Journal of Neuroscience Methods 198 (2011) 359–363

Contents lists available at ScienceDirect

Journal of Neuroscience Methods

journal homepage: www.elsevier.com/locate/jneumeth

Short communication

# Finite difference neuroelectric modeling software

Hung V. Dang[a,b], Kwong T. Ng[a,*]

[a] Klipsch School of Electrical and Computer Engineering, New Mexico State University, MSC 3-O, Las Cruces, NM 88003, United States
[b] Department of Computer Science, New Mexico State University, United States

ARTICLE INFO

*Article history:*
Received 20 October 2010
Received in revised form 28 March 2011
Accepted 29 March 2011

*Keywords:*
Finite difference method
FDM
Forward solver
Inverse problem
Source localization
Beamformer
LCMV

ABSTRACT

This paper describes a finite difference neuroelectric modeling software (FNS), written in C and MATLAB, which can be executed as a standalone program or integrated with other packages for electroencephalography (EEG) analysis. The package from the Oxford Center for Functional MRI of the Brain (FMRIB), FMRIB Software Library (FSL), is used to segment the anatomical magnetic resonance (MR) image for realistic head modeling. The EEG electrode array is fitted to the realistic head model using the Bioelectromagnetism MATLAB toolbox. The finite difference formulation for a general inhomogeneous anisotropic body is used to obtain the system matrix equation, which is then solved using the conjugate gradient algorithm. The reciprocity theorem is utilized to limit the number of required forward solutions to $N-1$, where $N$ is the number of electrodes. Results show that the forward solver only requires 500 MB of random-access memory (RAM) for a realistic $256 \times 256 \times 256$ head model and that the software can be conveniently combined with inverse algorithms such as beamformers and MUSIC. The software is freely available under the GNU Public License.



## 1. Introduction

Neuroelectromagnetic source reconstruction requires the repeated solution of the forward problem, in which the fields on or near to the scalp are obtained given a source distribution in the brain. To achieve accurate source imaging, a realistic head model should be used and the forward solution has to be obtained numerically (Baillet et al., 2001; Michel et al., 2004). Two common numerical techniques used by researchers are the boundary element method (BEM) and finite element method (FEM) (Wolters et al., 2002; Darvas et al., 2006; Acar and Makeig, 2010).

Recently, researchers have also used the finite difference method (FDM) to study different aspects of source localization (Lemieux et al., 1996; Vanrumste et al., 2001; Hallez et al., 2005; Grech et al., 2008). The FDM is attractive because of its low computational complexity and the easy incorporation of the head structure from a high-resolution anatomical magnetic resonance (MR) image. Once the image is segmented and the tissue is classified for each image voxel, mesh generation can be accomplished simply by mapping each rectangular element in the mesh to an image voxel or a group of image voxels. Each element is then assigned a conductivity value according to its tissue type. Though packages for BEM and FEM head modeling are freely available (Wolters et al., 2002; Acar and Makeig, 2010), to our knowledge, such software does not exist in the public domain for FDM. Here, we introduce the finite difference neuroelectric modeling software (FNS) that interfaces a finite difference solver with the FMRIB Software Library (FSL (http://www.fmrib.ox.ac.uk/fsl/)) (Smith et al., 2004; Woolrich et al., 2009) for MR image segmentation and the Bioelectromagnetism MATLAB Toolbox (http://eeg.sourceforge.net/bioelectromagnetism.html) for co-registering the electroencephalography (EEG) electrode system to the 3D head model. It provides a comprehensive capability for subject-specific neuroelectric simulation. Further, the use of reciprocity is included so the number of forward solutions required for the inverse source localization is limited to $(N-1)$, where $N$ is the number of EEG electrodes.

## 2. Software functionality

FNS is written in C and MATLAB, and supports an interface to high performance computing libraries such as the Intel Math Kernel Library and GotoBLAS. Currently, the software runs on platforms with the Linux operating system. It implements all the functions required for realistic head modeling: MR image segmentation; electrode co-registration; forward solver; extraction of potential data for reciprocity application; lead field matrix computation; visualization. The above steps can be performed automatically using a MATLAB or bash automation script. Fig. 1 shows the structure of the software and automation script. Details of major functional blocks in the figure will be described in the following subsections. All

* Corresponding author. Tel.: +1 575646 6431; fax: +1 575646 1435.
*E-mail addresses:* hung.dang@mathworks.com (H.V. Dang), ngnsr@nmsu.edu (K.T. Ng).

0165-0270/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.jneumeth.2011.03.026

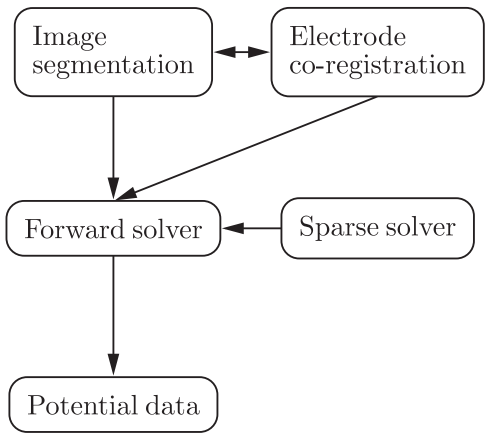


**Fig. 1.** The structure of the software and automation script.

core routines, including brain mask operators, forward solvers and model utility tools, are implemented with C code. The MATLAB code is used to read/write electrode information, to co-register electrodes to the scalp, and to combine all the steps using an automation script.

### 2.1. MR Image segmentation

To construct the head model for the forward solver, the brain extraction tool (BET) in FSL is first used to obtain the masks for the skull and scalp tissue and to extract the brain from the T1-weighted MR image (Smith, 2002; Jenkinson et al., 2005). Then FSL's automated segmentation tool (FAST) is used to identify the white matter, gray matter and cerebrospinal fluid from the image, and to form a separate mask for each tissue (Woolrich et al., 2009; Smith et al., 2004). Once the masks for different tissues have been collected, they can be combined to obtain the final segmented head. FSL supports various MR image formats. However, FNS currently only uses the ANALYZE format to store the segmented head model, which is also the only format accepted by the forward solver.

### 2.2. Electrode co-registration

After the segmented head model is obtained, the scalp nodes are extracted using the img_get_scalp command. Then locations of these nodes and positions of the electrodes used in the EEG experiment are read into MATLAB workspace. Routines from the Bioelectromagnetism MATLAB toolbox are used to co-register the electrodes with the scalp surface using three landmarks or fiducial points, nasion, right and left preauricular points (NAS, RPA and LPA). These steps are performed automatically given the segmented head model, electrode locations, and the three landmark points (in MR image coordinate format). A head model with 128 electrodes (distributed as in the Biosemi system), constructed from a T1-weighted MR image, is shown in Fig. 2(a).

### 2.3. Forward solver

Three types of input are required for the forward solver: (1) a volume image storing the segmented head model in ANALYZE format; (2) electrode locations, which are read into MATLAB workspace using Bioelectromagnetism routines; and (3) tissue conductivity table. The first two kinds of input are described in previous subsections, and the conductivity table is constructed using the values from Haueisen et al. (2002), Ramon et al. (2004, 2006), and Huiskamp (2008). The conductivity table is stored in a comma-separated values (CSV) file ('contable.csv') that allows users to view

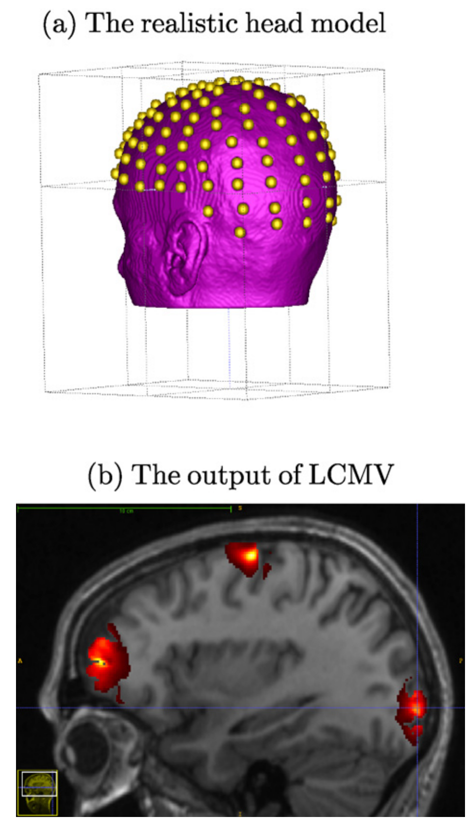


**Fig. 2.** Sample results: (a) the realistic head model and (b) the output of LCMV.

or modify with any text editor. The above input data allow FNS to specify the conductivity of each rectangular cell and the electrode locations in the finite difference grid. More instructions on modifying the conductivity table are given in the web site distributing the software.The forward sparse matrix is formed internally using the finite difference formula for a generally inhomogeneous and anisotropic medium (Saleheen and Ng, 1997). This sparse matrix is reordered and compressed in the software to reduce the matrix size. Forward solutions can be computed using either a conjugate gradient (CG) routine (Hestenes and Stiefel, 1952; Golub and Van Loan, 1996) included in FNS or the Intel Reverse Communication Interface (RCI) CG solver in the Intel Math Kernel Library. (http://software.intel.com/en-us/intel- 125mkl) The users can select the iterative solver during the compilation of FNS.

### 2.4. Extraction of potential data for reciprocity

Reciprocity has been used previously by researchers to limit the number of forward solutions required (Malmivuo and Plonsey, 1995; Hallez et al., 2005) in the inverse source reconstruction. For a given dipole source, to obtain the potential difference $\phi_{12}$ between any two surface electrodes 1 and 2, a current $I_{21}$ which is injected at electrode 1 and removed at electrode 2 is used. The reciprocity theorem states that $\phi_{12} = p[\phi^I(\boldsymbol{r}_+) - \phi^I(\boldsymbol{r}_-)]/I_o l$, where $p$ is the dipole moment, $\phi^I$ is the potential due to the injected currents, $\boldsymbol{r}_+$ and $\boldsymbol{r}_-$ represent the source and sink locations of the dipole, $l$ is the separation distance between them, and $I_o$ is the current magnitude. With one electrode specified as the reference electrode, then there are $N-1$ electrode pairs with independent potential differences, where $N$ is the total number of EEG electrodes.

Once the potentials at predetermined finite difference nodes are computed for all electrode pairs, they can be stored and used to easily calculate the surface potentials and lead field matrix for a