Guide for use of online materials associated with the book
Resistivity and Induced Polarization: Theory and Applications to the Near-Surface Earth

Andrew Binley
January 2021

1. Introduction

In preparing the book, one thing that we were keen to include was a set of modelling tools that could allow the reader to gain some familiarity with forward and inverse modelling, and also experiment with example datasets covered in the book. We wanted to provide a modelling platform that is easy to use, but also one that offers a full range of features thus allowing the reader to analyse their own datasets. The first author's forward and inverse modelling codes (the R family of codes) have been available for some time and have been used in a wide range of studies. These codes have no graphical interface and thus their use can be challenging to some users. Our vision was to have a graphical interface that utilised these codes.

Guillaume Blanchy (Lancaster), assisted by Sina Saneiyan (Rutgers), Jimmy Boyd (Lancaster) and Paul McLachlan (Lancaster), took on the challenge to develop such an interface, which resulted in ResIPy – a modelling environment that permits forward and inverse modelling of resistivity and IP data. ResIP, written in Python, and available as open source, can be used as a graphical interface but can also be used as an application programming interface (API), e.g. within Jupyter notebooks, which can be effective for scripting tasks and also sharing results.

In this set of online materials to accompany the book, we provide a copy of ResIPy, a number of datasets covered in the book, some video tutorial on the use of ResIPy's graphical interface and some example Jupyter notebooks. Although ResIPy is effectively a harness for the author's R family of codes, we provide here some additional datasets from the book setup for use with these codes outside of the ResIPy environment to give the reader a broad range of modelling solutions.

Working with the example datasets should allow the reader to experiment with the codes and explore different modelling options. These will help the reader to develop sufficient experience in modelling before embarking on analysis of their own datasets. The codes continue to be developed and it is recommended that the user accesses the most up to date versions. However, in order to ensure compatibility with example input files provided here, a specific version of each code is provided.

As explained in the Appendix of the book, there is a vast range of software tools available. Appendix 1 of this document lists the codes that the author is aware of, along with references and web links. Some of these are commercial codes (i.e. paid for), some are available as open source, whilst some are only available in executable form.

2. Summary of online materials

A set of five folders are provided, the contents of which are summarised below.

`ResIPy installation`
Version 2.2.3 of ResIPy (details in section 4) is provided as a zip file containing a Windows executable (see section 3 for more details of ResIPy).

`ResIPy GUI tutorials`
A set of video tutorials have been produced to guide the reader in the use of the ResIPy. The tutorials (with YouTube links) are:

- Tutorial 01 (Getting started)
- Tutorial 02 (Regularization)
- Tutorial 03 (Noise effects)
- Tutorial 04 (Cross borehole resistivity) – Figure 5.13 in the book
- Tutorial 05 (2D resistivity field data) – Figure 5.1 in the book
- Tutorial 06 (3D resistivity field data)
- Tutorial 07 (3D resistivity column data) – Figure 5.18 in the book
- Tutorial 08 (2D IP field data) – Figure 5.28 in the book
- Tutorial 09 (Inversion of time lapse resistivity data) – Figure 6.11 in the book

The folders `Tutorial 04 files` to `Tutorial 09 files` contain datasets used for tutorials 04 to 09, respectively.

`Additional examples for ResIPy application`
Four additional datasets are provided to allow the reader to analyse datasets used in the book using ResIPy. These examples are:

- Lancaster Castle 2D resistivity - Figure 6.1 in the book
- Groundwater-surface water interface 2D resistivity - Figure 6.2 in the book
- Cross borehole resistivity - Figure 6.23 in the book
- Cross borehole IP - Figure 6.19 in the book

`Jupyter Notebooks`
Four Jupyter notebooks are included to illustrate their use with ResIPy. These cover:

- An introduction to 2D resistivity forward modelling
- An introduction to 2D resistivity inverse modelling
- 3D resistivity inversion
- Time-lapse resistivity inversion

A folder `Notebook examples` contains the data files needed for these notebooks. A guide for the use of the Jupyter notebooks is also provided (`Jupyter Notebooks (brief guide).pdf`).

`R family of codes`

The R family of codes (see section 3 for details) are used by ResIPy and thus provided in that package. However, they are also included here as standalone executable codes. The versions supplied are the ones used for the example datasets provided. These are:

- R2 version 4.01
- R3t version 2.1
- cR2 version 3.0
- cR3t version 1.0

The codes are provided in individual zip files along with a user guide and example datasets. In addition two R2 example datasets and two R3t example datasets are provided that relate to material covered in the book. These are in the folders

- `R2 example datasets`
    - Synthetic 2D resistivity forward-inverse modelling - Figure 5.11
    - Groundwater-surface water interface 2D resistivity - Figure 6.2

- `R3t example datasets`
    - River terrace 3D resistivity example - Figure 5.16
    - Hollin Hill landslide 3D resistivity example - Figure 6.12

3. R family of codes

The R family of codes (http://www.es.lancs.ac.uk/people/amb/Freeware/freeware.htm) have evolved since the early 1990s and have been available (in executable form) for non-commercial use. The codes were designed to allow users, without access to budgets for commercial software, to carry out forward and inverse modelling. They do not include graphical interfaces but can utilise freely available software for mesh generation (e.g. Gmsh, http://gmsh.info) and presentation of 2D and 3D images (e.g. ParaView, https://www.paraview.org). The codes have been used in a wide range of applications (see list in the book's appendix).

The codes are based on potential field modelling using the finite element method, on structured and unstructured meshes (triangles and quadrilaterals in 2D; tetrahedra and triangular prisms in 3D). 2D analysis is done assuming 3D current flow. The modelled region is defined by the user and semi-infinite conditions are not imposed, allowing the analysis of bounded regions. Electrodes can be placed anywhere in the modelled region, e.g. on a line or plane representing the ground surface, or at depth (e.g. representing borehole arrays). Inversion is based on the L2 norm regularisation, although disconnected regularisation can be applied. In all codes robust inversion can be applied. Resistivity data are input as transfer resistances; IP data are input as transfer impedances (magnitude and phase angle). IP modelling is carried out in the frequency domain, i.e. as complex resistivity. Output from any inversion is accompanied with the cumulative sensitivity matrix allowing mapping of data coverage.

The principle code is R2. This is a 2D resistivity forward and inverse modelling program. cR2 is the complex resistivity equivalent of R2 for IP modelling. R3t is a 3D resistivity forward and inverse modelling program (the "t" suffix was added to differentiate R3t from an earlier (now obsolete) hexahedral element-based code R3). cR3t is a complex resistivity version of R3t for 3D IP modelling. All codes have similar data structure requirements. R2, cR2, R3t and cR3t form the R family of codes.

4. ResIPy

The codes described in the previous section can be executed independently as standalone software. In order to provide a more user-friendly tool for the reader to apply these codes, the ResIPy interface (https://gitlab.com/hkex/resipy) was written. Blanchy et al. (2020) describe the code structure and provide examples, illustrating its use; Boyd et al. (2019) illustrate the use of ResIPy for 3D inverse modelling. The code allows the user to carry out forward and inverse modelling of resistivity and IP data. In both cases, the user is able to define the geometry of the problem and then design a modelling mesh (2D or 3D). Unstructured meshing is done by ResIPy calls to the finite element meshing code, Gmsh (http://gmsh.info). Forward modelling can assist in survey design and allow the user to understand the sensitivity of different measurement configurations. Inverse modelling in ResIPy includes data quality checking and the construction of an error model (e.g. Figure 4.15 in the book) if reciprocal measurements are available. ResIPy includes graphical output of inversion results. Exporting of results allows use of the powerful (and freely available) ParaView environment (https://www.paraview.org) for visualisation. ResIPy is available in source code form or as a standalone executable. The former, allows the user to customise the interface, should they so wish. A standalone executable of ResIPy (for Windows, macOS or Linux) includes the R family of codes (R2, cR2, R3t, cR3t).

ResIPy is made up of three layers. The bottom layer contains the executables (R2, cR2, R3t, cR3t, along with Gmsh). A central layer is composed of the Python application programming interface (API). This interface contains a set of functions allowing the writing of input files to the executables along with reading of their outputs. The Python API also contains specific processing routines, for example for data filtering and error model construction. It can be independently downloaded from pypi (https://pypi.org/project/resipy/) and used in Jupyter Notebook, for instance, which can be useful for automated operations. The upper layer of ResIPy is composed of a set of visualisation tools that provide a graphical environment to the user.

The version of ResIPy supplied allows forward (2D) and inverse (2D and 3D) modelling of resistivity and IP data (a more recent version also permits 3D forward modelling). The first stage is to define the geometry of the problem (topography, electrode positions) and design a mesh. In 2D mode, the mesh can be a structured (quadrilateral element) or unstructured (triangular element) mesh. In 3D mode an unstructured (tetrahedral element) mesh is used. Refinement of an unstructured mesh is achieved by defining characteristic lengths (spacing near electrodes) and growth factors (rate of increase in mesh size away from electrodes). A custom mesh can also be imported for closed or more complex geometry.

For forward modelling (2D) the user can design a geometrical arrangement of resistivity (and IP) regions within the mesh; this is achieved by selecting rectangles, polygons or lines (e.g., as vertical boundaries for horizontally infinite layers) to define such geometry (see Figure A.1). Next, the electrode configuration is selected; full flexibility of quadrupole geometry is permitted: the user can choose predefined standard schemes (Wenner, dipole-dipole, Schlumberger), combine them, or import a custom sequence. Having selected a measurement sequence, the forward model is computed, which is shown graphically as a pseudosection. Gaussian noise can be added to the computed response (as a relative error for resistivity and absolute error for phase angle), allowing the user to run an inverse model and assess sensitivity of the measurement to the selected resistivity (and IP) geometry (see Figure A.2 of the book).

For inverse modelling (2D and 3D), the measurement set is first imported along with the electrode geometry. 2D datasets can be displayed as a pseudosection, allowing removal of any apparent outliers. If reciprocal measurements are available these can be analysed to produce an error model (e.g. Figure 4.21 of the book for DC resistivity, or Figure 4.39 of the book for IP). The mesh is then designed, as in a forward modelling phase (see Figure A.1 of the book) and, after defining inversion parameters such as choice of regularisation options, the inversion is run. Figure A.3 of the book shows an example screenshot from an inversion of a field dataset. Post processing of the model is then possible, e.g. to investigate the distribution of individual model misfits (see Figure 5.14 of the book). For both forward and inverse modelling, results can be stored, e.g. in vtk format, allowing presentation of model results in ParaView. This is particularly useful for 3D inverse modelling.

Additional online materials for ResIPy include:
Access to the current version and other executable formats:
https://gitlab.com/hkex/resipy#downloads

Wiki page with FAQs and common issues you may encounter during working with ResIPy.
https://gitlab.com/hkex/resipy/-/wikis/home

ResIPy YouTube channel:
https://www.youtube.com/channel/UCkg2drwtfaVAo_Tuyeg_z5Q

References

Bérubé, C. L., Chouteau, M., Shamsipour, P., Enkin, R. J. and Olivo, G. R. (2017) 'Bayesian inference of spectral induced polarization parameters for laboratory complex resistivity measurements of rocks and soils', Computers & Geosciences. Elsevier, 105, pp. 51–64.

Blanchy, G., Saneiyan, S., Boyd, J., McLachlan, P., Binley, A. (2020) 'ResIPy, an intuitive open source software for complex geoelectrical inversion/modeling in 2D space', Computers and Geosciences, 137, DOI: 10.1016/j.cageo.2020.104423.

Boyd, J., Blanchy, G., Saneiyan, S., McLachlan, P., Binley, A. (2019) '3D Geoelectrical Problems With ResiPy, an Open Source Graphical User Interface for Geoelectrical Data Processing', Fast Times, 24(4), pp. 85-92.

Cockett, R., Kang, S., Heagy, L. J., Pidlisecky, A. and Oldenburg, D. W. (2015) 'SimPEG: An open source framework for simulation and gradient based parameter estimation in geophysical applications', Computers & Geosciences. 85, pp. 142–154.

Ekinci, Y. L. and Demirci, A. (2008) 'A damped least-squares inversion program for the interpretation of Schlumberger sounding curves', Journal of Applied Sciences, 8(22), pp. 4070–4078.

Gernez, S., Bouchedda, A., Gloaguen, E. and Paradis, D. (2020) 'AIM4RES, an open-source 2.5D finite differences MATLAB library for anisotropic electrical resistivity modeling', Computers and Geosciences. 135, p. 104401. doi: 10.1016/j.cageo.2019.104401

Ghorbani, A., Camerlynck, C. and Florsch, N. (2009) 'CR1Dinv: A Matlab program to invert 1D spectral induced polarization data for the Cole–Cole model including electromagnetic effects', Computers & Geosciences. Elsevier, 35(2), pp. 255–266.

Günther, T., Rücker, C. and Spitzer, K. (2006) 'Three-dimensional modelling and inversion of dc resistivity data incorporating topography - II. Inversion', Geophysical Journal International, 166(2), pp. 506–517. doi: 10.1111/j.1365-246X.2006.03011.x.

Karaoulis, M., Revil, A., Tsourlos, P., Werkema, D. D. and Minsley, B. J. (2013) 'IP4DI: A software for time-lapse 2D/3D DC-resistivity and induced polarization tomography', Computers and Geosciences, 54, pp. 164–170. doi: 10.1016/j.cageo.2013.01.008.

Nath, S. K., Shahid, S. and Dewangan, P. (2000) 'SEISRES—a visual C++ program for the sequential inversion of seismic refraction and geoelectric data', Computers & Geosciences. 26(2), pp. 177–200.

Pidlisecky, A. and Knight, R. (2008) 'FW2_5D: A MATLAB 2.5-D electrical resistivity modeling code', Computers & Geosciences. 34(12), pp. 1645–1654.

Pidlisecky, A., Haber, E. and Knight, R. (2007) 'RESINVM3D: A 3D resistivity inversion package', Geophysics. 72(2), pp. H1–H10.

Rücker, C., Günther, T. and Wagner, F. M. (2017) 'pyGIMLi: An open-source library for modelling and inversion in geophysics', Computers & Geosciences. 109, pp. 106–123.

Weigand, M. and Kemna, A. (2016) 'Debye decomposition of time-lapse spectral induced polarisation data', Computers and Geosciences, 86, pp. 34–45. doi: 10.1016/j.cageo.2015.09.021.

Appendix 1. Currently available modelling tools

Name	Free*	Source	Brief description	Reference or web link
IPI2WIN	✓		1D resistivity and IP.	http://geophys.geol.msu.ru/ipi2win.htm
Sensing 1D	✓		1D resistivity and IP.	http://www.harbourdom.de/sensiv1d.htm
VES1dinv	✓	✓	1D resistivity. MATLAB® source.	Ekinci and Demirci (2008)
EarthImager 1D			1D resistivity commercial code.	https://www.agiusa.com/agi-earthimager-1d-ves
SPIA			1D VES can be used as part of the Aarhus Workbench (see below).	https://hgg.au.dk/software/spia-ves/
ZONDIP1D			1D resistivity and IP.	http://zond-geo.com/english/zond-software/ert-and-ves/zondip1d/
1X1Dv3			1D resistivity and IP.	http://www.interpex.com/ix1dv3/ix1dv3.htm
CR1Dinv	✓	✓	1D resistivity and IP relaxation modelling. MATLAB® source.	Ghorbani et al. (2009).

SEISRES	✓	✓	1D and 2D resistivity guided by seismic refraction inversion. Visual C++ code.	Nath et al.(2000). See also https://www.iamg.org/documents/oldftp/VOL26/v26-2-5.zip
FW2_5D	✓	✓	2D resistivity. MATLAB® source.	Pidlidesky and Knight (2008).
R2, cR2	✓		2D resistivity and FDIP.	http://www.es.lancs.ac.uk/people/amb/Freeware/Freeware.htm See also Section A.2 of the book.
DCIP2D	✓		2D resistivity and TDIP.	https://dcip2d.readthedocs.io/en/latest/
EarthImager 2D			2D resistivity commercial code.	https://www.agiusa.com/agi-earthimager-2d
X2IPI			2D resistivity and TDIP.	http://x2ipi.ru/en
ZONDRES2D			2D resistivity and TDIP/FDIP.	http://zond-geo.com/english/zond-software/ert-and-ves/zondres2d/
Res2DInv			Widely used 2D resistivity and IP commercial code.	https://www.geotomosoft.com/index.php
Aarhus Workbench			Commercial modelling platform for a range of applications including 2D resistivity and TDIP.	https://hgg.au.dk/software/aarhus-workbench/

E4D	✓		3D resistivity and FDIP. Designed for large problems with parallel computation.	https://e4d.pnnl.gov/Pages/Home.aspx
IP4DI	✓	✓	2D and 3D resistivity and TDIP/FDIP. MATLAB® source.	Karaoulis et al. (2013).
RESINVM3D	✓	✓	3D resistivity. MATLAB® source.	Pidlisecky et al. (2007). See also https://software.seg.org/2007/0001/
AIM4RES	✓	✓	2D anisotropic resistivity inverse code. MATLAB® source.	Gernez et al. (2020). See also https://github.com/Simoger/AIM4RES
DCIP3D	✓		3D resistivity and TDIP.	https://dcip3d.readthedocs.io/en/latest/
EarthImager 3D			3D resistivity and TDIP commercial code.	https://www.agiusa.com/agi-earthimager-3d
BERT	✓	✓	2D and 3D resistivity and FDIP.	https://gitlab.com/resistivity-net/bert See also Gunther et al. (2006) and *pyGIMLi* below
pyGIMLi	✓	✓	Generalised inversion suite allowing multi-dimensional inversion. Python source.	Rucker et al. (2017). See also https://www.pygimli.org/ and *BERT* above.

Eidors	✓	✓	2D and 3D resistivity targeted for biomedical applications. MATLAB® source.	http://eidors3d.sourceforge.net/
R3t, cR3t	✓		3D resistivity and FDIP.	http://www.es.lancs.ac.uk/people/amb/Freeware/Freeware.htm See also Section A.2 of the book.
Res3DInv			Widely used 3D resistivity and IP commercial code.	https://www.geotomosoft.com/index.php
ERTLab64			3D resistivity and IP commercial code.	http://ertlab64.com/
Geccoinv	✓	✓	Time-lapse SIP relaxation modelling using Debye decomposition. Python source.	Weigand and Kemna (2016). See also https://github.com/m-weigand/Debye_Decomposition_Tools
BISIP	✓	✓	Markov-chain Monte Carlo SIP relaxation modelling. Python source.	Bérubé et al. (2017). See also https://github.com/clberube/bisip
SimPEG	✓	✓	Simulation and gradient based parameter estimation in geophysical applications. Python source.	Cockett et al. (2005). See also https://simpeg.xyz/

ResIPy	✓	✓	Modelling environment for analysis and inversion of 2D and 3D resistivity and IP data. Python source.	Boyd et al. (2019), Blanchy et al. (2020). See also https://gitlab.com/hkex/resipy and Section A.3 of the book.

*Will be subject to licence agreement in some cases