AnyPetro - Universal parameter fitting tool for petrophysical laboratory data

DOCUMENTATION

Jana H. Börner[1]*, Volker Herdegen[2]

April 28, 2022

[1] Institute of Geophysics and Geoinformatics, TU Bergakademie Freiberg, Germany
[2] Institute of Thermal, Environmental and Natural Products' Process Engineering,
TU Bergakademie Freiberg, Germany

Abstract

AnyPetro is a MATLAB-based, GUI-controlled software for adjusting the parameters of arbitrary and non-linear petrophysical models to laboratory data. A Gauss-Newton scheme is applied for the minimization of a damped least-squares objective function. Thereby, the Jacobian matrix is calculated explicitly with the perturbation method. Data weighting, model parameter transformations and different regularizations are provided. The petrophysical model, resp., the forward operator is introduced by the user in form of a simple text file based on the specific application's needs. Example data files and forward operators as well as Matlab App and stand-alone installers are provided. The software tool has been developed for and successfully applied to the fitting of various petrophysical data sets (e.g. porosity, specific surface, electrical conductivity, spectral induced polarization) from fluid, unconsolidated, solid and crushed samples to non-linear, multi-parameter models (e.g. electrical CO_2-saline water interaction, Debye decomposition, crushed rock conductivity).

1 Licence

GNU General Public Licence Version 3 (29 June 2007)

2 Citation

When using the software please cite:

Börner, Jana H.; Herdegen, Volker (2022): AnyPetro - Universal parameter fitting tool for petrophysical laboratory data. V. 1.0. GFZ Data Services. https://doi.org/10.5880/fidgeo.2021.039

*Corresponding author: jana.boerner@geophysik.tu-freiberg.de

Contents

1 Licence 1

2 Citation 1

3 Introduction 3

4 How to install 6
- 4.1 MATLAB-App 6
- 4.2 Stand-alone application 7

5 User guide 8
- 5.1 Graphical user interface 8
- 5.2 Input files 8
  - 5.2.1 Data file 8
  - 5.2.2 Forward operator file 9
- 5.3 Output files 11
  - 5.3.1 Output data table 11
  - 5.3.2 Output model table 11
  - 5.3.3 Log file 11
  - 5.3.4 Output figures 12
- 5.4 The inversion workflow 12
  - 5.4.1 Load data file 12
  - 5.4.2 View or edit data table [OPTIONAL] 12
  - 5.4.3 Select column for data use 13
  - 5.4.4 Select data type column 13
  - 5.4.5 Select data column 13
  - 5.4.6 Select data weight column 14
  - 5.4.7 Use $\log_{10}$ of data [OPTIONAL] 14
  - 5.4.8 Select columns with sample specifics [OPTIONAL] 14
  - 5.4.9 Generate data vectors 14
  - 5.4.10 Select forward operator 14
  - 5.4.11 Parameter transformation [OPTIONAL] 15
  - 5.4.12 Regularization [OPTIONAL] 15
  - 5.4.13 Inversion parameters [OPTIONAL] 15
  - 5.4.14 Start inversion 15
  - 5.4.15 Clear command line [OPTIONAL] 16
  - 5.4.16 Export result [OPTIONAL] 16

6 Examples 17
- 6.1 Waxman-Smits model for electrical interface conductivity 17
- 6.2 Wyllie's time-average equation for p-wave velocity 18
- 6.3 Cole-Cole model for frequency-dependent, complex electrical conductivity 19
- 6.4 Debye decomposition 20
- 6.5 Recovering formation properties from measurements on crushed rock 21

7 Appendix 23
- 7.1 Data files 23
- 7.2 Forward operator files 25

3 Introduction

Adjusting the parameters of a petrophysical model so that it reflects measured laboratory data is a common task in geophysical research. This task is mathematically speaking an inverse problem (e.g. Aster et al., 2005; Nocedal & Wright, 2006; Menke, 2018). At the same time, petrophysical models are

- often empirical and therefore various,
- subject to constant (further) development and adaptation in their mathematical form,
- explicitly formulatable functions, but which are
- often non-linear and
- may be complex-valued.

AnyPetro provides an inversion framework (see Fig. 1), which is designed to cope with these peculiarities of petrophysical models and multiple types of data at once. At the same time, the application allows for a highly flexible definition of the petrophysical model in order to support petrophysical model development and data interpretation. *AnyPetro* has been developed alongside with petrophysical research, which is documented in detail in e.g. Börner (2016) and Börner et al. (2022). This introduction is taken from these two sources in most parts.

No matter what petrophysical model is considered, the general task for the inversion is to provide an optimal set of model parameters $\mathbf{m} \in \mathbb{R}^M$ for a given forward operator $\mathbf{A}$ (the petrophysical model) to fit the vector of measured data $\mathbf{d}_{\mathrm{obs}} \in \mathbb{R}^N$. *Model parameters* are those variables in the model that are not known from laboratory analyses (e.g. the cementation exponent in Archies law). Quantities, which are included in the model, but which we know (could be e.g. the porosity of a sample or its pore-water electrical conductivity), we call *sample specifics* $\mathbf{s}$. Model parameters and sample specifics are defined by the user and treated separately by *AnyPetro*.

Since the petrophysical models considered here are explicitly-formulatable systems of equations, the forward calculation $\mathbf{d}_{\mathrm{calc}} = \mathbf{A}(\mathbf{m}, \mathbf{s})$, which is a central part of every inversion procedure, does not require numerical simulation techniques but function evaluations only. *AnyPetro* understands the optimal set of model parameters as that set, which minimizes the following damped least-squares objective function Ψ with the data misfit norm Ψ_{d} and the model norm Ψ_{m} (e.g. Aster et al., 2005):

$$\begin{aligned} \Psi(\mathbf{m}^{\mathrm{t}}) &= \Psi_{\mathrm{d}}(\mathbf{m}^{\mathrm{t}}) + \lambda_0 \Psi_{\mathrm{m}}(\mathbf{m}^{\mathrm{t}}) \\ &= \frac{1}{2} \left\| \mathbf{C}_{\mathrm{d}} \left(\mathbf{d}_{\mathrm{obs}} - \mathbf{A}(\mathbf{m}^{\mathrm{t}}, \mathbf{s}) \right) \right\|^2 + \frac{\lambda_0}{2} \left\| \mathbf{C}_{\mathrm{m}} \left(\mathbf{m}^{\mathrm{t}} - \mathbf{m}^{\mathrm{t}}_{\mathrm{ref}} \right) \right\|^2 \rightarrow \min. \end{aligned} \tag{1}$$

where $\|\cdot\|$ denotes the euclidean norm. In order to enforce a petrophysically meaningful inversion result and to reduce ambiguities, the user can choose to provide a range of allowed values for the model parameters. A meaningful result is achieved by inverting for the transformed model parameters $\mathbf{m}^{\mathrm{t}}$ instead of $\mathbf{m}$ itself (e.g. Rücker, 2011; Wang et al., 2018), with

$$\mathbf{m}^{\mathrm{t}} = \log \left(\frac{\mathbf{m} - \mathbf{m}^{\mathrm{l}}}{\mathbf{m}^{\mathrm{u}} - \mathbf{m}} \right) \tag{2}$$

where $\mathbf{m}^{\mathrm{l}}$ and $\mathbf{m}^{\mathrm{u}}$ are the vectors of the lower and, resp., upper model parameter bounds. The matrices $\mathbf{C}_{\mathrm{d}} \in \mathbb{R}^{N \times N}$ and $\mathbf{C}_{\mathrm{m}} \in \mathbb{R}^{T \times M}$ are the data weighting and regularization matrix, respectively. $\mathbf{m}^{\mathrm{t}}_{\mathrm{ref}}$ denotes a reference model and Ψ_{m} therefore punishes deviations of $\mathbf{m}^{\mathrm{t}}$ from $\mathbf{m}^{\mathrm{t}}_{\mathrm{ref}}$. The regularization parameter λ_0 balances the two norms.

The data weighting matrix $\mathbf{C}_{\mathrm{d}}$ is a diagonal matrix, which reflects the confidence in each data value, e.g. in terms of a data error or data weight. The weights can be provided by the user. The regularization matrix $\mathbf{C}_{\mathrm{m}}$ can contain various constraints on the model, such as model parameter weights or interrelations of model parameters, in terms of a Tikhonov regularization (Tikhonov et al., 1995). In the default case, a unity matrix $\mathbf{C}_0$ is used, which causes models with small model norms to be preferred. In the case of correlated petrophysical parameters, $\mathbf{C}_{\mathrm{m}}$ could be designed to reflect that. *AnyPetro* allows for including first-derivative ($\mathbf{C}_1$) and second-derivative ($\mathbf{C}_2$) operators in addition to $\mathbf{C}_0$, which cause the preferred resulting model to be smooth. The three contributions to $\mathbf{C}_{\mathrm{m}}$ are balanced by the parameters λ_1 and λ_2 so that $\mathbf{C}_{\mathrm{m}}$ reads

$$\mathbf{C}_{\mathrm{m}} = \begin{bmatrix} \mathbf{C}_0 \\ \lambda_1 \mathbf{C}_1 \\ \lambda_2 \mathbf{C}_2 \end{bmatrix} \tag{3}$$

Note that $\mathbf{C}_\mathrm{m}$ has as many columns as there are model parameters (M) and a varying number of rows ($T \geq M$). Since the forward operator $\mathbf{A}$ is non-linear in general, the minimization of Ψ requires a linearization and as a result has to be realized in an iterative manner, where – beginning with a starting model $\mathbf{m}_0^\mathrm{t}$ – a model update $\Delta\mathbf{m}^\mathrm{t}$ is computed in every iteration. For the minimization, *AnyPetro* applies a Gauss-Newton scheme, which requires solving the following system of normal equations in each iteration j (e.g. Aster et al., 2005):

$$\Delta\mathbf{m}_j^\mathrm{t} = \left(\mathbf{J}_j^\mathrm{T}\mathbf{C}_\mathrm{d}^\mathrm{T}\mathbf{C}_\mathrm{d}\mathbf{J}_j + \lambda_0\mathbf{C}_\mathrm{m}^\mathrm{T}\mathbf{C}_\mathrm{m}\right)^{-1}\left(\mathbf{J}_j^\mathrm{T}\mathbf{C}_\mathrm{d}^\mathrm{T}\mathbf{C}_\mathrm{d}\mathbf{\Delta d}_j - \lambda_0\mathbf{C}_\mathrm{m}^\mathrm{T}\mathbf{C}_\mathrm{m}\left(\mathbf{m}_j^\mathrm{t} - \mathbf{m}_\mathrm{ref}^\mathrm{t}\right)\right) \tag{4}$$

where $\Delta\mathbf{d}_j = (\mathbf{d}_\mathrm{obs} - \mathbf{A}(\mathbf{m}_j, \mathbf{s}))$ is the vector of data residuals and $\mathbf{J}_j$ is the sensitivity or Jacobian matrix. $\mathbf{J}_j$ contains the first derivatives of the computed data with respect to the (transformed) model parameters. It has to be computed in every iteration, since it depends on the model itself. Due to the low computational cost of the forward calculation for petrophysical models, $\mathbf{J}_j$ is explicitly computed by *AnyPetro* with the perturbation method (e.g. Nocedal & Wright, 2006). Each column i of $\mathbf{J}_j$ is calculated by:

$$\mathbf{J}_{j,i} = \frac{\mathbf{A}(\mathbf{m}_j^\mathrm{t} + dm_i\mathbf{e}_i, \mathbf{s}) - \mathbf{A}(\mathbf{m}_j^\mathrm{t}, \mathbf{s})}{dm_i} \tag{5}$$

where $\mathbf{A}$ denotes a forward calculation, dm_i is a small perturbation of model parameter i and $\mathbf{e}_i$ is a unity vector of the same length as $\mathbf{m}_j^\mathrm{t}$, which applies dm_i on the i-th entry of $\mathbf{m}^\mathrm{t}$. $\mathbf{J}_j$ has as many columns as there are model parameters and as many rows as data exist. Note that the parameter transformation (Eq. 2) affects the sensitivity matrix, which reflects the partial derivatives with respect to $\mathbf{m}_j^\mathrm{t}$. After each iteration, the model is updated by means of a line search to avoid any overshooting of the solution:

$$\mathbf{m}_{j+1}^\mathrm{t} = \mathbf{m}_j^\mathrm{t} + \tau\Delta\mathbf{m}_j^\mathrm{t} \tag{6}$$

The line search parameter $0 < \tau \leq 1$ is determined in each iteration by an iterative sub-process (e.g. Aster et al., 2005; Scheunert et al., 2016). The iteration is stopped either when the maximum number of allowed iterations is reached or when the objective function stagnates and further iteration does not improve the minimization anymore. Finally, the transformation (Eq. 2) is reversed to get the final vector of model parameters $\mathbf{m}$. Differences in magnitude between data types can be compensated in *AnyPetro* during inversion by using the logarithm of the data vector $\log_{10}(\mathbf{d}_\mathrm{obs})$.

After the actual inversion is complete, a confidence interval can be estimated for the calculated model parameters, if the inverse problem is overdetermined. In that case, the empirical variances of the transformed final model parameters $\mathbf{m}^\mathrm{t}$ can be computed from the data weighting matrix $\mathbf{C}_\mathrm{d}$ and the Jacobian matrix of the final iteration step $\mathbf{J}_\mathrm{final}$ (see Malecki et al., 2020, for details). Those are finally used to compute a rough estimate for the variance of the final results for the target model parameters by means of a first-order Taylor approximation (e.g. Marsden & Wright, 2010). The confidence intervals can be used to evaluate the quality of the *AnyPetro* inversion result.

In the following, the installation, running of an inversion and adaptation to user-defined petrophysical models with *AnyPetro* is explained. It is recommended to run one of the working examples first and compare the results with the reference solutions provided in this documentation and alongside the download of the application.

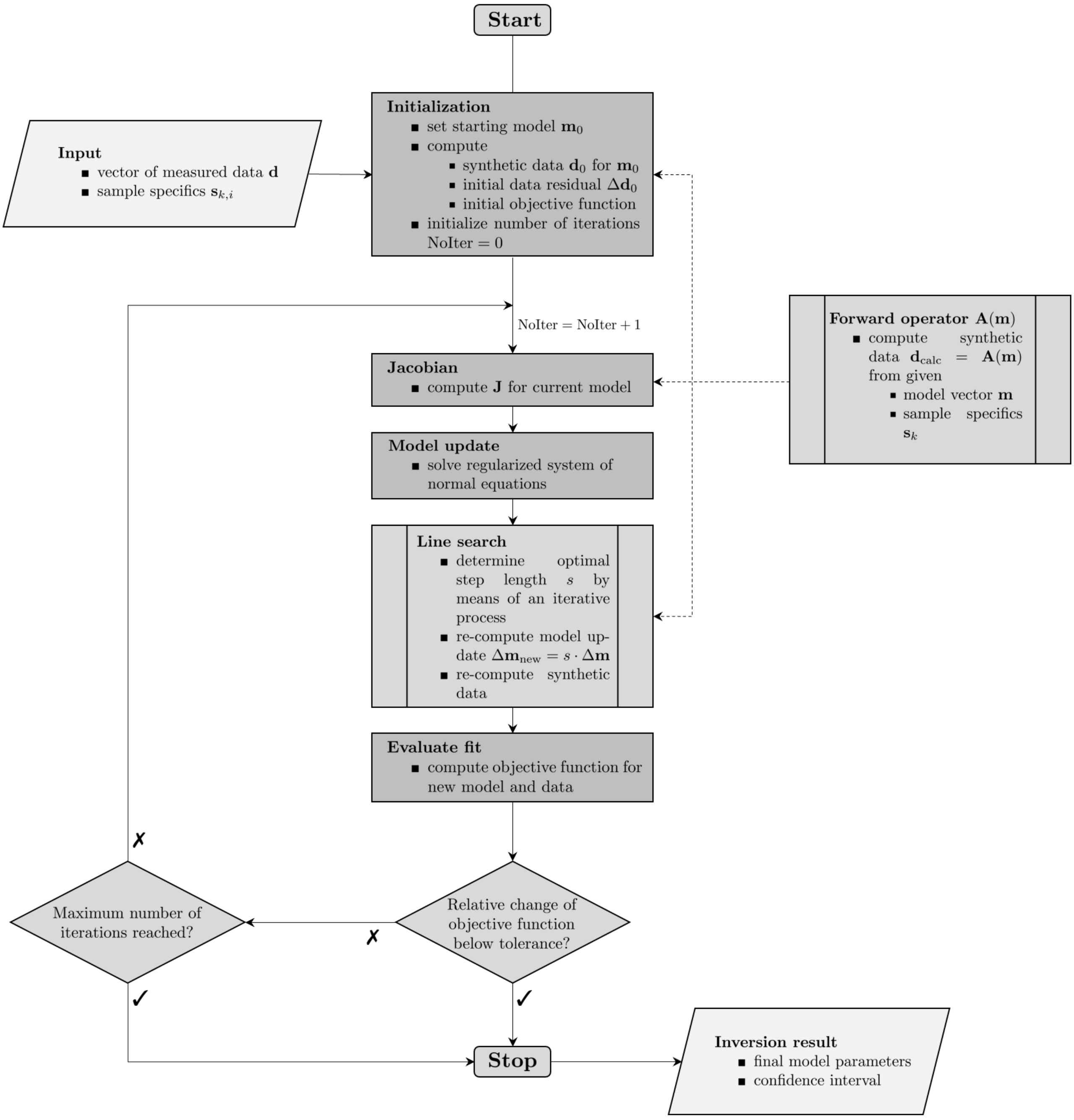

Start
Input
vector of measured data $\mathbf{d}$
sample specifics $\mathbf{s}_{k,i}$
Initialization
set starting model $\mathbf{m}_0$
compute
synthetic data $\mathbf{d}_0$ for $\mathbf{m}_0$
initial data residual $\Delta\mathbf{d}_0$
initial objective function
initialize number of iterations NoIter = 0
NoIter = NoIter + 1
Jacobian
compute $\mathbf{J}$ for current model
Forward operator $\mathbf{A}(\mathbf{m})$
compute synthetic data $\mathbf{d}_{\text{calc}} = \mathbf{A}(\mathbf{m})$ from given
model vector $\mathbf{m}$
sample specifics $\mathbf{s}_k$
Model update
solve regularized system of normal equations
Line search
determine optimal step length s by means of an iterative process
re-compute model update $\Delta\mathbf{m}_{\text{new}} = s \cdot \Delta\mathbf{m}$
re-compute synthetic data
Evaluate fit
compute objective function for new model and data
✗
Maximum number of iterations reached?
Relative change of objective function below tolerance?
✗
✓
✓
Stop
Inversion result
final model parameters
confidence interval

Figure 1: Flowchart of *AnyPetro* for the inversion scheme.

4 How to install

You can download *AnyPetro* from two sources:

- GFZ data services [DOWNLOADLINK]
- GitHub [DOWNLOADLINK].

Please download and unpack the repository. The main directory *AnyPetro* is divided into subfolders as follows:

```
AnyPetro
├─ documentation ........................ contains this pdf user guide
├─ examples ............. input and output files for all examples in Section 6
│  ├─ dataFiles ..................... input data files for all examples
│  ├─ forwardOperators ......... forward operator files for all examples
│  └─ results ....................... output files for all examples
│     ├─ ColeCole ................................. Cole-Cole model
│     ├─ CrushedRock ................ multi-data model for crushed rock
│     ├─ DebyeDecomposition ................... Debye decomposition
│     ├─ WaxmanSmits ........................ Waxman-Smits equation
│     └─ Wyllie ............ Wyllie's equation (time-average equation)
├─ installationFiles ..................................... installation files
│  ├─ MATLAB-App ....... file for installation as MATLAB-APP (MATLAB required)
│  └─ Stand-alone ...... file for stand-alone installation (MATLAB not required)
└─ sourceCode ........ ASCII-readable, MATLAB executable files with the source code
```

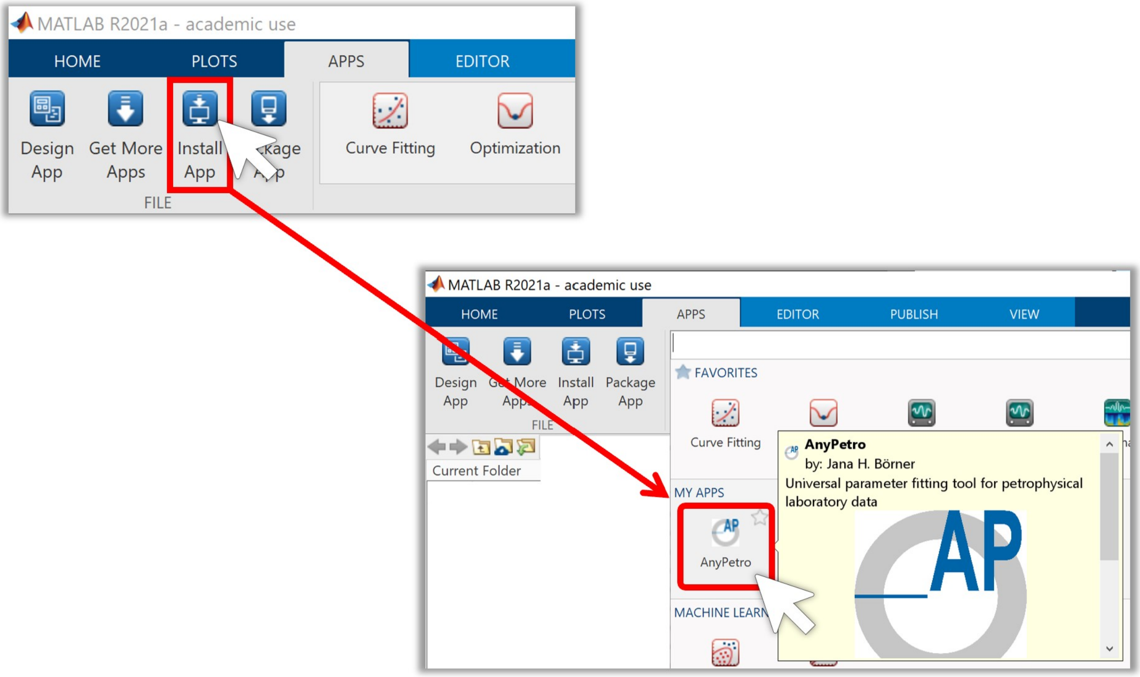
There are two ways for installing *AnyPetro*. If you are a licensed MATLAB user, it is recommended to install *AnyPetro* as a MATLAB-App within the MATLAB development environment. This is platform independent. Alternatively, you can install *AnyPetro* as stand-alone application on your Windows PC. The latter requires the installation of a MATLAB runtime environment, which means that the program takes up more virtual memory.

4.1 MATLAB-App

To install *AnyPetro* as MATLAB-App,

1. open MATLAB
2. go to the *APPS* menu and choose *Install App*
3. browse to the subfolder *MATLAB-App* of the *AnyPetro* directory and select the file *AnyPetro.mlappinstall.*

The app is then installed. When finished, you find *AnyPetro* in the *MY APPS* section of the installed apps, as shown also in the following screen shots:



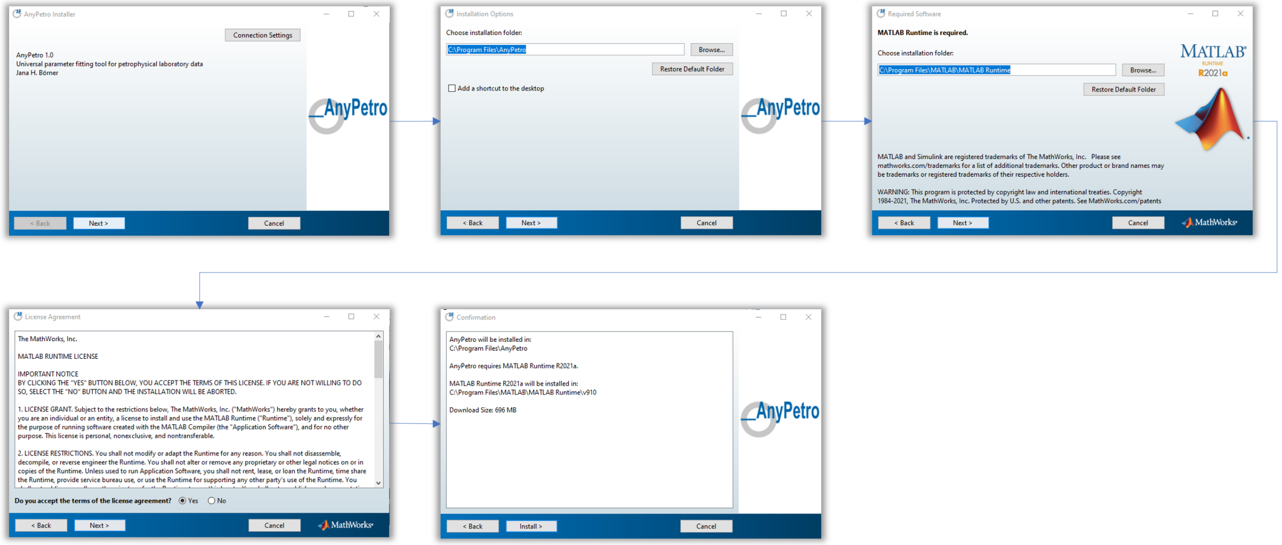
AnyPetro
by: Jana H. Börner
Universal parameter fitting tool for petrophysical laboratory data

4.2 Stand-alone application

To install *AnyPetro* as stand-alone application,

1. browse to the subfolder *Stand-alone* of the *AnyPetro* directory and run the file *AnyPetro_standalone.exe*
2. allow for execution of the installation file
3. follow the instructions (choose installation folder for *AnyPetro*, choose installation folder for *MATLAB runtime*, license agreement).

Note that you need to be connected to the internet to allow for downloading and installing of the runtime engine (696 MB). When finished, you can access *AnyPetro* as every other installed software. Due to the runtime, starting *AnyPetro* can take a while. The installation steps are summarized in the following screen shots:



AnyPetro 1.0
Universal parameter fitting tool for petrophysical laboratory data
Jana H. Börner

Choose installation folder:
C:\Program Files\AnyPetro

□ Add a shortcut to the desktop

MATLAB Runtime is required.

Choose installation folder:
C:\Program Files\MATLAB\MATLAB Runtime

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. Please see mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

WARNING: This program is protected by copyright law and international treaties. Copyright 1984-2021, The MathWorks, Inc. Protected by U.S. and other patents. See MathWorks.com/patents

The MathWorks, Inc.

MATLAB RUNTIME LICENSE

IMPORTANT NOTICE
BY CLICKING THE "YES" BUTTON BELOW, YOU ACCEPT THE TERMS OF THIS LICENSE. IF YOU ARE NOT WILLING TO DO SO, SELECT THE "NO" BUTTON AND THE INSTALLATION WILL BE ABORTED.

1. LICENSE GRANT. Subject to the restrictions below, The MathWorks, Inc. ("MathWorks") hereby grants to you, whether you are an individual or an entity, a license to install and use the MATLAB Runtime ("Runtime"), solely and expressly for the purpose of running software created with the MATLAB Compiler (the "Application Software"), and for no other purpose. This license is personal, nonexclusive, and nontransferable.

2. LICENSE RESTRICTIONS. You shall not modify or adapt the Runtime for any reason. You shall not disassemble, decompile, or reverse engineer the Runtime. You shall not alter or remove any proprietary or other legal notices on or in copies of the Runtime. Unless used to run Application Software, you shall not rent, lease, or loan the Runtime, time share the Runtime, provide service bureau use, or use the Runtime for supporting any other party's use of the Runtime. You

Do you accept the terms of the license agreement? ● Yes ○ No

AnyPetro will be installed in:
C:\Program Files\AnyPetro

AnyPetro requires MATLAB Runtime R2021a.

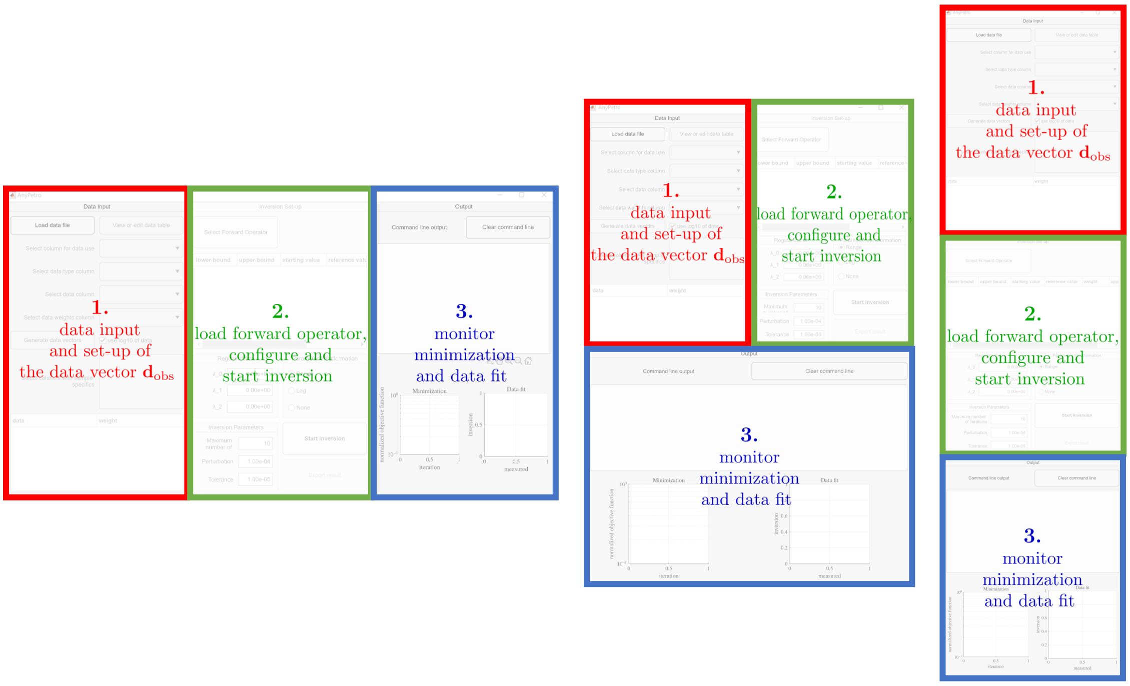
MATLAB Runtime R2021a will be installed in:
C:\Program Files\MATLAB\MATLAB Runtime\v910

Download Size: 696 MB

5 User guide

5.1 Graphical user interface

The graphical user interface of *AnyPetro* is organized in three panels. The panels reorder and adapt automatically to the user's screen aspect ratio. Within the first panel, the data input and the setting up of the data vectors $\mathbf{d}_{\text{obs}}$ is carried out. The second panel contains the set-up of the inversion in terms of the forward operator and the inversion parameters. In the third panel, the inversion progress can be observed in real time by means of a command line window and two figures. The arrangement of the panels is adapted to the screen aspect ratio automatically:

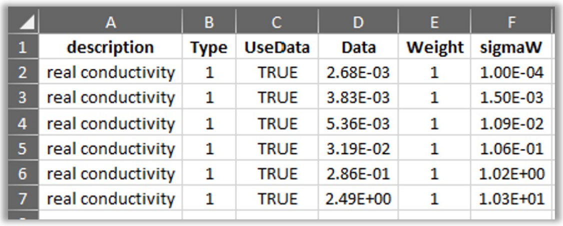


For performing an inversion with *AnyPetro*, the user has to provide the specific petrophysical model (*forward operator*), the parameters of which are supposed to be determined by the inversion, and the measured data to which the model parameters should be fitted. This input is realized with input files. The inversion results as well as the documentation of the minimization process can be exported to output files at the end.

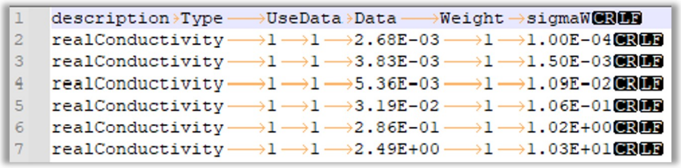
5.2 Input files

5.2.1 Data file

The data input file is organized as a single table, where each row refers to one single data point. A single-row header with user-defined column names is used. The table must have at least four columns (one each for data type, data use, data value and data weight), but there can be any number more. The order of the columns as well as of the rows can be arbitrary. The data table can be provided as *.xlsx file or as tab-separated ASCII file:



	A	B	C	D	E	F
1	description	Type	UseData	Data	Weight	sigmaW
2	real conductivity	1	TRUE	2.68E-03	1	1.00E-04
3	real conductivity	1	TRUE	3.83E-03	1	1.50E-03
4	real conductivity	1	TRUE	5.36E-03	1	1.09E-02
5	real conductivity	1	TRUE	3.19E-02	1	1.06E-01
6	real conductivity	1	TRUE	2.86E-01	1	1.02E+00
7	real conductivity	1	TRUE	2.49E+00	1	1.03E+01



```
description→Type→UseData→Data→Weight→sigmaW CRLF
realConductivity→1→1→2.68E-03→1→1.00E-04CRLF
realConductivity→1→1→3.83E-03→1→1.50E-03CRLF
realConductivity→1→1→5.36E-03→1→1.09E-02CRLF
realConductivity→1→1→3.19E-02→1→1.06E-01CRLF
realConductivity→1→1→2.86E-01→1→1.02E+00CRLF
realConductivity→1→1→2.49E+00→1→1.03E+01CRLF
```

Data type column: Indicate by means of an integer number, what is the data type of the row. Use the same integer identifiers as in the synthetic data calculation block of the forward operator file.

Data use column: Indicate by means of a boolean (for *.xlsx files) or 0 (= false) / 1 (= true) for ACII files whether the data row should be used during inversion or whether the data should be ignored. This allows for storing a whole data set in one table and selecting subsets of data for inversion, e.g. to test the influence of single data points.

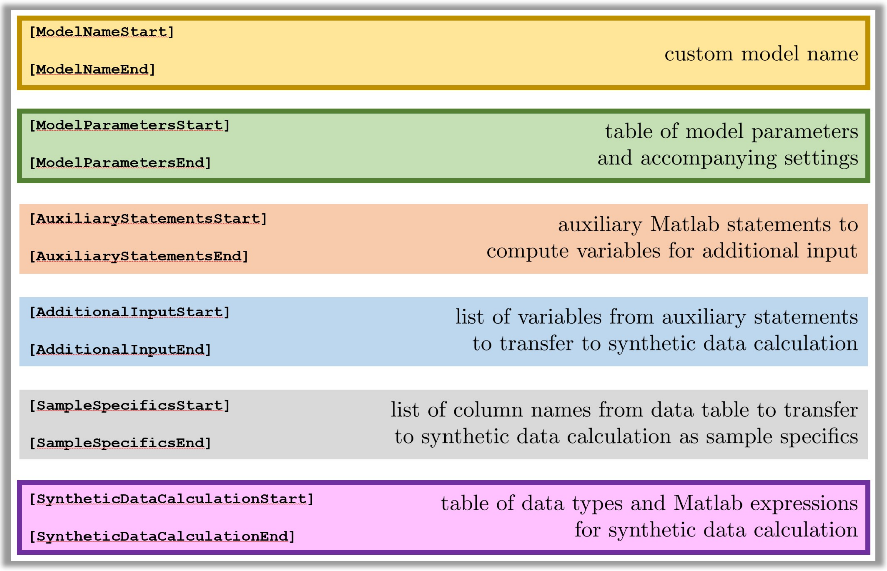
Data column: Give here the actual data value as a single, real-valued number.

Data weight column: Give here the weight that should be used for the data point during inversion. High values put a strong weight to the data and should be used for data with high confidence. The largest weight should be set to 1.0, lower weights defined relative to that.

Additional columns: Additional columns may contain a data description for example or the sample specifics, which are used during inversion later. If multiple data share the same sample specifics, the latter have nevertheless to be given for each data point individually. Surplus columns are ignored during inversion.

5.2.2 Forward operator file

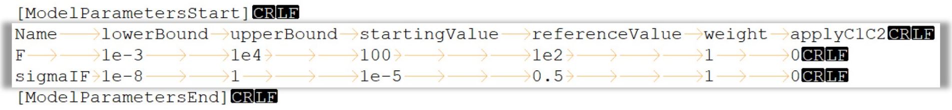
General concept: The forward operator file is organized in blocks, whose start and end lines are indicated by identifiers in square brackets. *AnyPetro* recognizes six blocks: *ModelName*, *ModelParameters*, *AuxiliaryStatements*, *AdditionalInput*, *SampleSpecifics* and *SyntheticDataCalculation*. Thereby, *ModelName*, *ModelParameters*, and *SyntheticDataCalculation* are mandatory (bold outlined), the others are optional:


[ModelNameStart]
[ModelNameEnd]
custom model name
[ModelParametersStart]
[ModelParametersEnd]
table of model parameters and accompanying settings
[AuxiliaryStatementsStart]
[AuxiliaryStatementsEnd]
auxiliary Matlab statements to compute variables for additional input
[AdditionalInputStart]
[AdditionalInputEnd]
list of variables from auxiliary statements to transfer to synthetic data calculation
[SampleSpecificsStart]
[SampleSpecificsEnd]
list of column names from data table to transfer to synthetic data calculation as sample specifics
[SyntheticDataCalculationStart]
[SyntheticDataCalculationEnd]
table of data types and Matlab expressions for synthetic data calculation

Statements and expressions defined in the forward operator file have to be in correct Matlab syntax. All parts of the file outside of the pre-defined blocks are ignored during import. This can be used for storing explanations or comments in the file (see Section 6.2 for an example). Multiple examples for valid forward operator files are provided in Section 6.

ModelName block: Define an arbitrary model name in this block. The model name has to be in one line. The name should be short and special characters should be avoided.

ModelParameters block: In this block the model parameters and associated settings are specified. This is done with a tab-separated table (multiple tabs are possible and treated as one), which bears one row for each model parameter and has to be formatted like in this example:



```
[ModelParametersStart]CRLF
Name    →lowerBound→upperBound→startingValue  →referenceValue→weight→applyC1C2CRLF
F→      →1e-3   →  →1e4→    →  →100→    →   →  →1e2→   →   →  →1→   →0CRLF
sigmaIF→1e-8    →  →1→  →   →  →1e-5    →   →  →0.5→   →   →  →1→   →0CRLF
[ModelParametersEnd]CRLF
```

The column headers and their sequence have to be exactly as given in the example. The table can be prepared e.g. in a *.xlsx file and copied to the forward operator file later.

In the first column (*Name*), provide names for the model parameters. The model parameter names have to be in accordance with the Matlab variable naming rules. You should use unique, short names without spaces, special characters or leading numbers.

The second and third column (*lowerBound*, *upperBound*) contain the parameter range to which the model parameter can be restricted during inversion. Whether this range is actually used, can be specified during inversion set-up later-on.

Provide a reasonable starting value in column four (*startingValue*) and a reference value in column five (*referenceValue*). The starting and reference value, resp., have to lie within the parameter range and must not be identical to the lower or upper bound. Starting and reference values can be the same. In column six (*weight*), provide weights for the model parameters (positive numbers). Using weights that differ from 1 can be helpful for correlated model parameters. Choose all weights equal to 1 as default.

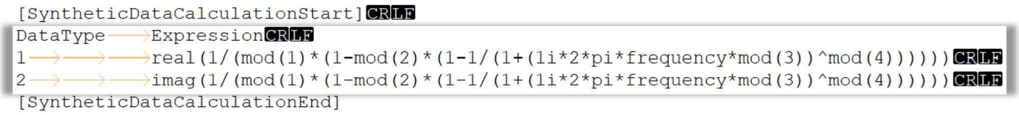
The last column (*applyC1C2*) specifies, whether the $\mathbf{C}_1$ and/or $\mathbf{C}_2$ regularization (if enabled during inversion set-up) is applied to the model parameter. Provide the specification as boolean (0 or 1). If set to 0, the model parameter is excluded from $\mathbf{C}_1$ and/or $\mathbf{C}_2$ regularization. Choose 0 as default (see Section 6.4 for an example with parameter-wise enabled/disabled $\mathbf{C}_1$ and/or $\mathbf{C}_2$ regularization). You can manipulate the model parameter table within the GUI later-on.

AuxiliaryStatements block [OPTIONAL]: Include here all statements that should be executed before synthetic data calculation. You can provide an arbitrary number of lines. Make sure that all statements are in accordance to Matlab syntax and do not use line breaks within one statement. For example, you can include some calculations here that are associated to the model parameters (see Section 6.4 for an example, where a relaxation time τ is calculated). A further possibility is to define anonymous functions in the *Auxiliary Statements* block and call them with specific input during synthetic data calculation (see Section 6.5 for an example). Calculations, which explicitly include model parameters, are not possible.

AdditionalInput block [OPTIONAL]: Provide a comma-separated list of variable names here that are called during synthetic data calculation. The variables have to be previously defined within the AuxiliaryStatements block (see Section 6.4 for an example). Pay attention to the correct upper and lower case letters. Note that anonymous functions defined in the AuxiliaryStatements block are not listed in the AdditionalInput block.

SampleSpecifics block [OPTIONAL]: Provide a comma-separated list of column names appearing in the data table that are called as sample specifics during synthetic data calculation (see Section 6 for multiple examples). Pay attention to the correct upper and lower case letters. Although including this block is optional, in the vast majority of cases at least on sample specific is required.

SyntheticDataCalculation block: Within this block the actual forward calculation for each data type is defined. This is done with a tab-separated table (multiple tabs are possible and treated as one), which bears one row for each data type and has to be formatted like in this example:



```
[SyntheticDataCalculationStart]CRLF
DataType	Expression CRLF
1			real(1/(mod(1)*(1-mod(2)*(1-1/(1+(1i*2*pi*frequency*mod(3))^mod(4))))))CRLF
2			imag(1/(mod(1)*(1-mod(2)*(1-1/(1+(1i*2*pi*frequency*mod(3))^mod(4))))))CRLF
[SyntheticDataCalculationEnd]
```

The column headers and their sequence have to be exactly as given in the example. In the first column (*DataType*), provide a running integer number specifying the data type. This number has to correspond to the entries in the data type column of the data table. For each data type that appears in the data table, a row for synthetic data calculation has to be defined. In the second column (*Expression*), provide an expression in correct Matlab syntax for calculating data of the according type. Note the following requirements and suggestions for the expressions:

- You can use spaces for readability in the expression, but no tabs.
- Each expression has to be given in one line.
- Include the model parameters as `mod(i)`, with `i` being the row number in the model parameter table. Do not use the model parameter names, these are only for user convenience. Multiple model parameters (e.g. as a vector) can be addressed as `mod(i:j)` (see Section 6.4).
- The variables and sample specifics, resp., introduced in the AdditionalInput and SampleSpecifics blocks can be addressed directly in the expression (see the use of `frequency` in the example above).
- Functions defined as anonymous function handles in the AuxiliaryStatements block can be called in the same manner as original Matlab functions with model parameters, additional input variables or sample specifics as arguments (see Section 6.5 for an example with anonymous functions).

- For expressions including vectors, the Matlab syntax differentiates between true array operations and element-wise operations (see Section 6.4 for an example with element-wise operations).

- To check the syntax of an expression, you can use MATLAB or, e.g., Octave.

5.3 Output files

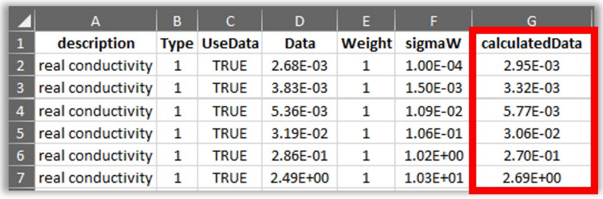
The inversion results can be exported after the calculation is complete. Thereby, seven files with different file extensions are stored, which are all named as follows:

*AnyPetro_inversionResult_YYYY-MM-DD_hhmmss_IDX.**,

where *YYYY-MM-DD* refers to the date (year-month-day format) and *hhmmss* to the time (hours minutes seconds) of the export. *IDX* is an indicator to the file content. Two tables (each of them respectively as *.xlsx and *.txt file with identical content) are stored for data and model, resp., as well as one ASCII log file and two figures. Note that if you apply changes to the data or model parameter table within the GUI, these changes are stored in the output files accordingly.

5.3.1 Output data table

The output data table (*IDX*: *data*) is identical to the input data table with one column *calculatedData* added. This column contains the data for the corresponding row calculated with the final model from inversion. For evaluating the achieved data fit, this calculated data can be compared to the input data column directly. For rows with disabled data use, no calculated data is provided.

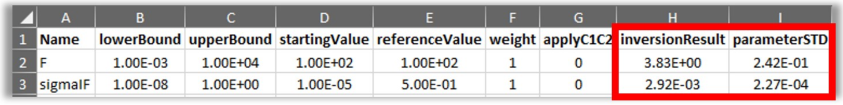


	A	B	C	D	E	F	G
1	description	Type	UseData	Data	Weight	sigmaW	calculatedData
2	real conductivity	1	TRUE	2.68E-03	1	1.00E-04	2.95E-03
3	real conductivity	1	TRUE	3.83E-03	1	1.50E-03	3.32E-03
4	real conductivity	1	TRUE	5.36E-03	1	1.09E-02	5.77E-03
5	real conductivity	1	TRUE	3.19E-02	1	1.06E-01	3.06E-02
6	real conductivity	1	TRUE	2.86E-01	1	1.02E+00	2.70E-01
7	real conductivity	1	TRUE	2.49E+00	1	1.03E+01	2.69E+00

additional column
with calculated data

5.3.2 Output model table

The output model table (*IDX*: *model*) is identical to the model parameter table within the forward operator file with two columns *inversion result* and *parameterSTD* added. This column contains the model parameters as determined by the inversion (*inversionResult*) as well as an estimate of the parameter confidence (*parameterSTD*). This confidence should not be treated as a true standard deviation but as an orientation. A model parameter is well resolved, if the confidence is at least one order of magnitude smaller then the actual parameter.

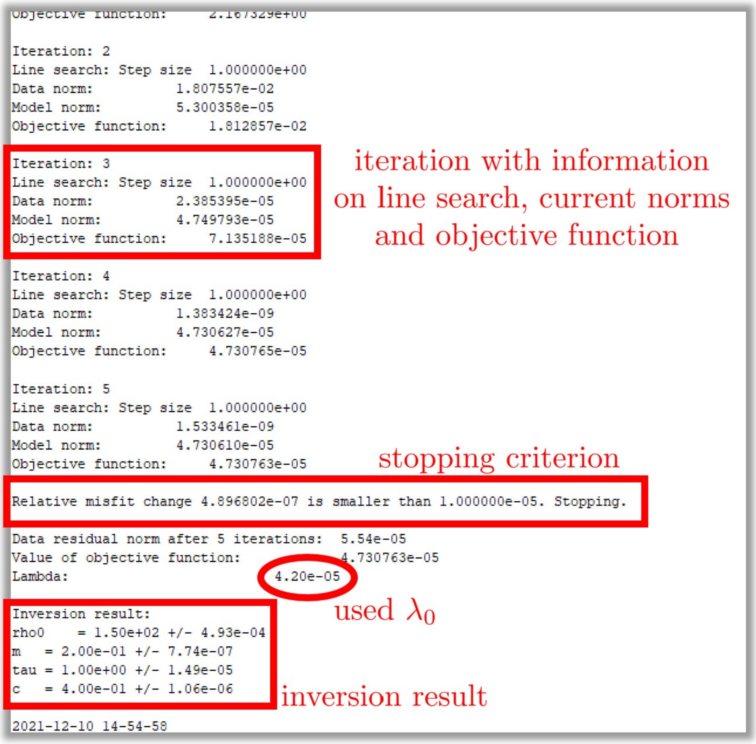


	A	B	C	D	E	F	G	H	I
1	Name	lowerBound	upperBound	startingValue	referenceValue	weight	applyC1C2	inversionResult	parameterSTD
2	F	1.00E-03	1.00E+04	1.00E+02	1.00E+02	1	0	3.83E+00	2.42E-01
3	sigmaIF	1.00E-08	1.00E+00	1.00E-05	5.00E-01	1	0	2.92E-03	2.27E-04

additional columns
with inversion result and
according parameter confidence

5.3.3 Log file

In the ASCII log file, a copy of the command line window is stored. It contains details of the inversion progress, the applied λ_0 (useful if the automatic calculation is used) as well as the inversion result.



```
Objective function:     2.167329e+00

Iteration: 2
Line search: Step size  1.000000e+00
Data norm:            1.807557e-02
Model norm:           5.300358e-05
Objective function:     1.812857e-02

Iteration: 3
Line search: Step size  1.000000e+00
Data norm:            2.385395e-05
Model norm:           4.749793e-05
Objective function:     7.135188e-05

Iteration: 4
Line search: Step size  1.000000e+00
Data norm:            1.383424e-09
Model norm:           4.730627e-05
Objective function:     4.730765e-05

Iteration: 5
Line search: Step size  1.000000e+00
Data norm:            1.533461e-09
Model norm:           4.730610e-05
Objective function:     4.730763e-05

Relative misfit change 4.896802e-07 is smaller than 1.000000e-05. Stopping.

Data residual norm after 5 iterations:  5.54e-05
Value of objective function:            4.730763e-05
Lambda:                           4.20e-05

Inversion result:
rho0    = 1.50e+02 +/- 4.93e-04
m   = 2.00e-01 +/- 7.74e-07
tau = 1.00e+00 +/- 1.49e-05
c   = 4.00e-01 +/- 1.06e-06

2021-12-10 14-54-58
```

iteration with information on line search, current norms and objective function

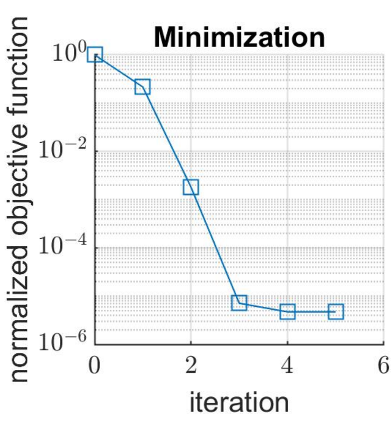
stopping criterion

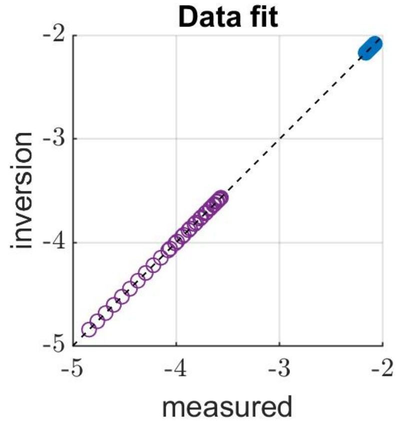
used λ_0

inversion result

5.3.4 Output figures

The two plots from the GUI are stored as raster graphics files. The data fit shows different data types with different colors. If the $\log_{10}$ of the data has been used (see inversion workflow), the data fit plot shows logarithmized data.





5.4 The inversion workflow

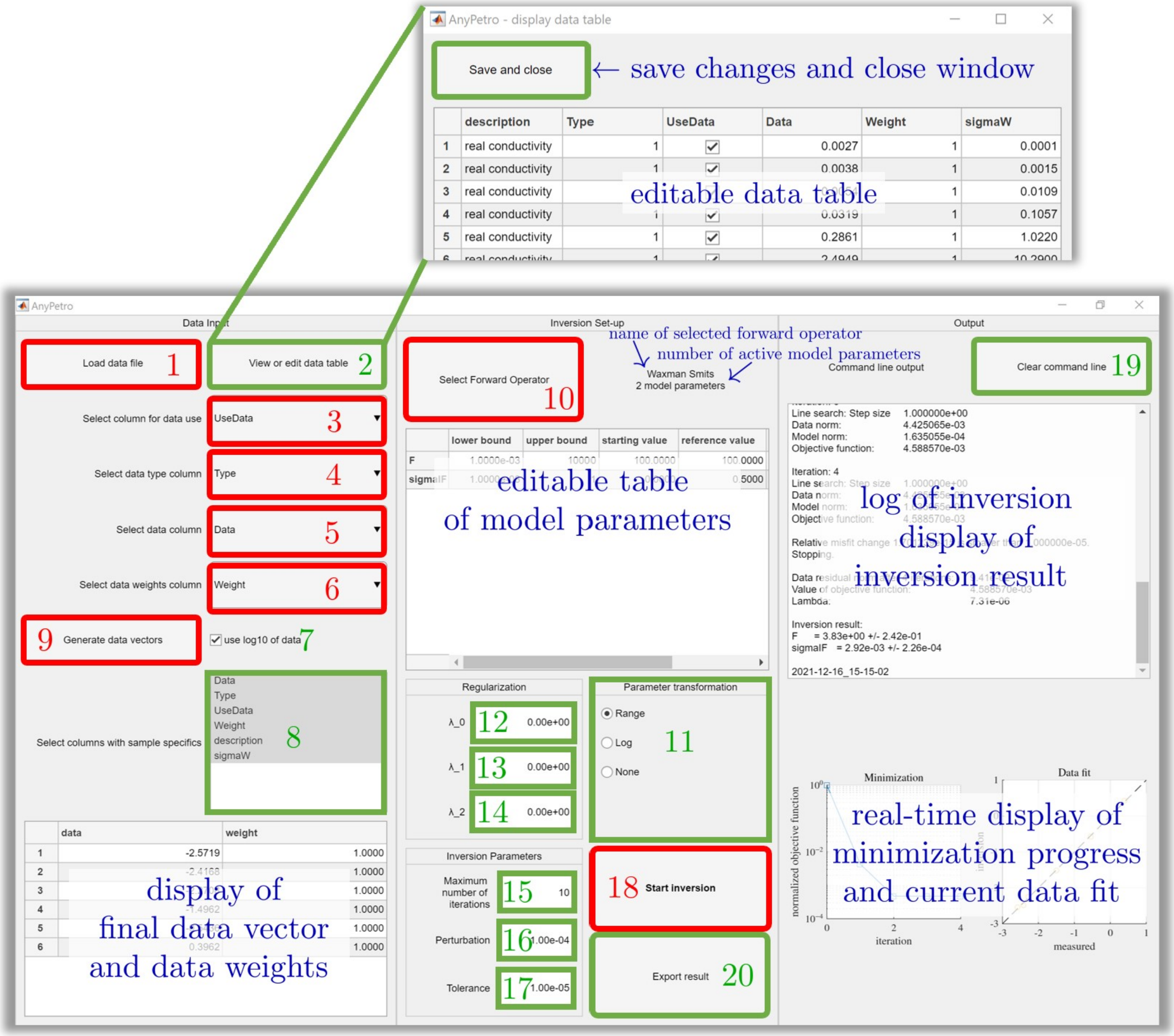
For performing an inversion with *AnyPetro*, the Steps 1 to 20 have to be addressed in a sequence (see Tab. 1 and Fig. 2). In the following, the individual steps and user options are explained in detail.

5.4.1 Load data file

From the open file dialogue, choose the file containing the data table for inversion. Make sure the file is in *.txt or *.xlsx format and is set up in accordance with the data file rules. The files is required to contain a column each for data use, data type, data value and data weight. It can contain arbitrary additional columns for use as sample specifics or for documentation. See Section 7.1 in the appendix for example data files.

5.4.2 View or edit data table [OPTIONAL]

Open an extra window, where the loaded data table can be inspected and manipulated. Here, you can check whether the data table import succeeded and you can change any values or check boxes. Note that you have to re-generate the data vectors if you manipulate the data table at a later stage. The current data table is saved



← save changes and close window

editable data table

	description	Type	UseData	Data	Weight	sigmaW
1	real conductivity	1	☑	0.0027	1	0.0001
2	real conductivity	1	☑	0.0038	1	0.0015
3	real conductivity	[illegible]	[illegible]	0.0054	1	0.0109
4	real conductivity	1	☑	0.0319	1	0.1057
5	real conductivity	1	☑	0.2861	1	1.0220
6	real conductivity	1	☑	2.4949	1	10.2900

name of selected forward operator

number of active model parameters

editable table of model parameters

	lower bound	upper bound	starting value	reference value
F	1.0000e-03	10000	100.0000	100.0000
sigmaIF	1.0000e-05	1	0.5000	0.5000

log of inversion display of inversion result

```
Line search: Step size     1.000000e+00
Data norm:                 4.425065e-03
Model norm:                1.635055e-04
Objective function:        4.588570e-03

Iteration: 4
Line search: Step size     1.000000e+00
Data norm:                 4.425065e-03
Model norm:                1.635055e-04
Objective function:        4.588570e-03

Relative misfit change 1.70[illegible]e-15 is smaller than 1.000000e-05.
Stopping.

Data residual norm after iterations:    9.41[illegible]
Value of objective function:            4.588570e-03
Lambda:                                 7.31e-06

Inversion result:
F      = 3.83e+00 +/- 2.42e-01
sigmaIF   = 2.92e-03 +/- 2.26e-04

2021-12-16_15-15-02
```

real-time display of minimization progress and current data fit

display of final data vector and data weights

	data	weight
1	-2.5719	1.0000
2	-2.4168	1.0000
3	[illegible]	1.0000
4	-1.4962	1.0000
5	[illegible]	1.0000
6	0.3962	1.0000

Figure 2: Workflow for performing an inversion with *AnyPetro*. Steps with a red outline are mandatory, green steps are optional ones. Comments are given in blue.

during the optional export of the inversion results. Save any manipulations of the data table and close the extra window by pushing the *Save and close* button.

5.4.3 Select column for data use

From the drop-down menu, choose the column of the loaded data table, which contains the information on data use (boolean). A column is pre-selected. For this pre-selection, *AnyPetro* searches the column headers for the word *use* and selects the first header found.

5.4.4 Select data type column

From the drop-down menu, choose the column of the loaded data table, which contains the information on data type (integer). A column is pre-selected. For this pre-selection, *AnyPetro* searches the column headers for the word *type* and selects the first header found.

5.4.5 Select data column

From the drop-down menu, choose the column of the loaded data table, which contains the data values. A column is pre-selected. For this pre-selection, *AnyPetro* searches the column headers for the word *data* and selects the first header found.

Table 1: Steps for performing an inversion with *AnyPetro*. The steps are numbered in accordance with Fig. 2. All steps are linked to the corresponding explanation paragraphs. Mandatory steps are indicated.

step	mandatory	description	default
1	✗	load data file	–
2		view or edit data table	–
3	✗	select column for data use	automatically preselected column
4	✗	select data type column	automatically preselected column
5	✗	select data column	automatically preselected column
6	✗	select data weight column	automatically preselected column
7		use $\log_{10}$ of data	yes
8		select columns with sample specifics	all columns of data table
9	✗	generate data vectors	–
10	✗	select forward operator	–
11		parameter transformation	range
12		λ_0	computed automatically
13		λ_1	0 (i.e. $\mathbf{C_1}$ regularization disabled)
14		λ_2	0 (i.e. $\mathbf{C_2}$ regularization disabled)
15		maximum number of iterations	10
16		perturbation	$1E-04$
17		tolerance	$1E-05$
18	✗	start inversion	–
19		clear command line	–
20		export result	–

5.4.6 Select data weight column

From the drop-down menu, choose the column of the loaded data table, which contains the information on data weights. A column is pre-selected. For this pre-selection, *AnyPetro* searches the column headers for the word *weight* and selects the first header found.

5.4.7 Use $\log_{10}$ of data [OPTIONAL]

Choose whether the data values from the loaded data table should be transformed by $\log_{10}$. This is usually advantageous for data covering several orders of magnitude, e.g. due to a high data dynamics or due to multiple data types. As default, the $\log_{10}$ transformation is used. If it is not wanted, un-check the check box. Please note that using the $\log_{10}$ is only possible, if all data are positive values.

5.4.8 Select columns with sample specifics [OPTIONAL]

Select from the list box those columns, which should be transferred to the forward operator as sample specifics (accompanying parameters provided for the data, which are needed for calculating synthetic data, e.g. frequency for SIP data). As default, all columns are selected. Choosing more columns than strictly necessary is not a problem. Make sure that at least those columns are selected which are specified as sample specifics in the forward operator file.

5.4.9 Generate data vectors

Push this button when the data input and configuration is finished. The vector of observed data $\mathbf{d}_{\text{obs}}$ and the according vector of data weights is generated and displayed (read only) in the window below. Length and entries of the vectors can be checked. If any adjustment is applied (changes in steps described in 5.4.1 – 5.4.8), the *Generate data vectors* button has to be pushed again in order to update the data vectors. After the initial generation of the data vectors, the *Inversion Set-up* panel is enabled and you can proceed to load the forward operator. If the data vectors are re-generated, the forward operator is cleared and has to be loaded again.

5.4.10 Select forward operator

From the open file dialogue, choose the file containing the forward operator. Make sure the file is in *.txt format and is set up in accordance with the forward operator file rules. The file is required to contain information on the model parameters, the parameter ranges, and the starting and reference model. It contains the calculation rules for synthetic data (see Section 6 for example files of forward operators). The name of the forward operator and the number of detected model parameters are displayed to the right of the button. Once loaded, the model parameters and the according ranges, starting and reference values are displayed in the table below. The parameter table can be manipulated, e.g. to test different starting models. A table of the current model parameter is saved during the optional export of the inversion results.

5.4.11 Parameter transformation [OPTIONAL]

Choose a model parameter transformation by selecting on of the three suggestions: *Range*, *Log* or *None*. *Range* is pre-selected and used as default. Choose *Log* if you want to enforce model parameters to be positive values, but do not want to limit their range. Choose *None* for disabling the transformation of model parameters.

5.4.12 Regularization [OPTIONAL]

5.4.12.1 λ_0: Set manually the λ_0 regularization parameter by entering a positive value here. If λ_0 is set to 0, as in the default case, a λ_0 value is calculated automatically from the initial data and model norms. The calculated value is given in the command line window at the end of the inversion, just above the inversion result. Since λ_0 balances the data and the model norm, higher λ_0 values strengthen the influence of the starting and reference model on the minimization process. Therefore, high λ_0 values tend to stabilize the minimization process. At the same time, more iterations are needed for convergence.

5.4.12.2 λ_1: Set manually the λ_1 factor by entering a positive value here. If $\lambda_1 > 0$, the $\mathbf{C}_1$ regularization is enabled in addition to the standard $\mathbf{C}_0$ regularization (unity matrix). With $\mathbf{C}_1$, a coupling of model parameters, which are neighbours in the model parameter table, according to the first derivative is introduced. Thereby, λ_1 acts as a factor, which balances $\mathbf{C}_1$ against $\mathbf{C}_0$. For example, $\lambda_1 = 10$ means that $\mathbf{C}_1$ has a weight of 10 over $\mathbf{C}_0$. Choosing high values for λ_1 will lead to a smoother model. Generally, introducing λ_1 is only sensible for correlated, interdependent model parameters. λ_1 and λ_2 can be used at the same time. The weight of $\mathbf{C}_1$ over $\mathbf{C}_2$ is $\frac{\lambda_1}{\lambda_2}$.

5.4.12.3 λ_2: Set manually the λ_2 factor by entering a positive value here. If $\lambda_2 > 0$, the $\mathbf{C}_2$ regularization is enabled in addition to the standard $\mathbf{C}_0$ regularization (unity matrix). With $\mathbf{C}_2$, a coupling of model parameters, which are neighbours in the model parameter table, according to the second derivative is introduced. Thereby, λ_2 acts as a factor, which balances $\mathbf{C}_2$ against $\mathbf{C}_0$. For example, $\lambda_2 = 10$ means that $\mathbf{C}_2$ has a weight of 10 over $\mathbf{C}_0$. Choosing high values for λ_2 will lead to a smoother model. Generally, introducing λ_2 is only sensible for correlated, interdependent model parameters. λ_1 and λ_2 can be used at the same time. The weight of $\mathbf{C}_2$ over $\mathbf{C}_1$ is $\frac{\lambda_2}{\lambda_1}$.

5.4.13 Inversion parameters [OPTIONAL]

Maximum number of iterations: The inversion stops when the maximum number of iterations specified by the user is reached, unless the tolerance criterion is fulfilled at an earlier stage. A default value of 10 is used as maximum number of iterations. Change this number to higher positive integer value if it is clear from the minimization plot that the inverse problem has not yet converged.

Perturbation: The specified value is used as fractional perturbation of each model parameter during the calculation of the Jacobian matrix. Changing the value is usually not necessary. In rare cases, adapting the perturbation, e.g. by increasing it slightly, can help stabilizing the inversion.

Tolerance: Stop criterion used for detecting convergence and stopping the inversion before the maximum number of iterations would be reached. The inversion stops when the reduction of the objective function between two iterations falls below the specified value. Increase the tolerance if you fear to over-fit the data. Decrease the tolerance if you want the inversion to continue the minimization.

5.4.14 Start inversion

When data input, forward operator, and inversion parameters are set, push the *Start inversion* button to run the inversion. It is possible to re-run the inversion multiple times, e.g. after changing an inversion parameter.

5.4.15 Clear command line [OPTIONAL]

Push this button to delete the log printed in the command line window. Any imported data, forward operators, parameters or inversion results are not affected. Clearing the command line window can be advantageous when re-running the inversion multiple times.

5.4.16 Export result [OPTIONAL]

A folder browser dialogue opens. Choose a target folder where to export the inversion output files. The content of the command line window is stored in the log-file. The file names are generated automatically and contain the time stamp of the export in order to avoid overwriting of previously exported results.

6 Examples

In the following, several examples for performing an inversion with *AnyPetro* are given. All input and output files for each example are given in the *examples* subfolder of the *AnyPetro* directory and in the appendix of this user guide. If not stated otherwise, default values and settings are used.

6.1 Waxman-Smits model for electrical interface conductivity

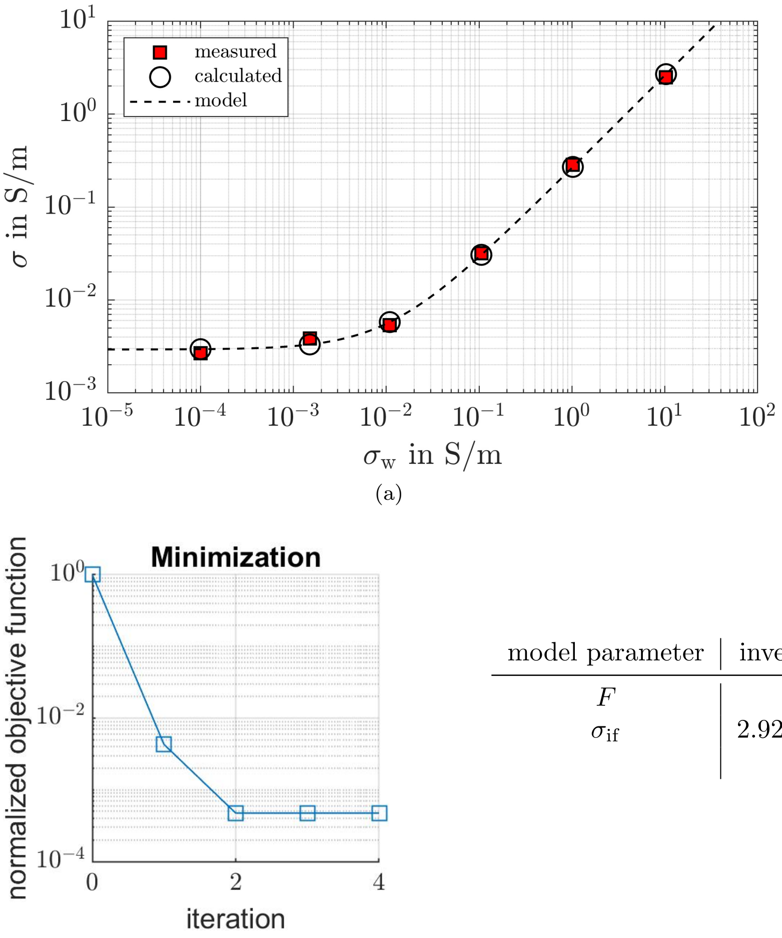
The Waxman-Smits equation is used to determine the electrical interface conductivity based on multi-salinity experiments. It can be formulated in the most simple case as follows (Waxman & Smits, 1968; Schön, 2015):

$$\sigma = \frac{1}{F}\sigma_{\mathrm{w}} + \sigma_{\mathrm{if}}, \tag{7}$$

where σ is electrical rock conductivity, σ_{w} is electrical pore-water conductivity, F is the formation factor and σ_{if} is electrical interface conductivity. For the inverse problem this means:

- 1 data type: measured σ
- 1 sample specific: measured σ_{w} for each σ
- 2 unknown model parameters: F and σ_{if}.

An exemplary multi-salinity data set of a clean quartz sand at multiple σ_{w} is given in the data file TXT_data_table_multiSalinity.txt / XLSX_data_table_multiSalinity.xlsx. The according forward operator file is WaxmanSmits.txt. Measured data, calculated data fit, minimization progress, and inverted model parameters are shown as results in Fig. 3.



(a)

model parameter	inversion result
F	3.83
σ_{if}	$2.92E-03\,\mathrm{S/m}$

(b)

(c)

Figure 3: Inversion result for Waxman-Smits example.

6.2 Wyllie's time-average equation for p-wave velocity

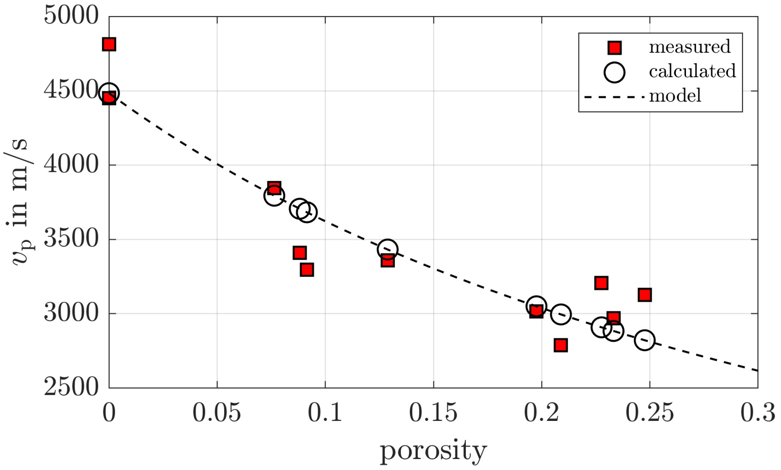
Wyllie's time-average equation describes the porosity dependence of p-wave velocity for water-saturated porous rocks. It reads in terms of p-wave velocity v_p (Wyllie et al., 1956; Schön, 2015):

$$\frac{1}{v_\mathrm{p}} = \frac{1-\Phi}{v_\mathrm{p,matrix}} + \frac{\Phi}{v_\mathrm{fluid}} \tag{8}$$

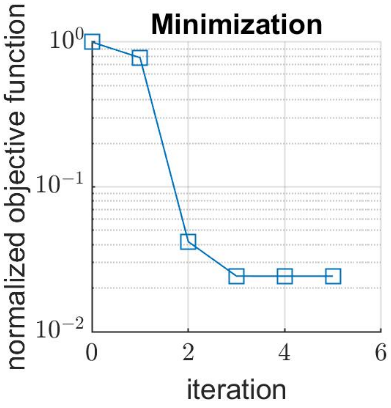
where Φ denotes porosity, $v_\mathrm{p,matrix}$ is the p-wave velocity of the solid rock matrix and v_fluid is the p-wave velocity of the pore filling. For the inverse problem this means:

- 1 data type: measured v_p
- 1 sample specific: measured Φ for each v_p
- 2 unknown model parameters: $v_\mathrm{p,matrix}$ and v_fluid.

An demonstration data set is given in the data file TXT_data_table_pWaveVelocity.txt / XLSX_data_table_pWaveVelocity.xlsx. The according forward operator file is Wyllie.txt. Deviating from default, the use of the $\log_{10}$ of the measured data should be disabled. Measured data, calculated data fit, minimization progress and inverted model parameters are shown in Fig. 4.



(a)



(b)

model parameter	inversion result
$v_\mathrm{p,matrix}$	4 484 m/s
v_fluid	1 327 m/s

(c)

Figure 4: Inversion result for Wyllie's time-average equation example.

6.3 Cole-Cole model for frequency-dependent, complex electrical conductivity

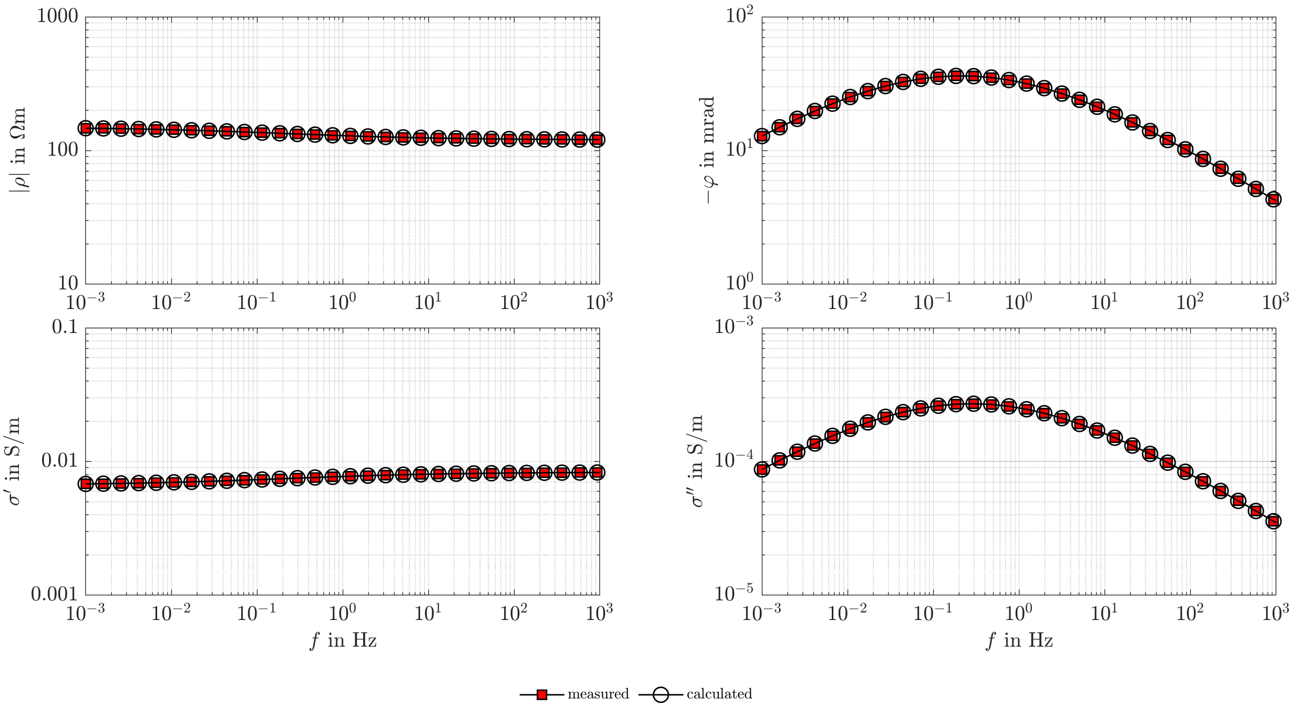
The Cole-Cole model for complex conductivity σ^* reads (Cole & Cole, 1942; Pelton et al., 1978):

$$\sigma^*(\omega) = \sigma'(\omega) + i\sigma''(\omega) = \frac{1}{\rho_0}\left[1 - m\left(1 - \frac{1}{1+(i\omega\tau)^c}\right)\right]^{-1} \quad , \tag{9}$$

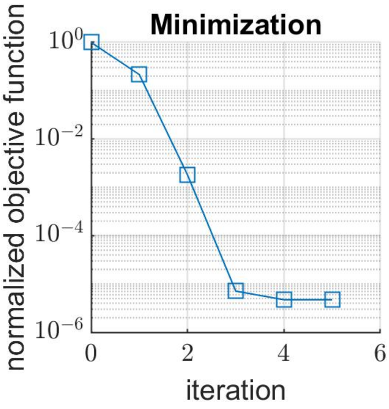
where ρ_0 is the DC resistivity, m is the chargeability, τ is the relaxation time, c is called frequency exponent and $\omega = 2\pi f$ with frequency f. For inverting one complex conductivity spectrum this means the following:

- 2 data types: measured real σ' and imaginary σ'' part of electrical conductivity
- 1 sample specific: frequency f
- 4 unknown model parameters: ρ_0, m, τ and c.

A synthetic complex electrical conductivity spectrum serves as data in this example and is given in the data file TXT_data_table_complexConductivity.txt / XLSX_data_table_complexConductivity.xlsx. The according forward operator file is ColeCole.txt. Measured data, calculated data fit, minimization progress and inverted model parameters are shown in Fig. 5.



(a)



(b)

model parameter	inversion result
ρ_0	150 Ωm
m	0.20
τ	1.00 s
c	0.40

(c)

Figure 5: Inversion result for Cole-Cole example.

6.4 Debye decomposition

Alternatively to the Cole-Cole model, a complex electrical conductivity spectrum can be represented by a series of Debye relaxations (Debye, 1923; Nordsiek & Weller, 2008):

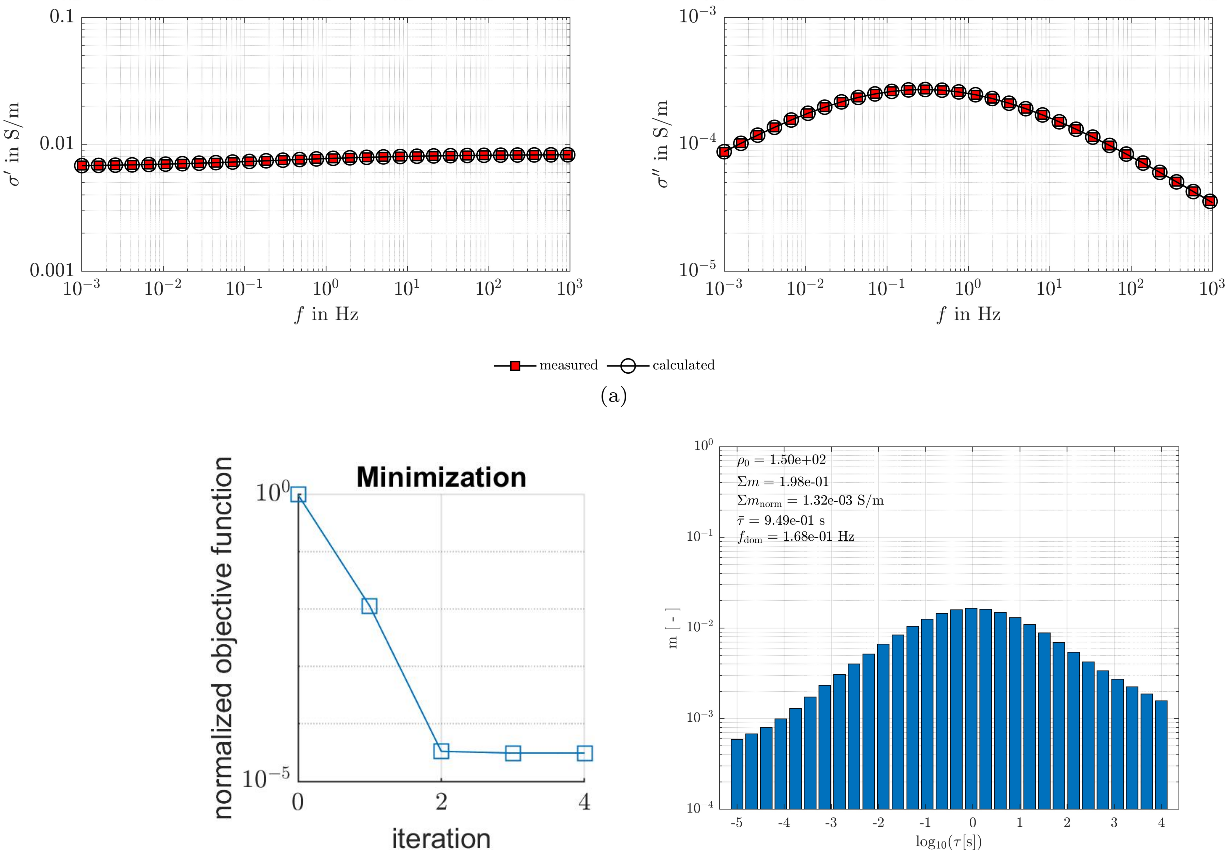
$$\sigma^*(\omega) = \frac{1}{\rho^*} = \frac{1}{\rho_0}\left(1 - \sum_\ell m_\ell \left(1 - \frac{1}{1 + i\omega\tau_\ell}\right)\right)^{-1} \tag{10}$$

with a direct current resistivity ρ_0, a set of relaxation times τ_ℓ and chargeabilities m_ℓ and $\omega = 2\pi f$ with frequency f. For inverting one complex electrical conductivity spectrum and using a decomposition with 30 relaxation times, this means:

- 2 data types: measured real σ' and imaginary σ'' part of electrical conductivity
- 1 sample specific: frequency f
- 1 additional input: relaxation time τ (is defined in the *AuxiliaryStatements* block)
- 31 unknown model parameters: ρ_0, 30 chargeabilities m_ℓ, one for each τ_ℓ.

A synthetic complex electrical conductivity spectrum (same as for the Cole-Cole example) serves as data in this example and is given in the data file TXT_data_table_complexConductivity.txt / XLSX_data_table_complexConductivity.xlsx. The according forward operator file is DebyeDecomposition.txt. For the Debye decomposition, the $\mathbf{C}_1$ and $\mathbf{C}_2$ regularization are enabled. In order to do so, choose $\lambda_1 = 10$ and $\lambda_2 = 100$. Measured data, calculated data fit, minimization progress and inverted model parameters are shown in Fig. 6.

If you want to change the range of possible relaxation times (1E-05 to 1E04 in the example), change the variable `limTau` in the *AuxiliaryStatements* block of the forward operator file according to your needs: `limTau = log10([<lowerLimit> <upperlimit>]);` . If you want to use a different number of relaxation times (30 in the example), adapt the *ModelParameters* block in the forward operator file to contain the correct number of rows for the chargeabilities (m01 to m30 in the example). Then, change the variable `nTau` in the *AuxiliaryStatements* block according to your needs: `nTau = <number of relaxation times>;` .



(a)

(b)

(c)

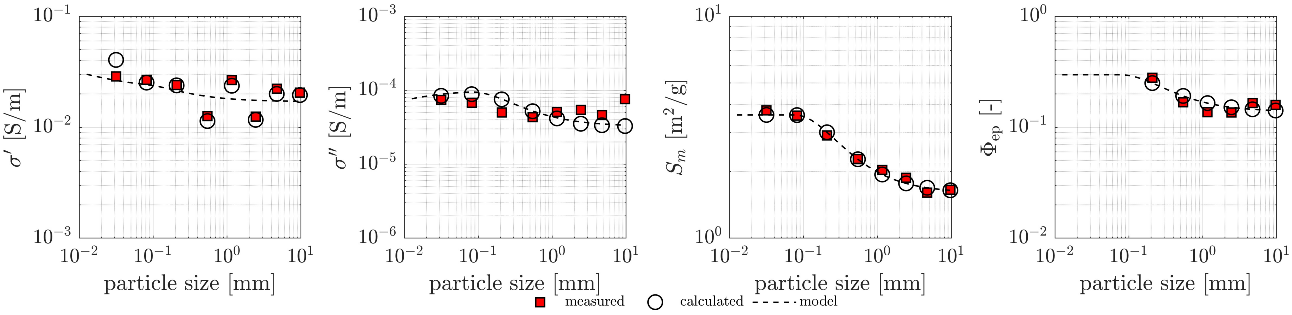
Figure 6: Inversion result for Debye decomposition example.

6.5 Recovering formation properties from measurements on crushed rock

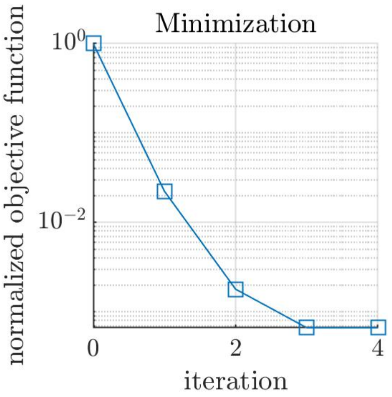
This example demonstrates the use of *AnyPetro* for more complex petrophysical models by means of anonymous functions. The model describes the specific surface (S_m [m^2/g]), the porosity (Φ [-]) and the complex electrical conductivity (σ^* [S/m]) and their interrelations for packings of crushed rock particles. The goal of the inversion is to reconstruct the original formation properties from the packing measurements. All details can be found in Börner et al. (2022) as the corresponding publication for the first application of the software *AnyPetro*. The inverse problem can be summarized as follows:

- 4 data types: measured specific surface S_m, porosity Φ, real σ' and imaginary σ'' part of electrical conductivity for multiple packings
- 5 sample specific: particle size, inter-particle porosity, matrix density, pore-water conductivity, ratio between real and imaginary part of electical interface conductivity
- 7 unknown model parameters: properties of the undisturbed formation (specific surface, porosity, cementation exponent, surface conductance) and particle properties (thickness of crushing zone, porosity change in zone, surface change in zone).

An exemplary data set of measurements on crushed rock of varying particle size is given in the data file TXT_data_table_complexConductivity.txt / XLSX_data_table_complexConductivity.xlsx. The according forward operator file is CrushedRock.txt. Measured data, calculated data fit, minimization progress and inverted model parameters are shown in Fig. 7.



(a)



(b)

model parameter	inversion result
$S_{\mathrm{m,c}}$	1.60 m^2/g
Φ_c	0.14
m	1.94
$-\log_{10}(\Sigma[\mathrm{S}])$	9.00
dr	34.5 μm
f_S	2.24
f_Φ	2.15

(c)

Figure 7: Inversion result for crushed rock example.

References

Aster, R. C., Borchers, B., & Thurber, C. H., 2005. *Parameter Estimation and Inverse Problems*, Elsevier.

Börner, J. H., 2016. *Electrical phenomena during CO_2–rock interaction under reservoir conditions – experimental investigations and their implications for electromagnetic monitoring applications*, Ph.D. thesis, Technical University Bergakademie Freiberg.

Börner, J. H., Herdegen, V., Repke, J.-U., & Spitzer, K., 2022. Retrieving electrical and structural carbonate formation properties from measurements on crushed rock using a multi-data inversion approach, *Geophysical Journal International*, in press.

Cole, K. S. & Cole, R. H., 1942. Dispersion and absorption in dielectrics II. direct current characteristics, *The Journal of Chemical Physics*, **10**(2), 98–105.

Debye, P., 1923. On ions and their activity, *Chemisch Weekblad*, **20**, 562–568.

Malecki, S., Börner, R.-U., & Spitzer, K., 2020. A new method for absolute underground positioning based on transient electromagnetics, *Geophysical Journal International*, **221**(1), 87–96.

Marsden, P. & Wright, J., 2010. *Handbook of survey research*, Bingley, UK: Emerald Group Publishing Limited.

Menke, W., 2018. *Geophysical data analysis: Discrete inverse theory*, Academic press.

Nocedal, J. & Wright, S. J., 2006. *Numerical Optimization*, Springer, 2nd edn.

Nordsiek, S. & Weller, A., 2008. A new approach to fitting induced-polarization spectra, *Geophysics*, **73**(6), F235–F245.

Pelton, W., Ward, S., Hallof, P., Sill, W., & Nelson, P., 1978. Mineral discrimination and removal of inductive coupling with multifrequency IP, *Geophysics*, **43**(3), 588–609.

Rücker, C., 2011. *Advanced electrical resistivity modelling and inversion using unstructured discretization*, Ph.D. thesis, Universität Leipzig.

Scheunert, M., Ullmann, A., Afanasjew, M., Börner, R.-U., Siemon, B., & Spitzer, K., 2016. A cut-&-paste strategy for the 3-D inversion of helicopter-borne electromagnetic data — I. 3-d inversion using the explicit jacobian and a tensor-based formulation, *Journal of Applied Geophysics*, **129**, 209–221.

Schön, J., 2015. *Physical Properties of Rocks - Fundamentals and Principles of Petrophysics*, vol. 18 of **Handbook of Geophysical Exploration**, Elsevier, 2nd edn.

Tikhonov, A. N., Goncharsky, A., Stepanov, V., & Yagola, A. G., 1995. *Numerical methods for the solution of ill-posed problems*, vol. 328 of **Mathematics and Its Applications**, Springer Science & Business Media.

Wang, F., Morten, J. P., & Spitzer, K., 2018. Anisotropic three-dimensional inversion of CSEM data using finite-element techniques on unstructured grids, *Geophysical Journal International*, **213**(2), 1056–1072.

Waxman, M. H. & Smits, L. J. M., 1968. Electrical conductivities in oil-bearing shaly sands, *Society of Petroleum Engineers Journal*, **8**, 107–122.

Wyllie, M. R. J., Gregory, A. R., & Gardner, L. W., 1956. Elastic wave velocities in heterogeneous and porous media, *Geophysics*, **21**(1), 41–70.

7 Appendix

7.1 Data files

Listing 1: and use with the Waxman-Smits forward operator (List. 5). Water conductivity sigmaW is in S/m.]Example data file *TXT_data_table_multiSalinity.txt* for electrical conductivity data [S/m] and use with the Waxman-Smits forward operator (List. 5). Water conductivity sigmaW is in S/m.

```
description Type    UseData Data     Weight   sigmaW
realConductivity    1    1    2.68E-03    1    1.00E-04
realConductivity    1    1    3.83E-03    1    1.50E-03
realConductivity    1    1    5.36E-03    1    1.09E-02
realConductivity    1    1    3.19E-02    1    1.06E-01
realConductivity    1    1    2.86E-01    1    1.02E+00
realConductivity    1    1    2.49E+00    1    1.03E+01
```

Listing 2: and use with the Wyllie's equation (List. 6). Density is given in g/cm^3 (not used by List. 6), porosity is dimensionless.]Example data file *TXT_data_table_pWaveVelocity.txt* for p-wave velocity data [m/s] and use with the Wyllie's equation (List. 6). Density is given in g/cm^3 (not used by List. 6), porosity is dimensionless.

```
description Type    UseData Data     Weight   density porosity
Rotliegend   1    1    3846     1    2.45     0.08
Buntsandstein1   1    1    3358     1    2.31     0.13
Buntsandstein2   1    1    2970     1    2.03     0.23
Sandstone2   1    1    3207     1    2.05     0.23
SandstoneOttendorf   1    1    3296     1    2.41     0.09
EolianSandstone  1    1    3410     1    2.42     0.09
GoethitSandstone    1    1    3127     1    1.99     0.25
Sandstone3   1    1    3015     1    2.13     0.20
Sandstone4   1    1    2788     1    2.10     0.21
Gneis    1    1    4814     1    2.65     0.00
Granite  1    1    4450     1    2.68     0.00
Marble   1    0    7023     1    2.64     0.02
SeewerKalk   1    0    5765     1    2.62     0.03
Jurakalk     1    0    6105     1    2.58     0.05
Wellenkalk   1    0    4470     1    2.57     0.05
darkMarble   1    0    4421     1    2.45     0.10
MadigerKalk  1    0    4134     1    2.16     0.20
Schaumkalk   1    0    2444     1    1.92     0.29
```

Listing 3: Example data file *TXT_data_table_complexConductivity.txt* for multi-frequency complex electrical conductivity data (all in S/m, frequency in Hz).

```
description Type    UseData Data     Weight   frequency
realConductivity    1    1    6.80E-03    1    1.00E-03
realConductivity    1    1    6.83E-03    1    1.61E-03
realConductivity    1    1    6.86E-03    1    2.58E-03
realConductivity    1    1    6.90E-03    1    4.15E-03
realConductivity    1    1    6.94E-03    1    6.66E-03
realConductivity    1    1    6.99E-03    1    1.07E-02
realConductivity    1    1    7.05E-03    1    1.72E-02
realConductivity    1    1    7.11E-03    1    2.76E-02
realConductivity    1    1    7.18E-03    1    4.44E-02
realConductivity    1    1    7.26E-03    1    7.13E-02
realConductivity    1    1    7.34E-03    1    1.15E-01
realConductivity    1    1    7.42E-03    1    1.84E-01
realConductivity    1    1    7.51E-03    1    2.96E-01
realConductivity    1    1    7.60E-03    1    4.75E-01
realConductivity    1    1    7.68E-03    1    7.63E-01
realConductivity    1    1    7.76E-03    1    1.23E+00
realConductivity    1    1    7.84E-03    1    1.97E+00
realConductivity    1    1    7.91E-03    1    3.16E+00
realConductivity    1    1    7.97E-03    1    5.08E+00
realConductivity    1    1    8.02E-03    1    8.16E+00
realConductivity    1    1    8.07E-03    1    1.31E+01
realConductivity    1    1    8.11E-03    1    2.11E+01
realConductivity     1    1     8.15E-03      1     3.38E+01
realConductivity     1    1     8.18E-03      1     5.44E+01
realConductivity     1    1     8.20E-03      1     8.73E+01
realConductivity     1    1     8.23E-03      1     1.40E+02
realConductivity     1    1     8.24E-03      1     2.25E+02
realConductivity     1    1     8.26E-03      1     3.62E+02
realConductivity     1    1     8.27E-03      1     5.82E+02
realConductivity     1    1     8.28E-03      1     9.35E+02
realConductivity     1    1     8.29E-03      1     1.50E+03
realConductivity     1    1     8.30E-03      1     2.41E+03
realConductivity     1    1     8.30E-03      1     3.87E+03
realConductivity     1    1     8.31E-03      1     6.22E+03
realConductivity     1    1     8.31E-03      1     1.00E+04
imaginaryConductivity    2    1     8.73E-05      1     1.00E-03
imaginaryConductivity    2    1     1.02E-04      1     1.61E-03
imaginaryConductivity    2    1     1.18E-04      1     2.58E-03
imaginaryConductivity    2    1     1.36E-04      1     4.15E-03
imaginaryConductivity    2    1     1.56E-04      1     6.66E-03
imaginaryConductivity    2    1     1.76E-04      1     1.07E-02
imaginaryConductivity    2    1     1.96E-04      1     1.72E-02
imaginaryConductivity    2    1     2.16E-04      1     2.76E-02
imaginaryConductivity    2    1     2.34E-04      1     4.44E-02
imaginaryConductivity    2    1     2.50E-04      1     7.13E-02
imaginaryConductivity    2    1     2.62E-04      1     1.15E-01
imaginaryConductivity    2    1     2.69E-04      1     1.84E-01
imaginaryConductivity    2    1     2.71E-04      1     2.96E-01
imaginaryConductivity    2    1     2.67E-04      1     4.75E-01
imaginaryConductivity    2    1     2.59E-04      1     7.63E-01
imaginaryConductivity    2    1     2.46E-04      1     1.23E+00
imaginaryConductivity    2    1     2.30E-04      1     1.97E+00
imaginaryConductivity    2    1     2.11E-04      1     3.16E+00
imaginaryConductivity    2    1     1.91E-04      1     5.08E+00
imaginaryConductivity    2    1     1.71E-04      1     8.16E+00
imaginaryConductivity    2    1     1.51E-04      1     1.31E+01
imaginaryConductivity    2    1     1.32E-04      1     2.11E+01
imaginaryConductivity    2    1     1.14E-04      1     3.38E+01
imaginaryConductivity    2    1     9.81E-05      1     5.44E+01
imaginaryConductivity    2    1     8.38E-05      1     8.73E+01
imaginaryConductivity    2    1     7.12E-05      1     1.40E+02
imaginaryConductivity    2    1     6.03E-05      1     2.25E+02
imaginaryConductivity    2    1     5.08E-05      1     3.62E+02
imaginaryConductivity    2    1     4.27E-05      1     5.82E+02
imaginaryConductivity    2    1     3.58E-05      1     9.35E+02
imaginaryConductivity    2    1     2.99E-05      1     1.50E+03
imaginaryConductivity    2    1     2.50E-05      1     2.41E+03
imaginaryConductivity    2    1     2.08E-05      1     3.87E+03
imaginaryConductivity    2    1     1.73E-05      1     6.22E+03
imaginaryConductivity    2    1     1.44E-05      1     1.00E+04
```

Listing 4: Example data file *TXT_data_table_crushedRock.txt* for specific surface, porosity and complex electrical conductivity data of crushed rock.

```
description        Type      Data      UseData  Weight   particleSize      density  total_porosity
      PhiinterNeu         sigmaW    l
realConductivity         1          2.05E-02          1          1.21E-01          9.66E+00
            2.71E+00          5.50E-01          4.64E-01          5.50E-02          0.05
realConductivity         1          2.23E-02          1          1.24E-01          4.70E+00
            2.71E+00          5.60E-01          4.73E-01          5.48E-02          0.05
realConductivity         1          1.24E-02          1          1.08E-01          2.44E+00
            2.71E+00          5.46E-01          4.74E-01          3.08E-02          0.05
realConductivity         1          2.66E-02          1          1.30E-01          1.16E+00
            2.71E+00          5.46E-01          4.64E-01          6.51E-02          0.05
realConductivity         1          1.26E-02          1          1.08E-01          5.43E-01
            2.71E+00          5.34E-01          4.40E-01          3.04E-02          0.05
realConductivity         1          2.39E-02          1          1.27E-01          2.08E-01
            2.71E+00          4.71E-01          4.71E-01          5.65E-02          0.05
```

```
realConductivity          1         2.67E-02         1         1.30E-01         8.15E-02
            2.71E+00          5.30E-01          5.30E-01          4.95E-02          0.05
realConductivity          1         2.87E-02         1         1.33E-01         3.15E-02
            2.71E+00          5.72E-01          5.72E-01          7.54E-02          0.05
imaginaryConductivity     2         7.56E-05         1         4.98E-02         9.66E+00
            2.71E+00          5.50E-01          4.64E-01          5.50E-02          0.05
imaginaryConductivity     2         4.61E-05         1         4.73E-02         4.70E+00
            2.71E+00          5.60E-01          4.73E-01          5.48E-02          0.05
imaginaryConductivity     2         5.44E-05         1         4.81E-02         2.44E+00
            2.71E+00          5.46E-01          4.74E-01          3.08E-02          0.05
imaginaryConductivity     2         5.08E-05         1         4.78E-02         1.16E+00
            2.71E+00          5.46E-01          4.64E-01          6.51E-02          0.05
imaginaryConductivity     2         4.34E-05         1         4.70E-02         5.43E-01
            2.71E+00          5.34E-01          4.40E-01          3.04E-02          0.05
imaginaryConductivity     2         4.99E-05         1         4.77E-02         2.08E-01
            2.71E+00          4.71E-01          4.71E-01          5.65E-02          0.05
imaginaryConductivity     2         6.74E-05         1         4.92E-02         8.15E-02
            2.71E+00          5.30E-01          5.30E-01          4.95E-02          0.05
imaginaryConductivity     2         7.37E-05         1         4.96E-02         3.15E-02
            2.71E+00          5.72E-01          5.72E-01          7.54E-02          0.05
specificInternalSurface   3         1.65E+00         1         9.39E-01         9.66E+00
            2.71E+00          5.50E-01          4.64E-01          5.50E-02          0.05
specificInternalSurface   3         1.60E+00         1         1.00E+00         4.70E+00
            2.71E+00          5.60E-01          4.73E-01          5.48E-02          0.05
specificInternalSurface   3         1.87E+00         1         7.52E-01         2.44E+00
            2.71E+00          5.46E-01          4.74E-01          3.08E-02          0.05
specificInternalSurface   3         2.03E+00         1         6.67E-01         1.16E+00
            2.71E+00          5.46E-01          4.64E-01          6.51E-02          0.05
specificInternalSurface   3         2.28E+00         1         5.74E-01         5.43E-01
            2.71E+00          5.34E-01          4.40E-01          3.04E-02          0.05
specificInternalSurface   3         2.90E+00         1         4.43E-01         2.08E-01
            2.71E+00          4.71E-01          4.71E-01          5.65E-02          0.05
specificInternalSurface   3         3.57E+00         1         3.71E-01         8.15E-02
            2.71E+00          5.30E-01          5.30E-01          4.95E-02          0.05
specificInternalSurface   3         3.77E+00         1         3.56E-01         3.15E-02
            2.71E+00          5.72E-01          5.72E-01          7.54E-02          0.05
intra-particlePorosity    4         1.59E-01         1         2.57E-01         9.66E+00
            2.71E+00          5.50E-01          4.64E-01          5.50E-02          0.05
intra-particlePorosity    4         1.65E-01         1         2.62E-01         4.70E+00
            2.71E+00          5.60E-01          4.73E-01          5.48E-02          0.05
intra-particlePorosity    4         1.36E-01         1         2.37E-01         2.44E+00
            2.71E+00          5.46E-01          4.74E-01          3.08E-02          0.05
intra-particlePorosity    4         1.36E-01         1         2.37E-01         1.16E+00
            2.71E+00          5.46E-01          4.64E-01          6.51E-02          0.05
intra-particlePorosity    4         1.68E-01         1         2.65E-01         5.43E-01
            2.71E+00          5.34E-01          4.40E-01          3.04E-02          0.05
intra-particlePorosity    4         2.79E-01         1         3.70E-01         2.08E-01
            2.71E+00          4.71E-01          4.71E-01          5.65E-02          0.05
```

7.2 Forward operator files

Listing 5: Forward operator definition file *WaxmanSmits.txt* for the Waxman-Smits model for electrical interface conductivity.

```
[ModelNameStart]
Waxman Smits
[ModelNameEnd]

[ModelParametersStart]
Name      lowerBound   upperBound   startingValue    referenceValue   weight   applyC1C2
F         1e-3         1e4          100              1e2              1        0
sigmaIF   1e-8         1            1e-5             0.5              1        0
[ModelParametersEnd]

[SampleSpecificsStart]
```

```
sigmaW
[SampleSpecificsEnd]

[SyntheticDataCalculationStart]
DataType      Expression
1             (1/(mod(1)))*sigmaW+mod(2)
[SyntheticDataCalculationEnd]
```

Listing 6: Forward operator definition file *Wyllie.txt* for the Wyllie's equation for p-wave velocity.

```
[ModelNameStart]
Wyllie's equation
[ModelNameEnd]

[ModelParametersStart]
Name          lowerBound    upperBound    startingValue    referenceValue    weight    applyC1C2
vPmatrix      0             8000          7500             6000              1         0
vfluid        0             8000          500              1000              1         0
[ModelParametersEnd]

[SampleSpecificsStart]
porosity
[SampleSpecificsEnd]

[SyntheticDataCalculationStart]
DataType      Expression
1             1/((1-porosity)*(1/mod(1)) + porosity*(1/mod(2)))
[SyntheticDataCalculationEnd]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% NOTES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
- use data directly (un-check "use log10 of data" check box)
```

Listing 7: Forward operator definition file *ColeCole.txt* for the Cole-Cole model of frequency-dependent, complex electrical conductivity.

```
[ModelNameStart]
Cole Cole
[ModelNameEnd]

[ModelParametersStart]
Name      lowerBound    upperBound    startingValue    referenceValue    weight    applyC1C2
rho0      1e-3          1e4           5e2              1e2               1         0
m         1e-6          1             0.5              0.5               1         0
tau       1e-7          1e6           100              1                 1         0
c         1e-6          1             0.5              0.5               1         0
[ModelParametersEnd]

[SampleSpecificsStart]
frequency
[SampleSpecificsEnd]

[SyntheticDataCalculationStart]
DataType      Expression
1             real(1/(mod(1)*(1-mod(2)*(1-1/(1+(1i*2*pi*frequency*mod(3))^mod(4))))))
2             imag(1/(mod(1)*(1-mod(2)*(1-1/(1+(1i*2*pi*frequency*mod(3))^mod(4))))))
[SyntheticDataCalculationEnd]
```

Listing 8: Forward operator definition file *DebyeDecomposition.txt* for the Debye Decomposition of frequency-dependent, complex electrical conductivity.

```
[ModelNameStart]
Debye Decomposition
[ModelNameEnd]

```

```
[ModelParametersStart]
Name     lowerBound   upperBound   startingValue   referenceValue   weight   applyC1C2
rho0     1e-3         1e4          5e2             1e2              10       0
m01      1e-6         1            1e-3            1e-3             1        1
m02      1e-6         1            1e-3            1e-3             1        1
m03      1e-6         1            1e-3            1e-3             1        1
m04      1e-6         1            1e-3            1e-3             1        1
m05      1e-6         1            1e-3            1e-3             1        1
m06      1e-6         1            1e-3            1e-3             1        1
m07      1e-6         1            1e-3            1e-3             1        1
m08      1e-6         1            1e-3            1e-3             1        1
m09      1e-6         1            1e-3            1e-3             1        1
m10      1e-6         1            1e-3            1e-3             1        1
m11      1e-6         1            1e-3            1e-3             1        1
m12      1e-6         1            1e-3            1e-3             1        1
m13      1e-6         1            1e-3            1e-3             1        1
m14      1e-6         1            1e-3            1e-3             1        1
m15      1e-6         1            1e-3            1e-3             1        1
m16      1e-6         1            1e-3            1e-3             1        1
m17      1e-6         1            1e-3            1e-3             1        1
m18      1e-6         1            1e-3            1e-3             1        1
m19      1e-6         1            1e-3            1e-3             1        1
m20      1e-6         1            1e-3            1e-3             1        1
m21      1e-6         1            1e-3            1e-3             1        1
m22      1e-6         1            1e-3            1e-3             1        1
m23      1e-6         1            1e-3            1e-3             1        1
m24      1e-6         1            1e-3            1e-3             1        1
m25      1e-6         1            1e-3            1e-3             1        1
m26      1e-6         1            1e-3            1e-3             1        1
m27      1e-6         1            1e-3            1e-3             1        1
m28      1e-6         1            1e-3            1e-3             1        1
m29      1e-6         1            1e-3            1e-3             1        1
m30      1e-6         1            1e-3            1e-3             1        1
[ModelParametersEnd]

[AuxiliaryStatementsStart]
nTau    = 30;
limTau  = log10([1e-5 1e4]);
tau     = logspace(limTau(1),limTau(2),nTau)';
[AuxiliaryStatementsEnd]

[AdditionalInputStart]
tau
[AdditionalInputEnd]

[SampleSpecificsStart]
frequency
[SampleSpecificsEnd]

[SyntheticDataCalculationStart]
DataType    Expression
1           real(1./(mod(1).*(1-sum(mod(2:end).*(1-(1./(1+1i*2*pi*frequency*tau)))))))
2           imag(1./(mod(1).*(1-sum(mod(2:end).*(1-(1./(1+1i*2*pi*frequency*tau)))))))
[SyntheticDataCalculationEnd]
```

Listing 9: Forward operator definition file *CrushedRock.txt* for recovering formation properties from measurements on crushed rock.

```
[ModelNameStart]
Crushed Rock
[ModelNameEnd]

[ModelParametersStart]
Name        lowerBound   upperBound   startingValue   referenceValue   weight   applyC1C2
zone        1e-6         1e-4         1e-5            2.0e-5           1        0
zoneFac     0.1          5            3.0             2                1        0
```

```
PhiIntra     5e-3     3e-1     0.01     0.02     1     0
m            1.66     2.6      2.0      2        1     0
phiFac       1        3        2.0      2        1     0
ImSigma      4        14       12       10       1     0
Splug        0.05     3        0.1      0.3      1     0
[ModelParametersEnd]

[AuxiliaryStatementsStart]
theta_shell= @(mod, particleSize) min([1;(1-(1-mod(1)./(0.5*particleSize*1e-3)).^3)]);
sigma_plug = @(mod,sigmaW,l,density) (mod(3)^mod(4)).*(sigmaW) +(1+1i*l)*((10.^(-mod(6)))*mod(7)*density*1e6*mod(3)^mod(4)*(1-mod(3))/mod(3));
sigma_shell= @(mod,sigmaW,l,density) ((mod(3)*mod(5))^mod(4)).*(sigmaW) +(1+1i*l)*((10.^(-mod(6)))*mod(7)*mod(2)*density*1e6*(mod(3)*mod(5))^mod(4)*(1-(mod(3)*mod(5)))/(mod(3)*mod(5)));
sigma_ep    = @(sigma_c,sigma_a,theta_shell) sigma_c*((2*theta_shell*sigma_c+(3-2*theta_shell)*sigma_a)/((3-theta_shell)*sigma_c+theta_shell*sigma_a));
F           = @(sigma_eff,sigma_w,sigma_i,PhiinterNeu) ((sigma_i-sigma_eff)/(sigma_i-sigma_w))*((sigma_w/sigma_eff).^(1/3))-PhiinterNeu;
[AuxiliaryStatementsEnd]

[SampleSpecificsStart]
particleSize, PhiinterNeu, density, sigmaW, l
[SampleSpecificsEnd]

[SyntheticDataCalculationStart]
DataType     Expression
1            real(fsolve(@(x) F(x,sigmaW,sigma_ep(sigma_shell(mod,sigmaW,l,density),sigma_plug(mod,sigmaW,l,density),theta_shell(mod,particleSize)),PhiinterNeu),1e-3+1i*1e-5,optimset('Display','off')))
2            imag(fsolve(@(x) F(x,sigmaW,sigma_ep(sigma_shell(mod,sigmaW,l,density),sigma_plug(mod,sigmaW,l,density),theta_shell(mod,particleSize)),PhiinterNeu),1e-3+1i*1e-5,optimset('Display','off')))
3            theta_shell(mod,particleSize)*mod(2)*mod(7) + (1-theta_shell(mod,particleSize))*mod(7)
4            theta_shell(mod,particleSize)*mod(5)*mod(3) + (1-theta_shell(mod,particleSize))*mod(3)
[SyntheticDataCalculationEnd]
```