Induced polarization with pyGIMLi and pyBERT

Thomas Günther

Leibniz Institute for Applied Geophysics (LIAG), Hannover, Germany

September 07, 2023

Content

Introduction

- some background and motivation
- presentation of the modules

Lab data

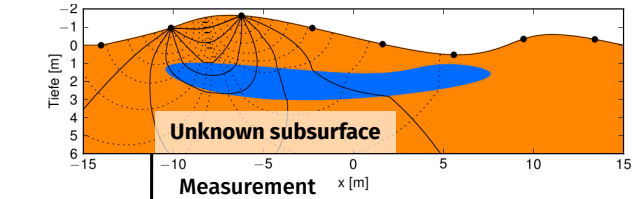
- Read in data & display it
- quality check (+Kramers-Kronig)
- filtering
- fitting Cole-Cole models
- Debye decomposition
- saving and export figures

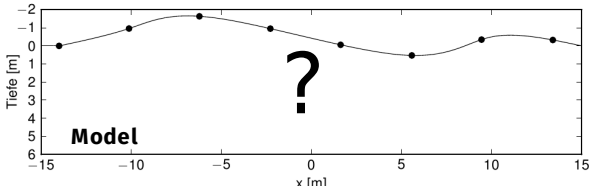
Field data

- Importing data from RES file
- quality check
- filtering and processing
- single-frequency inversion
- multi-frequency inversion
- Cole-Cole parameters

Geophysical imaging: A typical workflow

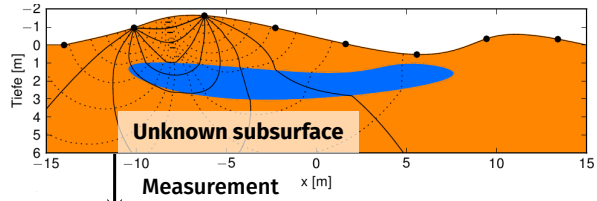
1. Data acquisition
2. Preprocessing (quality check and filtering)
3. Parameterization (i.e., mesh generation)
4. Inversion
5. Evaluate fit between measured & simulated data
6. Postprocessing & visualization of final model(s)
7. Interpretation

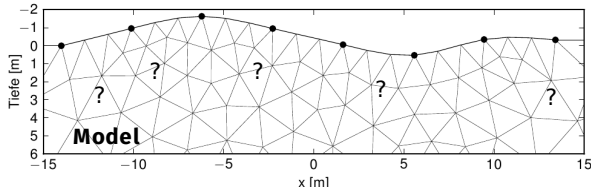




Geophysical imaging: A typical workflow

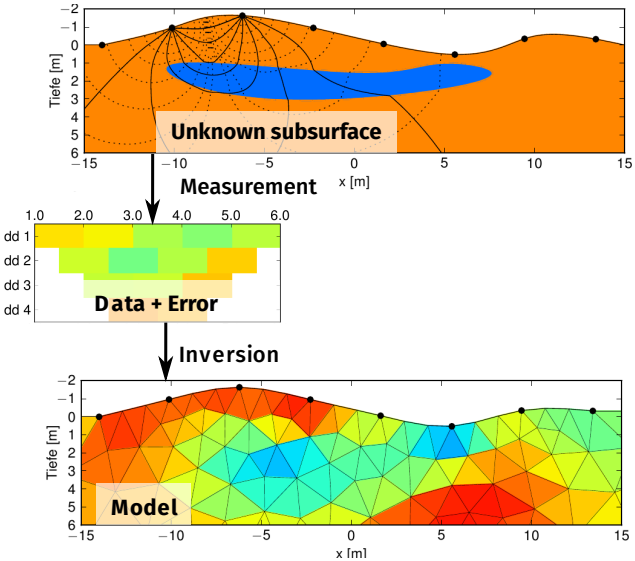
1. Data acquisition
2. Preprocessing (quality check and filtering)
3. Parameterization (i.e., mesh generation)
4. Inversion
5. Evaluate fit between measured & simulated data
6. Postprocessing & visualization of final model(s)
7. Interpretation





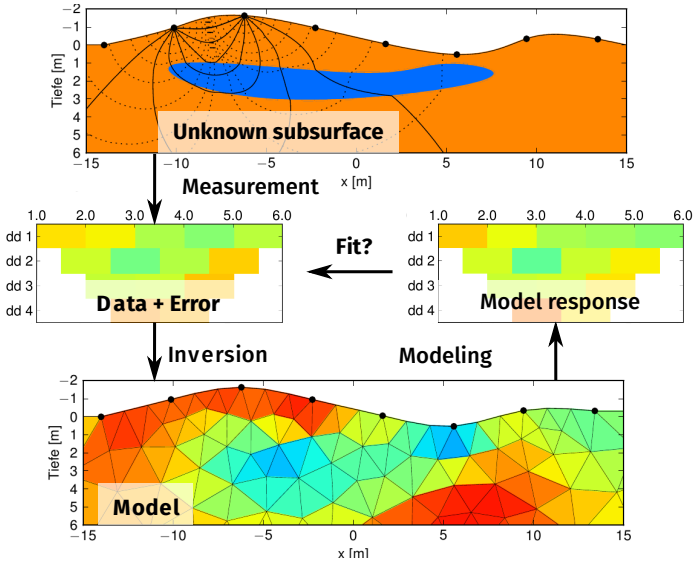
Geophysical imaging: A typical workflow

1. Data acquisition
2. Preprocessing (quality check and filtering)
3. Parameterization (i.e., mesh generation)
4. Inversion
5. Evaluate fit between measured & simulated data
6. Postprocessing & visualization of final model(s)
7. Interpretation



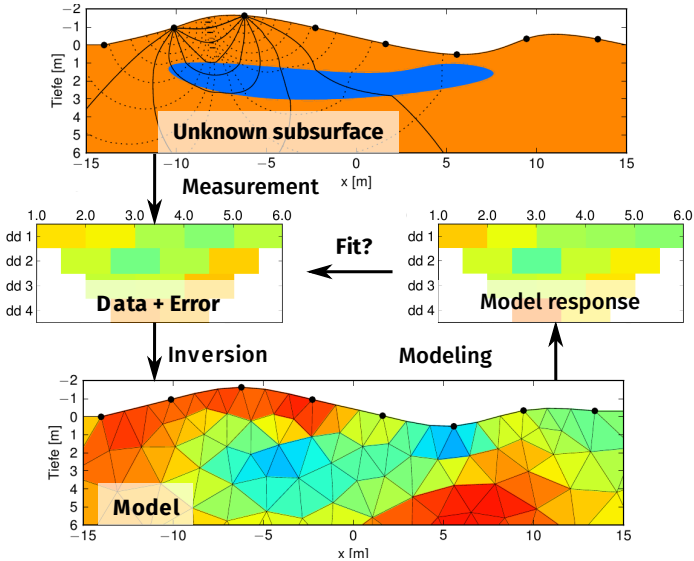
Geophysical imaging: A typical workflow

1. Data acquisition
2. Preprocessing (quality check and filtering)
3. Parameterization (i.e., mesh generation)
4. Inversion
5. Evaluate fit between measured & simulated data
6. Postprocessing & visualization of final model(s)
7. Interpretation



Geophysical imaging: A typical workflow

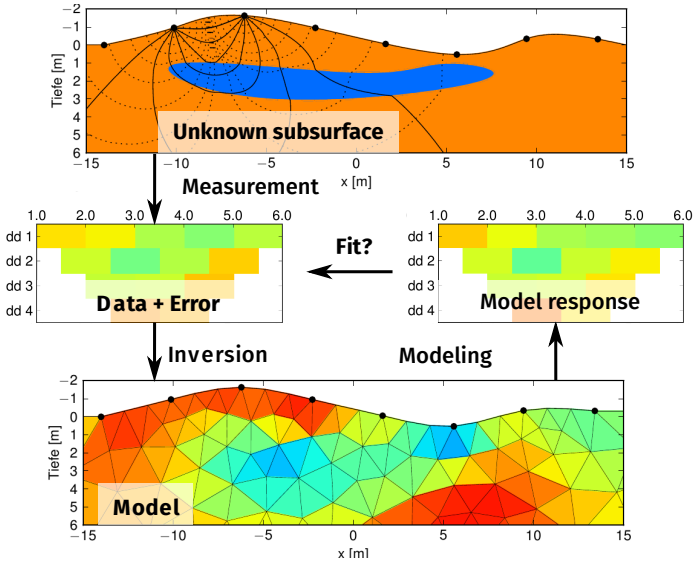
1. Data acquisition
2. Preprocessing (quality check and filtering)
3. Parameterization (i.e., mesh generation)
4. Inversion
5. Evaluate fit between measured & simulated data
6. Postprocessing & visualization of final model(s)
7. Interpretation



Geophysical imaging: A typical workflow

1. Data acquisition
2. Preprocessing (quality check and filtering)
3. Parameterization (i.e., mesh generation)
4. Inversion
5. Evaluate fit between measured & simulated data
6. Postprocessing & visualization of final model(s)
7. Interpretation

limited sensitivity and resolution, inherent (petrophysical) ambiguities

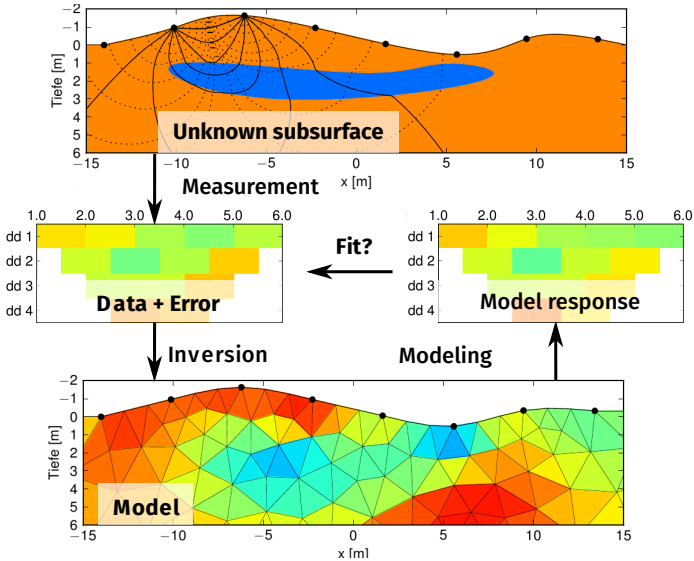


Geophysical imaging: A typical workflow

1. Data acquisition
2. Preprocessing (quality check and filtering)
3. Parameterization (i.e., mesh generation)
4. Inversion
5. Evaluate fit between measured & simulated data
6. Postprocessing & visualization of final model(s)
7. Interpretation

limited sensitivity and resolution, inherent (petrophysical) ambiguities

→ **requires integrating different (geophysical) methods & geology**



pyGIMLi: An open-source library for modelling and inversion in geophysics

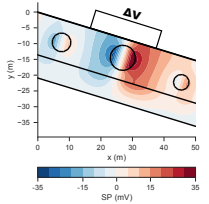
pyGIMLi

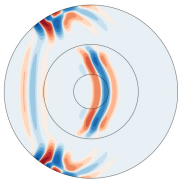
Geophysical Inversion & Modelling Library

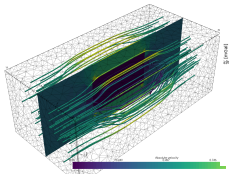
Acknowledge

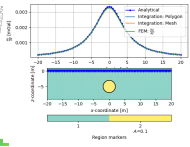
- Dr. Carsten Rücker (TU Berlin)
- Dr. Florian Wagner (RWTH Aachen)

- management of **regular and irregular meshes** in 2D & 3D
- efficient **finite-element and finite-volume solvers**
- **various geophysical forward operators**
- Gauss-Newton frameworks for **constrained, joint and coupled inversions** with **region-specific regularization**
- **open-source, cross-platform**, documented & tested
- well suited for **teaching & reproducible research**









Website & Documentation: `www.pygimli.org`

Rücker, C., Günther, T., Wagner, F.M., 2017. pyGIMLi: An open-source library for modelling and inversion in geophysics, *Computers and Geosciences*, 109, 106-123, doi: 10.1016/j.cageo.2017.07.011.

Why you should use open-source software

- don't reinvent the wheel $\Rightarrow$ fast progress
- you got almost all the tools you wish for
- flexibility to implement you own ideas
- sustainable use in your future (access, licenses)
- create reproducible science (FAIR principles)
- be part of lively community and get helped
- cooperations with other users
- open science through open access and data publications

Links to be used, further reading

- pygimli.org website with examples, tutorials API doc etc.
- Notebook collection on https://github.com/gimli-org/notebooks
  folder IP/Summerschool
- Notebooks from Transform2021 workshop at https://github.com/gimli-org/transform2021
  - associated Youtube video https://youtu.be/w3pu0H3dXe8
- Notebooks from Transform2022 workshop at https://github.com/gimli-org/transform2022
  - associated Youtube video https://youtu.be/2Hu4gDnRzlU
- github project page https://github.com/gimli-org/gimli with Issues etc.
- BERT gitlab project page https://gitlab.com/resistivity-net/bert with examples etc.
- published papers and associated data / scripts, e.g. https://zenodo.org/record/4419736
- NumPy (numerics) https://numpy.org, SciPy (scientific computing) https://scipy.org, Matplotlib (graphics) https://matplotlib.org

The SIPSpectrum module in pyGIMLi

Task

Reading, plotting, filtering of SIP spectra

Access

```
from pygimli.physics import SIPSpectrum
```

- data import (most common formats) and filtering
- synthetic modelling of real or complex conductivity
- ERT manager for handling all processing steps
- error estimation and inversion
- post-processing and export5

The ERT module in pyGIMLi

```
from pygimli.physics import ert
```

- data import (most common formats) and filtering
- synthetic modelling of real or complex conductivity
- ERT manager for handling all processing steps
- error estimation and inversion
- post-processing and export5

What's pyBERT about?

- many more examples
- command-line tools
- importers for a huge number of formats
- multi-frequency data processing (FDIP)
- analysis of whole TDIP decays (TDIP)

The FDIP module in pyBERT

```
from pybert import FDIP
```

- data import and filtering
- synthetic modelling using Cole-Cole model
- ERT manager for handling all processing steps
- error estimation and inversion
- post-processing and export5

The TDIP module in pyBERT

```
from pybert import TDIP
```

- data import and filtering
- synthetic modelling using Cole-Cole model
- ERT manager for handling all processing steps
- error estimation and inversion
- post-processing and export5

Input formats

ERT

Ascii columns (ABEM, Syscal, Resecs), 4-point light, Ares, Syscal, res2dinv, SuperSting, Geotom, ...

FDIP

Radic (SIP256C/D), MPT-DAS1, AarhusInv tx3,

TDIP

TDIP gated data:

- Syscal Pro export file (*.txt)
- ABEM TXT export file (*.txt or raw time series)
- Syscal Pro binary file (*.bin)
- GDD format (*.gdd), Ares II format (*.2dm)
- Aarhus Workbench processed data (*.tx2 and *.dip)
- res2dinv data

References

Günther, T., Rücker, C. & Spitzer, K. (2006): Three-dimensional modeling and inversion of dc resistivity data incorporating topography – II: Inversion. Geophys. J. Int. 166, 506-517, doi:10.1111/j.1365-246X.2006.03011.x.

Rücker, C., Günther, T. & Spitzer, K. (2006): Three-dimensional modeling and inversion of dc resistivity data incorporating topography – I: Modeling. Geophys. J. Int. 166, 495-505, doi:10.1111/j.1365-246X.2006.03010.x.

Rücker, C., Günther, T., Wagner, F.M. (2017): pyGIMLi: An open-source library for modelling and inversion in geophysics, Computers & Geosciences 109, 106-123, doi:10.1016/j.cageo.2017.07.011

Günther, T. & Martin, T. (2016): Spectral two-dimensional inversion of frequency-domain induced polarisation data from a mining slag heap. Journal of Applied Geophysics 135, 436-448, doi:10.1016/j.jappgeo.2016.01.008.

Martin, T., Günther, T., Orozco, A.F. & Dahlin, T. (2020): Evaluation of spectral induced polarization field mea- surements in time and frequency domain, J. Appl. Geophys. 180, 104141, doi:10.1016/j.jappgeo.2020.104141.