Open-source landscape for 3D CSEM modelling

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

ABSTRACT

I will draft a first abstract after the manuscript has been reviewed by everyone once.

For comments in the margin, use:

    \mycom[YOURINITIALS]{comment}

and for inline comments, use:

    \imycom[YOURINITIALS]{comment}

We will have to decide on a Journal. Current suggestions:
- GMD: Geoscientific Model Development
- SIG: Surveys in Geophysics
- SE: Solid Earth
- GJI: Geophysical Journal International
- GP: Geophysical Prospecting
- GEO: Geophysics
- CAG: Computers and Geosciences

I think *Geoscientific Model Development* and *Surveys in Geophysics* were the favourite ones when we discussed it ones. I personally would be very interested in GMD. What do you think? But also interesting would be the GJI, because the MT comparison was published there, so it would be in the same place. I just had a very bad experience with GP, so I am not inclined on that at all.

[**RR**]: GMD looks very interesting, my only concern is that it might not be in the scope of EM modelers, but we can probably reach modeleres from other communities as well.

INTRODUCTION

[**RR**]: We need to take care to not mix British English with AE, I'm very used to AE, but Dieter started with (at least what I can identify) mainly BE, so I tried to follow (mainly s instead of z and ou instead of o in some words). I hope there are not too many inconsistencies, as probably Lindsey has to fix all these mistakes in the end:-)

Controlled-source electromagnetic (CSEM) measurements are a frequently applied method in various geophysical exploration fields, such as geothermal and groundwater, oil and gas, mining, civil engineering, or geo-hazards. Modelling these electromagnetic fields is therefore of great interest to design survey layouts, understand the measured data and 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).

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) (Schwarzbach et al., 2011; Commer and Newman, 2004; da Silva et al., 2012; Puzyrev et al., 2013; Grayver et al., 2013; Zhang and Key, 2016), and finite volumes (FV) (Madsen and Ziolkowski, 1990; Haber and Heldmann, 2007; Jahandari and Farquharson, 2014; Clemens and Weiland, 2001; Mulder, 2006). 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; Grayver and Kolev, 2015; Chung et al., 2014; Jaysaval et al., 2014; 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).

Probably the main reason for the numerous amount of code developments was already outlined by Avdeev (2005), who finished his review 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.* » Some major exploration service companies as well as certain research consortia and research groups own their 3D CSEM codes, but how about codes that are open-source? Academic codes were traditionally often available upon request by authors. Codes distributed that way were often provided with the request to not share the code, and they often come without a license attached which means that they are not open-source. Also, at times there can be significant hurdles to install a code.

The meaning of the term open-source evolved rapidly in the last two decades. Today, open-source generally not only means that a code is available (with a proper license), but also that the code is hosted online, is under version control, has the possibility to file issues and make pull requests to fix bugs. These are important requirements for reproducible 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). Well maintained codes often have continuous integration that includes testing of the code base and online hosted documentation, often including examples and tutorials. As such, the term has shifted from purely open-source codes to developments in the open with increasingly building a global community. Related projects within the realm of geophysical exploration are e.g., pyGIMLi (Rücker et al., 2017) or Fatiando (Uieda, 2018).

[DW]: Others coming to your mind?

[RR]: i suggest to use also "emph" style for all other package names, not only ours

[RR]: no thoughts about geophysics, but there are many projects for related topics such as gmt, gis, mesh-gen

We continue addressing the topic of open-source developments with respect to 3D CSEM modeling. Even though it comprises still the minority of developments, the landscape in this field of research changed in the last five years. Before, there were only closed-source codes owned by companies or consortia and individual projects, often without a proper documentation, that could be requested from the authors. We introduce and compare four recent open-source projects with focus on 3D CSEM modeling. All presented codes are in the Python ecosystem and use either the FD 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 project. Analytical and semi-analytical solutions only exist for simple halfspace or layered-Earth models, which served mostly to verify new codes. Beyond them, 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, and 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 modeling 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 FD 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 continue not only extending the landscape of open-source codes and also creating a landscape of open-source benchmark models.

[**RR**]: Regarding "It is worth mentioning that there are many more 3D EM codes developed in other fields than geophysics such as communication systems (antennas, radar, satellites), medical imaging, or civil engineering. There are various open-source codes in these fields too, e.g., Elmer ([DW]: Add 2-3 proper references) (csc.fi/elmer, couldn't find a publication, just a poster). While these codes could potentially be used for the same goals as presented here we restrict our review to codes purpose-built and ready without further adjustments for geophysical applications."
I thought a lot about this but I just think the previous paragraph is not required in our introduction and haven't find a location where it fits considering different logical structures. I think it is also not important for this paper to tell "that there are more codes in EM than only geophysics", that's quite obvious. Maybe delete or shorten to 1 sentences to add Elmer and place this sentence somewhere else? I think, an appropriate list of all non-geophysical open-source EM codes should be very long anyway, we just don't know them or at least I'm not familiar with them.

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., 2019), and `SimPEG` (Cockett et al., 2015). 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 a quick rundown 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.

[**DW**]: are we all using the weak form?

[**DW**]: are we all using the diff. approx?

[**RR**]: too much info (redundancy to text) for table caption

	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	conda	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: All codes solve the weak form of Maxwell's equations in its differential form (DE) under the quasistatic approximation. 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. 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.

Each code should be briefly summarized by three points:
- Brief intro
- Main selling point
- Planned features
I think in a first round **everyone should just write what they want to write and think is important for their code**. In a second round we should then homogenize it, such that all codes have more or less the same length/weight and also similar topic/layout.

custEM

The customisable electromagnetic modeling 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 modeling, 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 modeling 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

TODO

SimPEG

TODO

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

> **[RR]**: I think the miensopust argument is rather suited to motivate our work in the introduction, i hope its fine that i moved this argument

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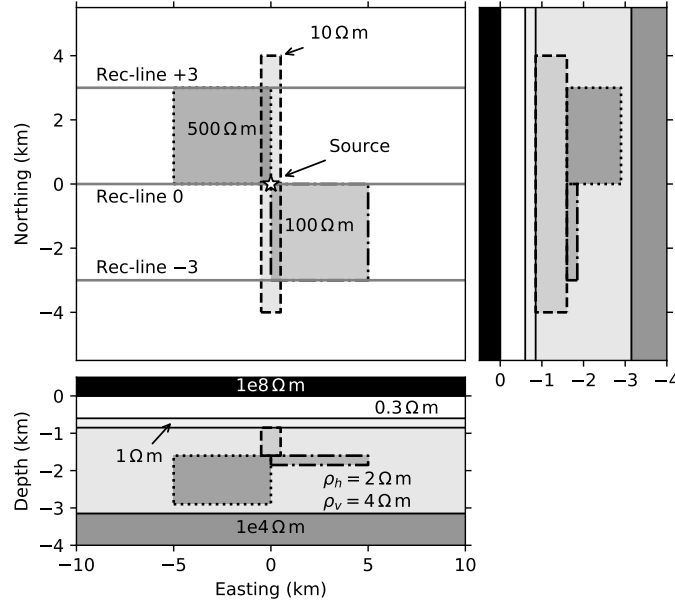
Figure 1: Sketch of the block model. The layered model consists of an air layer, a water layer, a thin top-layer followed by a thick, anisotropic background layer, and at the bottom a resistive basement layer. All three blocks are embedded in the thick background layer, which has VTI with $\lambda =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

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 $1\,\Omega\,\text{m}$, a 3.3 km thick, anisotropic layer of $\rho_{\text{h}} = 2\,\Omega\,\text{m}$ and $\rho_{\text{v}} = 4\,\Omega\,\text{m}$, and finally a resistive basement consisting of a lower halfspace of $1000\,\Omega\,\text{m}$. The 200 m long, x-directed source is located 50 m above the seafloor from $x = -50$ to $x = 50\,\text{m}$ in x-direction, at $y = 0\,\text{m}$. The x-directed receivers are placed on the seafloor every 200 m from $x = -10\,\text{km}$ to $x = +10\,\text{km}$ in three lines for $y = -3, 0, 3\,\text{km}$.

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 $y = 0$ km and $y = \pm 3$ km, which are identical.

Figure 2 illustrates the inline responses in the top-row for offsets $r > 0.5$ 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, whereas emg3d seems to do a very good job . Close to the 200 m long source, emg3d is less precise which is attributed to comparatively large cell sizes in this region.

[RR]: maybe reformulate to something like: whereas emg3d appears to be not affected by boundary effects
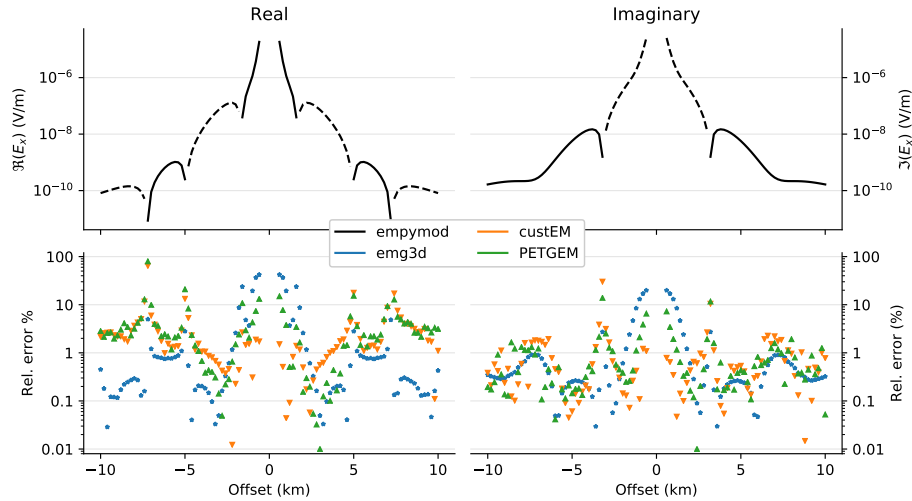
Figure 2: Comparison of the layered model for the inline receivers ($y = 0$ km). Top row shows the semi-analytical responses from empymod, and bottom row shows the relative error (%) of custEM, emg3d, PETGEM, and SimPEG.

[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 $y = \pm 3$ km are shown in Figure 3. The CPU and RAM requirements are listed in Table 3.
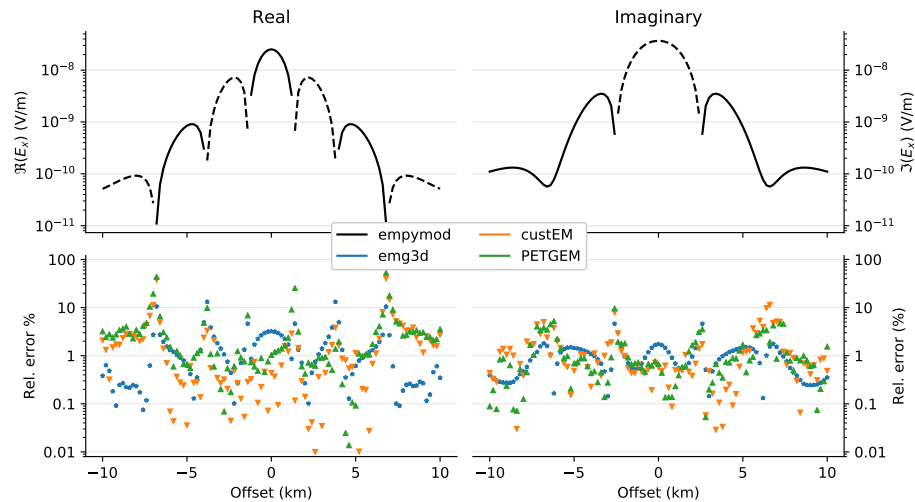
Figure 3: Comparison of the layered model for the broadside receivers ($y = -3$ km). Top row shows the semi-analytical responses from `empymod`, and bottom row show the relative error (%) of `custEM`, `emg3d`, `PETGEM`, and `SimPEG`.

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**]: I will fill this table out once we all submitted the final versions of the results.

In a second step, three resistive blocks were added in the $3.3\,\mathrm{km}$-thick, anisotropic background layer as shown in Figure 1. They have resistivities of $\rho = 10\,\Omega\,\mathrm{m}$ (shallow beam perpendicular to survey lines), $\rho = 100\,\Omega\,\mathrm{m}$ (thin plate, South-East), and $\rho = 500\,\Omega\,\mathrm{m}$ (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

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

[**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.

The results for the three receiver lines $y = -3, 0, 3\,\mathrm{km}$ 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`.

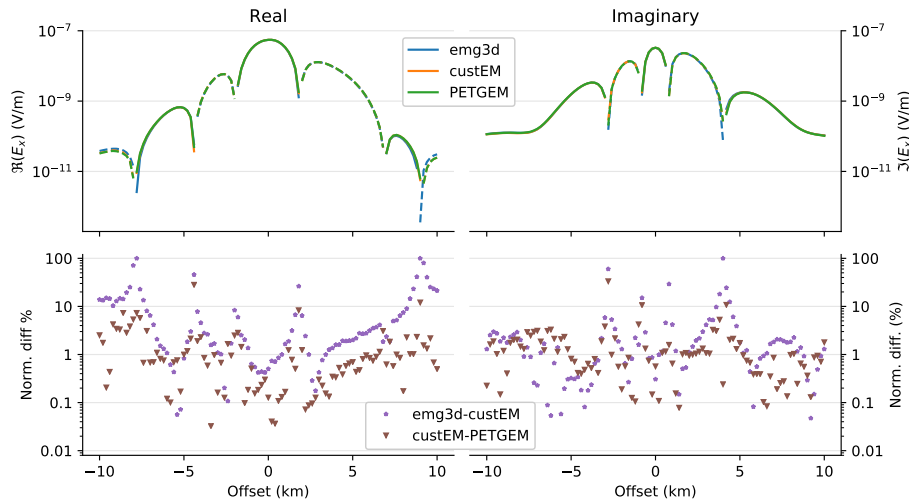[**DW**]: Write in detail when all results are available

Figure 4: Comparison of the block model for the broadside receivers at $y = -3\,\mathrm{km}$. Top row shows the responses and bottom row the NRMSDs (%) between `emg3d`–`custEM`, `custEM`–`PETGEM`, and `emg3d`–`SimPEG`.
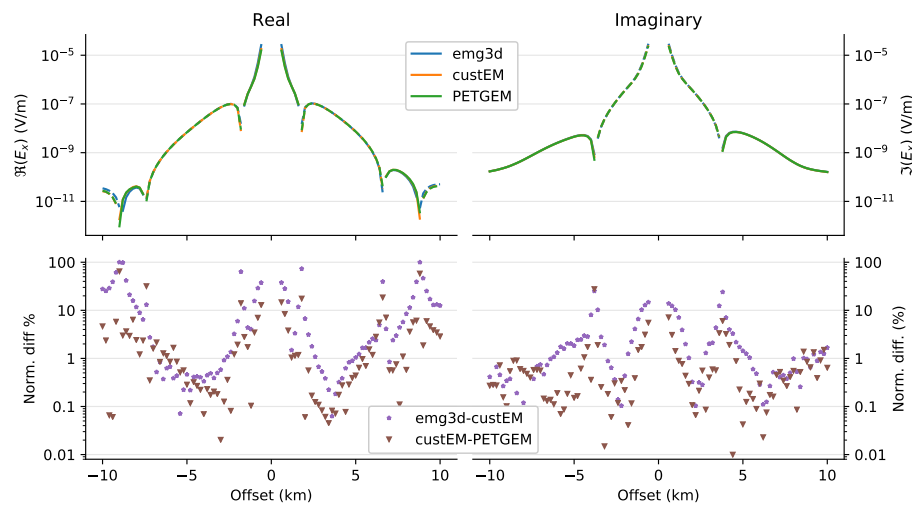
Figure 5: Comparison of the block model for the inline receivers at $y = 0$ km. Top row shows the responses and bottom row the NRMSDs (%) between `emg3d`–`custEM`, `custEM`–`PETGEM`, and `emg3d`–`SimPEG`.
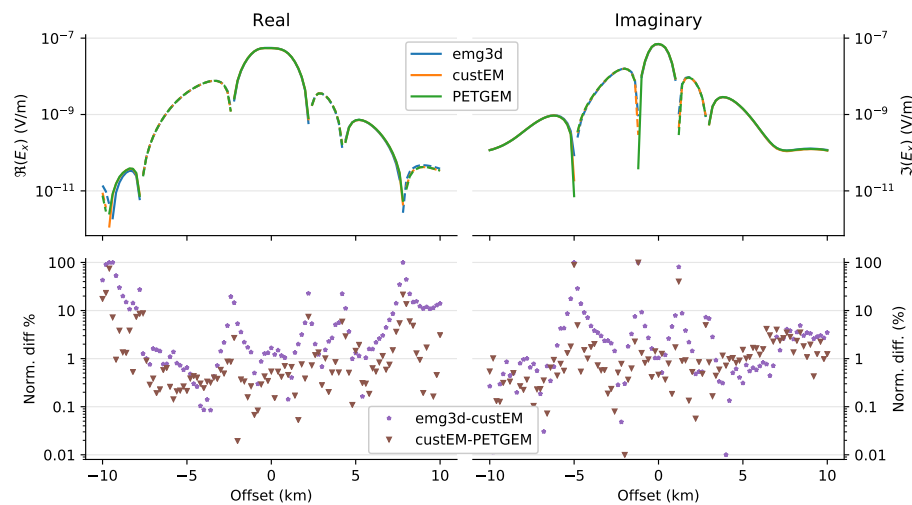

Figure 6: Comparison of the block model for the broadside receivers at $y = +3$ km. Top row shows the responses and bottom row the NRMSDs (%) between `emg3d`–`custEM`, `custEM`–`PETGEM`, and `emg3d`–`SimPEG`.

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.

[**DW**]: I will fill this table out once we all submitted the final versions of the results.

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. The thin layer was therefore limited in its horizontal extent to ???. This is the reason for the increased NRMSD at large offsets, and it has nothing to do with the accuracy of the computation itself.

[**DW**]: Raphael, to what x/y exactly?

[**RR**]: see imycom below figure 6

[**RR**]: 60 x 60 km (30 km in each direction from center). I already added a paragraph about mesh generation in the beginning, I suggest to remove this description here and jsut refer to like "Similar to the layered earth model, the largest errors occur at offsets > 8 km due to the differences in mesh design." We should also mention that relative errors are related to the magnitudes, as real is so much weaker at larger offsets, it is clear that errors increase significantly. Alternatively, we could show absolute magnitude or phase errors such as in the Marlim model.

Figure 7: 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 northeastern 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.

[**RR**]: I would delete this sentence, not that important and a bit redundant

The detailed reservoir model consists of 1022 by 371 by 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.

[**RR**]: I used gmt the last time 7 years ago , in between I tried the gmt4py tools, but they did not work well on my machine. I suggest to use two plots next to each other, a similar figure as in the notebook and a simple x-y coord sketch (identical extent as the bird perspective view in the other plot) with all Tx / Rx locations, highlighting the relevant line and Rx for us.

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.

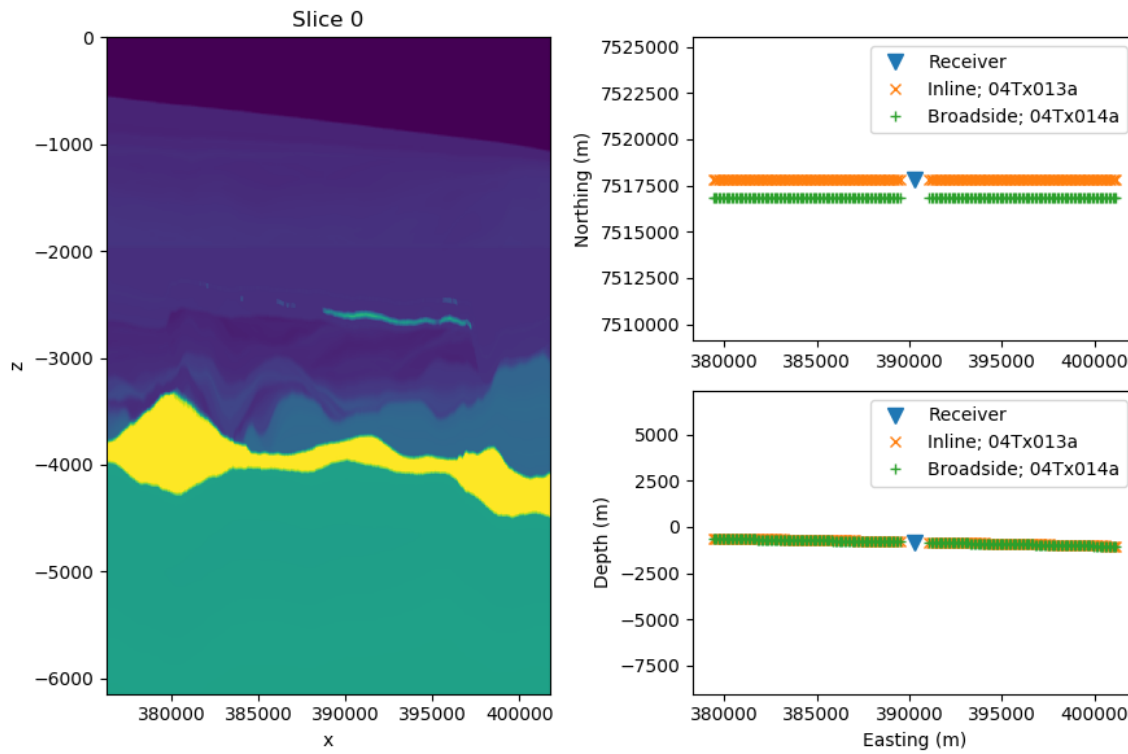- 0.125 Hz and 0.5 Hz: on the right side, `custEM` seems to very similar to MR3D.

Figure 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. If someone is familiar with GMT and happy to make that would be great. Also, the cross-section is currently heavily vertically exaggerated.

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.

[**DW**]: I will fill this table out once we all submitted the final versions of the results.
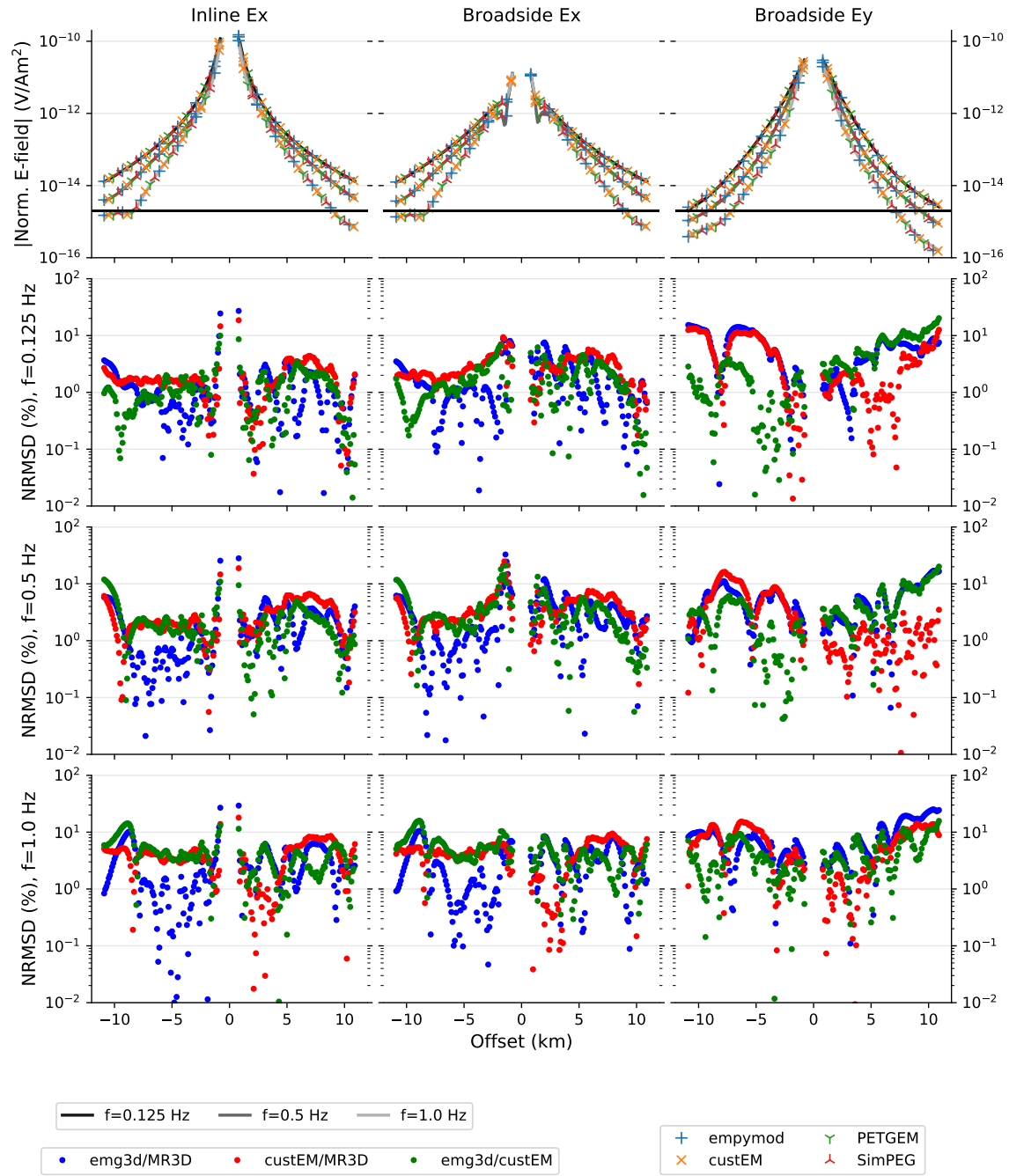
Figure 9: This is one heck of a figure. Just an idea, please make suggestions how to improve it (currently, `PETGEM` data is a copy of `custEM` data, and `SimPEG` data is a copy of `emg3d` data).

DISCUSSION

[**DW**]: Maybe we should create a Google-Doc for working on the Discussion collaboratively at the same time?

[**RR**]: if you prefer ok, but I don't think this is necessary, if everyone adds the own thoughts when he receives the manuscript. Same for conclusions. We need to take care to not mix discussion and conclusions, some of the statements from here belong rather to the conclusions, I hope it's fine that I reorganized the content. Aside from that, I already added all thoughts of mine.

Even if modeling 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 finite-difference codes, since they used either regular grids or octree-meshes, whereas the two finite-element 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 $\approx 9\,\mathrm{M}$ dof instead of $\approx 90\,\mathrm{M}$ 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 modeling 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 modeling 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 modeling 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 in-

crease 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.

CONCLUSIONS & OUTLOOK

[**DW**]: Maybe we should create a Google-Doc for working on the Conclusions collaboratively at the same time?

[**DW**]: I obviously haven't really done this section, just af few thoughts that I noted over time. I think the discussion (as the abstract) has to be tackled once the rest is at least in a draft status and all data is ready.

[**RR**]: I also rearranged some of your thoughts and formulated mine. I moved some of them (the few sentences that were rather redundant motivation content) to the introdcution.

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

[**DW**]: Write here your fundings; and potentially also minor contributors etc.

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.

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, 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.

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., 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.

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.

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 simula-

tion 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.

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.

Weiland, T., 1977, Eine Methode zur Lösung der Maxwellschen Gleichungen für sechskom-

ponentige 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.