# Documentation for 2.5D electrical resistivity modeling

D. Roubinet

January 29, 2019

**Please cite the use of this software as: Caballero-Sanz, V., D. Roubinet, S. Demirel, and J. Irving (2017), 2.5D discrete-dual-porosity model for simulating geoelectrical experiments in fractured rock, Geophysical Journal International, 209(2), 1099-1110, doi:10.1093/gji/ggx080**

The folders related to the software ERT_DDP_2.5D are organized as follows:

- *Documentation* contains the documentation of the software
- *Input* contains the input files that are necessary to run the code
- *Output* contains the output files that are generated by the code
- *ElectricalResistivity* contains the C++ source files of the code

## 1 Description of the input files

In the folder *Input*, the text file *SimuType.txt* gives the name of the folder containing the input files that will be read by the code. This folder must be located in the folder *Input*.

**The input files are listed and described in detail below for simulating Electrical Resistivity Tomography experiments. The input files provided with the code correspond to the parameter options written in bold.**

### 1.1 *FileNames.txt*

*FileNames.txt* contains the names of the parameter files that will be used by the code in the following order:

- Name of the file describing the kind of simulation and the corresponding numerical parameters
- Name of the file describing the domain properties
- Name of the file describing the considered fractures
- Name of the file describing the boundary conditions
- Name of the boundary condition file when the singularity-removal method is applied (with the boolean *singularity* set to *true* in *PERFORM.cpp*)
- Name of the file for the wave values
- Name of the file describing the positions of the considered electrodes

## 1.2 *Simu.txt*

This file describes the kind of simulation that the user wants to run (simulation_option) and the corresponding numerical parameters

- simulation_option: DFN_GENERATION or **FOURIER_POTENTIAL_COMPUTATION**

When simulation_option = FOURIER_POTENTIAL_COMPUTATION, the following lines of the file correspond to:

- model_option: EPM (equivalent porous media) or DDP (discrete-dual porosity domains)
- $N_x$: number of blocks used in x-direction to discretize the matrix domain
- $N_y$: number of blocks used in y-direction to discretize the matrix domain

## 1.3 *DomainProperties.txt*

- $L_x$: domain length in x-direction
- $L_y$: domain length in y-direction
- $\sigma_m$: matrix electrical conductivity; if $\sigma_m = -1$, the conductivity distribution is described in the file *porous_cond.txt*)

## 1.4 *DFN.txt*

- DFN_option: PARALLEL, SIERPINSKI,SUGAR_BOX, RANDOM, RANDOM_CONNEC, or **DETERMINISTIC1**

If DFN_option=PARALLEL:

- number of fracture family, and for each family:
  - number of fractures
  - angle of the fractures
  - aperture of the fractures
  - conductivity of the fractures
  - fracture spacing (= $-1$ if the fracture spacing is determined by the number of fractures)

If DFN_option=PARALLEL_BORDER:

- aperture of the fractures
- conductivity of the fractures

If DFN_option=SIERPINSKI:

- number of division ($N_d$)
- level of division ($N_l$)
- seed for random generation
- aperture of the fractures
- conductivity of the fractures

If DFN_option=SUGAR_BOX:

- fracture spacing ($B$)
- aperture of the fractures
- conductivity of the fractures

If DFN_option=RANDOM:

- power law length exponent ($a$)
- percolation parameter ($p$)
- fracture aperture
- fracture conductivity

**If DFN_option=DETERMINISTIC1:**

- number of fractures
- coordinates of the first extremity $x_1$ $y_1$
- coordinates of the second extremity $x_2$ $y_2$
- fracture aperture
- fracture conductivity

### 1.5 *BoundaryConditions.txt* (standard and singularity)

Successively for the left, right, top, and bottom borders:

- Type of conditions (DIRICHLET, NEUMANN or MIXED)
- Value of the boundary condition

### 1.6 *WaveValues.txt*

- Number of wave values
- Values of the variables in the Fourier domain

### 1.7 *ElectrodesPosition.txt*

Each line defines an experiment for which the source point injection is defined with

- Electrode number
- x-position of the electrode
- y-position of the electrode

## 2 Output file

In the folder *Output*, the results will be located in a folder having the same name as the folder containing the input files (this folder must be created by the user).

- For EPM simulations: a list of *Resultsk.txt* files containing the Fourier transform of the electric potential distribution in the discretized matrix for experiment $k$ (i.e., for a unit point-source injection in electrode $k$)
- For DDP simulations: a list of *Fracture_Potentialk.txt* and *Matrix_Potentialk.txt* files containing the Fourier transform of the electric potential distribution in the discretized fractures and matrix, respectively, for experiment $k$ (i.e., for a unit point-source injection in electrode $k$)

# 3 Libraries

The software ERT_DDP_2D requires the installation of the following open-access libraries: CGAL, boost, and umfpack.

# 4 Data post-processing

The electric potential distribution is provided in the Fourier domain for the wave values that are set in the input files. This data can be inverted by using the post-processing code written in Matlab and provided in the folder *Post_Processing*. In the provided example, the data located into the *Output* folder for both EPM and DDP simulations are inverted and the spatial distribution of the relative apparent resistivity is plotted.