

elfe3D

Modelling with the total **e**lectric field approach using **f**inite **e**lements in **3D**

User manual

Paula Rulff
Delft University of Technology (TU Delft)
formerly at Uppsala University
p.rulff@tudelft.nl

April 24, 2025

Contents

| | | |
|----------|--------------------------|----------|
| 1 | About | 2 |
| 1.1 | Theory | 2 |
| 1.2 | Implementation | 3 |
| 2 | Getting Started | 3 |
| 2.1 | Dependencies | 3 |
| 2.2 | Installation | 3 |
| 2.3 | Input files | 5 |
| 2.4 | Example | 7 |
| 3 | Citation | 7 |
| 4 | License | 9 |
| | Bibliography | 9 |

1 About

`elfe3D` is a 3D forward modelling code that simulates electric and magnetic field responses from frequency-domain controlled-source electromagnetic set-ups. It uses tetrahedral meshes and first-order finite-element approximations. `elfe3D` was validated in [7].

To balance problem sizes and solution accuracy, adaptive mesh refinement approaches are implemented. They are based on error estimators that consist of face jumps in the normal current density, face jumps in the tangential magnetic field and residuals and can be combined with amplitude-dependent weights. Global mesh quality improvement (q -refinement) can be applied during the refinement procedure. See [7, 2, 6] for details.

`elfe3D` is designed in modern `Fortran` and uses shared-memory parallelisation with `OpenMP`. The system of equations is solved with a direct solver. Isotropic electric resistivities and magnetic permeabilities are variable model parameters. Extended line or loop sources are modelled along element edges.

A earlier version of the code that `elfe3D` is based on was developed by Paula Rulff with contributions from Laura Maria Buntin and Thomas Kalscheuer at Uppsala University from 2018-2023. The initial code development was financed by the Smart Exploration project (European Union's Horizon 2020 funding, grant agreement No. 775971). This initial code is implemented in the implemented in the inversion software `emilia` [5, 4, 3, 8] available upon request for purely academic purposes from Thomas Kalscheuer.

The present version of `elfe3D` was released in 2024 under the Apache License, Version 2.0. Further developments of `elfe3D` by Paula Rulff, now at Delft University of Technology, are ongoing. Suggestions for improvements are welcome via opening an issue on github or via p.rulff@tudelft.nl.

1.1 Theory

Please have a look at pages 26-30 in [6].

1.2 Implementation

Please have a look at pages 34-40 in [6].

2 Getting Started

2.1 Dependencies

To install `elfe3D` on a Linux system, you need to have the following installed:

- a modern Fortran compiler (2008 standard)
- OpenBLAS
- make
- TetGen
- MUMPS

The mesh generator `TetGen` and the solver `MUMPS` are open source packages. Installation instructions are given below. Note that additional libraries have to be installed during the installation of `MUMPS`. Installation instructions are given below and in the `MUMPS` Makefile.

2.2 Installation

`elfe3D` can be compiled with `gfortran` or `ifort`. The provided Makefile is based on `gfortran` compilation. `gfortran` should be part of your system in Gnu compiler collection (gcc). Also, `OpenBLAS` and `make` packages are required. The latest compilation was performed on Ubuntu 22.04.

The following steps guide you through the `elfe3D` compilation:

- `TetGen`: The open source mesh generator `TetGen` must be installed. It can be downloaded from <https://wias-berlin.de/software/index.jsp?id=TetGen> or directly installed via typing in your terminal:

```
$ sudo apt install tetgen
```

If you would like to run the provided example model, you can create the mesh input files in your `elfe3D/in` folder with

```
$ tetgen -pq1.3kAaen CSEM_input_model.poly
```

- MUMPS is an open source direct solver available at <https://mumps-solver.org>. Read the MUMPS documentation for installation instructions! Link the MUMPS routines in your Makefile.
Short example of MUMPS_5.7.3 compilation:

```
$ tar zxvf MUMPS_5.7.3.tar.gz
$ cd MUMPS_5.7.3
$ cp Make.inc/Makefile.debian.SEQ Makefile.inc
$ sudo apt install libmetis-dev libparmetis-dev
$ sudo apt install libscotch-dev libptscotch-dev
$ sudo apt install libatlas-base-dev openmpi-bin libopenmpi-dev
$ sudo apt install liblapack-dev libscalapack-openmpi-dev
$ make all
```

Copy the following files, MUMPS is your MUMPS folder and elfe3D is your elfe3D folder:

```
$ cp MUMPS/libseq/mpif.h elfe3D/elfe3D/.
$ cp MUMPS/include/zmumps_root.h elfe3D/elfe3D/.
$ cp MUMPS/include/zmumps_struct.h elfe3D/elfe3D/.
```

- Modify your makefile as appropriate. At least, adjust LIBDIR in the elfe3D Makefile:

```
$ LIBDIR = /path/to/your/MUMPS_5.7.3/lib
```

- Compile elfe3D in your elfe3D folder by typing in your terminal:

```
$ make all
```

- elfe3D uses OpenMP parallelisation. If it is allowed to use all available threads by default, the performance might drop due to oversubscription. Ensure that elfe3D uses a number of threads that suits the problem you want to solve by setting:

```
$ OMP_NUM_THREADS= (max. total number of CPUs you have)
```

- Run elfe3D with

```
$ ./elfe3d
```

2.3 Input files

Input files for `elfe3D` are located in
`elfe3D/in`

- `elfe3D_input.txt`: This is the most important input file. It contains model and mesh refinement information as well as specifications for in and output files. `elfe3D_input.txt` must contain the following keywords:
 - `solver`: Followed by an integer. The only option that is currently available is option 2: **MUMPS**.
 - `model_size`: Define the following below keyword `model_size`
 - minimum x,y,z coordinates of model
 - maximum x,y,z coordinates of model
 - `num_freq`: Followed by an integer that specifies the number of frequencies. List the actual frequencies in the lines below.
 - `num_rec`: Followed by an integer that specifies the number of receivers. List the receiver coordinates in the lines below.
 - `output_E_file`: Specify the path and filename of the output file for electric field components behind this keyword. The output file will contain the frequencies and real and imaginary electric field components in the following order: frequency — Ex — Ey — Ez
 - `output_H_file`: Specify the path and filename of the output file for magnetic field components behind this keyword. The output file will contain the frequencies and real and imaginary magnetic field components in the following order: frequency — Hx — Hy — Hz
 - `source_type`: Followed by an integer. Several options are implemented. If you specify the source corners in the `source.txt` file, you can choose either option 6 (segmented line source) or option 7 (segmented loop source). Coordinates of source start and endpoints can also be specified below this keyword.
 - `current_direction`: Followed by an integer that specifies the current direction. Line source: current in positive direction (0), current in negative direction (1). Loop source: clockwise current (0), anticlockwise current (1).

- source_moment: Followed by a number that specifies the source moment $m = Idl$
- PEC_present: Followed by an integer that specifies the presence of a perfect electric conductor (PEC). See [2] for more information. No PEC present (0), PEC present (1).
- num_PEC: Followed by the number of PECs with start and end coordinates below.
- model_file_name: Specify the path and filename of the model input file, without file name extension, but with a . at the end.
- maxRefSteps: Followed by an integer that specifies the maximum number of mesh refinement steps. Set to 0 for forward simulations for several frequencies without mesh refinement.
- maxUnknowns: Followed by an integer that specifies the maximum number of unknowns for the mesh refinement.
- betaRef: threshold for the number of elements to be refined
- accuracyTol: accuracy tolerance < 1
- vtk: Followed by an integer that specifies if `yourmodel.vtk` files should be written during the refinement (1), set to 0 for forward simulations only without refinement.
- errorEst_method: Method for error estimation. residuals (1), residuals and face jumps J (2), residuals and face jumps J and H (3), face jumps J (4), face jumps H (5), face jumps J & H (6)
- refStrategy: Refinement strategy. Constant quality factor (0), maxRefSteps-1 on low-quality mesh, last step high-quality mesh (1), increasing quality factor (2), increasing quality factor on mesh with detailed subsurface anomaly (-T and -d option added) (3)
- `yourmodel.poly`: The required model files can be generated with the mesh generator `tetgen` described in the `tetgen` manual. Region numbers have to be specified in the input model file (`.poly` file), receiver locations must be within small elements and edges must be placed along the source cable locations. No node or edge markers are required. `yourmodel.node`, `yourmodel.edge`, `yourmodel.ele`, `yourmodel.neigh` and `yourmodel.vtk` files are expected as input files.

- **regionparameters.txt**: This file specifies the model parameters within the model regions. The following is an example for a model with three different regions (eleattr): air (1), half space (2) and a conductive anomaly (3) and their resistivities (rho), relative magnetic permeabilities (**mu_r**) and relative electric permittivities (**epsilon_r**). (**epsilon_r**) is currently not used in the forward simulations.

```
# eleattr
3
# eleattr rho mu_r epsilon_r
1 100000000.0 1.0 0.0
2 100.0 1.0 0.0
3 10.0 1.0 0.0
```

- **source.txt**: This file specifies the number and locations of source corner points. For a straight line source, it contains only start and endpoints as e.g.

```
2
-50.0 0 0
50.0 0 0
```

2.4 Example

You find an exemplary 3D CSEM resistivity model (Fig. 1) in
elfe3D/in

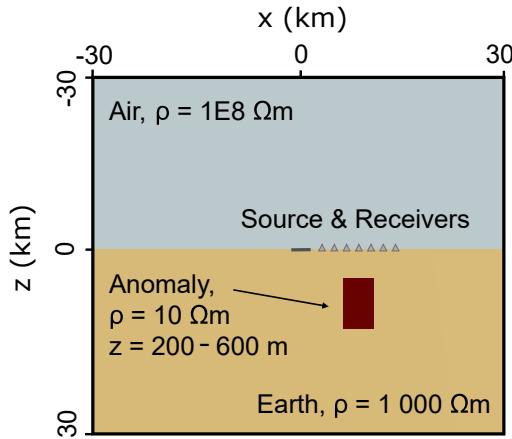
Information about the model are in
elfe3D/in/readme.md

3 Citation

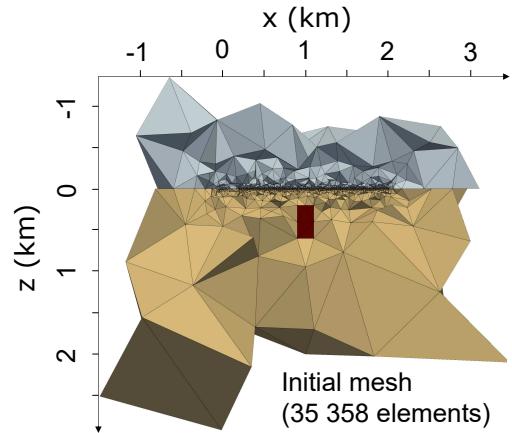
If you publish results generated with **elfe3D**, please give credit to the **elfe3D** developers by citing:

Paula Rulff, Laura M Buntin, Thomas Kalscheuer, Efficient goal-oriented mesh refinement in 3-D finite-element modelling adapted for controlled source

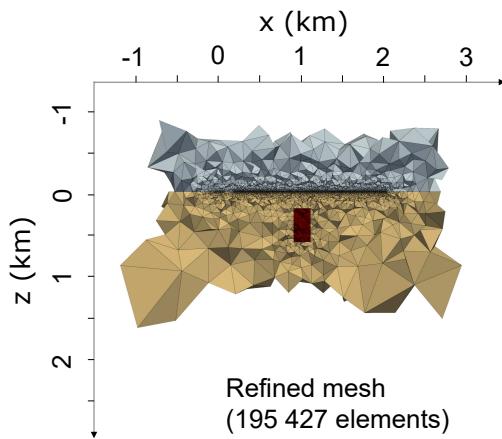
Step I: Model



Step II: Mesh



Step III: Refinement



Step IV: Validation

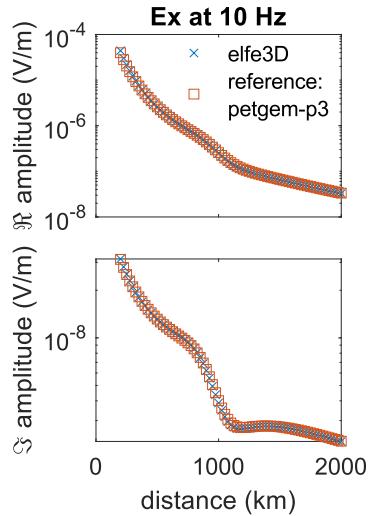


Figure 1: Key steps of the FE forward modelling procedure including the choice of a subsurface model and source-receiver setup (Step I), the initial meshing of the modelling domain (Step II), the mesh refinement (Step III) and the validation against a reference solution (Step IV). The reference solution for this example was computed with PETGEM [1] using third order interpolation functions. Figure and caption from [6].

electromagnetic surveys, Geophysical Journal International, Volume 227, Issue 3, December 2021, Pages 1624–1645, <https://doi.org/10.1093/gji/ggab264>

and refer to the `elfe3D` the version you used using the ZENODO DOI:
Link to Zenodo DOI

Do not forget to acknowledge MUMPS and tetgen developers!

4 License

`elfe3D` is licensed under the Apache License, Version 2.0 (the "License"); you may not use any `elfe3D` files except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Bibliography

- [1] Octavio Castillo-Reyes, Josep de la Puente, and José María Cela. PET-GEM: A parallel code for 3D CSEM forward modeling using edge finite elements. *Computers & Geosciences*, 119:126–136, 2018.
- [2] Octavio Castillo-Reyes, Paula Rulff, Evan Schankee, and Um Adrian. Meshing strategies for 3d geo-electromagnetic modeling. *Computational Geosciences*, 2023.
- [3] Thomas Kalscheuer, Sarah Blake, Joel E. Podgorski, Frederic Wagner, Alan G. Green, Mark Muller, Alan G. Jones, Hansruedi Maurer, Ongkopotse Ntibinyane, and Gomotsang Tshoso. Joint inversions of three

types of electromagnetic data explicitly constrained by seismic observations: results from the central Okavango Delta, Botswana. *Geophysical Journal International*, 202(3):1429–1452, 2015.

- [4] Thomas Kalscheuer, Maria de los Angeles Garcia Juanatey, Naser Meqbel, and Laust B. Pedersen. Non-linear model error and resolution properties from two-dimensional single and joint inversions of direct current resistivity and radiomagnetotelluric data. *Geophysical Journal International*, 182(3):1174–1188, 2010.
- [5] Thomas Kalscheuer, Laust B. Pedersen, and Weerachai Siripunvaraporn. Radiomagnetotelluric two-dimensional forward and inverse modelling accounting for displacement currents. *Geophysical Journal International*, 175(2):486–514, 2008.
- [6] Paula Rulff. *Three-dimensional forward modelling and inversion of controlled-source electromagnetic data using the edge-based finite-element method*. PhD thesis, Uppsala University, 2023.
- [7] Paula Rulff, Laura M Buntin, and Thomas Kalscheuer. Efficient goal-oriented mesh refinement in 3D finite-element modelling adapted for controlled-source electromagnetic surveys. *Geophysical Journal International*, 227:1624–1645, 2021. doi: 10.1093/gji/ggab264.
- [8] Paula Rulff and Thomas Kalscheuer. Research note: A comparison between normalized controlled-source electromagnetic field components and transfer functions as input data for three-dimensional non-linear conjugate gradient inversion. *Geophysical Prospecting*, 2024.