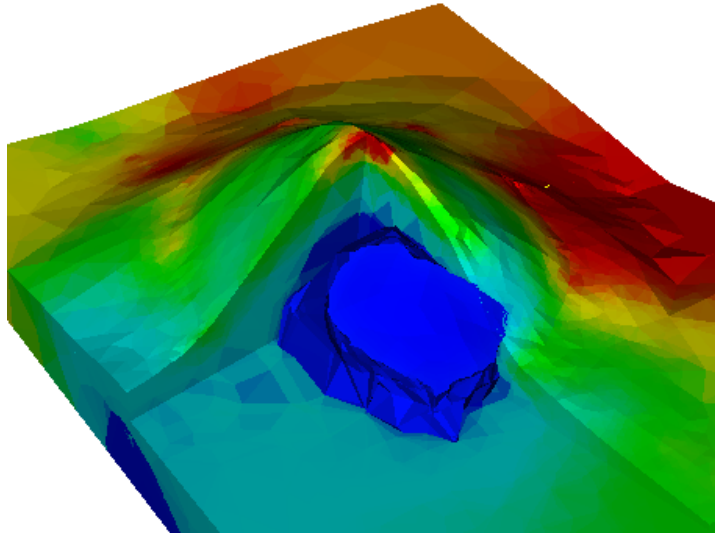


Boundless Electrical Resistivity Tomography BERT – the user tutorial



Thomas Günther* & Carsten Rücker†

August 30, 2011
version 1.0

In this tutorial we like to show the users of the BERT inversion how to work with our software. Some small but instructive examples, all real field cases, are presented to show how the different options in the configuration file can be used to yield case-specific inversion results.

The examples start from 2d inversion of surface measurements with and without topography. We show how to include structural information and how buried electrodes are handled. 3d inversion is done on a flat surface and topography. For the latter two main techniques are presented.

Also, measurements on closed objects, such as trees, humans, soil columns and model tanks are shown. Finally we show how to handle time-lapse resistivity measurements. The user is invited to follow by processing the data in the examples directory.

*Leibniz Institute for Applied Geophysics, Hannover

†Institute of Geology and Geophysics, University of Leipzig

Contents

1. Introduction	3
1.1. BERT, DCFEMLib, GIMLi - Nomen est omen	3
1.2. Options and commands	4
1.3. Visualization	5
2. 2D surface measurements	6
2.1. A flat surface example	6
2.2. Regularisation and data fit	7
2.3. Mesh quality and refinement	7
2.4. Incorporation of topography	8
2.5. Structural constraints	9
2.6. A sophisticated example: The lake case	10
3. 3D surface measurements	12
3.1. Flat surface	13
3.2. 3D Topography	13
4. Closed geometries	15
4.1. 2d example - tree tomography	15
4.2. 3d closed geometry - model tanks	17
5. Buried electrodes	19
5.1. 2D cross-hole data	19
5.2. 3D-Crosshole measurements	19
5.3. Topography and buried electrodes	20
6. Time-lapse ERT	20
6.1. Crosshole timelapse measurements	21
6.2. Soil column measurements	21
7. Concluding remarks and outlook	22
A. BERT for Windows users	23
B. Files and programs	24
C. Complete list of options and their default values	25
D. BERT version 2 using GIMLi	27

1. Introduction

1.1. BERT, DCFEMLib, GIMLi - Nomen est omen

Direct current electrical measurements are used in a wide range of applications such as medical imaging, geophysical surface or subsurface measurements or the investigation of trees and soil probes. This inverse problem is known under the terms ERT (electrical resistivity tomography), ERI (... imaging), EIT (... impedance tomography) or DC resistivity inversion. The aim of our software is to present a unique solution for all these cases.

Main advantage is the possibility to work on arbitrary geometries. Therefore we decided to consequently use unstructured finite element meshes for forward calculation as well as for the parameter identification. By the use of triangles (2d) and tetrahedrons (3d) we can follow any geometry of the subsurface, probe or any structural information we have about the object to be investigated. Due to this generality we decided to call it BERT - Boundless Electrical Resistivity Tomography.

BERT is a part of the software library DCFEMLib - Direct Current Finite Element Method Library. It is currently newly developed and will be called GIMLi - Generalized Inversion and Modelling Library. The latter is a much more general approach for arbitrary tomographic methods, as the functionality is growing BERT will more and more migrate to GIMLi¹.

DCFEMLib is licensed under the GPL (GNU public license). Our aim is to give back to the academic community without letting people earning money by just clicking a button. Additionally to the library itself it comprises executables for ERT modelling and inversion, tools for generating mesh input (the so-called poly-tools) and tools applying the mesh generator.

The theory and technology of BERT is described in our GJI paper (Günther et al., 2006) and bases on the finite element modelling techniques described by Rücker et al. (2006). It is represented by a triple-grid scheme: Most inversion algorithm use a dual-grid scheme, i.e. the forward calculation is calculated on a finer mesh. We add another one in order to use a secondary field approach and thus have a very fast forward calculation. Figure 1 shows the three grids: On a coarse and resolution-dependent grid the parameters are defined. On a globally refined and prolonged mesh the forward calculation is done. And a very fine primary mesh is used to calculate the primary potentials (for a homogeneous subsurface), but only once directly after the mesh creation.

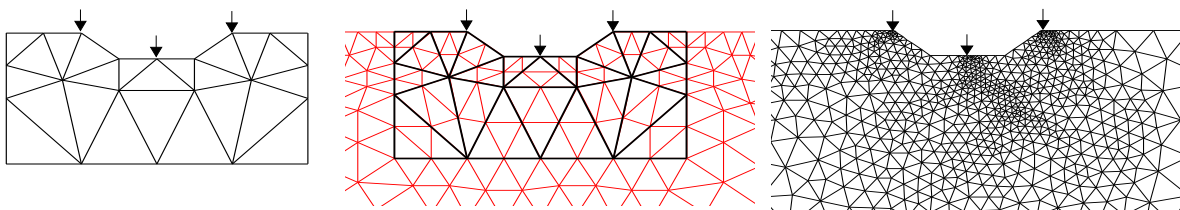


Figure 1: The three meshes of inversion for a 2d example, from Günther et al. (2006)

The overall scheme is visualised in Figure 2. It starts with the generation of the three meshes. Then the primary potentials are calculated and interpolated onto the secondary mesh. From this geometric factors are derived yielding the apparent resistivity and the sensitivity matrix is created for the homogeneous case. Finally the actual inversion is carried out: An inverse

¹See appendix for doing ERT inversions with GIMLi.

sub-problem is used to update the resistivity model, a forward calculation is carried out and checked against the data. The latter is done until the data are fitted well or the process stagnates.

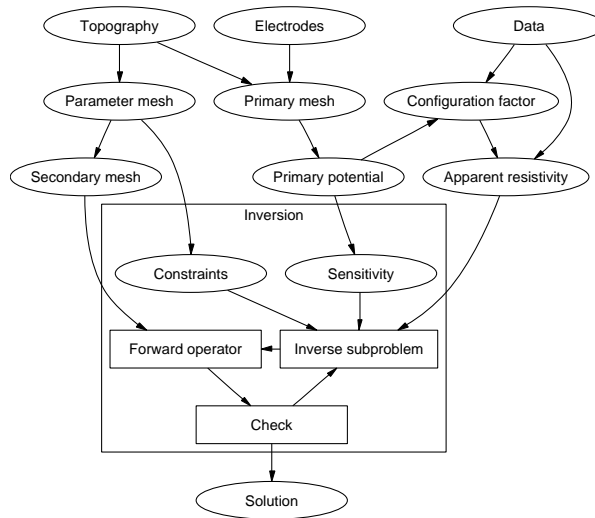


Figure 2: The BERT inversion scheme, from Günther et al. (2006): the geometrical information is used to prepare the actual inversion (rectangle).

BERT is available under Linux and Windows², either from pre-compiled binaries or self-compiled code³. The paths to the binaries and the library must be known, e.g. by setting

```

$ export PATH=$PATH:/path/to/DCFemlib/bin
$ export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/path/to/DCFemlib/lib

```

1.2. Options and commands

The inversion itself is controlled by the program `invert`, which reads the so called configuration (`cfg`) file. In the `cfg` file all necessary information is stored in form of lines consisting of `KEY=value` type, as in bash everything behind the `#`-sign is ignored and can be used for comments. Note that the key must be uppercase. There is only one mandatory key: the `DATAFILE` key holding the name of the data file. Other important keys are `DIMENSION` (2 or 3) and `TOPOGRAPHY` (0 or 1, meaning false or true). We suggest to create a new directory for each problem or also for different strategies to solve it. The data file must be in the unified resistivity.net format (see www.resistivity.net?unidata), but so far without the topography part and the end. `DC2dInvRes` and `DC3dInvRes` (graphical FD-based inversion software also available on www.resistivity.net) can easily be used to convert from different formats, such as `res2dinv` or different equipment files (`ABEM`, `Syscal`, `Geotom`, `Sting`, `4pointLight`). Use the `Save ohm file` call for doing so.

For list of possible options (with default values) see appendix C or call `invert opts`. However only few of them are of frequent use⁴. In order to create a new project there are special

²See appendix A for using BERT in Windows.

³See www.resistivity.net for information about how to obtain the binaries/codes and to compile the code

⁴In order to change default options permanently, we suggest to create a file `$HOME/.bertrc` that will be read if

commands for the individual tasks, namely `invertNew2d`, `invertNew2dTopo`, `invertNew3d`, `invertNew3dTopo`, `invertNew2dCirc` and `invertNew3dCyl` for the cases 2d/3d with or without topography and for circle/cylinder geometry. For example,

```
$ invertNew2d datafile.dat > inv.cfg
```

creates a new configuration file `inv.cfg` with the lines `DATAFILE=datafile.dat`, `DIMENSION=2` and `TOPOGRAPHY=0`, but also adds a lot of possible options for this case with an explanation, most of them inactive/commented. The user can now (or later) change the options and run single steps or the whole of inversion by

```
$ invert cfgfile commands
```

where `cfgfile` is the configuration file and `commands` can consist of the following:

all makes all, that is probably the first step in most (at least small) cases

meshs just makes the meshes, so that these can be optimized (suggested for bigger problems)

nomeshs do everything else but the meshes (after a successful mesh generation)

primPot interpolate calculates primary potentials and interpolates it to the secondary mesh

calcSensM actual inversion including sensitivity calculation (first inversion)

calc inversion with changed inversion options

newPara after creating a new parameter model (keeps primary mesh and potential)

save saves all important results (model&response for each iteration, log file, cfg file, meshes) in a directory called `result<date>_<time>`

clean cleans the directory from temporary results

mrproper deletes all stuff except input and result directories (releases disk memory fully)

show creates a vtk file (can be used in the main directory or in a result directory)

1.3. Visualization

BERT creates a lot of files that can be used for post-processing, see appendix. You can visualize in-place or save the state first using `invert inv.cfg save`. A result-directory is created out of the model file data and time and the files are moved in. Later you can change into the directory to do the visualization. There are several ways for doing it:

1. The command `invSummary` creates a pdf with all essential information and figures. This requires DCMatLab installed and is a easy method but only feasible for 2D.
2. You can use DCMatLab directly using

```
>> Mesh=loadmesh('mesh/meshParaDomain.bms'); %loads mesh into struct
>> res=load('model_iter.final'); %loads ascii resistivity vector
>> tripatchmod(Mesh,res); % help tripatchmod for options
```

present. Typical entries are, e.g., `SENSMATMAXMEM=3000` (available memory in MB), `LINSOLVER=2` (for LDL instead of `CHOLMOD` or `taucs`), or favourite mesh options

3. There is a graphical tool called bert2dpost for doing it automatically, where you can specify some options.
4. For 3D but also 2D you can call `invertinv.cfg show` which will create a file `showmodel.vtk` and opens ParaView with it. After `show` you can specify a different model file.

Paraview⁵ is a highly sophisticated and free tool for 3d (and 2d) visualisation. It handles both complete inversion results (obtained by `invert cfgfile show`), mesh inputs (*.poly, to be converted by `polyConvert -V -S` to vtk or stl) and meshes (to be converted by `meshconvert`, if not already done). In order to have a quick look at 3d meshes when finding optimal parameters, we also suggest MEDIT⁶, a free and fast 3d viewer.

The usual proceeding is (i) create meshes and look whether they are suitable, (ii) run a full inversion with `all`, (iii) look at the results (`show`, ParaView), (iv) refine options and do a `calc`, save good results using `save`. The first step is usually omitted for small projects. We now start to prepare and run inversions using the exemplary files in `dcfemlib/examples/inversion`, starting from easy-going to complicated ones.

2. 2D surface measurements

2.1. A flat surface example

The example in `dcfemlib/examples/inversion/2dflat/gallery` was friendly provided by the University of Mining and Technology, Freiberg (F. Donner). It is a very small profile over a known mining gallery that is used for draining water out of the mines. It represents a perfect two-dimensional anomaly since it strikes perpendicular to the profile and is a 2x2m cavity.

On a profile using 21 electrodes with 2m spacing, dipole-dipole measurements have been applied, the data quality was very good. The input in the data file `gallery.dat` is already the apparent resistivity. Standard deviations have been measured and 1% were added.

We create a configuration file using

```
$ invertNew2d gallery.dat > inv.cfg
```

just holding the data file and the dimension. Figure 3 shows the result of `invert inv.cfg` all with default options that converged with a chi-squared value of 0.99 (data fitted within error bounds). It clearly shows the cavity at about 20m and another anomaly whose origin is not completely clear from 2d measurements. See section 3.1 for inversion of a 3d data set.

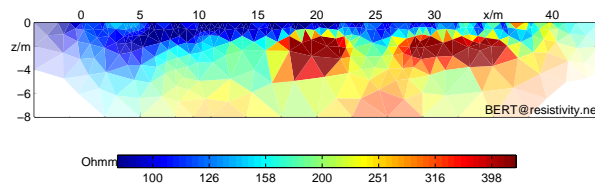


Figure 3: Result of the gallery example using default options.

⁵see ParaView home page www.paraview.org

⁶INRIA Rocquencourt, see <http://www-rocq.inria.fr/gamma/medit/medit.html>

2.2. Regularisation and data fit

We now might to change the characteristics of the model. The most important key for that is the regularization parameter `LAMBDA`. It controls the strengths of the smoothness constraints and thus defines how smooth the model will be. Therefore we test different values by commenting out the line with the parameter `LAMBDA` and setting it once to 200 and once to 2 instead of the default value of 20. After changing it we call

```
$ invert inv.cfg calc
```

and save the results as well using `save`.

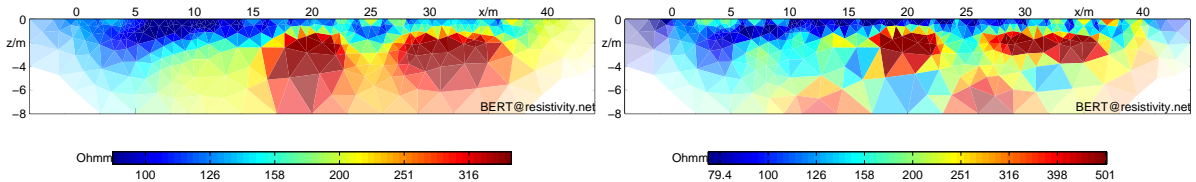


Figure 4: Result for a regularization parameter of 200 (left) and 2 (right).

Figure 4 shows the result for the two cases. The one for $\lambda = 200$ is clearly over-smoothed and cannot fit the data appropriately ($\chi^2 = 4.4$, $\text{RMS}=2.5\%$). The low value can fit the data but produces too much small-scaled anomalies that are not necessary.

Note that these data have a very good quality documented in the error values between 1 and 2.3%. An over-estimation of the errors will lead (if $\chi^2 = 1$) to an over-smoothed result as well. Errors (in the meaning of "how well can I fit the data") should therefore be measured or estimated by reciprocal data or repetition. Always add a fixed value that will account for error sources outside our measurements⁷. In the absence of errors in the data file an estimation is made using a fixed percentage (`INPUTERRLEVEL`) and a voltage error (`INPUTERRVOLTAGE`). If the current is not in the file, a value of 100 mA is assumed. For different current strengths the voltage error has to be adapted.

Alternatively to a fixed regularization it can be optimized using the L-curve by `LAMBDAOPT=1`. To decrease the regularization parameter in the course of iterations (cooling scheme) by, e.g., the factor 0.8 use `LAMBDADECREASE=0.8`.

2.3. Mesh quality and refinement

The depth of the modelling domain is by default automatically determined based on 1d sensitivity studies, but can be adjusted using `PARADEPTH`. The value of `PARABOUNDARY` defines how far (in % of the extension, default=5) the boundary is outside of the electrodes. You might of course think that the triangles in Figure 3 are quite coarse and that we can resolve the cavity better. Although we don't need it in this case according to Occam's razor, we illustrate how to do it.

Generally, all meshes must be fine near the electrodes (to achieve high accuracy in the forward calculation and because the resolution is high there) and coarse at the boundaries. The fineness is achieved by placing additional nodes in the vicinity of the electrodes, `PARADX` holds the distance (in electrode spacings a). A value of 0.2 means two points are created $0.2a$ to the right and left of each electrode. By `EQUIDISTBOUNDARY=1` several points are equally distributed.

⁷ Therefore `DCxdInvRes` ask for an error percentage to be added to the measured if those are too small.

Usually the points are linearly interpolated, `SPLINEBOUNDARY=1` forces a spline interpolation, which is useful for "round" geometries or smooth topography.

The coarse-ness towards the boundary is controlled by the mesh quality `PARA2DQUALITY`, which denotes a minimum angle. The higher the quality is, the more accurate are the results but with an increasing number of cells and thus run-time. In `triangle`⁸ (Shewchuk, 1996) version 1.6, our favoured 2d mesh generator, the range goes from 25-30 (bad quality) to 34-35 (good quality). Default value is 33.8, a trade-off between accuracy and run-time. Another way of avoiding a too coarse mesh is the maximum cell size by setting `PARAMAXCELLSIZE` (in m²).

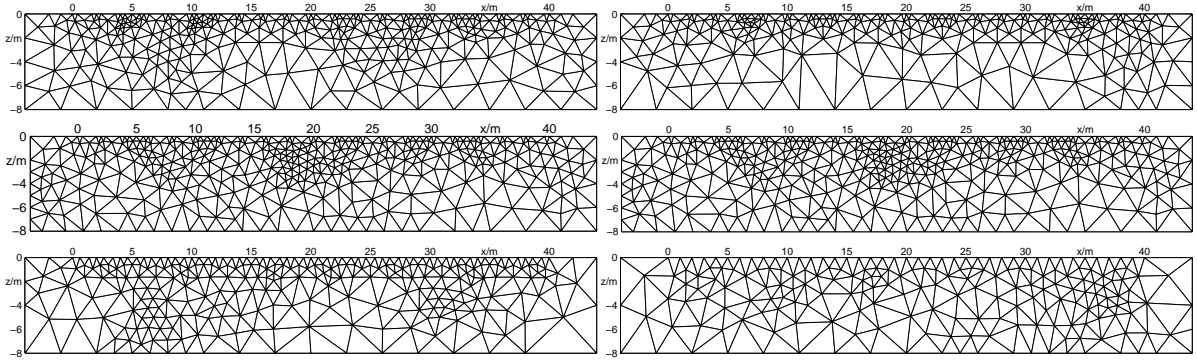


Figure 5: Meshes with different parameters: default values $dx=0.2$ & $q=33.8$ (top left), $dx=0.2$ & $q=33$ (top right), $dx=0.2$ & $q=34.5$ (center left), $dx=0.2$ equidist & $q=33.8$ (center right), $dx=0.3$ equidist & $q=33.8$ (bottom left), $dx=0.5$ equidist & $q=33.8$ (bottom right)

In Figure 5 the resulting parameter meshes for different settings are displayed. A lower quality is only recommended in case of topography. A too large quality value produces nicer triangles but can lead to heterogeneous mesh density even for equidistant electrodes. The `EQUIDISTANT` switch produces a nice near-surface triangulation that can yield too much parameters. The latter can be avoided by a larger `PARADX` value. Finally the user has to find the optimum settings for his case (and taste).

2.4. Incorporation of topography

In 2d inversion, topography is easily integrated by setting the heights of the electrodes. All the rest should be done automatically, if necessary, additional electrodes must be inserted. However, rarely all electrodes will be measured topographically. Often it is sufficient to have a few points. Note that in the current stage BERT requires the topographical information in the first section, not at the end of the file. For this case we recommend the use of `DC2dInvRes` (Günther, 2007) that will roll the positions along the surface. For this task, use `Data:Save Ohm file` and specify whether the x values are along measure tape or real x.

The slagdump profile

The data in `dcfemlib/examples/inversion/2dtopo/slagdump` were friendly provided by the Federal Institute of Geology and Natural Resources (BGR), Hannover. It was measured by

⁸see <http://www.cs.cmu.edu/~quake/triangle.research.html>

M. Furche and is one of several profiles over a slag dump. A Wenner array with $a = 2\text{m}$ spacing was applied yielding 222 data points. The topography was measured at 8 points by levelling and appended to the original file before converting it with DC2dInvRes. We initialize the inversion with standard options using the command:

```
$ invertNew2dTopo slagdump.ohm > inv.cfg
```

Note that this produces additional lines in the file for the primary mesh. Similar to the parameter mesh, the PRIMDX value specifies the (absolute, not relative!) refinement at the electrodes and PRIM2DQUALITY defines the mesh growth. Additionally, we decide by PRIMP2MESH whether the primary potentials shall be computed by quadratic shape functions. As stated by Rucker et al. (2006) those are more efficient. The necessary refinement for a P2 mesh is about $a/10$ and $a/100$ for a P1 mesh.

Figure 6 shows the topographic effect and the resistivity image. The upper pseudo-section shows the apparent resistivities using flat-earth geometry factors. The middle one is the pure geometric effect, $t = G_{flat} \cdot u_{topo}(\sigma = 1\text{S/m})$, and the lower image is based on the geometric factors out of the primary potential. Several anomalies can be explained solely by topographical undulations. The resistivity distribution shows a conductive interior and a resistive hard pan.

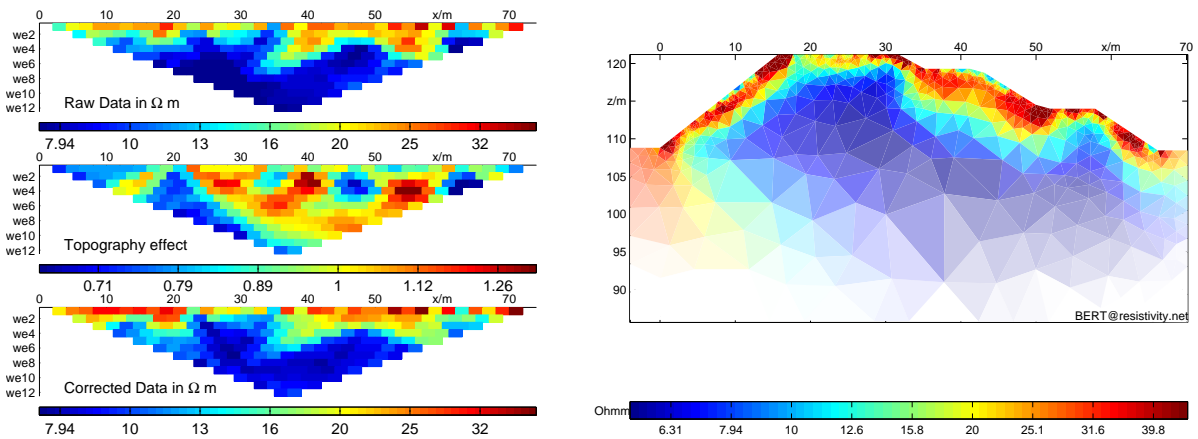


Figure 6: Topographic effect and inversion result of the slagdump site.

2.5. Structural constraints

Often there is additional information about the subsurface. An incorporation into the inversion process is always to be preferred over a comparison of the results. In (Günther and Rucker, 2006) we presented a more general minimisation approach that allows for arbitrary weights for each boundary between model cells. In existence of a known discontinuity this can be set to zero allowing for (but not enforcing) an arbitrary jump in resistivity.

The following example (`dcfemlib/examples/inversion/inversion/2dstruct`) was measured and friendly provided by the K-UTec GmbH Sondershausen (T. Schicht). Aim of the study was bedrock detection carried out with resistivity and refraction seismics. The velocity structure showed to be a very clear 2-layer case. So the result (layer boundary) of the refraction study can serve as structural information.

The file `bedrock.xz` contains the course of the boundary as x-z pairs. We now include this file into the configuration file using the INTERFACE option. In order to compare the result with

and without structure we call

```
$ invertNew2d bedrock.dat > inv.cfg
$ invert inv.cfg all save
$ echo INTERFACE=bedrock.xz >> inv.cfg
$ invert inv.cfg newPara save
```

The latter (*newPara*) switch is used if we change parameterisation. So the primary mesh and potentials do not have to be calculated anew, but they have only to be interpolated onto the forward mesh and the sensitivities are re-calculated as well. Figure 7 shows the subsurface images without and with the structural information.

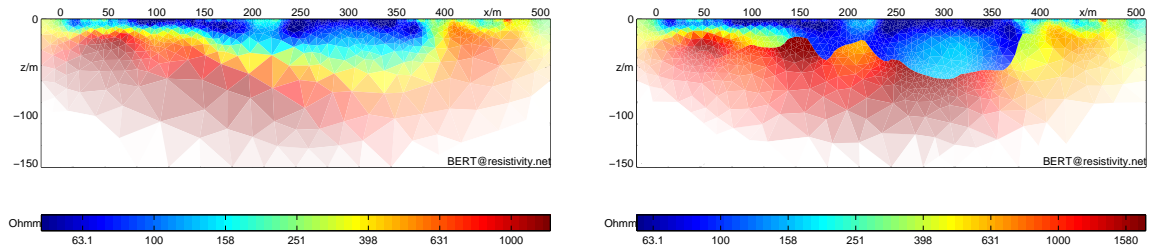


Figure 7: Resistivity distribution without (left) and with (right) structural information.

Obviously the additional information leads to a much clearer image of the subsurface. At most positions there is a sharp resistivity contrast at the boundary. However at some positions there is either a difference to velocity or the refraction result is ambiguous.

2.6. A sophisticated example: The lake case

The last 2d example was measured by the Leibniz Institute for Applied Geosciences, Hannover (W. Südekum and T. Günther). Its aim was to delineate sedimentation structures beneath the Feldungel lake near Osnabrueck. Electrodes have been spread out from one shore along the lake bottom onto the other shore. The spacing was 2m and both Wenner-alpha and Wenner-beta were measured and combined. Since the lake resistivity is known ($22.5 \Omega\text{m}$) this is excluded from inversion by hand:

1. We put the electrode positions (0 to -2.6m height) and resistances into the input file
2. We start as for a topographic case and generate the meshing input

```
$ invert inv.cfg domain
```
3. As a result we obtain the poly file *mesh/mesh.poly* which we copy to *mymesh.poly*⁹
4. We need to add the water surface by an edge between the left and the right shore. A view into the poly file shows these being represented by the points 3 and 138. So add another edge at the end of the edges (line 303) by inserting the line *151 3 138 -1* (number n1 n2 edgemarker) and increase the number of edges in line 152 from 151 to 152

⁹see triangle page <http://www.cs.cmu.edu/~quake/triangle.research.html> for file description

- Finally we add a region marker somewhere in the lake with marker 1 (not inverted) by appending the line `2 50 -1 1 0.0` (number x y marker maxtrsize) and increasing the number of regions from 2 to 3.

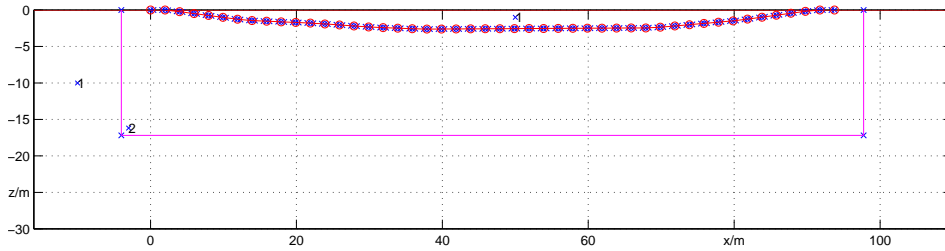


Figure 8: Representation of the input PLC for the lake case.

Figure 8 shows a section of the input PLC (the outer space needed for forward calculation is much bigger). All nodes are denoted by a cross. The electrodes, which are refined by additional nodes on both sides, are marked by a circle. They form the topography as in the 2d topographic examples. The red rectangle defines the region to be inverted. Therefore the marker 2 (inversion region) is in this region and the marker 1 (no inversion) is outside.

We use this altered poly file in the inversion by introducing into the cfg file

```
PARAGEOMETRY='cp mymesh.poly mesh/mesh.poly'
```

This means that mesh.poly is created by copying mymesh.poly. Alternatively we can put here a shell script or matlab call that makes up our geometry. Furthermore we add the following options (obsolete in GIMLi due to better automatics and the region technique):

```
TOPOGRAPHY=0      # ensures that the primary potentials are really analytic
SPACECONFIG=2     # ensures that mirror sources will be used (obsolete?)
RHOSTART=22.5    # Start(=background) resistivity
NOPROLONGATION=1 # no resistivity prolongation (outside & lake stay const)
SURFACESMOOTH=0  # prevent smoothing (edge swapping possible)
```

Furthermore we use the following options

```
ZPOWER=0.3       # enhances layered (sediment) structures
OVERRIDEERROR=1  # do not use the measured errors in file (optimistic), but:
INPUTERRLEVEL=2   # 2% plus
INPUTERRVOLTAGE=20e-6 # 20 microvolts
```

and the inversion converges at $1 < \chi^2 < 2$. Since the data file also contains induced polarisation (IP) phases, they are automatically inverted at the end using a single inverse sub-step with the same options as for the amplitude. Figure 9 shows resistivity and phase distribution. The lake sediments show generally by low resistivities and phases. Particularly the youngest sediments are almost as conductive as the water. Below different sediment layers are visible giving hints how the historic topography was covered by sediments step by step.

Note that for pure underwater measurements there is an additional option `UNDERWATER` which forms the sea bottom topography by the electrodes and adds the water surface at $z = 0$ automatically.

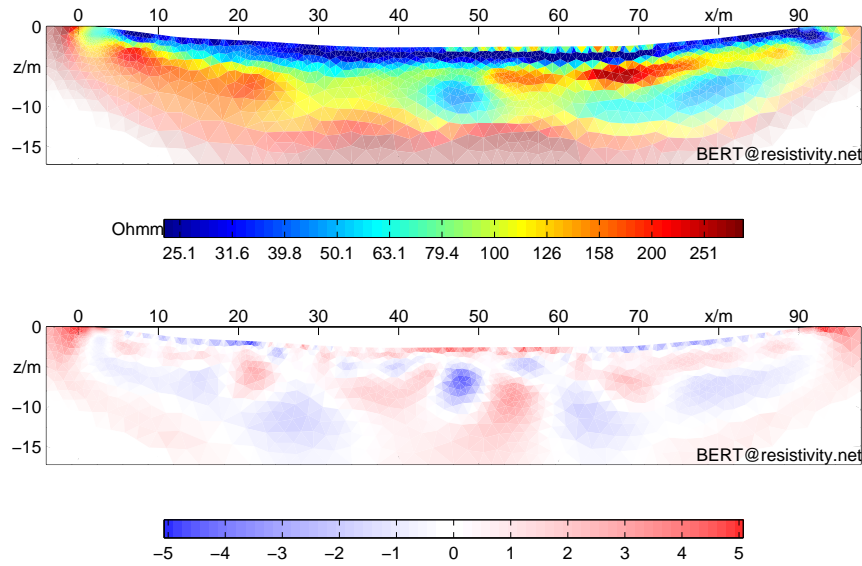


Figure 9: Inversion result of the water case: resistivity (top) and phase (bottom). Note that the water is not part of the inversion mesh and thus not plotted.

3. 3D surface measurements

3D surface measurements can be carried out in several variants:

1. Layout of an electrode grid. However, due to the limited electrode number grids are restricted to small areas.
2. Parallel (and perpendicular) profiles along the coordinate axes.
3. Profiles in arbitrary directions due to accessibility limits.
4. Non-profile layout, e.g. large-scale dipole-dipole experiments.

In any case, the electrode positions and measurements must be defined according to the unified data format. The data for the first two types can be easily organized by hand. For number 3 (and 2) we suggest to prepare 2d files and to write a pro-file containing of lines with the 2d file name and x-y pairs of points where the line is going. This file can be read into DC3dInvRes Günther (2008) and used to write the 3d file. In case of topography it is best to do the tape correction on the 2d files before using DC2dInvRes and Export Ohm.

The most flexible element in 3d is the tetrahedron. The tetrahedralization is done by a mesh generator. Our choice is Tetgen (Si, 2008), a free and versatile quality mesh generator. The quality measure is different from 2d and describes a radius-to-edge ratio, note that small values point to higher quality. Appropriate values for (primary field) forward calculation are 1.12 to 1.2, for the inverse (and thus secondary) mesh values of 1.2-1.5 are appropriate, the keys are called PRIM3DQUALITY and PARA3DQUALITY.

3.1. Flat surface

In `dcfemlib/examples/inversion/3dflat/gallery` is a data set in the field where the 2dflat example (section 2.1) was measured. It comprises a grid of 9x14 electrodes. Dipole-dipole measurements have been measured on all x and y profiles. In the data file is an error of constant 0.0 that will be overridden automatically. An inversion project file with default parameters is created by:

```
$ invertNew3d rothschoen.dat > inv.cfg
```

The inversion is then fully run (with command `all`) converging to a chi-squared misfit of about 2 (rrms=4-5%). In order to fit the data better, the regularization parameter is decreased using `LAMBDA=5`, which leads to an relative rms error of about 3% ($\chi^2 = 1$). Another way is to use an anisotropic regularization (`CONSTRAINT=6`) also leading to $\chi^2 \approx 1$. (Note that constraint 6 is somewhat deprecated and replaced by the `ZPOWER` options, where we can control the flatness continuously. Try out different `ZPOWER` values between 0 and 1.)

The result is saved and converted to a vtk file¹⁰ using

```
$ invert inv.cfg show
```

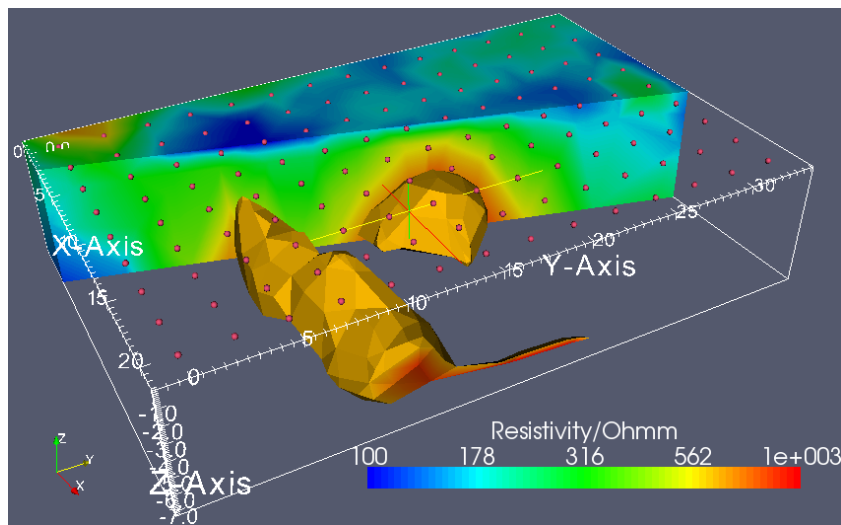


Figure 10: Inversion result of the 3d gallery data set using a smoothed iso-surface of 650 Ω m and a Plane Clip, the red spheres are the used electrodes.

Figure 10 shows a Paraview visualisation that has been created by the following steps: i) Cell Data To Point Data, ii) Clip by Scalar 650 (Ω m), iii) Extract Surface, iv) Smooth Surface, v) Another Clip based on Cell2Point with Plane, vi) representation of the input as Outline and Cube Axes. The color bar is logarithmic with a manual range of 100-1000 Ω m. The electrodes have been included as point vtk file and displayed by Glyph as Spheres of radius 0.05. After some exercise the reader will be able to create nice images, plots and calculate results such as extensions or volumes of geological bodies.

3.2. 3D Topography

The definition of a 3d topography is much more complicated than in 2d, where every shape can be described by a simple polygon. The input PLC consists of faces instead of edges, the

¹⁰can be displayed in 3d software ParaView, see <http://www.paraview.org>

resulting poly file has a similar but different format¹¹. Generally the proceeding is the following: i) create a flat surface mesh, ii) interpolate heights from topographic information, iii) make a small (inversion mesh) and a large (forward mesh) box around it, iv) make refinement, if necessary, and v) create the mesh using tetgen.

For specifying topography, there are two different ways:

- the electrodes in the data file have an elevation and all other points are interpolated
- there is a digital elevation model (DEM) or at least a list of measured topo points (in a 3-column file containing x,y and z)

Whereas the first case is sufficient for smooth topography and/or dense electrode coverage, the latter is more general. The topographic points are Delaunay triangulated. For every point of the meshes, also the electrodes, the elevation is linearly interpolated. Therefore electrodes with measured elevations should be included in the topo file as well to make sure their z values are correct. We specify this topographical list by the line `TOPOPOINTS=filename`.

In `dcfemlib/examples/inversion/examples/acucar` there is a project measured by the Federal Institute of Geology and Natural Resources (BGR) Hannover (M. Furche¹²). The site is an old slag dump that comprises a topography reminding on the sugar hat in Rio. Two resistivity profiles have been measured crossing the top of the isolated hill. Another profile was realised around the hill in a more or less constant elevation. Although this is not a dense sampling as an electrode grid it should be sufficient to obtain a rough image.

Additionally to the electrodes, some topographical points have been measured and put into the file `points.xyz`. So we create a new project using

```
$ invertNew3dTopo acucar.ohm > inv.cfg
```

and add the line `TOPOPOINTS=points.xyz` to the model. If we now call `invert inv.cfg meshes` we see the mesh does not show the hill, since the topography overrides the electrode elevation. Therefore we have to add the electrode definition (lines 3-230) to the topography file and see then the hill (Figure 11 left). However due to the point density the electrode line appears as a sharp edge that is not really the truth but sufficient in this case.

In other cases we might have a digital elevation model. In order to show this on the same example, we created one by cubic interpolation of the available points on a regular grid of 2m spacing. In order to avoid interpolation errors between the electrodes we created a polygon file `poly.xyz` for the three profiles¹³ and introduce it by `TOPOPOLY=poly.xyz`. Figure 11 shows the surface mesh of both variants. The sharp edges are now disappeared.

Finally the inversion result is visualised in Figure 12. It shows a conductive interior of the slag dump and different sediments at the surface, e.g. a resistive top. Of course the data coverage is low between the profiles and at the model boundaries. Therefore the model becomes more or less interpolated by the smoothness constraints.

¹¹See <http://tetgen.berlios.de>

¹²Now at Leibniz institute of Applied Geosciences

¹³Several polygons are separated by a blank line.

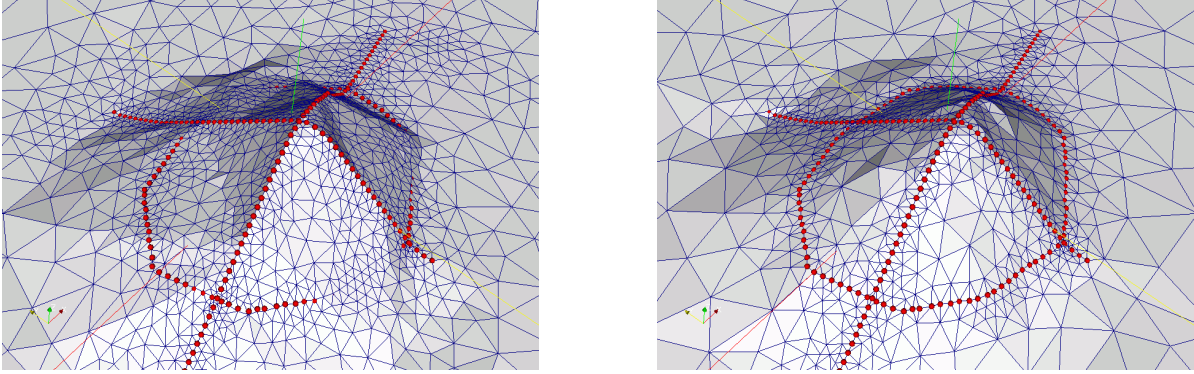


Figure 11: Surface mesh for the point-wise topographic information (left) and the digital elevation model (right), the electrodes are shown as red points.

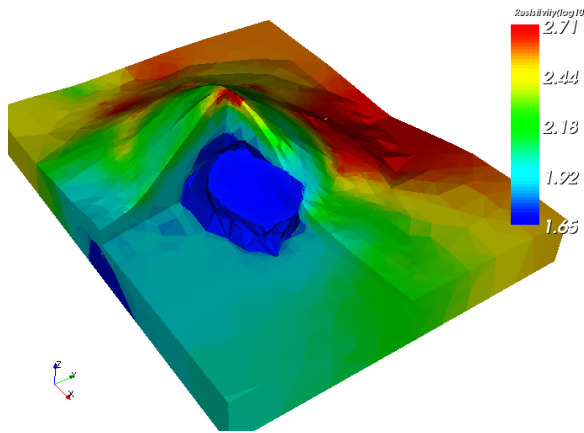


Figure 12: Inversion result of the 3dtopo case.

4. Closed geometries

Closed geometries are actually easier than open ones since we do not need a mesh prolongation and two different regions. However since the whole boundary is of Neumann type, we must ensure two additional conditions that are not necessary in the open case:

- The current cannot vanish in infinity, therefore we must use dipole sources, e.g. by a reference current node.
- Since only derivatives are present in the boundary value problem, we must make the forward solution unique, e.g. by adding a reference potential node, whose potential is forced to zero.

4.1. 2d example - tree tomography

For 2d bodies the electrodes are usually on the boundary and the PLC can easily be formed by a polygon through it as in the 2d mesh generation. If additional points between the electrodes are

needed, they can be introduced by fake electrodes. For tree (or human) geometry a dedicated GUI named TreeBERT (before DC2dTree) was created making it easy to process the data visually.

Tree tomography

EIT on trees has been successfully established to investigate decay of trees. The example in *dcfemlib/examples/circle/tree* was measured and friendly provided by Niels Hoffmann, HAWK Göttingen. It represents a lime tree, measured by 24 steel electrodes that are plugged into the bark. Dipole-dipole measurements have been applied using a Geotom equipment. The configuration file reads as follows

```
DATAFILE=hollow_limetree.ohm
DIMENSION=2
TOPOGRAPHY=1      # activates the primary mesh
CYLINDER=1       # defines a closed geometry
SURFACESMOOTH=1  # makes a nicer surface
EQUIDISTBOUNDARY=1 # equidistant refinement
PARADX=0.2       # 5 segments between the electrodes
PARA2DQUALITY=34.8 # very good quality, almost the upper limit
SPLINEBOUNDARY=1 # round geometry
PRIMDX_R=0.001  # refinement of primary mesh in radial direction
LAMBDA=10       # regularisation strength
BLOCKYMODEL=1   # enhance contrasts by robust (L1) methods
```

For this case an equidistant refinement, the use of splines and a high quality ensures a nice mesh with a round boundary. The primary refinement is done in radial direction. Additionally we used the robust modelling in order to obtain a clearer contrast of the high resistivity.

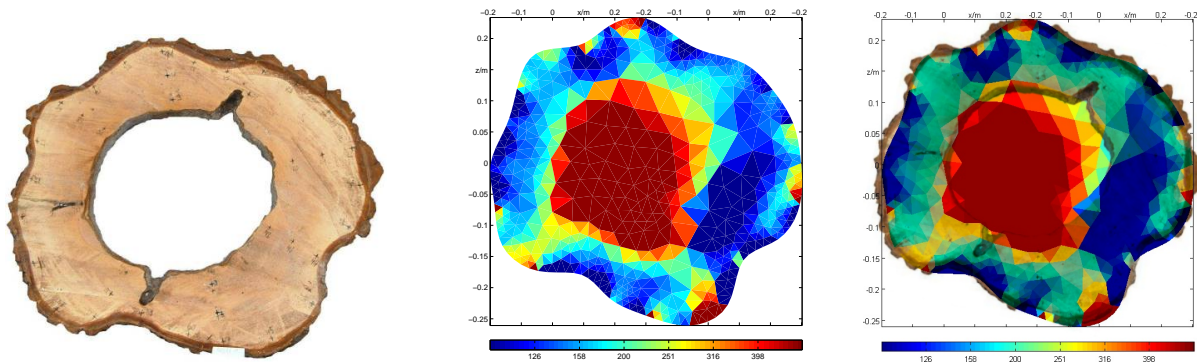


Figure 13: Tree cut (left), inversion result (center) and overlay.

After the measurements the tree was cut and revealed a cavity inside caused by decay. Figure 13 shows a photograph, the inversion result and an overlay of both. Clearly the cavity is marked by high resistivity that is in almost perfect accordance with the photo.

4.2. 3d closed geometry - model tanks

In the Federal Institute of Geology and Natural Resources (BGR) Hannover, a cylindrical model tank was created in order to make infiltration experiments with material from the slag dump. The column has diameter of 30cm and a height of 80cm. In each of 5 rings with 5cm vertical distance 24 steel electrodes of 2cm length were installed. Dipole-dipole measurements have been applied to all rings yielding a number of 320 data. The example is located in *dcfemlib/examples/inversion/3dtank*.

Since the parameterization cannot be detected automatically from the file, we have to create the mesh input, i.e. the PLC in *mesh/mesh.poly* by hand using a script. There is a poly tool *polyCreateCube* creating a unit cube. With the option *-Z* it creates a unit cylinder instead, which has to be scaled appropriately. Then we put in the electrodes as points¹⁴ with the marker *-99*. We insert two additional nodes with markers *-999* and *-1000* that are used for current reference and potential reference. So the script reads:

```
MESH=mesh/mesh # PLC name
polyCreateCube -v -Z -s 48 -m 2 $MESH # create unit cylinder with 48 segments
polyTranslate -z -0.5 $MESH # moves it such that top is zero
polyScale -x 0.3 -y 0.3 -z 0.8 $MESH # scale to radius 0.15 & height 0.8
cat soil_column.dat | head -n 82 | tail -n 80 > elec.xyz # extract electrodes
polyAddVIP -m -99 -f elec.xyz $MESH # add electrodes to mesh
polyAddVIP -m -999 -x 0 -y 0 -z 0 $MESH # current reference node
polyAddVIP -m -1000 -x 0 -y 0 -z -0.8 $MESH # potential reference node
polyConvert -V -o $MESH-poly $MESH # convert to vtk to load it to paraview
```

We create an empty cfg file (or use *invertNew3dCyl*) with the lines

```
DATAFILE=soil_column.dat
DIMENSION=3
TOPOGRAPHY=1
CYLINDER=1 # ensures the closed geometry
```

We add our PLC script to the *PARAGEOMETRY* variable such that *mesh/mesh.poly* will be created by it.

```
PARAGEOMETRY=mymesh.sh # make sure that mymesh.sh is executable
```

A further refinement can be achieved by quality improvement (*PARAQUALITY*), local refinement (*PARADX*) or maximum cell size (*PARAMAXCELLSIZE*). In order to obtain an accurate we use a refinement for the primary mesh of 1cm and quadratic shape functions ending in about 32000 nodes.

```
PRIMDX=0.01
PRIMP2MESH=1
```

Figure 14 shows the course from the mesh input via the parameter mesh to the final result.

¹⁴Since the electrodes cannot be show a significant extension compared to the column size, we put the points not onto the surface but moved it 1cm inside.

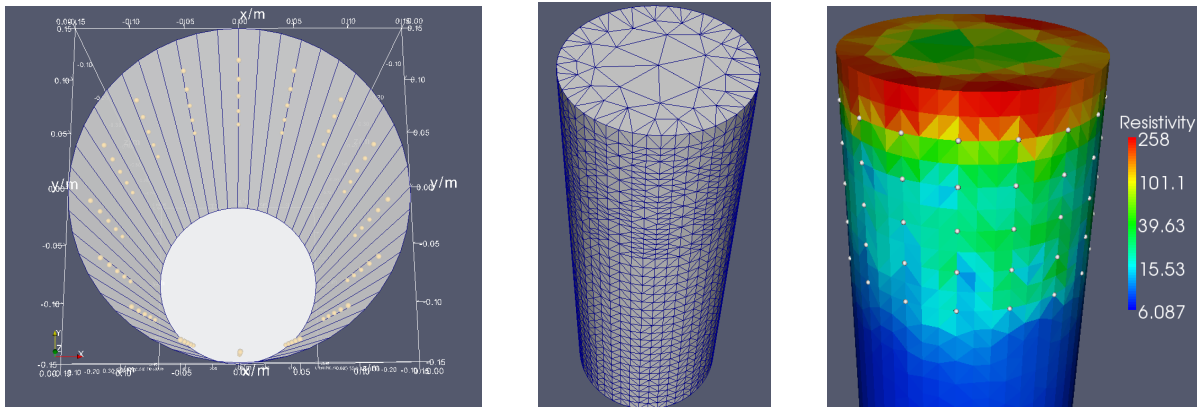


Figure 14: PLC (left), parameter mesh (center) and inversion result (right) of the soil column experiment.

5. Buried electrodes

5.1. 2D cross-hole data

Of course cross-hole measurements can also be inverted using BERT. The height of each electrode must be set to the elevation minus depth. However, since we cannot distinguish whether it is topography or a buried electrode we must create the geometry by hand. Note, that for this reason we introduced the tokens `h` (height) and `d` (depth below surface) to the data format, but this is only recognized by GIMLi.

The example in `dcfemlib/examples/inversion/2dxh` was produced by O. Kuras of the British Geological Survey (BGS) in the ALERT project (Kuras et al. (2009)). It represents about 1300 data obtained by cross-hole measurement between 5 very shallow (0-1.6m) boreholes. In order to create an inversion mesh we would create a small box with marker 2 (inversion) inside of a big box that is used for forward calculation (marker 1) by 8 points and 8 edges.

This is more rigorously implemented by the script `polyFlatWorld` which automatically calculates the size of the model and the boundary around the electrodes from the parameters `BOUNDARY`, `PARABOUNDARY` and `PARADEPTH`. To pass the variables we use the Unix command `source` and introduce `PARAGEOMETRY='source polyFlatPara 2dxh.ohm'` into the `cfg` file. Additionally we set `SPACECONFIG=2` for subsurface sources and a pretty good quality for the parameter mesh `PARA2DQUALITY=34.5`. As before, we can use `PARADX` to refine the model at the electrodes. In this case we introduce a node between each of the 0.1m separated electrodes using `PARADX=0.05`. Note that, different from 2d surface measurements, it is treated by `polyFlatWorld` as an absolute value. We use a relatively large regularization parameter `LAMBDA=300` for suppressing oscillations at the electrodes. Figure 15 shows the obtained resistivity distribution at the very beginning of a tracer experiment. See section 6.1 for time-lapse inversion.

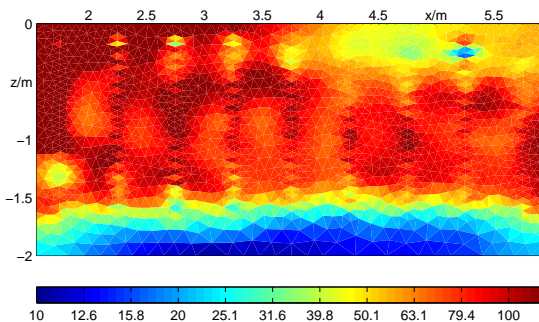


Figure 15: Inversion result of the crosshole data set.

5.2. 3D-Crosshole measurements

Crosshole measurements can of course be applied three-dimensionally. The example in `dcfemlib/examples/inversion/3dxh` was presented by J. Doetsch from ETH Zurich. The data file `3dxh.ohm` comprises 753 data between 4 boreholes in the saturated zone ($d=4-10\text{m}$) and is part of a monitoring experiment. We create a `cfg` file using

```
DATAFILE=3dxh.ohm
DIMENSION=3
```

```
SPACECONFIG=2      # for mirror sources at z=0
PARABOUNDARY=15   # to get a bit more space around the electrodes
PARAGEOMETRY="source polyFlatWorld $DATAFILE"
```

By using `ZPOWER=0.3` we can enhance the predominantly layered structures. The inversion converges then with defaults down to about $\chi^2 = 1$. Figure 16 shows the final result.

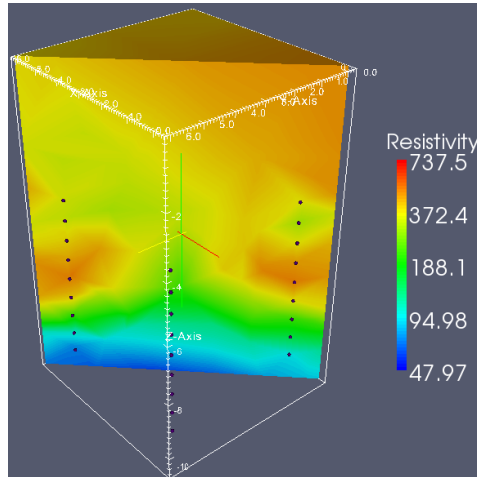


Figure 16: Inversion result for the 3D crosshole case.

5.3. Topography and buried electrodes

If both heights and depths below surface are given, we cannot use `createSurface/createParaMesh` and `polyFlatWorld`. In this case we suggest to create a data file with the surface electrodes (either with topo or with zeros and a topo file) and a list of subsurface electrodes including real. The normal PLC generation is done with the first and the latter are then added using `polyAddVIP`. Importantly the order of electrodes must be set such that first the surface electrodes and then the buries electrodes appear. The resulting script is introduced using `PARAGEOMETRY=myscript.sh` which can look as follows:

```
cp inv.cfg surface.cfg      # makes a copy with all options
echo DATAFILE=datafile-without-electrodes.dat >> surface.cfg
invert surface.cfg domain   # creates mesh/mesh.poly
polyAddVIP -m -99 -f borehole-electrodes.xyz mesh/mesh.poly
```

Finally all electrodes are in the PLC in the right order and all should be well.

6. Time-lapse ERT

We are often interested in ongoing physical processes and use ERT for monitoring experiments. There are many different time-lapse inversion approaches. Here we use a very simple one that solves the linear subsystem for the resistivity ratio of two data sets after the inversion. An arbitrary number of subsequent data sets can be processed by writing their file names in a text file and pass it by `TIMESTEPS=filename`. Up to now, the data files must contain identical configurations in the same order.

6.1. Crosshole timelapse measurements

Let's go back to the crosshole case 2dxh (see section 5.1) and unpack the time data files in 2dxh-timelapse.zip. In 2006 the BGS injected a highly saline tracer in borehole number 8 and measured 36 data sets every 40 minutes such that a whole day was covered. The subsequent files are named 01.dat, 02.dat, ... and are assembled in timesteps.txt.

By including `TIMESTEPS=timesteps.txt` and calling `invert inv.cfg calc` again. As a result we obtain a lot of `model_i.res.vector` containing the resistivity values for one time step each. By a `show` command they are transferred into paraview files ParaView recognizes this names as time steps and allows an easy scrolling through the times. Figure 17 shows some selected time steps that allow for seeing the tracer flow toward the left boundary. Note that these are only preliminary results that are used to present how BERT is working. With more sophisticated time lapse strategies the monitoring process can be traced more accurately.

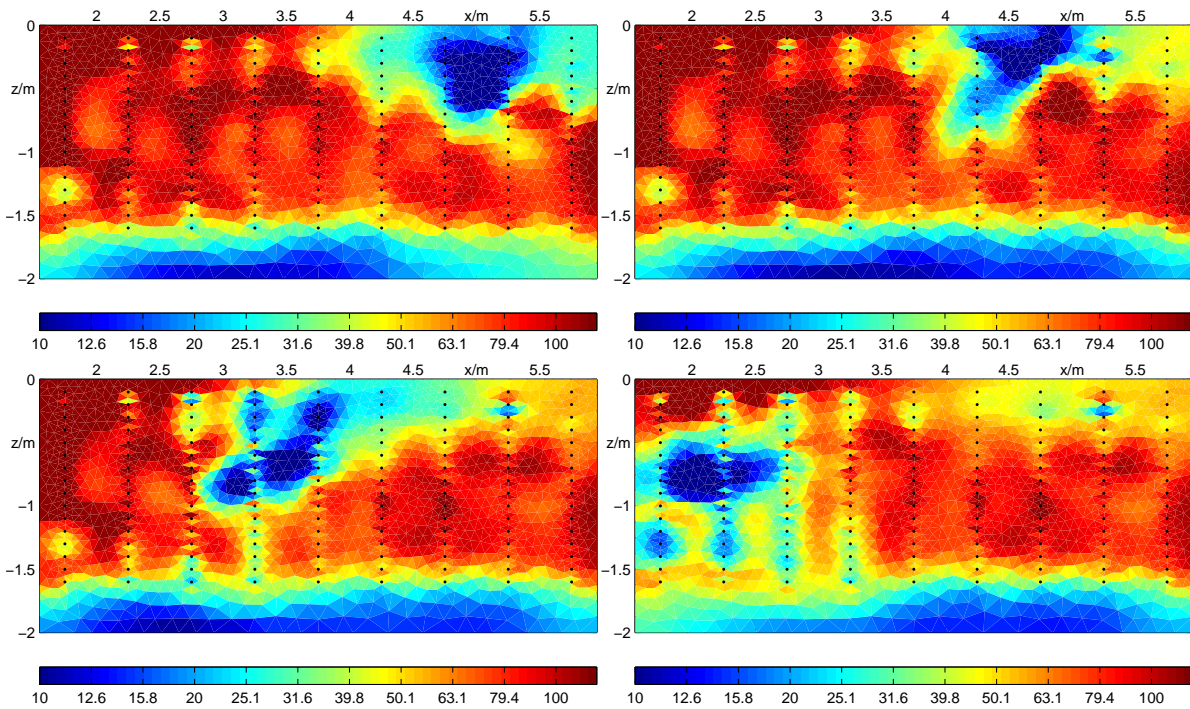


Figure 17: Inversion results 3 hours (upper left), 7 hours (upper right), 12 hours (lower left) and 16.5 hours (lower right) after tracer injection.

6.2. Soil column measurements

We go back to the soil column example from section 4.2. After irrigating a certain amount of water, every 2 hours a complete data set was measured and included in the `TIMESERIES` file. Since the changes are relatively low, we take a look at the relative differences in the files `diff_i.vtk` with respect to the initial resistivity. Figure 18 shows 5 selected time steps. We can see the water front intruding but at a certain stage the column is drying out again.

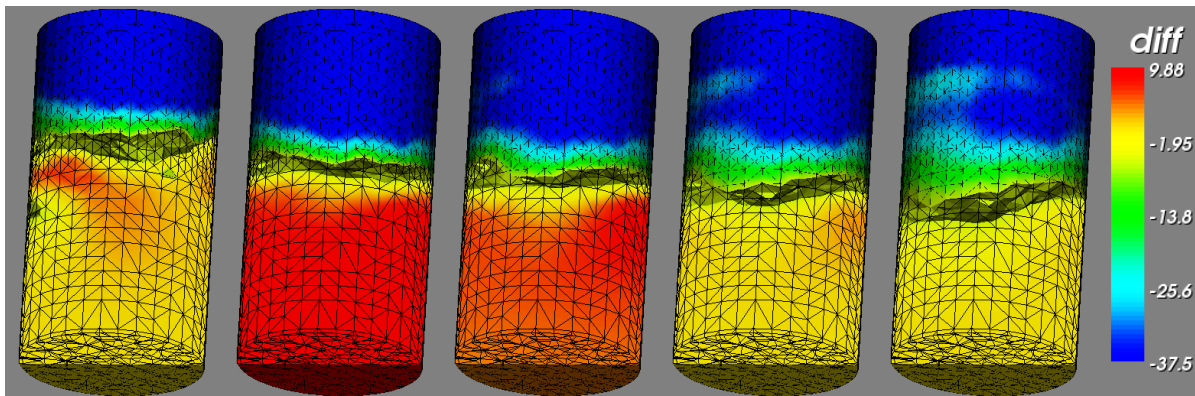


Figure 18: Relative resistivity difference (in %) for the repeated measurements at about 2, 4, 6, 10 and 16 hours after irrigation.

7. Concluding remarks and outlook

We hope to have demonstrated that our software can be used for ERT inversion on very different geometries using different kinds of data. BERT is not a black-box dummy tool and probably it will never be. Thus a lot of experience is required to obtain excellent results. However, for special cases it is worth to branch out more user-friendly tools with immediate visualization and data analysis. We started this by TreeBERT, CylBERT and WallBERT. There are some disadvantages in the use of BERT that are due to the historic development. BERT as part of DCFEMLib will mainly stay the way it is now, only bugs are fixed and minor issues are addressed. However, the technique BERT is included among other methods in the library GIMLi for geophysical inversion and modelling. The new and independent multi-physics library has some advances over DCFEMLib

- A much more flexible style of programming using template techniques.
- Connection of different data by classical or structural joint inversion.
- Use of arbitrary transform functions.
- More rigorous techniques for time-lapse inversion.
- Application of a region technique for sophisticated constraints.

See also appendix D for BERT within the scope of GIMLi.

Acknowledgements

We like to thank all the guys that provided the very instructive data: Folker Donner (formerly University of Mining and Technology, Freiberg), Markus Furche and Ulla Noell (BGR Hannover), Thomas Schicht (K-UTec GmbH Sondershausen), Niels Hoffmann (formerly HAWK Göttingen), Oliver Kuras (British Geological Survey), Joseph Doetsch and Ilaria Coscia (ETH Zurich). Furthermore we acknowledge all the users and testers of BERT that made the software what it is now, a powerful expert tool.

References

- Günther, T. (2002-2007). *DC2dInvRes - Direct Current 2d Inversion and Resolution*. resistivity.net productions, <http://dc2dinvres.resistivity.net>.
- Günther, T. (2003-2008). *DC3dInvRes - Direct Current 3d Inversion and Resolution*. resistivity.net productions, <http://dc3dinvres.resistivity.net>.
- Günther, T. and Rücker, C. (2006). A general approach for introducing structural information - from constraints to joint inversion. In *Ext. Abstract, EAGE Near Surface Geophysics Workshop. 3.-6.9.06, Helsinki(Finland)*.
- Günther, T., Rücker, C., and Spitzer, K. (2006). 3-d modeling and inversion of dc resistivity data incorporating topography - part II: Inversion. *Geophys. J. Int.*, 166(2):506–517.
- Kuras, O., Pritchard, J., Meldrum, P. I., Chambers, J. E., Wilkinson, P. B., Ogilvy, R. D., and Wealthall, G. P. (2009). Monitoring hydraulic processes with automated time-lapse electrical resistivity tomography (ALERT). *Compte Rendus Geosciences - Special issue on Hydrogeophysics*, 341(10-11):868–885.
- Rücker, C., Günther, T., and Spitzer, K. (2006). 3-d modeling and inversion of dc resistivity data incorporating topography - part I: Modeling. *Geophys. J. Int.*, 166(2):495–505.
- Shewchuk, J. R. (1996). Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In Lin, M. C. and Manocha, D., editors, *Applied Computational Geometry: Towards Geometric Engineering*, volume 1148 of *Lecture Notes in Computer Science*, pages 203–222. Springer-Verlag. From the First ACM Workshop on Applied Computational Geometry.
- Si, H. (2002-2008). *TetGen - a quality-constrained tetrahedral mesh generator*. Weierstrass institute, Berlin, <http://tetgen.berlios.de>.

A. BERT for Windows users

BERT is successfully applied on Windows platforms, however it can be slower for bigger problems. Most easily it is distributed under the name BERT4WIN-Suite comprising the binaries from `dfemlib`. Additionally it holds `demod` and `dcinv` and the `gimli` library (see Appendix D). Since BERT is controlled on the command line, Windows users need a command environment as well such as the widely known Cygwin. We recommend MSYS (minimal system), a very small but nice bash-like command shell. Download the current MinGW installer (`mingw-get*.exe`) from <http://www.mingw.org> and install it. Install BERT4WIN, assuming under `d:\software\BERT`¹⁵. Then this path must be known inside the shell, either by changing the environment variable `Path` under *System Control - System - Environment Variables* or in the shell by typing

```
$ export PATH=$PATH:/d/software/BERT
```

The latter can also be done automatically at startup by insertion into a `.bashrc` file in the home directory.

¹⁵Although it should work from version 1.2.5 on, we recommend to use pathnames without blank for both installation and working files.

Note that file names are in different from Windows, i.e. /c instead of c:, slash instead of backslash and names are case-sensitive. Under Windows Vista and 7, please use the original (English) names as /c/Users instead of the translated ones visible in the explorer. If you prepared your data and configuration file in a directory, you should go there with the shell by i.e.

```
$ cd /c/data/profile1/trial and run the inversion using invert inv.cfg all and so on.
```

B. Files and programs

Created files and their meaning

File types:

- *.poly triangle (2d) or tetgen (3d) PLC format
- *.bms binary mesh (house) format
- *.vtk visual toolkit mesh or poly format (paraview)
- *.mesh MEdit mesh format
- *.vector ascii vector of floats
- *.collect potential matrix of all electrodes

Directories and their content:

mesh	meshes and mesh input (meshPara,meshSec,meshPrim)
primaryPot/primary	primary potentials on primary mesh
primaryPot/interpolated	primary potentials on secondary mesh
sensM	sensitivity columns (smatrix.*) or rows smatrixCol.*
result*	saved result directory with most important files

Files in project or result directory:

command.history	history of commands executed by invert
*.data	filtered data file with apparent resistivities and errors
inv.log	inversion log file
model_iter.final	final resistivity distribution
model_iter.*.vector	model vectors of individual iterations
modelResponse.*.vector	model response vectors for each iteration
modelReciprocity.*.vector	model response reciprocity for each iteration
sensCov.vector	coverage (sum of absolute sensitivities)
mesh/mesh.poly	mesh input PLC
mesh/meshParaDomain	pure parameter mesh (use for visualization)

Program calls used for BERT

Inversion and parameterization:

invert - BERT cfg file interpretation

invertNew2d/2dTopo/2dCirc - CFG file generators for 2d cases (flat,topo,circle)

invertNew3d/3dTopo/2dCyl - CFG file generators for 3d cases (flat,topo,tank)

inversion - actual inversion routine

paradepth - estimate appropriate model depth by 1D sensitivities

dcfem - FEM modelling routine
setingvalue - set potentials at electrodes
collectdata - assemble potentials from dcfem
filterdata - filter data yielding apparent resistivity and error estimate

Vector functions:

createStartModel create starting model vector
va2b convert ascii to binary vector
va2b convert binary to ascii vector
interpolate interpolate quantity from one mesh to another

Mesh creation and alteration:

createParaMesh create parameter mesh from dat file
createSecondaryMesh create secondary mesh out of parameters
createSurface create 3d surface mesh from xyz point list
closeSurface close 3d surface mesh by surrounding boxes
prepareMeshRefinement insert refinemeht points
dctriangle triangle call
meshconvert convert mesh between various import formats

Poly tools - creating PLC objects:

polyCreateWorld makes a world with 2 surface and interior boundary
polyFlatWorld make inversion world (2 regions) around electrodes
polyCreateCube create (unit) cube around origin
polyTranslate translate PLC
polyScale scale PLC
polyRotate rotate PLC
polyMerge merge 2 PLCs into a new one
polyAddVIP add points (e.g. electrodes) to PLC
polyAddProfile add profile of electrodes
polyRefineVIPS refine points by local refinement
polyConvert convert PLC to VTK or STL format
polyScripts.sh various functions

C. Complete list of options and their default values

```
# Global settings
DATAFILE=datafile # defines the project filename (required)
DIMENSION=3      # defines the dimension of the problem (2 for 2d or 3 for 3d)
TOPOGRAPHY=0     # defines if topography is present (0 or 1)
TOPOPOINTS=      # file with additional coordinates for topography (x y z)
TOPOPOLY=        # file with additional polygons for topography (x0 y0 z0)
TIMESTEPS=       # file with names of additional datafiles for timelapse inversion
PARAGEOMETRY=    # defines command for creating the PLC by hand
INTERFACE=       # file with known interface(s) (currently only 2d)
```

```

CYLINDER=0      # defines cylindrical geometry (0 or 1)
ELECTRODENODES=1 # electrodes are represented as nodes (default)
SPACECONFIG=0   # geometric factor (0=half space, 1=full space, 2=mirror source)
UNDERWATER=0    # defines underwater survey, sets SPACECONFIG=2

# Data settings
INPUTOHM=0      # input data is in Ohm
OVERRIDEERROR=0 # overrides given errors with INPUTERRLEVEL/INPUTERRVOLTAGE
INPUTERRLEVEL=3 # sets input error level (in percent) if no error given
INPUTERRVOLTAGE=100e-6 # sets input voltage error (V) if no error given
FILTERVALS=0    # defines filter values (experimental)

# Inversion settings
RHOSTART=0      # resistivity of start model (0 = median of the data)
NOPROLONGATION=0 # disables resistivity prolongation into non-inversion regions
MAXITER=20      # maximum number of iteration steps
SINGVALUE=-1    # potential value at electrodes, for sensitivity (internal)
LAMBDA=20       # regularization parameter
LAMBDAOPT=0     # optimize lambda by using l-curve (0 or 1)
LAMBDADECREASE=1 # decrease lambda with each iteration
CONSTRAINT=1    # order 0, 1, 2 (experimental)
ZPOWER=0        # weight for vertical gradients
BLOCKYMODEL=0   # iteratively reweighted model (L1)
ROBUSTDATA=0    # iteratively reweighted data (L1)
LOWERBOUND=0.0  # lower resistivity bound (logarithmic barrier)
UPPERBOUND=0.0  # upper resistivity bound (0.0 = deactivated)
SENSMATUPDATE=0 # update sensitivity matrix with each iteration step
SENSMATDROPTOL=0 # only for very large problems (internal)
SENSMATMAXMEM=800 # only for very large problems (internal)

# Mesh settings
PARAMAXCELLSIZE=0 # maximum cell size volume (m3) or area (m2) for para mesh
PRIMMAXCELLSIZE=0 # maximum cell size volume (m3) or area (m2) for prim mesh
PARADEPTH=0       # maximum depth of parameter domain in meter (0 = automatic)
PARABOUNDARY=5    # boundary around electrodes in parameter domain (percent)
SPLINEBOUNDARY=0 # use splines for additional surface points
EQUIDISTBOUNDARY=0 # equidistant refined space between electrodes
BOUNDARY=500      # size of boundary area around parameter domain
MESHGEN=tetgen    # 3d mesh generator
TETGENTOLERANCE=1e-12 # tetgen tolerance limit for face checks
TETGENPRESERVEBOUNDARY=0 # prevents tetgen from splitting boundary facets
PARADX=0.0        # refinement for para mesh (values>0.5 is forced to 0.5)
PRIMDX=0.1        # ref. for prim mesh towards mesh center(2d: electrode)
PRIMDX_R=0.0      # ref. for prim mesh towards electr. center (overrides PRIMDX)
PARA2DQUALITY=33.0 # parameter grid (from 20 (bad) to 33.4 (good))
PRIM2DQUALITY=33.4 # primary grid (from 20 (bad) to 33.4 (good))
PARA3DQUALITY=1.5 # parameter grid (from 1.1 (good) to 2 (bad))

```

```

PRIM3DQUALITY=1.2 # primary grid (from 1.1 (good) to 2 (bad))
SURFACEQUALITY=30 # quality of topographical surface grid (20 (bad) - 33.4 (good))
SURFACEMAXTRISIZE=0.0 # maximal triangle area of parametric surface grid
SURFACESMOOTH=0 # improve quality of topographical surface grid
ICDROPTOL=0.0 # if number of nodes 200k drop tolerance is set for ICCG solver
LINSOLVER=1 # linear solver: 1=CHOLMOD, 2=LDL, 3=PCG (very slow), 4=TAUCS
SECMESHREFINE=1 # refinement steps for secondary mesh (use with caution)
SECP2MESH=0 # quadratic shapefunction for secmesh (experimental)
PRIMP2MESH=0 # use primary p2 mesh (use with caution)

```

```
# Directory settings
```

```

MESHBASENAME=mesh # basename for mesh files
DIRMESHS=mesh # directory name for mesh files
DIRPOT=primaryPot # directory name for primary and interpolated potentials
DIRPRIMPOT=potentials # subdirectory name for primary potentials
DIRINTERPOLPOT=interpolated # subdirectory name for interpolated potentials
DIRFEM=femM # directory name for finite element solutions
DIRSENS=sensM # directory name for sensitivity matrix
OLDPRIMMESHSTYLE=0# for internal use only

```

D. BERT version 2 using GIMLi

DCFEMLib as a library for dc resistivity modelling and inversion is not extended anymore. Only a few improvements and bug fixes will keep it updated as standard tool. However, the BERT technology is further developed within the open source library GIMLi - Geophysical Inversion and Modelling Library. It represents a completely new-written multi-method C++ class library for inversion of very different data. So BERT as a technology consisting of mesh generation, primary potential calculation and finally inversion, still persists. GIMLi::BERT has some advantages over DCFEMLib::BERT:

- mix of electrode nodes, node-free or finite electrodes
- faster runtime and easier build procedure
- more rigorous minimization approach, mainly time-lapse
- better support for induced polarization data
- improved options such as arbitrary transformations
- region technique for sophisticated regularization
- options for data filtering and restructuring
- python bindings for in-place data and model visualization
- solutions for specific users

There are no mesh generation procedures in GIMLi, so before v.2.1 we will still use the `invert` poly and mesh functions. Main difference is to replace `invert` by `bert` and analogously

bertNew* etc. The changes are documented in the migration guide that can be retrieved from the authors as well as the current (beta) version.