NFT & NIST
Neuroelectromagnetic Forward and Inverse Head Modeling Toolbox

Zeynep AKALIN ACAR
22nd EEGLAB Workshop, San Diego
November, 2016
A complete framework for accurate forward problem solution.

Easy-to-use MATLAB environment with GUI and command-line functions.

Ability to use available subject information
  – T1-weighted 3D MR images
  – Digitized sensor (electrode) locations
Comparison with Dipfit

- The realistic model in Dipfit is a three-layer MNI head model represented with 3000 vertices.
  - The forward matrices are pre-calculated, so there is no need for FP calculations.

- NFT generates subject-specific models.
  - NFT does model generation and forward problem calculations.
  - More accurate.
Head modeling from MR images

From a magnetic resonance image:
- Image Segmentation
- Mesh Generation
- Source Space Generation
- Electrode Co-Registration

MR image → T1-weighted → Segmentation → BEM mesh

Electrode Registration
Preparing the MR Image

- Using FreeSurfer
  - Inhomogeneity correction
  - Convert to 1x1x1 volume
  - Arrange direction of the image
  - Save in analyze format
Image Segmentation

Classifies four tissues from T1-weighted images
Scalp, Skull, CSF and Brain
Starting NFT

- To start from EEGLAB
  
  EEGLAB -> Tools -> NFT

- To start as a standalone toolbox

  addpath NFT directory

  Type ‘NFT’ in Matlab

For demo: go to NFT-2.4_demo folder
Subject Selection

- Select subject folder
- Specify subject name
- Specify session name
Subject Selection

Select current folder as subject folder
Enter “jc” as subject name
Enter “s1” as the session name
Image Segmentation

NFT: MR segmentation

1. Anisotropic Filtering
   - Number of iterations: 5
   - Image diffusion: 3

2. Scalp Segmentation

3. Brain Segmentation
   - Cerebellar low point: x = 135, y = 135, z = 110
   - White matter seed point: x = 135, y = 135, z = 110

4. Outer Skull Segmentation
   - Center of one eye: z = 110

5. Inner Skull Segmentation

Save Results
- Output Folder: /data/projects/zevnesi/comm/hone/zevnesi/io/oneneme
Image Segmentation

1. Anisotropic Filtering
   - Number of iterations: 5
   - Image diffusion: 3

2. Scalp Segmentation

3. Brain Segmentation
   - Cerebellar low point: x = 135, y = 135, z = 110
   - White matter seed point
   - Fill level: 0.4
   - Threshold: 0.4

4. Outer Skull Segmentation
   - Center of one eye: z = 110

5. Inner Skull Segmentation

Load image

File

Open...

Close

Display Image
- MR image
- Filtered image
- Scalp mask
- Brain mask
- Outer skull mask
- Inner skull mask

Save Results
- Output Folder: /data/projects/zevnap/common/home/zevnap10/deneme

NFT: MR segmentation
Segmentation

Load image test2c.img
Segmentation

Run filtering
Segmentation
Segmentation

View filtered image
Segmentation
Click ‘Next’ for scalp segmentation and run scalp segmentation
Segmentation

View scalp mask
Segmentation
Click ‘Next’ for brain segmentation
Selection of cerebellar low point
Segmentation

Selection of a white matter point
Segmentation

View brain mask
Segmentation

Click ‘Next’ for skull segmentation

Image Segmentation

1. Anisotropic Filtering
   - Number of iterations
   - Image diffusion

2. Scalp Segmentation

3. Brain Segmentation
   - Cerebellar low point
   - White matter seed point

4. Outer Skull Segmentation
   - Center of one eye

5. Inner Skull Segmentation

Save Results

Output Folder: /data/projects/zevnew/common/home/zevnew/to/deneme

Brain segmented!
Segmentation

Select a slice for eyes and click ‘set’
Segmentation

Click ‘Run’ for skull segmentation
Segmentation

Click on the eyes

Image Segmentation

1. Anisotropic Filtering
   - Number of iterations: 5
   - Image diffusion: 3

2. Scalp Segmentation

3. Brain Segmentation
   - Cerebellar low point: 67
   - White matter seed point: 172 (x), 188 (y), 150 (z)
   - Fill level: 0.1
   - Threshold: 0.1

4. Outer Skull Segmentation
   - Center of one eye: 95

5. Inner Skull Segmentation

Save Results

Output Folder: /data/projects/zeigen/commong/home/zeigen/01/deneme
Filtered Image
Segmentation
Segmentation

Image Segmentation

1. Anisotropic Filtering
   - Number of Iterations: 5
   - Image Diffusion: 3

2. Scalp Segmentation

3. Brain Segmentation
   - Cerebellar low point: x = 172, y = 188, z = 150
   - White matter seed point: x = 7, y = 178, z = 150
   - Fill level: 0.1
   - Threshold: 0.1

4. Outer Skull Segmentation
   - Center of one eye: x = 95

5. Inner Skull Segmentation
   - Output Folder: /data/projects/zeypex/comm
   - Segmentation: Filtered Image
Segmentation

View skull segmentation
Segmentation

Click ‘Next’ for CSF segmentation
Segmentation

Click ‘Run’ for CSF segmentation
Segmentation
Segmentation

View CSF segmentation

1. Anisotropic Filtering
   - Number of iterations: 5
   - Image diffusion: 3

2. Scalp Segmentation

3. Brain Segmentation
   - Cerebellar low point: 67
   - White matter seed point: 172, 158
   - Fill level: 0.1
   - Threshold: 0.1

4. Outer Skull Segmentation
   - Center of one eye: 95

5. Inner Skull Segmentation

Save Results
- Output Folder: /data/projects/zeynep/common/home/zeynep/to/deneme
- Filtered Image
- Segmentation

Segmentation completed
Segmentation

NFT: MR segmentation

(x,y,z) = (150, 160, 139)

1. Anisotropic Filtering
   - Number of iterations: 5
   - Image diffusion: 3

2. Scalp Segmentation

3. Brain Segmentation
   - Cerebellar low point: (67, Set)
   - White matter seed point: (172, Set)

4. Outer Skull Segmentation
   - Center of one eye: (95, Set)

5. Inner Skull Segmentation

Save Results:
- Output Folder: /data/projects/yezcan/comm/homedenene
- Filtered Image
- Segmentation

Saving filtered image as SubjectA_filtered.mat
Segmentation

NFT: MR segmentation

(x,y,z) = (150, 160, 139)

Image Segmentation
- Swap L/R
- Check inhomogeneity

1. Anisotropic Filtering
   - Number of iterations: 5
   - Image diffusion: 3

2. Scalp Segmentation

3. Brain Segmentation
   - Cerebellar low point: x = 67
   - White matter seed point: y = 172, z = 158

4. Outer Skull Segmentation
   - Center of one eye: z = 95

5. Inner Skull Segmentation

Display Image
- MR image
- Filtered image
- Scalp mask
- Brain mask
- Outer skull mask
- Inner skull mask

Save Results
- Output Folder: /data/projects/zevnev/comm
- Filtered Image
- Segmentation

Filtered image saved as SubjectA_filtered.mat
Segmentation

NFT: MR segmentation

1. Anisotropic Filtering
   - Number of iterations: 5
   - Image diffusion: 3

2. Scalp Segmentation

3. Brain Segmentation
   - Cerebellar low point: (172, 188, 150)
   - White matter seed point: (50, 150)
   - Fill level: 0.1
   - Threshold: 0.1

4. Outer Skull Segmentation
   - Center of one eye: (95, 150)

5. Inner Skull Segmentation

- Output Folder: /data/projects/zeynep/common/home/zeynep/10/deneme
- Run
- Next

Saving segmentation as: SubjectA_segments.mat
Segmentation

Image Segmentation

1. Anisotropic Filtering
   - Number of iterations: 5
   - Image diffusion: 3

2. Scalp Segmentation

3. Brain Segmentation
   - Cerebellar low point: 67
   - White matter seed point: 172
   - Fill level: 0.1
   - Threshold: 0.1

4. Outer Skull Segmentation
   - Center of one eye: 95

5. Inner Skull Segmentation

Save Results:
- Output Folder: /data/projects/zeynep/common/home/zeynep/lo/deneme
- Segmentation

Segmentation saved as SubjectA_segments.mat
Image Segmentation

```
>> dir SubjectA*
SubjectA_mri.mat    SubjectASegments.mat

>> load SubjectA_mri
>> mri

mri =

    dim: [256 256 256]
    xgrid: [1x256 double]
    ygrid: [1x256 double]
    zgrid: [1x256 double]
    anatomy: [256x256x256 double]
    transform: [4x4 double]
    hdr: []

>> load SubjectA_segments
>> Segm

Segm =

    scalpmask: [256x256x256 logical]
    brainmask: [256x256x256 logical]
    outerskullmask: [256x256x256 logical]
    innerskullmask: [256x256x256 logical]
```
Mesh Generation

Generate Mesh for a 3 or 4 layer head model
Mesh generation

Click local mesh refinement
Mesh generation

NFT: Mesh generation (on juggling-0-6.local)

Load Segmentation: /data/cta/zeypol/NFTdene/NFTplugin_demo_dipole/jo_segments

Output Folder: /data/cta/zeypol/NFTdene/NFTplugin_demo_dipole

Mesh name: jc

# of layers: 4

Mesh name: jc

Number of nodes per layer: 7000

Local mesh refinement

Edge length/Distance between meshes: 2.1

Start Mesh Generation

Mesh saved!

Generate linear FEM mesh

Generate quadratic FEM mesh
Mesh generation

Load Segmentation: /data/cta/zeynep/NFTdene/NFTplugin_demo_dipole/jo_segments

Output Folder: /data/cta/zeynep/NFTdene/NFTplugin_demo_dipole

# of layers: 4

Mesh name: jo

Number of nodes per layer: 7000

Local mesh refinement: on (2.1)

Edge length/Distance between meshes: 2.1

Start Mesh Generation

Mesh saved!

Generate linear FEM mesh

Generate quadratic FEM mesh
- \([C, E] = \text{ReadSMF('Scalp.smf',0,0,0,1)};\)
- \(\text{Plotmesh(E(:,2:4),C(:,2:4))}\)
Mesh generation
Source Space Generation

Generates a simple source space:
Regular Grid inside the brain
With a given spacing and distance to the mesh
Source Space Generation

From a magnetic Resonance Image

- Image Segmentation
- Mesh Generation
- Source Space Generation

Electrode Co-Registration

NFT: Source space generation (on juggling-0-6.local)

Mesh Folder: /data/cta/zeynep/NFTdene/NFTplugin_demo_dipole

Grid spacing (mm): 3
Min. distance from the mesh (mm): 2

Generate Regular Source Space
Generate Symmetric Source Space

Source space saved!
Electrode Co-registration

From a magnetic Resonance Image

- Image Segmentation
- Mesh Generation
- Source Space Generation
- Electrode Co-Registration

NFT: Electrode co-registration (on juggling-0-6.local)

- Load sensor locations
- Electrode file name: /data/cta/zeynap/NFTdene/NFTplugin_demo_dipole
- Mesh Folder
- Initial co-registration
- Translation
- Rotation
- Complete co-registration
- Translation
- Rotation
- Save initial reg.
- Save complete reg.
Electrode Co-registration

From a magnetic Resonance Image

Image Segmentation

Mesh Generation

Source Space Generation

Electrode Co-Registration
Electrode Co-registration
Electrode Co-registration
Electrode Co-registration

Load sensor locations

Mesh Folder

Initial co-registration

Translation: 3
Rotation: 0 0 0

Complete co-registration

Translation
Rotation

Save initial reg.

Save complete reg.
Electrode co-registration

NFT: Electrode co-registration (on juggling-0-6.local)

Load sensor locations
/data/cta/zeynep/NFTdene/NFTplugin_demo_dipole/jc_fid.sfp

Mesh Folder
/data/cta/zeynep/NFTdene/NFTplugin_demo_dipole

Initial co-registration
Translation: -3 -10 0
Rotation: 0 0 0

Complete co-registration

Save initial reg.
Save complete reg.

Computing translation and rotation parameters...
Electrode co-registration

Figure: Co-registered electrode locations (on juggling).

Translation and rotation parameters are computed.
Mesh generation

NFT: Electrode co-registration (on juggling-0-6.local)

Load sensor locations /data/cta/zeynep/NFTdene/NFTplugin_demo_dipole/jc_fid.sfp

Mesh Folder /data/cta/zeynep/NFTdene/NFTplugin_demo_dipo2o

Initial co-registration
Translation -3 -10 0
Rotation 0 0 0

Complete co-registration
Translation 10.3303 -1.18985 -11.5649
Rotation 1.6886 -3.3001 0.42156

Save initial reg.  Save complete reg.

Automatic registration is saved.
>> sens = load('jc_s1.sensors', '-mat')

sens =

    fn: '/data/cta/zeynep/NFTdene/JC/jc_fid.sfp'
    eloc: [1x208 struct]
    pnt: [208x3 double]
    ind: [1x208 double]
    param: [1x1 struct]

>> sens.param

ans =

    init: [-3  -10  0  0  0  0  1  1  1]
    auto: [10.3303  -1.1899  -11.5649  1.6886  -3.3001  0.4216]

>> |
Dipole source localization
Select EEG data
Forward Problem Solver

- MATLAB interface to numerical solvers
- Boundary Element Method or Finite Element Method
  - EEG Only (for now)
  - Interfaces to the Matrix generator executable written in C++
- Other computation done in MATLAB
- Generated matrices are stored on disk for future use.
Forward Problem Solution with BEM
Forward Problem Solution with BEM
Forward Problem Solution with BEM
Forward Problem Solution with BEM
Forward Problem Solution with BEM

NFT: Forward problem solution (on juggling-0-6.local)

Forward Model Generation

BEM Mesh Info
- Mesh Name: jo
- Number of Layers: 4
- Number of Nodes: 16151
- Number of Elements: 32286
- Number of Nodes/Element: 3

BEM Model
- Model Name: jo
- Enter conductivity values:
  - Skull: 0.0132
  - Scalp: 0.33
  - Brain: 0.88
  - CSF: 1.79
- Modified (Isolated Problem Approach)
- Create Model
- BEM Model Loaded

Session
- Session Name: s1
- Load Sensors
  - Mesh Coordinates
  - Mesh Node List
- Sensors Loaded: 208
- Generate transfer matrix
- Session Loaded

Forward Problem Solution
- Load Source Space
- Compute Lead Field Matrix
- Plot Potential Distribution
  - For Dipole
Forward Problem Solution with BEM
Forward Problem Solution with BEM

BEM Mesh Info
- Mesh Name: jo
- Number of Layers: 4
- Number of Nodes: 16161
- Number of Elements: 32286
- Number of Nodes/Element: 3

BEM Model
- Model Name: jo
- Conductivity values:
  - Scalp: 0.33
  - Brain: 0.33
  - Skull: 0.0132
  - CSF: 1.79
- Check box: Modified (Isolated Problem Approach)
- Create Model

Session
- Session Name: s1
- Load Sensors:
  - Mesh Coordinates
  - Mesh Node List
- Sensors Loaded: 208

Forward Problem Solution
- Load Source Space: 7479 Dipoles Loaded
- Compute Lead Field Matrix
- Plot Potential Distribution
Forward Problem Solution with BEM

**Forward Model Generation**

- **BEM Mesh Info**
  - Mesh Name: jo
  - Number of Layers: 4
  - Number of Nodes: 16151
  - Number of Elements: 32286
  - Number of Nodes/Element: 3

- **BEM Model**
  - Model Name: jo
  - Conductivity values:
    - Scalp: 0.33
    - Brain: 0.33
    - Skull: 0.0132
    - CSF: 1.79
  - Modified (Isolated Problem Approach)

- **Session**
  - Session Name: s1
  - Sensors Loaded: 208

**Forward Problem Solution**

- **Load Source Space**
  - Dipoles Loaded: 7479

- **Compute Lead Field Matrix**
  - LFM Computed

- **Plot Potential Distribution**
  - For Dipole
Forward Problem Solution with BEM
Inverse Problem Solution with BEM
Dipole source localization is saved in EEG structure, under EEG.etc.nft.

After source localization with NFT, you can continue using EEGLAB;

EEG.dipfit.model = EEG.etc.nft.model;
Distributed Source localization

Go to the folder NFTplugin_demo_cortical
addpath
Distributed Source Localization

Neuroelectromagnetic Forward Head Modeling Toolbox (on juggling-0-5.local)
Select EEG data
NIST – Generation of a cortical source space

- Load Freesurface cortical surface
- Downsample to 80,000 vertices
- Co-register with the NFT brain surface
- Re-generate NFT head model
- Calculate normals for each vertex on the cortical surface
- Save the cortical source space as: Subject_name_FS_ss.dip
- Calculate node area of each vertex for source localization, save as Node_area
- Check if the sensor locations need to be updated according to the new NFT model.
Distributed Source Localization

- Load MRI
- Start Freesurfer
- Cortical source space

MRI file

80000
# of dipoles in source space

10 mm

Generate patches

Forward Problem Solution

FP Solution with BEM
FP Solution with FEM

Cortical Source Localization

Component indices

Select Source Localization Method

Start Source Localization

IC #

Visualization
Load MRI
Run Freesurfer

![Freesurfer Interface](image_url)
NIST – Patch generation

NIST has options to generate Gaussian patches with 10 mm, 6 mm, 3 mm in radius.
Forward Problem Solution with FEM

- Tetgen for mesh generation
  - Uses BEM meshes as boundaries
- METU-FEM to generate transfer matrix
  - Compiled from source
  - Requires PETSc for matrix operations
- metufem .mex file for forward solutions in MATLAB
- Instructions available under README.FEM file.
FEM Mesh Info
- Mesh Name: jcFS.1.msh
- Number of Layers: 4
- Number of Nodes: 297956
- Number of Nodes/Element: 4

FEM Session
- Session Name: s1
- Enter conductivity values:
  - Scalp: 0.33
  - Skull: 0.0132
  - Brain: 0.33
  - CSF: 1.79
- Sensors Loaded: 208
- Create Session
  - FEM Session Created

Forward Problem Solution
- Load Source Space
  - Dipoles Loaded: 80150
- Compute Lead Field Matrix
FEM Mesh Info

Mesh Name: jcFS.1.msh
Number of Layers: 4
Number of Nodes: 297956
Number of Nodes/Element: 4

FEM Session

Session Name: s1
Enter conductivity values:
- Scalp: 0.33
- Skull: 0.0132
- Brain: 0.33
- CSF: 1.79
Load sensors: 208
Create Session

FEM Session Loaded

Forward Problem Solution

Load Source Space
Dipoles Loaded: 80150

Compute Lead Field Matrix
LFM Computed
Output

- Source estimates are saved as cortex_source_scs and/or cortex_source_sbl
Child Head modeling

- Segmentation of infant/child head into scalp, skull, CSF, and brain tissues.
- Electrical head mesh generation using NFT, making possible
- Non-invasive conductivity estimation of major head tissues, and also
- Making available accurate, age-specific developmental template head models
Generation of individual head models

NFT (sccn.ucsd.edu/nft/) was used to generate four-layer Finite Element (FEM) head models.

6 months

1,441,777 tetrahedral volume elements

12 months

1,025,643 tetrahedral volume elements
Richards Database for child head modeling

scalp  skull  CSF
brain  white matter

5-layer template BEM head model for three-year olds.
Source localization results

source

SCS

Patch-based SBL

sLORETA

Cheng Cao, 2012
c/o A. Ojeda
EEG source localization

scalp
skull
CSF
brain
Source localization for 1-year old
Visualization

- NIST uses Showmesh for visualization of potentials on the cortical surface.
- Showmesh loads a mesh in .smf format,
- Loads potential distribution.
- There are options to load a point set, zoom in, out, rotate, take snapshots.

SHOWMESH TUTORIAL
NFT Matlab Scripts

- Start EEGLAB and set your parameters:

```matlab
eeglab
EEG = pop_loadset('filename',eeg_file,'filepath',eeg_path);
[ALLEEG, EEG, CURRENTSET] = eeg_store(ALLEEG, EEG, 0);
% set 'of' (output folder), subject_name, session_name, and elec_file
subject_name = 'SubjectA';
session_name = 's1';
nl = 4; % number of layers
plotting = 1;
comp_index = 1:20; % component index for source localization
```
NFT Matlab Scripts

- Realistic modeling from MRI
  
  % Do segmentation using the GUI
  nft_mesh_generation(subject_name, of, nl)
  nft_source_space_generation(subject_name, of)
  % Do co-registration using the GUI
  nft_forward_problem_solution(subject_name, session_name, of);
  
  dip1 = nft_inverse_problem_solution(subject_name, session_name, of, EEG, comp_index, plotting, elec_file)
NFT Matlab Scripts

- BEM warping mesh

  nft_warping_mesh(subject_name, session_name, elec_file, nl, of, 0, 0);

  nft_forward_problem_solution(subject_name, session_name, of);

  dip1 = nft_inverse_problem_solution(subject_name, session_name, of, EEG, comp_index, plotting, elec_file)
NFT Matlab Scripts

- **FEM warping mesh**
  
  ```matlab
  session_name='s1_fem';
  nft_warping_mesh(subject_name, session_name, 
                   elec_file, nl, of, 0, 1);
  
  nft_fem_forward_problem_solution(subject_name, 
                                    session_name, of);
  
  dip1 = nft_inverse_problem_solution(subject_name, 
                                      session_name, of, EEG, comp_index, plotting, 
                                      elec_file)
  ```
**NFT Matlab Scripts**

- Set NFT dipole structure to EEGLAB dipole structure

```matlab
eeglab_folder = dirname(which('eeglab'));
mri_file = [eeglab_folder /
            'plugins/dipfit2.2/standard_BEM/standard_mri.mat'];
EEG.dipfit.mrifile = mri_file;
EEG.dipfit.model = EEG.etc.nft.model;
```
NFT download and reference

- [http://www.sccn.ucsd.edu/nft](http://www.sccn.ucsd.edu/nft)