

## A MATLAB PROGRAM FOR THE ANALYSIS AND INTERPRETATION OF TRANSIENT ELECTROMAGNETIC SOUNDING DATA

**Andreas Tzanis**

*University of Athens, Department of Geophysics and Geothermy, Zografou 15784, Greece,  
atzanis@geol.uoa.gr*

### **Abstract**

*Herein we present a software system, written in MATLAB, to interpret TEM sounding data. The program, dubbed maTEM, is designed to process, model and invert multiple soundings, either individually, or simultaneously along profiles. The latter capability allows for laterally constrained inversion, so as to generate pseudo-2D or 2D resistivity sections based on the program EMIDINV v2.13 by the Hydro-Geophysics Group of the University of Aarhus, Denmark. Using maTEM, the analyst may import and display data multiple data sets, denoise and smooth the data, perform approximate inversions, design 1-D model(s) graphically, perform forward modelling and inversion and generate/update data base in which to store the results. Finally, the analyst may use the data base to create 2-D and 3-D displays of the geoelectric structure with built-in graphical functions. maTEM is highly modular so that additional functions can be added at any time, at minimal programming cost. Although the software presented herein is focused on the analysis of TEM data, the maTEM concept has been designed ready to incorporate additional electrical and EM geophysical sounding methods and to mutually constrained analysis of different geophysical data sets.*

**Key words:** *Electromagnetic sounding, TEM, MATLAB, geophysical interpretation.*

### **1. Introduction**

The Transient Electromagnetic (TEM) exploration method is becoming increasingly popular for its effectiveness and robustness in hydrogeological, environmental, engineering geological applications, as well as general geological and geophysical mapping. In fact, due to its sensitivity in resolving small contrasts in generally highly conducting structures, it has become the EM method of choice for such applications, as the detection and mapping of subterranean water concentrations and flow paths, detection and monitoring of salination processes in coastal aquifer systems, detection and monitoring of aquifer pollution and pollutant migration, detection and mapping of buried faults, deep lifelines, leakage, UXO etc.

A pre-requisite for TEM applications in hydrogeological and environmental studies is the quality of the data and interpretation scheme, with particular emphasis on the latter. This is because in most modern applications, it is not sufficient to map the conductive targets only, but also to measure their volume, delineate and characterize the hosting structures and cover materials, measure the volumetric water content, the differences in water conductivity due to pollution etc.

Due to the rather involved physics of the TEM method, the existing analysis and interpretation software is relatively limited, both in numbers and availability. In general one can find: (1) High cost

commercial software from a handful of manufacturers. (2) Extreme scarcity of open source software. In fact, the author only knows of the MATLAB forward code by Ingeman-Nielsen and Baumgartner (2006) and the old inversion codes of Walter Anderson; the latter are written in plain FORTRAN and are available from the USGS (Anderson, 1982a-c), but cannot be put to work by a person unfamiliar with old and contemporary editions of FORTRAN, let alone the complete absence of a visual presentation of the results. (3) Limited availability of inversion software, which is usually offered as an executable, without (or with limited) data management and display facilities; this is usually provided by academic institutions, e.g. the University of Oulu, Finland and the University of Aarhus, Denmark.

With the above two exceptions aside, the academic free software community has hitherto not produced remedies. The software presented herein, *maTEM* (*Matlab Transient ElectroMagnetics*), marks the beginning of an effort to create an advanced TEM analysis and interpretation package, which may also be expandable and customizable to the needs of a particular user, with little programming effort. Although easy to say, this would be a truly demanding and ambitious undertaking, had it not been for the existence of powerful, general purpose computing engines. On this basis, the realization of such a project is feasible because a computing engine will generally provide a complete high level programming environment, inclusive of graphics, which facilitates the development of advanced software because all the complexities pertaining to low level interfacing, programming and functionality are taken care by the engines themselves. This is much easier and faster than building programs from scratch in some conventional high level language such as C, C++ or FORTRAN.

## 2. Specifications and programming environment

A modern program, like *maTEM* aspires to be, should be able to offer a complete solution to the problem of analysis, interpretation and visualization of TEM survey data, allowing the user to:

1. Import and handle *multiple* TEM sounding data, i.e. from more than one sounding locations and more than one type of instruments.
2. Visualize and inspect the data.
3. Process the data with utilities enabling noise suppression, smoothing etc.
4. Perform fast approximate inversions (inverse mappings). These may assist in the interpretation effort, but their primary function would be to provide insight and guidelines for the design of 1-D interpretative models.
5. Design 1-D interpretative models.
6. Perform interactive, 1-D forward modelling and/or inversion for each sounding separately.
7. Perform simultaneous inversion of multiple soundings, either 2-D, or 1-D with application of lateral constraints (in quasi-layered earth structures this amounts to proxy-2D inversion).
8. Display the interpretation in the form of resistivity cross-sections with publication quality graphics; moreover, generate composite displays by overlaying more than one cross-sections.
9. Have the data and results organized in a data base in an automated and transparent manner, so that he/she may be able to concentrate on the analysis only.
10. Be able to repeat any one of the steps 1-9 above at *any* time, in order to improve/ correct the interpretation and seamlessly introduce the results into the data base.

The realization of these specifications requires an appropriate programming and computational environment and careful selection of the interpretation (modelling and inversion) approach.

General purpose scientific computing engines are not many: The most commonly available are the proprietary IDL™ and MATLAB™ and the freeware OCTAVE and SCILAB. Of course, there is an abundance of other (proprietary or non-proprietary) scientific data analysis software, which is generally not suitable, as it is usually specialized (e.g. statistical analysis, curve fitting, data modelling, graphics etc.) and does not provide a sufficiently general and versatile enough programming environment for the development of integrated and advanced software. Of all those, MATLAB™ is the most widespread, diverse and versatile to such an extent, that it has become the *de facto* standard in scientific and technical computing.

MATLAB was not available until the later part of the 1980's, and prior to that, FORTRAN was the language of choice for serious numerical computation. C was making its first steps towards establishing a foothold in the world of scientific computation, but it didn't offer built-in facilities for doing complex arithmetic and this was a serious drawback. Conversely, FORTRAN lacked some of C's advantages such as structures, pointers, and dynamic memory allocation.

The appearance of MATLAB and its imitators made a big impact in the scientific community. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects, familiar to FORTRAN programmers for being robust collections of tools for linear algebra. To do this, MATLAB also introduced a new vector-oriented programming language, an interactive environment, and built-in graphics. These features offered many advantages and boosted productivity in comparison to more traditional application development environments. Since, MATLAB has evolved to embed a large collection of state-of-the-art numerical tools, high quality graphics, object-oriented extensions, a built-in interactive debugger, web services and a host of other facilities. Of course, C and FORTRAN have also evolved, with some advanced editions also offering high-level (visual) application development environments. However, neither offers as many treats as MATLAB, graphics for instance being a major issue.

The requirement for a suitable and accurate modelling and inversion method is quite more difficult due to the scarcity of the publicly available software. Anderson's (1982a-c) pioneering FORTRAN codes are robust but insufficient because they deal only with the single/ coincident loop and central loop configurations. Karmis (2003) has developed FORTRAN programs to perform inversion of single- and multiple sounding data, which he has dubbed "*pointem*" and "*linetem*" respectively. While maTEM was being developed, these programs were not publicly available.

The remaining options were either to develop new modeling and inversion code, or, until this is done, to use the program "em1divn" developed by the Hydro-Geophysics Group of the University of Aarhus, Denmark. This program is well known and tested by the EM geophysical community and is delivered by its author(s) in stand-alone executable form, upon application. It may perform inversion of single or multiple soundings simultaneously; in the latter case, by introducing lateral constraints between soundings, i.e. restraining the variation between homologous layers with respect to each other, it may also perform quasi-2D inversion in layered earth structures (Auken and Christiansen, 2004; Christiansen and Auken, 2004; Auken et al, 2005). An additional important advantage is the possibility to adjust the transmitter waveform to the specifications of different instrument manufacturers. On the downside, the program cannot, at present, handle earth models with complex and frequency-dependent resistivity. Thus, in some cases it may fail to interpret data affected by induced polarization effects. Moreover, because it assumes paramagnetic earth materials, it cannot handle super-paramagnetic phenomena appearing in magnetized or iron-rich soils.

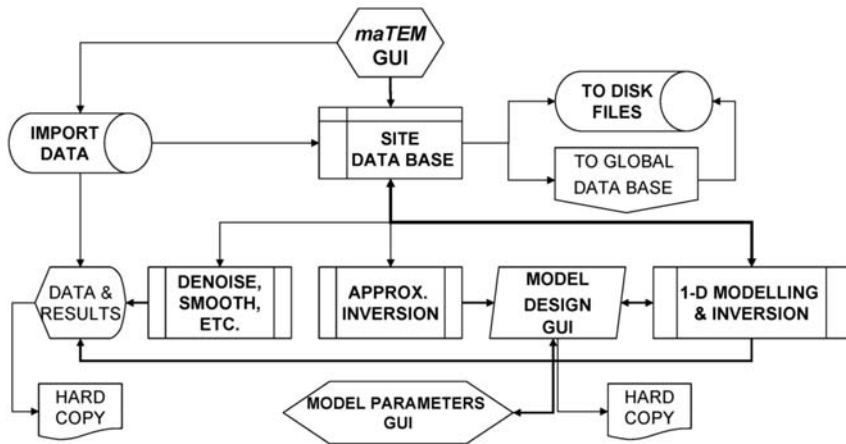


Fig. 1: Flowchart of a data analysis child process.

### 3. Program architecture

It should be apparent, even to the novice, that the specifications set at the beginning of Section 2, may only be realized by enabling the execution of multiple parallel processes. *maTEM* is designed so that each one of these processes will comprise the analysis of an individual TEM sounding at a given site (a *data analysis process*). The data and the results are assembled in a site-specific data base (DB). The individual site-DBs, in turn, contribute to the formation of a global-DB which can be independently handled, exported and re-imported. The global-DB also feeds with data and initial earth models the process of for multiple simultaneous inversions and assimilates the results for permanent storage and display. A GUI driven *parent* process, the *maTEM GUI*, controls the *data analysis child processes*. This design philosophy is illustrated and explained in the flowcharts of Figures 1-3.

Fig. 1 shows the (more or less self-explanatory) flow of work and information in a data analysis child process. Data processing (de-noising, smoothing etc.) are controlled by the *maTEM GUI*; the results are fed back to the site-DB and are hence available for the approximate and 1-D layered modelling and inversion operations. The site-specific Earth Model is designed and handled interactively (graphically) in an independent *Model Design GUI* which, in fact, is *child* of the *maTEM GUI*. This is supplied with the results of approximate inversion, which can be displayed at the user's discretion. The model design GUI generates a child process, the *Model Parameters GUI*, in which the user may manually introduce layer resistivities, thicknesses and most importantly, vertical and lateral constraints for the inversion. The site-DB can be exported in the form of a sounding data file (*tem-file*) and a sounding model file (*mod-file*).

Fig. 2 shows that the site-DBs of the parallel analysis child processes are collated to form the global-DB, in a manner that will be explained later. At present, the global-DB can be stored as a MATLAB binary file (*mat-file*). The information necessary for multiple simultaneous inversions (data and initial earth models) is extracted from the global-DB and the results are re-introduced for permanent storage. In addition, the results of the inversions can be extracted from the global-DB and stored independently as an ASCII file, the *inv-file*, to be available for visual inspection/ editing and display in the form of cross sections. An existing *inv-file* is not destroyed by subsequent write operations – any new results are merely appended to it, therefore caution should be exercised in order to keep the file free of unwanted repetitions.

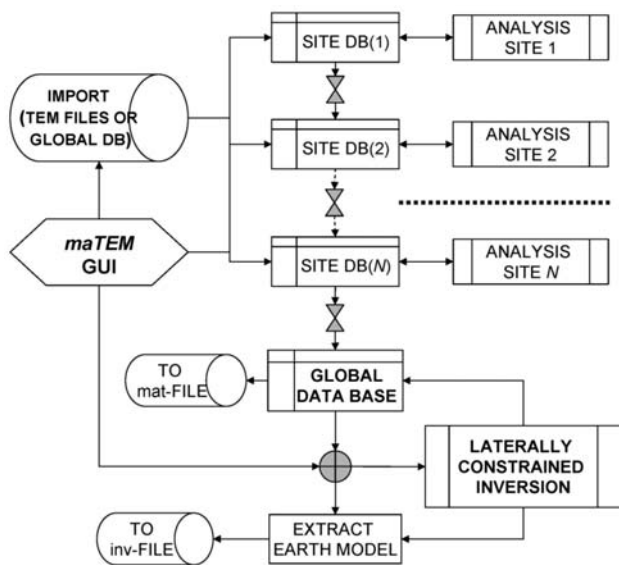


Fig. 2: Outline of the simultaneous analysis of multiple TEM soundings. Each “ANALYSIS” box corresponds to a data analysis child process.

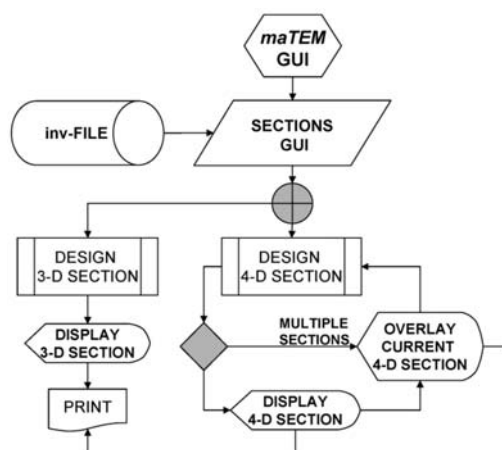


Fig. 3: Flowchart for the construction and display of interpretative cross-sections.

Fig. 3 shows how maTEM constructs and displays interpretative cross sections. The maTEM GUI launches yet another child process, the *Sections GUI*, through which the interpretation is imported from the inv-file, or extracted from the global-DB. The soundings available in the inv-file are displayed in map form and the user configures the cross-section by pointing and clicking. Cross-sections can be displayed in either 3-D (planar) or 4-D (volume) mode. In the latter case, multiple cross-sections can be overlaid in a single graph, thus enabling visualization of lateral correlations and changes in the geoelectric structure of the entire study area.

As is already clear, maTEM has been designed to be GUI-driven at all levels. This requires object-oriented programming and defines the frame and course of realizing the design specifications. In addition, one has to consider that all child processes must be executed in parallel and that the site- and global-DBs must be updated automatically every time there's some action by the user. Given that in the GUI-driven, object-oriented environment of MATLAB, the information pertaining to modifica-

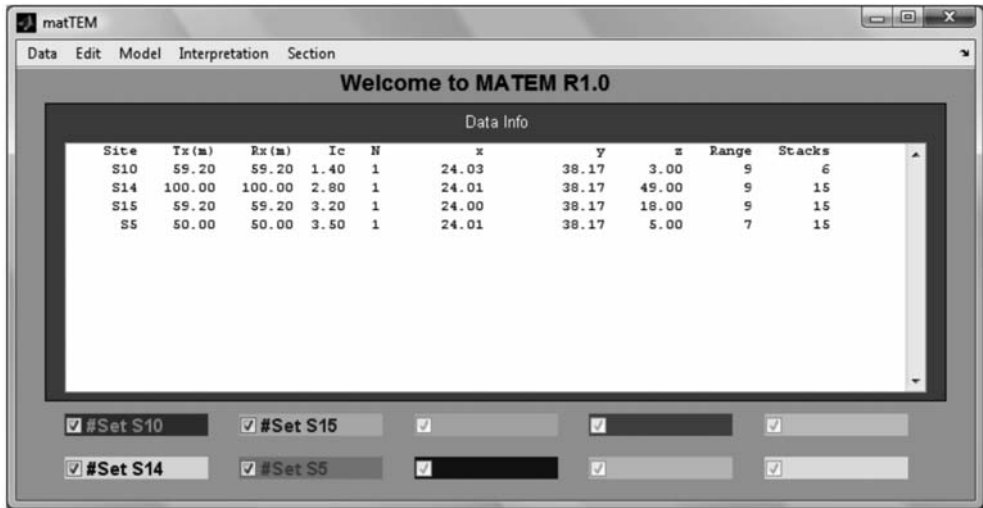
tions of graphical objects (in our case the graphical representations of data and earth models) is directly accessible through their respective handle structures, the following method for passing information between processes and data bases was adopted:

Let the data comprise  $n = 1, \dots, N$  soundings, with each sounding comprising  $m_n = 1, \dots, M_n$  channels (gates). Then, the handles of the graphical objects can be organized in data constructs comprising *homologous*  $1 \times N$  data structure arrays of data structures, in such a manner, that the  $n$ -th element of a data structure array will comprise a  $1 \times M_n$  data structure array whose fields are the handles of the  $M_n$  *homologous* graphical objects of the  $n$ -th sounding. Given that the  $m_n$ -th handle is actually a data structure conveying the properties of the graphical object, a strict hierarchy of data structures is thus created, organizing the available information in a manner directly accessible by reference. Any modification of the  $m_n$ -th graphical object is registered in the respective  $m_n$ -th handle and the hierarchical data construct is instantaneously updated with the new information.

The homologous structure arrays of graphical objects used in maTEM are associated with:

- a. Data manipulation and management: This construct contains the graphical objects representing the data of the  $n$ -th sounding. It allows the user to hide/show entire soundings for convenience, and, more importantly, to earmark any  $m_n$ -th data channel for *de-activation/re-activation*. The term *de-activation* implies the *exclusion* of the  $m_n$ -th data channel from further consideration and is applied to measurements heavily distorted by early time effects or by noise, to the point of being irrecoverable.
- b. Error bar manipulation and management: This construct contains the graphical objects representing the *observational error* associated with the  $m_n$ -th data channel in the  $n$ -th sounding. It allows the user to hide/show the error bars associated with the data of each sounding separately, thus facilitating the visual inspection and evaluation of individual soundings.
- c. Earth Model manipulation and management: This construct contains the handles of the graphical objects representing the 1-D geoelectric structural model used for the interpretation of the  $n$ -th sounding and is displayed in the “Model Design GUI”.
- d. Inversion parameters management: This contains the vertical and horizontal constraints applied to the inversion of the  $n$ -th model during the interpretation of the  $n$ -th sounding.

The site-DB is a data structure whose fields are alphanumeric and numeric constants and arrays containing all the useful information pertaining to an individual sounding. This includes survey-and-site-related information from the header(s) of the sounding data file, site coordinates, Tx/Rx size(s) and configuration, ramp duration, the channel central times, the sounding data in the form of impulse response, late-time apparent resistivity and true resistivity with associated uncertainties, the earmarks of the deactivated data channels, the approximate inverses, the interpretative 1-D Earth model and the associated theoretical response curves. It should be expected that the global-DB would be a  $1 \times N$  data structure array of the site-DB data structures, symbolically named  $\mathbf{D}(n)$ . When some graphical object pertaining to the  $n$ -th sounding data or model parameters is interactively modified during the course of the analysis, the corresponding field of the  $\mathbf{D}(n)$  is automatically updated via the pertinent handle. Likewise, if, after such a modification an approximate inversion or layered 1-D modeling or inversion is executed, the pertinent fields of  $\mathbf{D}(n)$  are automatically updated by direct substitution of the results. The global-DB can be exported as a MATLAB mat-file, together with the “inversion parameters” construct. This amounts to saving the analysis session. Re-importing the global-DB amounts to re-instating the session at the point at where it has been saved. This is an effective way to ensure the seamless continuity of work.



**Fig. 4:** The maTEM GUI after importation of data from four soundings collected with the TEM-FAST 48 system. The GUI displays key information about the soundings.

Every other graphical object not included in the above description is assigned with a unique *identification tag*”, using the ‘tag’ property of the object. The retrieval and registering of information in these objects is based upon acknowledgement of their ID tag. Finally, it should be noted that maTEM also features a collection of constants (keywords) and variables by which the user can choose between different modes of data and model displays, and tune the execution of em1dinv. The assignment of these parameters is also GUI driven and they can change during the course of analysis (runtime variables). They are organized in yet another global data structure whence they are passed to the appropriate calling routine. The specification and explanation of these variables is beyond the scope of this work.

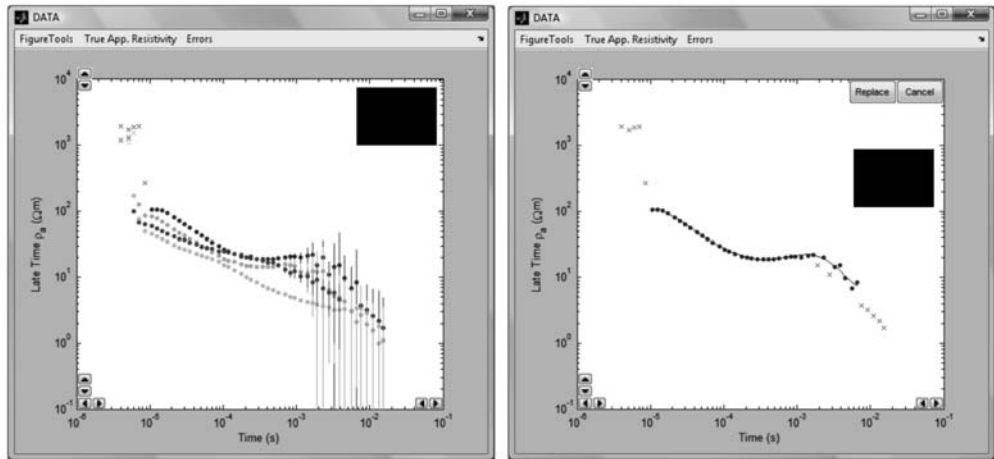
#### 4. Implementation

The maTEM GUI can be seen in Fig. 4, after a data set has been imported for analysis. The “Data” menu facilitates importation/ exportation of data and results, as well as general functions pertaining to the mode of data display (impulse response or apparent resistivity) and the type of the coordinates adopted for the survey. At present maTEM may accept the following types of data:

- Multiple *single-site* TEM data files in the TEM-FAST 48, AMIRA and em1dinv formats (each file contains one sounding).
- Multi-site TEM data files in the TEM-FAST 48 and AMIRA formats (each file contains more than one soundings)
- maTEM mat-files (i.e. the global-DB) in MATLAB binary format.

Upon importation the maTEM GUI displays important information as per Fig. 4. The (coloured) boxes indicate the colour associated with the data and models of each sounding and the checkboxes can be used to hide/ show the respective soundings for convenience.

The data is displayed in the “DATA” window as per Fig. 5a; this also comprises a GUI with menus



**Fig. 5:** (a) The four soundings shown in Fig.4, displayed as late time apparent resistivities. (b) Sounding with concealed errors, deactivated channels (x-markers) and a smooth model (continuous line) computed with a 9th order exponential spectrum and pending acceptance.

facilitating concealment/ projection of true apparent resistivity curves and error bars. The “Edit” menu offers de-noising and smoothing utilities. At this point de-noising comprises the de-activation of distorted data channels. This is done by pointing and clicking on the graphical representation of a data channel (solid circle). A deactivated channel is marked with x and may be re-activated in the very same way. Smoothing is possible with least squares smoothing splines (e.g. Wahba, 1990) or by fitting the decaying impulse response data with a function of the form

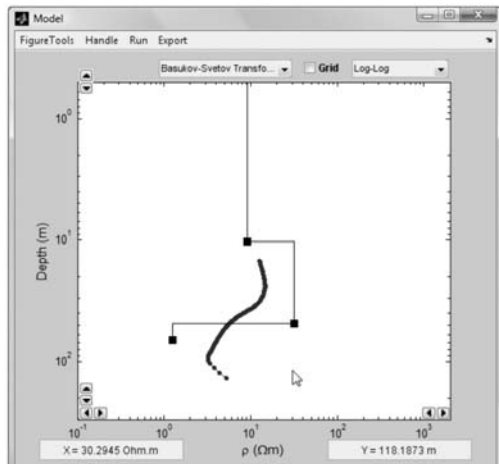
and recalculating the late time and true apparent resistivities (exponential spectrum method). An example is shown in Fig. 5b.

The “Model” menu initializes the Model Design GUI, in which layered Earth models can be created and manipulated via the dedicated “Handle” menu (Fig. 6). The models are introduced graphically, by pointing and clicking; for convenience and precision, the cursor coordinates (depth and resistivity) are indicated at the bottom of the Model Design GUI. The results of approximate inversion may also be overlaid to assist in the design. A model can be updated with insertion or deletion of layers and can be modified by capturing and shifting the breakpoints of the model (solid squares in Fig. 6 and 7a) with the mouse. There can be one layered model per sounding.

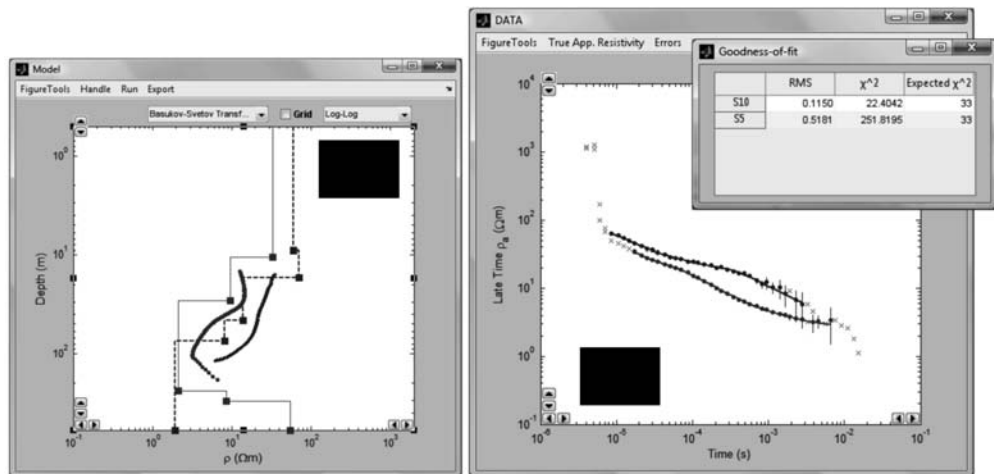
Upon completion of the graphical construction of an Earth model, the respective “Model Parameters GUI” is displayed. This, offers additional means of controlling the interpretation by allowing the manual fine-tuning of the model resistivities and thicknesses, as well as the definition of “prior information weights” (to fix or restrain the variation of a parameter depending on the existence of independent information), vertical constraints (restrain the variation between consecutive layers), and horizontal constraints (restrain the variation of homologous layers in neighbouring soundings during laterally-constrained inversion of multiple soundings along profiles).

The menu “Interpretation” is used to compute approximate inverses with the methods of Barsukov





**Fig. 6:** The Model Design GUI during the graphical construction of a layered earth model



**Fig. 7:** Simultaneous inversion of two soundings. (a) The final models are displayed in the Model Design GUI; approximate inverses are overlaid for comparison. (b) The model theoretical responses (continuous lines) are laid over the measured data. The “Goodness-of-fit” window displays the corresponding RMS and  $\chi^2$  metrics.

and Svetov (Barsukov et al, 2007) or S-inversion (Tartaras et al, 2000), to perform interactive forward modelling and inversion and to adjust the source waveform and the control parameters necessary for the execution of em1div. The latter (modelling and inversion) is also possible – and more versatile – via the “Run” menu of the Model Design GUI (Fig. 6 and 7a). Interactive forward modelling can be done either by capturing and shifting the breakpoints of the model (solid squares in Fig. 6 and 7a) with the mouse, or by typing the updated model parameters in the appropriate boxes of the Model Parameters GUI. The progress of the modelling exercise and the quality of the interpretation is monitored both visually and numerically by means of the RMS and  $c^2$  metrics projected on a separate, “Goodness-of-fit” window (Fig. 7b). Only one sounding can be modelled at a time in this manner. Conversely, more than one sounding can be inverted simultaneously, with or without vertical and lateral constraints. Fig. 7 displays the result of such an exercise for two de-noised soundings (S5 and S10 in Fig. 4). The results of the interpretation may be exported in the various available

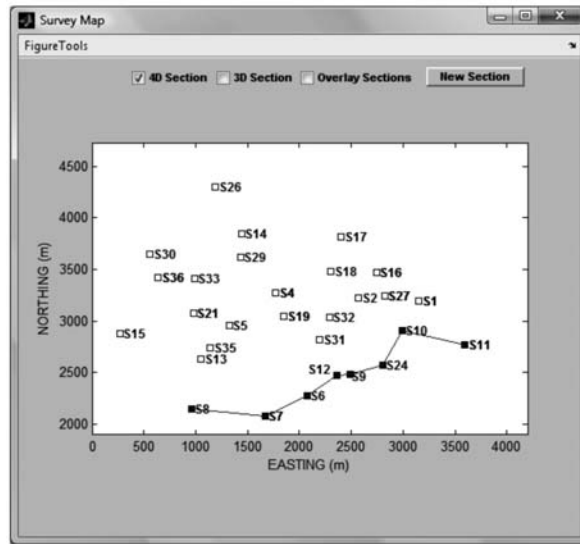


Fig. 8. The “Survey Map” GUI, used for designing a cross section.

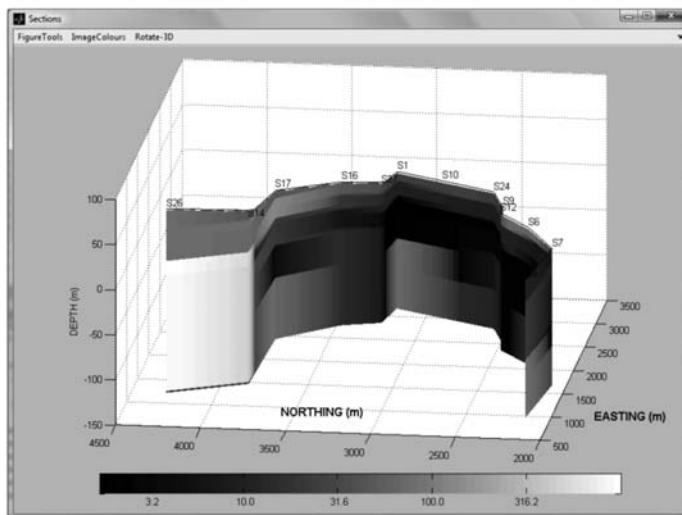


Fig. 9: A 4-D (volume) representation of two sections joined at S1, as viewed from the NE.

formats, via the “Export” menu of the Model Design GUI, or the appropriate menu entry under the “Data” menu of the maTEM GUI.

Finally, the menu “Section” allows the user to import the results of the interpretation from an inv-file and to build interpretative cross-sections. The locations of the interpreted soundings are displayed in the “Survey Map” GUI (open squares in Fig. 8), where the user may design the section by selecting the mode of presentation (3D or 4D) and pointing and clicking on the desired sounding locations. When in three-dimensional (4D) display mode, sections can be overlaid (multiple sections can be displayed on the same graph) as per Fig 9; they can also be freely rotated (menu “Rotate-3D”) in Fig. 9, and redrawn with a variety of different colour schemes (menu “ImageColors” in Fig. 9).

## 5. Epilogue

At the present stage of development maTEM provides a relatively broad and functional range of tools for the analysis of any type of TEM data (single loop, coincident loop and offset configurations). Overall, it offers a decent and in several aspects advanced and versatile means of treating TEM data, and may even stand competitively against commercial analysis programs.

Nevertheless, one can think many additional tools that can be included in future releases. One can think of several improvements in the code, addition of more advanced processing techniques and graphical utilities etc. One important development may be the incorporation of a means to model data with induced polarization effects. This would certainly increase the range of applications for the benefit of the user.

Another very important point, as well as one for which very few remedies exist, is the joint (mutually constrained) interpretation of different geophysical prospecting methods. It is common knowledge that every individual electrical or electromagnetic prospecting method is subject to physical limitations that restrain its scope of application. It is also common knowledge that two or more methods jointly applied and interpreted, are much more effective because they may compensate for each other's limitations. However, the geophysical software community is reacting slowly to the dearth of software solutions, at least in the public domain.

Em1div is one shiny exception: it is a pioneering application and very handy in this respect, because in addition to any kind of TEM loop configuration, it may handle dc resistivity data (Schlumberger and Wenner arrays), controlled-source frequency domain EM (FEM) and natural field EM (AMT/MT) sounding data in any combination. maTEM has an architecture readily expandable to incorporate any number of additional data analysis processes (different prospecting methods) and thus can be expanded to meet the challenge of mutually constrained geophysical data analysis with relatively minor programming effort.

Finally, maTEM may be available from the author upon request, subject to the applicant having secured a license to use em1div from its own authors. For an application to use em1div see <http://www.geofysiksamarbejdet.au.dk/page559.aspx>.

## 6. Acknowledgments

The author acknowledges limited support by the “Kapodistrias Programme” of the *Research Grants Secretariat*, National and Kapodistrian University of Athens. Ms Eleftheria Drosopoulou has *significantly* assisted in the development of this program, as part of her Postgraduate Dissertation at the Department of Geophysics and Geothermy, NKUA.

## 7. References

- Anderson, W.L., 1982a. Calculation of transient soundings for a coincident loop system (Program TCOLOOP). *USGS Open-File Report 82-378*.
- Anderson, W.L., 1982b. Nonlinear least-squares inversion of transient soundings for a coincident loop system (Program NLSTCO), *USGS Open-File Report 82-1064*
- Anderson, W.L., 1982c. Nonlinear least-squares inversion of transient soundings for a central induction loop system (Program NLSTCI), *USGS Open-File Report 82-1129*
- Auken E. and Christiansen A.V. 2004. Layered and laterally constrained 2D inversion of resistivity data. *Geophysics* 69, 752–761.

- Auken, E., Christiansen, A.V., Jacobsen, B.H., Foged, N. and Sørensen, K.I., 2005. Piecewise 1D laterally constrained inversion of resistivity data, *Geophysical Prospecting*, 53, 497–506.
- Barsukov, P.O., Fainberg, E.B. and Khabensky, E.O., 2007. Shallow investigations by TEM-FAST technique: Methodology and examples, in V.V. Spichak (editor), *Methods in Geochemistry and Geophysics, Volume 40*, 56-77.
- Christiansen, A. V. and Auken, E., 2004: Optimizing a layered and laterally constrained 2D inversion of resistivity data using Broyden's update and 1D derivatives. *Journal of Applied Geophysics*, 56, 247-261.
- Ingeman-Nielsen, T. and Baumgartner, F. 2006. CR1Dmod: A Matlab program to model 1D complex resistivity effects in electrical and electromagnetic surveys. *Computers & Geosciences*, 32, 1411–1419
- Karmis, P., 2003. Automated interpretation of transient electromagnetic soundings. Doctoral Dissertation, N.K University of Athens (in Greek).
- Tartaras, E., Zhdanov, M.S. Wada, K. Saito, A. and Hara, T., 2000. Fast Imaging of TDEM data based on S-inversion. *Journal of Applied Geophysics*, 43, 15–32.
- Wahba, G., 1990. Spline models for observational data. In: *CBMS-NSF Regional Conference Series in Applied Mathematics* 59, Soc. Industrial and Applied Mathematics, Philadelphia, Pennsylvania (1990), p. 169.