# PETGEM: A parallel code for 3D CSEM forward modeling using edge finite elements

Octavio Castillo-Reyes[a], Josep de la Puente[a], José María Cela[a]

[a]*Barcelona Supercomputing Center (BSC)*
*c/Jordi Girona, 29. 08034 Barcelona, Spain*

## Abstract

We present the capabilities and results of the Parallel Edge-based Tool for Geophysical Electromagnetic modeling (PETGEM), as well as the physical and numerical foundations upon which it has been developed. PETGEM is an open-source and distributed parallel Python code for fast and highly accurate modeling of 3D marine controlled-source electromagnetic (3D CSEM) problems. We employ the Nédélec Edge Finite Element Method (EFEM) which offers a good trade-off between accuracy and number of degrees of freedom, while naturally supporting unstructured tetrahedral meshes. We have particularised this new modeling tool to the 3D CSEM problem for infinitesimal point dipoles asumming arbitrarily isotropic media for low-frequencies approximations. In order to avoid source-singularities, PETGEM solves the frequency-domain Maxwell's equations of the secondary electric field, and the primary electric field is calculated analytically for homogeneous background media. We assess the PETGEM accuracy using classical tests with known analytical solutions as well as recent published data of real life geological scenarios. This assessment proves that this new modeling tool reproduces expected accurate solutions in the former tests, and its flexibility on realistic 3D electromagnetic problems. Furthermore, an automatic mesh adaptation strategy for a given frequency and specific source position is presented. We also include a scalability study based on fundamental metrics for high-performance computing (HPC) architectures.

*Keywords:* Marine electromagnetics, edge finite element, high-performance computing, numerical solutions

## 1. Introduction

The 3D marine controlled-source electromagnetic method (3D CSEM) is an essential technique in exploration geophysics (Constable, 2010). It is based on using an artificial time-varying EM source and analyzing the response at a series of receiver sites. 3D CSEM has been particularly successful finding hydrocarbon reservoirs in offshore exploration, due to hydrocarbon-filled rocks being very resistive (30–500 Ω·m) and surrounded by moderately conductive background media (0.5–2 Ω·m) (see e.g. Constable and Weiss, 2006;

[*] Corresponding author
*Email address:* octavio.castillo@bsc.es (Octavio Castillo-Reyes)

[1] Octavio Castillo-Reyes contributed to the parallel implementation and its documentation.

[2] Josep de la Puente contributed to the math background and to the meshes generation.

[3] José María Cela carried out simulations and its scalability analysis.

*Postprint of the paper DOI: https://doi.org/10.1016/j.cageo.2018.07.005 published in Computers & Geosciences*

Constable and Srnka, 2007; Key, 2009). As a key part of the electromagnetic exploration workflow, CSEM surveys for 3D media find two applications. Firstly, modeling allows geophysicists to carry out sensitivity analyses and falsate incorrect models. Secondly, its application is being the engine of an inversion scheme, where 3D CSEM data can be used in order to improve the quality of existing subsurface resistivity models. Although quasi-analytical and 2.5D approaches exist for simplistic scenarios, for really complex geological settings and 3D data surveys, full 3D CSEM modeling tools are necessary.

In particular, 3D CSEM modeling algorithms, like other geophysical modeling approaches, should present the following characteristics:

1. Efficiency. In terms of being able to obtain accurate results above the noise level expected in the data in a reasonable time at a reasonable cost.
2. Fidelity. Although academic scenarios might be over-simplified, realistic cases often display very complicated geologies. Bathymetry alone can be fairly complex.
3. Scalability. Schemes must be scalable, easily modifiable and should run equally well on workstations or in compute clusters.

With these three goals in mind, we present a 3D parallel code for 3D CSEM forward modeling in geophysics, namely, PETGEM: Parallel Edge-based Tool for Geophysical Electromagnetic Modeling. We provide this new tool as open source so that it can be used, modified and redistributed freely with the aims of fostering reproducibility and promoting its use for geophysical 3D CSEM modeling. The code works in parallel for distributed memory computers and it is based upon unstructured tetrahedral meshes for a better representation of geometry due to its simple code structure, it should be fairly easy to upgrade and modify in order to suit the needs of different research applications.

The rest of the paper is organized as follows. In Section 2 we give an overview of 3D CSEM modeling theory and its mathematical formulation. In Section 3 we present the analysis and development of our discretisation approach, namely, Nédélec Edge Finite Elements (EFEM). In Section 4 we provide a description of the code design and its parallel features. Through comparison with other state of the art algorithms, in Section 5 we describe the application of the code to realistic 3D CSEM models. Finally, we discuss our numerical results and remark PETGEM flexibility, accuracy and capabilities.

## 2. 3D CSEM forward modeling

In marine 3D CSEM, also referred to as seabed logging (Eidesmo et al., 2002), a deep-towed electric dipole transmitter is used to produce a low-frequency electromagnetic signal which interacts with the electrically conductive Earth and induces eddy currents that become sources of a new electromagnetic signal. The aggregate of both fields is measured by remote receivers placed on the seabed. Since the electromagnetic field at low frequencies, for which displacement currents are negligible, depends mainly on the electric conductivity distribution of the ground, it is possible to detect thin resistive layers beneath the seabed by studying the received signal (Koldan, 2013). Operating frequencies of transmitters in 3D CSEM may range between 0.1 and 10 Hz, although in most studies typical frequencies vary from 0.25 to 3 Hz. At such frequencies, and for typical source-receiver offsets of 10–12 km, the penetration depth of the method can extend to several kilometres below the seabed (Hanif et al., 2011; Koldan, 2013). The main disadvantage of marine 3D CSEM is its relatively low resolution compared to seismic imaging. Therefore, marine 3D CSEM is often used in cooperation with seismic surveying as the latter helps to constrain the geometry of the resistivity model. Marine 3D CSEM is nowadays a well-known geophysical prospecting tool in the industry (see, e.g., Constable, 2006; Boulaenko et al., 2007; Constable and Srnka, 2007; Orange et al., 2009; Constable, 2010).

We consider a quasi-static approximation of the electromagnetic field and ignore the displacement current (Zhdanov, 2009). The magnetic permeability of the Earth can be approximated by its value in the free space (Cai et al., 2017). Therefore, the electromagnetic 3D field in an unbounded domain $\Gamma$ can be obtained by solving Maxwell's equations in their diffusive form (Zhdanov, 2009)

$$\boldsymbol{\nabla} \times \mathbf{E} = i\omega\mu_0\mathbf{H}, \tag{1}$$

$$\boldsymbol{\nabla} \times \mathbf{H} = \mathbf{J}_{\mathrm{s}} + \tilde{\sigma}\mathbf{E}, \tag{2}$$

where we have ommitted the harmonic time dependence $\mathrm{e}^{-i\omega t}$. Above, $\omega$ is the angular frequency, $\mathbf{E}$ the electric field, $\mathbf{H}$ the magnetizing field, $\mu_0$ the free space magnetic permeability, $\mathbf{J}_{\mathrm{s}}$ the distribution of source current, $\tilde{\sigma}\mathbf{E}$ the induced current in the conductive Earth and $\tilde{\sigma}$ the electrical conductivity, which is assumed isotropic for simplicity. After substituting eq. (1) into eq. (2), we obtain

$$\boldsymbol{\nabla} \times \boldsymbol{\nabla} \times \mathbf{E} + i\omega\mu_0\tilde{\sigma}\mathbf{E} = i\omega\mu_0\mathbf{J}_{\mathrm{s}}, \tag{3}$$

which is known as the curl-curl form of the problem (Newman and Alumbaugh, 2002). One technique to solving eq. (3) is to switch to a primary/secondary field formulation in order to capture the rapid change of the primary field without large grid refinement requirements (Cai et al., 2014). In this case, also referred to as scattering formulation (Zhdanov, 2009; Cai et al., 2014), the total electric field $\mathbf{E}$ is obtained as

$$\mathbf{E} = \mathbf{E_p} + \mathbf{E_s}, \tag{4}$$

$$\tilde{\sigma} = \tilde{\sigma_s} + \Delta\tilde{\sigma}, \tag{5}$$

where subscripts $\mathbf{p}$ and $\mathbf{s}$ represent a primary field and secondary field, respectively. For a general layered Earth model, $\mathbf{E_p}$ generated upon the simplified conductivity model $\sigma_s$ can be computed semi-analytically by using Hankel transform filters. For homogeneous background media, the development of $\mathbf{E_p}$ can be found in Appendix A. The equation system that must be solved in this case is

$$\boldsymbol{\nabla} \times \boldsymbol{\nabla} \times \mathbf{E_s} + i\omega\mu_0\tilde{\sigma}\mathbf{E_s} = -i\omega\mu_0\Delta\tilde{\sigma}\mathbf{E_p}. \tag{6}$$

For our modeling purposes we set homogeneous Dirichlet boundary conditions, $\mathbf{E} = 0$ on $\partial\Gamma$. The range of applicability of this conditions can be determined based on the skin depth of the electric field (Puzyrev et al., 2013).

## 3. Edge finite element theory

The most popular discretisation techniques for eq. (3) are the Finite Difference Method (FDM) and the Finite Element Method (FEM). FDM is nowadays the most widely employed discretisation scheme (see, e.g., Alumbaugh et al., 1996; Newman and Alumbaugh, 2002). There exist many successful FDM implementations, but the most practical and highly efficient parallel code was developed by Alumbaugh et al. (1996) and improved afterwards by some collaborators. However, the main disadvantage of FDM is its dependance on structured grids, which reduces or limits its accuracy and usability in cases where irregular and complicated geology has a significant influence on measurements. For instance, an imprecise representation of the seabed bathymetry could produce artefacts in images that can lead to false interpretation (Koldan, 2013).

On the other hand, FEM supports completely unstructured meshes as well as mesh refinement, which enables the representation of complex geometries and thus improves the solution accuracy. Nevertheless, FEM is still not as widely applied as FDM and a major obstacle for a wider adoption is that nodal FEM does not correctly take into account all the physical aspects of the vector field functions. In fact, there are three main problems when nodal-based finite elements, obtained by interpolating the nodal values, are employed to represent vector fields (electric or magnetic). The first one is the occurrence of spurious solutions or non-physical solutions, which is generally attributed to the lack of enforcement of the divergence condition (Beck and Hiptmair, 1999; Jin, 2002; Monk, 2003; Hiptmair, 2015). The second one is the inconvenience of imposing boundary conditions at material interfaces as well as at conducting surfaces (Hiptmair, 2015). Finally, the third problem is the difficulty on treating conducting and dielectric edges and corners due to field singularities associated with these structures (Monk, 2003). Consequently, most of the researchers who have employed FEM for 3D electromagnetic forward modeling have been primarily focused on overcoming these problems, as well as on solving other physical and numerical challenges, in order to obtain a proper and accurate numerical solution, leaving aside the performance of the codes (Koldan, 2013).

Vector basis functions exist that assign vector degrees of freedom (DOF) to the edges rather than to the nodes of each element, which can be used to build so-called Edge Finite elements. In the case of tetrahedral elements and first-order polynomials, a divergence-free basis exists that can be used to build a particular type of edge finite elements called Nédélec elements. Since Nédélec elements belong to the class of finite elements method, they are very well suited for honoring complex geometrical structures such as realistic geology or bathymetry. Furthermore, they offer a good trade-off between accuracy and number of DOF, i.e. size of the problem.

For the computation of $\mathbf{E}$ in eq. (3), or alternatively $\mathbf{E_s}$ in eq. (6), we have implemented tetrahedral EFEM of the lowest order which uses vector basis functions defined on the edges of the corresponding elements. These basis functions are divergence-free but not curl-free (Jin, 2002). Thus, EFEM naturally ensures tangential continuity and allows normal discontinuity of $\mathbf{E}$ (or $\mathbf{E}_s$) at material interfaces. Figure 1 shows the tetrahedral Nédélec elements together with their node and edge indexing, that we have implemented. In our approach,

| Edge $i$ | Node $i_1$ | Node $i_2$ |
|---|---|---|
| 1 | 1 | 2 |
| 2 | 1 | 3 |
| 3 | 1 | 4 |
| 4 | 2 | 3 |
| 5 | 4 | 2 |
| 6 | 3 | 4 |

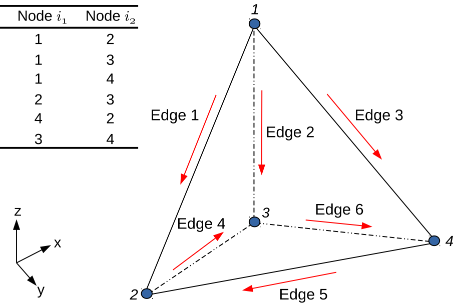

Figure 1: Tetrahedral Nédélec edge element with node/edge indexing that we have implemented in PETGEM.

we assign the tangential component of the electric field to the mesh edges. Therefore, all components of the electric field at a point $(x, y, z)$ located inside a tetrahedral element $e$ can be obtained as follows

$$\mathbf{E}^e(x,y,z) = \sum_{i=1}^{6} \mathbf{N}_i^e(x,y,z) E_i^e, \tag{7}$$

where $\mathbf{N}_i^e$ are the vector basis functions associated to each edge $i$, and $E_i^e$ their respective DOF (see Appendix  B for the mathematical development).

In the following, we show explicitly the development for solving $\mathbf{E_s}$. By substituting eq. (7) into eq. (3), and using Galerkin's approach, the weak form of the original differential equation becomes

$$Q_i = \int_\Omega \mathbf{N}_i \cdot [\boldsymbol{\nabla} \times \boldsymbol{\nabla} \times \mathbf{E_s} - i\omega\mu_0\tilde{\sigma}\mathbf{E_s} + i\omega\mu_0\Delta\tilde{\sigma}\mathbf{E_p}]dV. \tag{8}$$

The compact discretized form of eq. (8) is obtained after applying the Green's theorem

$$[K_{jk}^e + i\omega\tilde{\sigma}_e M_{jk}^e] \cdot \{E_{sk}\} = -i\omega\mu\Delta\tilde{\sigma}_e R_k^e, \tag{9}$$

where $K^e$ and $M^e$ are the elemental stiffness and mass matrices (see Appendix  B for mathematical details). These terms can be calculated analytically or numerically (Jin, 2002) whereas $R_k^e$ is the right hand side that requires numerical integration.

## 4. PETGEM: code work-flow

PETGEM is a Python 3 code for the scalable solution of 3D CSEM on tetrahedral meshes, as these are the easiest to scale-up to very large meshes of arbitrary shape. It is written mostly in Python 3 and relies on the scientific Python software stack with use of *mpi4py*  (Dalcín et al., 2011) and *petsc4py*  (Dalcín et al., 2011) packages for parallel computations. Other scientific Python packages used include: *Numpy* for efficient array manipulation and *Scipy* algorithms for numerical computations. PETGEM allow users the simulation of electromagnetic fields in realistic 3D CSEM on distributed-memory HPC platforms. Among others, the key drivers for the PETGEM development are the following:

1. Fill the relative scarcity of robust edge-based codes for 3D CSEM to reduce ambiguities in data interpretation for hydrocarbon exploration.

2. Model realistic scenarios that support 3D CSEM simulations in structurally complex geometries with a good trade-off between accuracy and number of DOF.
3. Provide synthetic results which can then compared to measured data.
4. Improve the degree integration of HPC using Python, EFEM, and geophysical methods such as 3D CSEM at realistic-scale.

Commonly the 3D CSEM is composed of four main tasks: discretisation of the geometry, elemental matrices computation, and global system assembly, solving the resulting system and post-processing the solution. The problem decomposition into independent modules is important because each region make use of methods that belong to different branches of mathematics. For instance, the design of algorithms for 3D meshing and iterative solvers for large scale modeling require knowledge going beyond the scope of this work. Therefore, this paper rely on well-known tools for domain discretisation and solving systems of linear equations and focuses on the kernel of 3D CSEM, namely, the core of PETGEM. An outline of the overall PETGEM work-flow is depicted in Figure 2. Furthermore, in Figure 3 a snapshot of the primary groups of modules in the code is given. In order to solve a 3D CSEM case, PETGEM requires a parameters file with all information associated with the model under consideration. For the sake of simplicity and in order to avoid a specific parser, the PETGEM parameters file is defined as a Python dictionary. This parameters file is divided into four sections: physical parameters (frequency, source position, source current, source length, conductivity model, background conductivity), mesh information (file path of nodal spatial coordinates, nodal element connectivity, edge element connectivity, edges nodes connectivity, and sparsity structure for *PETSc* matrix allocation (Balay et al., 2016)), solver parameters (solver and preconditioner type, tolerance, maximum number of iterations), results information (receivers position file path). Regarding discretisation formats, PETGEM is capable of importing tetrahedral finite element mesh files generated by *Gmsh* (Geuzaine and Remacle, 2008).

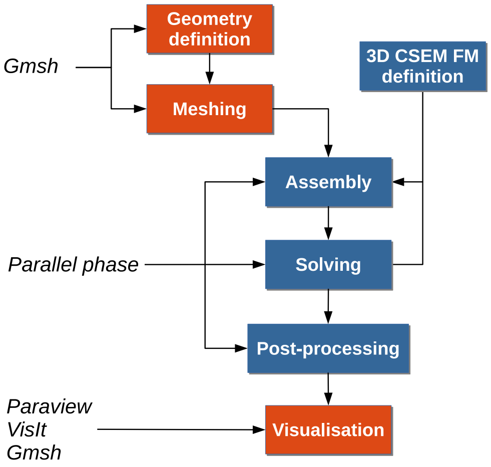

Figure 2: Outline of the overall PETGEM work-flow. On shared-memory architectures, the parallel phase is based on the Multiprocessing package. On the other hand, the *petsc4py* and *mpi4py* packages are used to provide parallel support on distributed-memory architectures.

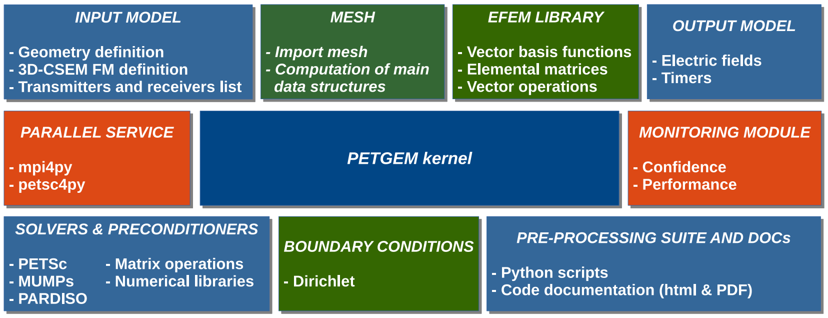

Figure 3: Upper view of PETGEM software stack.

Notice that geometry and meshing modules are independent of main PETGEM work-flow.

Once the parameter file is defined, PETGEM imports it and starts the assembly of the global linear system. Several processes of the kernel can be spawned, each responsible for its own subdomain, so that the whole domain is covered. Each process then assembles its local contributions to the global linear system that is solved. For this purpose, PETGEM use the *PETSc* library and its large collection of data structures and parallel iterative solvers, that can be used in Python through the *petsc4py* and *mpi4py* packages.

Once a solution of the 3D CSEM has been obtained, it should be post-processed by using a visualization program. PETGEM does not the visualization by itself, but it generates output files (*PETSc*, *MATLAB*, and *ASCII* format) with the electric field responses that can be easily imported by external visualization tools.

PETGEM is written in Python 3 because it is open source and functional on a wide number of platforms, including HPC environments. Furthermore, it uses a high level and very expressive language. The code structure is modular, simple, and flexible which allows exploiting not just PETGEM modules but also third party libraries. The HPC goal of this code involves using cutting-edge architectures. To that goal, the code is implemented in current state-of-the-art platforms such as Intel Skylake, Intel Haswell and Intel Xeon Phi processors, which offer high performance, flexibility, and power efficiency. Nevertheless, PETGEM support older architectures such as SandyBridge, for the sake of usability and to be able to compare performance.

## 5. Application scenarios

In order to verify our EFEM formulation and study the PETGEM capabilities, we simulate 3D CSEM forward modeling over different scenarios. These experiments have been performed on version III of the *Marenostrum* supercomputer at *BSC*. Marenostrum (MN3) supercomputer based on Intel SandyBridge processors, iDataPlex Compute Racks, a Linux Operating System and an Infiniband interconnection. It has 48 896 Intel SandyBridge-EP E52670 cores at 2.6 GHz grouped into 3 056 computing nodes, 103.5 TB of main memory (128 nodes with 128 Gb, 128 nodes 64 Gb, and 2 880 nodes with 32 Gb) as well as 1.9 PB of GPFS disk storage. Its peak perfomance is 1.1 Petaflops. Each computing node has two 8-core Intel Xeon processors E5-2670 with a frequency of 2.6 GHz and 20 MB cache memory.

Regarding numerical accuracy, we defined a relative misfit criteria for the electric responses in terms of its amplitude $|\mathbf{E}|$ and phase $\Phi$ as follows

$$\epsilon = \left| \frac{(S^{ref} - S^{efem})}{S^{ref}} \right| \times 100, \tag{10}$$

where $S^{ref}$ and $S^{efem}$ are the reference and EFEM solutions, respectively.

### *5.1. Canonical model of an off-shore hydrocarbon reservoir*

In the first test, we validate our approach and PETGEM against the quasi-analytical results of the canonical model by Constable and Weiss (2006). Our computational domain is defined by a $[-1, 4.5] \times [0, 3.5] \times [0, 3.5]$ km composed by four layers with different thickness and conductivity ($\sigma$): 1 000 m thick seawater (3.3 S/m), 1000 m thick sediments (1 S/m), 100 m thick oil (0.01 S/m), and 1400 m thick sediments (1 S/m). Figure 4 shows a 3D view of the model with its unstructured tetrahedral mesh for the halfspace $y > 1\,750$ m, with the color scale representing the electrical conductivity $\sigma$ for each layer. For this model, we use a 1 Hz $x$-directed dipole source as in Castillo et al. (2015), which is located at $x = 1\,750$ m, $y = 1\,750$ m, and $z = 975$ m. The receivers are placed in-line to the source position and along its orientation, directly above the seafloor ($z = 990$) with spacing of 58 m. For this case, we computed the electric fields using a tetrahedral mesh with $\approx 2$ millions of DOF. Also, a *PETSc* implementation of the GMRES solver has been used to solve the system of equations.

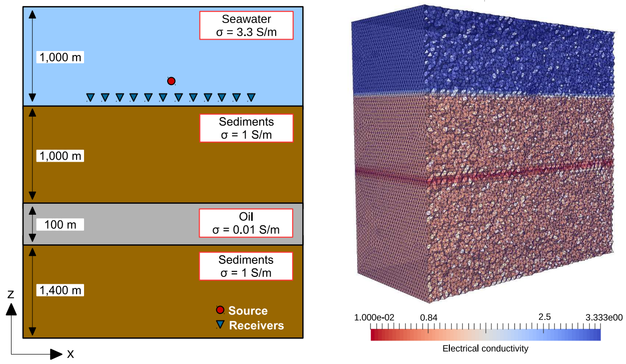

Figure 4: In-line canonical off-shore hydrocarbon model with its unstructured tetrahedral mesh for $y > 1\,750$ m. The color scale represents the electrical conductivity $\sigma$ for each layer.

Figure 5 shows the amplitude and phase comparison of $\mathbf{E}_x$ measurements between PETGEM and those obtained with the DIPOLE1D tool (Key, 2009). The top panel depicts the amplitude $|\mathbf{E}_x|$ of the electric responses along the receivers line, where is easy to see that PETGEM results are in good agreement with the reference data. Additionally, the phase ratio $\Phi_x$ comparison is presented in the bottom panel of Figure 5. Based on this verification, we computed the amplitude $\epsilon(|\mathbf{E}_x|)$ and phase $\epsilon(\Phi_x)$ misfits for inline receivers. The results are shown in Figure 6. In both cases, a good overall agreement is observed ($< 1\%$ of average relative misfit). Hence, the PETGEM solution shows a good agreement with the quasi-analytical results in canonical models.

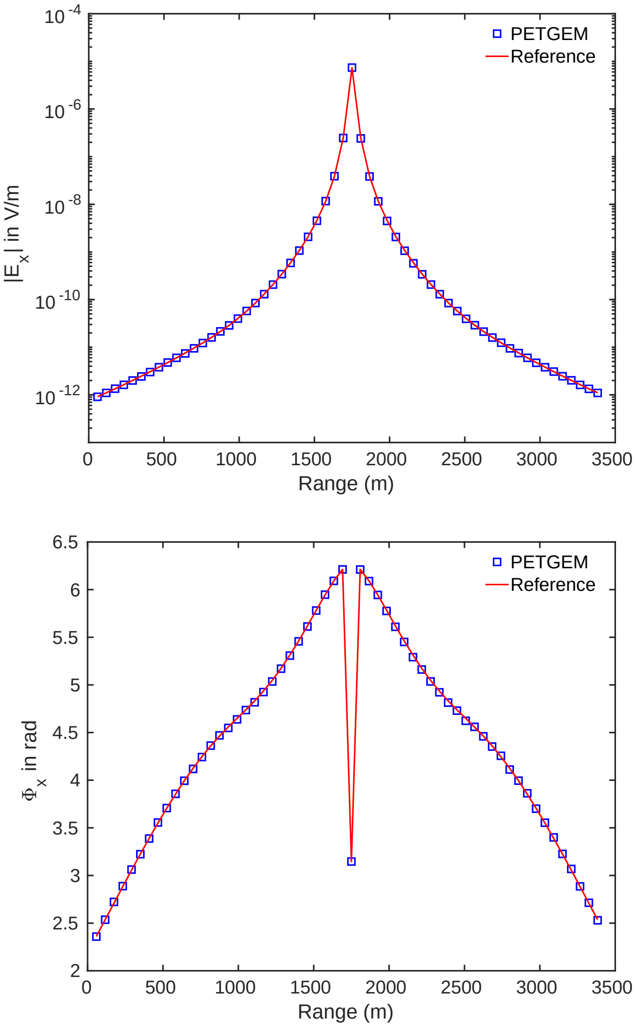

Figure 5: Comparison of inline electric field responses obtained from PETGEM and DIPOLE1D (Key, 2009). Amplitude $|\mathbf{E}_x|$ and phase $\Phi_x$ are plotted at top and bottom, respectively.

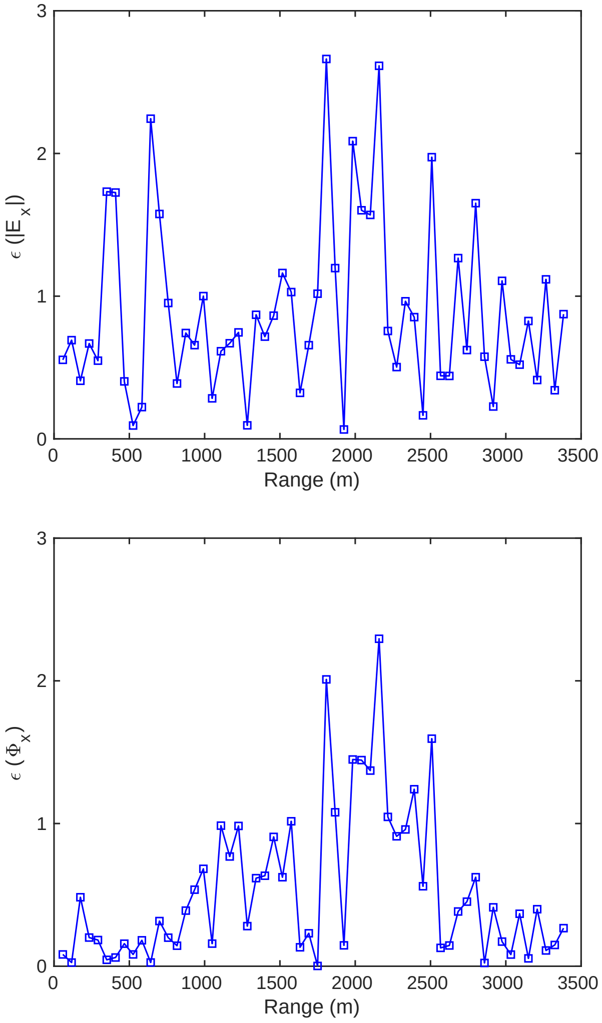

Figure 6: Amplitude misfit $\epsilon(|\mathbf{E}_x|)$ and phase misfit $\epsilon(\Phi_x)$ of $\mathbf{E}_x$ shown in Figure 5. The $\epsilon(|\mathbf{E}_x|)$ and $\epsilon(\Phi_x)$ misfits are plotted at top and bottom, respectively. In both cases, an average relative misfit $< 1\%$ is observed.

### *5.2. CSEM modeling with bathymetry*

The second test involves a 3D CSEM modeling with bathymetry. This model is especially interesting because a primary advantage of the FEM/EFEM over other techniques like the Finite Diference Method (FDM) is the precise representations of arbitrarily complex geological structures such as seafloor bathymetry, without critically increasing problem size. Furthermore, if not taken into account, bathymetry effects can produce large anomalies on the measured electric fields.

The reference dataset of this model was provided by Chung et al. (2014). Additionally, a nodal FEM solution of this modeling case is described in Um et al. (2013). The model consists of 26.6 km of air layer (1e-6 S/m), 2.4 km of seawater (3.3 S/m), and 25.0 km of sediments layer (1.4286 S/m). The computational domain is defined by a $[0, 30] \times [0, 32] \times [0, 54]$ km box as shown in Figure 7. The model has been discretized into 1 542 514 tetrahedral elements, resulting in 258 716 nodes and 1 814 928 DOFs. For this simulation, we use three x-oriented electric dipole sources, with a moment of 200 A·m and frequency of 0.25 Hz, located at $[-5, 0, -2.086]$ km, $[0, 0, -2.096]$ km, and $[5, 0, -2.001]$ km. The 41 receivers are placed in-line to the source positions and along its orientation, directly above the seafloor, as shown in Figure  8. The bathymetry in the area is very rough, and adequate spatial sampling can only be achieved by allowing receiver deployment in slopes or trenches. Several steep trenches go through the area from the shallow eastern part to the deeper western part. For this model, we use the multifrontal parallel solver MUMPS (Amestoy et al., 2006) to solve the resulting system of equations. For each source, Figures 9 and 10 compare the amplitude $|\mathbf{E}_x|$ and phase $\Phi_x$ of electric fields obtained from our modeling tool with those produced by Chung et al. (2014). One can clearly see a good agreement with the reference.

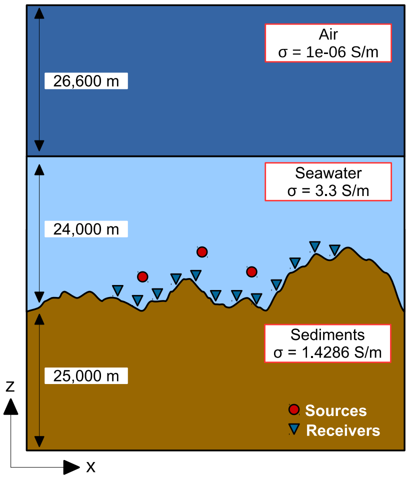

Figure 7: Bathymetry model used for simulating 3D CSEM surveys. The model was discretized into $1,542,514$ tetrahedral elements, resulting in $1,814,928$ unknowns. The dataset of this 3D CSEM model was provided by Chung et al. (2014).

As in the previous model, we compare PETGEM solution against the reference in terms of misfit ratios. Figures 11 and 12 present the amplitude $\epsilon(|\mathbf{E}_x|)$ and phase $\epsilon(\Phi_x)$ misfits, respectively. The overall agreement is reasonable ($< 2\%$ of average relative misfits) when considering that both results are subject to different

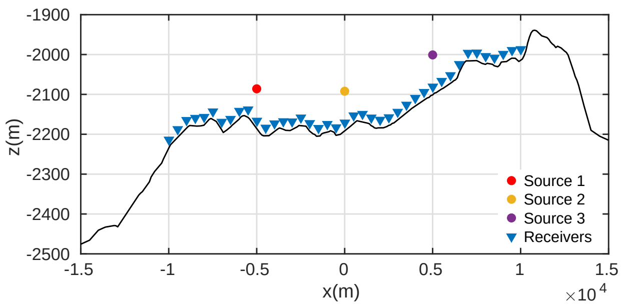

Figure 8: Cross-sectional view of the bathymetry model in Figure 7. Three 0.25 Hz $x-$directed dipoles are located at $[-5, 0, -2.086]$ km, $[0, 0, -2.096]$ km, and $[5, 0, -2.001]$ km. The 41 receivers are placed in-line to the source position and alongs its orientation, directly above the seafloor.

numerical inaccuracy comming from discretisation method, mesh quality, among others. On the other

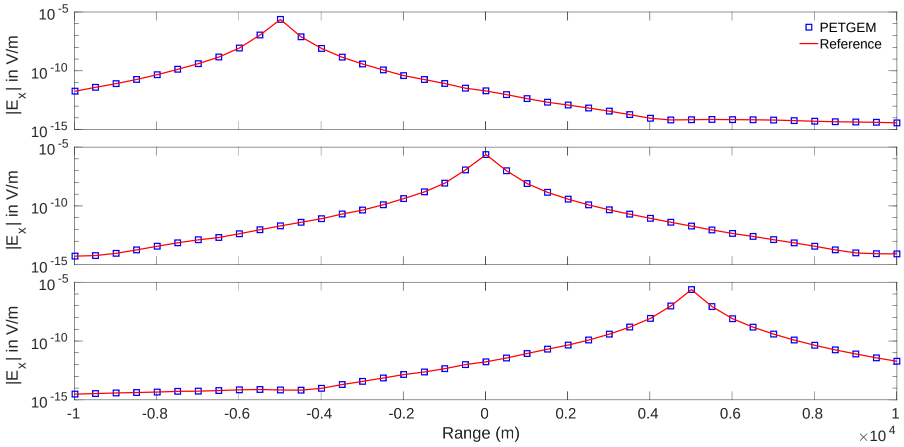

Figure 9: Amplitude $|\mathbf{E}_x|$ comparison between PETGEM and the results of Chung et al. (2014). The three 0.25 Hz sources located at $[-5, 0, -2.086]$ km, $[0, 0, -2.096]$ km, and $[5, 0, -2.001]$ km are plotted on the top, middle and bottom, respectively.

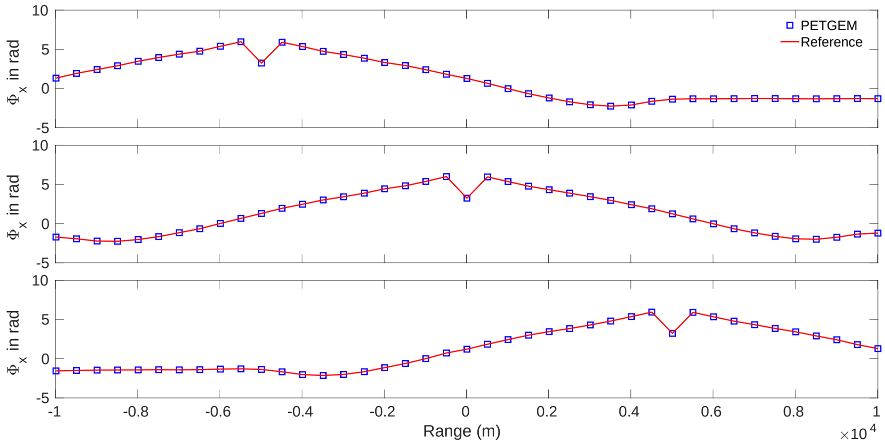

Figure 10: Phase $\Phi_x$ comparison between PETGEM and the results of Chung et al. (2014). The three 0.25 Hz sources located at $[-5, 0, -2.086]$ km, $[0, 0, -2.096]$ km and $[5, 0, -2.001]$ km are plotted on the top, middle, and bottom, respectively.

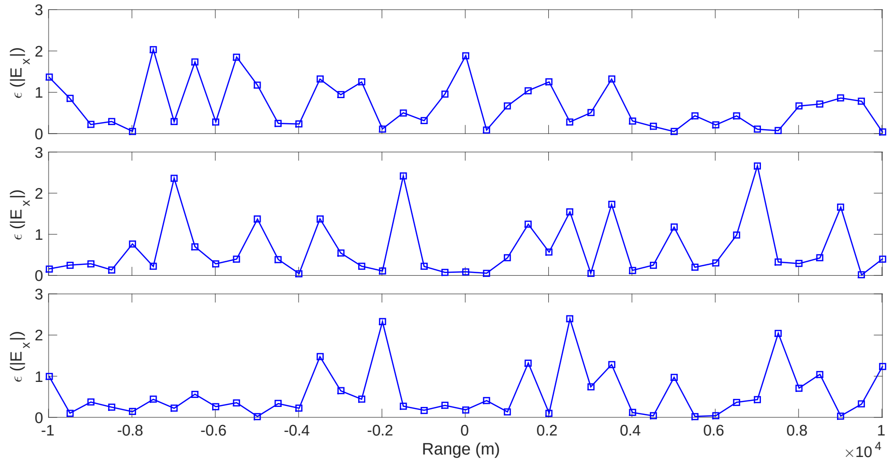

Figure 11: Amplitude misfits $\epsilon(|\mathbf{E}_x|)$ of Figure 9. The $\epsilon(|\mathbf{E}_x|)$ for the three 0.25 Hz sources located at $[-5, 0, -2.086]$ km, $[0, 0, -2.096]$ km, and $[5, 0, -2.001]$ km are plotted on the top, middle, and bottom, respectively.

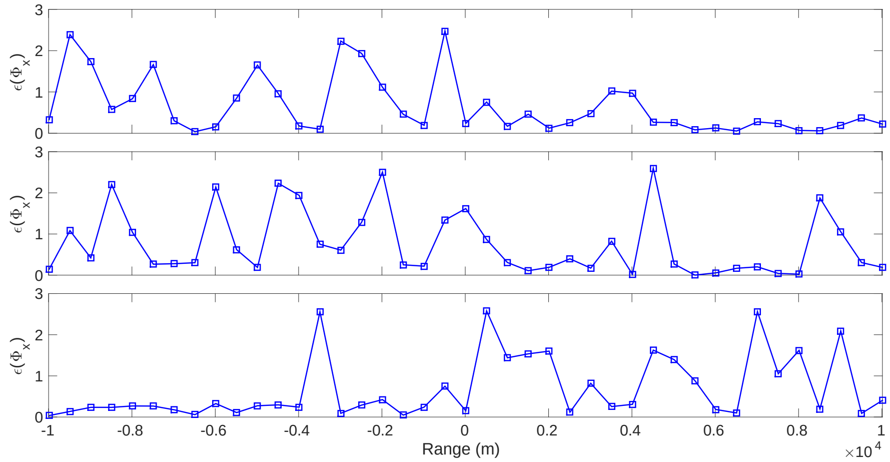

Figure 12: Phase misfits $\epsilon(\Phi_x)$ of Figure 10. The $\epsilon(\Phi_x)$ for the three 0.25 Hz sources located at $[-5, 0, -2.086]$ km, $[0, 0, -2.096]$ km, and $[5, 0, -2.001]$ km are plotted on the top, middle, and bottom, respectively.

hand, in Chung et al. (2014) the solution for the model was computed using EFEM over a hexahedral mesh with $265 \times 64 \times 73$ cells and 3 796 596 DOFs. Furthermore, the authors reported that the computation spent 3 872 seconds and required less than 88.5 Gb of memory on one node equipped with two Intel quad-core Xeon processors (resulting in eight cores) at 2.53 GHz and sharing 96 Gb of memory. In order to compare these numbers we executed our simulation with 8 MPI tasks. The mean runtime for this model is 2 439 seconds and required less than 64 Gb. This means a good efficiency of PETGEM when considering the difference between the DOF of each model. Again, PETGEM results show a good overall agreement with the reference solution.

### *5.3. Automatic mesh adaptation*

Nowadays, in the field of numerical simulations based on FEM and EFEM, automatic mesh adaptation has largely proved its efficiency for improving the accuracy of the numerical solution and capturing the behavior of physical phenomena by exploiting local mesh refinement. In principle, this technique allowing substantially reducing the number of DOF, thus favorably impacting CPU times, and to achieving a desired accuracy on computed solutions. Although iterative solvers are rather efficient even on oversampled meshes, memory requirements can be reduced if the computational mesh is adapted to the source location and to the frequency. However, when the source and receivers depth and the bathymetry are varying, it will be too cumbersome and impractical to ask the user to define a mesh per source/receivers and per frequency.

This test is devoted to the analysis of the automatic mesh adaptation approach developed by Plessix et al. (2007). This approach ensures, for a given frequency and a given source position, that the computational domain is consistent with the discretization of the EM equations. Its core is based on the skin-depth ($\delta$), defined as the effective depth of penetration of EM energy in a conducting medium, where the amplitude of a plane wave in a whole space has been attenuated to $1/e$ or 37% (Sheriff, 2002). Its formal definition is the following

$$\delta = \sqrt{\frac{2}{\mu_0 \omega \sigma}} \approx 503 \sqrt{\frac{1}{f\sigma}}, \tag{11}$$

where $\mu_0$ is the free space magnetic permeability (H/m), $\omega$ is the angular frequency (rad), $\sigma$ is the electric conductivity (S/m), and $f$ is the frequency (Hz). According to the formulation of Plessix et al. (2007), equation (11) gives a rule to automatically determine the spacing $d_\delta(f)$ at a frequency $f$

$$d_\delta(f) = \frac{\delta_{\min}(f)}{r_\delta}, \tag{12}$$

where $\delta_{\min}$ is the minimum skin-depth and $r_\delta$ is a number between two and three. In 3D CSEM surveys, $\delta_{\min}$ occurs in the water layer where $\sigma \approx 3.3$ S/m and $\delta_{\min} \approx 275/\sqrt{f}$. In order to obtain better approximations around source and receivers, we define the spacing $d_{\mathrm{s}}$ as follows

$$d_{\mathrm{s}} = \min\left(\frac{L_{\mathrm{s}}}{r_{\mathrm{s}}}, d_\delta(f)\right), \tag{13}$$

where $L_{\mathrm{s}}$ is the source dipole length and $r_{\mathrm{s}}$ a number between ten and fifteen. The value for $r_{\mathrm{s}}$ is different to those described by Plessix et al. (2007) (between two and four) because the authors used a finite-integration approach. However, in the test described below, we observe that this difference does not imply a significant increase in the computational cost.

We estimate the mesh dimensions from the average skin-depth ($\delta_{ave}$), that generally corresponds to the skin-depth in the sediment areas, i.e., $\delta_{ave} \approx 503/\sqrt{f}$ for a conductivity of 0.01 S/m. The computational domain, decomposed into a core domain and extra boundary layers, is defined as follows

$$[x_{\mathrm{s}} - r_x\delta_{ave}, x_{\mathrm{s}} + r_x\delta_{ave}] \times [y_{\mathrm{s}} - r_y\delta_{ave}, y_{\mathrm{s}} + r_y\delta_{ave}] \times [z_{air}, z_{\mathrm{s}} + r_z\delta_{ave}], \tag{14}$$

where $r_x$ and $r_y$ are numbers between four and eight (depending on the location of the receivers), $r_z$ a number around four, $z_{air}$ the depth of the air-water interface, and $[x_{\mathrm{s}}, y_{\mathrm{s}}, z_{\mathrm{s}}]$ the source position. In order to reduce boundary reflections (as can be seen in Section 5.1), we add extra boundary layers with a thickness of $t_b = r_b \delta_{ave}$, where $r_b$ is a number around four. The values of $r_x$, $r_y$, $r_z$, and $r_b$ where chosen in the same way as by Plessix et al. (2007), where reflections of EM fields are reduced by 98% over four skin-depths, and by 99.9% over eight skin-depths.

For this test, the core of the computational domain is centered at $x_{\mathrm{s}}$ and the number of points is limited by the following power-law stretching

$$x_i = x_{i-1} + \min(s_c^i d_{\mathrm{s}}, d_\delta(f)), \tag{15}$$

where $s_c$ is the stretching parameter equal to 1.04. Similarly, the boundary layers were stretched with a power-law defined as

$$x_i = x_{i-1} + s_b(x_{i-1} - x_{i-2}), \tag{16}$$

where $s_b$ is equal to 1.1. Finally, we have defined a constant conductivity value for each element of the computational domain.

To evaluate whether the aforementioned approach is satisfactory to model 3D CSEM surveys, we carried out several PETGEM simulations based on the model described in Section 5.1 for the following frequencies: 0.25 Hz, 0.5 Hz, 0.75 Hz, 1 Hz, 1.25 Hz, 1.5 Hz, 1.75 Hz, and 2 Hz. The strategy was as follows: The geometry of the domain, including the interfaces and separation between core and extra domains is generated beforehand in *Gmsh* (Geuzaine and Remacle, 2008). Then we set up an automatic meshing script that, given an input frequency, applies the discretization rules aformentioned to define typical spacings to each region. By running the script, *Gmsh* generates a new mesh in a completely automated way. For each frequency, the computational mesh was controlled by the paramters $r_\delta = 3$, $r_{\mathrm{s}} = 13$, $r_x = r_y = 8$, $r_z = 4$, and $r_b = 4$. A summary of the resulting meshes from this process is described in Table 1. All tests have been solved with a single process using the Symmetric Quasi-Minimal Residual (SQMR) solver with a Successive Over-relaxation (SOR) method as preconditioner. After executing each test separately (adapted tests) and aiming to investigate the effect of an oversampled mesh, we solved each frequency using a single mesh (i.e. the finest mesh which is that adapted to 2 Hz labeled H in Table 1). Figures 13 and 14 compare the amplitude $|\mathbf{E}_x|$ and phase $\Phi_x$ components of electric fields between PETGEM and those obtained by Constable and Weiss (2006). Here, the overall agreement is better than those presented in Section 5.1. However, simulation results with the oversampled mesh show higher differences with respect to the reference. Since these discrepancies have greater presence in the boundaries and around to source position, this accuracy loss is related to the quality of the elements on such mesh regions.

We computed the amplitude ($\epsilon(|\mathbf{E}_x|)$) and phase ($\epsilon(\Phi_x)$) misfits of the electric responses obtained on both meshes, adapted and oversampled. The $\epsilon(|\mathbf{E}_x|)$ and $\epsilon(\Phi_x)$ misfits are shown in Figures 15 and 16, respectively. One can clearly see the positive impact of frequency adapted meshes, e.g. the misfits associated to modeling

Table 1: Summary of resulting meshes based on automatic mesh adaptation.

| Label | Frequency (Hz) | Elements | Nodes | Edges | DOF |
|---|---|---|---|---|---|
| A | 0.25 | 717 572 | 122 461 | 852 100 | 815 896 |
| B | 0.5 | 811 710 | 137 708 | 961 485 | 925 281 |
| C | 0.75 | 909 821 | 153 579 | 1 075 467 | 1 039 263 |
| D | 1 | 1 029 484 | 172 868 | 1 214 419 | 1 178 215 |
| E | 1.25 | 1 156 378 | 193 420 | 1 361 865 | 1 325 661 |
| F | 1.5 | 1 298 556 | 216 346 | 1 526 969 | 1 490 765 |
| G | 1.75 | 1 455 924 | 241 750 | 1 709 741 | 1 673 537 |
| H | 2 | 1 636 021 | 270 745 | 1 918 833 | 1 882 629 |

results at 2 Hz are smaller than the ones shown in Figure 6. The responses are very similar, the amplitude misfits are smaller than 2%, and the phase misfits are around 0.3 rad. Notice that the results shown here, are a direct consequence of using unstructured tetrahedral meshes, which can be generated fully automatically given a proper geometry and spacing rules. This is different from conforming hexahedral grids that often require manual interaction to properly honor arbitrary spacings and geometries (Owen, 1998). These numerical examples depict the usefulness of frequency-adapted meshing for 3D CSEM design and scenario studies. To illustrate the efficiency of this technique, we have measured the times for the assembly and solving tasks in PETGEM. Table 2 lists the number of iterations, assembling and solution times on a single CPU. Solutions on oversampled meshes requires much more iterations than tests with adapted meshes and, as a consequence, need considerably more CPU time. On average, the adapted scheme is about four times faster in this set of experiments. In Figure 17, assembly and solution times of both test sets are displayed.

Table 2: Summary of results for mesh-adapted tests (M.A.) and oversampled mesh tests (O.S.). Times are expressed in minutes.

| Frequency (Hz) | Assembly | | Solver | | Iterations | |
|---|---|---|---|---|---|---|
| | M.A. | O.S. | M.A. | O.S. | M.A. | O.S. |
| 0.25 | 1.77 | 11.8 | 17.52 | 222.4 | 5 950 | 11 382 |
| 0.5 | 2.99 | 11.9 | 35.78 | 222.3 | 6 630 | 12 519 |
| 0.75 | 4.49 | 11.8 | 45.35 | 213.7 | 5 625 | 9 685 |
| 1 | 5.72 | 11.6 | 91.70 | 206.7 | 8 245 | 9 275 |
| 1.25 | 6.95 | 11.8 | 95.70 | 203.4 | 6 825 | 8 670 |
| 1.5 | 8.32 | 11.6 | 104.71 | 174.9 | 6 145 | 7 540 |
| 1.75 | 10.38 | 11.8 | 137.02 | 161.1 | 7 290 | 7 230 |
| 2 | 12.52 | 11.8 | 155.76 | 153.6 | 6 935 | 6 935 |

Previous results show the relevance of mesh adaptation for survey design in the 3D CSEM context.

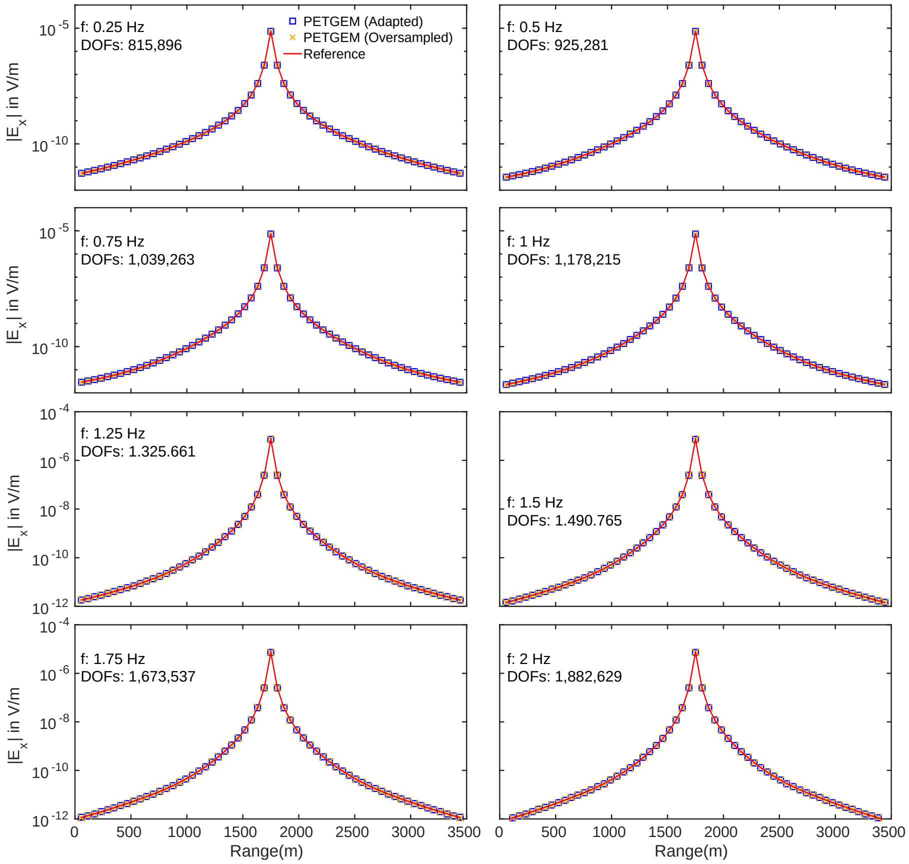

Figure 13: Comparison of inline electric field responses between PETGEM and those obtained with DIPOLE1D (Key, 2009). Here, the amplitude $|\mathbf{E}_x|$ is plotted for each frequency.

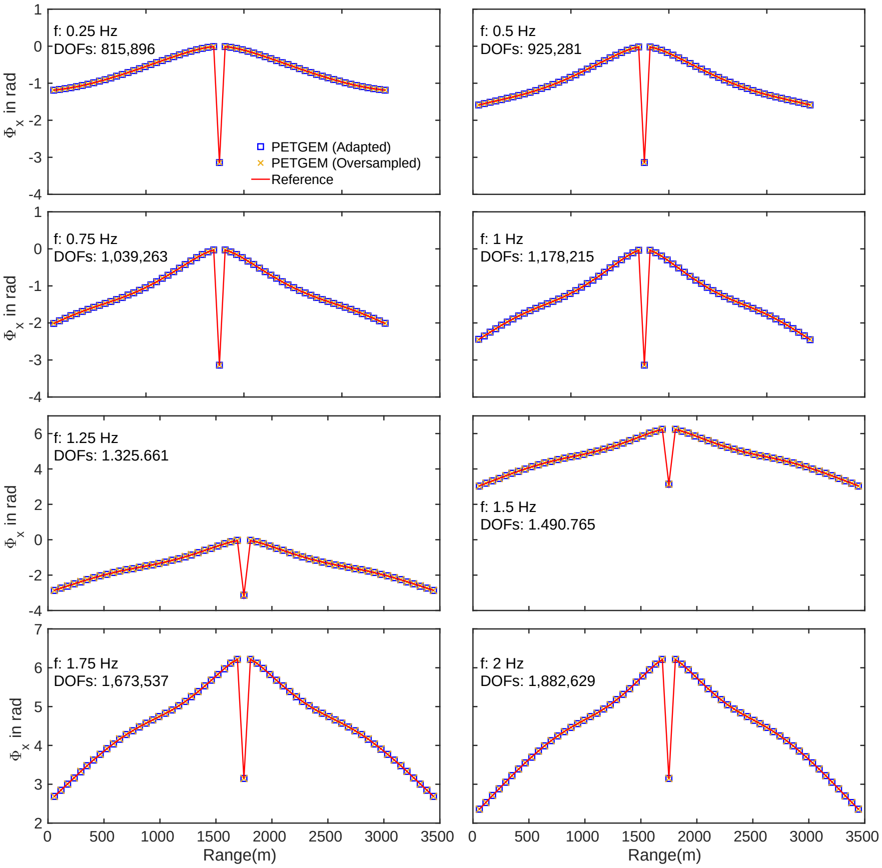

Figure 14: Comparison of inline electric field responses between PETGEM and those obtained with DIPOLE1D (Key, 2009). Here, the phase $\Phi_x$ is plotted for each frequency.

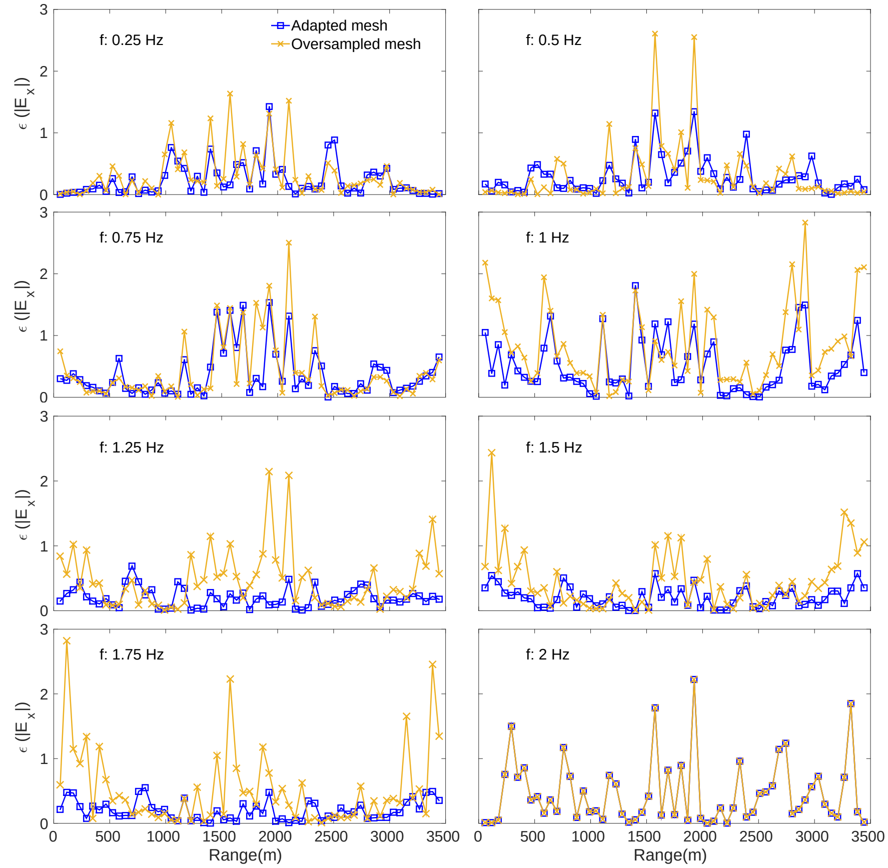

Figure 15: Amplitude misfits $\epsilon(|\mathbf{E}_x|)$ for Figure 13. An average relative misfit $< 2\%$ is observed.

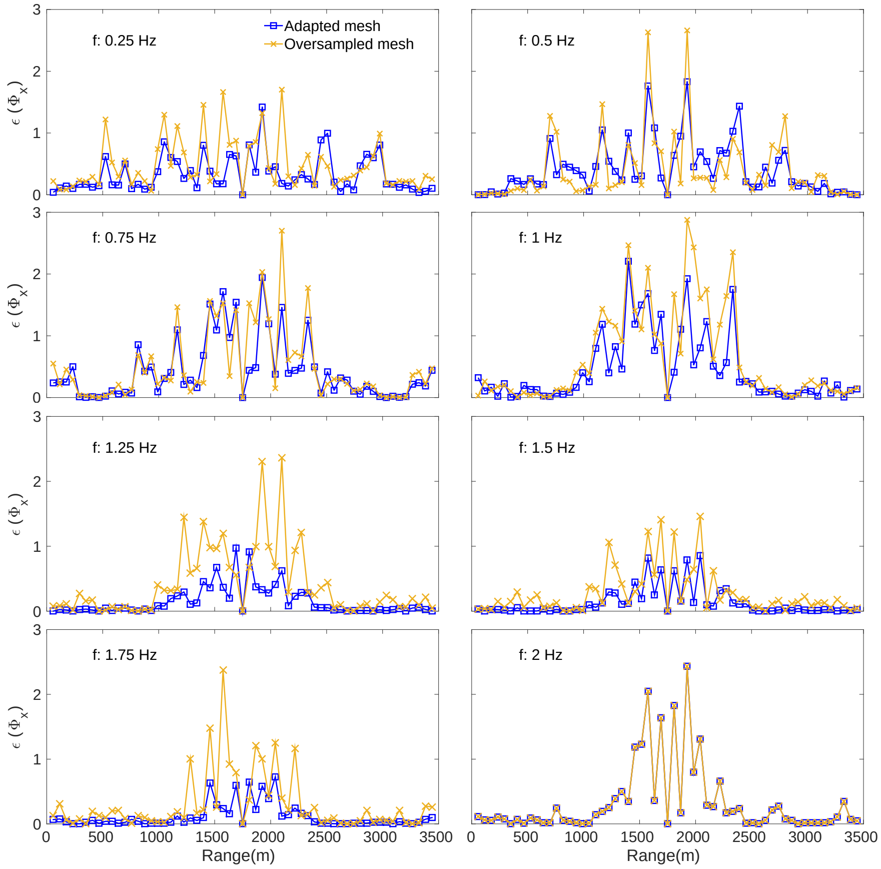

Figure 16: Phase misfits $\epsilon(\Phi_x)$ of $\mathbf{E}_x$ solution in Figure 14. Here, the $\epsilon(\Phi_x)$ is plotted for each frequency, where an average relative misfit $< 2\%$ is observed.

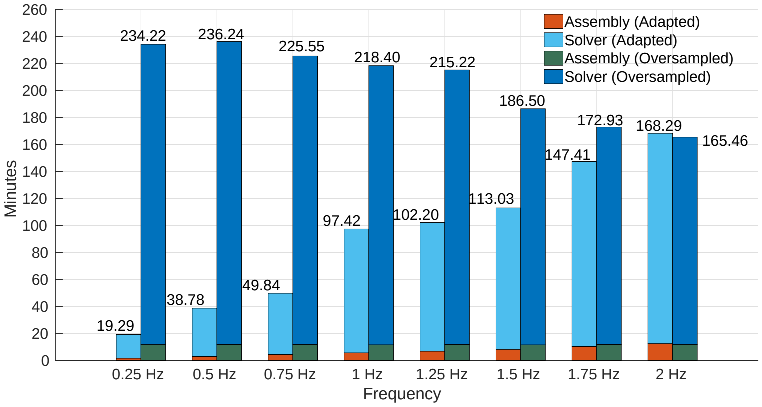

Figure 17: Time comparison between adapted and oversampled meshes. Assembly and solving times are considered.

## 6. Scalability tests

Finally, we perform a series of tests to examine the scaling of PETGEM on distributed-memory architectures by running the same problem for different number of CPUs working in parallel. In this set of experiments the most time-consuming sections have been considered, namely, assembly and solving tasks. All simulations have been carried out on version III of the *Marenostrum* (MN3) supercomputer at *BSC*.

The following tests are based on the canonical model described in Section 5.1. Its mesh has been created with *Gmsh* (Geuzaine and Remacle, 2008) and has 35 613 880 elements, 5 947 035 nodes and 41 753 430 edges. These tests have been carried out using 32, 64, 128, 256, 512, and 1 024 CPUs. These experiments are relevant because parallelism on distributed-memory platforms offers greater flexibility and capacity for realistic-scale 3D CSEM modeling.

Figure 18 shows speed-ups obtained for up to 1 024 CPUs of MN3 for case under consideration. The achieved scalability is almost linear for up to 256 CPUs. From this number on, the scalability stops its near-linear growth and slowly begins to saturate since the execution becomes dominated by exchange of messages between MPI tasks. However, the speed-ups keep growing constantly and significant reductions in runtime for more than thousand CPUs have been observed. Table 3 shows the runtime, speed-up and parallel efficiency of the modeling. Analysing these results, it is easy to see that the computation time has been reduced by increasing the number of processes (around 26 times when increasing the number of CPUs from 32 to 1 024). In order to perform a more thorough analysis of the MPI parallelism within PETGEM, we have carried out a set of simulations that has been been analyzed using *Paraver* (Pillet et al., 2017), which

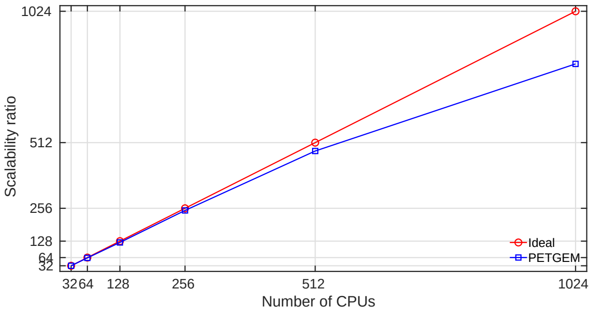

Figure 18: Scalability tests for the second-level mesh (41 753 430 DOFs) on MN3.

Table 3: Execution results for different number of CPUs on distributed-memory architectures

| # CPUs | 32 | 64 | 128 | 256 | 512 | 1 024 |
|---|---|---|---|---|---|---|
| Runtime (Min) | 945.30 | 482.14 | 246.09 | 122.96 | 63.08 | 36.92 |
| Speed-up | - | 1.96 | 3.84 | 7.68 | 14.98 | 25.60 |
| Efficiency | - | .98 | .96 | .96 | .94 | .80 |

is a parallel trace analyzer developed at *BSC*. This tool generates information that can be used in order to optimize a parallel application.

The analysis methodology begins with a set of *Paraver* traces for a number of MPI tasks, obtained from executing an instrumented version of PETGEM. Then, from a visual analysis of the traces clean, cuts are generated in order to identify the main computational phases. This allows discerning main computational regions from communication intensive stages, e.g. elemental matrix computations or solving phase from MPI calls. Furthermore, in this step, additional information such as useful computational duration and number of MPI calls could be measured. The net result of this analysis phase is depicted in Figure 19, where the color represents the duration of the computation burst (useful duration). This view gives a good perception of where are the major computation phases, and their balance across processors. As a result, in Figure 19 it is easy to see that assembly and solver phases are the main computational regions. Once the code structure has been identified, we analyze *Paraver* traces of PETGEM using 16, 32, and 64 CPUs in order to measure times for the aforementioned main computational regions. The main advantage of this strategy is that trace sizes are smaller and more manageable. In addition, all the effects that appear with the increase of the CPUs can be noticed much earlier.

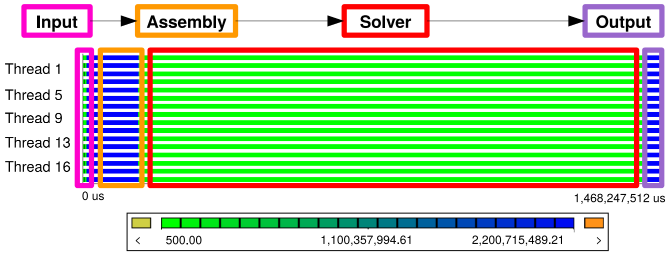

Figure 19: Main computational phases in PETGEM. Here, the useful duration is plotted for each CPU as function of time.

The scalability ratio for these experiments is shown in Figure 20, where it is easy to observe a quasi-linear ratio for up to 64 CPUs. Furthermore, the assembly task is slightly more efficient than solving because it is an embarrassingly parallel task. For the sake of clarity, we analyze the number of solver iterations for each *Paraver* trace. Thus, we cut a representative region of the solver phase and we measure the time (window size) for a number of iterations in the trace with 16 MPI tasks. Then, we count the number of iterations that fit in the same window size for remaining *Paraver* traces (32 and 64 CPUs). In our experiments, we fixed the initial number of solver iterations to 10, which produced a window size equal to 38 104 404 microseconds ($\mu s$). Results of this analysis are depicted in Figure 21, were it is easy to see an acceptable performance in terms of number of iterations when increasing number of CPUs. The examples show that PETGEM offers a good parallel performance for the solution of the problem under consideration.

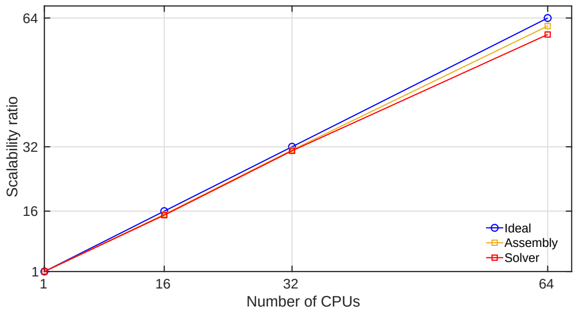

Figure 20: Scalability ratio of main computational phases in PETGEM

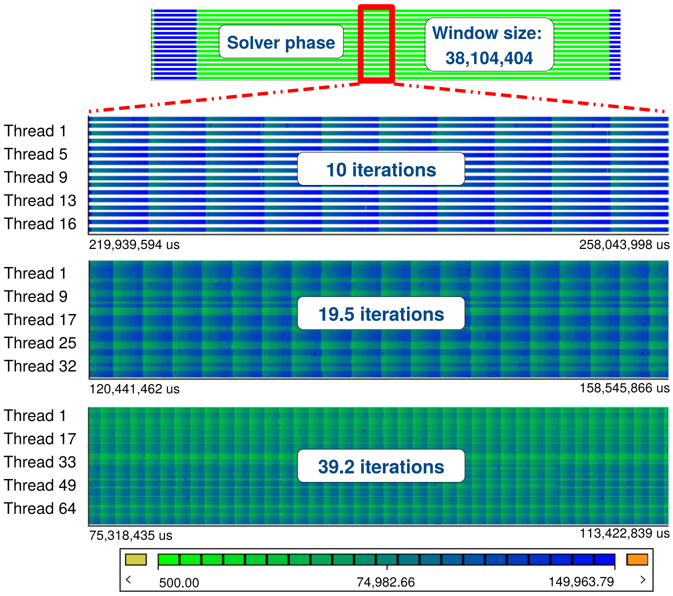

Figure 21: Solver scalability analysis for different number of CPUs using Paraver.

## 7. Conclusions

We have presented PETGEM, a 3D CSEM modeling tool that has been designed to cope with the topic of computational geophysics. Its structure has been designed to cope simultaneously with three key requirements: accuracy, flexibility, and efficiency. The proposed code is based on the Nédélec Edge Finite Element Method (EFEM) and pure Python 3 language. This is to our knowledge the first time this type of approach has been applied for running geophysical simulations on HPC platforms. By design, the code is very modular and flexible, which could allow users to easily switch its underlying numerics or expand its capabilities.

The accuracy and efficiency of this tool have been demonstrated on three different 3D CSEM models. The first model, a canonical test of an off-shore hydrocarbon reservoir, demonstrated a good agreement with the quasi-analytical solution. In the second model we have tested the code's capability to handle bathymetry. This model is especially interesting because it highlights one of the benefits of using an unstructured tetrahedral mesh, i.e. honoring complex geological structures without critically increasing the problem size. In the third test we exploit another advantage of unstructured tetrahedral meshes by building an automatic mesh adaptation technique for a multi-frequency study. This approach ensures, for a given frequency and source position, that the computational domain is consistent with the discretization of the EM equations with minimum amount of DOFs. For this simulation we observed that unnecessary and excessive refinement (over-sampled) examples required many more iterations than tests with adapted meshes, as a consequence needed considerably more CPU time. The adaptive mesh solutions had a factor of savings of up to four in time and storage compared to the fix-mesh result. Finally, a scalability study has shown that PETGEM offers an acceptable parallel performance for the physical problem under consideration.

Although PETGEM features will be beneficial for geophysicists interested in highly detailed and realistic 3D CSEM modeling, we anticipate that this algorithm will prove more useful as a kernel for inversions of EM datasets. By combining a good code structure, simple programming language and relying on external and robust preprocessing and solving libraries, PETGEM is a competitive, yet easy to use and expand, tool for 3D CSEM modeling. This effort, we hope, will foster research both in 3D CSEM modeling (and inversion) and in computer science.

## 8. Code availability

The PETGEM code is freely available at home page (*http://petgem.bsc.es*), from PyPI repository (*https://pypi.python.org/*
or by requesting the author (*octavio.castillo@bsc.es*, *ocastilloreyes@gmail.com*). In all cases, the code is supplied in a manner to ease the immediate execution under Linux platforms. User's manual and technical documentation (developer's guide) are provided in the PETGEM archive as well.

## 9. Acknowledgements

The authors are acknowledge the support of the *Mexican National Council for Science and Technology* (CONACyT). Octavio Castillo-Reyes expresses his gratitude to Prof. Yonghyun Chung of *Seoul National University* for the model and synthetic data used in Figure 9 and Figure 10. Thanks to the *BSC* (www.bsc.es) for allocating computation resources at *Marenostrum* supercomputer.

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under the *Marie Sklodowska-Curie* grant agreement No. 644202. In addition, the research leading to these results has received funding from the *European Union's Horizon 2020 Programme* (2014-2020) and from *Brazilian Ministry of Science, Technology and Innovation* through *Rede Nacional de Pesquisa* (RNP) under the HPC4E Project (www.hpc4e.eu), grant agreement No. 689772.

The authors gratefully acknowledge (in alphabetically order) to Dr. Eduardo Sánchez (*BSC*) and Dr. Otilio Rojas (*BSC*) for their valuable discussions and proofreading of this manuscript. Finally, last but not least, the authors are thankful to the reviewers for their valuable suggestions and comments.

## Appendix A. Source term formulation

If we consider eq. (1) and eq. (2) with harmonic time dependence $\mathrm{e}^{-i\omega t}$, $\omega$ as the angular frequency, $\mu_0$ as the free space magnetic permeability, $\mathbf{J}_s$ as the distribution of source current, for homogeneous media the source term for a x-directed dipole is given by

$$\mathbf{E} = \frac{I \cdot dS}{4\pi\sigma r^3} \cdot e^{-ikr} \begin{bmatrix} (\frac{d_x^2}{r^2} \cdot -k^2 \cdot r^2 + 3ikr + 3k^2r^2 - ikr - 1) \cdot \hat{i} \\ (\frac{d_x \cdot d_y}{r^2} \cdot -k^2r^2 + 3ikr + 3) \cdot \hat{j} \\ (\frac{d_x \cdot d_z}{r^2} \cdot -k^2r^2 + 3ikr + 3) \cdot \hat{k} \end{bmatrix}. \tag{A.1}$$

For a y-directed dipole, the source term is defined by

$$\mathbf{E} = \frac{I \cdot dS}{4\pi\sigma r^3} \cdot e^{-ikr} \begin{bmatrix} (\frac{d_x \cdot d_y}{r^2} \cdot -k^2r^2 + 3ikr + 3) \cdot \hat{i} \\ (\frac{d_y^2}{r^2} \cdot -k^2r^2 + 3ikr + 3 + k^2r^2 - ikr - 1) \cdot \hat{j} \\ (\frac{d_y \cdot d_z}{r^2} \cdot -k^2r^2 + 3ikr + 3) \cdot \hat{k} \end{bmatrix}. \tag{A.2}$$

Finally, following expression define the source term for a z-directed dipole

$$\mathbf{E} = \frac{I \cdot dS}{4\pi\sigma r^3} \cdot e^{-ikr} \begin{bmatrix} (\frac{d_x \cdot d_z}{r^2} \cdot -k^2 \cdot r^2 + 3ikr + 3) \cdot \hat{i} \\ (\frac{d_z \cdot d_y}{r^2} \cdot -k^2 \cdot r^2 + 3ikr + 3) \cdot \hat{j} \\ (\frac{d_z^2}{r^2} \cdot -k^2 \cdot r^2 + 3ikr + 3 + k^2r^2 - ikr - 1) \cdot \hat{k} \end{bmatrix}. \tag{A.3}$$

In eq. (A.1), eq. (A.2) and eq. (A.3), $I$ is the dipole current, $dS$ is the dipole length, $\sigma$ represent the background conductivity, $k$ is the propagation parameter (wavenumber), $r$ is the module between source position and evaluation point position, and $(d_x, d_y, d_z)$ is the distance between source position and evaluation point position.

## Appendix B. Nédélec element basis

If barycentric coordinates finite element defined in Zienkiewicz et al. (1977); Monk (2003) are renamed as $\boldsymbol{\lambda}_i^e = \mathbf{N}_i^e$, the vector basis functions for tetrahedral edge elements can be expressed in terms of first order of node-based Finite Element (Jin, 2002). Therefore, the vectorial functions for edge elements (see Figure 1) implemented in PETGEM are defined by

$$\mathbf{N}_1^e = W_{12}\ell_1^e = (\boldsymbol{\lambda}_1^e \nabla \lambda_2^e - \boldsymbol{\lambda}_2^e \nabla \lambda_1^e)\ell_1^e, \tag{B.1}$$

$$\mathbf{N}_2^e = W_{13}\ell_2^e = (\boldsymbol{\lambda}_1^e \nabla \lambda_3^e - \boldsymbol{\lambda}_3^e \nabla \lambda_1^e)\ell_2^e, \tag{B.2}$$

$$\mathbf{N}_3^e = W_{14}\ell_3^e = (\boldsymbol{\lambda}_1^e \nabla \lambda_4^e - \boldsymbol{\lambda}_4^e \nabla \lambda_1^e)\ell_3^e, \tag{B.3}$$

$$\mathbf{N}_4^e = W_{23}\ell_4^e = (\boldsymbol{\lambda}_2^e \nabla \lambda_3^e - \boldsymbol{\lambda}_3^e \nabla \lambda_2^e)\ell_4^e, \tag{B.4}$$

$$\mathbf{N}_5^e = W_{42}\ell_5^e = (\boldsymbol{\lambda}_4^e \nabla \lambda_2^e - \boldsymbol{\lambda}_2^e \nabla \lambda_4^e)\ell_5^e, \tag{B.5}$$

$$\mathbf{N}_6^e = W_{34}\ell_6^e = (\boldsymbol{\lambda}_3^e \nabla \lambda_4^e - \boldsymbol{\lambda}_4^e \nabla \lambda_3^e)\ell_6^e. \tag{B.6}$$

The gradients $\nabla \lambda_i^e$ can be expanded as

$$\nabla \lambda_i^e = \frac{1}{6V^e} \begin{bmatrix} b_i^e \cdot \hat{i} \\ c_i^e \cdot \hat{j} \\ d_i^e \cdot \hat{k} \end{bmatrix}. \tag{B.7}$$

Therefore, the expanded form of the edge basis functions for tetrahedral edge elements in PETGEM is the following

$$\mathbf{W}_{12}\ell_1^e = \frac{1}{(6V^e)^2}\begin{bmatrix} b_2^e(a_1^e + b_1^e x + c_1^e y + d_1^e z) - b_1^e(a_2^e + b_2^e x + c_2^e y + d_2^e z) \cdot \hat{i} \\ c_2^e(a_1^e + b_1^e x + c_1^e y + d_1^e z) - c_1^e(a_2^e + b_2^e x + c_2^e y + d_2^e z) \cdot \hat{j} \\ d_2^e(a_1^e + b_1^e x + c_1^e y + d_1^e z) - d_1^e(a_2^e + b_2^e x + c_2^e y + d_2^e z) \cdot \hat{k} \end{bmatrix} \ell_1, \tag{B.8}$$

$$\mathbf{W}_{13}\ell_2^e = \frac{1}{(6V^e)^2}\begin{bmatrix} b_3^e(a_1^e + b_1^e x + c_1^e y + d_1^e z) - b_1^e(a_3^e + b_3^e x + c_3^e y + d_3^e z) \cdot \hat{i} \\ c_3^e(a_1^e + b_1^e x + c_1^e y + d_1^e z) - c_1^e(a_3^e + b_3^e x + c_3^e y + d_3^e z) \cdot \hat{j} \\ d_3^e(a_1^e + b_1^e x + c_1^e y + d_1^e z) - d_1^e(a_3^e + b_3^e x + c_3^e y + d_3^e z) \cdot \hat{k} \end{bmatrix} \ell_2, \tag{B.9}$$

$$\mathbf{W}_{14}\ell_3^e = \frac{1}{(6V^e)^2}\begin{bmatrix} b_4^e(a_1^e + b_1^e x + c_1^e y + d_1^e z) - b_1^e(a_4^e + b_4^e x + c_4^e y + d_4^e z) \cdot \hat{i} \\ c_4^e(a_1^e + b_1^e x + c_1^e y + d_1^e z) - c_1^e(a_4^e + b_4^e x + c_4^e y + d_4^e z) \cdot \hat{j} \\ d_4^e(a_1^e + b_1^e x + c_1^e y + d_1^e z) - d_1^e(a_4^e + b_4^e x + c_4^e y + d_4^e z) \cdot \hat{k} \end{bmatrix} \ell_3, \tag{B.10}$$

$$\mathbf{W}_{23}\ell_4^e = \frac{1}{(6V^e)^2}\begin{bmatrix} b_3^e(a_2^e + b_2^e x + c_2^e y + d_2^e z) - b_2^e(a_3^e + b_3^e x + c_3^e y + d_3^e z) \cdot \hat{i} \\ c_3^e(a_2^e + b_2^e x + c_2^e y + d_2^e z) - c_2^e(a_3^e + b_3^e x + c_3^e y + d_3^e z) \cdot \hat{j} \\ d_3^e(a_2^e + b_2^e x + c_2^e y + d_2^e z) - d_2^e(a_3^e + b_3^e x + c_3^e y + d_3^e z) \cdot \hat{k} \end{bmatrix} \ell_4, \tag{B.11}$$

$$\mathbf{W}_{42}\ell_5^e = \frac{1}{(6V^e)^2}\begin{bmatrix} b_2^e(a_4^e + b_4^e x + c_4^e y + d_4^e z) - b_4^e(a_2^e + b_2^e x + c_2^e y + d_2^e z) \cdot \hat{i} \\ c_2^e(a_4^e + b_4^e x + c_4^e y + d_4^e z) - c_4^e(a_2^e + b_2^e x + c_2^e y + d_2^e z) \cdot \hat{j} \\ d_2^e(a_4^e + b_4^e x + c_4^e y + d_4^e z) - d_4^e(a_2^e + b_2^e x + c_2^e y + d_2^e z) \cdot \hat{k} \end{bmatrix} \ell_5, \tag{B.12}$$

$$\mathbf{W}_{34}\ell_6^e = \frac{1}{(6V^e)^2}\begin{bmatrix} b_4^e(a_3^e + b_3^e x + c_3^e y + d_3^e z) - b_3^e(a_4^e + b_4^e x + c_4^e y + d_4^e z) \cdot \hat{i} \\ c_4^e(a_3^e + b_3^e x + c_3^e y + d_3^e z) - c_3^e(a_4^e + b_4^e x + c_4^e y + d_4^e z) \cdot \hat{j} \\ d_4^e(a_3^e + b_3^e x + c_3^e y + d_3^e z) - d_3^e(a_4^e + b_4^e x + c_4^e y + d_4^e z) \cdot \hat{k} \end{bmatrix} \ell_6. \tag{B.13}$$

Eqs. (B.8) - (B.13) are divergence free but not curl free (Jin, 2002).

We redefine the stiffness matrix and mass matrix by Jin (2002) as follows

$$K_{ij}^e = \iiint_{V^e} (\boldsymbol{\nabla} \times \mathbf{N}_i^e \cdot S_i^e) \cdot (\boldsymbol{\nabla} \times \mathbf{N}_j^e \cdot S_j^e)\, dV, \tag{B.14}$$

$$M_{ij}^e = \iiint_{V^e} (\mathbf{N}_i^e \cdot S_i^e) \cdot (\mathbf{N}_j^e \cdot S_j^e)\, dV, \tag{B.15}$$

where, $S_i^e$ are coefficients equal 1 or $-1$ depending on the local and global direction of the *i-th* edge in the element *e*. In PETGEM these coefficients are computed as follows. If an edge adjoins two nodes $n_i$ and $n_j$, the direction of the edge is going from node $n_i$ to node $n_j$ if $i < j$. This simple algorithm gives a unique orientation of each edge in the mesh. On the other hand, the local orientation of edges within each element can be determined by his nodes indexes. Therefore, $S_i^e$ are given by the following vectorial function

$$S_i^e = \frac{node_{i2}^e - node_{i1}^e}{|node_{i2}^e - node_{i1}^e|} \qquad i = 1 \ldots 6, \tag{B.16}$$

where $i$ is the edge index within $e$-th element that adjoins $node_{i1}^e$ with $node_{i2}^e$. The main advantage of eq. (B.16) is that it allows to work with node numbering based on a clockwise or counter-clockwise in order to meet some conditions of FEM formulations such as element's volume computation, which must be positive in any case.

## References

## References

Alumbaugh, D.L., Newman, G.A., Prevost, L., Shadid, J.N., 1996. Three-dimensional wideband electromagnetic modeling on massively parallel computers. Radio Science 31, 1–23.

Amestoy, P.R., Guermouche, A., L'Excellent, J.Y., Pralet, S., 2006. Hybrid scheduling for the parallel solution of linear systems. Parallel Computing 32, 136–156.

Balay, S., Abhyankar, S., Adams, M.F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Eijkhout, V., Gropp, W.D., Kaushik, D., Knepley, M.G., McInnes, L.C., Rupp, K., Smith, B.F., Zampini, S., Zhang, H., Zhang, H., 2016. PETSc Web page. http://www.mcs.anl.gov/petsc. URL: http://www.mcs.anl.gov/petsc.

Beck, R., Hiptmair, R., 1999. Multilevel solution of the time-harmonic maxwell's equations based on edge elements. Internat. J. Numer. Methods Engrg. 45, 901–920.

Boulaenko, M., Hesthammer, J., Vereshagin, A., Gelting, P., Davies, R., Wedberg, T., 2007. Marine CSEM technologythe luva case. Houston Geological Society .

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

Cai, H., Xiong, B., Han, M., Zhdanov, M., 2014. 3D controlled-source electromagnetic modeling in anisotropic medium using edge-based finite element method. Computers & Geosciences 73, 164–176.

Castillo, O., de la Puente, J., Puzyrev, V., Cela, J.M., 2015. Edge-based electric field formulation in 3D CSEM simulations: a parallel approach, in: Proceedings of the 6th International Conference and Workshop on Computing and Communication, IEEE. Vancouver, Canada.

Chung, Y., Son, J.S., Lee, T.J., Kim, H.J., Shin, C., 2014. Three-dimensional modelling of controlled-source electromagnetic surveys using an edge finite-element method with a direct solver. Geophysical Prospecting 62, 1468–1483.

Constable, S., 2006. Marine electromagnetic methodsa new tool for offshore exploration. The Leading Edge 25, 438–444.

Constable, S., 2010. Ten years of marine CSEM for hydrocarbon exploration. Geophysics 75, 75A67–75A81.

Constable, S., Srnka, L.J., 2007. An introduction to marine controlled-source electromagnetic methods for hydrocarbon exploration. Geophysics 72, WA3–WA12.

Constable, S., Weiss, C.J., 2006. Mapping thin resistors and hydrocarbons with marine em methods: Insights from 1d modeling. Geophysics 71, G43–G51.

Dalcín, L., Paz, R., Kler, P., Cosimo, A., 2011. Parallel distributed computing using python. Advances in Water Resources 34, 1124–1139.

Eidesmo, T., Ellingsrud, S., MacGregor, L., Constable, S., Sinha, M., Johansen, S., Kong, F., Westerdahl, H., 2002. Sea bed logging (sbl), a new method for remote and direct identification of hydrocarbon filled layers in deepwater areas. First break 20, 144–152.

Geuzaine, C., Remacle, J.F., 2008. Gmsh: a three-dimensional finite element mesh generator with built-in pre-and post-processing facilities (2008). URL: http://www. geuz. org/gmsh .

Hanif, N.H.H.M., Hussain, N., Yahya, N., Daud, H., Yahya, N., Noh, M., 2011. 1d modeling of controlled-source electromagnetic (CSEM) data using finite element method for hydrocarbon detection in shallow water, in: Proceedings of the International MultiConference of Engineers and Computer Scientists.

Hiptmair, R., 2015. Maxwells Equations: Continuous and Discrete. Technical Report 2015-18. Seminar for Applied Mathematics, ETH Zürich. To appear in A. Bermdez de Castro, A. Valli (eds.), Computational Electromagnetism, Springer Lecture Notes in Mathematics 2148.

Jin, J., 2002. The Finite Element Method in Electromagnetics. Second ed., Wiley, New York.

Key, K., 2009. 1d inversion of multicomponent, multifrequency marine csem data: Methodology and synthetic studies for resolving thin resistive layers , F9–F20.

Koldan, J., 2013. Numerical solution of 3-D electromagnetic problems in exploration geophysics and its implementation on massively parallel computers. Ph.D. thesis. Polytechnic University of Catalonia.

Monk, P., 2003. Finite element methods for Maxwell's equations. First ed., Clarendon Press Oxford.

Newman, G.A., Alumbaugh, D.L., 2002. Three-dimensional induction logging problems, part 2: A finite-difference solution. Geophysics 67, 484–491.

Orange, A., Key, K., Constable, S., 2009. The feasibility of reservoir monitoring using time-lapse marine CSEM. Geophysics 74, F21–F29.

Owen, S.J., 1998. A survey of unstructured mesh generation technology., in: IMR, pp. 239–267.

Pillet, V., Labarta, J., Cortes, T., Girona, S., 2017. Paraver: a flexible performance analysis tool. URL http://www.bsc.es/computer-sciences/performance-tools/paraver/general-overview .

Plessix, R.E., Darnet, M., Mulder, W., 2007. An approach for 3d multisource, multifrequency csem modeling. Geophysics 72, SM177–SM184.

Puzyrev, V., Koldan, J., de la Puente, J., Houzeaux, G., Vázquez, M., Cela, J.M., 2013. A parallel finite-element method for three-dimensional controlled-source electromagnetic forward modelling. Geophysical Journal International , ggt027.

Sheriff, R.E., 2002. Encyclopedic dictionary of applied geophysics. Society of exploration geophysicists.

Um, E.S., Commer, M., Newman, G.A., 2013. Efficient pre-conditioned iterative solution strategies for the electromagnetic diffusion in the earth: finite-element frequency-domain approach. Geophysical Journal International , ggt071.

Zhdanov, M.S., 2009. Geophysical electromagnetic theory and methods. volume 43. Elsevier.

Zienkiewicz, O.C., Taylor, R.L., Zienkiewicz, O.C., Taylor, R.L., 1977. The finite element method. volume 3. McGraw-hill London.