type

IEEE floating point with big-endian byte ordering and 64 bit long data type.

In Addition, the data element size must be specified in the menu options:

unsigned character, 8 bits.
signed character, 8 bits.
integer, 8 bits.
integer, 16 bits.
integer, 32 bits.
integer, 64 bits.
unsigned integer, 8 bits.
unsigned integer, 16 bits.
unsigned integer, 32 bits.
unsigned integer, 64 bits.
floating point, 32 bits.
floating point, 32 bits.
floating point, 64 bits.
floating point, 64 bits.

If the data is binary it must be stored in the data file in slice order. (Data channels for time T, Data channels for time $T + \Delta T$, etc.).

gen_header.txt

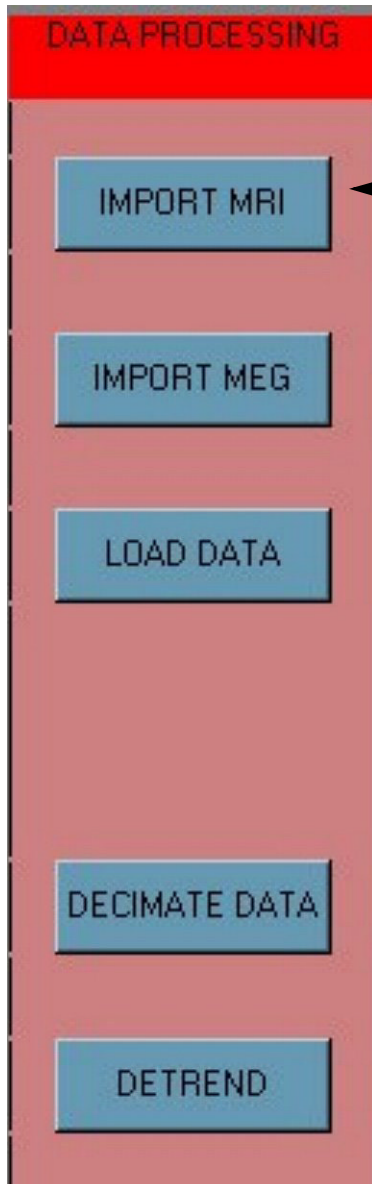
This file is used to associate each specific MEG, EEG, other time series data channel with the **order in which it is stored** in the binary or ASCII data file. Each MEG channel must be specified with the capital letter A#, (Example: A1, A33, A2, etc.). Each EEG channel must be specified with the capital letter E#, (Example: E1, E2, etc.). Other letters of the alphabet can be used for other channels in the data file such as T#, R#, (Example: T1, R1 for TRIGGER and RESPONSE channels in all 4D data. Assignment of a letter-number combination is required.) See the example file in the GENERIC_DATA_FORMAT directory that is included with this program installation. Also, this file contains conversion factors to convert the data amplitudes to units of TESLA for MEG data or VOLTS for EEG data.

gen_sen_loc.txt

This file Associates the channels described in the “gen_header.txt” file with their corresponding locations in the Subject coordinate system. Provisions for describing each sensor coil of magnetometers and gradiometers is included. It is important that the channel numbers in this file match those in the “gen_header.txt” file. Therefore, it is important to examine the examples of this file in the included demo data and in the appendix.

gen_hs_file

This file is used to coregister the MEG data to the MRI data and is not required until the IMPORT MRI utility is used. This file has a simple format that must be used exactly as demonstrated in the DEMO data examples.



IMPORT MRI

(A) Imports volumetric or MRI slice sequence files into MEG_tools structure.



(B) Creates 3D head of subject for use in forward modeling of MEG data and cortical source solutions. Also, this MRI derived head is used to display the relationship of the solution and anatomical structures.



(C) Coregisters the MRI and MEG coordinate system using the 3D MRI derived head and the index and head digitalization points obtained during a MEG study.

Import MRI Data

After pressing the <IMPORT MRI> button, the first of the first of three MRI data processing modules is activated. A data format menu appears such that the format of the MRI volumetric data file or sequence of MRI slices can be specified. Then the MRI data is converted these MRI data into three volumetric slice sequences in double float format required in matlab.

The following directories are created:

- (1) image_data\Subject_Directory\mrdata\axial
- (2) image_data\Subject_Directory\mrdata\coronal
- (3) image_data\Subject_Directory\mrdata\sagital

Each contains a full sequence of isotropic image files. While this duplication requires a large amount of disk space it greatly increases subsequent matlab imaging performance. However, these files can be deleted after creating the cortical source models that are used for MEG imaging. Each SOURCE_MODEL extracts a subset of the images in these directories that are required for visualizing the solution in three dimensions.

Importing Volumetric MRI Data

MRI data imported as a single file must be MRI volumetric data. In addition to the MRI data file an orientation file must be supplied. This file specifies the transformation matrix for converting from the row, column and slice coordinates of the MRI data to the x, y, and z coordinates of the subject. (The center of the subject coordinate system is equidistant from the left and right pre-auricular points. The +X axis passes through the nasion and the +Z axis is directed toward the top of the head perpendicular to the plane of the left, right PA, and Nasion points. The +Y axis points to the subject's left.) However, this orientation matrix does not have to be exact. After the MRI data is imported the program proceeds to MRI/MEG coregistration. This coregistration utility is used to correct the orientation matrix, as well as the final scaling of the MRI pixel dimensions in millimeter units. The MRI data file and orient_matrix.txt file must be placed in the Subject_Directory prior to using the import command.

The required import files and their locations are:

- (1) <drive>:\image_data\Subject_Directory\orient_matrix.txt (the file **must** have this name).

Example, (exact orientation matrix = coregistration matrix):

```
.002  -0.996  0.078  8.715
0.999  0.002  -0.000  6.990
0.000  0.078  0.996  51.041
0.000  0.000  0.000  1.000
```

Example, (approximate orientation matrix that will be refined during MEG/MRI coregistration):

```
0.000  -1.000  0.000  0.000
1.000  0.000  0.000  0.000
0.000  0.000  1.000  0.000
0.000  0.000  0.000  1.000
```

- (2) <drive>:\image_data\Subject_Directory\MRI_volume_file

Options are available on the import MRI menu, (shown on the next page), for specifying the

import format of this file and for describing the ROW, COLUMN, and SLICE pixel scaling, (mm/pixel) of the files that are imported. A tentative estimate of the pixel dimensions must be supplied before the program will allow the MRI data import to proceed. This is accomplished by altering the default pixel dimensions on the import menu and checking the APPLY box. However, during the coregistration phase of the MRI data import these pixel dimensions can be altered or fine tuned to optimize the MEG/MRI coregistration.

Importing MRI Slice Sequence Data

This requires the orient_matrix.txt file plus a subdirectory containing the full set of MRI slice files. Both of these must be placed in the Subject_Directory prior to using this import command. In addition, the row and column dimensions of the MRI slice pixels and slice thickness must be supplied. The MEG_tools program performs a nearest neighbor interpolation to generate the coronal, axial, and sagittal volumetric data with the desired row, column, and slice dimensions. The required import files and their locations are:

(1) <drive>:\image_data\Subject_Directory\orient_matrix.txt (the file **must** have this name).

Example:

```
.002  -0.996  0.078  8.715
0.999  0.002  -0.000  6.990
0.000  0.078  0.996  51.041
0.000  0.000  0.000  1.000
```

(2) <drive>:\image_data\Subject_Directory\MRI_slice_directory\MRI_slice_files
or <drive>:\image_data\Subject_Directory\MRI_volumetric_data_file

Alternatively, the original slice data files can be imported. The slice files should be in a separate slice series directory within the Subject_Directory, (SD = Subject_Directory). Each slice must have an embedded (integer or decimal) number representative of it's position in the slice sequence. File import is initiated by clicking on any file in the slice sequence. The program has limited ability to recognize and extract the numbering sequence. Therefore, if the data is not imported properly, change the file names to include a simple slice number identification, (example: slice1,slice2, etc.). Two sets of pixel dimensions must be supplied before the program will allow data import. First, the ROW, COLUMN and SLICE pixel dimensions of the MRI slice sequence files must be specified by altering the import menu default values and checking the APPLY box. Usually, the ROW and COLUMN dimensions are equal and the SLICE thickness larger. Next, the same must be done for the final (isotropic) pixel dimensions of the imported data. In general, these ROW, COLUMN and SLICE pixel dimensions are set equal such that the slice sequence is transformed to isotropic MRI volumetric data. A nearest neighbor interpolation method is used to set individual pixel amplitudes during this transformation to MRI volumetric data.

(SEE MENU FIGURES ON THE NEXT PAGE)

Importing a MRI volumetric file.
Must check pixel dimensions
APPLY box.

DATA PROCESSING

IMPORT MRI

IMPORT MEG

LOAD DATA

ASA Export

DECIMATE DATA

DETREND

Import Volumetric MRI Data and Transformation Matrix

MRI Machine Format: IEEE, big-endian

MRI Import Format: integer, 8 bits

MRI Data Type: Volumetric File

MRI Image dimensions (pixels): Hor. 256 Vert. 256

MRI Pixel Dimensions

row	column	slice
1	1	1

File Header Size, (Bytes): 0

IMPORT QUIT

Import Volumetric MRI Data and Transformation Matrix

MRI Machine Format: IEEE, big-endian

MRI Import Format: integer, 8 bits

MRI Data Type: Volumetric File

MRI Image dimensions (pixels): Hor. 256 Vert. 256

MRI Pixel Dimensions

row	column	slice
1	1	1

File Header Size, (Bytes): 0

IMPORT QUIT

IMPORT button activated

Import Volumetric MRI Data and Transformation Matrix

MRI Machine Format: IEEE, big-endian

MRI Import Format: integer, 8 bits

MRI Data Type: Slice Sequence

MRI Image dimensions (pixels): Hor. 256 Vert. 256

MRI Pixel Dimensions

row	column	slice
1	1	1

File Header Size, (Bytes): 0

IMPORT QUIT

Choose Slice Sequence option to import MRI slice series files. The additional pixel dimension menu extension appears. Thus both pixel dimension apply buttons must be checked after changing the default values. Then the IMPORT button is activated.

Import Volumetric MRI Data and Transformation Matrix

MRI Machine Format: IEEE, big-endian

MRI Import Format: integer, 8 bits

MRI Data Type: Slice Sequence

MRI Image dimensions (pixels): Hor. 256 Vert. 256

MRI Pixel Dimensions

row	column	slice
1	1	1

File Header Size, (Bytes): 0

IMPORT QUIT

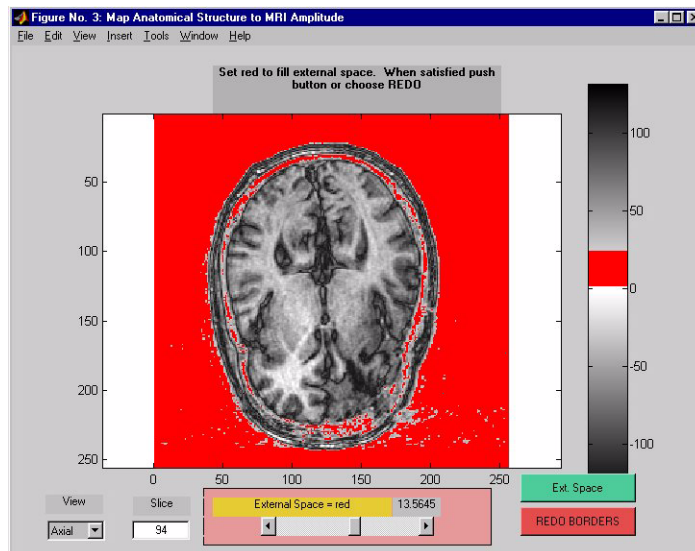
IMPORT button activated

Creating 3D Head from MRI data

This utility is automatically run immediately after completion of the IMPORT MRI application.

The head creation module performs the following tasks:

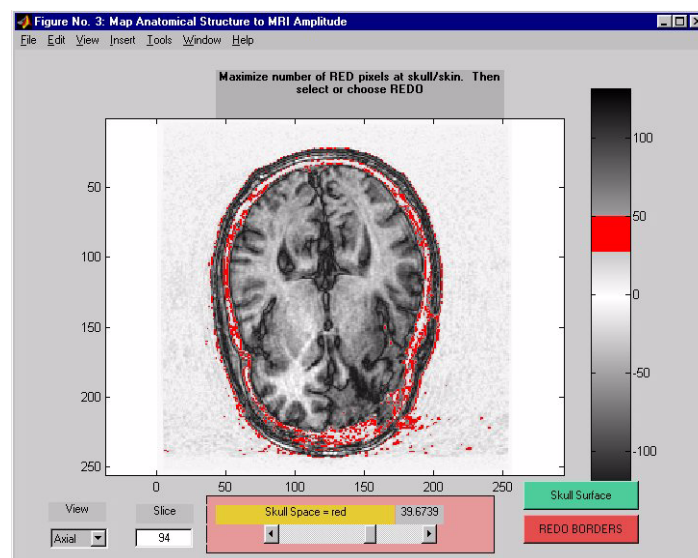
- (1) Associates MRI amplitudes with the following anatomical structures
 - External Space
 - Skull Surface
 - Cortical Gray Matter
 - Cortical White/Gray Matter Boundary
- (2) Sets the threshold amplitude gradients used to recognize the head boundary surface.
- (3) Generates a tentative head model that is examined, using 3D rotation, for quality.
- (4) Allows iterative adjustments to the settings in step (1) and especially step (2).
- (5) Regenerates the MRI based head model until it is acceptable.
- (6) Automatically saves the final head model.
- (7) Invokes the MRI/MEG coregistration utility is by pressing the <LOAD HSFILE> button.

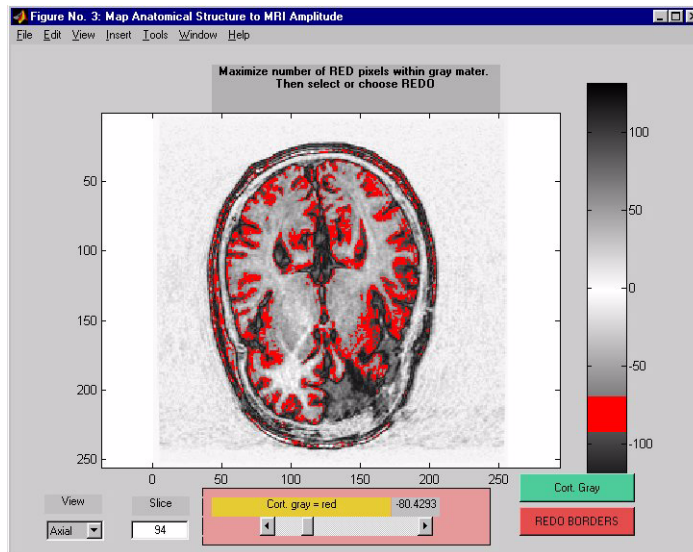


Immediately after the MRI data has been imported the window show above is presented. On this window there are controls to select the MRI image view, (axial, sagittal, or coronal), and the specific MRI slice of the sequence. The slider control is used to select the MRI pixel amplitude. Pixels within a narrow amplitude band centered on this amplitude are displayed in RED. In the above figure, the slider control has been adjusted until the external space is mostly RED. The figure instructions are at the top. Other MRI slices and views can be selected at any time. However, the RED will temporarily disappear until a slight slider change is made.

When the optimal amplitude is determined. The green button is pushed to proceed or the red button to redo the MRI amplitude selections.

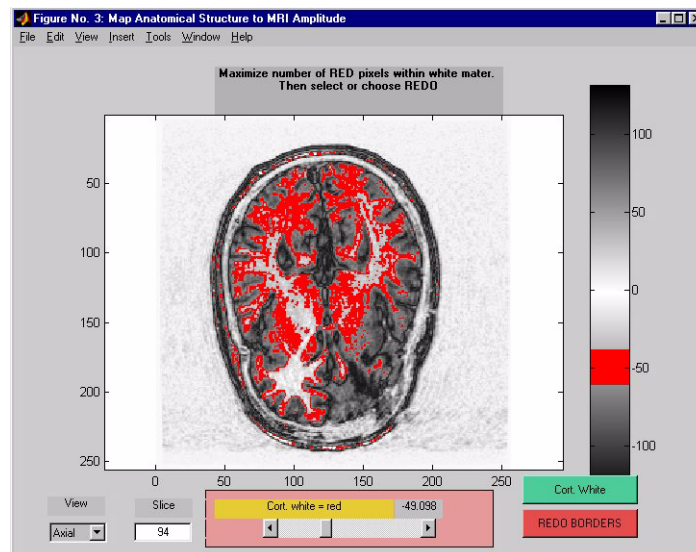
(BELOW), The next figure in the sequence requires the slider amplitude to approximate the skull surface amplitudes. Pressing the green button saves the result and proceed to the next identification.

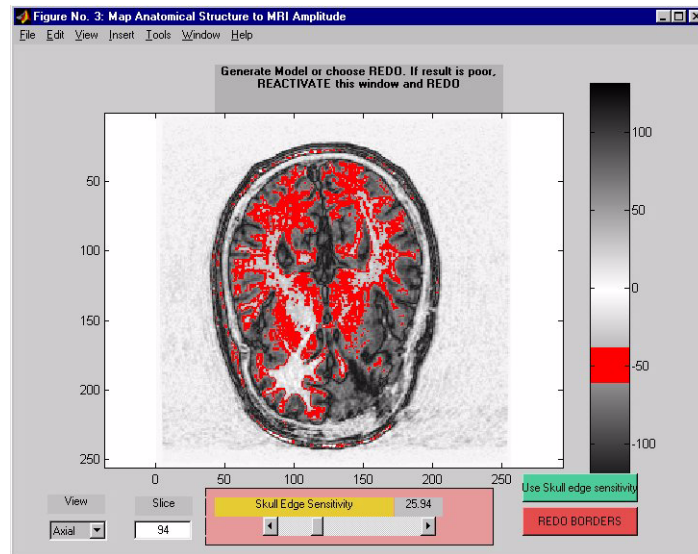




(ABOVE), The slider has been adjusted until the MRI amplitudes centered on the cortical gray matter have been selected. Then the green button is pushed to increment to the next selection task.

(BELOW), The slider has been adjusted to match MRI amplitudes of the white matter at the white/ gray matter boundary. Pushing the green button causes the program to proceed to the next task, (selection of a boundary sensitivity parameter used to identify the skin surface).

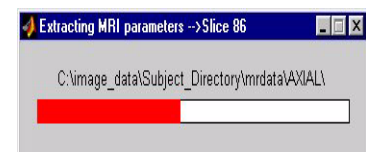
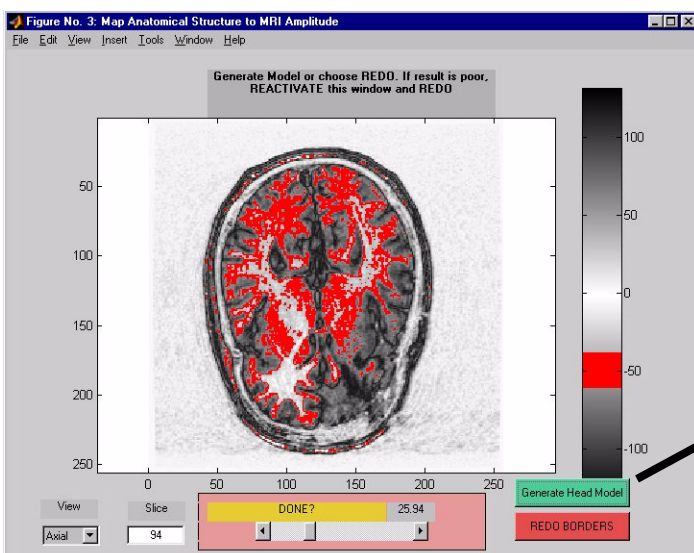




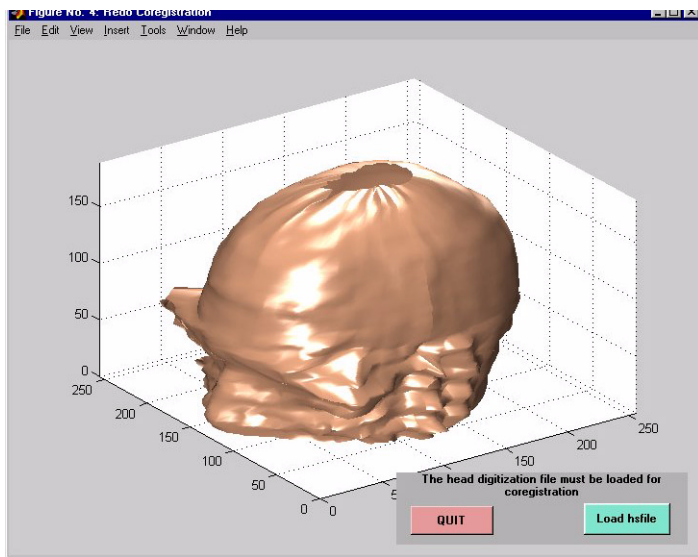
(ABOVE) The slider has been adjusted to a skull edge sensitivity of 25.94 from its default value of 20.0. In general, larger amplitudes make it less sensitive to noise in the external space. However, when too large a value is used the surface is blurred or local break through to internal brain structure is observed in the final MRI head model. If the model is unacceptable it can be deleted and this menu reappears. Thus a different edge sensitivity setting can be selected. This process can be repeated until the head model is acceptable. While setting the edge sensitivity, the RED colors of the MRI slice do not change.

(In general, if the MRI slices are sharp images with little noise, the final MRI head model will be very good. However, if the MRI slices are very poor quality, the MRI head will be very poor quality. Sometimes, it is useful to push the RED button and redo the MRI amplitude identifications to obtain an improved MRI head model. Slight changes in the external space identification and skull surface identification affect the quality of MRI slices displayed in the head model during the viewing of MEG imaging results.)

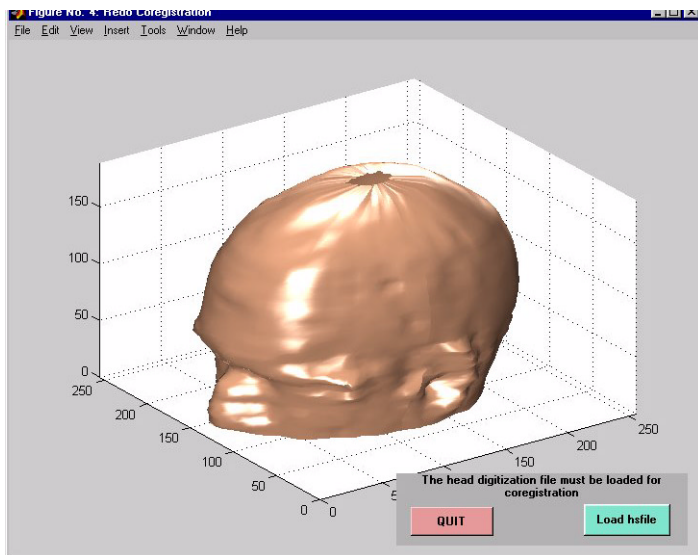
(BELOW) The green button increments to a <Generate Head Model> button. Pushing this button generates the head model which is displayed in a separate window.



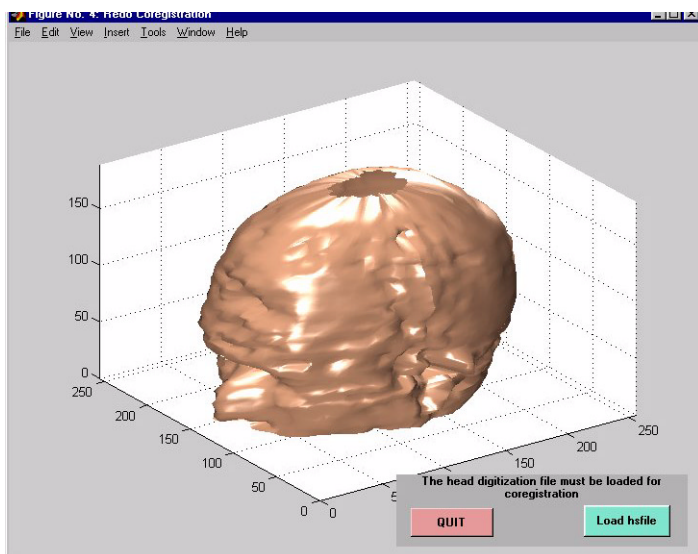
(ABOVE) Progress Meter updates during MRI head model generation.



(TOP FIGURE) The skull edge sensitivity used to generate this MRI head model was set too low. Therefore, the RED <Quit> button should be pushed to delete the head model and return to the previous edge sensitivity selection window. (Note the edge detection is significantly influenced by noise in the external space.)



(MIDDLE FIGURE) The skull edge sensitivity is optimal. Only slight noise distortions are present and ear and facial features are relatively sharp. The GREEN <Load hsf file> button should be pushed to activate the MRI/MEG coregistration module. This is necessary even if the orient_matrix.txt file is an exact MRI/MEG coregistration matrix. The coregistration module allows this matrix to be used without change if desired.



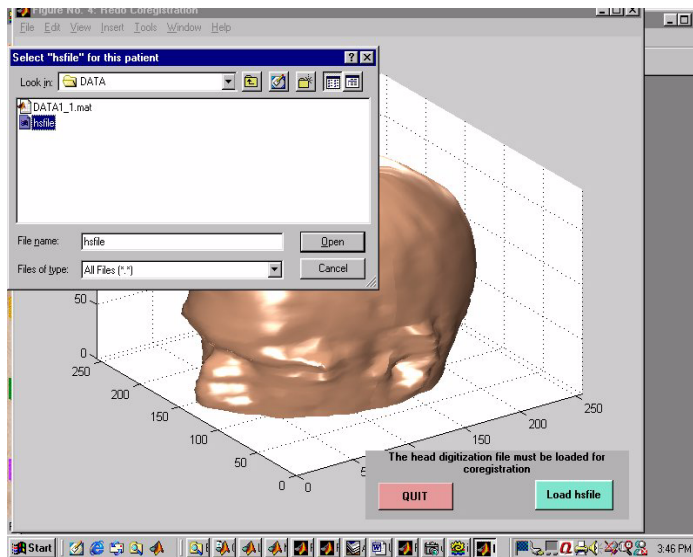
(BOTTOM FIGURE) The skull edge sensitivity is too high. Multiple small areas of break through into internal structure are present in this MRI head model. The RED <Quit> button should be pushed to return to the edge sensitivity selection window.

MEG/MRI COREGISTRATION

This utility is performed as an integrated part of importing MRI data. After the MRI data is imported and the MRI head model has been created, the COREGISTRATION module is started by pressing the green <Load hsfile> button. COREGISTRATION must be performed on all imported MRI data.

This utility does the following:

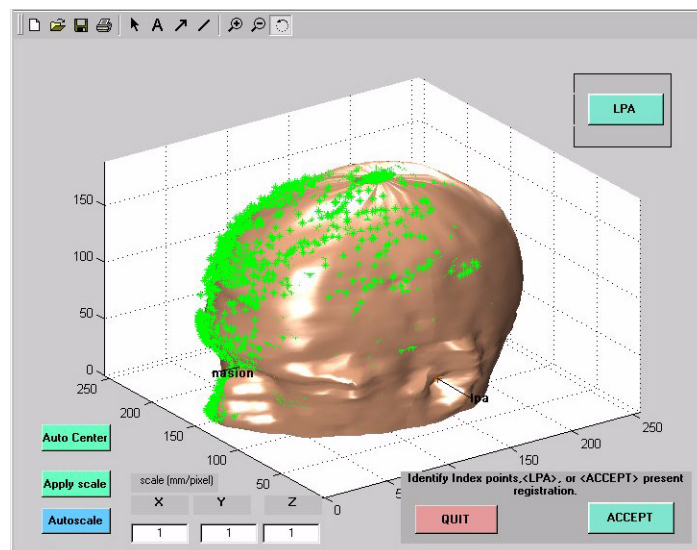
- (1) Uses the orient_matrix.txt as the initial MEG/MRI coregistration matrix.
- (2) Marks the location of the Left, Right and Nasion on the MRI head model using the coregistration matrix throughout the coregistration procedure.
- (3) Provides automatic and manual tools to:
 - (a) shift the center of the MEG coordinates to coincide with the MRI origin.
 - (b) Rotate the MEG coordinate axes to align with the MRI axes.
 - (c) Alter the MRI coordinate scaling such that the MRI head model matches the MEG digitization points over the head surface.
- (4) Saves the final MEG/MRI coregistration matrix with the MRI data and MRI head model.



After creating the MRI head model the program immediately advances to Coregistration: MRI coordinates with the MEG coordinates by pressing the <LOAD HSFILE> button.

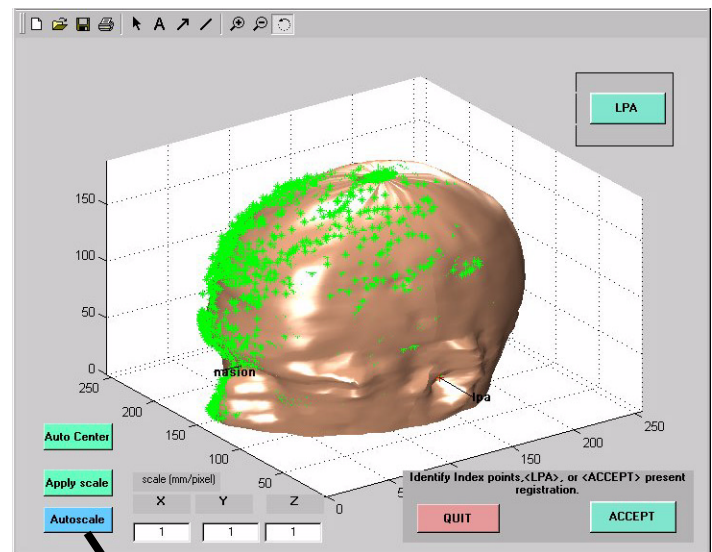
Coregistration steps

1. Press the <Load hsfile> button. The program will look for the hsfile in the <drive>\image_data\SD\ directory. If it is not found in this location a file open window is created. Using this utility you must locate and select the hsfile for the subject, (SEE FIGURE ABOVE). After, loading the hsfile the figure changes, as shown below. The orient_matrix that was imported with the MRI volumetric data is used as an initial coregistration matrix for the Left PA, Right PA, Nasion and digitization points.
2. Add digitization points by pressing the <Add digit pts> button. (SEE FIGURE BELOW)



- 3 If necessary, change the MRI pixel scaling (mm per pixel) for the X (back to front), Y (right to left), and Z (bottom to top) axes by typing in the appropriate edit box followed by pushing the <Apply scale> button. Do not use the <AUTOSCALE> button until the MEG coordinate system is properly centered and the X,Y, and Z scaling is approximately correct. Add the digitization points with the new scaling factors applied by pushing the <add digit pts> button.
- 4 Rotate the figure to examine the coregistration fit of these points as follows:
 (Matlab 5) Click on window Tool menu > Show Toolbar > toggle the Rotate 3D button to ON > drag figure with mouse to different orientations.
 (Matlab 6) Click on window toolbar > View > Figure Toolbar then toggle the Rotate 3D icon button ON > Drag the figure to different orientations using the mouse.
- 5 If MEG index and digit points are located properly on the MRI head model, press <ACCEPT> button to save the final coregistration.
6. However, if the orientation of the MEG coordinate system is correct but the origin is not centered properly, the <AUTOCENTER> button should be used. In the figure below the digitization points are shifted a small distance in the positive X direction. Therefore, the <Auto Center> button is used to shift the origin of the MEG coordinate system. (SEE FIGURES BELOW)

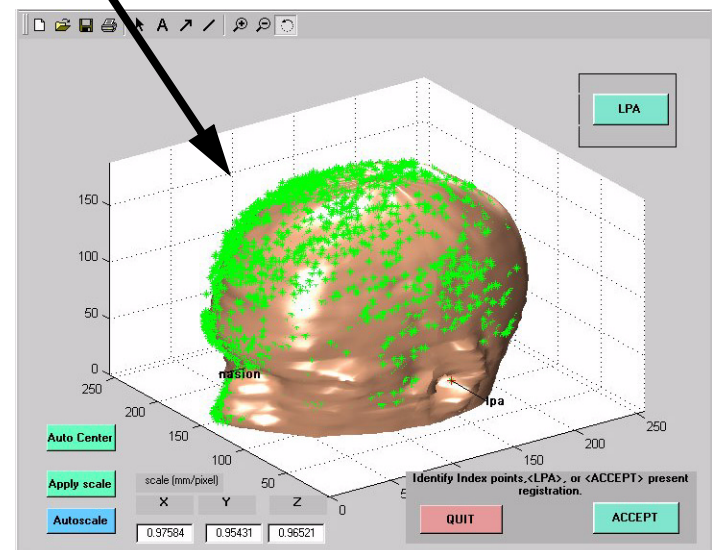
Beginning Coregistration using the orient_matrix.txt as the MRI/MEG coregistration matrix. The pixel scaling used in this figure is 1.0 mm/pixel which is an approximation of the correct pixel scaling. The orient_matrix.txt was created from STA_R coregistration matrix of the 4D imaging program.

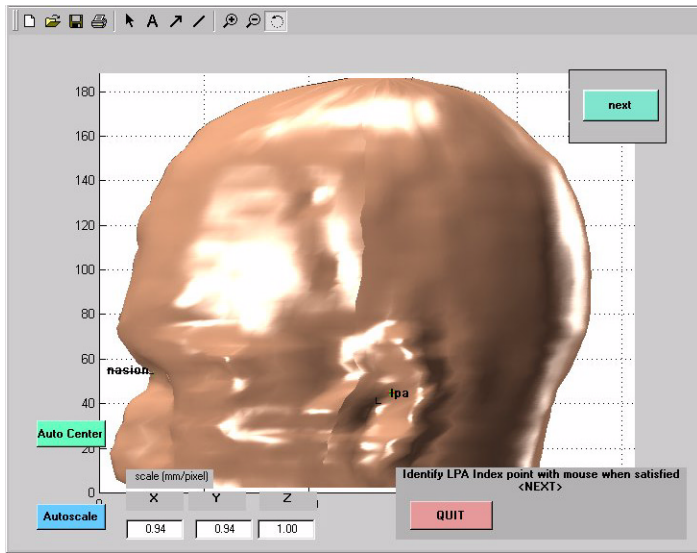


MRI/ MEG coregistration after applying:

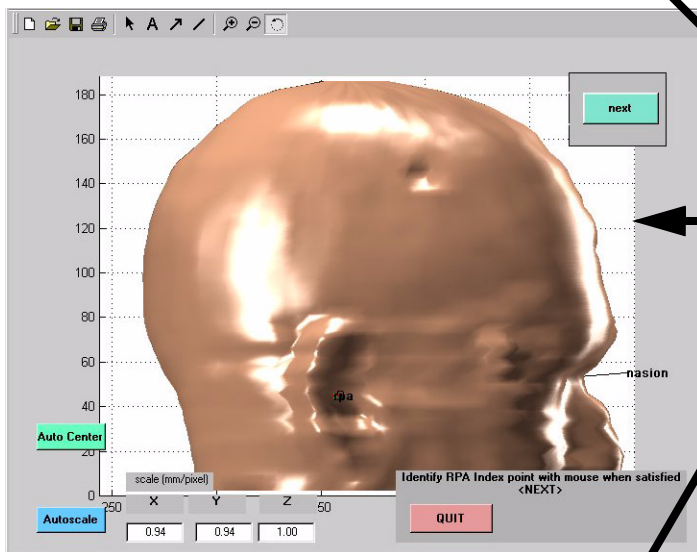
- (1) <Auto Center>
 - (2) <Auto Scale>
 - (3) <Auto Center>
 - (4) <Auto Scale>
- then
- (5) <Add digit pts>

The final X, Y, Z pixel scaling is:
 .97, .95, .96

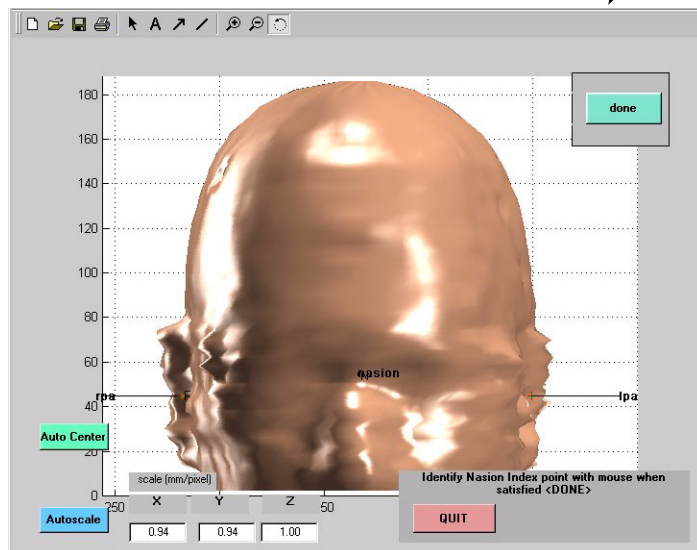




7. If the coordinate system needs to be rotated to a new orientation a complete MRI/MEG coregistration must be performed. Press the <LPA> button to perform a complete MRI/MEG coregistration. A sequence of 3 views of the MRI generated head will be presented to reselect the Left PA, Right PA, and Nasion. **DO NOT ROTATE THESE COREGISTRATION FIGURES.**



- Identify the Left PA by clicking on the head with the mouse. Then press the <NEXT> button.
- Identify the Right PA by clicking on the head with the mouse. Then press the <NEXT> button.
- Identify the Nasion by clicking on the head with the mouse. Then press the <DONE> button.
- Add the digit points. Toggle the rotate 3D button OFF the ON again to rotate the figure and view the results of the coregistration.
- Repeat steps 6a through 6d until the coregistration fit is acceptable. Then press the <ACCEPT> button.



8. <QUIT> terminates this SOURCE MODEL generation module without saving results.
9. <AUTOSCALE> can be used to enhance to MEG/MRI coregistration fit by separately adjusting X,Y,Z mm/pixel scaling to achieve the best fit of the digit points to the MRI head surface. It is not necessary to press <Apply Scale> button after <AUTOSCALE>.
 - a. This would increase coregistration accuracy by compensating for minor MRI distortion and/or minor coregistration errors, (More accurate external alignment of model \rightarrow more accurate internal structure alignment with sources).
 - b. In addition, for studies of multiple individuals with no MRI data. A cortical model for each individual could be generated by scaling an appropriate normal brain MRI to fit each individual separately. This single re-scaled cortical model is a better model of the brain than a uniform sphere. In addition, it would facilitate the comparisons of relative source locations between individuals because the results would be displayed in the common MRI pixel units. However, the relationship between cortical structure and source activity would be only approximate for this type of study result.

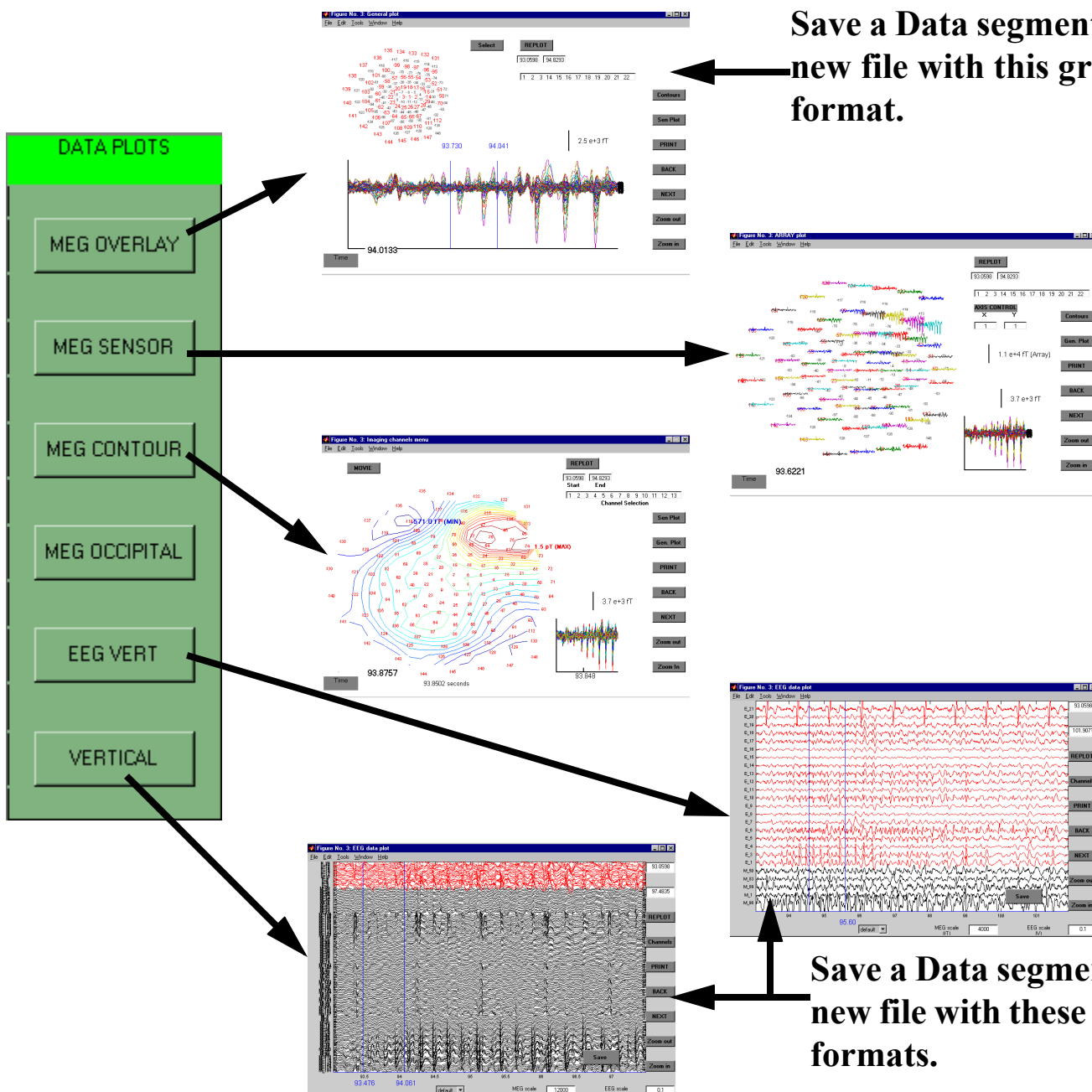
Above, after successfully MEG/MRI coregistration, the MEG, MRI common index points and the final coregistration matrix for the input MRI volumetric data is displayed.

WHAT CAN GO WRONG during COREGISTRATION?

1. Occasionally, after rotating the head figure rapidly then performing the coregistration, the final results have the head upside down or backwards, etc.
 - a. TO FIX THIS PROBLEM redo the coregistration procedure. Use a best guess at where the true LPA, RPA and Nasion should be located. This should cause the head to reorient to approximately the right orientation.
 - b. Next, this result can be refined to the final result.
2. Each display operation deactivates the Rotate 3D selection even though it appears to be selected. The solution is to toggle the Rotate 3D to OFF then toggle it ON.
3. This Coregistration utility was developed to utilize 4Dneuroimaging index points and head digitization point as well as 8 bit scaled MRI and coregistration matrix from the STA_R imaging utility incorporated into their software. The ability to import other types of MRI data is a recent addition to the program. Therefore, if you have difficulty please contact the MEG_tools author. The EMAIL address is moran@neurnis.neuro.hfh.edu
4. FUTURE PLANS: In addition to T1 weighted MRI volumetric data, each MRI pixel location can be associated with T2 weighted image data, fMRI amplitudes, PET amplitudes. Therefore, including this data in the design of a CORTICAL SOURCE MODEL requires each MRI pixel to be described as a vector rather than an amplitude. The subsequent construction of the source model will require the use of vector metric's to specify the model locations and initialization amplitudes for subsequent imaging. We will be developing a menu/mouse GUI application for this task as part of our continuing research.

DATA PLOTS

- DATA PLOT buttons are used to select the type of plot used to display MEG and EEG data.
- Each of the data plots has control buttons and edit boxes for changing the plotting parameters and to create the other types of data plots.
- A segment of the data can be selected then saved in a new data file. These small segments of data can be loaded for further analysis, such as, MEG source imaging. Segments of data can be selected when using the OVERLAY, EEG VERTICAL, and VERTICAL plot types.



Save a Data segment to a new file with this graph format.

Save a Data segment to a new file with these graph formats.

MEG OVERLAY: Can select data segments that are automatically stored as an EPOCH structures. These data segments can be loaded, (Use LOAD DATA button), for plotting and MEG cortical source imaging.

MEG SENSOR: Time series data is plotted on top view of sensor array with front of array toward the right.

MEG CONTOUR: Contour plot of MEG data viewed from above the sensor. Can make movie of the data.

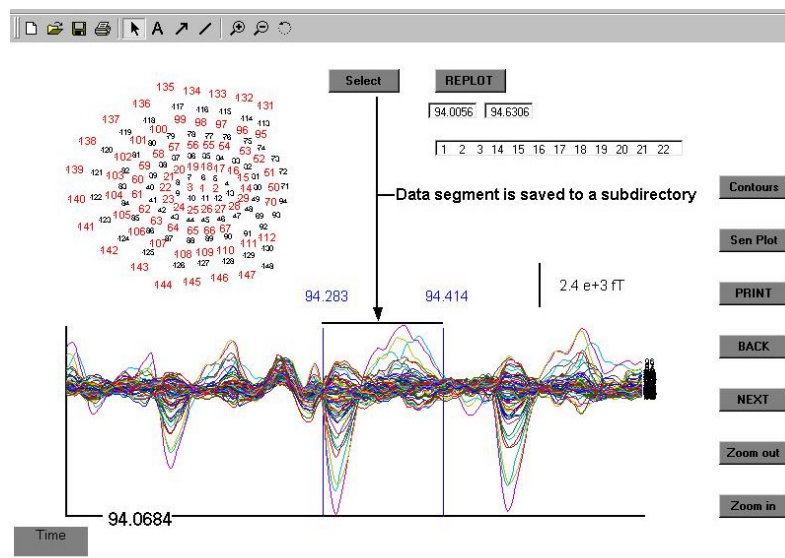
EEG VERT: Time series plot of EEG channel data and a few selected MEG channels. Can select data segment.

VERTICAL: Time series plot of MEG and EEG data. Can select data segment.

Vertical and horizontal axis scaling and channel selection controls are on each graph window.

OVERLAY PLOT

1. The channels plotted are the highlighted large red numbers in the top view of the sensor layout in the upper left corner of the plot figure. In this layout the nose is to the right and the left ear is at the top.
2. Channels can be included or excluded by using the mouse to toggle the channels on/off in the sensor layout or by inserting a channel number, (any place), in the channel edit box. Or delete channels by highlight and backspace to remove from the edit box. If the edit box is used the replot button must also be selected.
3. Changing the time segment of data plotted:
 - a **Edit start and end** of data segment boxes and push **replot** button.
 - b Press **next** button to plot the next segment of data of the same segment length.
 - c Press the **back** to plot the previous data segment.
 - c Click mouse at center of desired data segment then **zoom in** or **zoom out** button.
4. Invoking a new graph of a different type
 - a Push **Contours** button to get a contour plot of the start of the present data segment.
 - b Push the **Sen Plot** button to get the sensor layout plot of the present data segment.
5. **Saving a segment of data as new data file**
 - a Click mouse at the beginning and end of the data segment, (the order is not important).
 - b Push the **select** button that appears to save the chosen data. A subdirectory of the loaded data directory will be created, (named by the date), and the data saved as an EPOCH structure in a file named dataseg1.mat or dataseg2.mat, etc. depending on the number of data segments selected until the date changes. For example, d:\image_data\subject_directory\run4\data\p04-aug-2000\dataseg1.mat. After being created each dataseg# file can be independently loaded, plotted and used for MEG cortical imaging. It is also possible to select a segment of data from a dataseg file. A new subdirectory of this subdirectory would be automatically created to contain this new selection. For example, d:\image_data\subject_directory\run4\data\p04-aug-2000\p04-aug-2000\dataseg1.mat.
 - c All the MEG and EEG data channels in the original data are saved for the data segment rather than just those that are presently plotted.
 - d Data selection is terminated by clicking the mouse in the upper right corner of the plot window. This causes both the selection limit lines and selection button to disappear.



CONTOUR PLOTS

1. Control buttons are the same as OVERLAY PLOTS.
2. To change the time point of the contour plot, click the mouse on the small time series graph of the data segment to change the cursor to a full cross-hair type then click at the time point desired.
3. To use **zoom in** and **zoom out**, click the zoom in/out button first, then click the center of the desired data segment in the small graph. Finally, chose the time point for the contour plot as described in (2).
4. Create a contour data **movie** corresponding to 20 equally spaced time slices of data throughout the plotted time series data by pressing the movie button. After the movie is created, **play movie**, **fast**, and **slow** buttons appear.
5. Data segments cannot be selected from contour plots.
6. By default, all MEG channels are included in contour plots. To deselect/reselect channels use the mouse to toggle channels on/off or edit the channel selection box, (left/right arrow keys, <enter> will index the edit box to channels outside the box limits).
7. In these contour plots the nose is on the right. The back of the head is to the left. The left ear is at the top and the right ear at the bottom.

SENSOR PLOTS

1. Control buttons work like the corresponding CONTOUR PLOTS.
2. Zoom in/out by pressing the button first and clicking on the center of the desired interval second.
3. Axis Control Edit boxes are used to change the X and Y scaling of the sensor layout waveforms. If 0.5 is put in the Y axis control box followed by <REPLOT> then the vertical extent of the waveforms is reduced by 50 percent. If the fraction 2.0 is subsequently entered in this box, the original scaling is restored because $0.5 \times 2 = 1$. Similarly, the X axis control edit box changes the time axis scaling.
4. Channels can be toggled on/off with the mouse or the channel edit box can be used.
5. Data segments cannot be selected and saved from these plots.
6. In these sensor layout plots the nose is on the right. The back of the head is to the left. The left ear is at the top and the right ear at the bottom.

VERTICAL LAYOUT PLOTS

1. Zoom in/out controls are different than contour and sensor plots. First, the mouse is used to select the center of the desired plot interval by clicking on the graph at the appropriate latency. Next, the zoom in or the zoom out button is pushed.
2. The channels button opens a window containing the sensor layout and EEG channels of the data. Channels are added or removed by using the mouse to toggle the appropriate channels on/off. Then the <USE> button to apply the channel selection or <QUIT> to discard channel changes and return to the plot.
3. The vertical scaling of the MEG and EEG channels can be separately changed by using the edit boxes at the bottom with the <REPLOT> button pressed to apply the scale changes.
4. Data segments can be selected and are saved to a data subdirectory that is automatically created. The name of this subdirectory is determined by the date, as previously described in OVERLAY PLOTS. The data segments are saved as EPOCH structures with file names dataseg1, dataseg2, etc., if multiple segments are sequentially selected on the same day. Any of these data files can be loaded using <LOAD DATA> of the main imaging menu.
5. The VERTICAL plots are identical to the EEG VERTICAL plots except all MEG and EEG channels of data are initially plotted. The CHANNELS button can be used to edit the channels plotted.

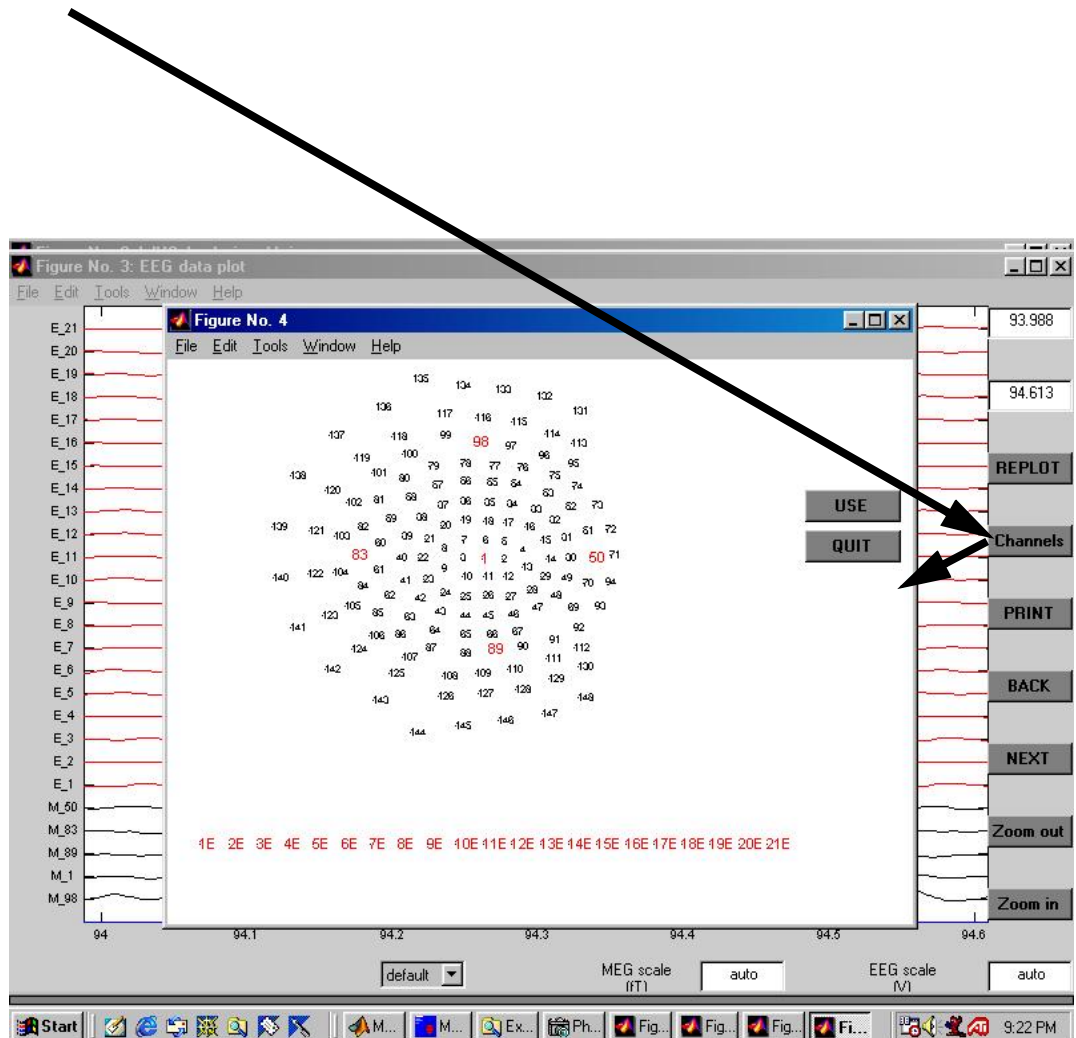
ALL GRAPH NOTES and COMMENTS:

All graphs and their controls were designed to plot and align correctly at a screen resolution of 800 x 600 pixels. At other resolutions, control buttons and boxes may be misplaced. These resolution dependent characteristics will be addressed in the future.

In addition, some graph types utilize common variables to control plotting. This can create unexpected behavior if more than one graph type has been plotted and the window buttons are used to switch between graph types. In the future, this interdependence will be eliminated.

Therefore, if a graph is not responding properly, close the graph window, return to the main menu and regenerate the desired plot.

In the figure below, the channel selection window is used to change the channels used in the **EEG vert and the VERTICAL layout plots**. If the screen resolution is not correct the <USE>, <QUIT> button may not be visible without maximizing the screen size of this menu.



Adaptive Noise Filter

As described in CORTICAL SOURCE IMAGING of this user manual, an optimal inverse filter is a combination of an inverse matrix operator and a data filter operator which is optimized to eliminate noise and apply other constraints on the source solution. However, the solution techniques incorporated in this software do not utilize an inverse filter, thus adaptive filtering of noise is performed as a separate operation. The adaptive noise filter utilized in the CORTICAL SOURCE IMAGING section is designed to attenuate noise when the signal amplitude is greater than the noise amplitude. However, a significant number of patients requiring MEG study and analysis have dental appliances that generate very large magnetic artifact despite efforts to demagnetize them. These MEG data require an adaptive filter designed to remove the large included noise and leave the relatively small amplitude signal. This is accomplished by utilizing a variation of the noise filter described in CORTICAL SOURCE IMAGING.

The optimal noise filter is:

$$\mathbf{F}_{\text{noise}} = (\mathbf{B} - \mathbf{N}) \mathbf{B}^{-1} \mathbf{b}$$

The data covariance matrix, \mathbf{B} , is decomposed into the eigenvector/eigenvalue format: $\mathbf{B} = \mathbf{U}\mathbf{\lambda}\mathbf{U}^T$. Similarly, the noise covariance matrix is $\mathbf{N} = \mathbf{U}\mathbf{\lambda}_N\mathbf{U}^T$. With these substitutions, the optimal noise filter becomes:

$$\mathbf{F}_{\text{noise}} = \mathbf{U}(\mathbf{\lambda} - \mathbf{\lambda}_N)\mathbf{\lambda}^{-1} \mathbf{U}^T = \mathbf{U}\mathbf{\lambda}_{\text{sig}}\mathbf{U}^T$$

The exact spectrum of the noise is unknown but concentrated in the largest eigen value/vector components of $\mathbf{\lambda}$. Therefore, the diagonal elements of are chosen to be:

$$\text{diag}\mathbf{\lambda}_{\text{sig}} = 1 - \beta(\lambda_i / (0.02\lambda_i + 0.98\lambda_{\text{largest}}))$$

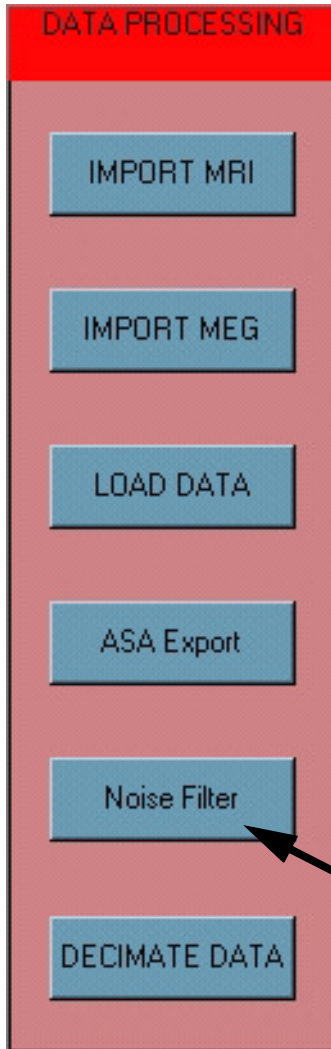
$$\beta = 1 - \text{mean}(\log(\lambda_{\text{largest}} / \lambda_{(1 \text{ through } n)}))$$

When this filter is applied to MEG data to remove large amplitude noise the following operations are performed.

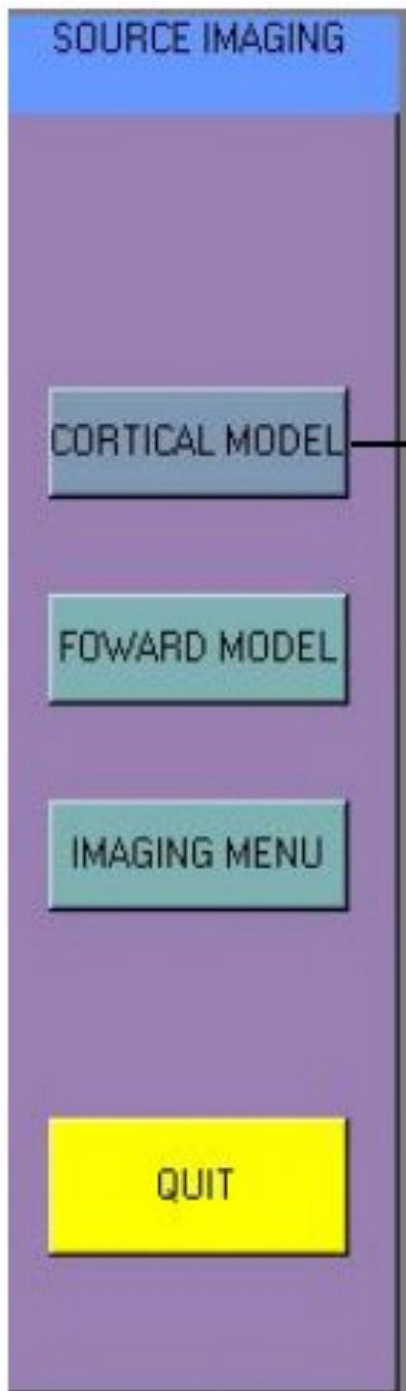
(A) If the data consisted of a sequence of individual evoked responses: Each epoch is adaptive filtered and the filtered average is saved in a separate subdirectory.

(B) If the data is a long time series divided by MEG Tools into a sequence of epochs: Each epoch is adaptive filtered and overwritten by the filtered data.

(C) If the MEG data is contained in a single EPOCH structure file: The MEG data is adaptive filtered and available for plotting and imaging. However, it is not saved to disk,(use the save data segment procedure if desired).

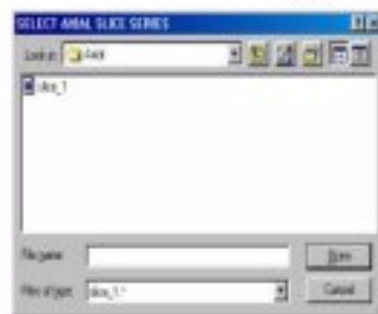
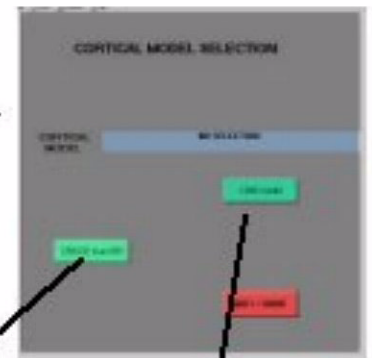


CREATING A CORTICAL MODEL



CORTICAL MODEL This button is used to invoke the cortical model menu with options for:

- Creating a cortical model from volumetric MRI data previously imported.
- Loading a previously constructed model



The **CREATE from MRI** button invokes a file open menu with the appropriate axial series slice_1.mat file in the window. Highlight this file and press OPEN button or cancel if creation process is to be terminated.

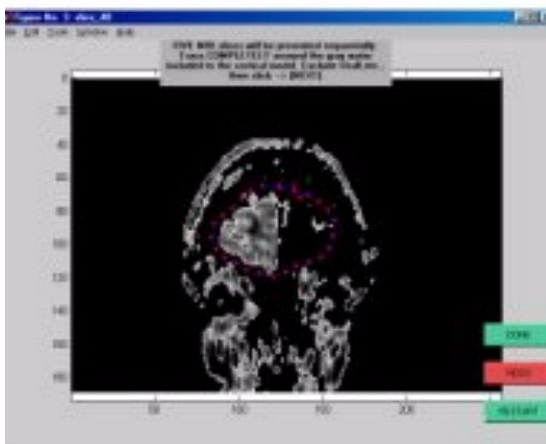
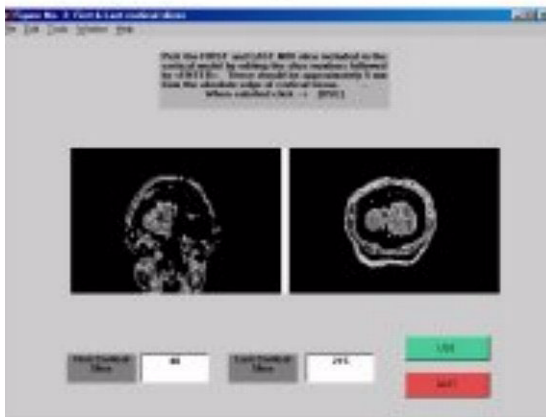
The **Load model** button opens a menu with the list of cortical models that have been created. Highlight the desired model then the <Ok> button. The model that is loaded will be listed in the status bar on this menu and the main menu.

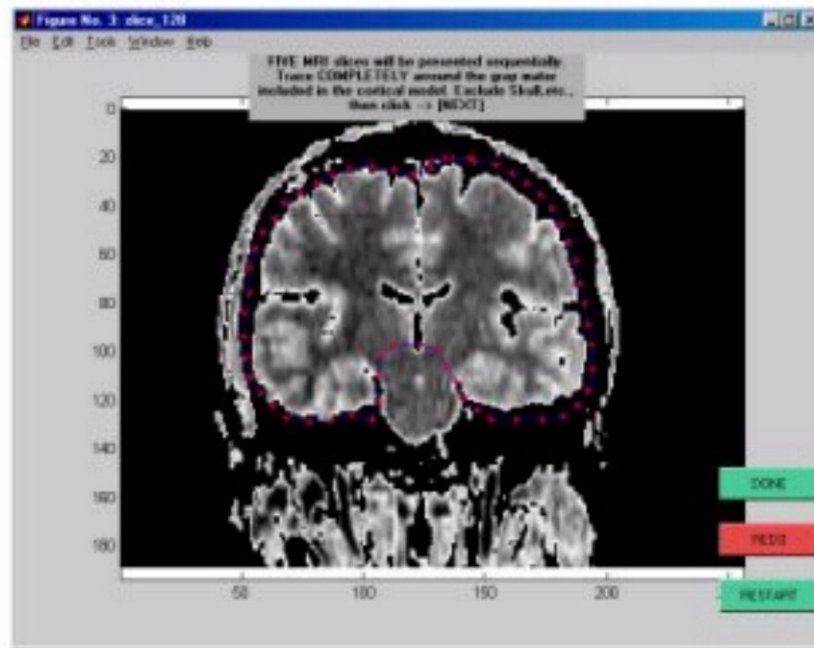
Steps required to create cortical model:

1. Defining an outer limit surface boundary within which the desired cortical structures are located and specifying a range of MRI amplitudes that uniquely determine the cortically active structures.
2. Searching the entire MRI coronal slice sequence for cortical pixel locations consistent with these parameter limits.
3. Distributing 3000 source locations throughout this volume on a set of 15 equally spaced coronal slices such that the distribution of the source locations matches the cortical structure.

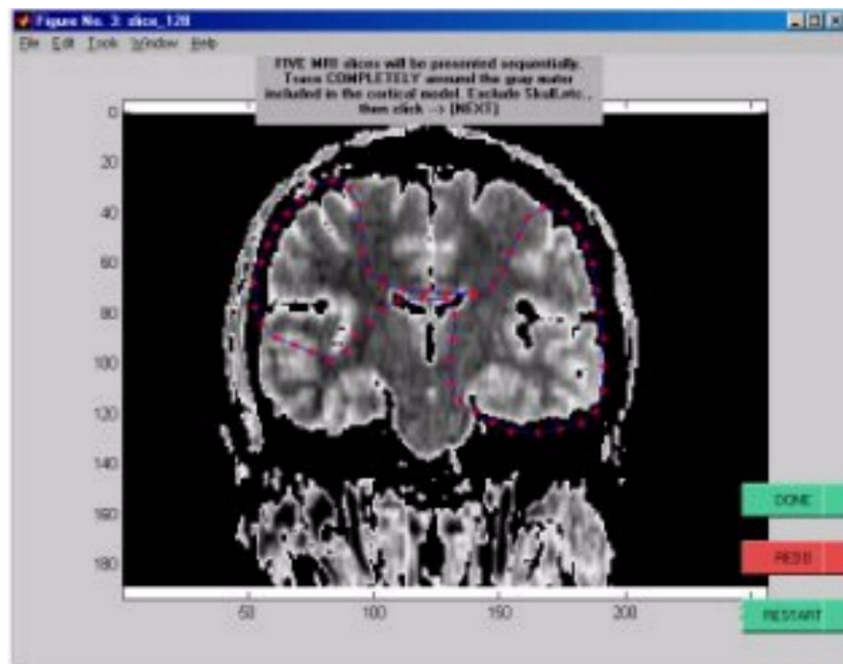
Defining the outer limit surface

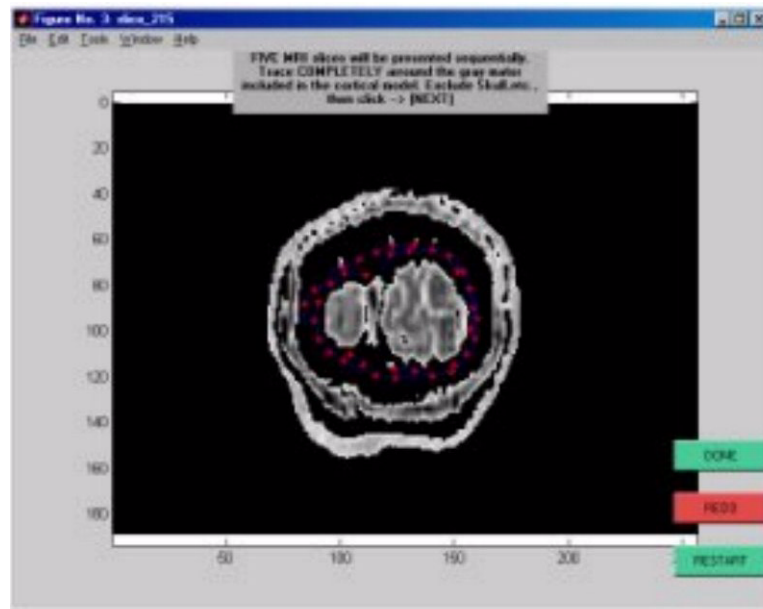
1. After file selection and processing the axial slice series, (see previous page), the menu at the left is presented.
2. Enter the first and last coronal slices within which the desired cortical model is contained. It is not necessary to include the entire cortex if the sources corresponding to the magnetic field data are known to lie within a specific region. (see illustration , left side).
3. Draw the lateral boundaries of the cortical model on the set of 5 coronal slices that are sequentially presented.
 - a. Hold the mouse button down while tracing the outer border.
 - b. Line tracing does not have to be continuous. A continuous line boundary or a sequence of overlapping line segments can be used. All border discontinuities must overlap.
 - c. <DONE> next slice of 5 slice sequence
 - d. <REDO> redo this slice only
 - e. <RESTART> redo all 5 slices





Above, the outer boundary was defined by holding the mouse button depressed while tracing around the region of the cortex that is to be included in the final cortical source model. Multiple regions cannot be selected. However, within limits, the boundary can be complex as shown below.

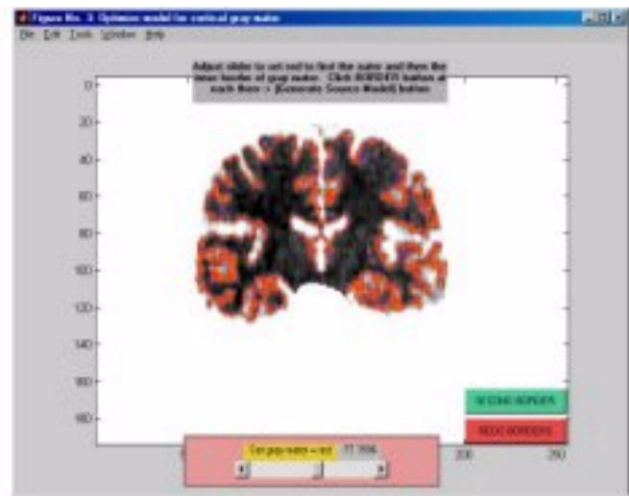
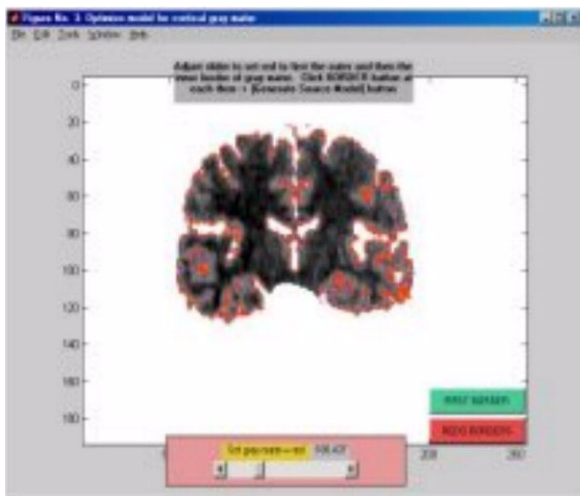




The boundary in the above figure is defined by multiple overlapping line segments. Overlapping each line segment is important to prevent gaps in the boundary. Likewise, even if a continuous line is used to define the boundary, the beginning and end of this single boundary line must be overlapped.

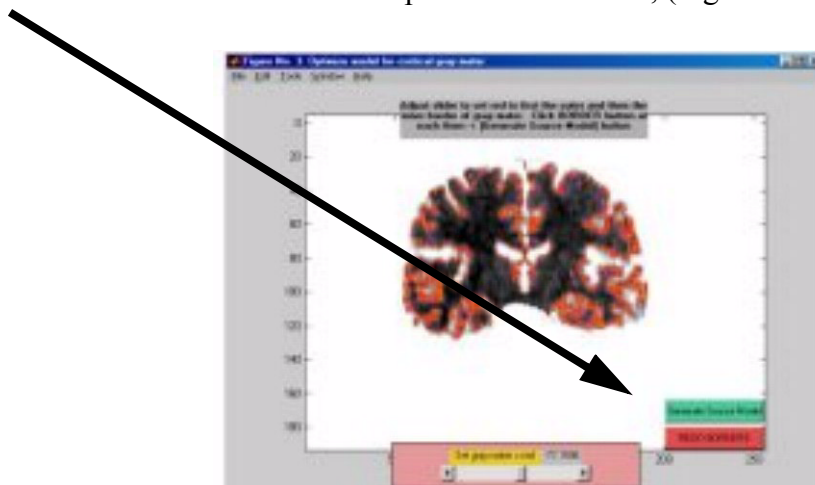
Specifying the range of MRI amplitudes unique to cortical sources

Gray matter and white matter have different ranges of MRI amplitudes. Further, maximum and minimum MRI amplitudes associated with cortical gray matter correspond to the outer (border with csf) and inner (border with the white matter) physical borders of the gray matter. Therefore, the location of all cortical gray matter within the selected boundary envelope is found using the MRI amplitude range. The MRI amplitude range is established by defining it's physical borders with the procedure below.



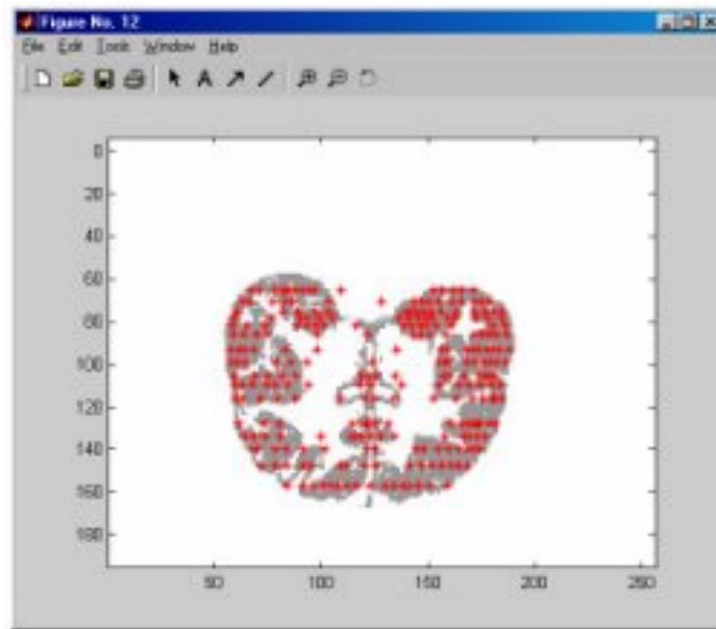
In the figure, (above, left), the slider has been moved until the red pixels lie on the outer border of the cortical gray matter, (border with the csf surrounding the brain). When satisfied with the border selection, the <FIRST BORDER> button is pushed.

In the figure, (above, right), the slider has been adjusted until the red pixels coincide with gray matter adjacent to the white matter. When satisfied with the selection, the <SECOND BORDER> button is pushed. Alternatively, the MRI border selection can be redone by pushing the <REDO BORDER> button. Finally, push the <Generate Source Model> button to create the final 3000 point source model, (Figure below).



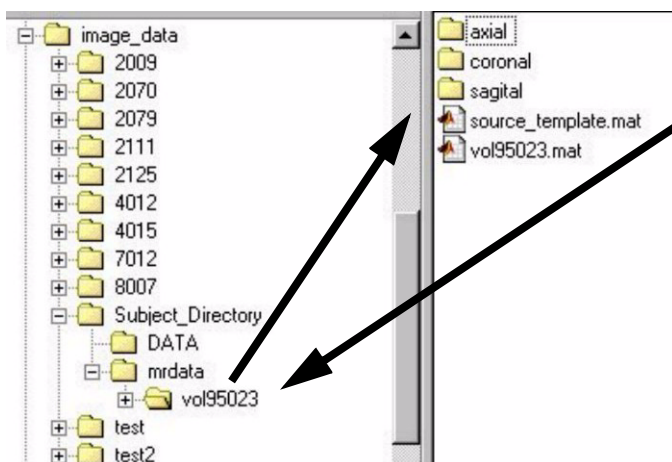
Distributing source points in gray matter

The 4 of the 15 slices containing the source locations are displayed in separate windows. These coronal slices are not necessarily oriented in the "standard view" used in the previous figures. The location of gray matter represented by each slice is solid gray and the distribution of source locations are red stars (*). If the cortical model depicted in these figures is acceptable these windows can be deleted and/or printed for documentation or publication, etc.



The figure above is one of 15 slices of the final cortical model, (top of the head is at the bottom, left side is to the right of the figure). The gray matter in this figure is a composite of the gray matter in the MRI coronal slices 106 through 117. The source points have been distributed such that each represents approximately the same volume of gray matter.

The cortical model is stored as a data structure => SOURCES with a file name such as vol95023. If a source structure is loaded it can be examined by typing SOURCES <ENTER> in the MATLAB command window. In addition, a select set of axial, coronal, and sagittal slices are created and saved with the source structure in a subdirectory, <drive>\image_data\Subject_directory\mrdata\ vol95023, (See Figure below).



FORWARD MODEL (GAIN MATRIX CALCULATIONS)

This module uses the loaded data structure, (EPOCH), and loaded source model structure, (SOURCES), to create the forward solution model for x, y, and z oriented current dipoles of unit amplitude, (1 nanoamp-meter). The calculations are based on a spherical head model with options available to fit the model to match the skull geometry utilizing either the MRI head model or the head digit points acquired during the MEG study. The forward model matrix is stored in a data structure, (FORWARD_MODEL), that is saved in a subdirectory, (forward_model), of the loaded data directory.

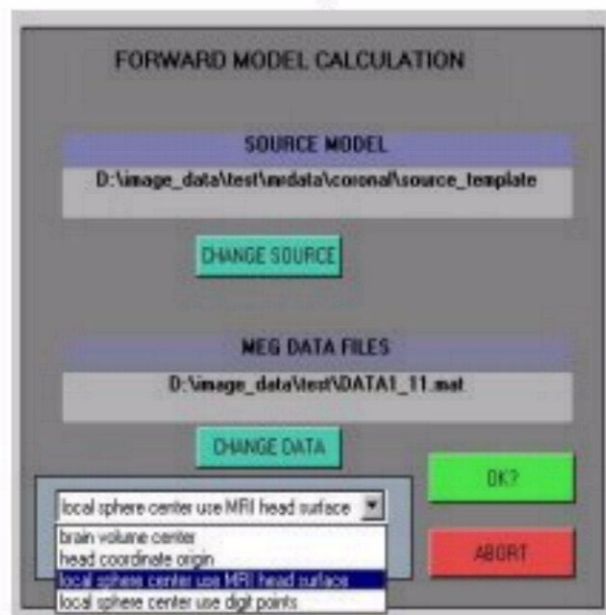
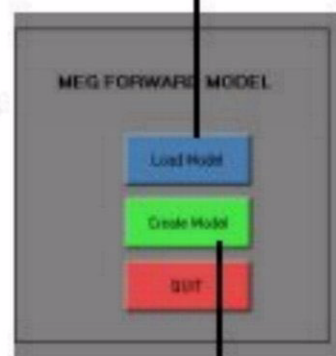
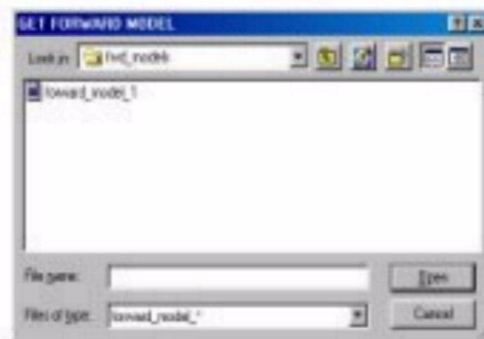
SOURCE IMAGING

CORTICAL MODEL

FORWARD MODEL

IMAGING MENU

QUIT

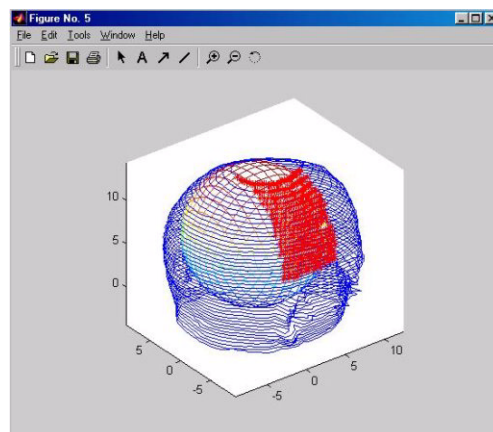


FORWARD MODEL Utility

- Used to create the Gain Matrix corresponding to the loaded data and loaded cortical model. Utilizes a spherical head approximation with the sphere center chosen to fit the head as defined by the head shape digitization points or by the head shape derived from the MRI data.
- Alternatively, a forward model Gain matrix previous calculated and corresponding to the data

CREATION STEPS: FORWARD MODEL GAIN MATRIX

- 1 Load MEG data file for which the Gain matrix will be utilized. (The EPOCH data structure contains the location of the MEG sensors).
- 2 Load the Source Model for which the Gain matrix will be calculated. (The SOURCES structure contains the source locations and coregistration matrix.)
- 3 Press
 - a the <FORWARD MODEL> button on the main menu
 - b press the <Create Model> button that appears
 - c choose the method of determining the center of the spherical model in the FORWARD MODEL CALCULATION menu
 - brain volume center = center of the loaded Source Model cortical locations.
 - head coordinate origin = (0,0,0) of the MEG coordinate system.
 - local sphere center use MRI head surface = 6 regional spheres fit to MRI head shape data
 - local sphere center use digit points = 6 regional spheres fit to the MEG hsfile digit points.
 - For the local sphere center forward calculations, the sources within a each specific region are assigned the appropriate sphere center.
 - d Verify the MEG data selection is correct. Note: Immediately after creating a SOURCE model, the MEG path is incorrect and the <change data> button should be used to reload the desired data file. This does not occur if a source model that has been previously created is reloaded.
 - e Verify that the desired cortical model is selected.
 - f Press the <OK> button to perform Gain Matrix calculations for the x,y and z oriented dipole components at each of the SOURCE MODEL locations. The results are stored as a FORWARD_MODEL structure in a subdirectory, (forward_model), of the MEG data directory that is loaded.
 - g The program displays in separate windows the local sphere fit to the surface data used for each region. Next, the X,Y, and Z forward model calculations are performed and the data points in each sphere fit region are shown in a wire schematic of the sensor array. These figures should be reviewed to avoid errors. Use the print, save, export utilities from the window menu as desired, then delete these windows.



Cortical Source Imaging

Module Topics:

1. Create a new **SOLUTION** using:

- a. the loaded EPOCH DATA STRUCTURE.
- b. the loaded CORTICAL MODEL STRUCTURE.
- c. the loaded FORWARD MODEL STRUCTURE.

Available Solution Types and MEG data transformation options:

A Two Dimensional Inverse Imaging with independent time slices

Data types: Conventional time series or Time Derivative data

Source structure initialization: Single Current Dipole metric

Random amplitudes.

User supplied source structure

Time series continuity constraint

B Multi-resolution FOCUS

Data types: Conventional time series or Time Derivative data

Source structure initialization: Single Current Dipole metric

Random amplitudes.

User supplied source structure

C Single Current Dipole metric

D Data types available for source imaging:

Conventional time series data

Time Derivative data

Noise filtered conventional time series

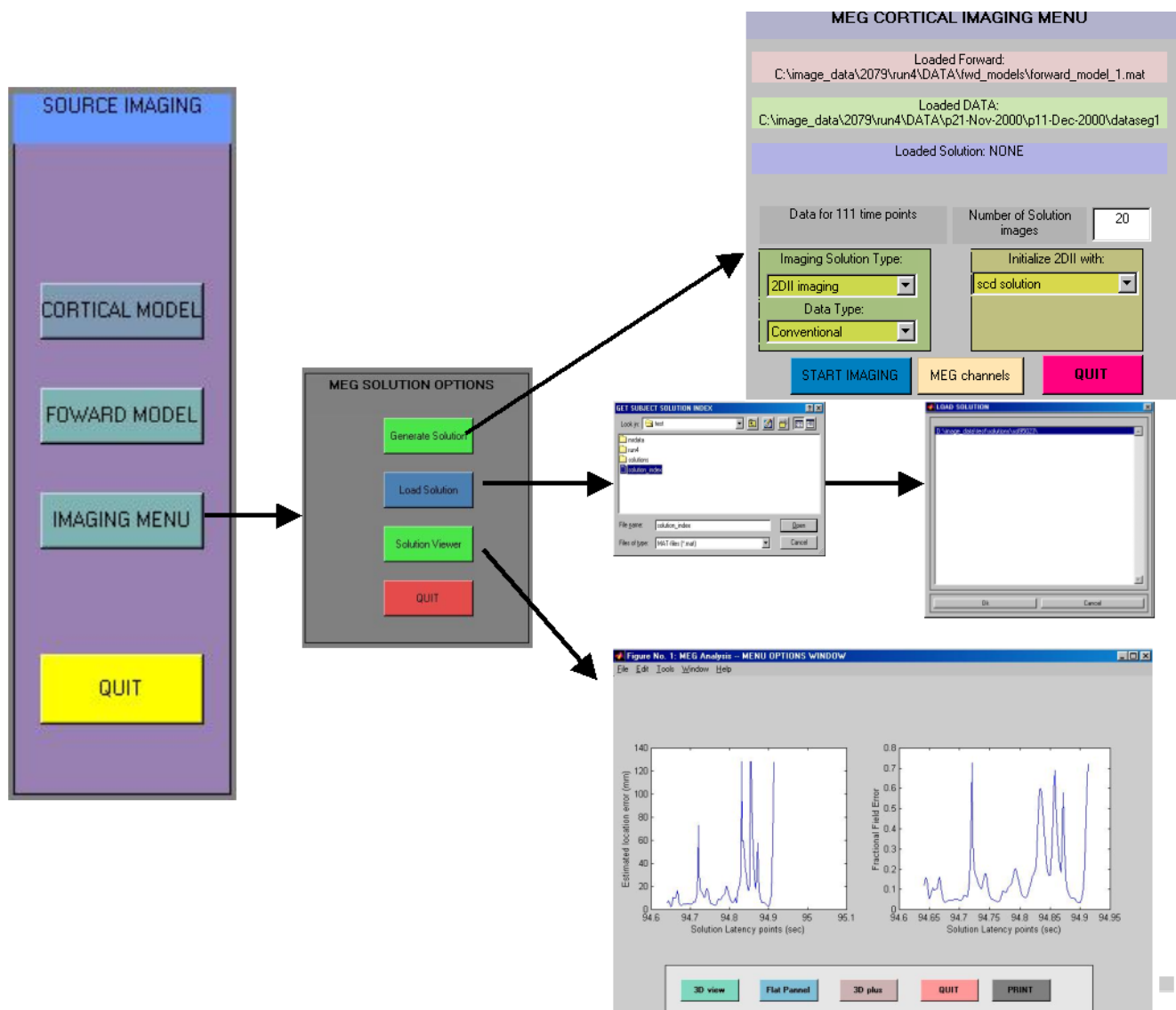
Noise filtered time derivative data

2. Load a previously created **SOLUTION** for viewing and analysis.

3. Use the **SOLUTION VIEWER** to view loaded SOLUTION STRUCTURE.

Viewer Formats:

- a. 3D Head with active source overlay with MEG data time series and contour plots.
- b. 3D Head with active source overlay only.
- c. Flat panel overlay of active sources on MRI slices.



Source Imaging Graphical Interface:

Press <Imaging Menu> button to:

1. Access the MEG CORTICAL IMAGING MENU
2. Load a previously generated solution.
3. Access the SOLUTION VIEWER MENU

Three source imaging techniques are implemented:

- (A) Two Dimensional Inverse Imaging, (2DII), is capable of imaging small compact source activity and extended cortical source activity. Faster calculation time with time continuity constraint.
- (B) Multi-Resolution FOCUS similar imaging capabilities as 2DII but less resolution of focal sources. However, calculation of source solutions are performed faster than corresponding 2DII solution. Least calculation time with time continuity constraint on solution amplitudes.
- (C) The Single current dipole metric is the goodness of fit corresponding to a single current dipole solution calculated at each point of the cortical model. (Rapid solution calculation time)

MEG CORTICAL IMAGING MENU

Loaded Forward:
C:\image_data\2079\run4\DATA\ fwd_models\forward_model_1.mat

Loaded DATA:
C:\image_data\2079\run4\DATA\p21-Nov-2000\p11-Dec-2000\dataseg1

Loaded Solution: NONE

Data for 111 time points Number of Solution images: **111**

Imaging Solution Type:
Multi-resolution Focus

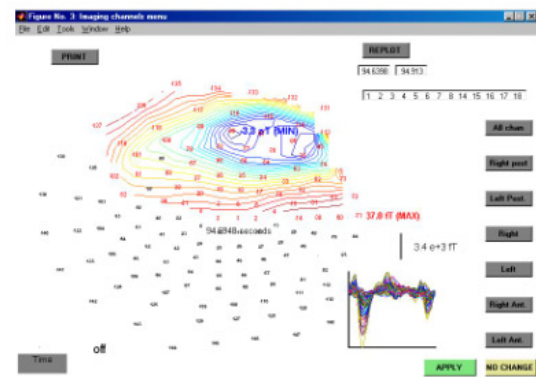
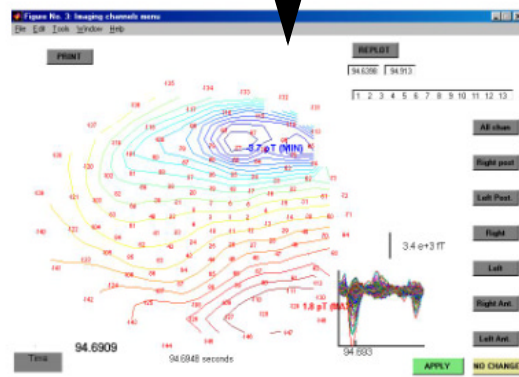
Data Type:
Conventional

Initialize 2DII with:
time series correlation

2DII Initialization Weights:
uniform random

START IMAGING MEG channels QUIT

Set to 111 solution image points which matches the number of data points in the MEG data segment

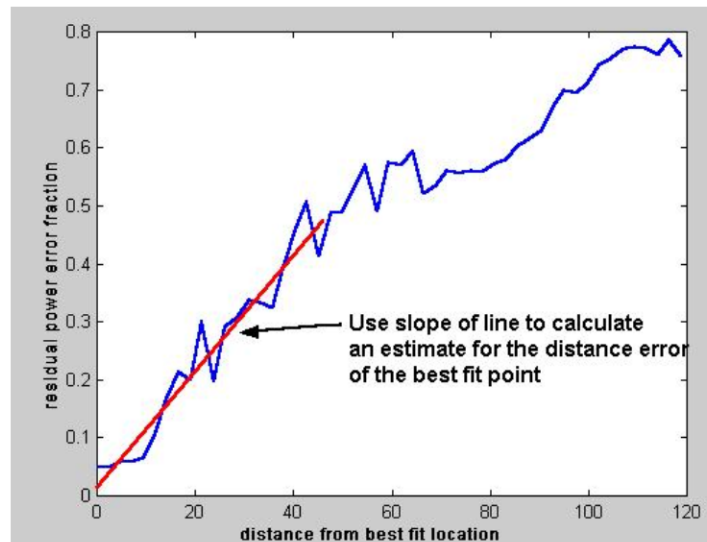


GENERATE A SOLUTION

- S1. Push the <Generate Solution> button of the MEG SOLUTION OPTIONS menu. This will open the MEG CORTICAL IMAGING MENU.
- S2. The Number of Data time points is listed. Change the number of solutions desired to a value less than or equal to the number of Data time points. The program will choose a time point spacing between solution points to be an interger multiple of the Data sampling rate that will generate a number of solution close to your requested number.
- S3. Push the <MEG channels> button to produce a contour map of the loaded data. Select the channels to be used to generate the solution. The initial contour map includes the time series plots of each channel on the sensor layout plot, therefore, bad channels can be located and deleted by using the mouse to toggle the channel off. The buttons on the right can be used to select specific regions and additional channels can be included or excluded by using the mouse to toggle channels on/off. Push <apply> button when satisfied with the channels selected or <no change> if more appropriate. The time point used in the contour map can be changed by clicking the mouse arrow pointer on the small time series graph (lower right corner) such that the pointer is a full cross hair then clicking this graph at the desired latency. (See figure above)

SCD METRIC (single equivalent dipole goodness of fit metric)

1. After performing steps S1-S3 described on the previous page, push the <SCD Metric> button.
 - a At each of the approximate 3000 source locations in the cortical model, the appropriate X,Y,Z gain vectors of the loaded FORWARD_MODEL structure are used to calculate the best fit single equivalent dipole amplitude, (qx qy qz), and goodness of fit = $1 - r^2/b^2$ where b is the magnetic field data at a specific latency, $r = b - b_{\text{forward}}$, b_{forward} is the forward solution which minimizes r^2 . The technique used to calculate the forward solution for each source location is detailed in the reference appendix, (Source Space Minimization Technique for MEG Source Analysis). In order to dramatically decrease execution time, this technique was implemented as a C++ dll application integrated into matlab with their mex file application interface.
 - b In addition to the x,y,z source amplitudes, at each latency, the program calculates the rate at which the goodness of fit changes with the distance away from the location with the best goodness of fit, (the single equivalent dipole solution point). This rate is used to estimate the distance error of this best fit location, (where it is assumed location error is the only factor decreasing the goodness of fit to less than 1.0). (see figure below for an example).



- c. A list of the best single equivalent source locations can be obtained by typing the m-file command, **img_get_bestfits** <Enter> at the prompt in the main MATLAB command window. This allows the entire list or selected latencies of best goodness of fit locations and corresponding source amplitudes to be copied and pasted to other documents as desired. The 5 columns of the command window output are for the best goodness of fit points:

latency (seconds) , X Y Z location (cm, MEG coordinates), distance error]

followed by:

latency (seconds), X Y Z amplitude (nanoAmp-m) , distance error]

SCD Metric SOLUTION structure

TYPE: 'SCD'
INIT_TYPE: []
INIT_ID_TAG: []
INIT_NAME: []
INIT_DIRECTORY: []
CHANNELS_USED: [66x1 double]
NUM_TIMEPTS: 139
SOURCES: [1x1 struct]
FWD_PARENT: [1x66 char]
ID_TAG: '95023160748689176245'
DATA_PARENT: 'D:\image_data\test\run4\DATA\p07-Sep-2000\dataseg1.mat'
NAME: 'solution_1'
DIRECTORY: 'D:\image_data\test\solutions\sol95023\
X_AMP: [2919x139 double]
Y_AMP: [2919x139 double]
Z_AMP: [2919x139 double]
ERROR: [2919x139 double]
DATA_TIMEPTS: [1x139 double]
DIST_ERROR: [1x139 double]
XYZmatrix: [51x139 double]
ERRORmtx: [51x139 double]

2DII (Two Dimensional Inverse Imaging)

1. Perform steps 1-3 described above to select the number of solution time points and the channels to be used. The 2DII technique is capable of imaging simultaneously active source structures. Therefore, usually all the MEG channels should be utilized. However, if the cortical model includes only a portion of the complete cortex then it is important to restrict the MEG channels used to those containing significant signals from the restricted source space. Restricting the source model causes the 15 source model slices to be spaced closely such that the spatial resolution of the source model is increased. The channels used should be matched to the distribution of source locations in the cortical source model.
2. Choose the method of initializing the 2DII source solution. The 2DII technique requires an initial estimate of the cortical source structure. This initial estimate is altered to obtain the final solution by performing a sequence of focal and global cortical source amplitude perturbations which simultaneously reduces the residual field error and reduces the correlation of this residual field with each individual X,Y, and Z Gain matrix component, (Gain matrix is stored in the FORWARD_MODEL structure). At the present time, three methods of initializing the 2DII solution are implemented:
 - a. SCD metric (goodness of fit) amplitudes for each source
 - b. Random source amplitudes with uniform weight bias or SCD metric bias.
 - c. Time continuity constraint on source amplitudes combined with either (a) or (b).However, future development is planned such that fMRI and PET data can be used to initialize 2DII

solutions. As a result, these 2DII solutions will satisfy both MEG and fMRI, PET, etc. constraints.

2DII SOLUTION

SCD goodness of fit metric initialization.

Initialize with: **SCD solution** —————> <Start Imaging> button (See figure below.)

At each latency the SCD goodness of fit metric and single current dipole source orientation are calculated for each location of the source model. The 2DII initializes each source location with an amplitude set proportional to the goodness of fit but with the single equivalent dipole source orientation. This initialization biases the final solution toward including those source locations with a better goodness of fit. Thus, it is especially useful for imaging combinations of small discrete source structures. Only a single 2DII source solution is calculated for each latency, (10 iterations per solution). Therefore, a sequence of 2DII solutions is calculated relatively quickly.

During each of the 10 iterations of a 2DII solution:

- A multi resolution decomposition is performed for the cortical source structure,

$$\mathbf{q} = (\mathbf{q}_x \mathbf{q}_y \mathbf{q}_z) \text{ with forward solution, } \mathbf{b}_q.$$

- Then the best decomposition component, $\mathbf{q}_P = (\mathbf{q}_{Px} \mathbf{q}_{Py} \mathbf{q}_{Pz})$ is selected

$$\text{with a forward solution, } \mathbf{b}_P = (\mathbf{b}_{Px} \mathbf{b}_{Py} \mathbf{b}_{Pz})$$

$$\text{that minimizes } (\mathbf{b} - \alpha \mathbf{b}_q - \beta_x \mathbf{b}_{Px} - \beta_y \mathbf{b}_{Py} - \beta_z \mathbf{b}_{Pz})^2$$

where \mathbf{b} = magnetic field data

$$\mathbf{q} \text{ is updated for next iteration: } \mathbf{q}_{\text{update}} = \alpha \mathbf{q} + \beta_x \mathbf{q}_{Px} + \beta_y \mathbf{q}_{Py} + \beta_z \mathbf{q}_{Pz}$$

and $(\alpha \beta_x \beta_y \beta_z)$ are least square fit coefficients determined using the methods documented in the Reference appendix, (Source Space Minimization Technique for MEG Source Analysis)

Initialize 2DII solution with
scd metric

2DII SOLUTION

Random amplitude initialization source structure and:

Source amplitude weight options:

uniform random source amplitudes

SCD metric amplitude random weights

User defined amplitude random weights

a *with uniform random weights:*

Initialize with: **Random amplitudes** —————> Initialization weights: **Uniform Random**. —————>
<**Start Imaging**>

In this initialization scheme:

- First, each cortical source in the structure is initialized with unit amplitude and an orientation corresponding to the single equivalent dipole least-square solution at the source location.
- Next, the x, y, and z component amplitudes are separately multiplied by random amplitudes with an expected mean value of zero and variance of 1, (generated by MATLAB function randn).
- Finally, twenty 2DII source solutions, (10 iterations per solution), are generated and averaged for each latency. Thus the mean initialization amplitude of each source location is approximately zero such that the final composite 2DII cortical source solution is independent of any initialization structure. This initialization scheme is useful when no estimate of the true source structure is available, (such as spontaneous MEG data). However, these 2DII solutions will still be biased toward sources whose forward solution is most correlated with the magnetic field data. Also, at latencies where the MEG signal is dominated by noise, the source amplitudes may become unstable. As a general rule, if the residual field error is greater than 10 to 15 percent, significant instabilities of the source amplitude between time points can be expected. This instability can be reduced or eliminated by initializing the 2DII solution with the TIME SERIES CORRELATION Technique.

b *with SCD solution weights:*

Initialize with: **Random amplitudes** —> Initialization weights: **SCD Solution**. —> <**Start Imaging**>

This is a more robust version of the SCD metric initialization. Each source is initialized with the SCD metric solution calculated for it's location. Next, each x,y,z component is multiplied by a random amplitude generated by the Matlab function randn. As with the uniform weighted initialization, for a large set of these initialization structures, the expected mean source amplitudes are zero. Therefore, the corresponding average 2DII solution does not contain a direct contribution from an initialization structure. However, the mean absolute amplitude of each source is proportional to it's SCD metric amplitude. Thus, the corresponding average 2DII source solution reflects the SCD metric weight factors. At each time point of the data to be imaged, a set of 20 2DII solutions are generated and averaged.

c *with user chosen amplitudes:*

Initialize 2DII with: **Random amplitudes** —> Initialization weights: **User File** —> <**Start Imaging**>

This option allows the user to import custom weights for the 2DII solution. These weights could be based on fMRI data etc. The number of rows in this data file matrix must equal the number of source in the SOURCES model and the number of columns must match the number of time slices of data that will be imaged. As with SCD initialization weights, these amplitudes are multiplied by random amplitudes and a set of 20 2DII solutions are generated and averaged for each time point imaged.

A file open box is used to select the user supplied matrix file in ASCII or matlabfile.mat format.

MEG CORTICAL IMAGING MENU

Loaded Forward:
C:\image_data\2079\run4\DATA\ fwd_models\forward_model_1.mat

Loaded DATA:
C:\image_data\2079\run4\DATA\p21-Nov-2000\p11-Dec-2000\dataseg1

Loaded Solution: NONE

Data for 111 time points Number of Solution images: 20

Imaging Solution Type: 2DII imaging
Data Type: Conventional

Type of Initialization: random amplitudes
Initialization Weights: uniform random

START IMAGING MEG channels QUIT

2DII solution:

Initialized with Random amplitudes

and one of the following options:

- a) uniform amplitude weights
- b) SCD amplitude weights
- c) user defined weights

2DII solution

user file initialization structure:

Initialize with: **Obtain from file** -----> <Start Imaging>

After pressing the start image button a file open menu appears. Use this menu to select the file containing the initialization structure amplitudes for each latency that is to be imaged, (a single amplitude for each source location at each latency). The source orientations of the initialization structures is set to the orientation of the optimal equivalent current dipole at each source location. Therefore, the data matrix in the user file must have the number of rows equal to the number of sources in the SOURCES structure and the number of columns must be equal to the number of time slices of data that will be imaged for the solution. The user file can be located anywhere that is accessible with the file open utility and can be ASCII with blank space separators or a matlab_file.mat format.

A file open menu is used to select the desired file.

MEG CORTICAL IMAGING MENU

Loaded Forward:
C:\image_data\2079\run4\DATA\ fwd_models\forward_model_1.mat

Loaded DATA:
C:\image_data\2079\run4\DATA\p21-Nov-2000\p11-Dec-2000\dataseg1

Loaded Solution: NONE

Data for 111 time points Number of Solution images: 20

Imaging Solution Type: 2DII imaging
Data Type: Conventional

Type of Initialization: obtain from file

START IMAGING MEG channels QUIT

2DII solution initialized with user file

2DII solution:

Time continuity initialization

Initialize with: time continuity constraint

Initialization weights: Uniform Random or User File or SCD Metric —————> <Start Imaging> button

One of the main deficiencies of the 2DII imaging technique is that the solution of the magnetic inverse problem at a specific latency is performed independent of all other latencies. Thus, at latencies when the signal to noise amplitude is low the 2DII solution can exhibit large discontinuities of source structure amplitude between adjacent time samples of data. This behavior can be greatly reduced by introducing a continuity of change constraint on the source structure.

For sources that change amplitude smoothly with time in a segment of MEG data, the source structure, \mathbf{q}_{tk} at time, t_k , can be represented as a weighted sum of the source structure at all sampled time points in the data segment interval, (for a sufficiently long interval).

$$\boxed{\mathbf{q}_{tk} = \sum \mathbf{w}_{jk} \mathbf{q}_{tj}} \quad \text{summed for all time interval points}$$

Therefore, for the MEG data at time t_k

$$\mathbf{b}_{tk} = \mathbf{G} \mathbf{q}_{tk} \longrightarrow \boxed{\mathbf{b}_{tk} = \sum \mathbf{w}_{jk} \mathbf{b}_{tj}}$$

This system of equations, ($k = 1, \dots, n$), can be solved for the weights, \mathbf{w}_{jk} , from the MEG data alone.

These weights are incorporated into the 2DII source solution performing a single iterative step solution for each time point of MEG data. Then, the time continuity constraint is applied. Using the continuity weighting factors, \mathbf{w}_{jk} , and the individually calculated 2DII source structures, \mathbf{q}_{tk} , a continuous solution is generated which is used as an initialization structure for the next iterative step, (see equation for \mathbf{q}_{tk} above). This technique can be used with the Multi-Resolution FOCUSS source solution also.

Initialization Weight Factors with time continuity constraint:

- Uniform Random: At each time point, a set of 20 2DII source solutions with the uniform random source initialization structure described previously are generated with only one iterative step and subsequently averaged into a single 2DII solution. Next, the time continuity constraint is applied to this sequence of solutions, as described above. Finally, the last 9 iterative steps are performed with the time continuity constraint applied at each of these steps except the last. This initialization is good for spontaneous MEG data and epileptic MEG seizure data.
- SCD metric initialization: Each time point is initialized with the SCD metric source structure; (The 2DII initializes each source location with an amplitude set proportional to the goodness of fit but with the single equivalent dipole source orientation) Next, 10 iterative steps of the 2DII algorithm are performed with the time continuity constraint applied on the first 9 steps. This initialization structure is especially useful for evoked response data.
- User file initialization: This initialization procedure is identical to the SCD metric initialization except the initialization structure amplitudes are supplied by the user for each time slice imaged. A file open menu is utilized to access this file in ASCII or matlabfile.mat format.

2DII solution with time continuity constraint

MEG CORTICAL IMAGING MENU

Loaded Forward:
C:\image_data\2079\run4\DATA\ fwd_models\forward_model_1.mat

Loaded DATA:
C:\image_data\2079\run4\DATA\p21-Nov-2000\p11-Dec-2000\dataseg1

Loaded Solution: NONE

Data for 111 time points Number of Solution images: 20

Imaging Solution Type:
2DII imaging

Data Type:
Conventional

Type of Initialization:
time continuity constraint

Initialization Weights:
uniform random

START IMAGING MEG channels QUIT

Type of Initialization:
Time continuity constraint

Initialization Weights:
uniform random
user file
SCD metric

Multi-Resolution FOCUSS

FOCUSS is an iterative imaging technique that utilizes an a weighted minimum norm solution at each iterative step. Therefore, with the source structure, \mathbf{q} , determined by the solution of $\mathbf{b} = \mathbf{G}\mathbf{q}$, the FOCUSS algorithm becomes:

$$\mathbf{b} = \mathbf{G}\mathbf{W}\mathbf{W}^{-1}\mathbf{q}_k \text{ for the } k^{\text{th}} \text{ iterative step}$$

where

$$\mathbf{W} = \text{diag}(\mathbf{q}_{k-1}) = \text{diagonal matrix of source amplitudes of previous step}$$

and

$$\mathbf{q}_k = \mathbf{W}(\mathbf{G}\mathbf{W})^+\mathbf{b} = \mathbf{W}\mathbf{W}^T\mathbf{G}^T(\mathbf{G}\mathbf{W}\mathbf{W}^T\mathbf{G}^T)^{-1}\mathbf{b}, \quad (\mathbf{G}\mathbf{W})^+ = \text{pseudoinverse of } \mathbf{G}\mathbf{W}$$

In the above algorithm the vector has thousands of individual source amplitudes and regularization techniques are required to perform the necessary matrix inversion.

The Multi-Resolution FOCUSS algorithm avoids these matrix inversion problem by utilizing the 2DII multi resolution decomposition of the source structure to construct a low resolution projection matrix \mathbf{W}_P with the structural resolution chosen as eight source populations, (16 or 32, etc. source population resolution could have been selected). Therefore, introducing this subspace projector in the above equation.

$$\mathbf{b} = \mathbf{G}\mathbf{W}\mathbf{W}_P\mathbf{W}_P^T\mathbf{W}^{-1}\mathbf{q}_k \longrightarrow \boxed{\mathbf{b} = \mathbf{G}_P\mathbf{q}_p^k}$$

where

$$\mathbf{q}_p^k = \mathbf{W}_P^T\mathbf{W}^{-1}\mathbf{q}_k \text{ and } \mathbf{G}_P = \mathbf{G}\mathbf{W}\mathbf{W}_P \text{ for the } k^{\text{th}} \text{ iterative step}$$

and

$$\mathbf{W}_P\mathbf{W}_P^T = \text{block diagonal matrix, (after reordering sources in } \mathbf{q}_k)$$

with block diagonal element amplitudes = N_j^{-1} where N_j is the number of sources in the j^{th} source population of the eight populations created.

The vector, \mathbf{q}_p^k , has only eight sources, each with x, y, and z orientations. Therefore, only 24 source structure amplitudes are calculated each iterative step. A least squares solution for these source amplitudes are calculated directly without a matrix inversion using the Source Space Minimization Technique described in the APPENDIX. Finally, the full source structure estimate of the k^{th} step is equal to:

$$\boxed{\mathbf{q}_k = \mathbf{W}\mathbf{W}_P\mathbf{q}_p^k}$$

Therefore, this technique differs from the 2DII technique only in the number of source structures that are altered each iterative step. Thus, the Multi-Resolution FOCUSS algorithm performs less focal alterations of the source structure than 2DII and the final source solutions are, in general, less focal than the corresponding 2DII source solution. Also, the Multi-Resolution FOCUSS algorithm converges more rapidly than 2DII and iteration must be terminated to avoid including source structure reflecting noise that can actually degrade accuracy.

In summary

The Multi-Resolution FOCUSS technique produces less focal source solutions than the 2DII technique. However, the Multi-Resolution source solutions are generated more rapidly than the corresponding 2DII technique.

The source structure initialization options available for the 2DII technique are available for the Multi-Resolution FOCUSS technique. FOCUSS Reference: Gorodnitsky I.F., George J.S., Rao B.D., Neuromagnetic source imaging with FOCUSS: a recursive weighted minimum norm algorithm, Elcetroencephal. Clin. Neurophysiol. 95 pp231-251(1995)

Multi-Resolution Focuss

MEG CORTICAL IMAGING MENU

Loaded Forward:
C:\image_data\2079\run4\DATA\ fwd_models\forward_model_1.mat

Loaded DATA:
C:\image_data\2079\run4\DATA\p21-Nov-2000\p11-Dec-2000\dataseg1

Loaded Solution: NONE

Data for 111 time points Number of Solution images: 111

Imaging Solution Type:
Multi-resolution Focuss

Data Type:
Conventional

Type of Initialization:
time continuity constraint

Initialization Weights:
uniform random

START IMAGING MEG channels QUIT

Type of Initialization:
random amplitudes
scd solution
obtain from file
time continuity constraint

Initialization weights:
uniform random
user file
SCD metric

Imaging solution type:
Multi-resolution FOCUSS

Multi-resolution FOCUSS: This cortical source imaging technique utilizes the same source initialization options as the 2DII imaging technique. Descriptions of these source initialization and amplitude weight factors are described in the 2DII section of this user manual. The influence of the source initialization on the final source solution is similar for both imaging methods. In general, the 2DII imaging technique produces more focal source structure solutions than the Multi-resolution FOCUSS technique. However, Multi-resolution FOCUSS source solutions are generated more quickly than a corresponding 2DII technique with the same initialization source structure.

Data Types: available data transformations

MEG CORTICAL IMAGING MENU

Loaded Forward:
D:\image_data\7013\gary1050\1\DATA1\ fwd_models\forward_model_1.mat

Loaded DATA:
D:\image_data\7013\gary1050\1\DATA1\p14-Jan-2001\dataseg1

Loaded Solution: D:\image_data\7013\solutions\sol87481\solution_1

Data for 90 time points Number of Solution images: 90

Imaging Solution Type:
2DII imaging

Data Type:
Conventional

Type of Initialization:
scd solution

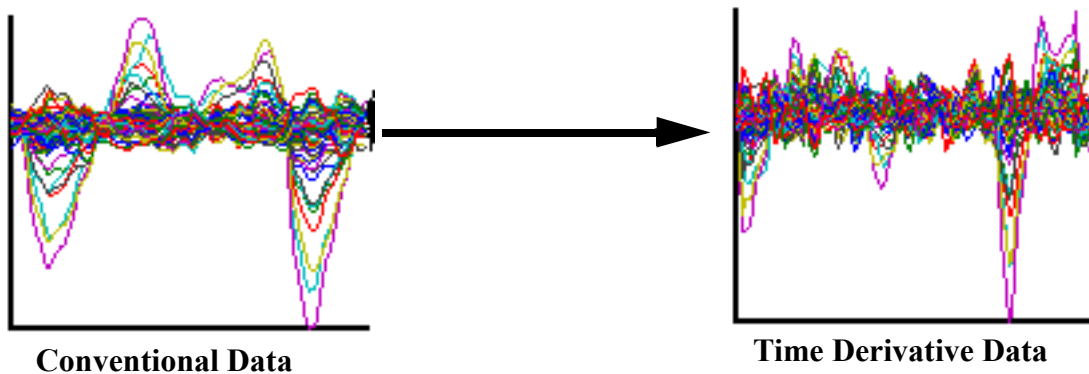
START IMAGING MEG channels QUIT

Data Type: Conventional
Time Derivative
Conventional + Noise Filter
Time Derivative + Noise Filter

Data Types available for cortical source imaging:

- a. **Conventional time series data:** This is the data as imported into the EPOCH data structure of the MEG_TOOLS program.
- b. **Time Derivative data:** A eleventh order differentiator is applied to the conventional time series data prior to generating the sequence of cortical source solutions by the imaging method of choice. Thus, on/off responses of cortical sources are selectively imaged using time derivative data. Further, on/off phases of cortical source activity are brief thus temporal overlap of source activity is reduced. Also, the size of cortical sources tend to be small and focal during the on/off intervals. Therefore this data transformation can be useful for imaging neuronal activity at the onset of epileptic activity.

Note: This transformation is not appropriate for very short time segments of data because of filter convolution artifact for the first and last ten time points of data. These beginning and end segment of filtered data must not be utilized. Also, the signal to noise ratio may be reduced if high frequency noise is not filtered from the data before or after applying this filter. An increase in the fraction of high frequency noise and signal, as well as, the signal distortion at the beginning and end of the data segment is observed in the figure below.



Data Types available for cortical source imaging:

c. **Noise filtered data:** This transformation of the MEG data is closely related to an inverse filter derived from a variety of source imaging optimizations, such as minimization of expected source error and Wiener filter operator, (primary references: Liu A.K., Belliveau J.W., Dale A.M., Spatiotemporal imaging of human brain activity using functional MRI constrained magnetoencephalography data: Monte Carlo simulations Proc. Natl. Acad. Sci. USA 95 pp/ 8945-8950,1998).

The Inverse operator is applied to the MEG data to obtain an estimate of the cortical source structure:

with $\mathbf{q} = \mathbf{F}\mathbf{b}$ where $\mathbf{F} = \mathbf{Q}\mathbf{G}^T(\mathbf{G}\mathbf{Q}\mathbf{G}^T + \mathbf{N})^{-1}$ = optimal inverse filter

\mathbf{b} = $\mathbf{G}\mathbf{q}_a + \mathbf{n}$ time slice of magnetic field data

\mathbf{q} = estimate of cortical source structure for time slice of data

\mathbf{q}_a = actual source structure active at specified time point

\mathbf{n} = noise in the data \mathbf{b}

\mathbf{Q} = covariance estimate for source structure within the time interval of loaded MEG data.

\mathbf{N} = noise covariance matrix for same interval of data acquisition.

\mathbf{G} = Gain matrix for source structure and measurement array

$\mathbf{G}^\# = \mathbf{G}^T(\mathbf{G}\mathbf{G}^T)^{-1}$ = pseudoinverse of the Gain matrix

$\mathbf{B} = \mathbf{G}\mathbf{Q}\mathbf{G}^T + \mathbf{N}$ = MEG data covariance matrix for the time interval of data.

From the above equations:

$$\mathbf{F} = \mathbf{Q}\mathbf{G}^T\mathbf{B}^{-1} = \mathbf{G}^\#(\mathbf{B} - \mathbf{N})\mathbf{B}^{-1}$$

Therefore:

$$\text{MEG forward solution with optimal noise filter}$$

$$\mathbf{G}\mathbf{q} = (\mathbf{B} - \mathbf{N})\mathbf{B}^{-1}\mathbf{b} = \mathbf{F}_{\text{noise}}\mathbf{b}$$

Next the data covariance matrix, \mathbf{B} , is decomposed into the eigenvector/eigenvalue format: $\mathbf{B} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$

then

$$\text{Optimal noise filter}$$

$$\mathbf{F}_{\text{noise}} = \mathbf{U}(\mathbf{\Lambda} - \mathbf{\Lambda}_N)\mathbf{\Lambda}^{-1}\mathbf{U}^T = \mathbf{U}\mathbf{\Lambda}_{\text{sig}}\mathbf{U}^T$$

The eigenvalue matrices, $(\mathbf{\Lambda}, \mathbf{\Lambda}_N, \mathbf{\Lambda}^{-1})$, are diagonal. Therefore, these are easily combined into a single diagonal matrix, $\mathbf{\Lambda}_{\text{sig}}$, with diagonal elements equal to: $\lambda_{si} = (\lambda_i - \lambda_{Ni})/\lambda_i$ with the choice of the noise eigenvalues dependent on the estimate of noise in the data.

The noise filter implemented in the MEG Tools imaging software chooses the noise eigenvalues, λ_{Ni} , such that diagonal elements of $\mathbf{\Lambda}_{\text{sig}}$ are

$$\text{diag}\mathbf{\Lambda}_{\text{sig}} = \lambda_i / (0.9\lambda_i + 0.1\lambda_{\text{largest}})$$

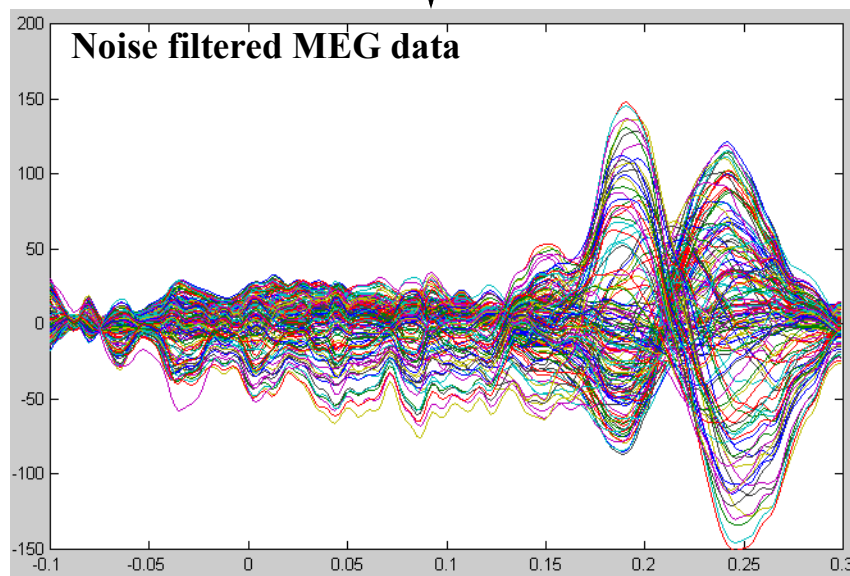
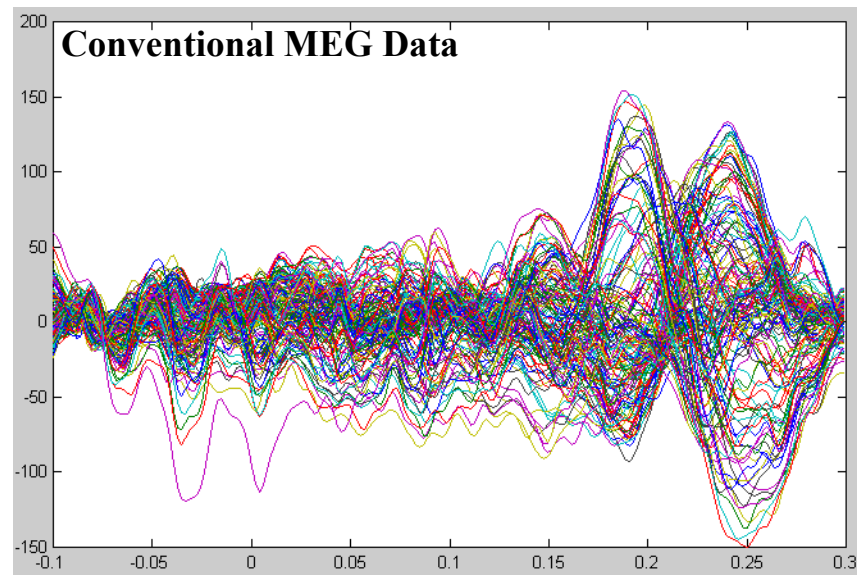
After applying this filter to the data, $\mathbf{G}\mathbf{q} = \mathbf{F}_{\text{noise}}\mathbf{b}$, the source solution is generated using 2DII or Multi resolution FOCUSS or the Single current dipole metric.

Noise filtered data:

The filter can be applied to both conventional or time derivative data.

The present implementation assumes that the signal amplitude is greater than the noise in the data. This filter should not be used if the noise level is much larger than the data. Noise is not assumed to be uncorrelated. Rather, noise includes low amplitude contributions from ongoing background neuronal activity that are not of interest.

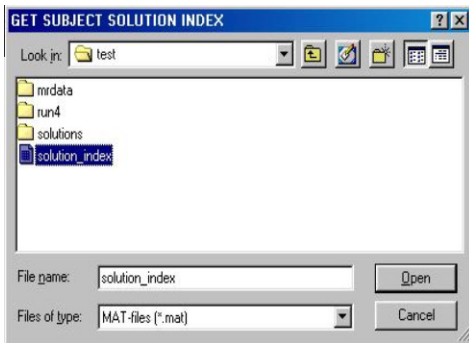
The effect of the filter on visual evoked MEG data is shown below.



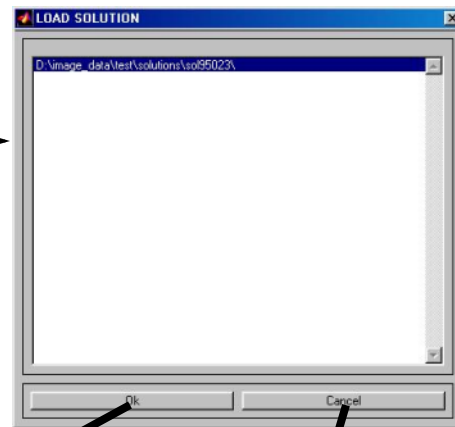
Loading a SOLUTION

The program keeps track of the solutions that have been generated for each subject in a file called solution_index.mat in the Subject_Directory.

1. To load a solution —> Open the solution_index for the specific subject. A second menu appears with a list of source solutions. Choose the desired solution then press <OK> button. (See the figure below.)
2. To dearchive (from CD-ROM etc.) and load a solution
 - a. Copy the specific solution directory (such as: sol83820 which contains the solution_1.mat file) into <drive>\image_data\subject_directory\solution\
 - b. Perform step 1 above (To load a solution), except the list of solutions will not contain the dearchived solution. Therefore, press the <CANCEL> button. This will cause a file open menu to appear. Locate and open the solution_1.mat file in the directory that was dearchived (such as the directory, sol83820). This will load the dearchived solution and update the solution_index file. (See the figure below).



<Open> Solution_index.mat



<OK> loads the highlighted solution

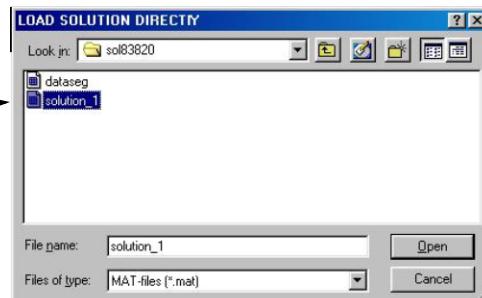
Example:

sol95023\solution_1.mat

**<Cancel> generates
FILE OPEN MENU
to load dearchived
SOLUTION**



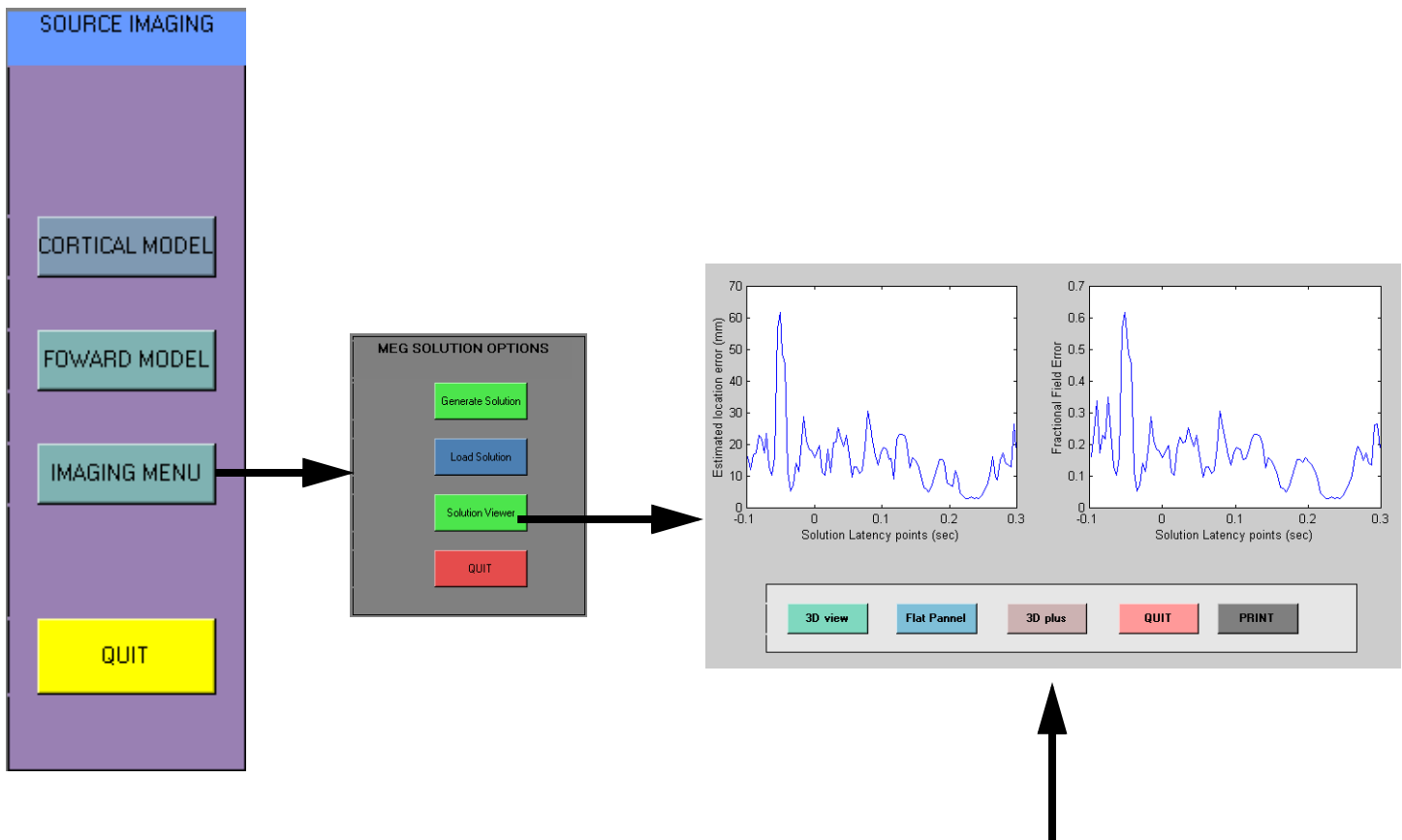
**Subject directory structure with
solutions**



SOLUTION VIEWER

This utility provides both 3 dimensional views and overlay MRI slice series views of the solution that is loaded in the program. (See previous page for loading a solution.)

Open the solution viewer menu by pressing <IMAGING MENU> —> <Solution Viewer> as shown below.



Solution viewer graphs

Left graph

- For SCD solutions, the estimated error of the best fit source location is displayed for each latency. (See generate a solution, SCD Metric, for a description of this parameter.)
- For 2DII solutions, the maximum amplitude of the source structure is displayed. This parameter is not particularly useful. Therefore, this graph will be probably be changed in future upgrades.

Right graph

- For both SCD and 2DII solutions, the fractional residual field power error is displayed. This is equal to $(1.0 - \text{goodness of fit})$.
- For 2DII solutions, this parameter is based on the entire 2DII source structure.
- For SCD solutions, this parameter is based only on the best fit source location.

Solution Viewer buttons

<3D view> This control generates a three dimensional view of the head derived from the subject MRI data that was created during MRI DATA IMPORT and COREGISTRATION. The top of the head is removed and the Solution is visualized within the head with an MRI slice used as a cortical structure reference.

- The amplitudes displayed for SCD solutions are proportional to the inverse of the fractional field error of the corresponding cortical location. This is a nonlinear scaling which enhances the visibility of locations with the best goodness of fit that is the most important parameter for assessing active sites.

Red corresponds to locations with the smallest residual field error. If the fractional field error of a point is 0.05 then the plot metric amplitude for this site is $1/0.05 = 20$.

Blue corresponds to 15 percent of the maximum display metric.

Source locations with a display amplitude less than 15 percent of the maximum are not displayed.

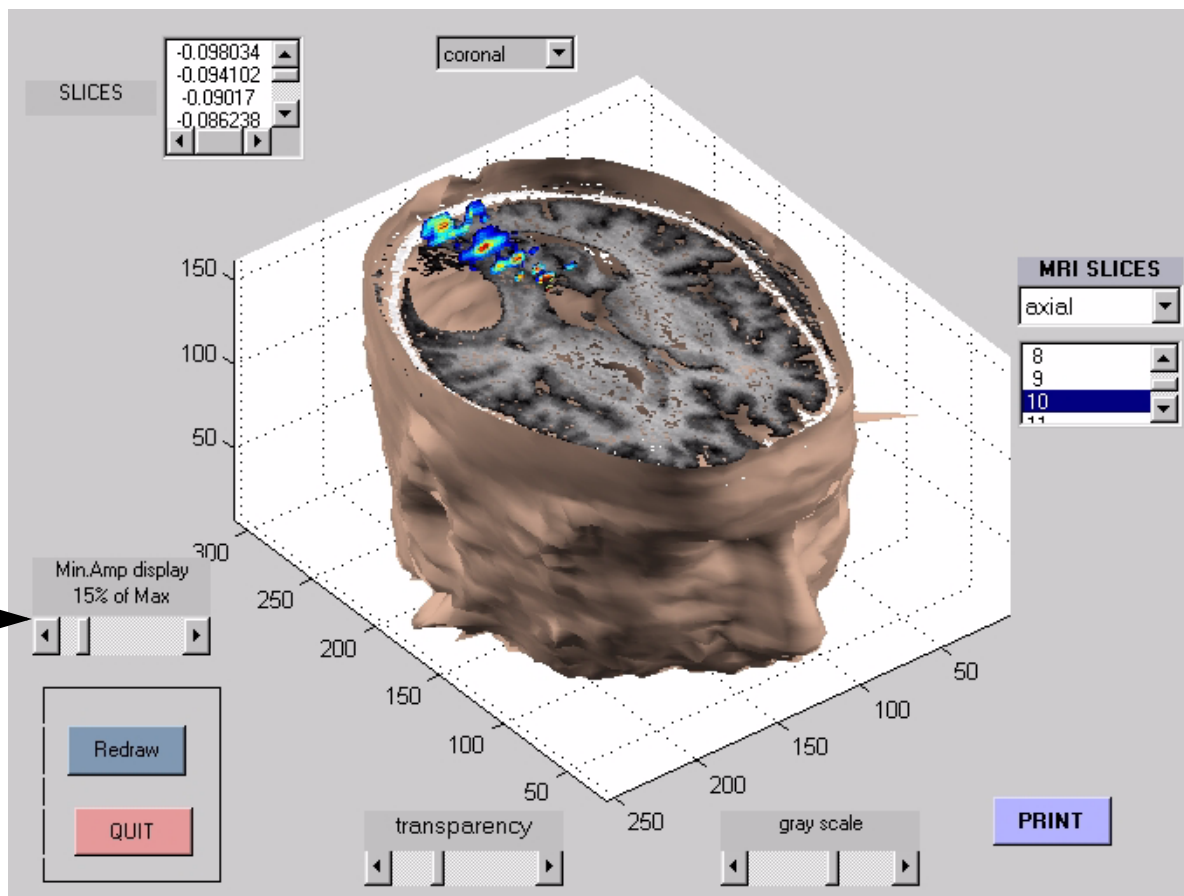
- The amplitudes displayed for 2DII solutions are proportional to the calculated 2DII structural amplitudes.

Red corresponds to points of the cortical source structure that have the greatest amplitude.

Blue corresponds to points of the cortical source structure at a threshold amplitude chosen using the MIN. AMP DISPLAY slider control of the figure, (Percent of maximum cortical source amplitude units).

Source locations whose amplitude is less than 15 percent of the maximum amplitude are not displayed.

- The 3D figure of the solution can be rotated by using the Window menu —> Tools —> Rotate 3D.
- The figure contains other buttons and edit menus that allow the display of the solution to be changed.
- Press <Redraw> to incorporate and changes made with other controls. (See the figure below)
- The Slices scroll menu is for False color SOLUTION slices.
- MRI SLICES controls the MRI slice that is displayed.
- TRANSPARENCY and GRAY SCALE control how the MRI slice is rendered.
- <PRINT> uses the print setup and page setup options set from the Figure window <FILE> menu bar.

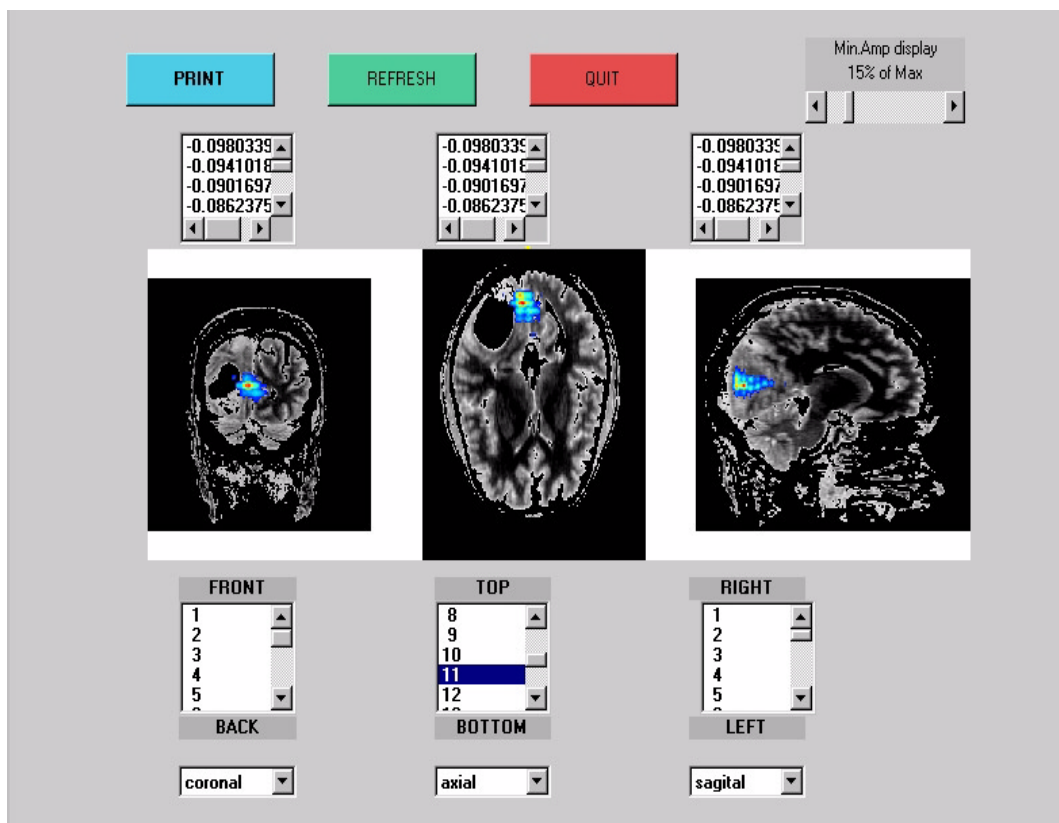


Flat Panel

<Flat Panel> Activation of this solution display option causes the solution to be displayed on a false color overlay on a set of three MRI slices, (default display is coronal, axial, sagittal). The slices and latencies of the solution in the individual slices can be changed as desired using the MENU CONTROLS then pushing the <REFRESH> button. (SEE FIGURE BELOW)

The default MRI SLICE VIEW ORIENTATION is:

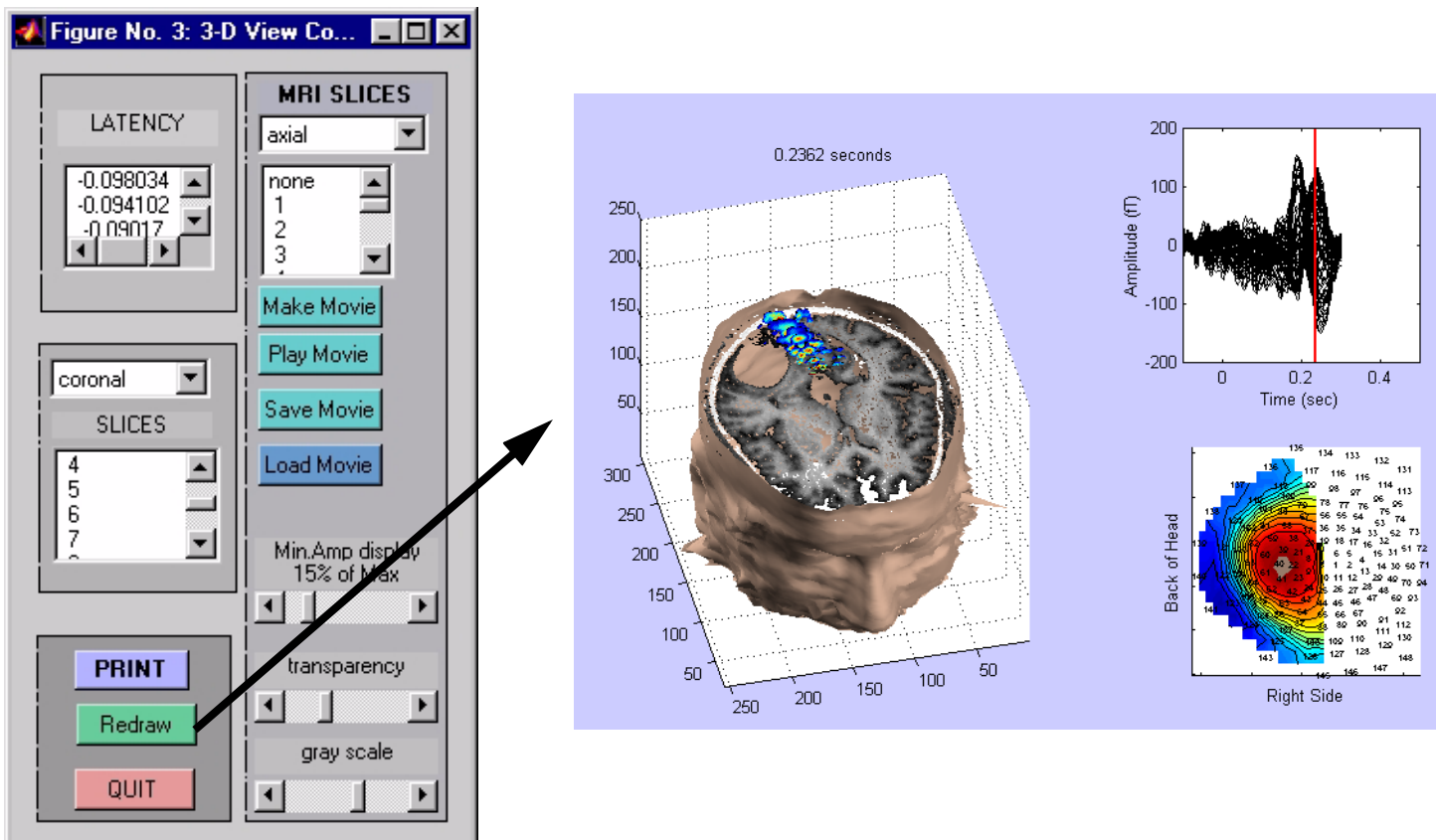
- CORONAL VIEW: The view is toward the subject from the front. Therefore, the subject's left ear is on the right of the MRI figure and the top of the head is at the top;
- AXIAL VIEW: The view is from above looking down on the top of the head. Therefore, the left ear is on the right of the MRI figure and the nose is at the bottom.
- SAGITAL VIEW: The view is toward the subject from the right side. Therefore the nose is on the right side of the MRI figure and the top of the head is at the top of the MRI figure.
- Solution color code is the same as for the three dimensional display previously described.
- <PRINT> Prints the three MRI views on one sheet without the GUI controls. Use the window <FILE> pull down menu to access <Print Setup> and <Page Setup> for further control of how these MRI views are printed.

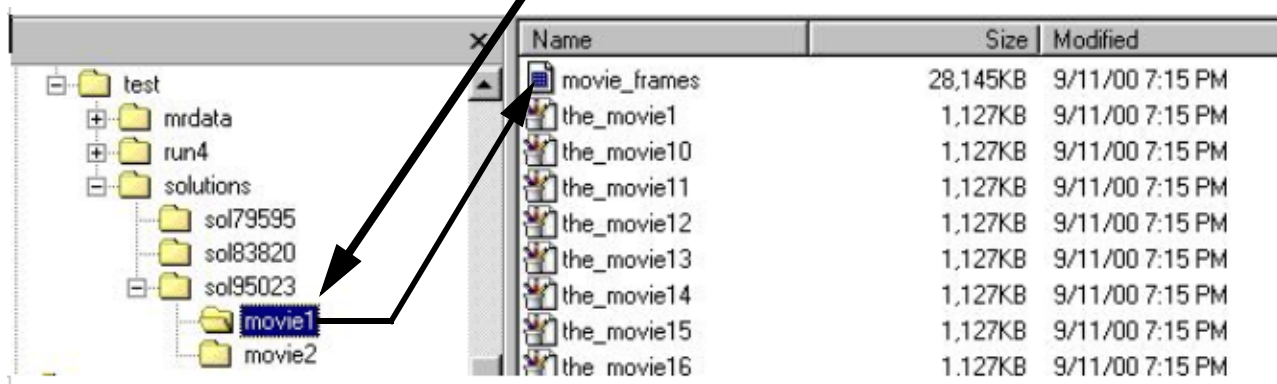
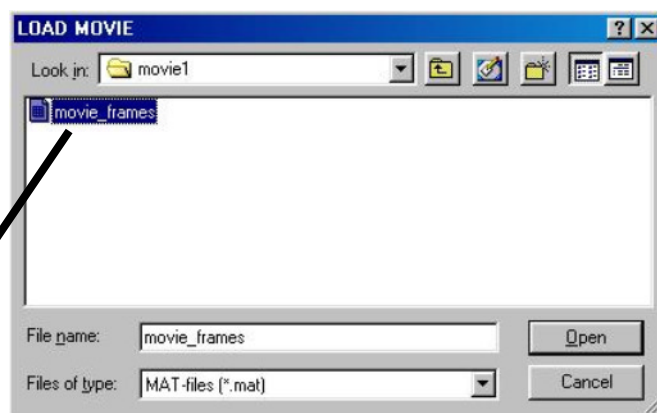
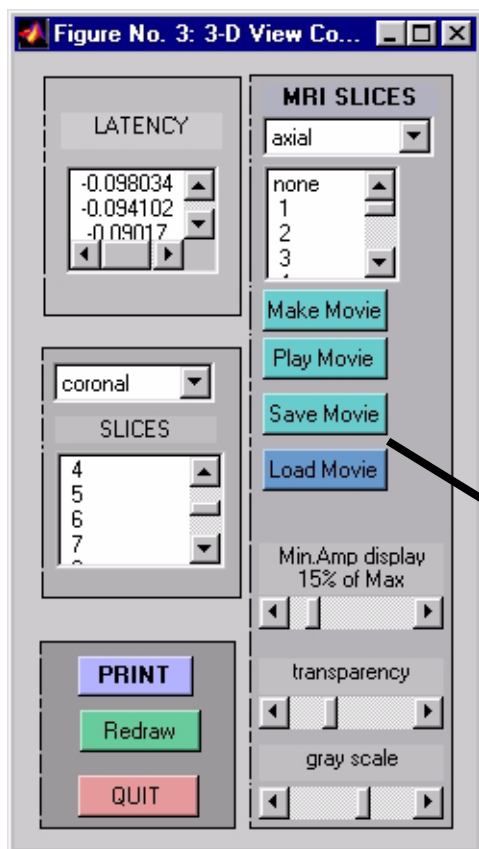


3D view plus time series plot and field contour plot

<3D plus> plots the three dimensional solution as well as the corresponding time series data and a contour map of the data at the latency corresponding to the displayed solution.

- The controls for this figure are in a separate window figure. This figure is moved to the bottom of the computer screen after the figure solution has been displayed. To make additional changes, use the mouse to drag the control menu back into full view.
- The three dimensional head can be rotated by dragging with the mouse, (choose on the window menu, tools —> Rotate 3D—> drag MRI/solution head to desired orientation. Alternatively, 3D rotation can be activated by toggling the rotation button on the tools menu bar.
- Change the other display parameters as desired. Then press **<Redraw>** to apply these changes.
- **<Make Movie>** button: This control sequentially plots the solution at each latency and saves the result in a frame buffer structure (threeD_movie in the matlab environment). In addition this utility activates a **<Play Movie>** and **<Save Movie>** button. The orientation and MRI slice parameters are the same as the figure has been adjusted to at the time of the Make Movie button selection.
- **<Play Movie>** Plays the movie that is loaded.
- **<Save Movie>** Saves the movie in a Subdirectory of the solution directory of the loaded solution. If previous movies have been made, the movie is placed in a new directory, (sequentially named, movie1 ... movie33, etc). In addition, each movie frame is saved as a separate bmp file, (the_movie25.bmp, etc), that can be utilized by other software to create a movie that can be played outside matlab.
- **<Load Movie>** Any movie from any subject and solution can be loaded and played.
- MRI display controls allow the slice orientation and location to be chosen as well as visibility.
- Contour Map and Time series Plot are for channels selected for imaging, (see figure below).





The directory tree for the location of the movie1\movie_frames.mat file is show for the subject identified as test.

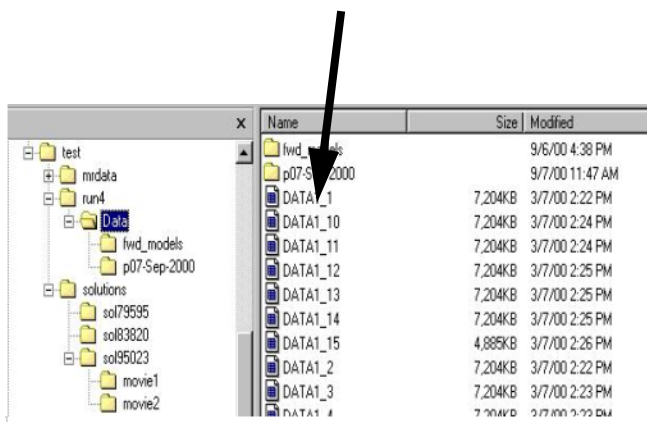
Appendix: Data Structures

Meg/EEG Data

MEG Tools creates the EPOCH data structure for these data during data import. The EPOCH structure contains the MEG/EEG sensor locations, (MEG head coordinates), in addition to the MEG and EEG data. The EPOCH structure can be displayed by typing <EPOCH> at the matlab command line as shown in right column.

EPOCH structures are stored in DATA subdirectories or sub-subdirectories, etc. of the corresponding SUBJECT directory located in the Image_data directory. A strict database is not implemented. However, this program creates the data directory structure based on specific rules. Therefore, moving files or renaming files and directories may cause the program to not function properly. If data storage questions arise, the author of this program should be contacted especially if trial and error will result in permanent loss of data. Long time series recordings of time series data are stored in a directory as a sequence of EPOCH structures, each contain 10 or 15 seconds of data. These structures are stored with names such as DATA1_1.mat and dataseg.mat. It is important not to change these names. The directory containing these files can be renamed. However, after forward models and source image solutions have been generated, the directory name should not be changed because the directory name is referenced in these structures and could cause confusion in your records if names do not match.

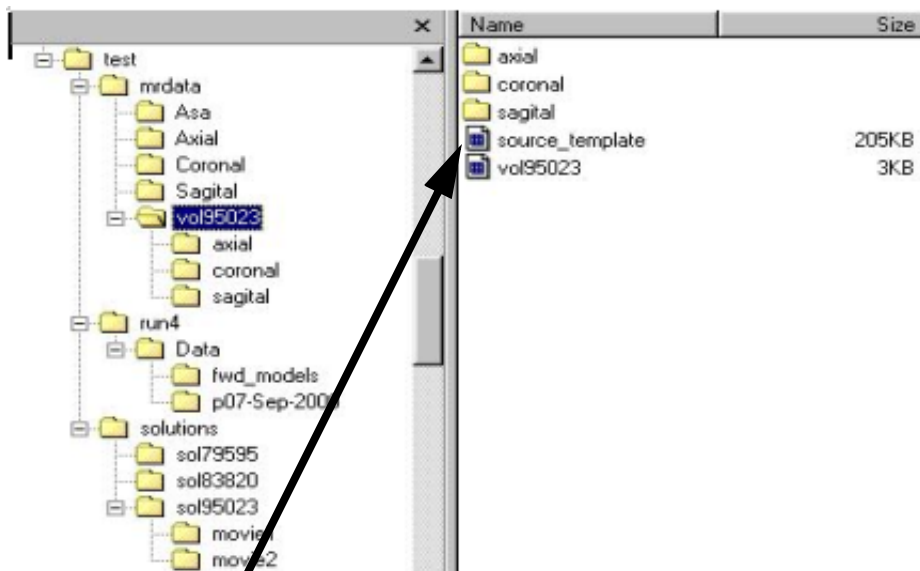
See EPOCH data files ,(Data1_#), below



EPOCH =

BTINAME: 'c,rhfp0.1Hz'
SAVENAME: 'DATA1'
SUBJECT_INFO: [4x24 char]
ID_T: 1
T_LABEL: 'TRIGGER'
ID_R: 2
R_LABEL: 'RESPONSE'
ID_U: 9
U_LABEL: 'UACurrent'
FIELD_ID: [8x1 char]
ID_MEG: [148x1 double]
MEG_LOC: [148x3 double]
MEG_DIR: [148x3 double]
MEG_WGHT: [148x1 double]
MEG_LABELS: [148x4 char]
ID_EEG: [8x1 double]
EEG_LOC: [8x3 double]
EEG_WGHT: [8x1 double]
EEG_LABELS: [8x3 char]
G_LOC: [5x6 double]
G_DIR: [5x6 double]
G_LABEL: [5x4 char]
M_LOC: [6x3 double]
M_DIR: [6x3 double]
M_LABEL: [6x4 char]
ID_M: [12 14 8 92 94 88]
ID_G: [19 21 13 15 11]
START_B: 0
END_B: 288.3236
EPOCH_PTS_B: 146650
CHANNEL_TYPE: [1x169 char]
CHAN_NO: [1x169 double]
PERIOD: 0.0020
LAST_B_EPH: 1
LAST_M_EPH: 20
PRESENT_B: 1
PRESENT_M: 1
START: 0
END: 14.9991
DATA: [169x7629 double]

Source Model Data Structure



SOURCES =

PARENT: 'D:\image_data\test\mrdata\coronal\'

ID_TAG: '95023160748689176245'

SLICE_LIMITS: [40 215]

COMPOSITION: [15x2 double]

LOCATION: [2919x3 double] **(In [ROW,COLUMN,SLICE] of coronal slice sequence)**

MEAN_SLICE_LOCATION: [1x15 double]

REGISTRATION: [9x4 double]

HEAD_TEMPLATE: [1x1 struct]

HEADSHAPE: [1x1 struct]

VOL_PATH: 'D:\image_data\test\mrdata\vol95023\vol95023'

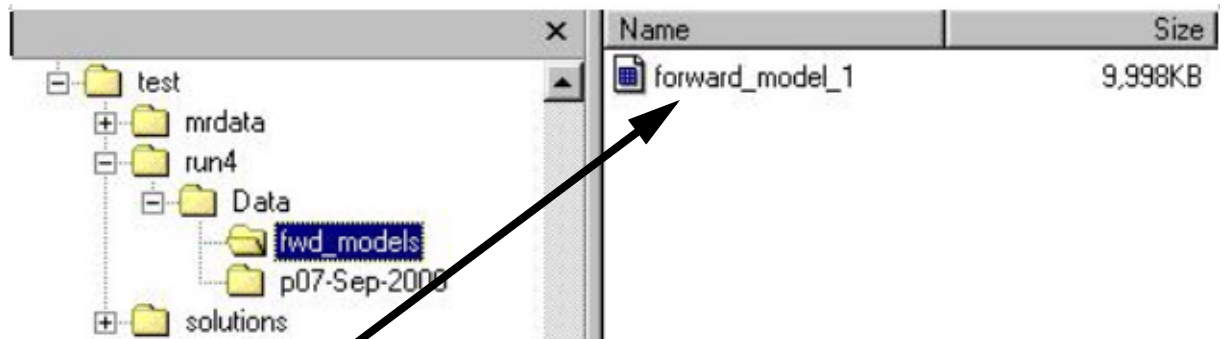
VOLSLICE_DIR: {1x3 cell}

SRC_NAME: 'source_template'

This structure is created using the <CORTICAL MODEL> utilities from the main menu.

The SOURCES structure contains substructures, SOURCES.HEAD_TEMPLATE (3d MRI head rendering matrices) and SOURCES.HEADSHAPE (head shape and index points). The SOURCES structure is stored on disk in the subject directory tree in the file named source_template.mat. However, the file vol95023.mat contains additional information concerning the associated MRI slices and file locations. Therefore the file vol#####.mat is used for loading the corresponding SOURCES structure into the MEG Tools work space.

Forward Model Structure

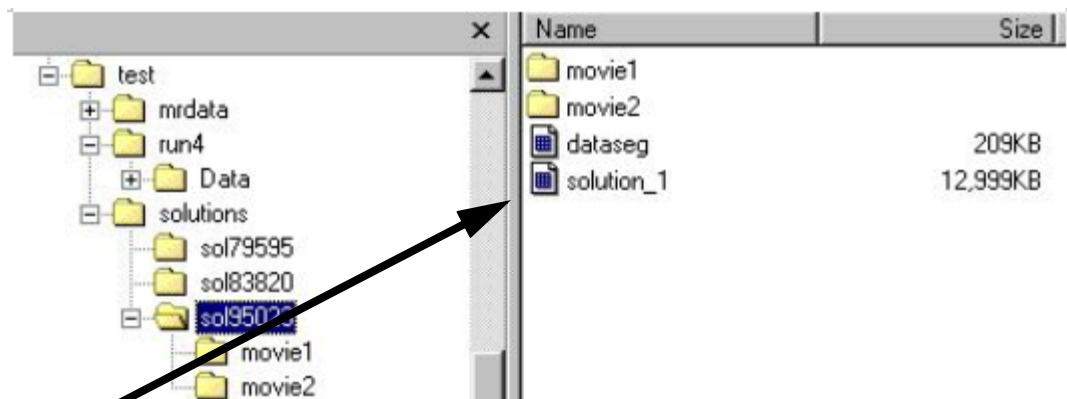


FORWARD_MODEL =

DATA_PARENT: 'c,rfhp0.1Hz,f3-100n,o-DATA'
SOURCES_PARENT: 'D:\image_data\test\mrdata\vol95023\source_template'
ID_TAG: '83820681379832503709'
SRC_ID_TAG: '95023160748689176245'
XLEAD: [146x2919 double] (X oriented dipoles for sources at SOURCES.LOCATION)
YLEAD: [146x2919 double] (Y oriented relative to subject coordinates)
ZLEAD: [146x2919 double] (Z oriented relative to subject coordinates)
ORIGIN: [3x6 double]
ID_MEG: [148x1 double]
MEG_LOC: [148x3 double]
MEG_DIR: [148x3 double]
EPOCH_PARENT: 'D:\image_data\test\run4\DATA\
DIRECTORY: 'D:\image_data\test\run4\DATA\fwd_models\
NAME: 'forward_model_1'

The FORWARD_MODEL data structure contains the X,Y, Z gain matrices, (XLEAD, YLEAD, ZLEAD). Also, FORWARD_MODEL.MEG_LOC and FORWARD_MODEL.MEG_DIR are compared to these fields in the EPOCH data structure before generating a SOLUTION. If they do not match exactly then an Error Message will be generated rather than a SOLUTION. Therefore, this FORWARD_MODEL structure is compatible with the original data or any EPOCH subset or EPOCH parent. In addition, the SRC_ID_TAG field must match the corresponding SOURCES.ID_TAG field or an Error Message is generated rather than a SOLUTION.

Solution Structure



SOLUTION =

```
TYPE: 'SCD'
INIT_TYPE: [ ]
INIT_ID_TAG: [ ]
INIT_NAME: [ ]
INIT_DIRECTORY: [ ]
CHANNELS_USED: [66x1 double]
NUM_TIME_PTS: 139
SOURCES: [1x1 struct]
FWD_PARENT: [1x66 char]
ID_TAG: '95023160748689176245'
DATA_PARENT: 'D:\image_data\test\run4\DATA\p07-Sep-2000\dataseg1.mat'
NAME: 'solution_1'
DIRECTORY: 'D:\image_data\test\solutions\sol95023\'
X_AMP: [2919x139 double]      (For X oriented dipoles located at SOURCES.LOCATION)
Y_AMP: [2919x139 double]      (For Y oriented sources relative to subject coordinates)
Z_AMP: [2919x139 double]      (For Z oriented sources relative to subject coordinates)
ERROR: [2919x139 double]
DATA_TIME_PTS: [1x139 double]
DIST_ERROR: [1x139 double]
XYZmatrix: [51x139 double]
ERRORmtx: [51x139 double]
```

The SOLUTION structure contains the X, Y, Z amplitudes of the solution for each latency and information about its construction. (Note: The source locations are in MRI coordinates in SOLUTION.SOURCES.LOCATION.) Also the directory that contains the SOLUTION structure also contains the EPOCH data used to construct it and all movies, (in separate movie directories). The sequence of movie#.bmp files can be imported to other image programs for viewing.

Technical References

- 1 Moran JE, Tepley N, A field gradient transformation of large array magnetoencephalographic data for cortical source imaging. In Recent Advances in Biomagnetism, T Yoshimoto, M Kotani, S Kuriki, H Karibe and N Nakasato, eds. Tohoku University Press, Sendai Japan, 1999, pp 266-269.
- 2 Moran JE, Tepley N, Two dimensional inverse imaging, (2D11), applied to large array magnetoencephalographic data. In Recent Advances in Biomagnetism, T Yoshimoto, M Kotani, S Kuriki, H Karibe and N Nakasato, eds. Tohoku University Press, Sendai Japan, 1999, pp 270-273.
- 3 Moran JE, Tepley N: Two dimensional inverse imaging of current sources in magnetoencephalography. Brain Topography, 12 no. 3 , 2000, pp 201-217.
- 4 Moran JE, Tepley N, Source space minimization technique for MEG source analysis In biomag2000 proceedings: available for download at <http://biomag2000.hut.fi/papers/178b.pdf>
- 5 Press WH, *Numerical Recipes in C*, 2ed., pp 413-416,(Cambridge University Press New York, 1996).
- 6 Gorodnitsky I.F., George J.S., Rao B.D., Neuromagnetic source imaging with FOCUSS: a recursive weighted minimum norm algorithm, Elctroencephal. Clin. Neurophysiol. 95 pp231-251(1995)
- 7 Liu A.K., Belliveau J.W., Dale A.M., Spatiotemporal imaging of human brain activity using functional MRI constrained magnetoencephalography data: Monte Carlo simulations Proc. Natl. Acad. Sci. USA 95 pp/ 8945-8950,1998

Abstract: Source Space Minimization Technique for MEG Source Analysis

J.E. Moran¹, and N. Tepley^{1,2}.

¹Henry Ford Hospital, Detroit, Michigan USA; ²Oakland University, Rochester Michigan USA
(tepley@oakland.edu)

Introduction

The magnetic inverse solution of $\mathbf{b} = \mathbf{G}\mathbf{q}$, can be transformed into a minimization of $[\mathbf{b} - \mathbf{G}\mathbf{q}]^2$ and solved by refining an initial solution estimate \mathbf{q}_0 , using a gradient descent method, $\mathbf{q}_{i+1} = \mathbf{q}_i + \alpha_i[\mathbf{G}^T\mathbf{G}]^{-1}\mathbf{G}^T\mathbf{r}_i$, $\mathbf{r}_{i+1} = \mathbf{r}_i - \mathbf{G}(\mathbf{q}_{i+1} - \mathbf{q}_i)$. However, the contribution of components of \mathbf{b} corresponding to low amplitude eigenvalues in spectrum of the matrix $[\mathbf{G}^T\mathbf{G}]$ are excessively amplified. This is undesirable, especially if \mathbf{b} contains noise. The problem can be avoided if the minimization of $[\mathbf{b} - \mathbf{G}\mathbf{q}]^2$ is transformed into the source space minimization of $[\mathbf{G}^T\mathbf{b} - \mathbf{G}^T\mathbf{G}\mathbf{q}]^2$ with the solution: $\mathbf{q}_{i+1} = \mathbf{q}_i + \alpha_i[\mathbf{G}^T\mathbf{G}]\mathbf{G}^T\mathbf{r}_i$, $\mathbf{r}_{i+1} = \mathbf{r}_i - \mathbf{G}(\mathbf{q}_{i+1} - \mathbf{q}_i)$ and $\alpha_i = \mathbf{r}_i^T[\mathbf{G}\mathbf{G}^T]^2\mathbf{r}_i / \mathbf{r}_i^T[\mathbf{G}\mathbf{G}^T]^4\mathbf{r}_i$. This recursive solution amplifies the importance of large eigenvalue components of the magnetic field rather than the small ones. Thus, it is useful for calculating equivalent current dipole amplitudes when the forward solution incorporates a realistic volume conductor model and radial sources generate detectable magnetic fields (Symbols as defined in [3], above)).

Methods The performance of this minimization technique was tested by calculating x,y, and z source amplitudes for a small grid of 16 source locations. Simulated data consisted of the lead field for one of these sources oriented in either the x,y, or z direction. In addition, we developed imaging software incorporating this technique to generate equivalent current dipole (ECD) and 2DII images [3] of somatosensory and visual evoked cortical activity.

Results Calculated source amplitudes exclude radial source components and accurately reflected the amplitude of tangential sources. The calculation of ECD amplitudes required only 2 iterative steps. Also, the same performance is demonstrated in the whole brain imaging applications incorporating an MRI derived model of the cortex.

Discussion This is a conjugate direction minimization technique [5]; the sequence of residual magnetic field vector changes are orthogonal, $\Delta\mathbf{r}_i\Delta\mathbf{r}_j = 0$. Also, the algorithm produces \mathbf{q} equal to a series

$$\sum \beta_i [\mathbf{G}^T\mathbf{G}]^{2i-2} \mathbf{G}^T\mathbf{b}, i \geq 2.$$

Appendix: Matlab Path

After installing the MEG_tools directory, the following directories must be added to the Matlab Path before running the MEG tools program:

```
C:\matlabR12\MEG_tools\WHSplot
C:\matlabR12\MEG_tools\imaging\soln_viewer
C:\matlabR12\MEG_tools\imaging
C:\matlabR12\MEG_tools\MR_image
C:\matlabR12\MEG_tools\imaging\twodii
C:\matlabR12\MEG_tools\imaging\soln_viewer\view_flat_pannel
C:\matlabR12\MEG_tools\imaging\soln_viewer\view_3d
C:\matlabR12\MEG_tools\imaging\soln_viewer\view_3d\movie
C:\matlabR12\MEG_tools\imaging\scd
C:\matlabR12\MEG_tools\imaging\channel_choice
C:\matlabR12\MEG_tools\WHSplot\svd_spectral_tools
C:\matlabR12\MEG_tools\WHSplot\sen_plot
C:\matlabR12\MEG_tools\WHSplot\overlay_plot
C:\matlabR12\MEG_tools\WHSplot\meg_vert_plots
C:\matlabR12\MEG_tools\WHSplot\graphs_common
C:\matlabR12\MEG_tools\WHSplot\filters
C:\matlabR12\MEG_tools\WHSplot\eeg_plots
C:\matlabR12\MEG_tools\WHSplot\database
C:\matlabR12\MEG_tools\WHSplot\data_control
C:\matlabR12\MEG_tools\WHSplot\contour_plot
C:\matlabR12\MEG_tools\MR_image\tools
C:\matlabR12\MEG_tools\MR_image\coregister
C:\matlabR12\MEG_tools\MR_image\ASA
C:\matlabR12\MEG_tools\Forward_model\matlab_dll
C:\matlabR12\MEG_tools\Forward_model
```

APPENDIX : GENERIC DATA FILE FORMATS

GEN_SEN_LOC.TXT

%*****GEN_SEN_LOC.TXT*****

This file contains the location and orientation of MEG and EEG channels
in the Subject coordinates.

ALL LOCATIONS must be in meters.

%*****

Top row must contain the text "Subject:" followed by any text that does not include "#Channel" or any channel identifier such as "A1"

EXAMPLE:

Subject: JOHN DOE

Second row of "gen_sen_loc.txt" is the Column Identifier row.

This row of labels is OPTIONAL. However, if this row of column identifies is included it must be exactly as shown below

#Channel	Label	Type	Pos(x)	Pos(y)	Pos(z)	Dir(x)	Dir(y)	Dir(z)	Weight
----------	-------	------	--------	--------	--------	--------	--------	--------	--------

Channel identifiers -

MEG CHANNELS = A# , Such as:

A50, A33, etc. must be used for MEG signal channels. This includes gradiometers as well as magnetometers.
Reference MEG channels should not use A#.

(In addition, MEG channels all have label, type, location, orientation, and weight field entries.)

EEG CHANNELS = E# ♦ E14, E9, etc. must be used for EEG signal channels (EEG channels have label, type, location and label field entries. No direction entries.)

OTHER CHANNELS = G,M,Q,B,L,T,R, and other letters of the alphabet. Such as:

G1, G2, etc for reference gradiometers.

T, R, for TRIGGER and RESPONSE channels.

Use Type 00 when these channels do not have position, orientation and weight fields.

(IMPORTANT !!!!! These channel letter/number combinations must be the same as those used in the "gen_header.txt" file that contains the labels of the acquired channels in the order in which they are stored in the time slice data file.

"gen_sen_loc.txt" may contain additional channel/numbers that are not included in the imported data header file.

However, it must contain all the channel identifiers found in the "gen_header.txt" file.

Label = any single word text or text number

Type

ma = magnetometer

gp = planar gradiometer

(graphical displays will use the average location of all rows with same channel)

ga = axial gradiometer, use with all first order, second order, etc.

(Graphical displays will use the location of first instance of channel identifier
in the input file shown below.)

ep = electric potential

00 = (zero, zero) = no position, orientation, or weight for this entry.

EXAMPLES

Typical 4D Neuroimaging Magnes 2500 sensor positions in Generic file format

The channel order only affects location of graphical display of repeated channel numbers (gradiometers,etc)

Subject: John Doe mrn: 123456-987

#Channel	label	Type	Pos(x)	Pos(y)	Pos(z)	Dir(x)	Dir(y)	Dir(z)	Weight
T1	TRIGGER	00							
R1	RESPONSE	00							
A68	M68	ma	0.0944	-0.0702	0.1021	0.5695	0.7829	0.2501	1.00
G1	GzyA	gp	0.1247	0.0951	0.3085	0.7004	0.1269	0.7023	1.00
G1	GzyA	gp	0.1397	-0.0386	0.3176	-0.7004	-0.1269	-0.7023	-1.00
M1	MxA	ma	0.1322	0.0282	0.3130	0.7049	0.0310	-0.7085	1.00
A75	M75	ma	0.0654	0.0975	0.0735	0.3046	0.9507	-0.0556	1.00
A1	M1	ma	0.0506	0.0097	0.1779	0.4521	0.1097	0.8851	1.00
E5	T7	ep	0.0150		0.0716		0.028		
E3	P9	ep	-0.0240		0.0679		0.0303		
E1	F9	ep	0.0568		0.0664		0.0117		

***** Generic file format entry for a first order gradiometer channel *****

A68	M68	ga	0.0945	-0.0703	0.1022	0.5695	-0.7830	0.2502	1.0000
A68	M68	ga	0.1173	-0.1016	0.1122	0.5695	-0.7830	0.2502	-1.000

%***** Generic file format entry for a 2nd order gradiometer channel *****

A68	M68	ga	0.0945	-0.0703	0.1022	0.5695	-0.7830	0.2502	1.0000
A68	M68	ga	0.1173	-0.1016	0.1122	0.5695	-0.7830	0.2502	-2.000
A68	M68	ga	0.1401	-0.1329	0.1222	0.5695	-0.7830	0.2502	1.0000

%***** Generic file format entries for magnetometer *****
 %***** with the Field calculated as weighted average of 7 positions within sensor coil*****

A1	M1	ma	-0.1210	0.000	0.1600	0.5695	-0.7830	0.2502	0.3770
A1	M1	ma	-0.1210	0.0592	0.1600	0.5695	-0.7830	0.2502	0.1885
A1	M1	ma	-0.1723	0.0296	0.1600	0.5695	-0.7830	0.2502	0.1885
A1	M1	ma	-0.1723	-0.0296	0.1600	0.5695	-0.7830	0.2502	0.1885
A1	M1	ma	-0.1210	-0.0592	0.1600	0.5695	-0.7830	0.2502	0.1885
A1	M1	ma	-0.0697	-0.0296	0.1600	0.5695	-0.7830	0.2502	0.1885
A1	M1	ma	-0.0697	0.0296	0.1600	0.5695	-0.7830	0.2502	0.1885

%***** Typical generic format entries for electrode positions *****
 %***** The electrode number can be a number as the label as in the last row *****

E1	F9	ep	0.0568	0.0664	0.0117
E2	T9	ep	0.0186	0.0667	-0.0075
E3	P9	ep	-0.0240	0.0678	0.0303
E4	F7	ep	0.0738	0.0609	0.0253
E5	T7	ep	0.0150	0.0716	0.0286
E6	P7	ep	-0.0322	0.0650	0.0574
E7	FC5	ep	0.0510	0.0700	0.0545
E8	CP5	ep	-0.0019	0.0730	0.0706
E9	E9	ep	0.1099	0.0301	0.0254

(Note: dimensions are in METERS)

Generic hs_file.txt

Subject: 7012

Index Points:

-0.006296	0.079491	0.000000
0.006296	-0.079491	-0.000000
0.098213	0.000000	0.000000

Digitization Points:

0.102471	-0.026683	0.039748
0.098768	-0.036678	0.046249
0.097131	-0.038696	0.047181
0.095118	-0.041440	0.048379
0.092618	-0.043837	0.049345
0.090017	-0.046152	0.050131

In the above file, the index point must be :

First line LPA (left preauricular)
 Second line RPA (right preauricular)
 Third line Nasion

Additional lines as desired. (Note: dimensions are in METERS)

Also the hs_file must contain digitization points. If none were obtained, then reuse the Index points as digitization points. The first three lines of index_points are used to define the subject coordinate system.

Total Channels: 160
Number of Epochs: 1
Sample Period: 3.440683e-03
First Latency: -1.000000e-01

Orientation matrix for MRI data

The last line has the x,y,z oriented mm/pixel correspondence. If zeros are found it defaults to 1 mm/pixel. The scaling is easily adjusted during the coregistration of the MEG coordinates to the MRI coordinates.

<drive>:image data\subject directory\Run#

In subject directory

In Run directory

(SEE DEMO DATA for example of these files)

Acknowledgement: Research supported by NIH/NINDS Grant RO1-NS30914