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.

INTRODUCTION

Controlled-source electromagnetic (CSEM) measurements are a frequently applied method in various geophysical exploration fields, such as geothermal and ground water, oil and gas, mining, civil engineering, or geo-hazards. Modelling these electromagnetic fields is therefore of great interest in order to understand the measured data. Publications regarding 3D modelling in electromagnetic methods started to appear as early as the 1970's and 1980's. These early publications where 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 in the publication of 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 publications. The amount of available numerical solutions can be overwhelming and is part of the reason why there are hundreds of publications about the topic. There are the different methods to solve Maxwell's equation, such as 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, for instance finite differences (FD) (Yee, 1966; Wang and Hohmann, 1993; Mackie et al., 1994; Druskin and Knizhnerman, 1994; Streich, 2009; Sommer et al., 2013), and finite elementes (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), finite volume (FV) (Madsen and Ziolkowski, 1990; Haber and Heldmann, 2007; Jahandari and Farquharson, 2014; Clemens and Weiland, 2001; Mulder, 2006). And these are just the most common ones.

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), OcTree meshes (Haber and Heldmann, 2007), or hexagonal meshes (Cai et al., 2014).

The biggest variety of all exists probably in the available solvers to solve the system of linear equations; 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).

A very well written overview up to the year 2005 of the different approaches to 3D EM modelling is given by Avdeev (2005), and also by Börner (2010). In the last 15 years the publications with regards to 3D EM modelling grew tremendously, driven by the ever increasing computing power. Avdeev 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.*»

It is this topic that we address, open-source 3D CSEM codes. Some major exploration service companies as well as certain research consortia and research groups have their 3D CSEM codes. But how about codes that are open-source? Research code was traditionally often available upon request by the author. Codes distributed that way often come with the request to not share the code, and they often come without a license attached at all (which means that they are not open-source). Also, at times there can be significant hurdles to install a code. However, the meaning of open-source evolved rapidly in the last decades. Today, open-source generally not only means that a code is available (with a proper license). The meaning includes much more these days, such as that the code is hosted online, is under version control, has the possibility to file issues and make pull requests to fix bugs. Well maintained codes often have continuous integration that includes testing of the code base and online hosted documentation which often includes code documentation, examples, and tutorials. As such the term has shifted from purely open-source code to development in the open and increasingly building a global community. There are a number of projects within the realm of geophysical exploration, besides the ones we present here, e.g. pyGIMLi

[**DW**]: Others coming to your mind?

(Rücker et al., 2017) and Fatiando (Uieda, 2018). Newer programming environments have also brought much simplifications in the installation process and have most importantly simplified it for code developers to make their codes easily available and installable. All our presented codes are in the Python ecosystem, and each code can be installed with a single command (which includes download and installation).

[**DW**]: wrong term, think about it

A second topic we want to address is verification. New codes are generally verified against analytical solutions. However, analytical and semi-analytical solutions only exist for very simple models. There is no possibility to check how good a computed response is, expect for comparing the result from different modellers. If different modellers using different discretisation and implementations of Maxwell's equations yield the same result it can give confidence in the accuracy. Providing these two models and results from four different codes should give a benchmark with which new codes can be compared to and validated with.

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 (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.

[**DW**]: Add 2-3 proper references

In the following section we introduce the codes under consideration. The numerical result consider two cases. First a layered background with vertical transverse isotropy containing three blocks, followed by a realistic, complex 3D model.

CODES

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 not only come with an open-source license, but they are also in an online-hosted version-control system with feedback and tracking possibilities (raising issues, filing pull requests), have extensive online documentation including many examples, have continuous integration to some degree, and also importantly are all easily installable, without any user-side compilations.

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?

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`	`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	`pip`; `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.

custEM

TODO

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

In this numerical section we compute the responses for two different models to verify that these different 3D codes yield the same electromagnetic responses. The first model is a simple one consisting of a layered, anisotropic background, in which three blocks are embedded, and the second model is a very complex, realistic model.

Block Model

Miensopust et al. (2013) published a review of two workshops dealing with the verification of magnetotelluric forward and inversion codes, hence a similar exercise to what we do here for CSEM codes. The block model is a derivation of the *Dublin Test Model 1*, a model from their first workshop. 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).

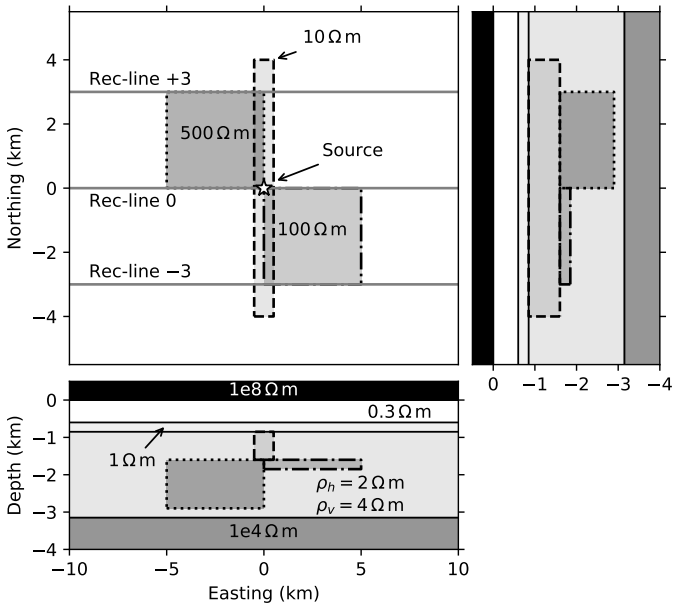

Figure 1: Sketch of the block model. The layered model consists of an air layer, a water layer, a thin top-layer followed by a thick, anisotropic background layer, and at the bottom a resistive basement layer. All three blocks are embedded in the thick background layer, which has VTI with $\lambda =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

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.

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 receiver line $y = 0\,\text{km}$ and $y = -3\,\text{km}$, respectively (for the layered background the responses for the two receiver-lines of $y = \pm 3\,\text{km}$ are identical, so we only show one here).

Figure 2 shows in the top-row the inline responses for offsets $r > 0.5\,\text{km}$. As all codes compute the same we only show the semi-analytical result for the real and imaginary amplitudes, and then in the bottom row the relative error of the 3D codes. The results show that the codes are literally yielding the same result, the relative error is for big parts below 1 %, and for almost all parts below a few percent. Some differences are due to the chosen discretizations. E.g., `emg3d` is less precise close to the 200 m long source. This could be improved by choosing smaller cell sizes, at the cost of computation time. At large offsets, on the other hand, `emg3d` seems to do a very good job.

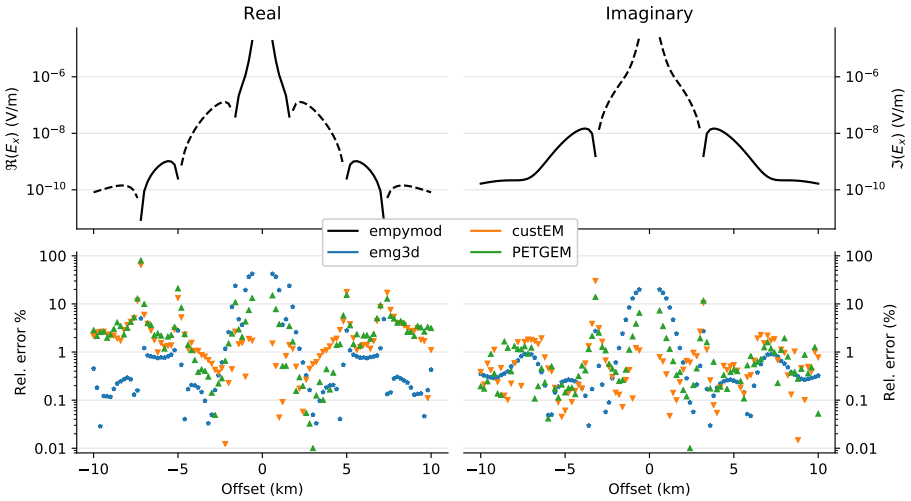

Figure 2: Comparison of the layered model for the inline receivers ($y = 0\,\text{km}$). Top row shows the semi-analytical responses from `empymod`, and bottom row shows the relative error (%) of `custEM`, `emg3d`, `PETGEM`, and `SimPEG`.

The corresponding broadside responses for $y = -3\,\text{km}$ is shown in Figure 3, and the required CPU and RAM is shown in Table 3.

Three resistive blocks are added in the 3.3 km-thick, anisotropic background layer, as

Code	CPU (s)	#Procs	RAM (GiB)	#Cells
`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.

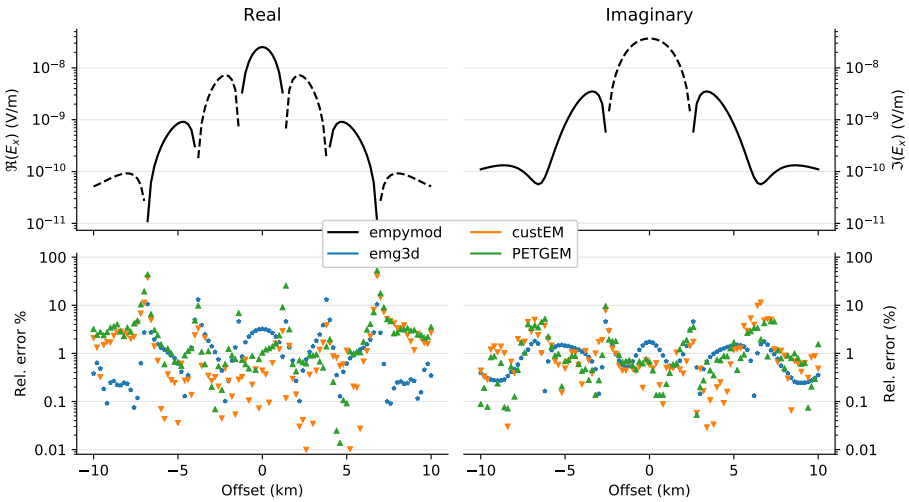

Figure 3: Comparison of the layered model for the broadside receivers ($y = -3\,\text{km}$). Top row shows the semi-analytical responses from `empymod`, and bottom row show the relative error (%) of `custEM`, `emg3d`, `PETGEM`, and `SimPEG`.

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), $\rho = 500\,\Omega\,\text{m}$ (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

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

[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?

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. What we can conclude from the NRMSD is 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

The difference becomes only significant at offsets larger than 8 km. However, 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?

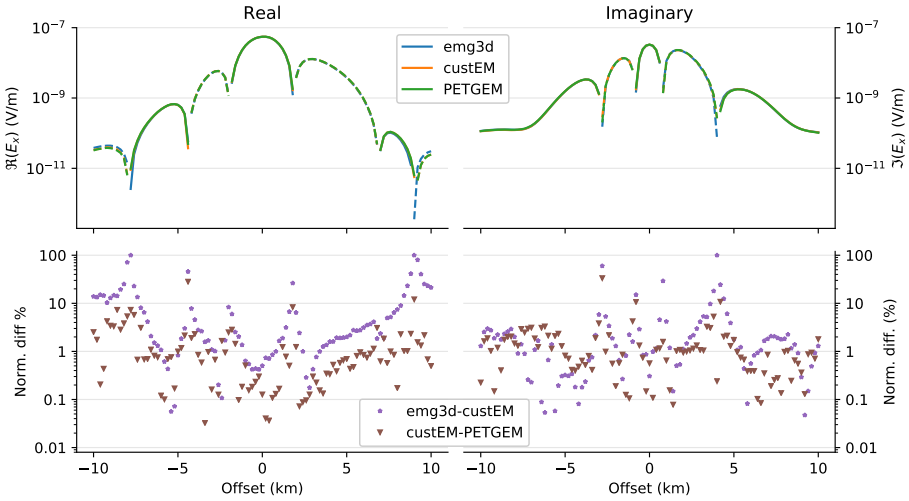

Figure 4: Comparison of the block model for the broadside receivers at $y = -3$ km. Top row shows the responses and bottom row the NRMSDs (%) between `emg3d`–`custEM`, `custEM`–`PETGEM`, and `emg3d`–`SimPEG`.

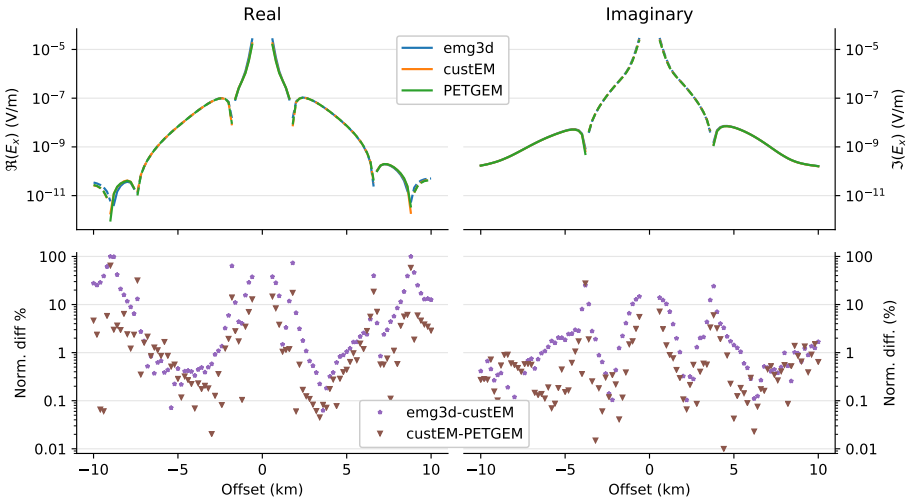

Figure 5: Comparison of the block model for the inline receivers at $y = 0$ km. Top row shows the responses and bottom row the NRMSDs (%) between `emg3d`–`custEM`, `custEM`–`PETGEM`, and `emg3d`–`SimPEG`.

Code	CPU (s)	#Procs	RAM (GiB)	#Cells
`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.

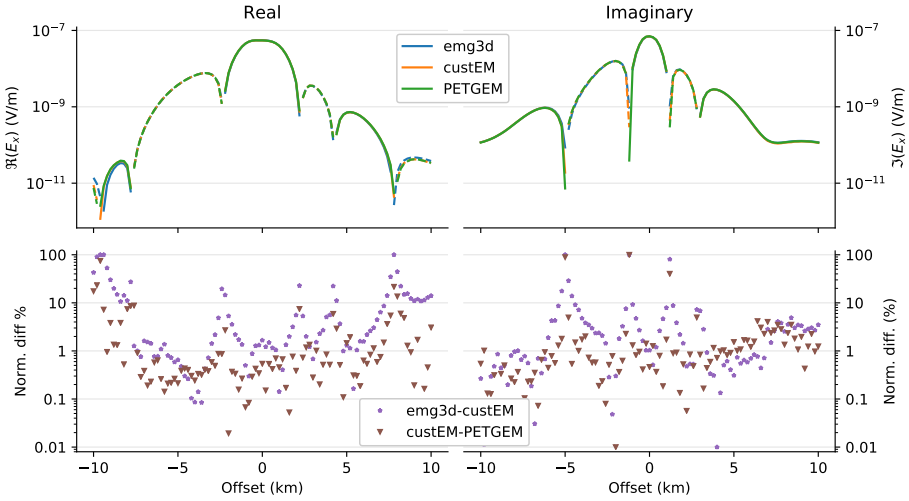

Figure 6: Comparison of the block model for the broadside receivers at $y = +3\,\text{km}$. Top row shows the responses and bottom row the NRMSDs (%) between `emg3d`–`custEM`, `custEM`–`PETGEM`, and `emg3d`–`SimPEG`.

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. Using this model Correa and Menezes (2019) computed CSEM data for six frequencies from 0.125 Hz to 1.25 Hz, which they also 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 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 7, together with the acquisition information.

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 8.

[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.

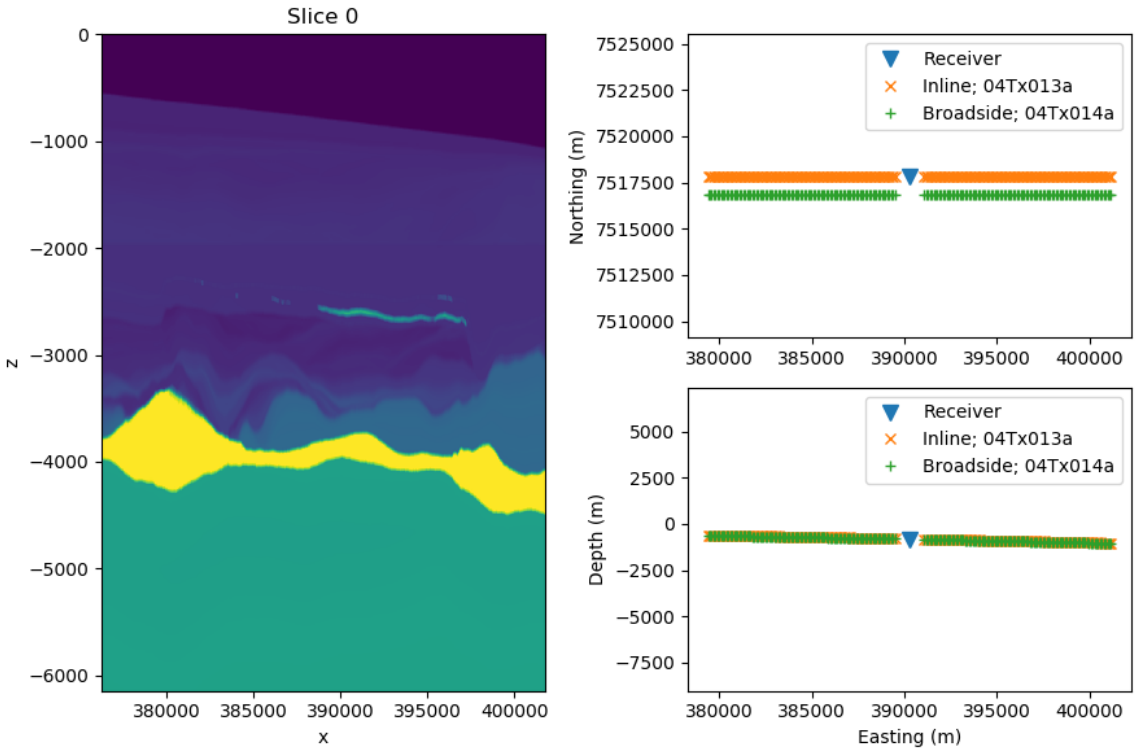

Figure 7: This figure obviously needs to be improved massively (at least a colorbar). What would you like to show from the actual data here? A nice figure would be Figure 2 in Correa and Menezes (2019); however, note that the drawn survey line in that figure are plainly wrong. If someone is familiar with GMT and happy to make a make that would be great. Also, the cross-section is currently heavily vertically exaggerated.

Code	CPU (s)	#Procs	RAM (GiB)	#Cells
`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.

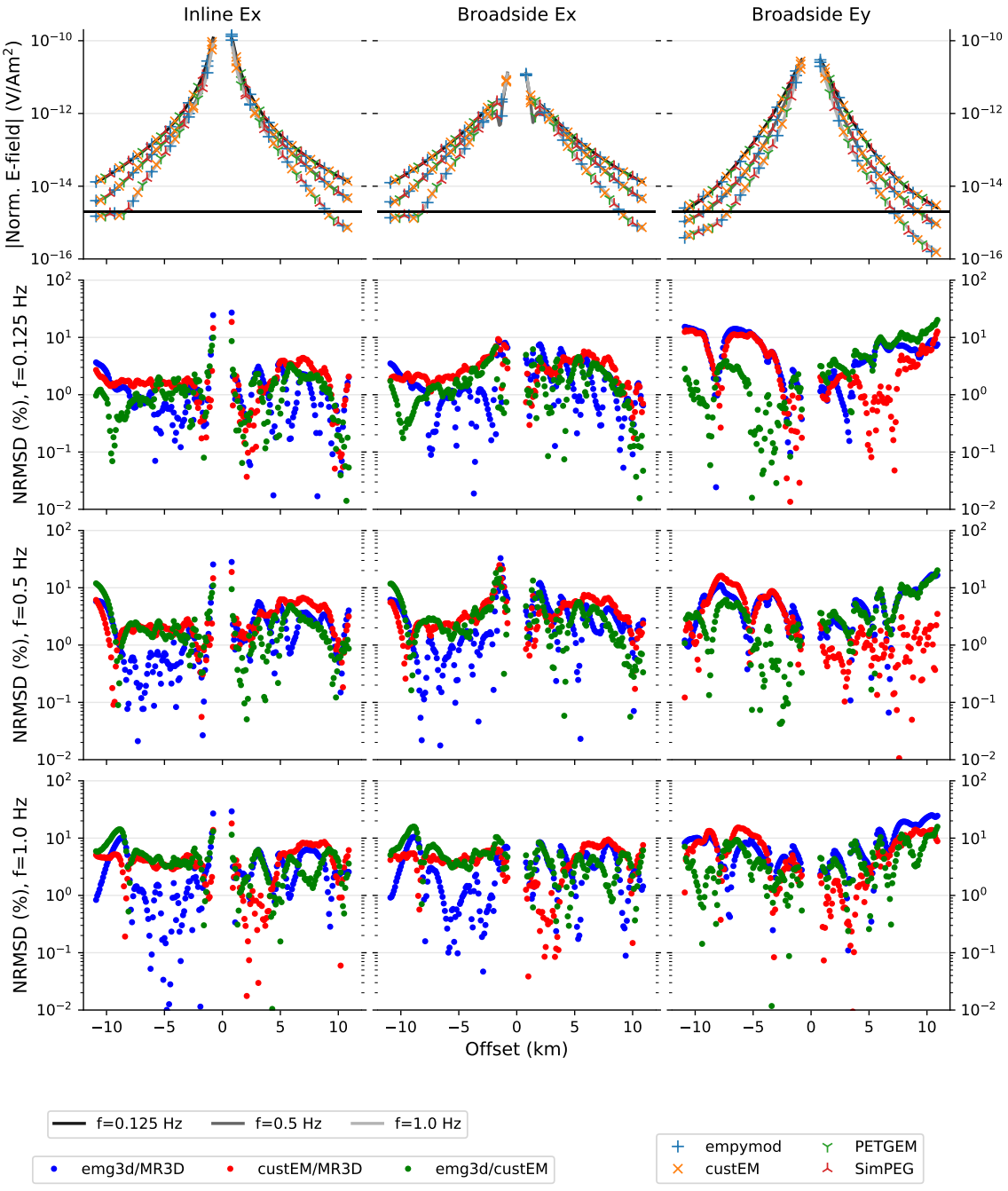

Figure 8: This is one heck of a figure. Just an idea, please make suggestions how to improve it (currently, `PETGEM` data is a copy of `custEM` data, and `SimPEG` data is a copy of `emg3d` data).

DISCUSSION

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

We compared four different, open-source 3D CSEM modellers by computing responses for (a) a layered, anisotropic model without and with three resistive blocks, and (b) a realistic 3D model. The comparison shows that the codes do yield the same responses. As such we could verify the accuracy of these four codes.

This is a very important insight. The possibilities to verify the accuracy of 3D CSEM codes are limited. One can compare them to analytical and semi-analytical solutions, but they only exist for very simple models. For complex models comparisons to other codes is the only possibility. It is therefore really important to have more test models, and importantly, for various scenarios, and that they are easily accessible. Our data should also make it very easy for new codes to have a readily available data set to test against and verify.

A similar exercise has be done for magnetotelluric (MT) 3D modelling and inversion by Miensopust et al. (2013), but we are not aware of anything like this for CSEM data. However, we do hope that more comparisons will emerge in the future, particularly with the current increase in open-source codes as well as open-source data.

The codes under consideration consisted of finite element and finite differences. However, it has to be noted that the chosen models are better suited for finite differences codes. The first model for obvious reasons, as it is a layered background containing blocks, all aligned in Cartesian coordinates. The second model, Marlim R3D, could be an ideal case for FE codes, as it comes with lithological horizons from seismic data. However, the corresponding, published resistivity model uses a Cartesian grid. An extension of this work could be to use a complex model tailored for a FE code, and compare how the FD codes cope with it.

The codes under consideration use direct and indirect solvers, ... blabla say something. Having calculated in each model only one source favours in the runtime comparison the iterative solver and therefore `emg3d`. For the iterative solvers additional sources would come at almost no cost, whereas for `emg3d` it would come at the same cost as the first source. This is something to keep in mind when choosing the appropriate modeller for a problem at hand.

[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.

Finally, the comparison in our study is limited to marine CSEM cases in the frequency-domain. We hope that similar studies will be carried out for land CSEM data and for time-domain data as well.

We see this only as the start. Much more comparisons and examples are required. 3D CSEM modelling is a difficult task, which requires many considerations: It starts with the selection of the right code for the problem; then the meshing is particularly difficult, 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, particularly for shallow and marine cases where the airwave has to be considered. The tricky part is that

even if the result is not correct it may look completely valid. The only options to verify results are (a) by comparing different discretizations, and (b) by comparing different codes.

CONCLUSIONS

[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.

The landscape in 3D CSEM modelling greatly changed in the last five years or so. While before there were only closed-source codes owned by companies or consortia, or codes that you had to obtain from the authors, often without much documentation, there was recently a wave of openly released codes.

What is missing for these emerging codes is a comparison and verification suite, test models beyond the usual, analytical fullspace and halfspace models and the semi-analytical layered Earth models. We hope that our efforts help to the confidence in the available open-source codes...

Also, the runtime and memory consumption comparison together with the meshing differences should help to decide which modeller is used best for which tasks, as there is not one method which is best for all problems.

ACKNOWLEDGMENT

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

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 EU Horizon 2020 research and innovation programme (No 728053); martera.eu.

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**, 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.

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**, 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.

Lebedev, V. I., 1964, Difference analogues of orthogonal decompositions, basic differential operators and some boundary problems of mathematical physics. I: USSR Computa-

tional 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.

Maaø, F. A., 2007, Fast finite-difference time-domain modeling for marine-subsurface electromagnetic problems: Geophysics, **72**, 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 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**, 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: Geo-

physical Journal International, **187**, no. 1, 63–74; doi: 10.1111/j.1365-246X.2011.05127.x.

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 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 mod-

eling of electromagnetic fields in complex structures with inhomogeneous background conductivity: Geophysics, **71**, no. 6, G333–G345; doi: 10.1190/1.2358403.