Geoscience Letters

REVIEW

Open Access

A review of open software resources in python for electrical resistivity modelling

Yonatan Garkebo Doyoro[1,2,4], Ping-Yu Chang[1,3*], Jordi Mahardika Puntu[1], Ding-Jiun Lin[1], Tran Van Huu[1], Diah Ayu Rahmalia[1] and Meng-Shiun Shie[1]

Abstract

Geophysical modelling performs to obtain subsurface structures in agreement with measured data. Freeware algorithms for geoelectrical data inversion have not been widely used in geophysical communities; however, different open-source modelling/inversion algorithms were developed in recent years. In this study, we review the structures and applications of openly Python-based inversion packages, such as pyGIMLi (Python Library for Inversion and Modelling in Geophysics), BERT (Boundless Electrical Resistivity Tomography), ResIPy (Resistivity and Induced Polarization with Python), *pyres* (Python wrapper for electrical resistivity modelling), and SimPEG (Simulation and Parameter Estimation in Geophysics). In addition, we examine the recovering ability of pyGIMLi, BERT, ResIPy, and SimPEG freeware through inversion of the same synthetic model forward responses. A versatile pyGIMLi freeware is highly suitable for various geophysical data inversion. The SimPEG framework is developed to allow the user to explore, experiment with, and iterate over multiple approaches to the inverse problem. In contrast, BERT, *pyres*, and ResIPy are exclusively designed for geoelectric data inversion. BERT and pyGIMLi codes can be easily modified for the intended applications. Both *pyres* and ResIPy use the same mesh designs and inversion algorithms, but *pyres* uses scripting language, while ResIPy uses a graphical user interface (GUI) that removes the need for text inputs. Our numerical modelling shows that all the tested inversion freeware could be effective for relatively larger targets. pyGIMLi and BERT could also obtain reasonable model resolutions and anomaly accuracies for small-sized subsurface structures. Based on the heterogeneous layered model and experimental target scenario results, the geoelectrical data inversion could be more effective in pyGIMLi, BERT, and SimPEG freeware packages. Moreover, this study can provide insight into implementing suitable inversion freeware for reproducible geophysical research, mainly for geoelectrical modelling.

Keywords: Resistivity modelling, Inversion freeware, pyGIMLi, BERT, ResIPy, Pyres, SimPEG

Introduction

The direct current resistivity method is widely used in geophysical near-surface prospecting, such as hydrogeological (Chambers et al. 2006; Chang et al. 2017; Zhang et al. 2016), geological structure (Caputo et al. 2003; Chang et al. 2015), engineering (Arjwech and Everett 2019; Lin et al. 2013), and environmental (Cardarelli et al. 2010; Van Schoor 2002) surveys. The resistivity method has been significantly advanced in data acquisitions and inversion techniques in recent decades. Modern resistivity acquisition obtains a large number of data in two dimension (2D) and three dimension (3D) to recover complex geological structures that are not possible with a one-dimensional (1D) survey (Dahlin and Zhou 2004; Sharma and Verma 2015). The measured apparent resistivity data can be inverted to reconstruct subsurface spatial resistivity distribution using inversion algorithms. Although several commercial inversion software have been established for geoelectric data inversion, they are less extensible to use and not easily accessible for independent researchers.

*Correspondence: pingyuc@ncu.edu.tw
[1] Department of Earth Science, National Central University, Taoyuan, Taiwan
Full list of author information is available at the end of the article

Technically versatile users can also commonly end up building individually tailored solutions by linking various existing potentially commercial software through scripts, which hinders the reproducibility of scientific researches (Peng 2011). This motivates and supports the need for modern freeware architectures for the numerical tasks in geophysical studies.

Accessible and extensible inversion freeware algorithms are developed through open-source initiatives using different interpretive languages (Cockett et al. 2015). Python is the most versatile and interpretive language among the open-source programming languages for developing various geoscientific packages. It also combines numerical computation performance with high-end graphical outputs and user interfaces. Furthermore, Python code can solve a scientific problem and allow every scientific community to advance the source codes by enriching feedback from different users and developers using social coding platforms, such as GitHub (https://github.com) and Read the Docs (https://readthedocs.org/) (Lin 2012; Rücker et al. 2017). Thus, geoelectric inversion freeware packages are mainly developed under the Python interface, which can interrogate the source codes and, where necessary, pertain and customize for needed applications.

This paper reviews the commonly used geoelectric data inversion freeware established in the Python interface. The freeware includes pyGIMLi (Python Library for Inversion and Modelling in Geophysics), BERT (Boundless Electrical Resistivity Tomography), ResIPy (Resistivity and Induced Polarization with Python), *pyres* (Python wrapper for electrical resistivity modelling), and SimPEG (Simulation and Parameter Estimation in Geophysics). We reviewed based on code structures, mesh designs, package dependencies, and applications.

Even though various studies implement inversion freeware for multiple applications (Benjamin et al. 2020; Gourdol et al. 2018; Klingler et al. 2020), there are no studies that assess the effectiveness of different freeware packages for geoelectric data inversion. We examine the pyGIMLi, BERT, ResIPy, and SimPEG freeware performances by inverting the same input models. A conceptual model with low and high resistive targets sets at different depths of the homogeneous host medium is used to measure the apparent resistivity data synthetically. The inverted models have shown variations based on the target size and buried depth in addition to the type of inversion freeware used. In addition, we have suggested suitable inversion freeware for subsurface structural studies. Overall, this review paper may encourage geoscientific communities to implement the inversion freeware for modelling and inverting geoelectrical datasets.

Freeware framework

pyGIMLi

pyGIMLi package is created using a Python programming script that provides modular functionality for different geophysical studies. The architecture of pyGIMLi constitutes three significant conceptual levels: the equation, the modelling, and the application levels. The equation level provides an interface to solve partial differential equations on a given mesh, comprising all geometric specifications, for instance, topography and known subsurface structures. The modelling level represents a collection of classes to solve a simulation task for a specific geophysical method by applying the equation level or using appropriate calculations. The application-level defines a general framework to solve basic and advanced inversion tasks, like time-lapse and joint inversion. All the conceptual levels interacted through a unified Python interface to resolve the forward and inverse problem of the resistivity method. A more comprehensive design and architecture of pyGIMLi freeware are explained by Rücker et al. (2017).

pyGIMLi provides a versatile meshing method for the forward problem formulation. Its modelling component allows the finite element and finite volume solvers to determine the differential equations on unstructured mesh. Discretizing unstructured triangular mesh (or tetrahedral meshes in 3D) produces an error for smaller or larger mesh angles. The larger mesh angle can increase interpolation error, while the smaller mesh angle can cause an ill-conditioned stiffness matrix, leading to inversion singularity (Du et al. 2009; Neumaier 1998). The quality checker in pyGIMLi can handle mesh discretization problems and provides optimum quality mesh. pyGIMLi can also import mesh from external mesh creators, such as Triangle (Shewchuk 1996), TetGen (Si 2015), and Gmsh (Geuzaine and Remacle 2009).

Figure 1 presents the generalized inversion procedure of pyGIMLi. Its inversion mainly implements a deterministic Gauss–Newton algorithm with flexible regularization (Rücker et al. 2017). The inversion algorithm can be controlled by applying various regularization methods, including parameter transformation, starting model, and anisotropic and stochastic regularizations. Furthermore, pyGIMLi post-processing routines can be provided to visualize results in 2D using Matplotlib (Hunter 2007) and 3D using ParaView (Ayachit 2015).

BERT

The functionality of BERT is based on a multi-method of GIMLi (Generalized Inversion and Modelling Library). The Python interface of GIMLi can be used to retrieve, build, and utilize the pyBERT code (Python binding of BERT). Therefore, *pygimli.meshtool*s library can be

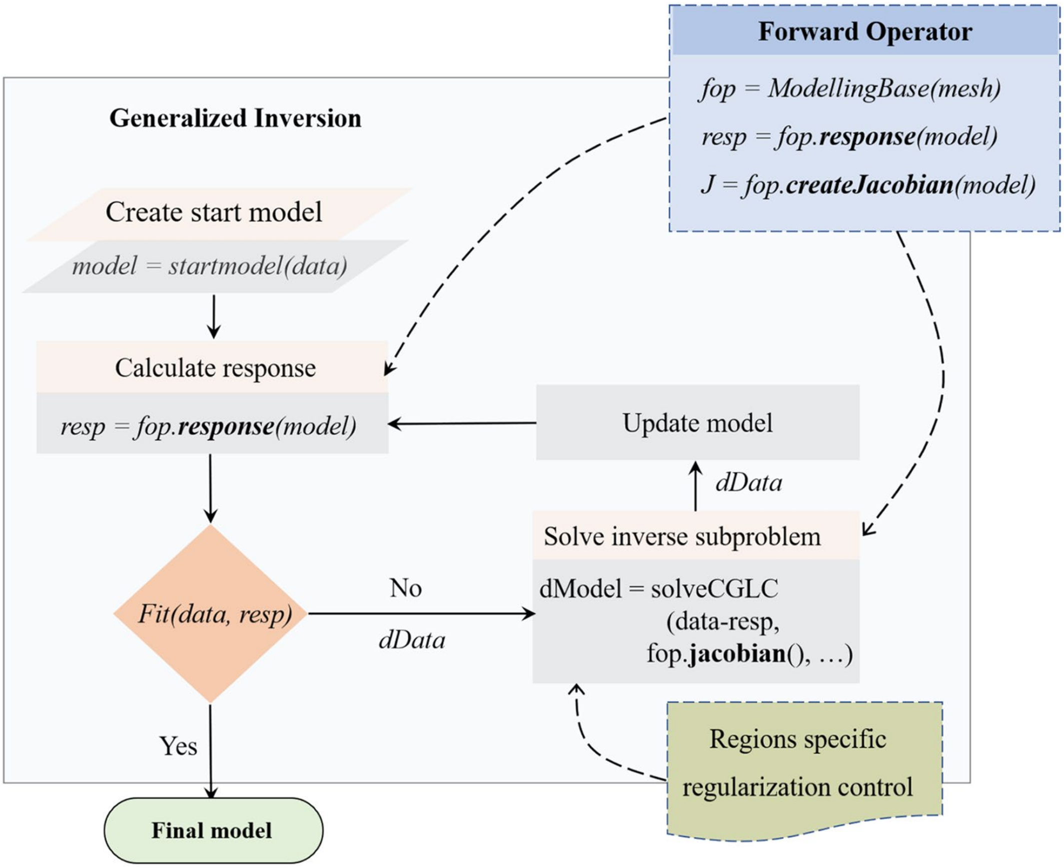
Forward Operator
fop = ModellingBase(mesh)
*resp = fop.**response**(model)*
*J = fop.**createJacobian**(model)*
Generalized Inversion
Create start model
model = startmodel(data)
Calculate response
*resp = fop.**response**(model)*
Update model
dData
Solve inverse subproblem
dModel = solveCGLC (data-resp, fop.**jacobian**(), …)
Fit(data, resp)
No
dData
Yes
Regions specific regularization control
Final model

Fig. 1 pyGIMLi generalized inversion scheme (Rücker et al. 2017)

imported in addition to pyBERT modules for modelling and inversion of geoelectric data. We direct the reader to Günther et al. (2006) for a more compressive description of BERT freeware theory and technology.

BERT applies efficient meshing approaches for resistivity problem formulation. It uses unstructured triangular mesh for 2D modelling while tetrahedral mesh for 3D modelling (Günther and Rücker 2015). Similar to pyGIMLi, the BERT freeware can control mesh quality that enhances the numerical accuracy of the forward calculations. It can also import a free and versatile mesh from external mesh generators, including TetGen (Si 2015) and Gmsh (Geuzaine and Remacle 2009).

Figure 2 displays the overall inversion procedure of BERT. First, the primary potentials are calculated and interpolated onto the secondary mesh. Then, the geometric factors yield the apparent resistivity and the sensitivity matrix for a homogeneous case. Finally, model inversion is performed: an inverse sub-problem is used to update the resistivity model and a forward calculation is carried out and checked. Moreover, the inversion implements a smoothness-constrained Gauss–Newton algorithm (Günther et al. 2006). It was later formulated as a flexible minimization and regularization scheme, which is furtherly explained by Rücker (2010).

pyres/ResIPy

pyres freeware and R2 package are explicitly designed for modelling and inversion of geoelectric data. The R2 package provides a robust toolset for the forward and inverse solutions. However, many users of the R2 package had created their codes for managing and manipulating the text files, there were no published codes that provide an object-oriented programming interface for R2 or that incorporate mesh creation and parameterization for R2. Hence, *pyres* Python wrapper package for R2 were used to create a powerful and flexible programmatic interface for modelling and inverting geoelectric datasets (Befus 2018). It contains a minimal number of dependent modules along with Python packages. For instance, *pyres* modules, such as *mesh_tools, plot_utils,* and *pyres_utils,* can help create an input file and run R2, construct the mesh, plot the output file, and calculate the depth of investigation. In addition to managing R2 inputs

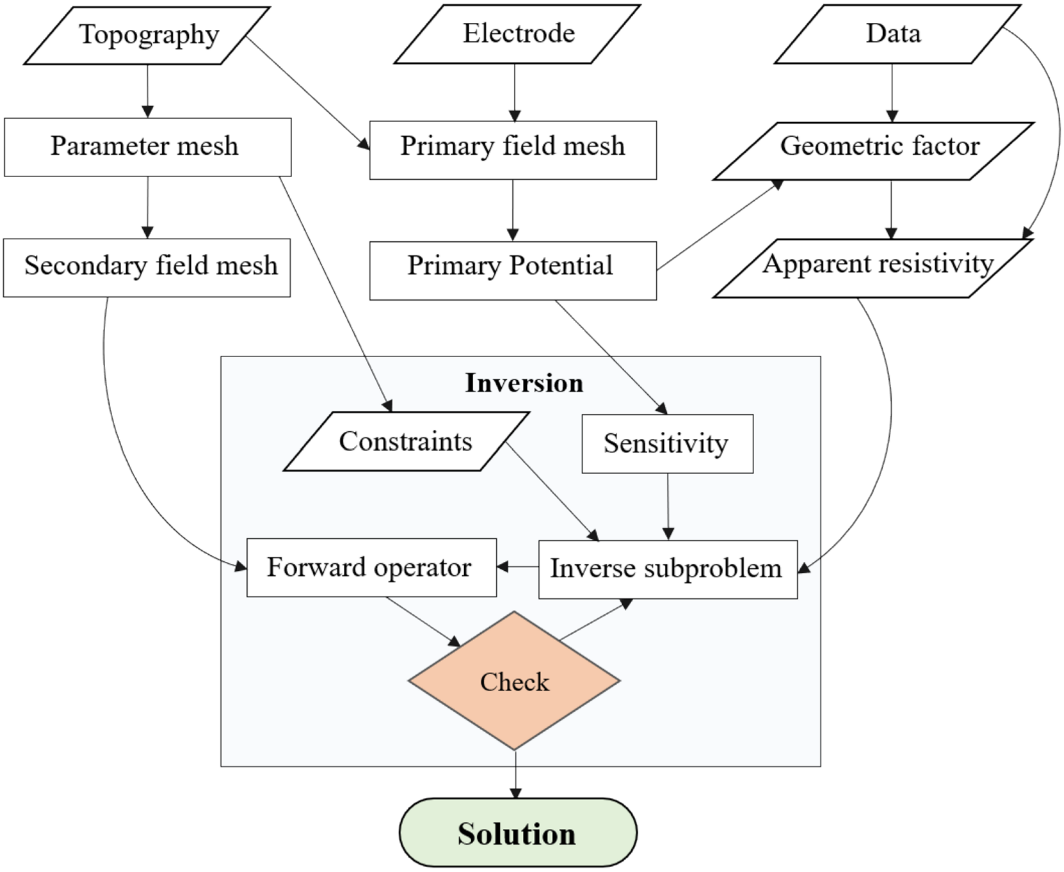

Fig. 2 BERT inversion scheme indicating the central loop of the forward and inverse step until the stopping criterion is reached and a solution is obtained (Günther et al. 2006)

and running R2 forward and inverse models, *pyres* contain several other input and output methods that can be incorporated with user-defined Python code for higher-level pre- or post-processing. The forward problem in *pyres* can be solved using finite element quadrilateral (structured or unstructured) or triangular (unstructured) mesh (Befus 2018). The quadrilateral mesh node can be defined using NumPy arrays (Bressert 2012), while the triangular mesh can be extracted from Gmsh (Geuzaine and Remacle 2009). Furthermore, *pyres* implements a weighted least-squares objective function coupled with a range of regularizations (Binley and Kemna 2005).

ResIPy can also enable the modelling and inversion of geoelectric datasets. It is established under the Python interface, and its source code is available on a GitLab repository (https://gitlab.com/hkex/pyr2). ResIPy applies the freely available codes, such as R2, R3t, and cR2. The R2 and R3t codes are developed to solve the 2D and 3D direct current resistivity. On the contrary, the cR2 code is designed to solve the induced polarization problem (Binley and Kemna 2005). These codes require formatted text files for input, forward and inverse model setting, and mesh construction. However, the graphic user interface (GUI) in ResIPy removes the need for such text input and assists the user in pre- and post-processing stages. As shown in Fig. 3, ResIPy implements structured quadrilateral and unstructured triangular finite element meshes for resistivity calculations. Additionally, it can import complex mesh from Gmsh (Geuzaine and Remacle 2009). We forward the reader to Blanchy et al. (2020) for further design aspects of ResIPy.

The general workflow of ResIPy is demonstrated in Fig. 3. The geoelectric data can be synthetically measured using forward problem formulation. The synthetic or field data can be imported and filtered out the error data points or electrodes. The filtered data and the generated mesh can be sent to the inversion pipeline. Inversion can be performed using a weighted least-squares objective function, coupled with a range of regularizations. The procedure ends with an inverted geoelectric section. Furthermore, a diagnostic pseudo-section of normalized

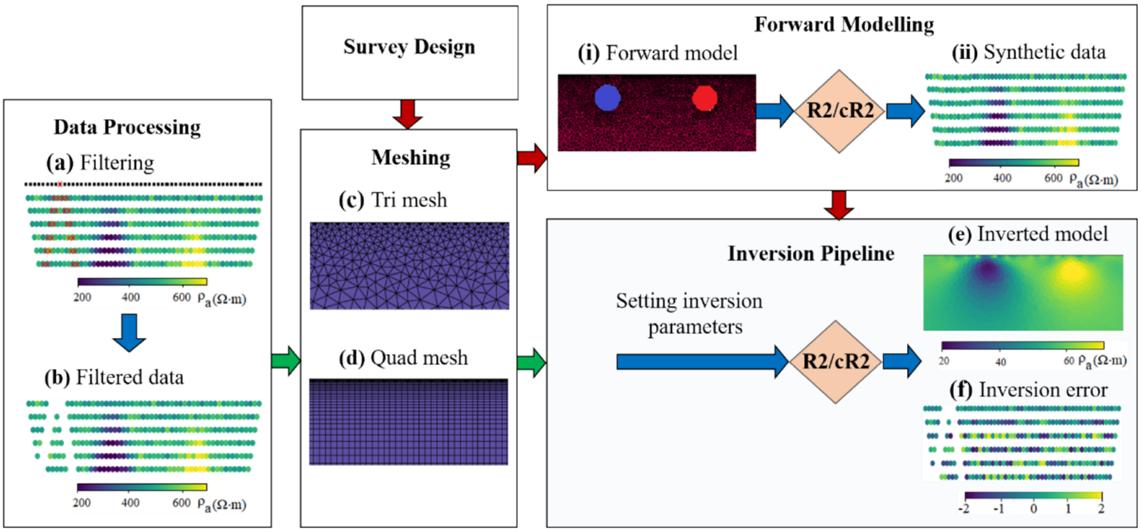

Fig. 3 ResIPy workflow. Inversion workflow (green arrows): **a** import the raw data and filter out the noises, **b** prepare data for inversion, **c** triangular mesh, **d** quadrilateral mesh, **e** inverted model, **f** normalized inversion error. Forward modelling workflow (red arrows): **i** synthetic model and mesh creation, and **ii** synthetically measure the geoelectric data (Blanchy et al. 2020)

inversion error can be produced (Binley and Kemna 2005).

SimPEG

SimPEG framework is also an open-source library written in Python interface to facilitate an experimental and empirical inversion for different applications. It interfaces numerical solver packages, model parameterizations, and visualization routines (Cockett et al. 2015). The development of SimPEG focused on modularity, usability, documentation, and extensive unit testing that can help as an interface create other inversion code in the geoscientific community (Cockett et al. 2015; Wilson et al. 2014).

SimPEG mainly implements a staggered grid and mimetic finite volume discretization on structured and semi-structured meshes (Hyman and Shashkov 1999). This approach requires definitions of variables at either cell centres, nodes, faces, or edges. Its forward resistivity calculation uses three different meshes: Tensor, Tree, and Curvilinear meshes (Cockett et al. 2015). The type of needed mesh can be imported from SimPEG libraries, including the supporting modules of NumPy and SciPy (Bressert 2012). Moreover, in a 1D direct current resistivity experiment, the governing equation with supplied boundary conditions can be solved using finite volume, finite element, integral equation, or semi-analytical method.

Figure 4 outlines the general procedure of SimPEG that implements the Tikhonov inversion approach. There are three main steps in SimPEG inversion: the input, the implementation, and the evaluation stages. The interrogation of these three stages can minimize the ill-posedness of an inverse problem. The input stage includes the geophysical data, governing equation, and *prior* knowledge (geological or petrophysical information). The two broad categories, such as forward and inverse modelling, are comprised in the implementation stage. The forward simulation solves the governing equation for a given model, while the inversion component iteratively modifies and updates the recovered model. SimPEG implements a gradient-based inversion to update the model through optimization routines (Cockett et al. 2015). The optimized inversion numerically determines data misfits and regularizes an inverse problem (Tikhonov et al. 2013). Finally, the inverted model can be evaluated before interpretation. The evaluation stage can re-assess the choices and assumptions made in both the input and implementation stages. The inversion setting can be modified in the evaluation stage to assess whether the inverted model result can fit the desired target or not. A complete SimPEG inversion procedure was described by Cockett et al. (2015).

Dependencies

There are several modern and user-level package managers that control the freeware environments and dependencies in Python. *conda* and *pip* are the prominent package managers that are usually used to install freeware (Lehe et al. 2020). The required packages and dependencies in Table 1 are needed to be installed before using the Python inversion freeware.

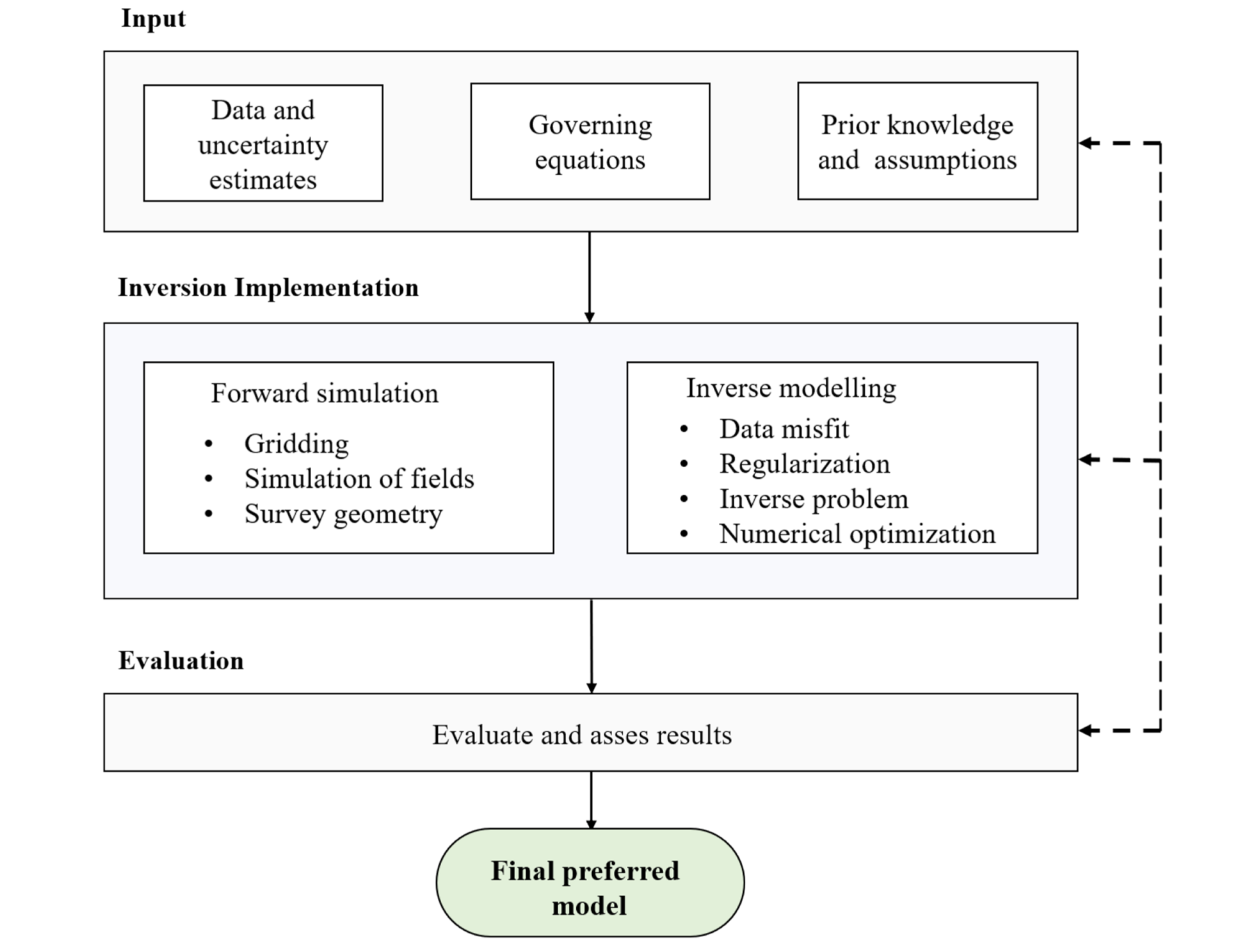

Fig. 4 SimPEG inversion procedure including inputs, implementation, evaluation and interpretation (Cockett et al. 2015)

Table 1 Pre-requests for Python-based inversion freeware

Python Freeware	Dependencies
PyGIMLi (https://www.pygimli.org)	Python, NumPy, Jupiter Notebook, or Spyder, SciPy, TetGen, and Matplotlib
BERT (http://resistivity.net/bert)	Python, NumPy, Jupiter Notebook, or Spyder, SciPy, TetGen, and Matplotlib
pyres (https://github.com/kbefus/pyres)	Python, NumPy, R2 software, Jupiter Notebook or Spyder, SciPy, and Matplotlib
SimPEG (https://docs.simpeg.xyz)	Python, NumPy, Cython, Jupiter Notebook or Spyder, SciPy, Matplotlib

Numerical experiment

Methodology

We develop a synthetic resistivity model to examine the effectiveness of Python-based inversion freeware, such as pyGIMLi, BERT, ResIPy, and SimPEG. This study uses the latest freeware versions: pyGIMLi 1.2.2, BERT 2.3.2, ResIPy 3.3.2, and SimPEG 0.15.1; their future inversion performance may be varied due to different advancements of the tested freeware codes. We use a conceptual model representing a horizontally stratified sedimentary layer and archaeological targets buried in a homogeneous host medium. Forward resistivity modelling is performed for a layered geologic model that consists of a sand layer with a resistivity value of 200 Ωm, a gravel layer with a resistivity value of 600 Ωm, and a moderately fractured sandstone with a resistivity value of 1000 Ωm, respectively, from top to bottom (Fig. 8a). Moreover, we conduct forward simulation for an archaeological model (Fig. 5a) comprising a conductive target (left side) with a resistivity value of 10 Ωm and a relatively resistive target

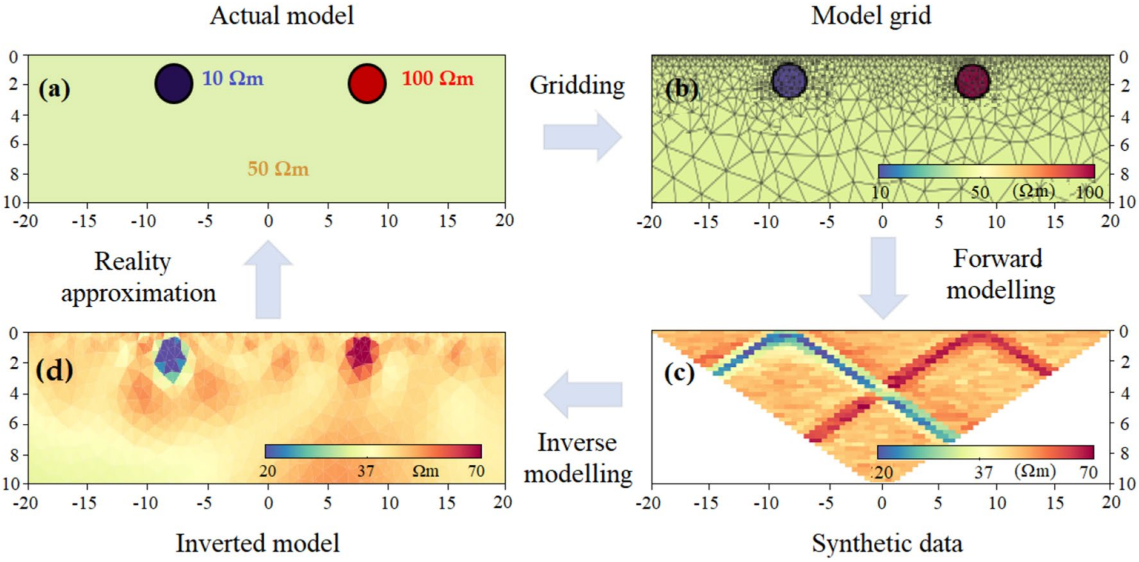

Fig. 5 The general framework of the forward-inverse modelling of geoelectric data: **a** synthetic model with two target bodies representing simple subsurface reality, **b** model discretization, **c** synthetic measure of apparent resistivity data, and **d** recovered model

(right side) with a resistivity value of 100 Ωm, buried in a silty clay host medium with a resistivity value of 50 Ωm (Keller 2017). We use four different scenarios based on the target sizes and survey depths. A 1 m target radius is set at three different depths to assess the depth effect on the performances of the inversion freeware; thus, the 1 m target is buried at 1.5 m deep in scenario one, at 3 m deep in scenario two, and 5 m deep in scenario three. We also use scenario four, a 2 m target radius situated at 3 m depth, to examine the size effect.

The freeware algorithms are applied to solve the forward resistivity problem. Since the dipole–dipole array provides a high resolution for buried targets (Doyoro et al. 2021; Seaton and Burbey 2002), we apply its arrangement for surface resistivity probing using 41 electrodes with 1 m spacing. We use the finite element triangular unstructured mesh for model discretization in pyGIMLi, BERT, and ResIPy as shown in Fig. 5b, while finite volume tensor mesh is implemented in SimPEG. Every tested freeware is used to generate its forward resistivity datasets; for instance, Fig. 5c displays forward resistivity datasets for pyGIMLi. As measurement error does not observe in the synthetic data, we integrate the acquisition error by perturbing 3% Gaussian noise in which the random value noises are generated for every data point with zero mean and 3% standard deviation.

The inversion algorithms discussed in each freeware package are used for the inversion of forward resistivity data. The regularization value is set to 20 initially, which is reduced iteratively. The iterative inversion process can modify the starting homogeneous model until an acceptable convergence between the model response and the observed apparent resistivity has to be achieved. The inverted models are compared to the actual model to examine the recovering ability of the inversion freeware (Fig. 5d).

Modelling algorithms may vary based on the efficiency of numerical calculations (Narayan et al. 1994). We use the horizontally stratified geologic model (Fig. 6) to evaluate forward resistivity variation among the freeware packages. The forward resistivity results of pyGIMLi, BERT, and ResIPy are examined for different sizes of unstructured triangular grids. The comparison does not

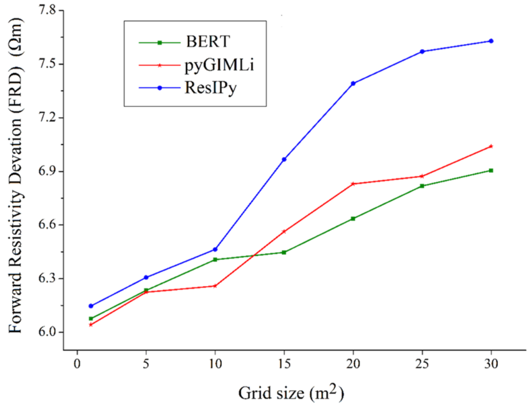

Fig. 6 Forward resistivity data variation among the pyGIMLi, BERT, and ResIPy freeware for different grid sizes. The grid size value indicates the maximum area of unstructured triangular mesh for forward resistivity formulation

include SimPEG forward results, as it implements finite volume tensor mesh. We determine the freeware forward resistivity deviation (FRD) by averaging the resistivity values of pyGIMLi, BERT, and ResIPy, and subtract the individual freeware resistivity values from the average values, which is divided by the total number of data points; it is expressed by

$$\mathrm{FRD} = \frac{\left|\rho_{\mathrm{avi}-F\rho_i}\right|}{n}, \tag{1}$$

where ρ_{avi} is the average value of forward resistivity data points of pyGIMLi, BERT, and ResIPy, $F\rho_i$ is the forward resistivity values for individual freeware packages, and n is the number of data points.

Militzer et al. (1979) first introduced the anomaly effect (AE) to study the measuring ability of the electrode arrays. The anomaly effect can also be used to examine the recovering effectiveness of arrays for different buried geological targets and inspects the corresponding image quality, and it is determined using the expression (Aizebeokhai et al. 2010; Militzer et al. 1979).

$$\mathrm{AE} = \left|\frac{\rho_{\mathrm{max}} - \rho_{\mathrm{min}}}{\rho_{\mathrm{av}}}\right|, \tag{2}$$

where ρ_{max}, ρ_{min}, and ρ_{av} indicate the maximum, minimum, and average apparent resistivity datasets, respectively. In this study, we apply the anomaly effect to assess forward resistivity calculating abilities of the inversion freeware (pyGIMLi, BERT, ResIPy, and SimPEG).

Modelling results

The obtained resistivity model resolution relies on the quality of measured datasets and the effectiveness of the inversion software. Even though the same input model is used, different inversion packages may show variations in the recovered model resolution and anomaly geometry. We examine the pyGIMLi, SimPEG, BERT, and ResIPy inversion freeware for horizontally layered geologic conditions and archaeological targets sets in a homogeneous medium.

We assess the forward resistivity datasets of pyGIMLi, BERT, and ResIPy for different grid sizes. Figure 6 shows the freeware forward resistivity deviations from the average value. Low resistivity variations are shown among the tested freeware for the small grid sizes, whereas relatively high resistivity variations are displayed for the large grid sizes. However, the result generally shows less considerable resistivity discrepancies among the tested freeware, particularly between BERT and pyGIMLi.

Forward resistivity data with maximum anomaly information can obtain a reliable inverted image (Okpoli 2013). Thus, we calculate the anomaly effect of forward resistivity data for simulated target scenarios using tested freeware packages. As presented in Fig. 7, the highest anomaly effect of 0.985, 0.967, 0.938, and 0.917 for the corresponding freeware packages of BERT, pyGIMLi, SimPEG, and ResIPy is exhibited for simulation of scenario four. In contrast, the modelling of scenario three displays the lowest anomaly effect of 0.123, 0.223, 0.231, and 0.250 for ResIPy, SimPEG, pyGIMLi, and BERT, respectively. The result shows a slight variation of anomaly effect for the tested freeware packages, mainly BERT, pyGIMLi, and SimPEG. The anomaly effect is considerably varied for the target size and buried depth. The large-sized scenario four depicts a more pronounced anomaly effect compared to scenario two, situated at the same depth. Scenario three, relatively located at deeper depth, shows the lowest anomaly effect than the tested scenarios. The increase of target buried depth can decrease the anomaly effect, indicating direct dependency on obtained image quality.

We assess inversion freeware performances for a horizontally layered sedimentary model (Fig. 8a). The obtained model results for the tested freeware packages are presented in Fig. 8b–e. All the freeware packages detect the sand layer; however, the thickness is slightly reduced in pyGIMLi and BERT. The gravel layer is more adequately identified in pyGIMLi, BERT, and SimPEG, but BERT's layer thickness is considerably decreased. Similarly, the sandstone layer is recovered in all the tested inversion freeware eÎpt ResIPy. However, a significant overestimation of sandstone layer thicknesses is depicted in pyGIMLi and BERT. The gravel and sandstone layers are combined in the inverted models of ResIPy, indicating two-layer structures instead of three model layers. The centre of the model profiles is well resolved in all the tested inversion freeware. However, smearing is observed

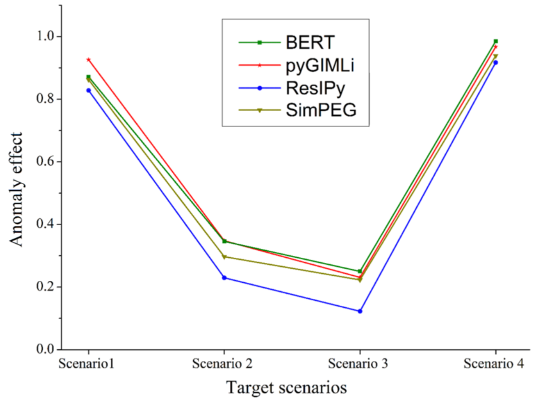

Fig. 7 The anomaly effect of the pyGIMLi, BERT, SimPEG, and ResIPy freeware for simulated buried targets

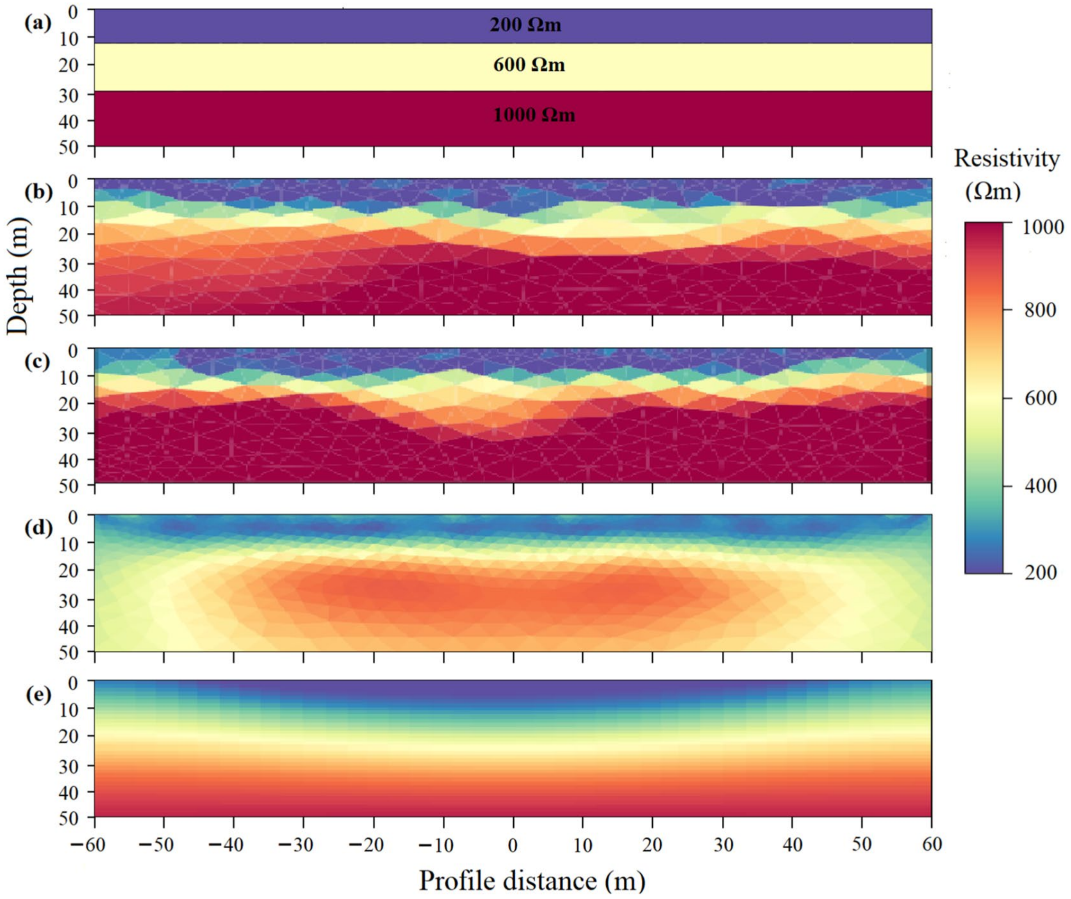

Fig. 8 **a** The layered conceptual model consists of sand, gravel, and moderately fractured sandstone from top to bottom. The inverted model results for the inversion freeware: **b** pyGIMLi, **c** BERT, **d** ResIPy, and **e** SimPEG. Unstructured triangular mesh is implemented in pyGIMLi, BERT, and ResIPy, whereas tensor mesh is used in SimPEG

at the end of the profiles, likely related to limited data points or no data coverages.

Inversion freeware packages are also examined for archaeological targets hosted in a homogeneous host medium. Figure 9 displays the inversion results for target scenario one with a 1 m target radius set at 1.5 m depth. The result exhibits high model resolution and accurate anomaly geometry for pyGIMLi and BERT, yet their anomalies have slightly deviated from the actual positions. However, a moderate resolution is observed in SimPEG, and relatively low resolution is exhibited in ResIPy. SimPEG and ResIPy correctly resolve the upper boundary of the target anomaly, while the base is slightly extended beyond the actual model. The background anomaly is reproduced accurately in all the tested freeware packages. Compared to the initial target resistivities, the inverted resistivity values are overestimated in the conductive target zone while underestimated in the resistive target. In all the tested inversion freeware, the recovered resistivity values of scenario one are overestimated about 7–10% for the conductive target while underestimated about 3–5% for the resistive target.

Scenario two shows the recovered models for a conceptual model with a 1 m target radius set at 3 m depth, as shown in Fig. 10. The target anomalies are moderately identified in pyGIMLi and BERT while slightly indicated in SimPEG and ResIPy. Besides, the obtained anomaly geometries are not correctly recovered in all the tested inversion freeware. Scenario two inversions in the tested freeware packages show considerable overestimations (33–38%) in the obtained resistivity for the conductive target while underestimations (21–26%) for the resistive target (Fig. 10). Overall, scenario two shows a noticeably lower model resolution than the shallowest target scenario (Fig. 9).

In scenario three, a conceptual model of a 1 m target radius set at 5 m depth is inverted to assess further depth effect on the performances of the inversion freeware (Fig. 11). pyGIMLi, BERT, and SimPEG provide blurred resolution with highly broadened anomalies. On

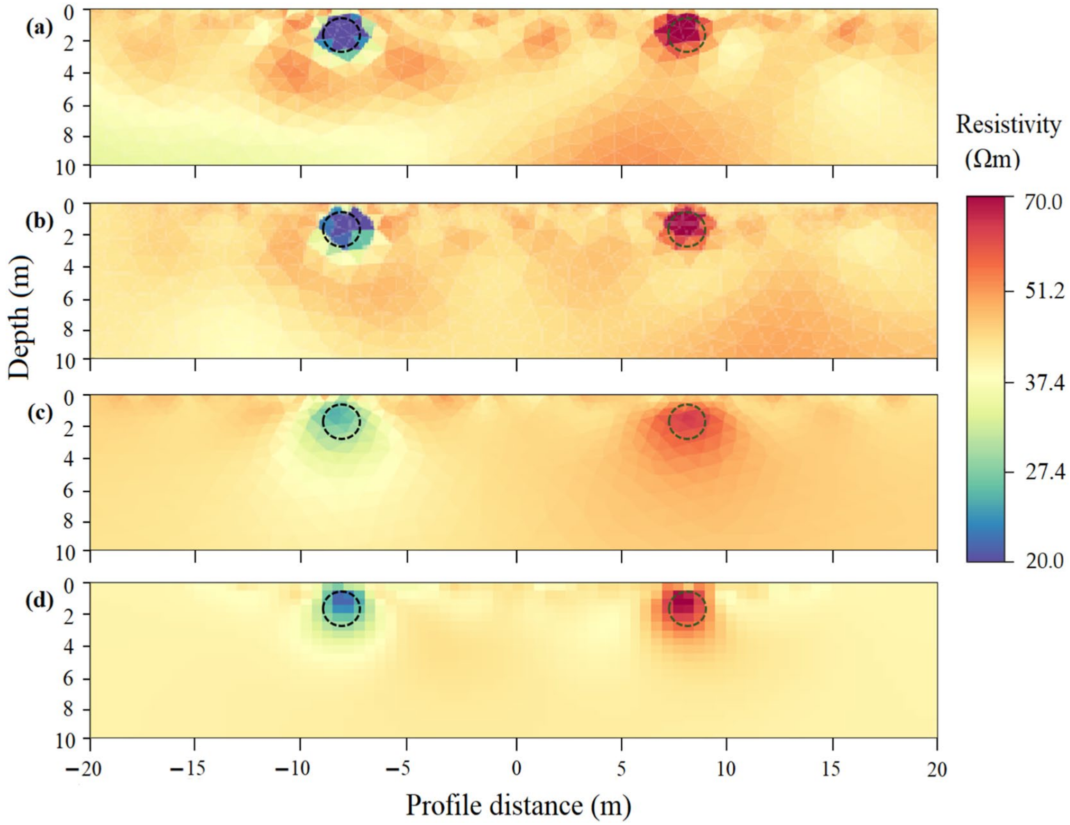

Fig. 9 The inverted model of scenario one for a 1 m target radius located at 1.5 m depth using the inversion freeware: **a** pyGIMLi, **b** BERT, **c** ResIPy, and **d** SimPEG. The broken circles indicate the actual targets; the left for conductive while the right for resistive target. Unstructured triangular mesh is implemented in pyGIMLi, BERT, and ResIPy, whereas tensor mesh is used in SimPEG

the other hand, target anomalies are not reproduced in ResIPy. As the target depth increases, significant resistivity overestimations (42–44%) have occurred in conductive anomaly zones, while underestimations (35–39%) are observed in resistive target anomaly zones for all the tested inversion freeware (Fig. 11).

Moreover, we examine the inversion freeware effectiveness using different target sizes, such as 0.25, 0.5, 1, and 2 m radius, all set at 3 m survey depth. Target sizes with 0.25 and 0.5 m radius are not recovered in all the tested freeware packages. Since the 1 m target radius is discussed in the earlier section as scenario two (Fig. 10), the inverted models for the 2 m target radius are presented as scenario four (as shown in Fig. 12). pyGIMLi, BERT, and SimPEG show high resolution for the inverted model of scenario four, while a relatively low resolution is exhibited in ResIPy (Fig. 12). The target geometries are accurately resolved in pyGIMLi and BERT. The upper boundary of the target anomaly in ResIPy is adequately recovered, whereas broadening occurs at the lower boundary. Its anomaly resolution is noticeably reduced as the survey depth increases. SimPEG yields a relatively broader target anomaly than the actual target. The model resolution in all the inversion freeware packages is significantly enhanced as the target size increases (Fig. 12). Inversion in scenario four overestimates about 11–18% of the obtained resistivity for the conductive target while underestimates about 4–9% for the resistive target. The larger target size of scenario four shows lower underestimation and overestimation in the obtained resistivity compared to the smaller target size of the same depth scenario. We also quantify the model accuracy of the freeware packages for scenario four as an example. The model accuracy is obtained by summing up the square of the difference between the inverted resistivity and the true model resistivity. The result shows a model accuracy of 21,051.37 for BERT, 22,854.23 for pyGIMLi, 25,024.35 for SimPEG, and 28,135.69 for ResIPy, indicating direct association with the inverted model resolution, as displayed in Fig. 12.

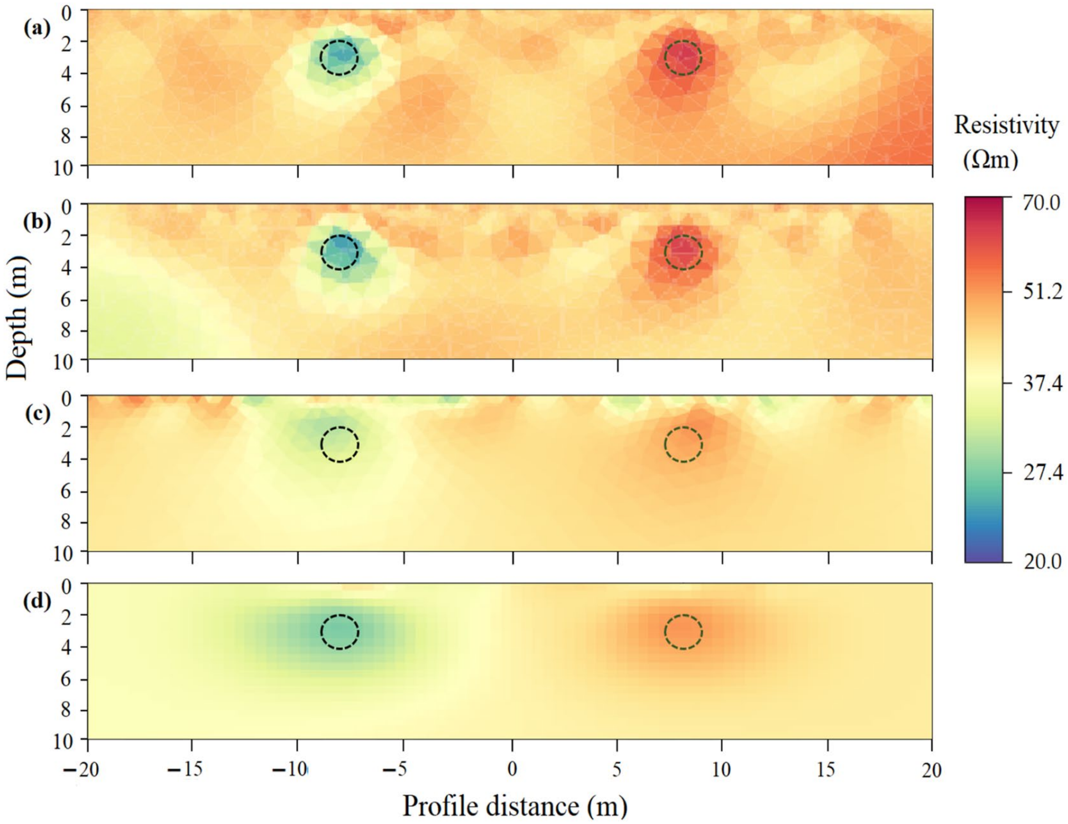

Fig. 10 The recovered model of scenario two for a 1 m target radius situated at 3 m depth applying the inversion freeware: **a** pyGIMLi, **b** BERT, **c** ResIPy, and **d** SimPEG. The broken circles show the actual targets; the left for conductive while the right for resistive target. pyGIMLi, BERT, and ResIPy use unstructured triangular mesh, while SimPEG implements tensor mesh

Discussion and conclusions

Inversion freeware performances are exclusively intended on how well the inverted model can estimate the subsurface features since no algorithms can recover complete realism. We review and discuss an open-source inversion freeware for geoelectric data that are developed under Python interface, including pyGIMLi, SimPEG, BERT, ResIPy, and *pyres*. The capability of inversion freeware may vary based on the code structure, mesh design, and type of dependency. In addition, this study examines the recovering performances of pyGIMLi, SimPEG, BERT, and ResIPy freeware by inverting the same input models. A conceptual model with low and high resistive targets sets at different depths of the homogeneous host medium is used for numerical experiments.

Table 2 shows the application of tested inversion freeware for reproducible researches. The most versatile pyGIMLi package is effectively applied for various geophysical data inversion (Rücker et al. 2017). The framework development of SimPEG is focused on modularity, usability, documentation, and extensive unit testing, which can be used as an interface to create new inversion codes (Cockett et al. 2015). pyGIMLi and SimPEG are mainly applied for inversion of geoelectric, travel time tomography, potential field, and electromagnetic data. They also perform joint inversion of direct current resistivity and magnetotelluric data to enhance model resolution. Moreover, pyGIMLi specifically performs a petrophysical coupled inversion of travel time and resistivity imaging data, which combines the benefit of the two methods to quantify water saturation. Thus, in addition to geoelectrical data modelling and inversion, pyGIMLi and SimPEG can be implemented for a wide range of geophysical studies.

In contrast, BERT, *pyres*, and ResIPy are mainly implemented for modelling and inversion of geoelectrical data. BERT applies similar code design and retrieval with pyGIMLi; thus, it can import pyGIMLi tools to facilitate the forward and inverse problems. Furthermore, the highly flexible mesh of BERT freeware can incorporate complex topographic features and subsurface structures (Audebert et al. 2014; Günther et al. 2006). The same mesh design and inversion algorithm are used in *pyres* and ResIPy. *pyres* uses scripting language, while

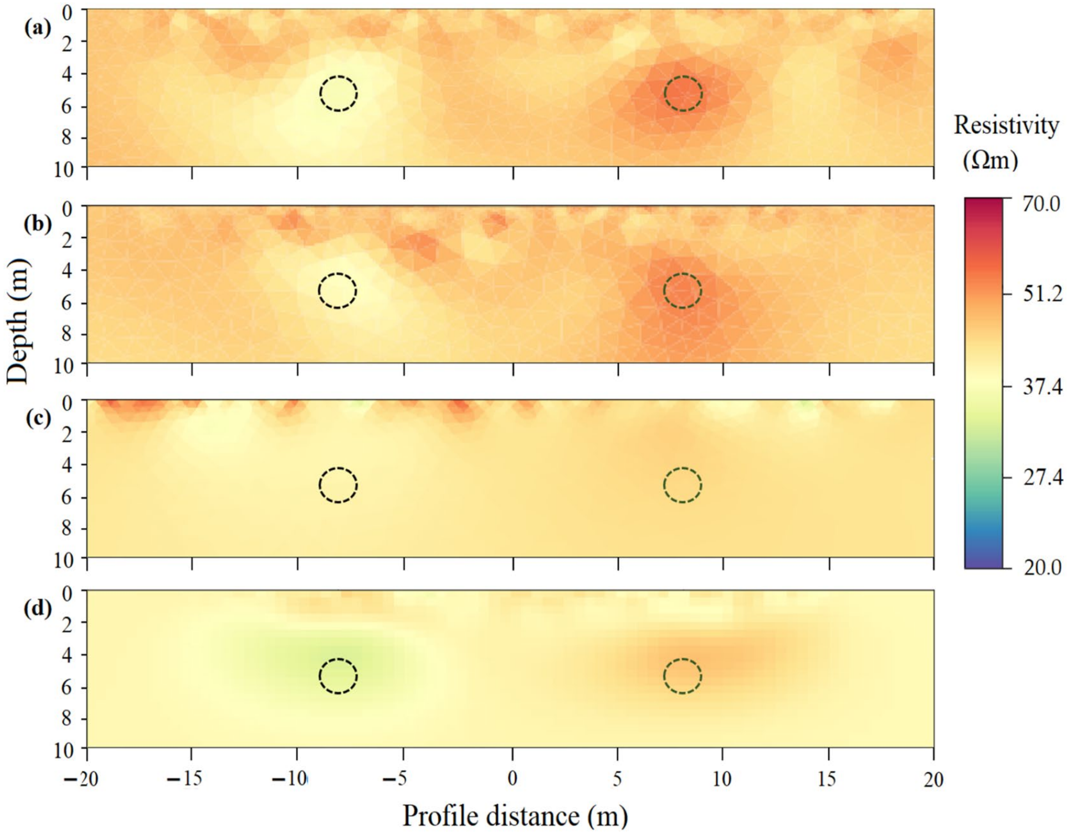

Fig. 11 The inverted model of scenario three for a 1 m target radius located at 5 m depth using the inversion freeware: **a** pyGIMLi, **b** BERT, **c** ResIPy, and **d** SimPEG. The broken circles indicate the actual targets; the left for conductive while the right for resistive target. Unstructured triangular mesh is implemented in pyGIMLi, BERT, and ResIPy, whereas tensor mesh is used in SimPEG

ResIPy applies a graphical user interface (GUI). The GUI in ResIPy removes the need for text inputs and assists the user in pre- and post-processing stages. In contrast, *pyres* is more suitable for inversion of larger datasets than ResIPy (Blanchy et al. 2020). However, its code is less extensible and flexible compared to the other tested freeware codes.

The tested inversion freeware varies based on the mesh design and forward formulations. For instance, SimPEG implements the finite volume of structured and semi-structured meshes, while pyGIMLi applies structured and unstructured meshes using the finite element and finite volume methods. On the other hand, finite element of unstructured mesh is used in BERT, whereas pyres and ResIPy implement structured and unstructured meshes using the finite element method. However, the structured mesh requires less memory and saves computational time; it is less flexible for numerical formulation in the gradient direction compared to the unstructured mesh (Sack and Urrutia 1999). The unstructured mesh refines a grid size in the vicinity to the electrode location and target zone where strong contrast of the simulated electric potential requires enhanced numerical accuracy and its grid size enlarges towards the computational boundaries to account for numerical calculation of low sensitive zone (Blanchy et al. 2020; Nguyen et al. 2005; Rücker et al. 2006). Besides, a quality checker in pyGIMLi and BERT can control the gridding error of unstructured mesh. Since the unstructured mesh is superior in representing complicated subsurface structures, topography, and complex acquisition, it is primarily applied for geoelectrical modelling and inversion in pyGIMLi, BERT, *pyres*, and ResIPy freeware packages. Thus, using the inversion freeware with flexible mesh type can enhance resistivity distributions and numerical calculations, consistent with other studies (Sack and Urrutia 1999; Tomita et al. 2012).

We evaluate the efficiency of the tested inversion freeware as the computational cost plays an essential role in geophysical modelling. Using a CORE i5 Intel computer with 20 GB random access memory (RAM), the processing time per cell takes 13 ms for SimPEG, 16 ms for ResIPy, 29 ms for BERT, and 38 ms for pyGIMLi. This indicates that SimPEG and ResIPy are computationally the most efficient, BERT is moderately

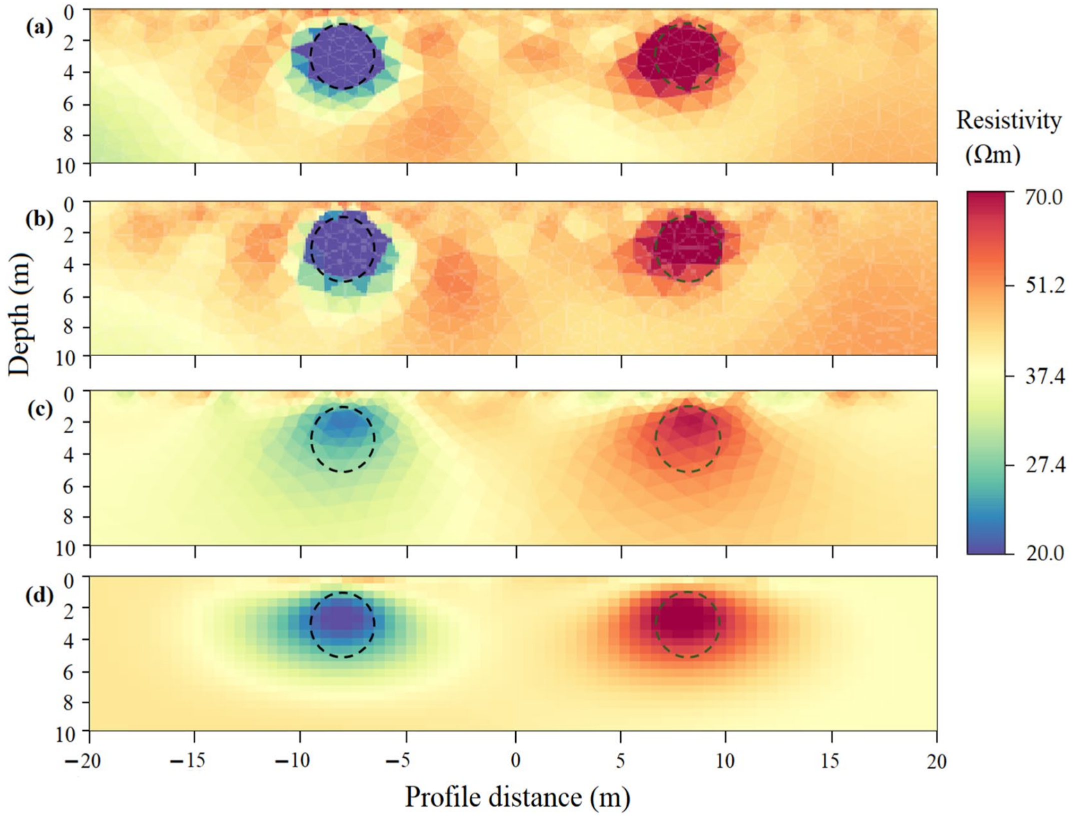

Fig. 12 The recovered model of scenario four for a 2 m target radius located at 3 m depth applying the inversion freeware: **a** pyGIMLi, **b** BERT, **c** ResIPy, and **d** SimPEG. The broken circles show the actual targets; the left for conductive while the right for resistive target. pyGIMLi, BERT, and ResIPy use unstructured triangular mesh, while SimPEG implements tensor mesh

Table 2 Implementation of inversion freeware for reproducible geophysical studies

Freeware	Applications	Key References
pyGIMLi	ERT/IP	(Günther and Martin 2016; Klingler et al. 2020; Nickschick et al. 2019)
	EM/MT	(Audebert et al. 2014; Rochlitz et al. 2019)
	Travel time tomography	(Whiteley et al. 2020)
	Joint/coupled inversion	(Jordi et al. 2020; Rochlitz et al. 2019; Rücker et al. 2017)
BERT	ERT/ IP	(Carriere et al. 2015; Gourdol et al. 2018; Udphuay et al. 2011)
pyres/ResIPy	ERT/IP	(Benjamin et al. 2020; Blanchy et al. 2020)
SimPEG	ERT/IP	(Mitchell 2020; Simpson and Heinson 2020)
	EM/MT	(Heagy et al. 2017; Kang et al. 2015; Oldenburg et al. 2020)
	Joint inversion	(Astic et al. 2021; Kang et al. 2018)

efficient, whereas pyGIMLi is the least efficient. The tested freeware packages can adequately work on a computer with more than 4 GB RAM. However, the more extensive models can require a higher RAM computer; for instance, the experimental model in Fig. 8 requires a computer processor greater than 6 GB RAM.

Effective calculation of forward resistivity dataset can play a vital role in the quality of the inverted model (McGillivray 1992). The numerical resistivity variations among the pyGIMLi, BERT, and ResIPy are examined for different grid sizes. Slight resistivity discrepancies are shown among the tested freeware, mainly for small grid sizes. Moreover, the anomaly effect shows the variation of freeware effectiveness for forward resistivity formulations. The anomaly effect of the forward resistivity dataset is less considerably varied for the tested freeware packages: pyGIMLi, BERT, SimPEG, and ResIPy. The anomaly effect for different electrode arrays and geologic conditions are noticeably varied (Dahlin and Zhou 2004). This study exhibits a significant change of anomaly effect for the target size and depth. The increase in target size can increase the anomaly effect. As target depth increases, the anomaly effect of the forward resistivity results of the tested freeware is decreased. This is likely associated with the measurement sensitivity of the electrode array. High measurement sensitivity of an array near the electrode position might enable a high anomaly effect than far electrode location (Doyoro et al. 2021; Loke 2013).

The inverted model resolution shows variation with regard to the target size and survey depth in addition to the type of freeware used. Our numerical experiment shows that the larger anomaly targets are correctly resolved in all the tested inversion freeware packages (Fig. 12), yet they do not display any signatures for the target radius less than 0.5 m set at 3 m depth. Moreover, the tested freeware recovery effectiveness is varied based on the target depth-to-dimension ratio. pyGIMLi, BERT, and SimPEG can recover a target diameter larger than one-fourth of its depth. On the other hand, ResIPy can reproduce a target diameter larger than one-third times its depth. The tested inversion freeware may detect relatively deeper targets if the anomaly geometry is not considered.

The decrease in model resolution and considerable anomaly size overestimations have occurred as the buried target depth increases. One factor explaining this limit is the declining measurement sensitivity of the surface resistivity probing. This direct consequence of fundamental physics law cannot be improved unless cross-hole resistivity imaging can be implemented (Barker 1989; Loke 2013). The second factor is decreasing of measured data density as the survey depth increases (Nguyen et al. 2005; Oldenburg and Li 1999). The measured resistivity data have an insignificant influence on the model inversion at greater depth, leading to noticeable uncertainty. The third factor is the smoothing effect of the freeware inversion algorithm (Portniaguine and Zhdanov 1999). It regularizes high resistivity contrast between the target and host medium, limiting the recovery of accurate anomaly geometry. Because of the inversion regularization, the inverted resistivity data are considerably overestimated for the conductive target while underestimated for the resistive target, particularly at deeper depths. The increase in the target size of the subsurface structure can substantially minimize the smoothness effect of the freeware inversion processes.

There are challenges in comparing different resistivity inversion freeware packages. The difference in discretization algorithms could create variations in the recovered models (Hellman et al. 2016). For example, comparing the inversion freeware packages with finite element and finite difference methods may significantly change the inverted model results. Furthermore, using the same colour scale for obtained models of different freeware packages can considerably affect the resolution, particularly in ResIPy and SimPEG. Despite those constraints of inversion freeware comparisons, our modelling shows relatively higher resolution in pyGIMLi, BERT, and SimPEG than the other tested freeware packages. In the same manner, the numerical resistivity data inversion by pyGIMLi (Jordi et al. 2020) and field data inversion by BERT display high resolutions (Flechsig et al. 2010).

Our review and numerical experiment can identify suitable freeware for geoelectric data inversion. Comparatively, all the tested inversion freeware packages could resolve the subsurface features having a size greater than or equal to the target size in scenario four. pyGIMLi and BERT could also be effective for inverting small-scale targets with smaller than or equal to the anomaly size in scenario two. The small-scale target could also be reproduced in SimPEG; however, the model resolutions and anomaly accuracies are considerably reduced. Horizontally stratified modelled layers are adequately recovered in all the tested freeware except ResIPy. Overall experimental results show that pyGIMLi, BERT, and SimPEG freeware packages could be more effective for geoelectrical data inversion. We hope that this study will promote and inspire reproducible research and communication of Python-based inversion freeware to help tackle some of the inherently multidisciplinary geophysical problems in addition to geoelectrical datasets.

Acknowledgements

The authors would like to thank the Minister of Science and Technology (MOST) of Taiwan for supporting this study under project number MOST 108-2638-E-008-001-MY2. This work is also partly supported by Earthquake-Disaster, Risk Evaluation and Management Centre, E-DREaM from The Featured Areas Research Centre Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan. The comments and suggestions from anonymous reviewers have significantly improved the paper.

Authors' contributions

YG contributed to conceptualization, methodology, and draft manuscript; PC contributed to conceptualization, review and modify, validation, and supervision; JM, DJ, TV, DA, and MS were involved in methodology, review, and editing.

Funding

Ministry of Science and Technology, Taiwan, MOST 108-2638-E-008-001-MY2, Earthquake-Disaster, Risk Evaluation and Management Centre, E-DREaM from

The Featured Areas Research Centre Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan.

Availability of data and materials

The data are available upon request to the authors.

Declarations

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Department of Earth Science, National Central University, Taoyuan, Taiwan. [2]Earth System Science, Taiwan International Graduate Program (TIGP), Academia Sinica, Taipei, Taiwan. [3]Earthquake-Disaster, Risk Evaluation and Management Centre, National Central University, Taoyuan, Taiwan. [4]Department of Applied Geology, School of Natural Sciences, Adama Science and Technology University, Adama, Ethiopia.

Received: 27 August 2021 Accepted: 5 January 2022
Published online: 13 January 2022

References

Aizebeokhai A, Olayinka A, Technology (2010) Anomaly effects of arrays for 3d geoelectrical resistivity imaging using orthogonal or parallel 2d profiles. AJEST 4(7):446–454

Arjwech R, Everett ME (2019) Electrical resistivity tomography at construction sites in northeast Thailand with implications for building foundation design. J Environ Eng Geophys 24(2):333–340. https://doi.org/10.2113/JEEG24.2.333

Astic T, Heagy LJ, Oldenburg DW (2021) Petrophysically and geologically guided multi-physics inversion using a dynamic Gaussian mixture model. Geophys J Int 224(1):40–68. https://doi.org/10.1093/gji/ggaa378

Audebert M, Clément R, Grossin-Debattista J, Günther T, Touze-Foltz N, Moreau S (2014) Influence of the geomembrane on time-lapse ERT measurements for leachate injection monitoring. J Waste Manag 34(4):780–790. https://doi.org/10.1016/j.wasman.2014.01.011

Ayachit U (2015) The paraview guide: a parallel visualization application. Kitware Inc, New York

Barker R (1989) Depth of investigation of collinear symmetrical four-electrode arrays. Geophys 54(8):1031–1037. https://doi.org/10.1190/1.1442728

Befus KM (2018) pyres: a Python wrapper for electrical resistivity modeling with R2. J Geophys Eng 15(2):338–346. https://doi.org/10.1088/1742-2140/aa93ad

Benjamin M, Peruzzo L, Boaga J, Cenni N, Schmutz M, Wu Y, Hubbard SS, Cassiani G (2020) Time-lapse monitoring of root water uptake using electrical resistivity tomography and mise-à-la-masse: a vineyard infiltration experiment. Soil 6(1):95–114. https://doi.org/10.5194/soil-6-95-2020

Binley A, Kemna A (2005) DC resistivity and induced polarization methods, Hydrogeophysics. Springer, Berlin, pp 129–156

Blanchy G, Saneiyan S, Boyd J, McLachlan P, Binley A (2020) ResIPy, an intuitive open source software for complex geoelectrical inversion/modeling. Comput and Geosci 137:104423. https://doi.org/10.1016/j.cageo.2020.104423

Bressert E (2012) SciPy and NumPy: an overview for developers. O'Reilly Media Inc, Sebastopol, CA

Caputo R, Piscitelli S, Oliveto A, Rizzo E, Lapenna V (2003) The use of electrical resistivity tomographies in active tectonics: examples from the Tyrnavos Basin, Greece. Geodyn 36(1–2):19–35. https://doi.org/10.1016/S0264-3707(03)00036-X

Cardarelli E, Cercato M, Cerreto A, Di Filippo G (2010) Electrical resistivity and seismic refraction tomography to detect buried cavities. Geophys Prospect 58(4):685–695. https://doi.org/10.1111/j.1365-2478.2009.00854.x

Carriere SD, Chalikakis K, Danquigny C, Clement R, Emblanch C (2015) Feasibility and limits of electrical resistivity tomography to monitor water infiltration through karst medium during a rainy event, Hydrogeological and environmental investigations in Karst systems. Springer, Berlin, pp 45–55

Chambers JE, Kuras O, Meldrum PI, Ogilvy RD, Hollands J (2006) Electrical resistivity tomography applied to geologic, hydrogeologic, and engineering investigations at a former waste-disposal site. Geophys 71(6):B231–B239. https://doi.org/10.1190/1.2360184

Chang P-Y, Liang-Chi C, Teh-Quei L, Yu-Chang C, Huei-Fen C (2015) Examining lake-bottom structures with the resistivity imaging method in Ilan's Da-Hu Lake in Northeastern Taiwan. J Appl Geophys 119:170–177. https://doi.org/10.1016/j.jappgeo.2015.05.016

Chang P-Y, Chang L-C, Hsu S-Y, Tsai J-P, Chen W-F (2017) Estimating the hydrogeological parameters of an unconfined aquifer with the time-lapse resistivity-imaging method during pumping tests: case studies at the Pengtsuo and Dajou sites. Taiwan J Appl Geophys 144:134–143. https://doi.org/10.1016/j.jappgeo.2017.06.014

Cockett R, Kang S, Heagy LJ, Pidlisecky A, Oldenburg DW (2015) SimPEG: An open source framework for simulation and gradient based parameter estimation in geophysical applications. Comput and Geosci 85:142–154. https://doi.org/10.1016/j.cageo.2015.09.015

Dahlin T, Zhou B (2004) A numerical comparison of 2D resistivity imaging with 10 electrode arrays. Geophys Prospect 52(5):379–398

Doyoro YG, Chang P-Y, Puntu JM (2021) Uncertainty of the 2D resistivity survey on the subsurface cavities. J Appl Sci 11(7):3143. https://doi.org/10.3390/app11073143

Du Q, Wang D, Zhu L (2009) On mesh geometry and stiffness matrix conditioning for general finite element spaces. SIAM J Numer Anal 47(2):1421–1444. https://doi.org/10.1137/080718486

Flechsig C, Fabig T, Rücker C, Schütze C (2010) Geoelectrical investigations in the Cheb Basin/W-Bohemia: an approach to evaluate the near-surface conductivity structure. Studia Geophys Geod 54(3):443–463. https://doi.org/10.1007/s11200-010-0026-6

Geuzaine C, Remacle JF (2009) Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities. Int J Numer Meth Eng 79(11):1309–1331. https://doi.org/10.1002/nme.2579

Gourdol L, Clément R, Juilleret J, Pfister L, Hissler C (2018) Large-scale ERT surveys for investigating shallow regolith properties and architecture. Hydrol Earth Syst Sci. https://doi.org/10.5194/hess-2018-519

Günther T, Martin T (2016) Spectral two-dimensional inversion of frequency-domain induced polarization data from a mining slag heap. J Appl Geophys 135:436–448. https://doi.org/10.1016/j.jappgeo.2016.01.008

Günther T, Rücker C, (2015) Boundless Electrical Resistivity Tomography BERT 2–the user tutorial.

Günther T, Rücker C, Spitzer K (2006) Three-dimensional modelling and inversion of DC resistivity data incorporating topography—II. Inversion Geophys J Int 166(2):506–517. https://doi.org/10.1111/j.1365-246X.2006.03010.x

Heagy LJ, Cockett R, Kang S, Rosenkjaer GK, Oldenburg DW (2017) A framework for simulation and inversion in electromagnetics. Comput and Geosci 107:1–19. https://doi.org/10.1016/j.cageo.2017.06.018

Hellman K, Johansson S, Olsson P and Dahlin T, (2016) Resistivity inversion software comparison, Near Surface Geoscience 2016–22nd European Meeting of Environmental and Engineering Geophysics. European Association of Geoscientists & Engineers.

Hunter JD (2007) Matplotlib: A 2D graphics environment. Comput Sci Eng 9(03):90–95. https://doi.org/10.1109/MCSE.2007.55

Hyman JM, Shashkov M (1999) Mimetic discretizations for Maxwell's equations. J Comput Phys 151(2):881–909. https://doi.org/10.1006/jcph.1999.6225

Jordi C, Doetsch J, Günther T, Schmelzbach C, Maurer H, Robertsson J (2020) Structural joint inversion on irregular meshes. Geophys J Int 220(3):1995–2008. https://doi.org/10.1093/gji/ggz550

Kang S, Cockett R, Heagy LJ and Oldenburg DW, (2015) Moving between dimensions in electromagnetic inversions, SEG Technical Program Expanded Abstracts 2015. Society of Exploration Geophysicists, pp. 5000–5004.

Kang S, Fournier D, Werthmuller D, Heagy LJ and Oldenburg D, (2018) SimPEG-EM1D: gradient-based 1D inversion software for large-scale airborne electromagnetic data, AGU Fall Meeting Abstracts, pp. NS53A-0557.

Keller GV (2017) Electrical properties of rocks and minerals. CRC Press

Klingler S, Leven C, Cirpka OA, Dietrich P (2020) Anomaly effect-driven optimization of direct-current geoelectric mapping surveys in large areas. J Appl Geophys 176:104002. https://doi.org/10.1016/j.jappgeo.2020.104002

Lehe R, Huebl A, Vay J-L, Friedman A, Thévenet M, Mitchell C, Bruhwiler D, Grote D, Cowan B and Vincenti H (2020) Embracing modern software tools and user-friendly practices, when distributing scientific codes. In: Snowmass21 LOI

Lin JW (2012) Why Python is the next wave in earth sciences computing. Bull Am Meteorol Soc 93(12):1823–1824. https://doi.org/10.1175/BAMS-D-12-00148

Lin C-P, Hung Y-C, Yu Z-H, Wu P-L (2013) Investigation of abnormal seepages in an earth dam using resistivity tomography. Geoengin 8(2):61–70. https://doi.org/10.6310/jog.2013.8(2).4

Loke M, (2013) Tutorial: 2-D and 3-D electrical imaging surveys. Geotomo Software Malaysia. Unpublished.

McGillivray PR (1992) Forward modeling and inversion of DC resistivity and MMR data. University of British Columbia, Vancouver

Militzer H, Rösler R, Lösch W (1979) Theoretical and experimental investigations for cavity research with geoelectrical resistivity methods. Geophys Prospect 27(3):640–652

Mitchell MA (2020) Methodologies for the use of electrical and electromagnetic methods in complex, subterranean environments. University of British Columbia, Vancouver

Narayan S, Dusseault MB, Nobes DC (1994) Inversion techniques applied to resistivity inverse problems. Inverse Probl 10(3):669

Neumaier A (1998) Solving ill-conditioned and singular linear systems: A tutorial on regularization. J Soc Ind Appl Math 40(3):636–666. https://doi.org/10.1137/S0036144597321909

Nguyen F, Garambois S, Jongmans D, Pirard E, Loke M (2005) Image processing of 2D resistivity data for imaging faults. J Appl Geophys 57(4):260–277. https://doi.org/10.1016/j.jappgeo.2005.02.001

Nickschick T, Flechsig C, Mrlina J, Oppermann F, Löbig F, Günther T (2019) Large-scale electrical resistivity tomography in the Cheb Basin (Eger Rift) at an International Continental Drilling Program (ICDP) monitoring site to image fluid-related structures. Solid Earth 10(6):1951–1969. https://doi.org/10.5194/se-10-1951-2019

Okpoli CC (2013) Sensitivity and resolution capacity of electrode configurations. Geophys J Int. https://doi.org/10.1155/2013/608037

Oldenburg DW, Li Y (1999) Estimating depth of investigation in dc resistivity and IP surveys. Geophys 64(2):403–416. https://doi.org/10.1190/1.1444545

Oldenburg DW, Heagy LJ, Kang S, Cockett R (2020) 3D electromagnetic modelling and inversion: a case for open source. J Expl Geophys 51(1):25–37. https://doi.org/10.1080/08123985.2019.1580118

Peng RD (2011) Reproducible research in computational science. Science 334(6060):1226–1227. https://doi.org/10.1126/science.1213847

Portniaguine O, Zhdanov MS (1999) Focusing Geophysical Inversion Images. Geophys 64(3):874–887. https://doi.org/10.1190/1.1444596

Rochlitz R, Skibbe N, Günther T (2019) custEM: customizable finite-element simulation of complex controlled-source electromagnetic data. Geophys 84(2):F17–F33. https://doi.org/10.1190/geo2018-0208.1

Rücker C (2010) Advanced electrical resistivity modelling and inversion using unstructured discretization. Universität Leipzig, Leipzig

Rücker C, Günther T, Spitzer K (2006) Three-dimensional modelling and inversion of dc resistivity data incorporating topography—I. Modelling Geophys J Int 166(2):495–505

Rücker C, Günther T, Wagner FM (2017) pyGIMLi: An open-source library for modelling and inversion in geophysics. Comput and Geosci 109:106–123. https://doi.org/10.1016/j.cageo.2017.07.011

Sack J-R, Urrutia J (1999) Handbook of computational geometry. Elsevier

Seaton WJ, Burbey TJ (2002) Evaluation of two-dimensional resistivity methods in a fractured crystalline-rock terrane. J Appl Geophys 51(1):21–41. https://doi.org/10.1016/S0926-9851(02)00212-4

Sharma S, Verma GK (2015) Inversion of electrical resistivity data: a review. J Comput Sci Eng 9(4):400–406. https://doi.org/10.5281/zenodo.1106169

Shewchuk JR, (1996) Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator, Workshop on Applied Computational Geometry. Berlin: Springer, pp. 203–222.

Si H (2015) TetGen, a Delaunay-based quality tetrahedral mesh generator. ACM Trans Math Softw 41(2):1–36. https://doi.org/10.1145/2629697

Simpson JM, Heinson G (2020) Synthetic modelling of downhole resistivity data to improve interpretation of basin morphology from magnetotelluric inversion. Earth Planets Space 72:1–21. https://doi.org/10.1186/s40623-020-01178-2

Tikhonov AN, Goncharsky A, Stepanov V, Yagola AG (2013) Numerical methods for the solution of ill-posed problems. Springer Science and Business Media, Berlin, p 328

Tomita JT, da Silva LM, da Silva DT (2012) Comparison between unstructured and structured meshes with different turbulence models for a high pressure turbine application. Power for Land, Sea, and Air. American Society of Mechanical Engineers, Turbo Expo, pp 1633–1645

Udphuay S, Günther T, Everett ME, Warden RR, Briaud J-L (2011) Three-dimensional resistivity tomography in extreme coastal terrain amidst dense cultural signals: application to cliff stability assessment at the historic D-Day site. Geophys J Int 185(1):201–220. https://doi.org/10.1111/j.1365-246X.2010.04915.x

Van Schoor M (2002) Detection of sinkholes using 2D electrical resistivity imaging. J Appl Geophys 50(4):393–399

Whiteley J, Chambers J, Uhlemann S, Boyd J, Cimpoiasu M, Holmes J, Inauen C, Watlet A, Hawley-Sibbett L, Sujitapan C (2020) Landslide monitoring using seismic refraction tomography—the importance of incorporating topographic variations. J Eng Geol 268:105525. https://doi.org/10.1016/j.enggeo.2020.105525

Wilson G, Aruliah DA, Brown CT, Hong NPC, Davis M, Guy RT, Haddock SH, Huff KD, Mitchell IM, Plumbley MD (2014) Best practices for scientific computing. PLoS Comput Biol 12(1):e1001745. https://doi.org/10.1371/journal.pbio.1001745

Zhang G, Zhang G-B, Chen C-C, Chang P-Y, Wang T-P, Yen H-Y, Dong J-J, Ni C-F, Chen S-C, Chen C-W (2016) Imaging rainfall infiltration processes with the time-lapse electrical resistivity imaging method. Pure Appl Geophys 173(6):2227–2239

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.