Novel technologies for greenfield exploration
Edited by Pertti Sarala
Geological Survey of Finland, Special Paper 57, 173–184, 2015

ARJUNGUI: 2D MODELLING AND INVERSION SOFTWARE FOR AIRBORNE TIME-DOMAIN EM DATA

by
Markku Pirttijärvi[1] and Heikki Salmirinne[2]

Pirttijärvi, M. & Salmirinne, H. 2015. ArjunGUI: 2D modelling and inversion software for airborne time-domain EM data. *Geological Survey of Finland, Special Paper 57,* 173–184, 11 figures.

ArjunGUI is a graphical user interface (GUI) for ArjunAir, a computer program for the modelling and interpretation of geophysical airborne electromagnetic (AEM) data from a single profile line using a two-dimensional (2D) model of electrical resistivity and magnetic susceptibility. ArjunAir was originally developed by Drs Glenn Wilson, Art Raiche and Fred Sugeng for the CSIRO/Amira consortia (project P223F). It became public domain in 2010 and is available for download from: http://p223suite.sourceforge.net/. The modelling method of ArjunAir is referred to as 2.5-dimensional, because the model is two-dimensional and the (dipolar) source–receiver system is three-dimensional. In forward modelling, the topography is flat and the flight altitude is constant. In inverse modelling, however, both the topography and variations in flight altitude are taken into account.

The inversion uses either an unconstrained (SVD) or a constrained (Occam) inversion method to update the model resistivity values so that the data error, i.e., the difference between the measured and the computed data, is minimized. In constrained Occam inversion, the roughness of the model, i.e., the variation in resistivity between neighbouring points, is minimized together with the data error. Occam inversion yields smoothly varying resistivity models. Manual editing, however, allows the creation of more realistic models with sharp conductivity boundaries. The resistivity (and susceptibility) values and the fix/free status of the cells can be manually edited to incorporate and fix known information (*a priori* data) in the model. Binary (on/off) weights can be assigned to individual data points, stations or channels to include them in or exclude them from the inversion.

ArjunGUI was developed for the rapid inversion of airborne TEM anomalies caused by conductive targets of possible economic interest. This paper discusses the background to ArjunGUI, presents the main components of the graphical user interface, and provides a modelling and inversion example and comparison with 1D inversion.

Keywords (GeoRef Thesaurus, AGI): computer programs, geophysical methods, airborne methods, electromagnetic methods, two-dimensional models, inverse problem, interpretation

[1] *University of Oulu, Department of Physics, P.O. Box 3000, FI-90014 University of Oulu*
E-mail: markku.pirttijarvi@gmail.com
[2] *Geological Survey of Finland, P.O. Box 77, FI-96101 Rovaniemi, Finland*
E-mail: heikki.salmirinne@gtk.fi

INTRODUCTION

Airborne time-domain electromagnetic (TEM) methods are widely used in mineral prospecting to locate and delineate targets of enhanced electrical conductivity. In TEM methods, an EM pulse is generated by abruptly switching off or on the electric current flowing through a large wire loop. The electromotive force due the time-changing current induces currents inside conductive earth (Faraday's law). These currents, in turn, produce a secondary EM field (Ampére's law), which is measured by a receiver loop or coil (dB/dt response) or (flux-gate, Cs or SQUID) magnetometer (B-field response) as a function of time. The more conductive the earth is, the stronger is the EM response at later time channels. In TEM methods, the measurement time typically ranges from a few microseconds (near-surface investigations) to a few seconds (deep EM studies).

One of the benefits of TEM methods compared with frequency-domain methods, in which a time harmonic (sinusoidal) current waveform is used, is the higher signal-to-noise ratio (SNR) due to the absence of the primary EM field, because the measurements are made off-time, i.e., after the primary EM field has vanished. The low noise level, together with the large transmitter moment, gives an increased depth of exploration. In TEM measurements, the depth of exploration can reach few hundreds of metres.

Although airborne measurements enable the investigation of large areas cost-effectively, they produce enormous amounts of data. The data are routinely processed using one-dimensional (1D) inversion or apparent resistivity–depth transformation. The results for each site are then combined side by side to yield conductivity depth images (CDI) or resistivity pseudo-sections. Due to the 1D model and the offset between the transmitter and the receiver, the pseudo-sections give incorrect and misleading conductivity images for 2D and 3D targets. Unfortunately, 2D and 3D modelling and inversion of TEM data are very time-consuming and require either expensive special software or services provided by commercial companies. Practical interpretation software for TEM data is needed.

ArjunAir is a computer program for the modelling and interpretation of geophysical airborne electromagnetic (AEM) data from a single profile using a two-dimensional (2D) model of electrical resistivity and magnetic susceptibility. ArjunAir was originally developed by Drs Glenn Wilson, Art Raiche and Fred Sugeng for the CSIRO/Amira consortia in project P223F (Wilson et al. 2006). Amira software, including ArjunAir, became public domain in 2010 (http://p223suite.sourceforge.net/). ArjunAir modelling is based on the finite element method. The method is referred to as 2.5-dimensional, because the model is two-dimensional and the (dipolar) source–receiver system is three-dimensional. ArjunAir is a stand-alone solver program for the EM modelling problem that uses text files for data input and output. After the Amira programs became public domain, the original graphical user interface (GUI), EMGui, developed in the P233F project, was made obsolete by Maxwell, commercial software by ElectroMagnetic Imaging Technology Ltd (EMIT) (http://www.electromag.com.au/).

ArjunGUI

ArjunGUI is a new graphical user interface (GUI) for ArjunAir developed for the participants of the NovTecEx project at the University of Oulu by Dr Markku Pirttijärvi. It is written in Fortran90 and uses the DISLIN graphics library for both the GUI and the graphics. Presently, ArjunGUI supports only time-domain (transient) EM data and measurement systems. Because the focus is on airborne applications, the source loop is approximated as a magnetic dipole, i.e., the actual size and shape of the transmitter loop are not taken into account. The receiver is either a small loop or a coil measuring the time derivative of the magnetic flux density dB/dt (T/s) or a magnetometer (flux-gate or squid) measuring the intensity of the magnetic flux density B (T). The response is the vertical (Z) component or the horizontal (X) component along the flight line, or both components.

ArjunGUI was primarily developed for inversion, but it can also be used for forward modelling. In forward modelling, the topography is flat and the flight altitude is constant. In inverse modelling, both the topography and the varying flight altitude can be taken into account. Inversion uses either an unconstrained (SVD) or a constrained (Occam) inversion method to update the model resistivity values so that the data error, i.e., the difference between the measured and the computed data, is minimized. In constrained inversion, the roughness of the model, i.e. the variation in resistivity between neighbouring points, is minimized together with the data error. Occam inversion produces a smoothly varying resistivity model if a large number of the cells are free to change. The initial resistivity (and susceptibility) values and the fix/free status of the parameters can be manually edited to construct models with step-wise conductivity changes and to incorporate *a priori* data in the models. Weights can be assigned to individual data points, stations or channels to either include them in or exclude them from the inversion.

ArjunGUI uses a modified version of ArjunAir developed by Drs Glenn Wilson, Art Raiche and Fred Sugeng for the CSIRO/Amira consortia (project P223F). The original source code in Fortran language (version 7.0.5) is available at: http://p223suite.sourceforge.net/. The main modifications are as follows: 1) a global module is used to pass large allocatable arrays between subroutines; 2) the unnecessary temporary file I/O is replaced with global arrays; 3) the sensitivity matrix (Jacobian) is trimmed by removing fixed resistivity cells and zero-weighted data; 4) the original SVD inversion method was replaced with a) unconstrained SVD-based inversion with adaptive damping, b) constrained (Occam) inversion with a (faster) iterative conjugate gradient solver and c) constrained Occam inversion using a (slower) SVD solver; 5) a wider and more carefully positioned mesh is extended on all four sides of the user-defined mesh; 6) old results can also be used in inversion, because both the frequency domain spectrum (*.FRQ) and the Jacobian (*.JCB) are saved after inversion; and 7) OpenMP parallelisation is used in the 64-bit version of ArjunAir (requires libiomp5md.dll file). The SVD algorithm is adapted from Press et al. (1988) and suits both under- and over-determined problems. The unconstrained SVD parameter optimization is based on a linearized inversion method in which singular value decomposition (SVD) with adaptive damping is used (Pirttijärvi 2003). The constrained Occam inversion algorithm is derived from GRABLOX2 gravity interpretation and modelling software using a 3D block model (Pirttijärvi 2011).

ArjunGUI was written in Fortran90 and compiled with Intel Fortran 15 (Intel Visual Fortran Composer XE). The graphical user interface is based on the DISLIN graphics library (version 10.4) by Helmut Michels (http://www.dislin.de). Since the DISLIN graphics library is available for other operating systems (Linux, Mac), ArjunGUI could be compiled and run on other operating systems without major modifications.

ArjunGUI USER INTERFACE

ArjunGUI consists of the Main GUI window presented in Figure 1 and nine separate GUI windows for special tasks, which are: 1) DATA for data input and pre-processing (with XYZ file support) (Fig. 2), 2) SYSTEM for setting system parameters (Fig. 3), 3) TIMES for defining time channels, 4) WAVES for defining the (pulse) waveform, 5) MODEL for model creation, visualization and editing (Fig. 4), 6) COMPUTE for forward and inverse computation (Fig. 5), and 7) RESULTS for the visualization of results as profile (and sounding) graphs and editing of data weights (Fig. 6). The GUI windows are activated by pressing the corresponding push buttons in the Main GUI (cf. Fig. 1).

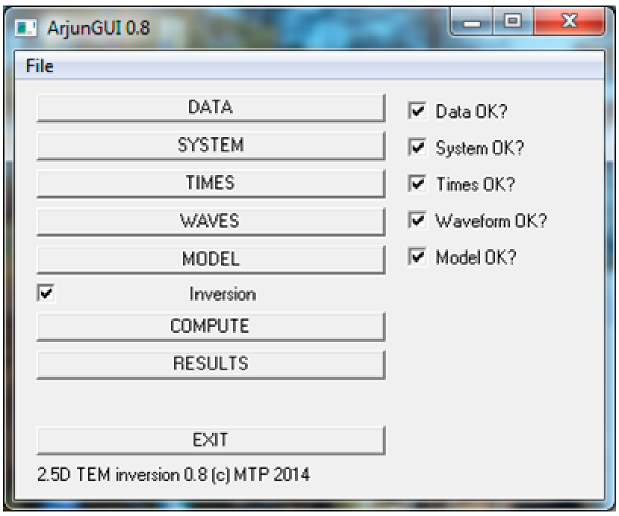

Fig. 1. ArjunGUI Main GUI window.

To perform modelling or inversion, the user needs to take care of the following prerequisites:

1) measured data must be read in or a profile defined for forward modelling,
2) the system, time channel and waveform parameters must be properly set, and
3) an appropriate 2D model must be defined.

The current status of the modelling problem can be stored in and read from a *.AGP project file. The project file contains all relevant information on the modelling problem, except for the measured data. On exit, the program automatically saves the current status of the modelling work into an AGUI.AGP project file, which is then read in the next time the program is started.

The Data GUI window is shown in Figure 2. The main purpose of the Data GUI is to:

1) visualize the TEM response as a function of profile distance,
2) select and cut smaller anomalous parts of the profile data, and
3) re-sample dense data and thus reduce the dimension of the linear problem.

ArjunGUI supports the reading of data from (Geosoft) XYZ files, but uses its own data format to store the data on a single anomaly in a single profile.

The system GUI window is displayed in Figure 3. Presently, frequency-domain computations are not supported, but an option is reserved for them. The TEM response is obtained by first computing the full FEM response over a wide frequency range for various values of wave number (Ky) along the strike direction. Fourier transform is then applied once to give the full FEM response at receiver positions, and another Fourier transform is carried out for the time domain step response. The TEM response is either the time derivative of the magnetic flux density (dB/dt, T/s = V/m^2) or intensity of the magnetic flux density (B-field, T) defined either as nano (10^{-9}), pico (10^{-12}) or femto (10^{-15}) T/s or nT, pT, fT in the case of B-field data.

The *Normalize with NIA* check box is needed to rescale measured dB/dt data if they have been normalized with the effective moment of the transmitter loop (NIA). By multiplying the data by the number of transmitter loop turns (N), the current (I) and the surface area of the loop (A), measured data defined as V/(Am^4) will become V/m^2 = T/s used in ArjunAir for the dB/dt response. The system parameters can be stored in and read from an AGS file.

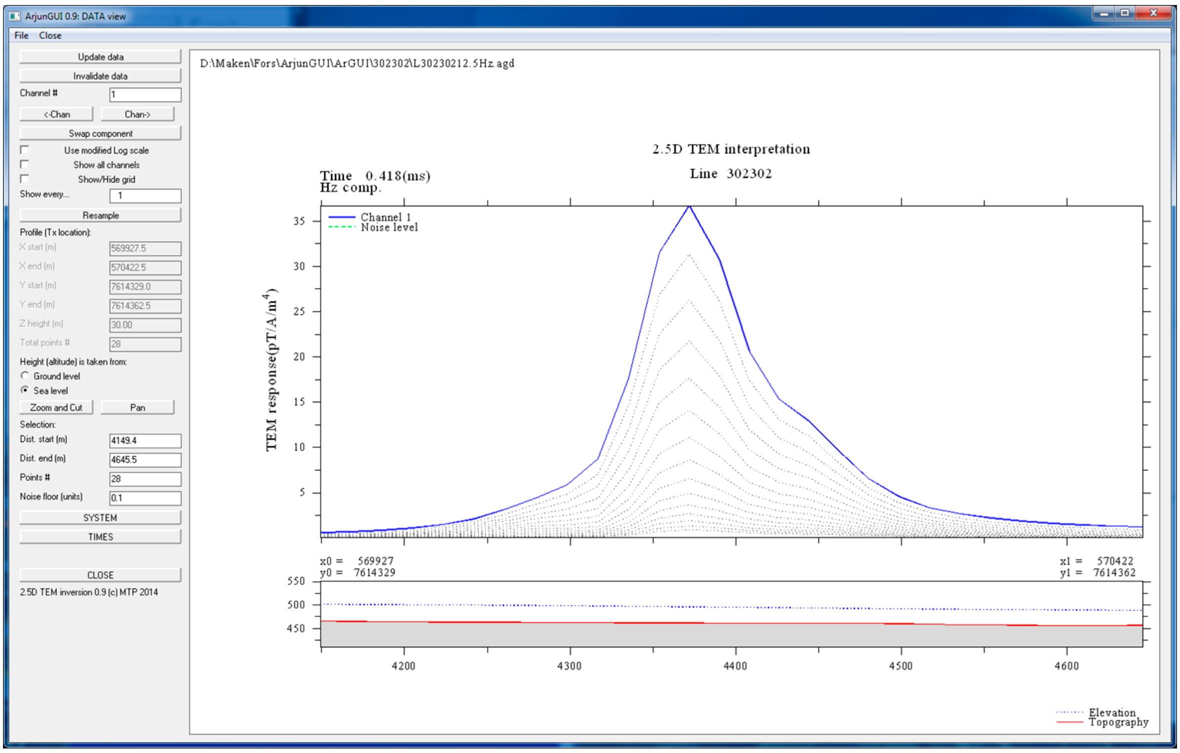

Fig. 2. ArjunGUI Data GUI for data input and pre-processing.

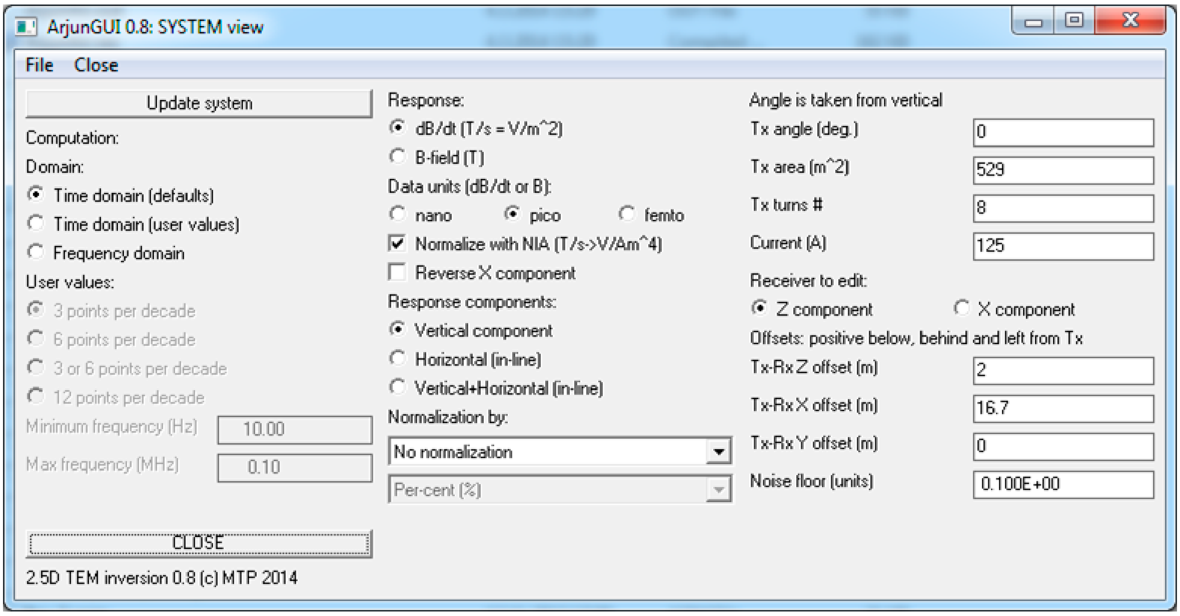

Fig. 3. ArjunGUI System GUI for setting the system parameters.

The Model GUI, illustrated in Figure 4, is perhaps the most advanced GUI window of the ArjunGUI application. In addition to visualization of the model, the Model GUI enables the creation of the initial 2D model mesh and interactive editing of the mesh nodes, parameter values and fix/free status of the cells. The graph area shows a vertical cross-section of the 2D resistivity (or susceptibility) model. The colours represent the 10-base logarithm of the resistivity (or susceptibility). The locations of TEM measurement points are indicated by small open circles above the model graph.

The *Generate initial model* button is used to create a totally new initial model based on the measured data or synthetic profile defined in the Data GUI and to redo the current model with new

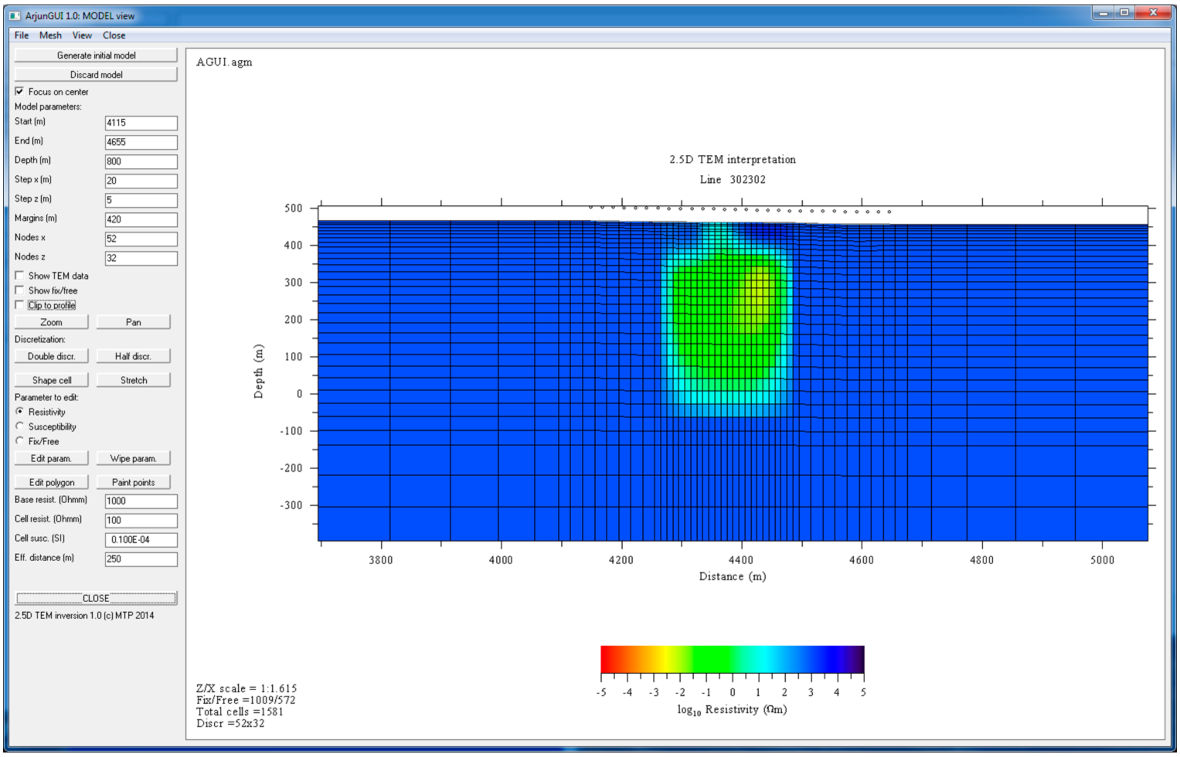

Fig. 4. ArjunGUI Model GUI for model creation, visualization and editing.

settings. Pressing the button will not only affect the model position and discretization but also resets any existing resistivity (and susceptibility) to its initial value. The initial value of magnetic susceptibility is always $k = 0$ SI, i.e., the host medium is considered non-magnetic. The initial resistivity value is equal to the user-given background value in forward modelling. In data interpretation, however, a special transformation method is used to compute the initial resistivity distribution based on the measured TEM data in a manner similar to conductivity-depth transformation. This aims to reduce the total number of inverse iterations needed for a good fit between the measured and computed TEM data.

The initial model consists of quasi-rectangular elements or cells defined by the four neighbouring corner nodes. The beginning and the end of the model are based on the start and end of the data profile (in both forward and inverse modelling). The nominal width and nominal height of the cells are based on the mean data sampling. The nominal size of the elements should depend on the resistivity of the model. The more conductive the model is, the smaller the size of the elements should be. The width of the elements increases inside the margin areas, which are used to model the decaying behaviour of the EM fields more accurately outside the profile. In addition to the visible user-defined mesh, ArjunAir automatically adds 11 layers for the air above the model and an additional 7 columns and rows of increasing width and height on both sides and at the bottom of the model. Thus, the actual computational mesh extends ca. 10–20 km away from the sides of the mesh visible to the user.

The height of the cells increases downwards by a factor of one. Thus, if DZ is the height of the topmost element, the height of the elements increases as: DZ, 2×DZ, 3×DZ, 4×DZ, etc. The topography along the profile is taken into account by varying the height of the elements so that the bottom of the model becomes horizontally flat. The depth to the bottom of the model is a user-defined parameter and it should be at least 500 m or about twice the profile length.

After creating the initial model, the user can interactively edit the geometry of the cell nodes, the resistivity and susceptibility values and the fix/free status of the cells. Three different parameter-editing modes are available: rectangular area, polygonal area and smooth point insertion. The model can be stored in and read from a *.AGM file. The model resistivity and susceptibility values (and their 10-base logarithm) and fix/free status can also be outputted to a general purpose *.DAT file to be used to visualize the resistivity sections in third-party software (e.g. GoCAD).

The Compute GUI, illustrated in Figure 5, consists of a single column of GUI widgets. Depending on the status of the *Inversion* check box in the Main (or Results) GUI, the push button at the top of the Compute GUI reads either *START COMPUTATION* or *START INVERSION*. The *Unconstrained (SVD)* inversion method should only be used when a model with a rather good fit already exists and additional iterations are needed to create as good a fit as possible. Alternatively, it can be used when an initial model exists and the fix/free status of the cells has been edited so that the inversion will optimize the resistivity of the elements of the body or its neighbouring cells only. *Constrained (Occam+CG)* inversion is the default method, which produces rather smooth models because it aims to minimize the model roughness together with data misfit. It uses an iterative conjugate gradient (CG) solver for the linear equations. *Constrained (Occam+SVD)* inversion provides the benefits of Occam's method and the stable SVD method in solving the linear equations. It can be very slow if the number of free cells is large, but it produces a good fit and the models that are not as smooth as in CG-based inversion. *Constrained (Occam+SVD)* inversion should be used when making the last iterations in a model found out using *Constrained (Occam+CG)* inversion.

When the *START COMPUTATION* button is pressed, ArjunGUI creates the ArjunAir.cfl control file and ArjunAir.res resistivity and susceptibility distribution file, and calls for ArjunAir.exe (32 or 64 bit version) to perform the computation. When inversion is started, an additional ArjunAir.inv file is used to pass the inversion parameters to the ArjunAir program. The original model resistivity definition based on a fixed number of lithology codes defined in the control file is replaced by the use of a separate *.RES file, which allows the resistivity values to change arbitrarily from one cell to another. The progress of the computation can be followed from the console window. After the computation ends, ArjunGUI reads the results from ArjunAir.mf1 (forward computation) or ArjunAir.mv1 and ArjunAir.res files (inverse computation) and closes the Compute GUI. Typically, the forward computation takes a few minutes or tens

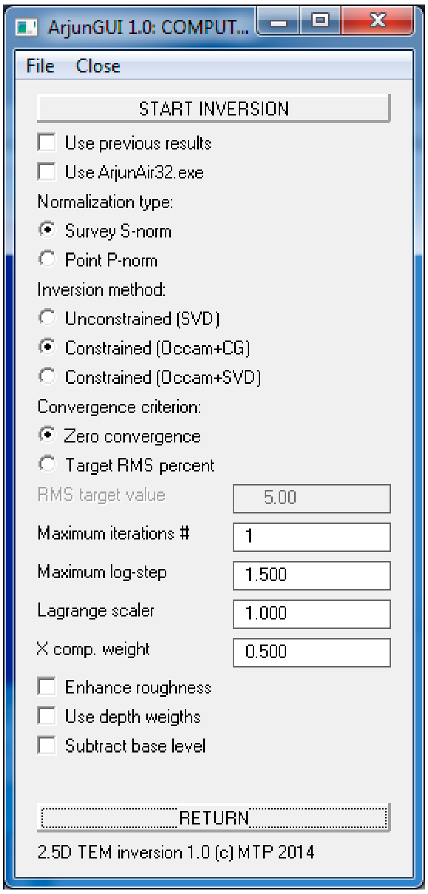

Fig. 5. ArjunGUI Compute GUI for forward and (in this case) inverse computation.

of minutes to compute. Because the sensitivity matrix (Jacobian) is obtained almost as a side product of the forward computation, the inversion itself is relatively fast and most of the time is spend in the forward computations.

The Results GUI in Figure 6 presents the computed response in the upper graph so that the active time channel is drawn as a solid blue line and other channels are shown by dotted curves. If data have been read in, the measured TEM data of the active time channel are drawn as small dots. The RMS error of the current channel and the global RMS error are shown in the lower left corner of the results graph. The lower graph shows either the topography and the flight path or the resistivity cross-section. The Results GUI is also used for the manual editing of data weights (*Edit null weight* and *Edit full weight* buttons) and removal of the whole time channel from the inversion (*Active/Inactive* check box). The computed (and measured) data can be saved to a *.AGD file for later analysis. The *Pick sounding site* button brings up the Sounding GUI, where the TEM response is plotted as a sounding curve, i.e., as a function of time per single receiver site, and the data can be saved to a *.AGZ file in sounding format.

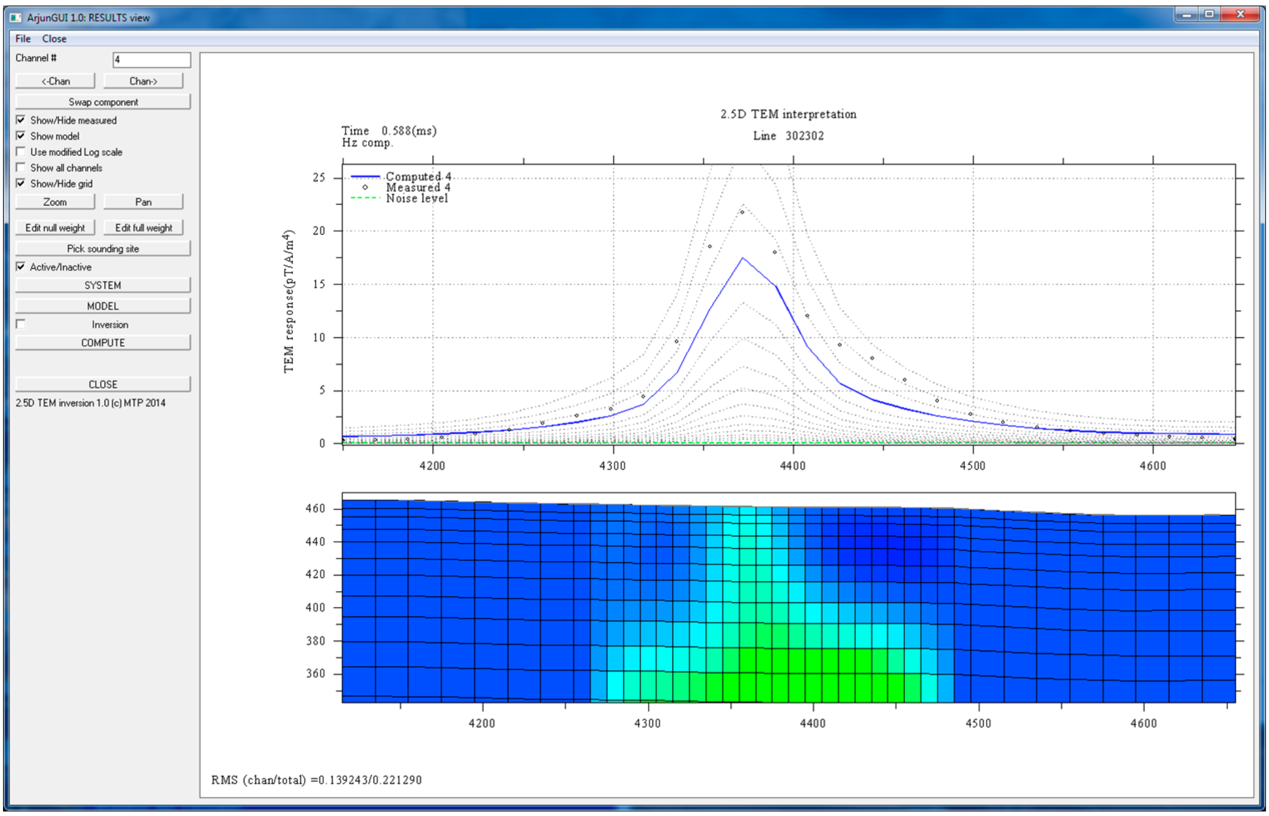

Fig. 6. ArjunGUI Results GUI for visualization of results as profile (and sounding) graphs and editing of data weights. The model is the same as in Fig. 4 (1:1 depth scale).

EXAMPLE

The data in the example were measured with the Danish Skytem system in 2012 in Enontekiö, Northern Finland. The resampled data around a single anomaly are presented in Figure 2. The system parameters are shown in Figure 3. Only the Z component is used in this example. The high moment data were measured using a 12.5 Hz base frequency. The length of the pulse was about 8.2 ms and the 15 time channels ranged from 0.4 to 5 ms. After setting the noise level at 0.1 pT/s, the total number of data points was 420.

One of the models obtained during the inversion is presented above in Figure 4. The model, the discretization of which is focused on the central part of the profile, has 1581 cells, of which 572 are free for optimization. The initial model, which is shown in Figure 7, is that of a homogeneous half-space with a constant background resistivity of 1000 Ωm. The discretization was manually doubled below the main anomaly. The fix/free status was set to free below the central part of the profile down to a depth of about 500 m. The model in Figure 4 was obtained after five Occam inverse iterations with a Lagrange value of 0.5 and base level removal. Single iteration took about 10 minutes to compute on a PC with a 3.3 GHz Intel i5-2500K CPU. The fit between the measured and computed data for the model in Figure 4 is illustrated in Figure 6 for time channel #4 (0.588 ms). The initial RMS error was about 40%. The total RMS error in Figure 4 is about 22%.

After the first five iterations, the discretization was doubled over the main target. The new model has 3657 cells, of which 1920 are free for optimization. The model obtained after eight additional iterations and the fit between the measured and computed data are presented in Figures 8 and 9. A single iteration took about 45 minutes to compute, so the inversion was performed overnight. The RMS error is about 12%, which can be considered as a moderately good value for the fit.

The inversion reveals that the main source of the TEM anomaly is 100–150 m wide and is located at a depth of 50–100 m. The true depth extent is difficult to assess without additional testing, but according to the model Figure 8 it is about 300 m deep. The resistivity of the main target is less than 10^{-4} Ωm, indicating that it is very conductive and possibly due to graphitic black schists. The conductivity and shape of the target are also difficult to assess without additional modelling tests.

The user should remember that the inversion results must not be taken as a true representation of the subsurface conductivity. Because of the inherent ambiguity of the inversion, other possible models can fit the measured data equally well.

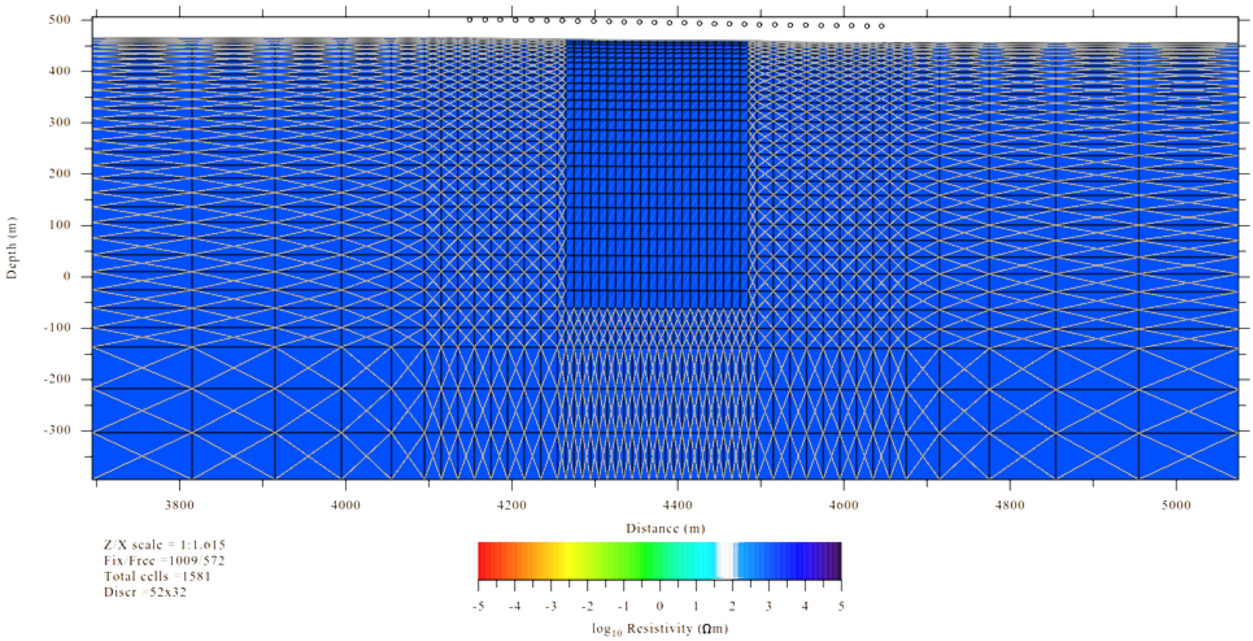

Fig. 7. The initial model used in the interpretation of the anomaly on line 302302. The cells marked with a cross are fixed in the inversion.

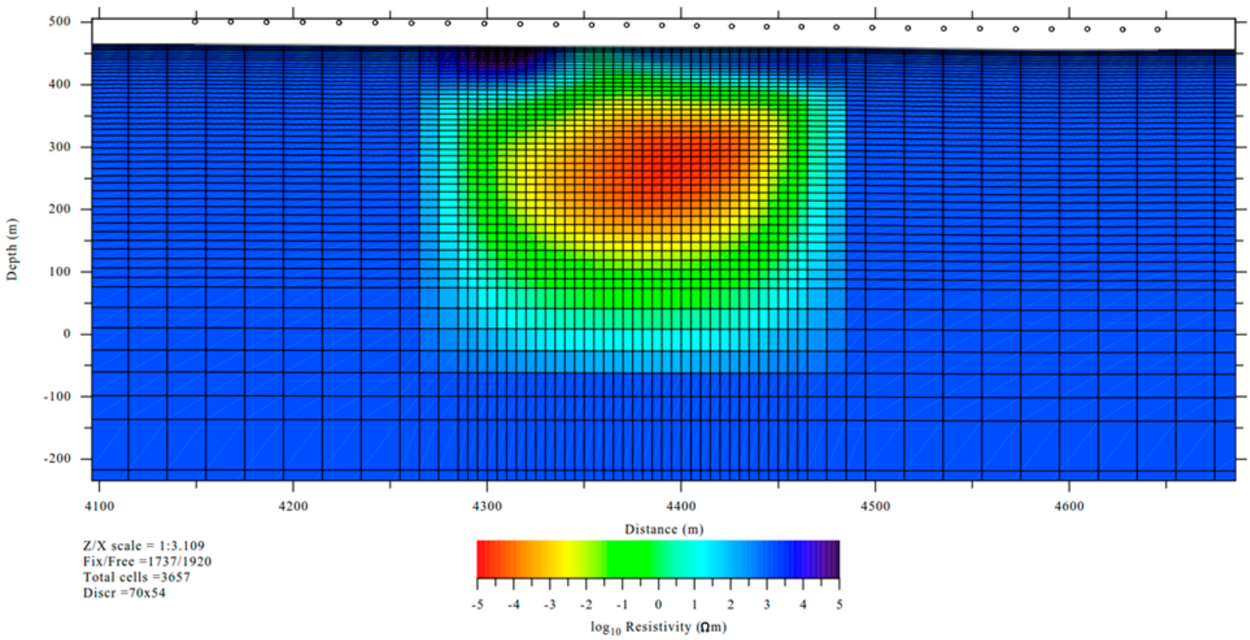

Fig. 8. The resistivity model for the anomaly on line 302302 after 13 iterations. The vertical axis is not in scale with the horizontal axis.

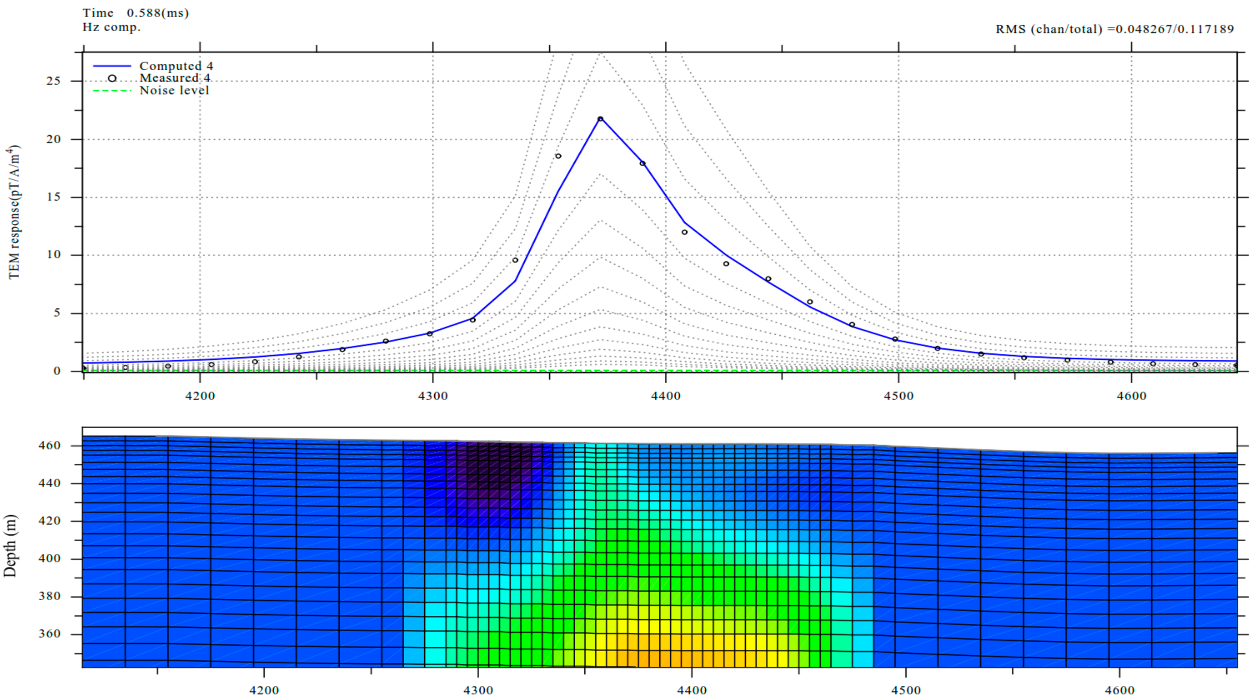

Fig. 9. The fit between measured and computed data for the anomaly on line 302302 after ten iterations and a detailed view of the upper part of the model (1:1 depth scale).

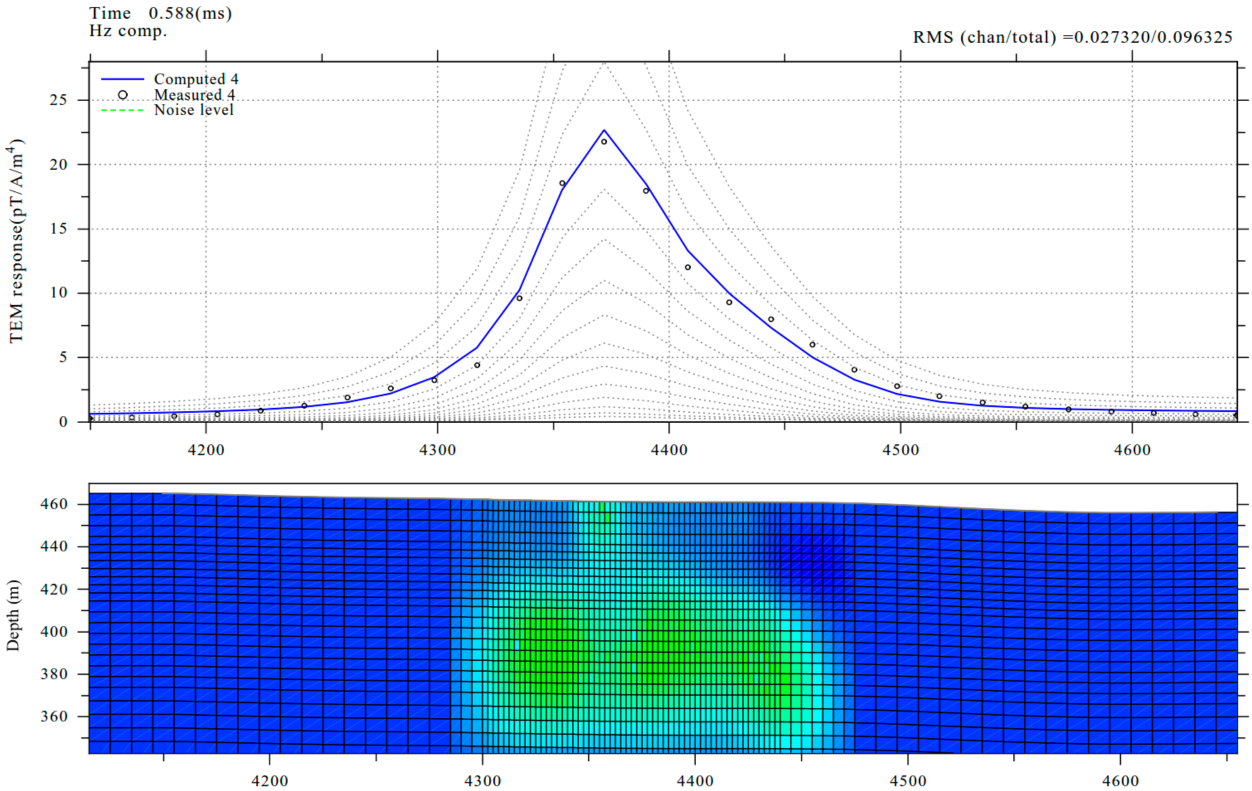

Fig. 10. The fit between measured and computed data for the anomaly on line 302302 and a detailed view of the upper part of the model (1:1 depth scale). The bottom of the model was constrained to the depth level of +330 m (a.s.l.). The colour scale is the same as in Figs. 8 and 9.

The model resulting from the linearized inversion method used in ArjunAir is affected by the starting model. A different initial model or different constraints imposed on the model will produce different results, as also will an alternative set-up of the fix/free status.

Figure 10 provides the inversion results based on a different initial model and inversion strategy. The main assumption was that the target is located closer to the surface. Therefore, the bottom of the free cells was limited to the depth of about +330 m above sea level. As before, the discretization was increased and the width of the area of free cells was increased during the inversion. The model consists of 3910 cells, of which 1716 were free in the inversion. The "final" RMS error after 20 iterations is 9.6%. The resistivity of the target is higher (c. 1 Ωm) than in Figures 8 and 9 because the conductor is closer to the surface. The peak TEM anomaly is due to a small outcropping conductor, whereas the generally horizontal main target is 150 m wide and is located at a depth of 40 m. According to this modelling, the bottom of the conductor is at a depth of about 110 m (+350 m above sea level).

Finally, to demonstrate the benefits of using a 2D model, the ArjunGUI results are compared with the 1D inversion results provided by SkyTEM. Figure 11 presents the 1D inversion results for the whole length of profile 302302. The black dashed rectangle delineates the part of the profile (X=569930-570420) used in the ArjunGUI examples (cf. Fig. 2). The 1D inversion is carried out point-by-point using a conductive horizontally layered earth model and lateral constraining. If the true geo-electrical structure is not horizontally layered, the stitched pseudo-section (e.g. Fig. 11) gives misleading results. Interpretation of an isolated 2D and 3D conductivity target using a 1D model tends to generate a so-called "pant-leg effect", where the conductor appears to be divided into two branches of a hyperbola extending to great depth. The 2D inversion results in Figures 8–10 are more realistic and demonstrate that the depth extent is likely to be limited. All in all, ArjunGUI and ArjunAir provide a means for practical and more accurate interpretation of airborne TEM data using a 2D model.

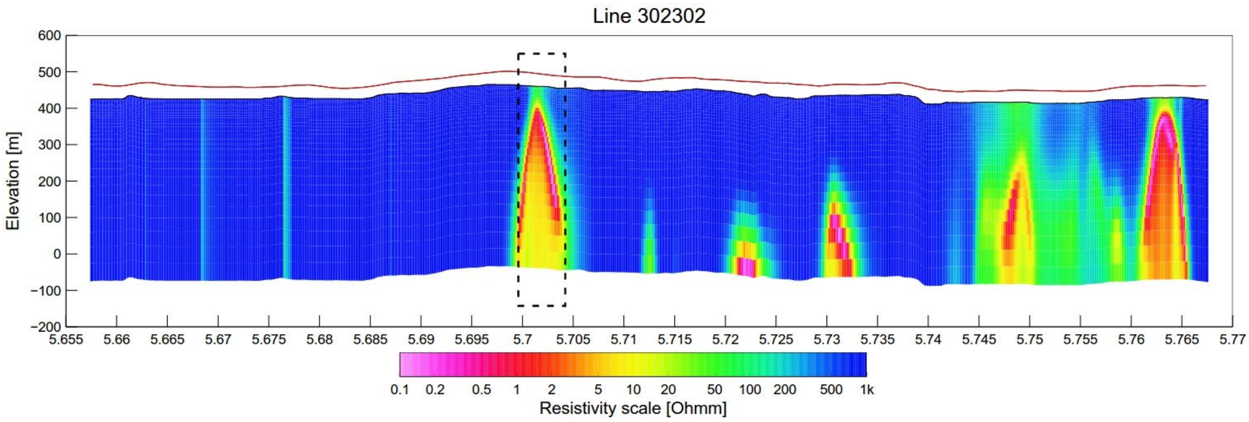

Fig. 11. SkyTEM's 1D inversion results from line 302302. The black dashed line delineates the part of profile between coordinates X=569930-570420 shown in Fig. 2 and Figs. 6, 9 and 10.

CONCLUSIONS

ArjunGUI is a new graphical user interface for ArjunAir, a computer program for the modelling and interpretation of geophysical airborne EM data using a two-dimensional (2D) model. The 2D model is a good compromise between a 1D layered earth model and a full 3D model. On one hand, the 2D model allows better interpretation of the true shape of complicated geological targets than stitched 1D models commonly used in conductivity-depth imaging. On the other hand, 2D modelling is computationally less demanding than 3D modelling. In addition, 2D models are easier to visualize and edit than 3D models.

The original ArjunAir (v. 7.0.5) program was modified for so-called Occam inversion, where the model roughness is minimized together with the data misfit. Occam inversion aims to derive the least complicated model that explains the data. The inversion results, however, should never be taken as a true representation of the subsurface conductivity. Because of the ambiguity of the inversion, other possible models can fit the measured data equally well. Different initial model or different constraints imposed on the model will produce different results, as will an alternative setup of the fix/free status. *A priori* information from drill-holes, for example, can be used to fix the resistivity in a certain part of the model to reduce the ambiguity.

Interpretation of TEM data is a process where the user needs to test different hypotheses regarding the most important properties of the target, such as the conductivity, the depth to the top, the depth extent, the shape and dimensions or dip angle and width. The models obtained from Occam inversion are usually too smooth to be geologically realistic. The interpreter should try to distinguish common features between different models and use this information as a starting point for a new, more rugged initial model. ArjunGUI helps this task and enables modelling of the EM responses of complicated geological structures. The inversion results and resistivity models can be saved in text files that can be imported into third-party 3D visualization and CAD software. Despite the improvements made to the computational method of ArjunAir, the overall inversion process is still time-consuming and practice is needed to learn the optimal inversion strategy and to assess the viability of the models.

Presently, ArjunGUI can handle only airborne TEM systems. Support for ground systems and frequency-domain EM systems would enable two-dimensional interpretation of GTK airborne EM data (multiple frequencies) and Sampo EM sounding data in the future. Combined inversion of EM and static magnetic field data would help in better constraining the inversion and estimating the shape and dimensions of the targets having both conductive and magnetic characteristics.

ACKNOWLEDGEMENTS

This work was financed by the Finnish Funding Agency for Innovation (TEKES) through the NovTecEx project (Green Mining programme). ArjunGUI version 1.0 is freeware and it can be downloaded from: https://wiki.oulu.fi/x/pIIrAw.

REFERENCES

Pirttijärvi, M. 2003. Numerical modeling and inversion of geophysical electromagnetic measurements using a thin plate model. PhD thesis, Acta Univ. Oul. A403, University of Oulu. Available at: http://urn.fi/urn:isbn:951427119X

Pirttijärvi, M. 2014. Grablox2 – Gravity interpretation and modeling software based on 3-D block models; User's guide to version 2.1. University of Oulu. Available at: https://wiki.oulu.fi/x/jYU7AQ

Press, W. H., Flannery, B. P., Teukolsky, S. A. & Vetterling, W. T. 1988. Numerical Recipes, The art of scientific computing. Cambridge University Press.

Wilson, G. A., Raiche, A. P. & Sugeng, F. 2006. 2.5D inversion of airborne electromagnetic data. Expl. Geoph. 37, 363–371.