The maetsim software for MATLAB

Philip Hoskins

Contents

1 Introduction to MAET **2**

1.1 Mathematical model of MAET 2

2 Software **6**

2.1 How to install 7

2.2 Geometric parameters 8

3 .d files and tools **9**

3.1 readd.m 9

3.2 ndwrite.m 9

3.3 viewd.m 9

4 Direct **10**

5 Inverse **12**

5.1 inverse.in 12

5.2 Curl configuration files 13

6 Examples provided **13**

1 Introduction to MAET

This manual documents some software tools developed for the numerical study of Magnetoacoustoeletric tomography (MAET). Since the basic procedures of data acquisition and image reconstruction in MAET are not currently well known, we will provide here a very brief summary.

MAET is a novel biomedical imaging modality with the goal of reconstructing the spatially varying conductivity of a biological specimen. A hypothetical schematic of a MAET procedure is illustrated in Figure 1. Data acquisition in MAET proceeds as follows. A conductive (non-uniform) object is placed in a tank filled with a saline solution of known conductivity. A uniform and constant magnetic field is provided by a powerful magnet. The object is irradiated by an acoustic wave produced, for example, by a piezoelectric ultrasound transducer. The propagating acoustic wave induces mechanical motion of free charges (ions/electrons) present in conductive tissues. The Lorentz force pushes charges of opposite sign in opposite directions, creating an electric potential in the tissues. This potential is picked up by electrodes immersed in saline solution and measured. The inverse problem of MAET is to recover the spatial conductivity distribution of the object from measurements of this potential.

1.1 Mathematical model of MAET

In order to make this document somewhat self-contained, we will describe what is being computed in mathematical terms. For a detailed account of the physics and mathematics underlying MAET, we defer to the works listed at the end of this section and references therein. We assume herein that you are familiar with concepts, terminology, and notation from vector calculus and

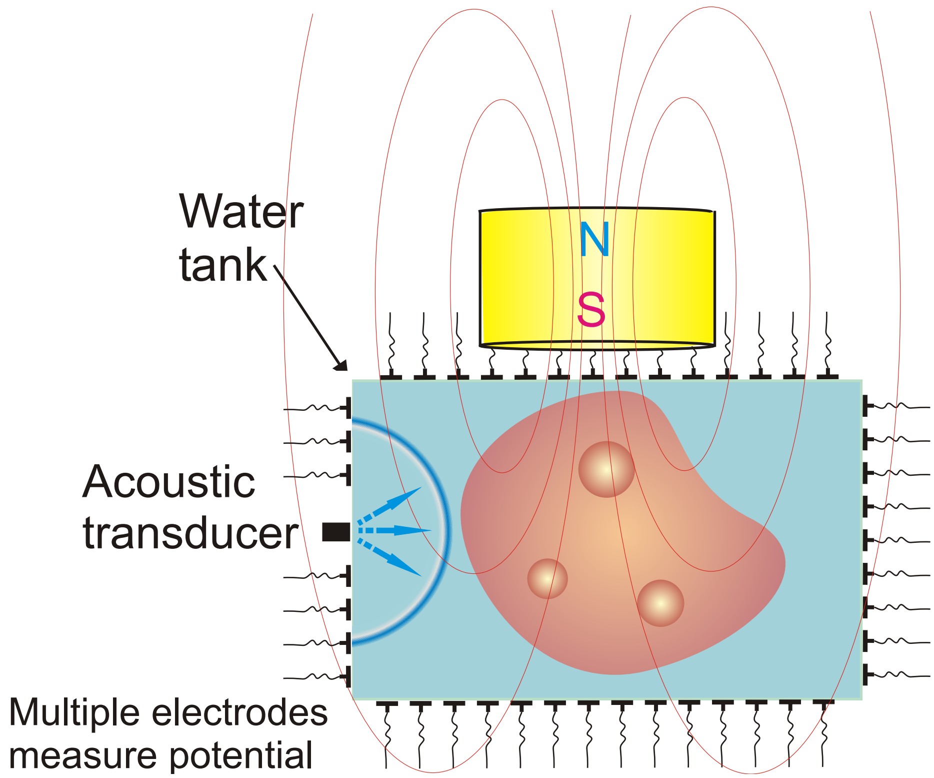

Figure 1: A schematic illustration of the MAET procedure (not an illustration of an actual device)

electromagnetism. Otherwise, see e.g. *Introduction to Electrodynamics* by David J. Griffiths (Cambridge University Press, 2017).

To assist in our presentation, we provide an illustration of the acquisition geometry assumed by our model in Figure 2. Measurements are made inside a circular chamber with an electrically insulated boundary. The radius of the chamber is fixed to be 1. Point-like electrodes are placed at equispaced points $(y_1, y_2, \ldots)$ on a ring of a certain radius R, where $0 < R < 1$. The object is assumed to be contained in a smaller inner chamber completed surrounded by the electrodes (white inner disk). The conductivity outside of this disk (shaded in gray) is assumed to be constant and equal to 1.

Let us denote the space- and time-dependent electric potential inside the chamber by $U(\mathbf{x}, t)$ where $\mathbf{x} = (x_1, x_2)$ is a 2D vector with real entries and $t > 0$ is a real number.

For an N electrode acquisition scheme, MAET measurements take the form of weighted average

$$M(t) = \sum_{j=1}^{N} W_j U(\mathbf{y}_j, t).$$

The points $\mathbf{y}_j$, $j = 1, \dots, N$, are the location of the electrodes. They are determined by fixing an initial angle α and using the formulas

$$\mathbf{y}_j = R(\cos\theta_j, \sin\theta_j), \quad \theta_j = \alpha + \frac{2\pi}{N}(j-1).$$

The weights W_j are determined by an angle γ and the formula

$$W_j = \cos(\theta_j - \gamma).$$

The simplest example is when $N = 2$ and $\gamma = \alpha$, so that $W_1 = 1$ and $W_2 = -1$. In this case, $M(t)$ is just the difference in potential between the two electrode points $\mathbf{y}_1$ and $\mathbf{y}_2$:

$$M(t) = U(\mathbf{y}_1, t) - U(\mathbf{y}_2, t).$$

A fundamental object of interest in the study of MAET is the so-called *lead current*. This is the electrostatic current that would flow inside the chamber if currents $W_1, \dots, W_n$ were injected through the electrode points in the absence of the magnetic field and acoustic excitation. The lead currents describe the sensitivity of the measurement system to the potential $U(\mathbf{x}, t)$. It is shown in [1,2] that the measurements $M(t)$ can be used to recover the curl of the lead current. The mathematical arguments are technical and we will not reproduce them here.

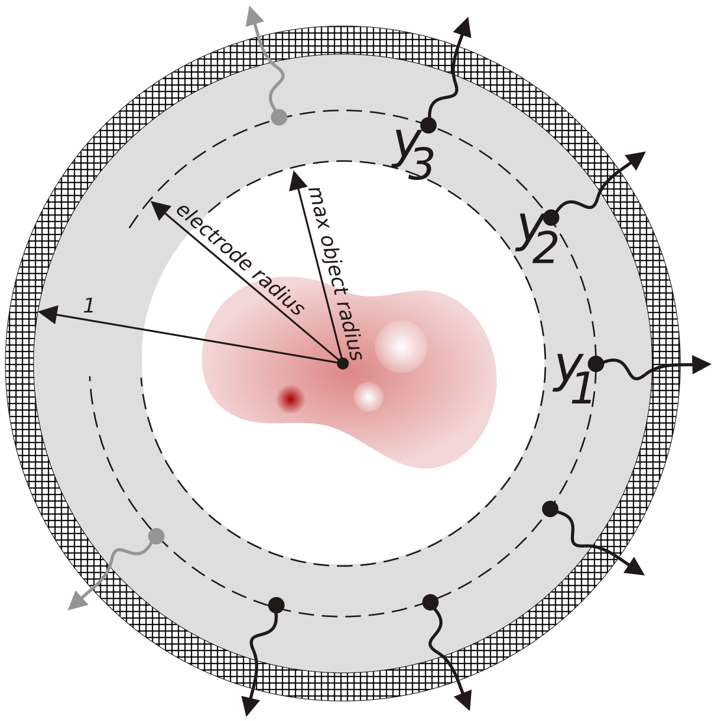

Figure 2: Data acquisition schematic of MAET in a 2D chamber.

Let us denote the lead currents by

$$\mathbf{J}(\mathbf{x}) = (J_1(\mathbf{x}), J_2(\mathbf{x})).$$

They are related to the *lead potential* $u(\mathbf{x})$ according to the equation

$$\mathbf{J}(\mathbf{x}) = \sigma(\mathbf{x})\nabla u(\mathbf{x}).$$

The (2D) curl of the lead current is given by the formula

$$C(\mathbf{x}) = \frac{\partial J_2}{\partial x_1} - \frac{\partial J_1}{\partial x_2}.$$

In [1, 2], it is shown that this curl can be expressed in terms of $\ln \sigma$ by the formula

$$C(\mathbf{x}) = J_2 \frac{\partial \ln \sigma}{\partial x_1} - J_1 \frac{\partial \ln \sigma}{\partial x_2}.$$

The problems we consider are

- (Direct problem). Determine the curl $C(\mathbf{x})$, given the logarithm of conductivity $\ln \sigma$ and weights $W_1, \ldots, W_N$.
- (Inverse problem). Reconstruct $\ln \sigma$ from one or more curls of lead currents (corresponding to different sets of weights).

The maetsim software contains numerical implementations for solving the direct and inverse problems. In the latter case, two sets of curls are used to reconstruct the logarithm of conductivity. Details of the implementations can be found in [1].

References

[1] Philip Hoskins, *Analytical and Numerical Study of Inverse Problems Arising in Some Novel Imaging Modalities*. PhD Thesis. University of Arizona. May 2020.

[2] Leonid Kunyansky et al, *Rotational Magneto-Acousto-Electric Tomography (MAET): Theory and Experimental Validation*. Phys Math Biol. 2017.

2 Software

The maetsim software is a collection of MATLAB scripts and functions which allow one to perform some simple simulations of the direct and inverse problems of 2D MAET inside a circular chamber. In particular, the functions in this package can be used to compute curls of currents from model density distributions of the logarithm of conductivity inside the chamber, and to reconstruct the logarithm of conductivity from two different curls.

The toolset consists of two scripts: direct.m and inverse.m, several subroutines called by these scripts. Several examples are included in the archive which can be used to check the installation and to demonstrate the functionality of the software. The software has been developed and tested in MATLAB version 9.5. No other MathWorks products or toolboxes are required.

2.1 How to install

It is recommended to follow these steps:

- Extract the archive in a directory of your choice. This creates a directory called maetsim with all the files of the software.
- Start MATLAB and navigate to the maetsim directory.
- Run the script startmaet.m. This adds the maetsim directory to the MATLAB search path for the current MATLAB session. If you wish to permanently add the directory to the search path for future sessions, follow the instructions for your operating system at

  https://www.mathworks.com/help/matlab/matlab_env/add-remove-or-reorder-folders-on-the-search-path.html

- Navigate to the maetsim/examples directory. (See Section 6 "Examples provided" below.)
  - Navigate to the maetsim/examples/direct directory. Run the direct script.
  - Navigate to the maetsim/examples/inverse directory. Run the inverse script.

2.2 Geometric parameters

The parameters determining the geometry of the simulation are placed in the file geometry.in. By default, this file reads:

```
513          = ndim
0.7          = objRadius
0.9          = electrodeDist
4            = nElectrodes
```

These parameters are:

ndim The size of the computational grid. That is, computations are performed on an ndim $\times$ ndim grid disceretization of the unit square. ndim must be an integer of the form $2^m + 1$ where m is a positive integer. For example, $513 = 2^9 + 1$.

objRadius The size of the inner chamber. This must be a floating point number between 0 and 1.

electrodeDist This is the distance of the electrode points from the center of the chamber. It must be a floating point number less than 1 and greater than objRadius.

nElectrodes An integer specifying the number of electrodes.

The geometric parameters must always be provided in geometry.in using the format exactly as described in this section.

3 .d files and tools

The maetsim software reads from and writes to files with the extension ".d". These are binary files which store data from a 2D floating point array. The data is written in the 32-bit IEEE single precision format in row-major order. Several functions are provided to read/write MATLAB arrays from/to .d files. These functions can be found in the maetsim/subroutines directory.

3.1 readd.m

The function **readd** reads a .d file into a MATLAB array. An example of how to use this function is as follows:

```
A = readd('file.d');
```

If executed in the MATLAB console, then the values stored in file.d will be read into the MATLAB array A.

3.2 ndwrite.m

The function **ndwrite** writes a MATLAB array into a .d file. An example of how to use this function is as follows

```
ndwrite(A,'file.d');
```

If the above code is executed in the MATLAB console, then the array A will be written into file.d.

3.3 viewd.m

The data stored in file.d can be displayed in a colormap plot by calling the **viewd** function as follows

```
viewd('file.d');
```

Alternative, the following syntax can also be used

```
viewd file.d;
```

4 Direct

The function direct loads a 2D phantom (stored in binary with extension ".d") representing the logarithm of conductivity and computes the lead potential, horizontal and vertical components of the lead current, and the curl of the lead current. The simplest way to call the function is to execute the following in the MATLAB console:

```
direct();
```

Along with the .d file of the phantom, the function also loads a file config.in. This file contains parameters used by the numerical routines. An example of a properly formatted config file is as follows:

```
45              = initAngle
45              = weightAngle
curl.d          = curlID
curx.d          = curxID
cury.d          = curyID
u.d             = potID
phant.d         = condlogID
```

The parameters are

initAngle The angle (in degrees) which determines the position of the first electrode. This is a floating point number between 0 and 360.

weightAngle The angle (in degrees) which determines the weights. This is a floating point number between 0 and 360.

curlID A string which gives the name of the file where the curl data will be written.

curxID The file where the horizontal component of the lead current will be written.

curyID The file where the vertical component of the lead current will be written.

potID The file where the lead potential will be written.

condlogID The file where the phantom data is stored.

Both the phantom data file and config.in must be located in the same directory where direct is called.

An alternative method for calling the function direct is to provide a string argument, e.g:

```
direct('myconfig.in');
```

In this case, the function will load a config file with the name myconfig.in. It is sometimes convenient to have several config files corresponding to different weights. For example, if config1.in and config2.in are located in the same directory, then two solutions of the direct problem can be computed by executing the following code in the MATLAB console:

```
direct('config1.in');
direct('config2.in');
```

5 Inverse

The function inverse loads two curl files and reconstructs the logarithm of the conductivity. To call inverse, execute the following line of code in the MATLAB console:

```
inverse();
```

The following files must be located in the same directory where inverse is called:

- inverse.in
- Data files for each curl in .d format
- For each curl, a corresponding configuration file containing parameters of the electrode configuration (2 total).

5.1 inverse.in

An example of inverse.in is given below:

```
config1.in        = leadConfig1
config2.in        = leadConfig2
recondlog.d       = condlogID
```

It contains three string parameters. The parameters are:

leadConfig1 The configuration file corresponding to the first curl

leadConfig2 The configuration file corresponding to the second curl

condlogID The data file where the reconstructed logarithm of conductivity will be written

5.2 Curl configuration files

Along with the curl files, inverse requires a configuration file corresponding to each curl to be located in the same directory. In the example of the previous subsection, these files were called config1.in and config2.in. An example of the contents of config1.in is given below:

```
45                        = initAngle
45                        = weightAngle
curl1.d                   = curlID
```

The file contains two floating point parameters which determine the electrode positions and weights and a string parameter which contains the name of the binary file containing the curl data.

6 Examples provided

The archive contains a directory entitled *examples*. This contains three further subdirectories containing example data and configuration files as described below. In order to run examples, copy the "examples" directory with all the subdirectory into your working directory while preserving the tree structure. The three subdirectories and instructions for running each example are described below:

direct Call the function direct() from the MATLAB console.

twocurls Execute the script runall from the MATLAB console.

inverse Call the function inverse() from the MATLAB console. Compare the result recondlog.d to the original phantom phant.d.