User manual for the FDEMtools package

G. P. Deidda[*] P. Díaz de Alba[†] C. Fenu[‡] G. Lovicu[§] G. Rodriguez[¶]

April 15, 2020

The FDEMtools package consists of a set of MATLAB routines for the inversion of frequency domain electromagnetic (FDEM) data. In addition to the forward problem computation and the solution routines, the package also includes 1D and 2D test problems on synthetic and experimental data sets, respectively, which allow the user to experiment with different numerical strategies, compare the results, and draw conclusions. An introduction to the toolbox is provided by the paper

- G. P. Deidda, P. Díaz de Alba, C. Fenu, G. Lovicu, and G. Rodriguez.
  FDEMtools: a MATLAB package for FDEM data inversion
  arXiv:1808.04779 [math.NA], 2019.

to which the software is attached.

The package is distributed as a compressed archive. Uncompressing the file creates the directory `FDEMtools`. To be able to use the package from other directories, its name must be added to the MATLAB search path, either by the command `addpath` or by using the menu in the graphical user interface of MATLAB. The toolbox content are described in detail in the `Contents.m` and `README.txt` files, located in the main directory. The software uses some functions from the package *Regularization Tools*, created by P. C. Hansen (`http://people.compute.dtu.dk/pcha/Regutools/index.html`. This software package should be installed and added to the search path, before using FDEMtools. The MATLAB PDF reader should also be configured.

For any comment or suggestion about the computer code, contact by e-mail any of the authors. New version of the package will be available at the web page `http://bugs.unica.it/cana/software`

1 The graphical user interface FDEMgui

In order to simplify experimenting with the functions of the package, the toolbox includes a MATLAB graphical user interface (GUI). It is started by issuing the command `FDEMgui` in the main MATLAB window; see Figure 1.

[*]Department of Civil and Environmental Engineering and Architecture, University of Cagliari, via Marengo 2, 09123 Cagliari, Italy (`gpdeidda@unica.it`).

[†]Department of Mathematics and Computer Science, University of Cagliari, viale Merello 92, 09123 Cagliari, Italy (`patricia.diazdealba@gmail.com`).

[‡]Department of Mathematics and Computer Science, University of Cagliari, viale Merello 92, 09123 Cagliari, Italy (`kate.fenu@unica.it`).

[§]Department of Electrical and Electronic Engineering, University of Cagliari, via Marengo 2, 09123 Cagliari, Italy (`lovicu.gabriele.gl@gmail.com`).

[¶]Department of Mathematics and Computer Science, University of Cagliari, viale Merello 92, 09123 Cagliari, Italy (`rodriguez@unica.it`).

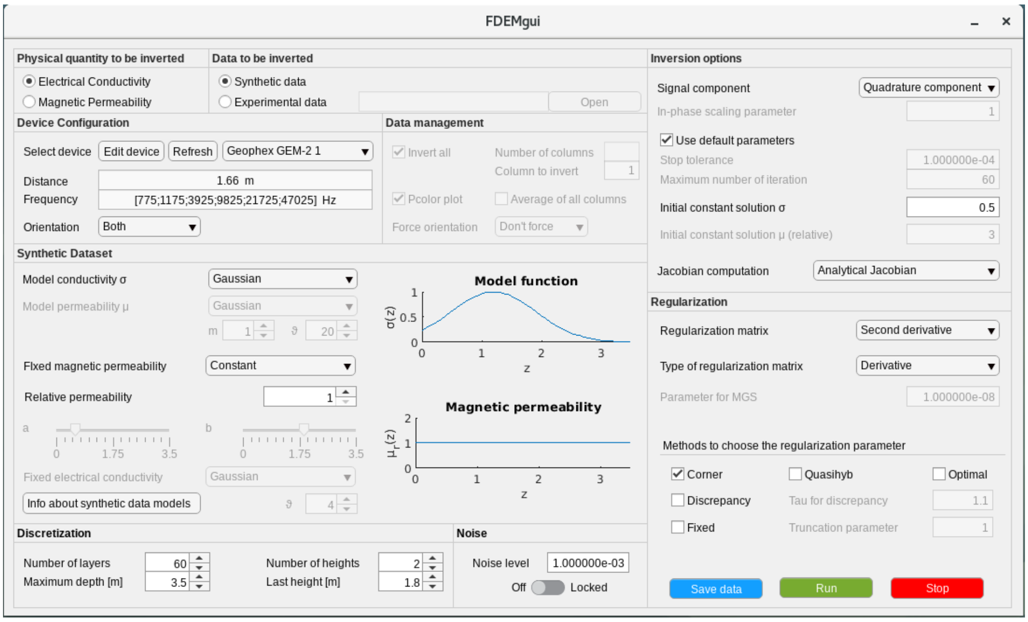


Figure 1: FDEMgui.

It is composed of a set of input panels that let the user generate or load data sets, choose different approaches and parameters for the inversion algorithm and, finally, visualize the computed results. Each panel is briefly described in the following paragraphs.

- **Physical quantity to be inverted**. This block, highlighted in Figure 2, allows the user to choose the physical quantity to be reconstructed, i.e., either the electrical conductivity or the magnetic permeability of the soil.

- **Data to be inverted**. Here, the user chooses between inverting synthetic or experimental data; see Figure 3.

  If synthetic data is chosen, the **Device configuration** panel is activated, where the user can select the device configuration for generating the data; see next paragraph for more information about this panel. If, on the contrary, the user wants to process experimental data, a `.mat` file must be loaded. The data set contained in the file has to be encoded using the data structure described in Section 3.

- **Device configuration.** When this panel is activated, the device can be selected from a list. Each stored device corresponds to a particular configuration, i.e., inter-coil distances ρ_i and frequencies f_i. The orientation of the device coils must be selected too.

  If a new configuration is needed, which is not present in the available list, it can be added by selecting the button *Edit device*. To refresh the whole panel, click the *Refresh* button.

- **Data management.** When this panel is activated, the user can choose between inverting all the columns (i.e., measurements) contained in a data set loaded from a file (*Invert all*), or select just one column by clicking on *Number of columns* and choosing the column to process. Another possibility, in the case of a data set containing repeated measurements, is to consider an average of all columns, by selecting the option *Average of all columns*. If more

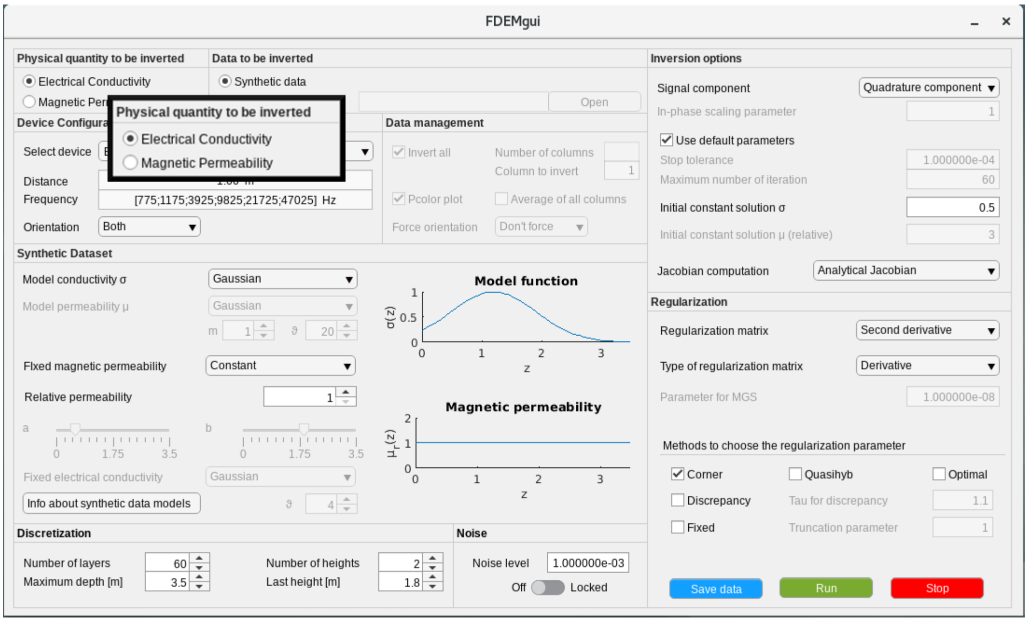


Figure 2: Physical quantity to be inverted.

than one column is being processed, a 2D plot will be displayed during the computation if *Pcolor plot* is selected.

- **Synthetic dataset.** Using this block the user can generate a data set, by choosing a model function and by tuning other parameters. A few different profiles for both the electrical conductivity and the magnetic permeability are provided; more information on them can be visualized by clicking on the *Info about synthetic data model* button.

- **Discretization.** This block contains some controls useful to define the discretization of the subsoil, such as *Number of layers*, *Maximum depth*, *Number of heights*, and *Maximal height*. In case multiple measurements are taken by placing the device at different heights above the ground surface, this section allows the user to configure the *Number of heights*, and the *Maximal height*. This last option represents the highest position at which the measurements are taken, in the case of multiple soundings.

- **Noise.** The *Noise* panel allows the user to select the noise level. This is useful to generate noisy synthetic data sets, but the selected noise level is also used for experimental data, in case the discrepancy principle is applied to estimating the regularization parameter. The *Locked* switch configures the random number generator so that exactly the same noise is generated upon subsequent computations; this is useful to experiment with a particular synthetic data set by varying some of the inversion parameters.

- **Inversion options.** In this section, the user can modify some parameters that influence the inversion process.

  - *Signal component*. Selection of the signal component to be inverted. It can be the in-phase component (i.e., the real part) of the signal, the quadrature component (the imaginary part), or the complex signal in its entirety.

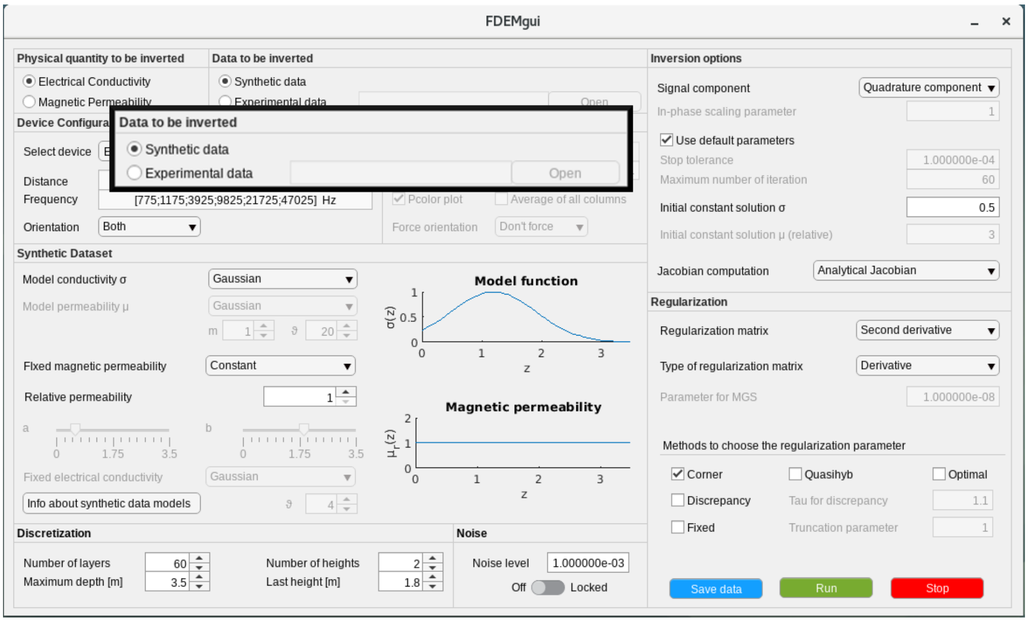


Figure 3: Choosing synthetic or experimental data.

  - *In-phase scaling parameter*. Positive parameter β used for scaling the in-phase component of the signal when complex data are inverted; see the paper.
  - The *Stop tolerance* and *Maximum number of iteration* are set to fixed values by default. If the user wants to modify them, unselect *Use default parameters*.
  - *Initial constant solution*. Sets the initial solution vector for the inversion of either the electrical conductivity or the magnetic permeability, respectively.
  - *Jacobian computation*. This drop menu selects the method for computing the Jacobian matrix: analytical Jacobian, Broyden update, and finite difference approximation.

- **Regularization**. Some regularization parameters can be set for the inversion algorithm.
  - *Regularization matrix*. The available choices are the identity matrix, and either the first derivative or the second derivative discrete approximations.
  - *Type of regularization matrix*. The user can choose between the direct use of the above regularization matrices (*Derivative*) or the minimum gradient support (*MGS*) sparsity promoting approach; see the paper. In the last case, a further parameter (*Parameter for MGS*) has to be chosen.
  - *Methods for choosing the regularization parameter*. The available methods for choosing the regularization parameter are the following: L-curve (*Corner*), a variation of quasi-optimality (*Quasihyb*), and the optimal parameter (*Optimal*, only for synthetic data sets, as it makes use of the exact solution). More simply, it is also possible to fix a value of the TSVD/TGSVD truncation parameter by the *Fixed* option.

The computation is started by the *Run* button; the computed results can be saved to both the MATLAB workspace and a data file by clicking on *Save data*. The *Stop* button allows for interrupting the computation, in case something goes wrong.

2 Tutorial for running numerical experiments

This section is a tutorial for running the test scripts `driver` and `driver2D` included in the package, and for repeating the same computation by using the FDEMgui graphical interface.

2.1 1D example

The 1D example included in the package runs a single inversion for the reconstruction of the electrical conductivity, starting from a complex synthetic data set. The data set is generated for a particular configuration of the Geophex GEM-2 instrument, corresponding to the inter-coil distance $\rho_1 = 1.66m$, six frequencies ($f_i = 775Hz, 1175Hz, 3925Hz, 9825Hz, 21725Hz, 47025Hz$), both orientation of the coils, placed at a height of $1m$. The algorithm is regularized by the second derivative operator and the L-curve method is used to choose the regularization parameter.

For repeating the computation performed by the script `driver` using the graphical interface FDEMgui, the user has to select *Electrical conductivity* as the physical quantity to be reconstructed, choose *synthetic data* as the data to invert, and enter the Geophex GEM-2.1 as the device configuration, with the orientation of the device set to *Both*. In the Synthetic Dataset panel the user should choose the Gaussian test function, fix the relative magnetic permeability to 1 and the magnetic permeability as a constant, and then define some parameters for the discretization of the problem:

- *Number of layers*: 35;
- *Maximum depth*: 3.5m;
- *Number of heights*: 1;
- *Maximal height*: 1m (in this case we consider the device placed at a fixed height, that is, the one to which "maximal height" corresponds);
- *Noise level*: 10^{-3}.

About the inversion options, the user should choose the *complex signal* as the signal component with the scaling factor for the in-phase component set as 1, click *use default parameters*, fix the *initial solution* as 0.5, and finally choose the analytical expression of the Jacobian matrix (*Analytical Jacobian* option).

Finally, the regularization matrix should be the *Second Derivative* operator, the type of regularization matrix fixed as *Derivative*, and the method to choose the regularization parameter set to *Corner*.

When the *Run* button is clicked, the computation is performed and the results are displayed.

Figure 4 displays the solutions computed by the regularized Gauss–Newton method. The graph on the left of Figure 5 reports the regularized solution selected by the L-curve method.

The graph on the right of Figure 5 shows the fitting between the input data set and the model prediction for the solution selected, for the two orientation of the coils. The order in which the data point are displayed is the internal ordering adopted in the package; see, e.g., the help for the `hratio` function.

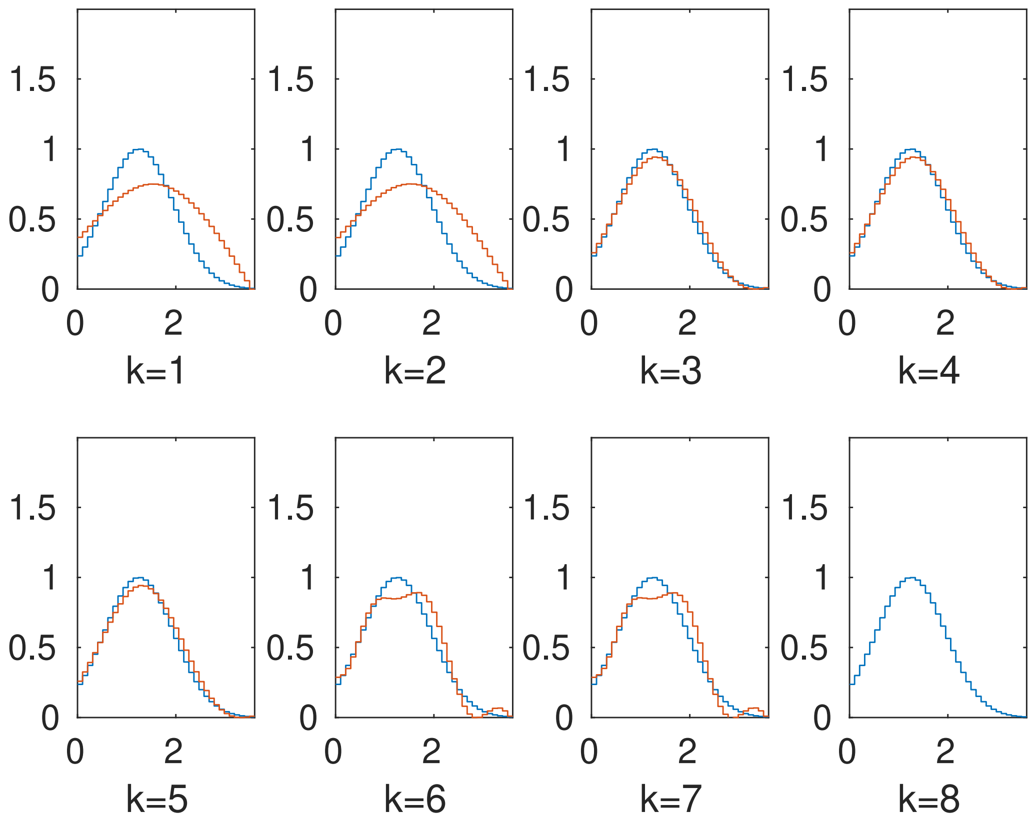


Figure 4: Reconstruction of the electrical conductivity from synthetic data. The figure displays the regularized solutions $\boldsymbol{\sigma}^{(k)}$ for $k = 1, \dots, 8$.

2.2 2D example

The `driver2D` test script performs a pseudo-2D inversion in order to reconstruct the electrical conductivity starting from an experimental data set. The data corresponds to the Geonics-EM38 configuration, with a pair of coils at distance $\rho_1 = 1m$, and operating at the frequency $f_1 = 14600Hz$. Also in this case, the algorithm is regularized by the second derivative operator and applies the L-curve method for the choice of the regularization parameter.

For performing the same computation using the graphical user interface FDEMgui, we fix the relative magnetic permeability to 1 and the magnetic permeability as a constant. We set the following parameters for the discretization:

- *Number of layers*: 60;
- *Maximum depth*: 3.5m;
- *Number of heights*: 16;
- *Maximal height*: 1.5m.

This data set contains just the quadrature component of the signal, and this configuration option is automatically selected by the GUI. We set the *initial solution* to 0.5, and choose the analytical expression of the Jacobian matrix (*Analytical Jacobian*).

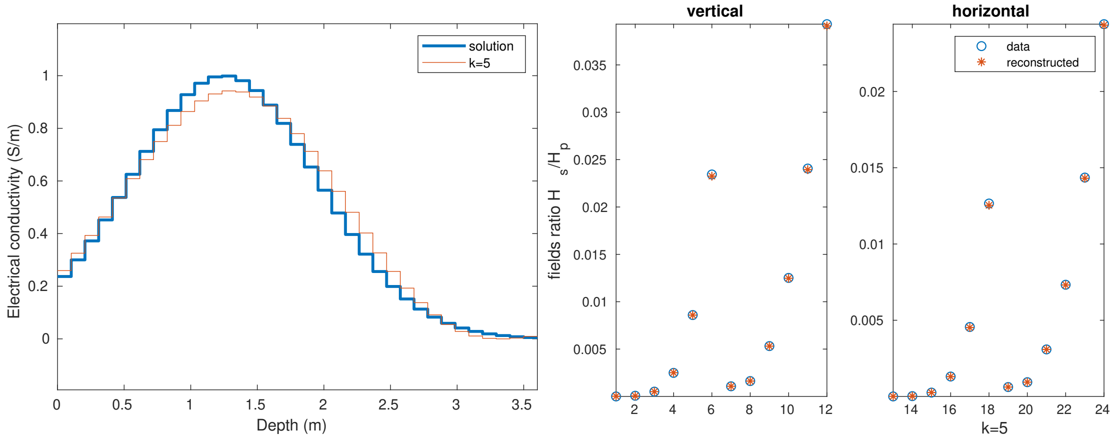


Figure 5: Reconstruction of the electrical conductivity from synthetic data. On the left, the solution selected by the L-curve method; on the right, the data fitting between the exact and reconstructed data.

Finally, the regularization matrix is the *Second derivative*, the type of regularization matrix is *Derivative*, and the method for the choice of the regularization parameter is *Corner*. In this case, rather than generating a synthetic data set, the user must load the file `test2Ddata.mat`, located in the subfolder *data* of the main FDEMtools directory. So, *Experimental data* must be checked, and the user should select, in the Data management panel, *Invert all* columns, *Don't force* the orientation, and make a *Pcolor plot*.

When the *Run* button is activated, the plot in Figure 6) is visualized. The results are saved in the file `FDEMoutput.mat` by clicking on *Save data*.

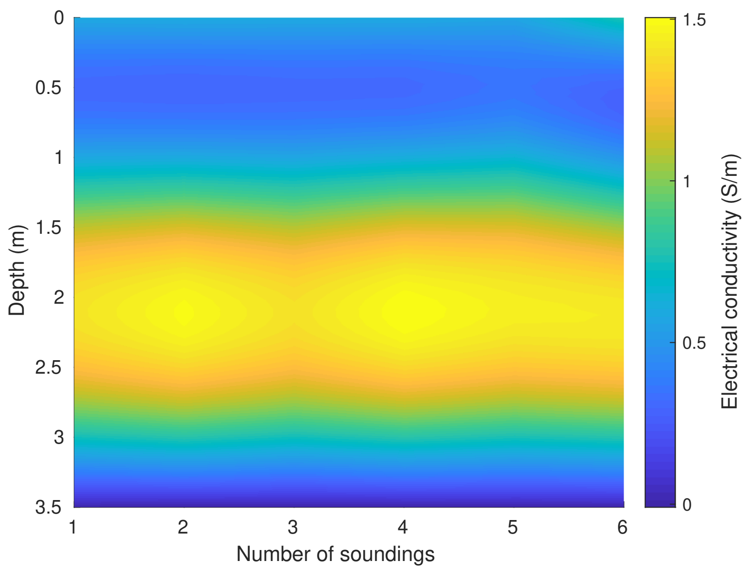


Figure 6: Test script `driver2D`: reconstruction of the electrical conductivity from experimental data.

3 File structure for loaded experimental data

In this section, we describe the data structure used by FDEMtools as an input format, and we briefly explain how to create such a data structure from an Excel file which contains experimental data.

The input data is encoded in a MATLAB structure array; type `help struct` in the MATLAB command window for more information. The fields for this structure array are listed in Table 1. Here are some remarks:

- if `type='complex'`, then both `vdata` and `hdata` are complex arrays, otherwise they are real arrays;
- if one of the orientations of the coils is not used, the corresponding data matrix is empty;
- the `coors` field is a matrix with three columns, each row containing the three coordinates (generally read by a GPS) for the corresponding sounding.

Table 1: Input data structure for FDEMtools.

`FDEMdata.name`	name for the data set (string)
`FDEMdata.R`	coil distances (scalar or vector)
`FDEMdata.freq`	device frequencies (scalar or vector)
`FDEMdata.height`	device heights (scalar or vector)
`FDEMdata.vdata`	data for the vertical orientation (matrix)
`FDEMdata.hdata`	data for the horizontal orientation (matrix)
`FDEMdata.orient`	orientation ('vertical', 'horizontal', or 'both')
`FDEMdata.coors`	geographical coordinates for each sounding (matrix)
`FDEMdata.type`	either 'real', 'imag', or 'complex'
`FDEMdata.device`	device model (string)
`FDEMdata.info`	information about the data set (string)

A similar structure array is used for encoding the output of the inversion algorithm, when it is written in a data file; its fields are displayed in Table 2.

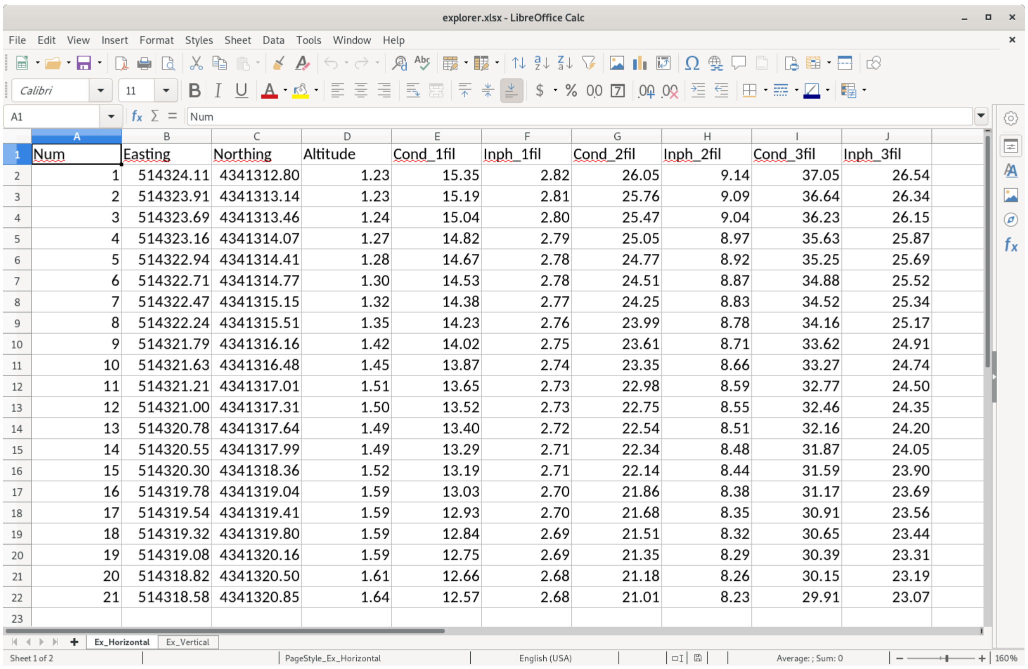
Table 2: Output data structure for FDEMtools.

`FDEMresults.solution`	computed solutions (matrix)
`FDEMresults.model`	forward model adopted (function handle)
`FDEMresults.z`	subsoil discretization (vector)
`FDEMresults.opts`	inversion parameters setting (`struct`)
`FDEMresults.parameter_choice`	method for choosing the regularization parameter (string)

The Excel file `explorer.xlsx`, contained in the `data` subdirectory of the FDEMtools main directory and represented in Figure 7, contains a set of soundings performed by a CMD-Explorer (three intercoil distances and one operating frequency). The two sheets contain the data recorded by the vertical and the horizontal orientations of the coil, respectively.

The Excel file is structured as follows:

- Column A: contains an order index for the measurements;
- Column B and C: correspond to the coordinates of the points at which the measurements have been taken;



Num	Easting	Northing	Altitude	Cond_1fil	Inph_1fil	Cond_2fil	Inph_2fil	Cond_3fil	Inph_3fil
1	514324.11	4341312.80	1.23	15.35	2.82	26.05	9.14	37.05	26.54
2	514323.91	4341313.14	1.23	15.19	2.81	25.76	9.09	36.64	26.34
3	514323.69	4341313.46	1.24	15.04	2.80	25.47	9.04	36.23	26.15
4	514323.16	4341314.07	1.27	14.82	2.79	25.05	8.97	35.63	25.87
5	514322.94	4341314.41	1.28	14.67	2.78	24.77	8.92	35.25	25.69
6	514322.71	4341314.77	1.30	14.53	2.78	24.51	8.87	34.88	25.52
7	514322.47	4341315.15	1.32	14.38	2.77	24.25	8.83	34.52	25.34
8	514322.24	4341315.51	1.35	14.23	2.76	23.99	8.78	34.16	25.17
9	514321.79	4341316.16	1.42	14.02	2.75	23.61	8.71	33.62	24.91
10	514321.63	4341316.48	1.45	13.87	2.74	23.35	8.66	33.27	24.74
11	514321.21	4341317.01	1.51	13.65	2.73	22.98	8.59	32.77	24.50
12	514321.00	4341317.31	1.50	13.52	2.73	22.75	8.55	32.46	24.35
13	514320.78	4341317.64	1.49	13.40	2.72	22.54	8.51	32.16	24.20
14	514320.55	4341317.99	1.49	13.29	2.71	22.34	8.48	31.87	24.05
15	514320.30	4341318.36	1.52	13.19	2.71	22.14	8.44	31.59	23.90
16	514319.78	4341319.04	1.59	13.03	2.70	21.86	8.38	31.17	23.69
17	514319.54	4341319.41	1.59	12.93	2.70	21.68	8.35	30.91	23.56
18	514319.32	4341319.80	1.59	12.84	2.69	21.51	8.32	30.65	23.44
19	514319.08	4341320.16	1.59	12.75	2.69	21.35	8.29	30.39	23.31
20	514318.82	4341320.50	1.61	12.66	2.68	21.18	8.26	30.15	23.19
21	514318.58	4341320.85	1.64	12.57	2.68	21.01	8.23	29.91	23.07

Figure 7: Excel file containing the experimental data to be inverted.

- Column D: represents the elevation in meters;
- Columns E, G and I: contain the apparent conductivity in mS/m (that is, a rescaling of the quadrature component of the signal) for each of the three different intercoil distances;
- Columns F, H, and J: contain the in-phase component in ppt (parts per thousands) for each different intercoil distance.

The MATLAB routine `converter.m`, located in the same *data* subfolder, reads the Excel file and creates the structure array `FDEMdata` containing the data set. Finally, it saves the data to a `.mat` file, which will be easily loaded using either the GUI, or the `driver.m` example script. Here is how the newly created structure array is represented in the MATLAB command window:

```
FDEMdata =
  struct with fields:
      name: 'explorer_data'
         R: [3x1 double]
      freq: 10000
    height: 0.900000000000000
     vdata: [3x20 double]
     hdata: [3x20 double]
    orient: 'both'
     coors: [3x20 double]
      type: 'complex'
    device: 'CMD-Explorer'
      info: 'conversion example for CMD-Explorer'
```

After inspecting the `converter.m` script, it should be easy for the user to modify it, in order to convert Excel files with a similar layout to a format suitable for FDEMtools.