Open-source landscape for 3D CSEM modelling

Dieter Werthmüller[*], *Raphael Rochlitz*[†], *Octavio Castillo-Reyes*[‡], *and Lindsey Heagy*[§]

ABSTRACT

Large-scale modelling of three-dimensional (3D) controlled-source electromagnetic (CSEM) used to be only possible for large companies or research consortia. However, this changed over the last few years, and today there exists a selection of different open-source codes which are up to the task. Four different open-source 3D CSEM codes are considered in this work, where in all cases at least the user-facing functionality is written in Python. Two codes, `SimPEG` and `emg3d`, use finite volumes, the other two codes, `custEM` and `PETGEM`, use finite elements. Three different mesh-types are used, simple tensor meshes, unstructured tetrahedral meshes, and octree meshes. Two models are used to compare the codes, a layered background containing three resistive blocks, and Marlim R3D, a complex, real-world, open-source resistivity data. The comparison shows that all four codes are able to compute the model to sufficient precision. The biggest difference between the codes are actually the used mesh types. The required runtime and memory varies greatly between the codes. However, these requirements depend mainly on the underlying solver, and not on the actual CSEM codes. The collaboration of four code maintainer trying to achieve the same task brought in the end all four codes a significant step further. We hope that these results may be useful for the entire CSEM community, and we invite and encourage the community to make more code, modelling scripts, and results publicly available. For a start, all the scripts to reproduce our shown results are available.

INTRODUCTION

Controlled-source electromagnetic (CSEM) measurements are a frequently applied method in various geophysical exploration fields, such as geothermal, groundwater, oil and gas, mining, civil engineering, or geo-hazards. Modelling these electromagnetic fields is therefore of great interest to design survey layouts, to understand the measured data, and for inversion purposes. Publications regarding 3D modelling in electromagnetic methods started to appear as early as the 1970's and 1980's. These early publications were integral equation (IE) methods, having an anomaly embedded within a layered medium, mostly for loop-loop type transient EM measurements (Raiche, 1974; Hohmann, 1975; Das and Verma, 1982; Newman et al., 1986) and magnetotelluric (MT) measurements (Wannamaker et al., 1984). Ward and Hohmann (1988) assemble many of these solutions in *Electromagnetic Theory for Geophysical Applications*, which is widely viewed as an authoritative publication on electromagnetic geophysics.

In the 1990's, computer became sufficiently powerful that 3D modelling gained traction, which resulted amongst other publications in the book *Three-Dimensional Electromagnetics* by the SEG (Oristaglio and Spies, 1999). Often cited publications from that time are

Mackie et al. (1994), 3D MT computation; Druskin and Knizhnerman (1994), frequency- and time-domain modelling using a Yee grid and a global Krylov subspace approximation; and Alumbaugh et al. (1996); Newman and Alumbaugh (1997), low-to-high frequency computation on massively parallel computers.

The continuous improvement of computing power and the CSEM boom in the early 2000's in the hydrocarbon industry led to a wealth of developed numerical solutions and according publications. The most common applied methods to solve Maxwell's equation are the IE method (Raiche, 1974; Hursán and Zhdanov, 2002; Zhdanov et al., 2006; Tehrani and Slob, 2010; Kruglyakov et al., 2016; Kruglyakov and Bloshanskaya, 2017) and different variations of the differential equation (DE) method, such as finite differences (FD) (Yee, 1966; Wang and Hohmann, 1993; Mackie et al., 1994; Druskin and Knizhnerman, 1994; Streich, 2009; Sommer et al., 2013), finite elements (FE) (Commer and Newman, 2004; Schwarzbach et al., 2011; da Silva et al., 2012; Grayver et al., 2013; Puzyrev et al., 2013; Zhang and Key, 2016), and finite volumes (FV) (Madsen and Ziolkowski, 1990; Haber and Ascher, 2001; Clemens and Weiland, 2001; Jahandari and Farquharson, 2014). There are also many different types of discretisation, where the most common ones are regular grids (Cartesian, rectilinear), mostly using a Yee grid (Yee, 1966) or a Lebedev grid (Lebedev, 1964), but also unstructured tetrahedral grids (Zhang and Key, 2016; Cai et al., 2017), hexagonal meshes (Cai et al., 2014), or OcTree meshes (Haber and Heldmann, 2007). Very well written overviews about the different approaches to 3D EM modelling are given by Avdeev (2005) and Börner (2010).

In the last 15 years, the publications with regards to 3D EM modelling grew tremendously, driven by the ever increasing computing power. One reason why there are so many reports about this topic results from the variety of solution techniques for the systems of linear equations (SLE). They can be distinguished between direct solvers (Streich, 2009; Chung et al., 2014; Jaysaval et al., 2014; Grayver and Kolev, 2015; Oh et al., 2015; Wang et al., 2018), indirect solvers (Mulder, 2006; Jaysaval et al., 2015), or a combination of both, so-called hybrid solvers (Liu et al., 2018). The solvers often use preconditioners such as the multigrid method (Aruliah and Ascher, 2002; Mulder, 2006; Jaysaval et al., 2016).

Many of the advancements made in the EM modelling community over the past decades have required that authors develop new implementation from scratch. These codes often provided the research group or company with a competitive advantage for a time, and thus the source codes were often kept internal. In some cases, executables have been made available for academic purposes upon request or to sponsoring consortium members. As the field continues to mature, advancements become more incremental, and particularly in an applied field, many advancements are driven by new use-cases and applications that were not considered by the original authors. In his review on EM modelling and inversion Avdeev (2005) concludes with the following statement: «*The most important challenge that faces the EM community today is to convince software developers to put their 3-D EM forward and inverse solutions into the public domain, at least after some time. This would have a strong impact on the whole subject and the developers would benefit from feedback regarding the real needs of the end-users.*» Further, Oldenburg et al. (2019) argue that an open-source paradigm has the potential to accelerate multidisciplinary research by encouraging the development of modular, interoperable tools that are supported by a community of researchers.

In many domains of science, it is becoming more common for researchers to release

source-code with an open license that allows use and modification (e.g., see the Open Source Initiative approved licenses: opensource.org/licenses). This is an important step for improving reproducibility of research, «*to provide the means by which the reader can verify the validity of the results and make use of them in further research*» (Broggini et al., 2017). Going a step beyond releasing open-source software, many groups adopt an open model of development where code is hosted and versioned in an online repository, all changes are public, and users can engage by submitting and track issues. Community oriented projects further engage by encouraging pull-requests, which are suggested changes to the code. Successful, well-maintained projects often have unit-testing and continuous integration that runs those tests with any changes to the code. Additionally, documentation that includes examples and tutorials is an important component for on-boarding new users and contributors. As a result, in many areas of the geosciences, we are seeing a shift away from a one-way distribution of code towards building global communities around open projects. Related projects within the realm of geophysical exploration are, e.g., `pyGIMLi` (Rücker et al., 2017) or `Fatiando` (Uieda, 2018)

In this paper, we discuss the topic of open-source developments with respect to 3D CSEM modelling. Even though it comprises still the minority of developments, the landscape in this field of research changed in the last five years. Previously, there were only closed-source codes owned by companies or consortia and individual projects, often without proper documentation, that could be requested from the authors. We introduce and compare four recent open-source projects with focus on 3D CSEM modelling. All presented codes are in the Python ecosystem and use either the FV method on structured grids or the FE method on unstructured tetrahedral meshes.

Furthermore, we deal with the topic of verification. Miensopust et al. (2013) presents a review of two workshops dealing with the verification of magnetotelluric forward and inversion codes, but we are not aware of any comparable study or benchmark suite for CSEM data. Hence, we first conduct a similar exercise for CSEM codes. We want our efforts to help to ensure the reliability of simulations, not only carried out with the available open-source codes, but also for individual closed-source projects. Analytical and semi-analytical solutions only exist for simple halfspace or layered-Earth models, which served mostly to verify new codes. Beyond these simple models, the only objective possibility to ensure the accuracy of solutions is by comparing results from different modellers. If different discretisations and implementations of Maxwell's equations yield the same result, it gives confidence in their accuracy.

We simulate EM fields for a layered background model with vertical transverse isotropy, containing three blocks, as well as for the complex marine Marlim R3D model. These two model designs as well as the corresponding results from four different codes provide a benchmark for new codes to be compared to and validated with. In the following section, we introduce the codes under consideration. Afterwards, we present the considered benchmark cases in detail and present the modelling results of our four codes in terms of accuracy and computational performance. Beyond that, we extensively discuss important points that control the performance and suitability of our FV and FE codes, including mainly considerations about the mesh design and the choice of solvers. We conclude with a summary of our results and a motivation for the EM community to not only continue to extend the landscape of open-source codes but to also create a landscape of open-source benchmark models.

Introduce Maxwell's Equations or not? Miensopust et al., 2013, don't.

CODES

[RR]: I think we should introduce the term degrees of freedom (dof), equivalent to the size of the SLE, as general measure of our problem sizes. If it is somewhat x*Nedges, or related to Ncells or Nnodes, who cares? We just need this one number for this paper in my opinion, more details in the jupyter notebooks - fine.

The four codes under consideration are, in alphabetical order, `custEM` (Rochlitz et al., 2019), `emg3d` (Werthmüller et al., 2019), `PETGEM` (Castillo-Reyes et al., 2018, 2019), and `SimPEG` (Cockett et al., 2015; Heagy et al., 2017). All four codes have their user-facing routines written in Python; all of them make heavy use of `NumPy` (van der Walt et al., 2011) and `SciPy` (Virtanen et al., 2020). The four of them are "modern" open-source projects, meaning that they come with both, an open-source license and an online-hosted version-control system with tracking possibilities (raising issues, filing pull requests). All developments comprise an extensive online documentation with many examples and have continuous integration to some degree. Newer package-management systems such as Conda, Docker, or pip significantly supported us to make our codes easily available and installable without any user-side compilations. Each one can be downloaded and installed with a single command.

In the following, we provide a quick overview of the codes. It is, however, beyond the scope of this article to go into every detail of the different modellers, and we refer to their documentations for more details. An overview comparison of the different codes is given in Table 1. All codes have in common that they solve the weak formulation of Maxwell's equation in its differential form (DE) under the quasistatic or diffusive approximation, hence neglecting displacement currents. The machines on which the different codes were run are listed in Table 2 together with the responsible operator.

	`custEM`	`emg3d`	`PETGEM`	`SimPEG`
Home	custem.rtfd.io	empymod.github.io	petgem.bsc.es	simpeg.xyz
License	GPL-3.0	Apache-2.0	GPL-3.0	MIT
Installation	`conda`	`pip`; `conda`	`pip`	`pip`; `conda`
Comp. Dom.	frequency & time	frequency	frequency	frequency & time
Method	FE	FV	FE	FV
Mesh	tetrahedral	rectilinear	tetrahedral	recti- & curvilinear
BC	PMC; PEC	PEC	PEC	PEC; PMC
Solver	`MUMPS`	`emg3d`	`PETSc`; `MUMPS`	`PARDISO`; `MUMPS`

Table 1: Comparison of the four codes under consideration. Note that `emg3d` is a solver on its own, while the other codes implement third-package solvers such as `PETSc` (Abhyankar et al., 2018), `MUMPS` (Amestoy et al., 2001), or `PARDISO` (Schenk and Gärtner, 2004).

Code	Computer and Operating System	Operator
`custEM`	**PowerEdge R940 server**; 144 Xeon Gold 6154 CPU @2.666 GHz; 3 TB DDR4 RAM; Ubuntu 18.04	Raphael Rochlitz
`emg3d`	**Laptop**; i7-6600U CPU@2.6 GHz x4; 16 GB of memory, Ubuntu 18.04	Dieter Werthmüller
`PETGEM`	**Marenostrum4 (supercomputer)**. Intel Xeon Platinum from Skylake generation; 2 sockets Intel Xeon Platinum 8160 CPU with 24 cores each @2.10GHz for a total of 48 cores per node; 386 Gb DDR4 RAM per node; SuSE Linux Enterprise	Octavio Castillo-Reyes
`SimPEG`	???	Lindsey Heagy

Table 2: List of computer and operating system (hardware and software) on which the different codes were run, together with the operator.

custEM

The customisable electromagnetic modelling Python toolbox custEM was developed for simulating arbitrary complex 3D CSEM geometries with focus on semi-airborne setups, but it supports also land-based, airborne, coastal or marine environments. Multiple electric or magnetic field or potential finite-element approaches were implemented as total or secondary field formulations. The finite-element kernel, including higher order basis functions and parallelisation, relies on the FEniCS project (Logg et al., 2012; Langtangen et al., 2016). The resulting systems of linear equations are solved with MUMPS (Amestoy et al., 2001), which is a very robust but memory consuming choice. Primary field solutions are supplied by the COMET (Skibbe et al., 2020) package.

The toolbox considers generally anisotropic petrophysical properties. Even though changes of the conductivity is mainly of interest for CSEM modelling, the electric permittivity and magnetic permeability can be taken into account using the preferred electric field approach on Nédélec elements. Recently, induced polarization parameters in frequency-domain and three methods for simulating time-domain responses were added to custEM. The provided meshing tools based on TetGen (Si, 2015) and functionalities of pyGIMLi (Rücker et al., 2017) facilitate the generation of tetrahedral meshes including layered-earth geometries with topography or bathymetry and anomalous bodies which are allowed to be connected or to reach the surface.

The custEM toolbox is under steady development and maintenance. Future enhancements comprise adding iterative solution techniques, provided by FEniCS, as well as further functionalities for automatically incorporating intersecting bodies such as folds or faults in meshes. Furthermore, current work is on using custEM for inverse modelling applications.

emg3d

The 3D CSEM modeller `emg3d` is a multigrid (Fedorenko, 1964) solver for 3D electromagnetic diffusion following Mulder (2006), with tri-axial electrical anisotropy, isotropic electric permittivity, and isotropic magnetic permeability. The matrix-free solver can be used as main solver or as preconditioner for one of the Krylov subspace methods implemented in SciPy, and the governing equations are discretized on a staggered grid by the finite-

integration technique (Weiland, 1977), which is a finite-volume generalization of a Yee grid. The code is written completely in Python using the NumPy and SciPy stacks, where the most time- and memory-consuming parts are sped up through jitted (just-in-time compiled) functions using Numba (Lam et al., 2015). The code computes the electric field due to an electric source in the frequency-domain, which is its main application (frequency-domain CSEM). However, there are also routines to obtain the magnetic field due to an electric source, to obtain the electric and magnetic fields due to a magnetic source, and to obtain time-domain responses.

The multigrid method is characterized by almost linear scaling both in terms of runtime (CPU) and memory (RAM) usage, and it is therefore a comparably low-memory consumption solver. It is also minimal in terms of requirements, only NumPy, SciPy, and Numba are required, with Python 3.7+. The current development is focused on adding basic inversion capabilities, and a further plan is to implement a hook in order to use `emg3d` as a frequency-domain CSEM solver within the grander `SimPEG` framework.

PETGEM

`PETGEM` is a parallel code for frequency-domain 3D CSEM data from marine and land surveys. The high-order edge FE method (HEFEM) is used for the discretization of the governing equations in its diffusive form. This technique provides a suitable mechanism to obtain stable numerical solutions as well as a good trade-off between number of dofand computational effort. The current implementation supports up to six-order tetrahedral vector basis functions. Moreover, because the HEFEM belongs to the FE family, the unstructured meshes can be used efficiently on complex geometries (e.g., models with topography and bathymetry).

[OCR]: I assume we have previously mentioned the term dofs

`PETGEM` permits to locate the source and receivers anywhere in the computational domain(e.g., sediments, seafloor, sea, ground), allowing to analyze the physical environment of the electric responses and how parameters impact them (e.g., frequency, conductivity, dependence on mesh setup, basis order, solver type, among others). Furthermore, `PETGEM` implements a semi-adaptive mesh strategy (*hp* mesh refinement) based on physical parameters and on polynomial order to satisfy quality criteria chosen by the user. Nonetheless, only Horizontal Electric Dipole (HED) has been implemented.

[DW]: I think this applies to all four codes

For the parallel forward modelling computations, a highly scalable MPI domain decomposition allows reducing run-times and the solution of large-scale modelling cases. This strategy is capable of exploiting the parallelism offered by both modest multi-core computers and cutting-edge clusters (e.g., High-performance Computing architectures).

Based on the current state of the `PETGEM` project, there are many possibilities to improve or to add additional features. For instance, in the short-term, more modelling routines and support for multi-source set-ups will be implemented. For the long-term, the obvious next step is to move to 3D CSEM data inversion.

SimPEG

`SimPEG` is a modular toolbox for simulations and gradient based inversions in geophysics. Current functionality includes modelling and inversion capabilities for gravity, magnetics, direct current resistivity, induced polarization, electromagnetics (time, frequency, as well as natural sources, e.g. magnetotellurics), and fluid flow (Richards equation). It is a community driven project that aims to support researchers by providing a flexible, extensible framework and to facilitate integration of data sets from a variety of geophysical methods, including joint inversions (Astic et al., 2020), by providing a common interface to each method.

Meshes and finite volume differential operators are implemented in the `discretize` package. It currently includes tensor, OcTree, cylindrical, and logically rectangular meshes. Each mesh type inherits from a common structure and uses the same naming conventions for methods and properties. This allows us to decouple the implementation of a discretized partial differential equation from the details of the mesh geometry and therefore we can write a single implementation of discretized partial differential equation (or set of partial differential equations) in `SimPEG` that will support all mesh types implemented in `discretize`.

The electromagnetics implementations use a staggered grid finite volume approach where physical properties are discretized at cell centers, fields on cell edges and fluxes on cell faces (Yee, 1966) (for an overview of discretizing Maxwell's equations, we recommend Haber (2014)). There are two different discretization strategies implemented for Maxwell's equations: (a) the EB-formulation, which discretizes the electric field ($\vec{e}$) on cell edges and the magnetic flux density ($\vec{b}$) on cell faces, and (b) the HJ-formulation, which discretizes the magnetic field ($\vec{h}$) on cell edges and the current density ($\vec{j}$) on cell faces. In both cases the physical properties: electrical conductivity / resistivity and magnetic permeability are discretized at cell centers. Having multiple implementations allows for testing that compares results from each approach, as well as the representation of both electrical and magnetic sources on cylindrically symmetric meshes. `SimPEG` supports variable magnetic permeability and full-tensor anisotropy for the physical properties.

`SimPEG` interfaces to various solvers including `PARDISO` (Schenk and Gärtner, 2004), `MUMPS` (Amestoy et al., 2001), as well as those implemented in the `scipy.sparse.linalg` module (Virtanen et al., 2020). For problems with many sources (and reciprocity is not a natural solution, e.g. in 3D airborne EM), `SimPEG` contains machinery for domain-decomposition approaches; see for example Fournier et al. (2020).

NUMERICAL VERIFICATIONS

We computed the responses for two different models to verify that all different 3D codes yield the same electromagnetic responses. The first simple *Block model* consists of a layered, anisotropic background with three embedded blocks. The second Marlim R3D model is based on a very complex, realistic marine CSEM setup.

Block Model

The block model is a derivation of the *Dublin Test Model 1* from the first EM modelling workshop described by Miensopust et al. (2013). We took the same layout of the blocks but adjusted the dimensions and resistivities to a typical marine CSEM problem, as shown in Figure 1. We additionally added a layered background with vertical transverse isotropy (VTI).

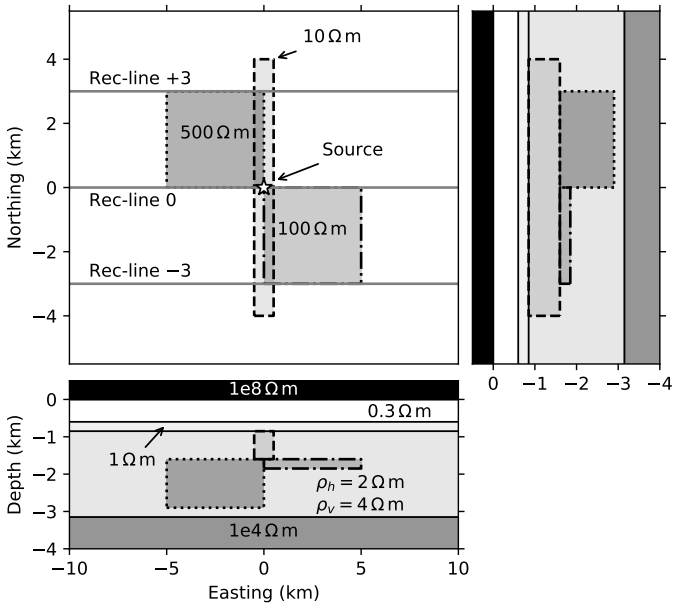

Figure 1: Sketch of the block model. The layered model consists of an air layer, a water layer, a thin top-layer followed by a thick, anisotropic background layer, and at the bottom a resistive basement layer. All three blocks are embedded in the thick background layer, which has VTI with $\lambda = \sqrt{2}$.

The layered model consists of an upper halfspace of air, a 600 m deep water layer, followed by a 150 m thick, isotropic layer of 1 Ω m, a 3.3 km thick, anisotropic layer of $\rho_\mathrm{h} = 2\,\Omega\,\mathrm{m}$ and $\rho_\mathrm{v} = 4\,\Omega\,\mathrm{m}$, and finally a resistive basement consisting of a lower halfspace of 1000 Ω m. The 200 m long, x-directed source is located 50 m above the seafloor from $x = -50$ to $x = 50$ m in x-direction, at $y = 0$ m. The x-directed receivers are placed on the seafloor every 200 m from $x = -10$ km to $x = +10$ km in three lines for $y = -3, 0, 3$ km.

The thin isotropic layer beneath the water requires many tetrahedra for a good-quality discretisation. Therefore, the FE codes `custEM` and `PETGEM` used the same tetrahedral mesh with a horizontal layered-earth extent of 30 km from the origin, surrounded by a ten times larger halfspace-like boundary mesh with water conductivities assigned to the subsurface. This approximation significantly reduced the problem size compared to considering the layered-earth geometry for the whole computational domain.

[**RR**]: Add short information about mesh design with SIMPEG and emg3d, including something like "comparatively coarse discretization in center to minimize computation times"

In a first step, we compare the layered background to the semi-analytical solutions of a

1D code, for which we use `empymod` (Werthmüller, 2017). Figures 2 and 3 show the actual responses in the top rows, and the relative error in the bottom rows for the receiver lines $y = 0\,\text{km}$ and $y = \pm 3\,\text{km}$, which are identical.

Figure 2 illustrates the inline responses in the top-row for offsets $r > 0.5\,\text{km}$. We only show the semi-analytical result for the real and imaginary amplitudes. The bottom row displays the relative error of the 3D codes. The results show that all codes are yielding accurate result with errors of around 1 % in most parts and overall below a few percent, disregarding locations affected by sign reversals. The differences can be mainly attributed to the chosen discretisations. The FE codes show the strongest misfits at large offsets, most likely caused by the utilized boundary mesh. Very low errors at large offsets are achieved by `emg3d`, whereas it suffers at short offsets from a coarse grid.

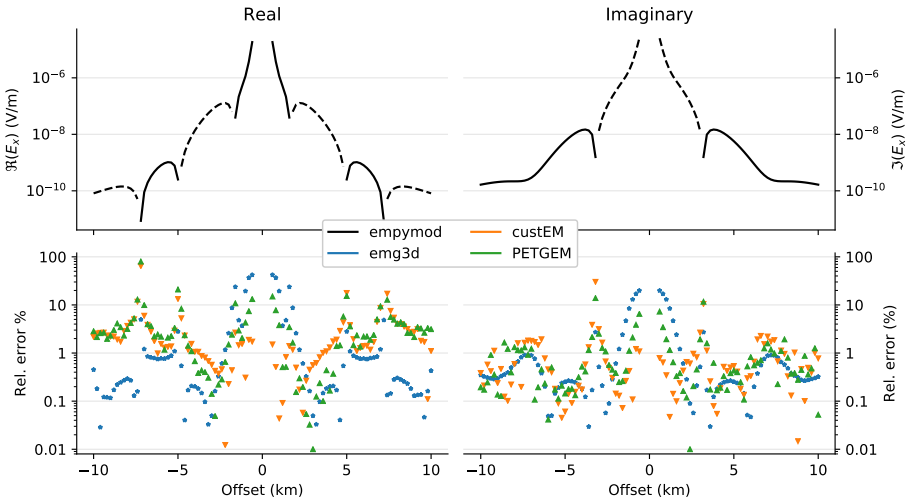

Figure 2: Comparison of the layered model for the inline receivers ($y = 0\,\text{km}$). Top row shows the semi-analytical responses from `empymod`, and bottom row shows the relative error (%) of `custEM`, `emg3d`, `PETGEM`, and `SimPEG`.

[RR]: I suggest to show y-component for broadside and the errors of the weak component (if not too bad), otherwise only broadside is required to save a plot, because inline should be better for Ex in general.

The corresponding broadside responses for $y = \pm 3\,\text{km}$ are shown in Figure 3. The CPU and RAM requirements are listed in Table 3.

In a second step, three resistive blocks were added in the 3.3 km-thick, anisotropic background layer as shown in Figure 1. They have resistivities of $\rho = 10\,\Omega\,\text{m}$ (shallow beam perpendicular to survey lines), $\rho = 100\,\Omega\,\text{m}$ (thin plate, South-East), and $\rho = 500\,\Omega\,\text{m}$ (cube, North-West). As there are no analytical solutions for this model we show the normalized difference between different codes, instead of the relative error, where the normalized difference of two responses p and q is given by

$$\text{NRMSD}\ (\%) = 200\frac{|p-q|}{|p|+|q|}\ . \tag{1}$$

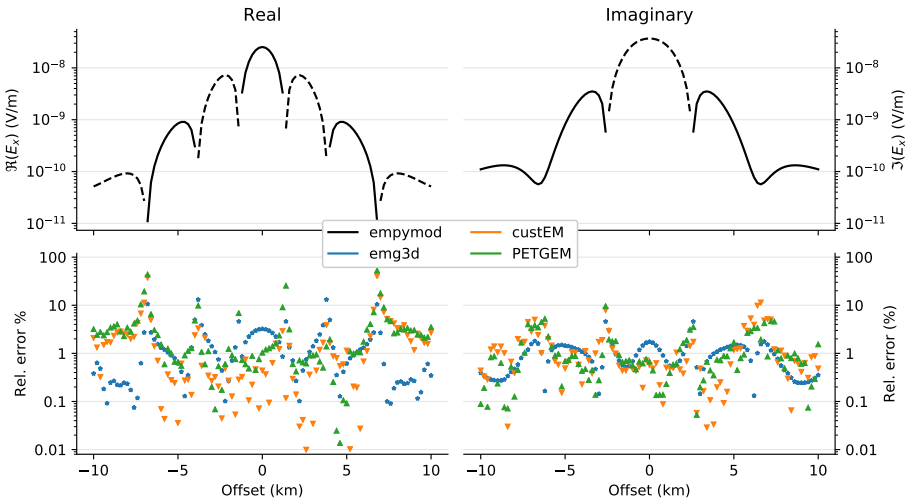

Figure 3: Comparison of the layered model for the broadside receivers ($y = -3\,\text{km}$). Top row shows the semi-analytical responses from `empymod`, and bottom row show the relative error (%) of `custEM`, `emg3d`, `PETGEM`, and `SimPEG`.

Code	CPU (s)	#Procs	RAM (GiB)	#dof
`custEM`	0.0	0	0.0	0
`emg3d`	0.0	0	0.0	0
`PETGEM`	0.0	0	0.0	0
`SimPEG`	0.0	0	0.0	0

Table 3: Comparison of used CPU and RAM and of number of cells used to discretize the layered model.

[DW]: NRMSD: Comparing all codes to all would make the figures too cluttered, and I don't think it would add a lot insights. My idea is therefore to compare `emg3d`/`SimPEG` (the two FV ones), `custEM`/`PETGEM` (the two FE ones), and `emg3d`/`custEM` (a cross-comparison). What do you think?

[RR]: Agreed. Another suggestion, I solved this visualization problem with a color coded matrix plot, showing 90th percentiles or median values of all errors along the line as meaningful comprehensive number. Look in my thesis - example 4. Comparison for real and imag could be in the lower and upper triangles, or we show absolute magnitudes for layered earth in lower triangle and for block model in upper triangle. With help of this plot, one could easily say if differences are higher for block model and which codes fit to each other well.

[RR]: If using the mentioned matrix plots, we would need only one of the figures 4-6, and could show the other data comprehensively. I added a python script to generate an example of my plotting suggestion in the model-block directory and a new Figure 7 in the manuscript that illustrates what I mean. Of course, any upgrade suggestion for this initial idea is welcome, aside from making it looking nice with labels and annotations.

[OCR]: I really liked this way of presenting the comparison, in addition to saving space, we gained in understanding. Looks really nice!

The results for the three receiver lines $y = -3, 0, 3\,\text{km}$ are shown in Figures 4, 5, and 6, respectively, and the corresponding CPU and RAM in Table 4. We can conclude from the NRMSD that similar codes have a smaller NRMSD than cross-comparison, hence `emg3d` and `SimPEG` are very similar, as well as are `custEM` and `PETGEM`.

[DW]: Write in detail when all results are available

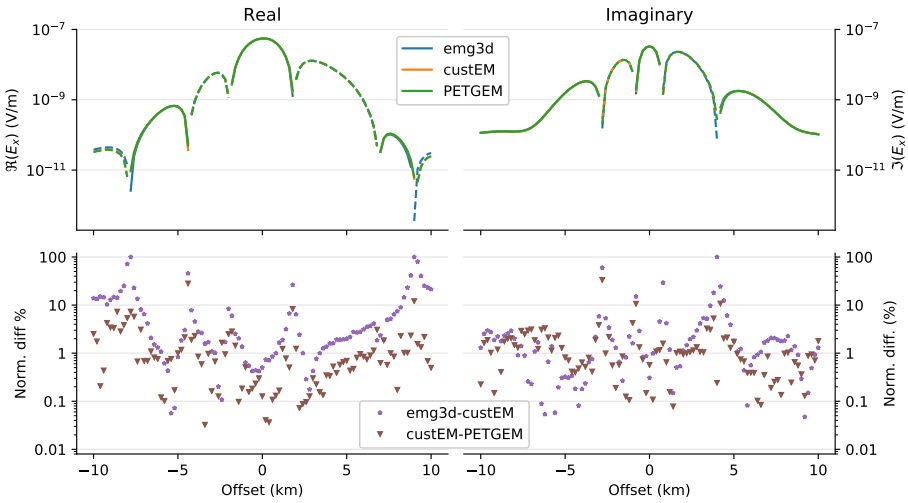

Figure 4: Comparison of the block model for the broadside receivers at $y = -3\,\text{km}$. Top row shows the responses and bottom row the NRMSDs (%) between `emg3d`–`custEM`, `custEM`–`PETGEM`, and `emg3d`–`SimPEG`.

The difference becomes only significant at offsets larger than 8 km. This is due to a practical difference in the meshing: This simple block model is advantageous for FV codes, as it consists of rectangular blocks and horizontal layers, simple geometric objects. While having a very thin, shallow layer is straight-forward for FV codes, it requires many cells for a FE code, and a similar approach as for the Block Model was used. This is the reason for the increased NRMSD at large offsets, and it has nothing to do with the accuracy of the computation itself.

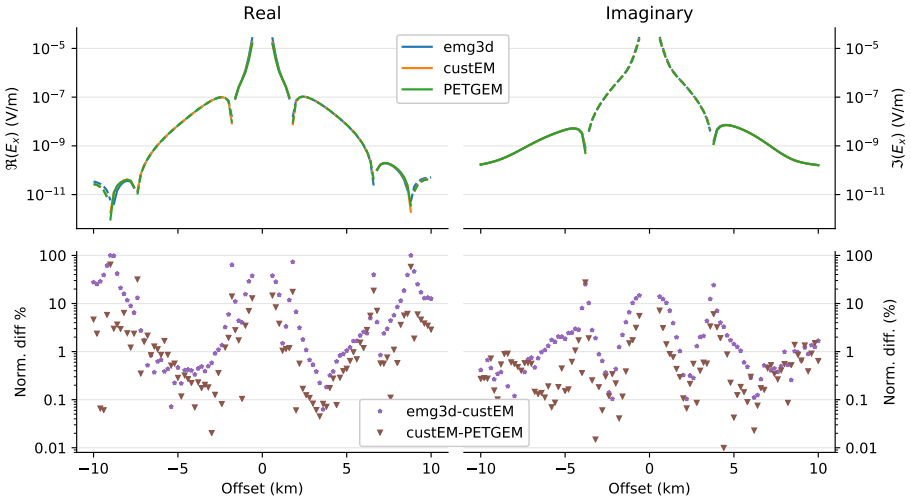

Figure 5: Comparison of the block model for the inline receivers at $y = 0$ km. Top row shows the responses and bottom row the NRMSDs (%) between `emg3d`–`custEM`, `custEM`–`PETGEM`, and `emg3d`–`SimPEG`.

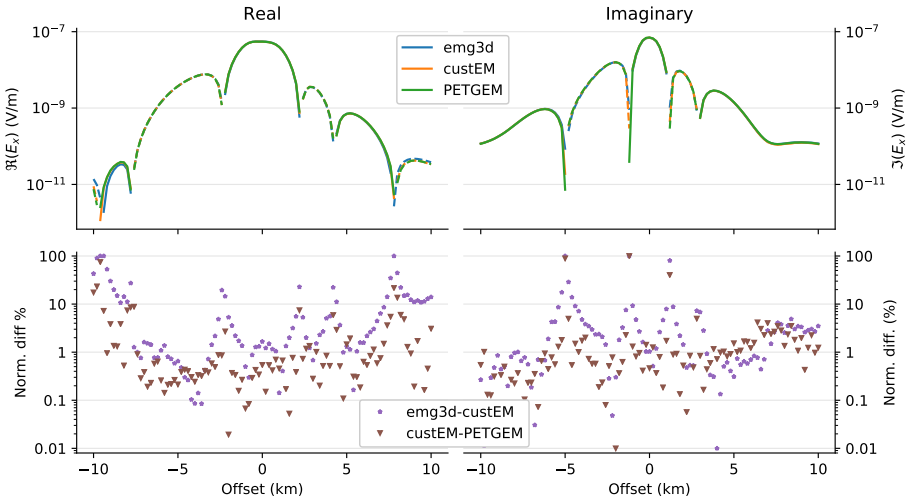

Figure 6: Comparison of the block model for the broadside receivers at $y = +3$ km. Top row shows the responses and bottom row the NRMSDs (%) between `emg3d`–`custEM`, `custEM`–`PETGEM`, and `emg3d`–`SimPEG`.

Code	CPU (s)	#Procs	RAM (GiB)	#dof
`custEM`	0.0	24	0.0	0
`emg3d`	0.0	1	0.0	0
`PETGEM`	0.0	48	0.0	0
`SimPEG`	0.0		0.0	0

Table 4: Comparison of used CPU and RAM and of number of cells used to discretize the block model.

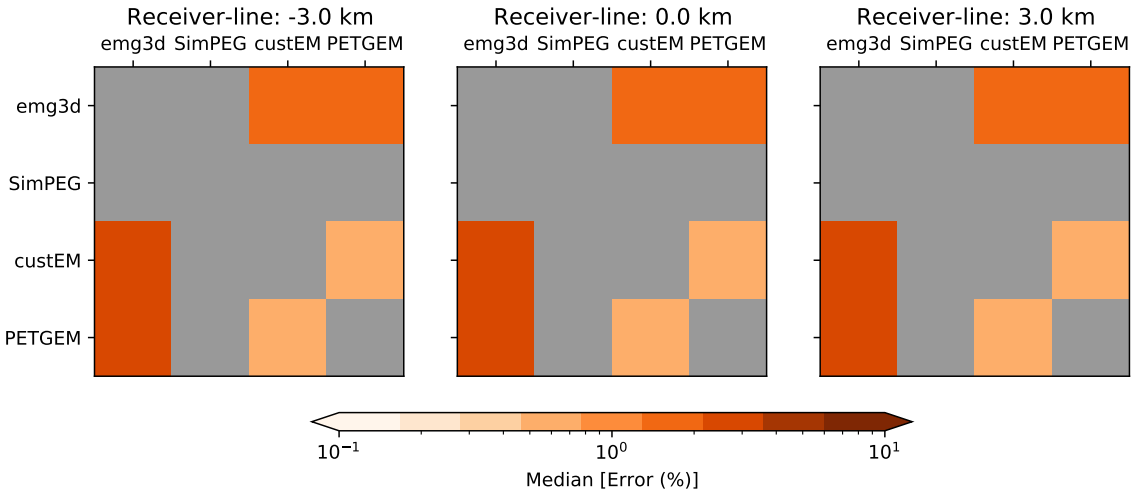

Figure 7:

[DW]: Suggestion from Raphael for a statistics plot.

Overall misfit relations between all codes for the three receiver lines y=-3, y=0, and y=3 (f.l.t.r.). Colors indicate the median values of the 101 calculated NRMSD arguments for each line between all codes. Errors for real parts are shown in the lower triangle, for imaginary parts in the upper triangle.

Marlim R3D

The Marlim oil field is a giant reservoir in a turbidite sandstone horizon in the north-eastern part of the Campos Basin, offshore Brazil, which was discovered in 1985. Carvalho and Menezes (2017) created from seismic data and well log data a realistic resistivity model with vertical transverse isotropy (VTI), Marlim R3D (MR3D), which they released under an open-source license. Correa and Menezes (2019) computed CSEM data for six frequencies from 0.125 Hz to 1.25 Hz, released under an open-source license. To compute the data they used a state-of-the-art code from the industry (Maaø, 2007). It is therefore an ideal case to verify our open-source codes against, as it is a complex, realistic model and the data were computed by an industry-proofed code. Additionally, that code is a time-domain code, whereas the four codes under consideration here compute the results all in the frequency-domain.

The detailed reservoir model consists of $1022 \times 371 \times 1229$ cells, totalling to almost 466 million cells, where each cell has dimensions of 25 by 75 by 5 m. For the computation the model is upscaled to 515 by 563 by 310 cells, totalling to almost 90 million cells, where each cell has dimensions of 100 by 100 by 20 m.

The published data set consists of a regular grid of receivers of 20 in eastern direction by 25 in northern direction, 500 receivers in total, with 1 km spacing located on the irregular seafloor. 45 source-towlines were located on the same grid, 50 m above the seafloor, with shots every 100 m. In Correa and Menezes (2019) the actual responses for one receiver with the corresponding East-West inline and one East-West broadside acquisition line are shown, which is the data we chose for our comparison. The cross-section of the resistivity model

along the chosen inline acquisition line is shown in Figure 8, together with the acquisition information.

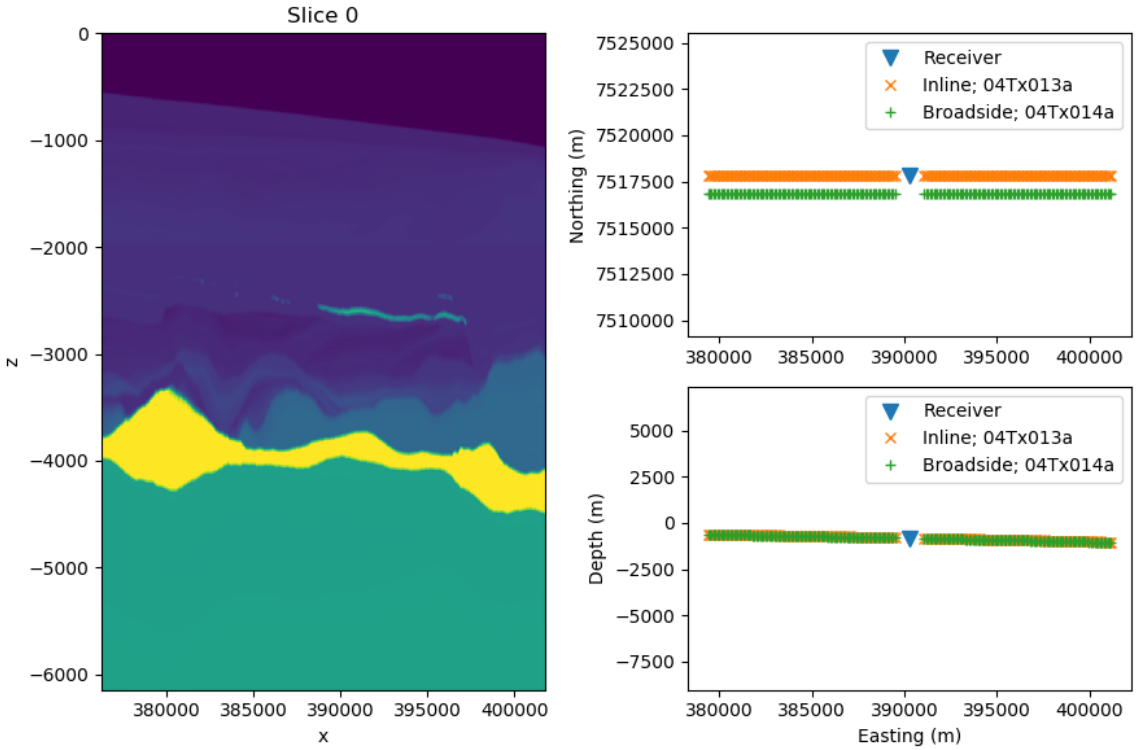

Figure 8: This figure obviously needs to be improved massively (at least a colorbar). What would you like to show from the actual data here? A nice figure would be Figure 2 in Correa and Menezes (2019); however, note that the drawn survey line in that figure are plainly wrong.

We computed all all six frequencies for all components (E_x, E_y, E_z) for both inline and broadside acquisition lines. However, we show here only the three frequencies and the responses with the strongest amplitudes (which are generally used in processing), which are E_x inline and broadside, and E_y broadside. The comparison is shown in Figure 9.

[DW]: There are some interesting features we should highlight. But again, we first need all results. E.g.

- 0.125 Hz: on the left side, `custEM` and `emg3d` are almost identical, but very different from MR3D. So there I actually trust more our calculation, or there is a difference in the model that we did not take into account.
- 0.125 Hz: on the right side, `custEM` is very similar to MR3D, whereas `emg3d` is off.
- 0.125 Hz and 0.5 Hz: on the right side, `custEM` seems to very similar to MR3D.

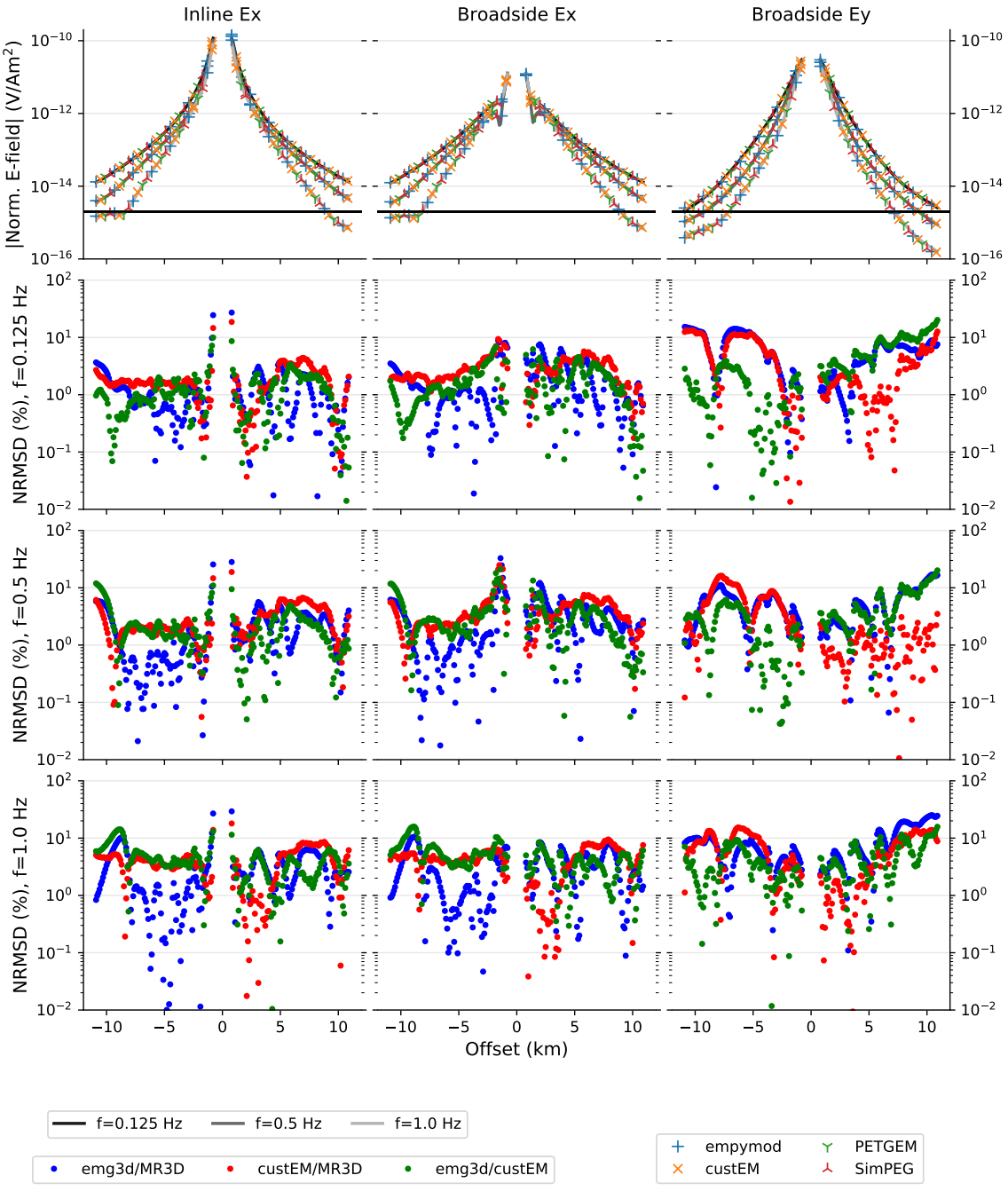

Figure 9: This is one heck of a figure. Just an idea, please make suggestions how to improve it (currently, `PETGEM` data is a copy of `custEM` data, and `SimPEG` data is a copy of `emg3d` data).

Code	CPU (s)	#Procs	RAM (GiB)	#dof
`custEM`	0 0	0	0 0	0
`emg3d`	0 0	0	0 0	0
`PETGEM`	0 0	0	0 0	0
`SimPEG`	0 0	0	0 0	0

Table 5: Comparison of used CPU and RAM and of number of cells used to discretize the Marlim R3D model.

DISCUSSION

Even if modelling results may look completely valid, they could not be correct at all. Verifying the performance of 3D CSEM codes for real 3D problems is only possible by cross-validating multiple solutions. Overall, we observed an excellent match between all solutions. The relative misfits, almost always smaller than 10 %, were related to the either weak or strong amplitudes of real or imaginary parts and can be attributed to particularities regarding the discretization, boundary effects, and interpolation. Considering this satisfactory result in terms of accuracy and robustness, it is more interesting to discuss the characteristics of the four considered codes with focus on the demand of computational resources and the capability of handling the investigated CSEM geometries, including the very important underlying mesh design. The following arguments are based on a comparison of only marine CSEM models, which are the focus of industry applications. Nevertheless, we are confident that many observations would hold in a similar manner for land-based, airborne, or mixed setups.

We point out that making a synthetic and a real benchmark model available to cross-validate any 3D CSEM code was the determining reason for their choice, regardless of the suitability of the considered FD and FE codes for these problems. The first model, including the block-anomalies variation, is exactly reproducible by using any numerical method with either regular or irregular computational domains. Aside from the idea of analyzing an industrially relevant benchmark example, the Marlim R3D model was also considered since the primary design from Correa and Menezes (2019) uses a regular discretisation. At least theoretically, the exact geometry might be reconstructed with unstructured meshes, but it is never possible to convert an irregular geometry to a regular grid without approximations. Hence, the chosen examples favoured the FD codes, since they used either regular grids or octree-meshes, whereas the two FE codes were required to tetrahedralise the model geometries.

In the first model, the necessity of discretising regular structures with tetrahedra, especially thin layers, led to comparatively large FE problem sizes and computational resource demands. FE codes with unstructured meshes were probably the poorest choice of all commonly used methods for this setup in terms of computational performance. However, thinking about introducing just one slight irregularity such as bathymetry, a thin dipping plate or an anticline, the performance behaviour could likely switch. All of these changes would barely change the problem size for the FE codes, but significantly increase it for the FD codes if the true geometry should be approximated sufficiently accurate.

The real Marlim R3D model was, ironically, the ideal case for FE codes because it's structure was characterized by irregular lithological horizons obtained from seismic data. Since the corresponding published resistivity model was defined on a regular grid, we were forced to approximate the comparatively fine regular discretization by an unstructured one and interpolate the resistivity data for being able to apply the FE codes to this problem. We are completely aware about the ineptness of this re-approximation procedure, unless it served for the cross-validation purposes. Nevertheless, the chosen tetrahedral discretization, especially in combination with second order basis functions, showed clearly the intrinsic strength of the FE codes, requiring significantly less dof for meshing irregular geometries in general. The p2 FE solution required only ≈9 M dof instead of ≈90 M dof for the FD system by Correa and Menezes (2019), disregarding the structure of the system matrices.

The codes under consideration use direct and indirect solvers. In general, the biggest advantage of iterative solvers with appropriate preconditioners are the comparatively small memory requirements. Direct solvers can be considered as most robust to solve any ill-conditioned SLE, which is most important for systems of the FE method on unstructured meshes. As the consumed computational resources indicate, a more powerful machine than a laptop is required for most 3D CSEM problems, but no high-performance-computing architecture.

The computation times can differ significantly for specific simulations, but there is no clear general advantage for one of the solver types. In our models, the runtime comparison favoured the iterative solver and therefore, `emg3d`. Even though not shown here, direct solvers exhibit their strongest advantage in terms of computation times if responses of multiple CSEM transmitters need to be calculated in the same computational domain. As the system matrix factorisation requires 98-99 % of the solution time, computations for additional sources come at almost no cost for direct solvers, i.e., MUMPS, whereas it would come at the same cost as the first source for `emg3d`. For the sake of completeness, note that there are also advanced techniques for iterative solvers to make use of preconditioners, factorisations, or intermediate solutions of the previous SLE to speed up the calculation for multiple sources, but this argumentation is beyond the scope of this work.

We see this work as the start for future elaboration on the validation of modelling codes, not necessarily restricted to CSEM problems. Much more comparisons and examples are required which not only result in the finding that a sufficient accuracy was observed. 3D CSEM modelling is a difficult task, which requires many considerations. It starts with the selection of the right code for the problem. We found that the meshing task, which is most performance determining, is particularly difficult and relevant, choosing cells small enough to appropriately represent the model yet to be as coarse as possible still achieving the desired precision. The required model extent has to be considered as well, thinking also about effects of the so-called airwave in marine setups or boundary meshes in general. This is not a new finding but rather well known fact. We are surprised that only the minority of publications in the field of 3D EM modelling considers a sufficiently detailed elaboration of the meshing process.

A completely objective evaluation of codes is only possible by cross-validation. Nevertheless, we point out that using different discretizations or formulations of the numerical approach, e.g. by choosing field or potential approaches, second or total field formulations, different polynomial order basis functions, and others, is a suitable alternative to the cross-comparison of multiple codes for self-validating complex 3D results. This is a very powerful

method to confirm the general functionality of codes for any modelling problem, which might not be covered by existing cross-validation benchmark examples. Considering for instance this study, it would be very optimistic to estimate the accuracy of the utilized codes for a land-based mineral exploration setup with a highly-conductive, steeply-dipping conductor. However, each modeller could run multiple simulations with different configurations to increase the reliability of the obtained results.

[DW]: I would like to capture as many as possible of the (I) differences, (II) difficulties, (III) advantages, and (IV) disadvantages, that we encountered in this exercise. This includes differences from direct to iterative solver and differences from FE to FD codes etc. Please just jot down things that come to your mind.

[OCR]: I think an important point is the user experience (modeler). I know this is an intangible element but perhaps we could add a line about it. It is my opinion

[LH]: From the discussion in slack - also a TODO on adding points on if there were any changes (or new motivation for development) based on this exercise –> goal: highlight benefits of open source collaborations.
Within SimPEG: This work has motivated feature development including: (a) interpolation and averaging strategies from mapping physical properties on a fine mesh to a coarser mesh for computation and (b) new examples to include in the documentation for designing OcTree meshes. Furthermore, as a part of a broader development objective of interoperating with other forward-simulation engines, connecting `emg3d` and `SimPEG` provided a motivating use-case for the latest refactor and release of `SimPEG` 0.14.0 (SimPEG contributors et al., 2020).

CONCLUSIONS & OUTLOOK

We compared four different open-source 3D CSEM modellers by computing responses for a layered, anisotropic model, including a modification with three resistive blocks, and the realistic marine Marlim R3D model. All comparisons exhibited an excellent reliability of the solutions. Our data should make it very easy for new codes to have a readily available data set to test against and verify. We are confident that the discussions about runtimes, memory consumptions, solver choices, and, most important, discretisations of the 3D CSEM problems help potential users not only to decide which modeller is best suited for specific tasks, but also to support their individual projects.

As none of the considered methods is best suited for all problems, it is important to have more test models for various scenarios in future. Our study is limited to marine CSEM cases in the frequency-domain. Reasonable extensions include providing benchmark models for land, airborne, or mixed CSEM environments as well as time-domain data. Another addition to this work would be focusing on complex irregular models, tailored for an FE or FV code based on unstructured meshes, and to compare how the FD codes can cope with it.

We encourage the community to not only work on new open-source code developments but also to create a landscape of easily accessible benchmark models for increasing the number of reliable and reproducible solutions. Our study could provide one of the primary inputs for this task. We are confident that the mentioned development will and should be taking place, taking into account the current general trend in science to open-source papers, codes or data.

ACKNOWLEDGMENT

We would like to thank Paulo Menezes for the help and explanations with regards to the Marlim R3D model and corresponding CSEM data, and for making their actual computation model available under an open-source license.

The work of D.W. was conducted within the Gitaro.JIM project funded through MarTERA, a *European Union's Horizon 2020* research and innovation programme, grant agreement N° 728053; martera.eu.

The development of `custEM` as part of the DESMEX project was funded by the Germany Ministry for Education and Research (BMBF) in the framework of the research and development program Fona-r4 under grant 033R130D.

The work of O.C-R. has received funding from the *European Union's Horizon 2020 programme* under the *Marie Sklodowska-Curie* grant agreement N° 777778. Further, the development of `PETGEM` has received funding from the *European Union's Horizon 2020 programme*, grant agreement N° 828947, and from the Mexican Department of Energy, CONACYT-SENER Hidrocarburos grant agreement N° B-S-69926. Furthermore, O.C-R. has been 65% cofinanced by the European Regional Development Fund (ERDF) through the Interreg V-A Spain-France-Andorra program (POCTEFA2014-2020). POCTEFA aims to reinforce the economic and social integration of the French-Spanish-Andorran border. Its support is focused on developing economic, social and environmental cross-border activities through joint strategies favouring sustainable territorial development.

DATA

All files to rerun the different models with the four codes and reproduce the shown results are available at ....

Put up on Zenodo, link here.

REFERENCES

Abhyankar, S., J. Brown, E. M. Constantinescu, D. Ghosh, B. F. Smith, and H. Zhang, 2018, PETSc/TS: A modern scalable ODE/DAE solver library: arXiv preprint arXiv:1806.01437.

Alumbaugh, D. L., G. A. Newman, L. Prevost, and J. N. Shadid, 1996, Three-dimensional wideband electromagnetic modeling on massively parallel computers: Radio Science, **31**, no. 1, 1–23; doi: 10.1029/95RS02815.

Amestoy, P. R., I. S. Duff, J. Koster, and J.-Y. L'Excellent, 2001, A fully asynchronous multifrontal solver using distributed dynamic scheduling: SIAM Journal on Matrix Analysis and Applications, **23**, no. 1, 15–41; doi: 10.1137/S0895479899358194.

Aruliah, D., and U. Ascher, 2002, Multigrid preconditioning for Krylov methods for time-harmonic Maxwell's equations in three dimensions: SIAM Journal on Scientific Computing, **24**, no. 2, 702–718; doi: 10.1137/S1064827501387358.

Astic, T., L. J. Heagy, and D. W. Oldenburg, 2020, Petrophysically and geologically guided multi-physics inversion using a dynamic Gaussian mixture model: (in review) Geophysical Journal International.

Avdeev, D. B., 2005, Three-dimensional electromagnetic modelling and inversion from theory to application: Surveys in Geophysics, **26**, no. 6, 767–799; doi: 10.1007/s10712-005-1836-x.

Broggini, F., J. Dellinger, S. Fomel, and Y. Liu, 2017, Reproducible research: Geophysics papers of the future–Introduction: Geophysics, **82**, no. 6, WBi–WBii; doi: 10.1190/geo2017-0918-spseintro.1.

Börner, R.-U., 2010, Numerical modelling in geo-electromagnetics: Advances and challenges: Surveys in Geophysics, **31**, no. 2, 225–245; doi: 10.1007/s10712-009-9087-x.

Cai, H., X. Hu, J. Li, M. Endo, and B. Xiong, 2017, Parallelized 3D CSEM modeling using edge-based finite element with total field formulation and unstructured mesh: Computers & Geosciences, **99**, 125–134; doi: 10.1016/j.cageo.2016.11.009.

Cai, H., B. Xiong, M. Han, and M. Zhdanov, 2014, 3D controlled-source electromagnetic modeling in anisotropic medium using edge-based finite element method: Computers & Geosciences, **73**, 164–176; doi: 10.1016/j.cageo.2014.09.008.

Carvalho, B. R., and P. T. L. Menezes, 2017, Marlim R3D: a realistic model for CSEM simulations–phase I: model building: Brazilian Journal of Geology, **47**, no. 4, 633–644; doi: 10.1590/2317-4889201720170088.

Castillo-Reyes, O., J. de la Puente, and J. M. Cela, 2018, PETGEM: A parallel code for 3D CSEM forward modeling using edge finite elements: Computers & Geosciences, **119**, 126–136; doi: 10.1016/j.cageo.2018.07.005.

Castillo-Reyes, O., J. de la Puente, L. E. García-Castillo, and J. M. Cela, 2019, Parallel 3D marine controlled-source electromagnetic modeling using high-order tetrahedral nédélec elements: Geophysical Journal International, **219**, 39–65; doi: 10.1093/gji/ggz285.

Chung, Y., J.-S. Son, T. J. Lee, H. J. Kim, and C. Shin, 2014, Three-dimensional modelling of controlled-source electromagnetic surveys using an edge finite-element method with a direct solver: Geophysical Prospecting, **62**, no. 6, 1468–1483; doi: 10.1111/1365-2478.12132.

Clemens, M., and T. Weiland, 2001, Discrete electromagnetism with the finite integration technique: PIER, **32**, 65–87; doi: 10.2528/PIER00080103.

Cockett, R., S. Kang, L. J. Heagy, A. Pidlisecky, and D. W. Oldenburg, 2015, SimPEG: An open source framework for simulation and gradient based parameter estimation in geophysical applications: Computers & Geosciences, **85**, 142–154; doi: 10.1016/j.cageo.2015.09.015.

Commer, M., and G. Newman, 2004, A parallel finite-difference approach for 3D transient electromagnetic modeling with galvanic sources: Geophysics, **69**, no. 5, 1192–1202; doi: 10.1190/1.1801936.

Correa, J. L., and P. T. L. Menezes, 2019, Marlim R3D: A realistic model for controlled-source electromagnetic simulations–Phase 2: The controlled-source electromagnetic data set: Geophysics, **84**, no. 5, E293–E299; doi: 10.1190/geo2018-0452.1.

da Silva, N. V., J. V. Morgan, L. MacGregor, and M. Warner, 2012, A finite element multifrontal method for 3D CSEM modeling in the frequency domain: Geophysics, **77**, no. 2, E101–E115; doi: 10.1190/geo2010-0398.1.

Das, U. C., and S. K. Verma, 1982, Electromagnetic response of an arbitrarily shaped three-dimensional conductor in a layered earth – numerical results: Geophysical Journal International, **69**, no. 1, 55–66; doi: 10.1111/j.1365-246X.1982.tb04935.x.

Druskin, V., and L. Knizhnerman, 1994, Spectral approach to solving three-dimensional maxwell's diffusion equations in the time and frequency domains: Radio Science, **29**, no. 4, 937–953; doi: 10.1029/94RS00747.

Fedorenko, R. P., 1964, The speed of convergence of one iterative process: USSR Computational Mathematics and Mathematical Physics, **4**, no. 3, 227–235; doi: 10.1016/0041-5553(64)90253-8.

Fournier, D., L. J. Heagy, and D. W. Oldenburg, 2020, Sparse magnetic vector inversion in spherical coordinates: Geophysics, **85**, no. 3, J33–J49.

Grayver, A. V., and T. V. Kolev, 2015, Large-scale 3D geoelectromagnetic modeling using parallel adaptive high-order finite element method: Geophysics, **80**, no. 6, E277–E291; doi: 10.1190/geo2015-0013.1.

Grayver, A. V., R. Streich, and O. Ritter, 2013, Three-dimensional parallel distributed inversion of CSEM data using a direct forward solver: Geophysical Journal International, **193**, no. 3, 1432–1446; doi: 10.1093/gji/ggt055.

Haber, E., 2014, Computational Methods in Geophysical Electromagnetics; doi: 10.1137/1.9781611973808.

Haber, E., and U. M. Ascher, 2001, Fast finite volume simulation of 3D electromagnetic problems with highly discontinuous coefficients: SIAM Journal on Scientific Computing, **22**, no. 6, 1943–1961; doi: 10.1137/S1064827599360741.

Haber, E., and S. Heldmann, 2007, An octree multigrid method for quasi-static maxwell's equations with highly discontinuous coefficients: Journal of Computational Physics, **223**, no. 2, 783–796; doi: https://doi.org/10.1016/j.jcp.2006.10.012.

Heagy, L. J., R. Cockett, S. Kang, G. K. Rosenkjaer, and D. W. Oldenburg, 2017, A framework for simulation and inversion in electromagnetics: Computers and Geosciences, **107**, 1–19.

Hohmann, G. W., 1975, Three-dimensional induced polarization and electromagnetic modeling: Geophysics, **40**, no. 2, 309–324; doi: 10.1190/1.1440527.

Hursán, G., and M. S. Zhdanov, 2002, Contraction integral equation method in three-dimensional electromagnetic modeling: Radio Science, **37**, no. 6, 1–1–1–13; doi: 10.1029/2001RS002513.

Jahandari, H., and C. G. Farquharson, 2014, A finite-volume solution to the geophysical electromagnetic forward problem using unstructured grids: Geophysics, **79**, no. 6, E287–E302; doi: 10.1190/geo2013-0312.1.

Jaysaval, P., D. Shantsev, and S. de la Kethulle de Ryhove, 2014, Fast multimodel finite-difference controlled-source electromagnetic simulations based on a Schur complement approach: Geophysics, **79**, no. 6, E315–E327; doi: 10.1190/geo2014-0043.1.

Jaysaval, P., D. V. Shantsev, and S. de la Kethulle de Ryhove, 2015, Efficient 3-D controlled-source electromagnetic modelling using an exponential finite-difference method: Geophysical Journal International, **203**, no. 3, 1541–1574; doi: 10.1093/gji/ggv377.

Jaysaval, P., D. V. Shantsev, S. de la Kethulle de Ryhove, and T. Bratteland, 2016, Fully anisotropic 3-D EM modelling on a Lebedev grid with a multigrid pre-conditioner: Geophysical Journal International, **207**, no. 3, 1554–1572; doi: 10.1093/gji/ggw352.

Kruglyakov, M., and L. Bloshanskaya, 2017, High-performance parallel solver for integral equations of electromagnetics based on Galerkin method: Mathematical Geosciences, **49**, no. 6, 751–776; doi: 10.1007/s11004-017-9677-y.

Kruglyakov, M., A. Geraskin, and A. Kuvshinov, 2016, Novel accurate and scalable 3-D MT forward solver based on a contracting integral equation method: Computers & Geosciences, **96**, 208–217; doi: 10.1016/j.cageo.2016.08.017.

Lam, S. K., A. Pitrou, and S. Seibert, 2015, Numba: A LLVM-based python JIT compiler: Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC, 1–6; doi: 10.1145/2833157.2833162.

Langtangen, H. P., A. Logg, and A. Tveito, 2016, Solving PDEs in Python: The FEniCS Tutorial I: Springer International Publishing, volume **3** *of* Simula SpringerBriefs on Computing; doi: 10.1007/978-3-319-52462-7.

Lebedev, V. I., 1964, Difference analogues of orthogonal decompositions, basic differential operators and some boundary problems of mathematical physics. I: USSR Computational Mathematics and Mathematical Physics, **4**, no. 3, 69–92; doi: 10.1016/0041-5553(64)90240-X.

Liu, R., R. Guo, J. Liu, C. Ma, and Z. Guo, 2018, A hybrid solver based on the integral equation method and vector finite-element method for 3D controlled-source electromagnetic method modeling: Geophysics, **83**, no. 5, E319–E333; doi: 10.1190/geo2017-0502.1.

Logg, A., K.-A. Mardal, and G. Wells, 2012, Automated solution of differential equations by the finite element method: The FEniCS book: Springer-Verlag Berlin Heidelberg, volume **84** *of* Lecture Notes in Computational Science and Engineering; doi: 10.1007/978-3-642-23099-8.

Maaø, F. A., 2007, Fast finite-difference time-domain modeling for marine-subsurface electromagnetic problems: Geophysics, **72**, no. 2, A19–A23; doi: 10.1190/1.2434781.

Mackie, R. L., J. T. Smith, and T. R. Madden, 1994, Three-dimensional electromagnetic modeling using finite difference equations: the magnetotelluric example.: Radio Science, **29**, no. 4, 923–935; doi: 10.1029/94RS00326.

Madsen, N. K., and R. W. Ziolkowski, 1990, A three-dimensional modified finite volume technique for Maxwell's equations: Electromagnetics, **10**, no. 1-2, 147–161; doi: 10.1080/02726349008908233.

Miensopust, M. P., P. Queralt, A. G. Jones, and the 3D MT modellers, 2013, Magnetotelluric 3-D inversion—a review of two successful workshops on forward and inversion code testing and comparison: Geophysical Journal International, **193**, no. 3, 1216–1238; doi: 10.1093/gji/ggt066.

Mulder, W. A., 2006, A multigrid solver for 3D electromagnetic diffusion: Geophysical Prospecting, **54**, no. 5, 633–649; doi: 10.1111/j.1365-2478.2006.00558.x.

Newman, G. A., and D. L. Alumbaugh, 1997, Three-dimensional massively parallel electromagnetic inversion–I. Theory: Geophysical Journal International, **128**, no. 2, 345–354; doi: 10.1111/j.1365-246X.1997.tb01559.x.

Newman, G. A., G. W. Hohmann, and W. L. Anderson, 1986, Transient electromagnetic response of a three-dimensional body in a layered earth: Geophysics, **51**, no. 8, 1608–1627; doi: 10.1190/1.1442212.

Oh, S., K. Noh, S. J. Seol, and J. Byun, 2015, 3D CSEM frequency-domain modeling and inversion algorithms including topography: SEG Technical Program Expanded Abstracts, 828–832; doi: 10.1190/segam2015-5898964.1.

Oldenburg, D. W., L. J. Heagy, S. Kang, and R. Cockett, 2019, 3D electromagnetic modelling and inversion: a case for open source: Exploration Geophysics, **0**, no. 0, 1–13.

Oristaglio, M., and B. Spies, 1999, Three-dimensional electromagnetics: SEG, volume **7** *of* Geophysical Developments; doi: 10.1190/1.9781560802154.

Puzyrev, V., J. Koldan, J. de la Puente, G. Houzeaux, M. Vázquez, and J. M. Cela, 2013, A parallel finite-element method for three-dimensional controlled-source electromagnetic forward modelling: Geophysical Journal International, **193**, no. 2, 678–693; doi: 10.1093/gji/ggt027.

Raiche, A. P., 1974, An integral equation approach to three-dimensional modelling: Geophysical Journal International, **36**, no. 2, 363–376; doi: 10.1111/j.1365-246X.1974.tb03645.x.

Rochlitz, R., N. Skibbe, and T. Günther, 2019, custEM: customizable finite element simulation of complex controlled-source electromagnetic data: Geophysics, **84**, no. 2, F17–F33; doi: 10.1190/geo2018-0208.1.

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

Schenk, O., and K. Gärtner, 2004, Solving unsymmetric sparse systems of linear equations with PARDISO: Future Generation Computer Systems, **20**, no. 3, 475–487; doi: 10.1016/j.future.2003.07.011.

Schwarzbach, C., R.-U. Börner, and K. Spitzer, 2011, Three-dimensional adaptive higher order finite element simulation for geo-electromagnetics—a marine csem example: Geophysical Journal International, **187**, no. 1, 63–74; doi: 10.1111/j.1365-246X.2011.05127.x.

Si, H., 2015, Tetgen, a Delaunay-based quality tetrahedral mesh generator: ACM Transactions on Mathematical Software (TOMS), **41**, no. 2, 1–36; doi: 10.1145/2629697.

SimPEG contributors, L. Heagy, S. Kang, D. Fournier, G. K. Rosenkjaer, J. Capriotti, T. Astic, D. C. Cowan, D. Marchant, M. Mitchell, J. Kuttai, D. Werthmüller, L. A. C. Mata, kai, F. Koch, B. Smithyman, K. Martens, C. Miller, C. Gohlke, S. Devriese, M. Wathen, and F. Perez, 2020, simpeg/simpeg: v0.14.0 simulation.

Skibbe, N., R. Rochlitz, T. Günther, and M. Müller-Petke, 2020, Coupled magnetic resonance and electrical resistivity tomography: An open-source toolbox for surface nuclear-magnetic resonance: Geophysics, **85**, no. 3, F53–F64; doi: 10.1190/geo2019-0484.1.

Sommer, M., S. Hölz, M. Moorkamp, A. Swidinsky, B. Heincke, C. Scholl, and M. Jegen, 2013, GPU parallelization of a three dimensional marine CSEM code: Computers & Geosciences, **58**, 91–99; doi: 10.1016/j.cageo.2013.04.004.

Streich, R., 2009, 3D finite-difference frequency-domain modeling of controlled-source electromagnetic data: Direct solution and optimization for high accuracy: Geophysics, **74**, no. 5, F95–F105; doi: 10.1190/1.3196241.

Tehrani, A. M., and E. Slob, 2010, Fast and accurate three-dimensional controlled source electromagnetic modelling†: Geophysical Prospecting, **58**, no. 6, 1133–1146; doi: 10.1111/j.1365-2478.2010.00876.x.

Uieda, L., 2018, Verde: Processing and gridding spatial data using Green's functions: Journal of Open Source Software, **3**, no. 29, 957.

van der Walt, S., S. C. Colbert, and G. Varoquaux, 2011, The NumPy array: A structure for efficient numerical computation: Computing in Science Engineering, **13**, no. 2, 22–30; doi: 10.1109/MCSE.2011.37.

Virtanen, P., R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, 2020, SciPy 1.0: Fundamental algorithms for scientific computing in python: Nature Methods, **17**, 261–272; doi: 10.1038/s41592-019-0686-2.

Wang, F., J. P. Morten, and K. Spitzer, 2018, Anisotropic three-dimensional inversion of CSEM data using finite-element techniques on unstructured grids: Geophysical Journal International, **213**, no. 2, 1056–1072; doi: 10.1093/gji/ggy029.

Wang, T., and G. W. Hohmann, 1993, A finite-difference, time-domain solution for three-dimensional electromagnetic modeling: Geophysics, **58**, no. 6, 797–809; doi:

10.1190/1.1443465.

Wannamaker, P. E., G. W. Hohmann, and S. H. Ward, 1984, Magnetotelluric responses of three-dimensional bodies in layered earths: Geophysics, **49**, no. 9, 1517–1533; doi: 10.1190/1.1441777.

Ward, S. H., and G. W. Hohmann, 1988, 4, *in* Electromagnetic Theory for Geophysical Applications: 130–311; doi: 10.1190/1.9781560802631.ch4.

Weiland, T., 1977, Eine Methode zur Lösung der Maxwellschen Gleichungen für sechskomponentige Felder auf diskreter Basis: Archiv für Elektronik und Übertragungstechnik, **31**, no. 3, 116–120; pdf: https://www.leibniz-publik.de/de/fs1/object/display/bsb00064886_00001.html.

Werthmüller, D., 2017, An open-source full 3D electromagnetic modeler for 1D VTI media in Python: empymod: Geophysics, **82**, no. 6, WB9–WB19; doi: 10.1190/geo2016-0626.1.

Werthmüller, D., W. A. Mulder, and E. C. Slob, 2019, emg3d: A multigrid solver for 3D electromagnetic diffusion: Journal of Open Source Software, **4**, no. 39, 1463; doi: 10.21105/joss.01463.

Yee, K., 1966, Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media: IEEE Transactions on Antennas and Propagation, **14**, no. 3, 302–307; doi: 10.1109/TAP.1966.1138693.

Zhang, Y., and K. Key, 2016, MARE3DEM: A three-dimensional CSEM inversion based on a parallel adaptive finite element method using unstructured meshes: SEG Technical Program Expanded Abstracts, 1009–1013; doi: 10.1190/segam2016-13681445.1.

Zhdanov, M. S., S. K. Lee, and K. Yoshioka, 2006, Integral equation method for 3D modeling of electromagnetic fields in complex structures with inhomogeneous background conductivity: Geophysics, **71**, no. 6, G333–G345; doi: 10.1190/1.2358403.