Originally published as:

Robust and scalable 3-D geo-electromagnetic modelling approach using the finite element method

Alexander V. Grayver[1] and Markus Bürg[2]

[1]*GFZ German Research Centre for Geosciences, Telegrafenberg, D-14473 Potsdam, Germany. E-mail: agrayver@gfz-potsdam.de*
[2]*Department of Mathematics, Texas A&M University, College Station, TX 77843, USA*

SUMMARY

We present a robust and scalable solver for time-harmonic Maxwell's equations for problems with large conductivity contrasts, wide range of frequencies, stretched grids and locally refined meshes. The solver is part of the fully distributed adaptive 3-D electromagnetic modelling scheme which employs the finite element method and unstructured non-conforming hexahedral meshes for spatial discretization using the open-source software deal.II. We use the complex-valued electric field formulation and split it into two real-valued equations for which we utilize an optimal block-diagonal pre-conditioner. Application of this pre-conditioner requires the solution of two smaller real-valued symmetric problems. We solve them by using either a direct solver or the conjugate gradient method pre-conditioned with the recently introduced auxiliary space technique. The auxiliary space pre-conditioner reformulates the original problem in form of several simpler ones, which are then solved using highly efficient algebraic multigrid methods. In this paper, we consider the magnetotelluric case and verify our numerical scheme by using COMMEMI 3-D models. Afterwards, we run a series of numerical experiments and demonstrate that the solver converges in a small number of iterations for a wide frequency range and variable problem sizes. The number of iterations is independent of the problem size, but exhibits a mild dependency on frequency. To test the stability of the method on locally refined meshes, we have implemented a residual-based *a posteriori* error estimator and compared it with uniform mesh refinement for problems up to 200 million unknowns. We test the scalability of the most time consuming parts of our code and show that they fulfill the strong scaling assumption as long as each MPI process possesses enough degrees of freedom to alleviate communication overburden. Finally, we refer back to a direct solver-based pre-conditioner and analyse its complexity in time. The results show that for multiple right-hand sides the direct solver-based pre-conditioner can still be faster for problems of medium size. On the other hand, it also shows non-linear growth in memory, whereas the auxiliary space method increases only linearly.

Key words: Numerical solutions; Electrical properties; Electromagnetic theory; Magnetotellurics.

1 INTRODUCTION

The necessity to perform calculations of electromagnetic (EM) fields in 3-D settings arises frequently in geo-EM methods. For simulations of realistic surveys, including large numbers of sources and receivers in 3-D volumes of complex geometry, and for inverting realistic 3-D data sets, highly efficient modelling algorithms are required. The behaviour of the EM fields is governed by Maxwell's equations. For a comprehensive 3-D model of domains with a complex distribution of electrical properties and sources, we need to adopt advanced spatial discretization techniques in order to calcu-late an approximation of the EM field within the domain subject to prescribed boundary conditions. In this work, we use the finite element method (FEM) for this task. Our choice is motivated by the fact that FEM for Maxwell's equations provides a well-established theory (Hiptmair 2002; Jin 2002; Monk 2003) which enables extensive error estimation analysis (e.g. Beck *et al.* 2000; Bürg 2012, and references therein) and enough flexibility to develop efficient adaptive schemes in 3-D (e.g. Bürg 2013; Ren *et al.* 2013). To large extent the success of FEM in an applied sense is supported by the existence of well designed and thoroughly tested and maintained open-source libraries (Kirk *et al.* 2006; Bangerth *et al.* 2007;

Logg & Wells 2010) and meshing tools. They simplify development of the specific application codes and their applicability to realistic problems.

There is a large number of papers published which use a variety of finite element formulations aimed at geo-EM modelling in 3-D (see review paper by Börner (2010) and more recent works by Schwarzbach *et al.* 2011; Farquharson & Miensopust 2011; Puzyrev *et al.* 2013; Ren *et al.* 2013). Usually, finite element discretizations of Maxwell's equations result in large, but sparse, systems of linear equations. The solution of these systems constitutes the most time-consuming part of the modelling algorithm. Hence, there is a huge demand for robust and scalable methods to solve these problems.

The classical formulation of time-harmonic Maxwell's equations involves the operator $\nabla \times$ which inherits a large and non-trivial null-space (Zaglmayr 2006; Schwarzbach 2009). This makes the solution of problems involving low frequencies difficult and the common remedy is an explicit application of divergence correction schemes (Smith 1996; Sasaki & Meju 2009; Farquharson & Miensopust 2011). Although divergence correction suppresses some spurious effects in the solution, the original problem remains ill-conditioned. Many realistic models contain air layers which have a much lower conductivity than any subsurface materials resulting in conductivity contrasts of 10^8 S m^{-1} or even larger. Generic iterative methods experience difficulties for this type of problems (Ernst & Gander 2011; Um *et al.* 2013) and converge slowly. Furthermore, non-uniform and locally refined meshes are often required to adequately discretize large 3-D modelling domains. Again, this may result in efficiency losses (Mulder 2006; Farquharson & Miensopust 2011) due to the negative impact on the condition number of the system matrix. In order to ensure accuracy and efficacy for a wide range of practical problems, the robustness of the applied method is crucial.

Modern high-performance platforms with distributed architectures allow solving ever larger complex problems. However, to utilize the computational power of these platforms, one needs to address a number of challenges regarding the efficiency of algorithms and methods in a distributed environment (Gropp 2005). For instance, distributed direct solvers (Schenk & Gärtner 2004; Li 2005; Amestoy *et al.* 2006; Maurer & Wieners 2011) are very robust and their performance is virtually independent of model parameters. Furthermore, they allow a user to obtain solutions for multiple right-hand sides at low cost, since only a single expensive factorization of the system matrix is required; followed by relatively inexpensive forward and backward substitutions (Operto *et al.* 2007; Streich 2009). Unfortunately, direct solvers lack scalability due to non-linear growth of time and memory complexities and low computation-to-communication ratios (Gould *et al.* 2007; Pardo *et al.* 2012) which, in practice, limits the size of problems they can be applied to. In contrast, multilevel methods such as domain decomposition and multigrid are considered scalable (Smith *et al.* 2004). Furthermore, these methods preserve their efficiency when run on the largest supercomputers available today (Adams 2012; Baker *et al.* 2012; Brown *et al.* 2013). Classic multilevel methods fail when applied to Maxwell's equations because of a large null-space of the curl operator (Zaglmayr 2006). Although, specific geometric multigrid schemes, which address this issue explicitly, have been devised (Hiptmair 1998; Arnold *et al.* 2000; Mulder 2006; Haber & Heldmann 2007; Nechaev *et al.* 2008). Nevertheless, the difficulty with geometric multigrid methods is that their scalable implementation for unstructured grids with highly discontinuous coefficients and stretched meshes can be very challenging and technically involved.

The main goal of this work is to find a robust and scalable solver for large-scale 3-D geo-EM forward modelling problems which takes into account all of the considerations above. In this contribution, we concentrate on 3-D EM modelling in frequency domain for the magnetotelluric method (MT). We make a corresponding plane-wave source assumption and neglect displacement currents as they are irrelevant in the frequency range under consideration (Berdichevskii & Dmitriev 2008). Nevertheless, the conclusions drawn and methods presented are valid for controlled-source EM methods as well.

Section 2 starts with the problem formulation. We briefly describe its discretization using the FEM in Section 2.2 and introduce an *a posteriori* error estimator in Section 2.3. We split the complex-valued equations, which have been derived in frequency domain, into two coupled real-valued equations and employ an efficient block-diagonal pre-conditioner for the resulting problem in Section 3. In Section 3.1, we show how a direct solver can be used efficiently on smaller blocks of the pre-conditioner to accelerate the solution process of the original system for problems of moderate size. For larger problems, we adopt the auxiliary space pre-conditioner (Hiptmair & Xu 2007; Kolev & Vassilevski 2009) which reformulates the original problem as a combination of simpler ones and solves them using highly efficient methods. Details are given in Section 3.2. The rationale for real-valued formulation is given in Section 3.3. Section 4 reports results for a series of numerical experiments aimed at analysing the robustness and scalability of the methods at hand by using benchmark COMMEMI 3-D models.

2 FORMULATION OF THE PROBLEM

2.1 Governing equations

We consider the time-harmonic electric field formulation of Maxwell's equations with time-dependence expressed by the factor $e^{i\omega t}$:

$$\nabla \times (\mu^{-1} \nabla \times \mathbf{E}) + i\omega\sigma\mathbf{E} = \mathbf{0} \quad \text{in} \quad \Omega$$

$$\nabla \cdot (\sigma\mathbf{E}) = 0 \quad \text{in} \quad \Omega, \tag{1}$$

where $\Omega \subseteq \mathbb{R}^3$ is some bounded Lipschitz domain, ω is an angular frequency, μ is the magnetic permeability of free space, $\mathbf{E} : \Omega \to \mathbb{C}^3$ is some complex-valued electric field, and $\sigma : \Omega \to \mathbb{R}_+$ denotes the electric conductivity. In order to guarantee the uniqueness of the solution (e.g. Monk 2003), we impose the Dirichlet boundary conditions

$$\mathbf{n} \times \mathbf{E} = \mathbf{n} \times \mathbf{E}^0 \quad \text{on} \quad \Gamma,$$

where $\mathbf{n}$ is an outward-pointing normal vector on Γ and $\mathbf{E}^0$ is a known exterior field. For simplicity, we concentrate on the isotropic case, although the implemented scheme can easily be adapted to support anisotropic media. Once eq. (1) is solved, the magnetic field is calculated by virtue of Faraday's law

$$\mathbf{H} = -\frac{1}{i\omega\mu} \nabla \times \mathbf{E}. \tag{2}$$

By taking the divergence of the first equation of (1) and using the fact that $\nabla \cdot (\nabla \times \mathbf{F}) = 0$ for all sufficiently smooth functions $\mathbf{F} : \Omega \to \mathbb{C}^3$ and $\sigma \neq 0$, we observe that the second equation follows directly from the first one. Thus, it suffices to consider the boundary value problem

$$\nabla \times (\mu^{-1} \nabla \times \mathbf{E}) + i\omega\sigma\mathbf{E} = \mathbf{0} \quad \text{in} \quad \Omega$$

$$\mathbf{n} \times \mathbf{E} = \mathbf{n} \times \mathbf{E}^0 \quad \text{on} \quad \Gamma. \tag{3}$$

By using the secondary field formulation (Newman & Alumbaugh 1995), this can be written as

$$\nabla \times (\mu^{-1}\nabla \times \mathbf{E}^s) + i\omega\sigma\mathbf{E}^s = -i\omega(\sigma - \sigma^0)\mathbf{E}^0 \quad \text{in} \quad \Omega$$

$$\mathbf{n} \times \mathbf{E}^s = 0 \quad \text{on} \quad \Gamma, \tag{4}$$

where σ^0 is some background model conductivity. The background model is represented by a horizontally layered medium and $\mathbf{E}^0 = \mathbf{E}^0(\omega, \sigma^0, z)$ is an incident electric field which can be calculated analytically. Then, the total electric and magnetic fields are calculated as

$$\mathbf{E} = \mathbf{E}^0 + \mathbf{E}^s$$

and

$$\mathbf{H} = \mathbf{H}^0 - \frac{1}{i\omega\mu}\nabla \times \mathbf{E}^s.$$

Inserting $\mathbf{E}^s := \mathbf{E}^s_R + i\mathbf{E}^s_I$ into (4) and splitting the result into a system of two real-valued equations yields

$$\nabla \times (\mu^{-1}\nabla \times \mathbf{E}^s_R) - \omega\sigma\mathbf{E}^s_I = \omega(\sigma - \sigma^0)\mathbf{E}^0_I$$

$$\nabla \times (\mu^{-1}\nabla \times \mathbf{E}^s_I) + \omega\sigma\mathbf{E}^s_R = -\omega(\sigma - \sigma^0)\mathbf{E}^0_R$$

$$\mathbf{n} \times \mathbf{E}^s_R = 0 \quad \text{on} \quad \Gamma$$

$$\mathbf{n} \times \mathbf{E}^s_I = 0 \quad \text{on} \quad \Gamma. \tag{5}$$

Analogously, we can separate complex-valued eq. (2) into two real-valued equations

$$\mathbf{H}^s_R = -\frac{1}{\omega\mu}\nabla \times \mathbf{E}^s_I + \mathbf{H}^0_R \tag{6}$$

and

$$\mathbf{H}^s_I = \frac{1}{\omega\mu}\nabla \times \mathbf{E}^s_R + \mathbf{H}^0_I. \tag{7}$$

In what follows, we will work with eqs (5)–(7). We defer the discussion of the motivation behind this split to Section 3.3.

2.2 Spatial discretization using finite elements

First, we define a space of 3-D vector functions whose curl is in L^2 by

$$\mathcal{H}(\text{curl}, \Omega) = \{\mathbf{u} \in (L^2(\Omega))^3 \mid \nabla \times \mathbf{u} \in (L^2(\Omega))^3\}.$$

By $\mathcal{H}_0(\text{curl}, \Omega)$ we denote the functions $\mathbf{u} \in \mathcal{H}(\text{curl}, \Omega)$ which additionally satisfy homogeneous Dirichlet boundary conditions, that is, $\mathbf{n} \times \mathbf{u} = 0$ on Γ.

Next, we multiply system (5) by test functions $\varphi, \psi \in \mathcal{H}_0(\text{curl}, \Omega)$ and use integration by parts to obtain the weak formulation for finding $\mathbf{E}^s_R, \mathbf{E}^s_I \in \mathcal{H}_0(\text{curl}, \Omega)$ such that

$$\int_\Omega \mu^{-1}\nabla \times \mathbf{E}^s_R \cdot \nabla \times \varphi - \omega\int_\Omega \sigma\mathbf{E}^s_I \cdot \varphi = \omega\int_\Omega (\sigma - \sigma^0)\mathbf{E}^0_I \cdot \varphi$$

$$\int_\Omega \mu^{-1}\nabla \times \mathbf{E}^s_I \cdot \nabla \times \psi + \omega\int_\Omega \sigma\mathbf{E}^s_R \cdot \psi = -\omega\int_\Omega (\sigma - \sigma^0)\mathbf{E}^0_R \cdot \psi \tag{8}$$

for all $\varphi, \psi \in \mathcal{H}_0(\text{curl}, \Omega)$. By defining the inner product

$$\langle \psi, \varphi \rangle_{L^2(\Omega)} := \int_\Omega \psi \cdot \varphi,$$

we can write (8) in the more compact form

$$\langle \mu^{-1}\nabla \times \mathbf{E}^s_R, \nabla \times \varphi \rangle - \omega\langle \sigma\mathbf{E}^s_I, \varphi \rangle = \omega\langle (\sigma - \sigma^0)\mathbf{E}^0_I, \varphi \rangle$$

$$\langle \mu^{-1}\nabla \times \mathbf{E}^s_I, \nabla \times \psi \rangle + \omega\langle \sigma\mathbf{E}^s_R, \psi \rangle = -\omega\langle (\sigma - \sigma^0)\mathbf{E}^0_R, \psi \rangle. \tag{9}$$

To discretize this problem, we split the continuous domain Ω into a set $\mathcal{K}$ of non-overlapping hexahedral, possibly non-parallelogrammic, cells. For electric fields, one usually wants to permit discontinuities of the normal components, but have continuous tangential components. These continuity requirements are satisfied by all functions from the space $\mathcal{H}_0(\text{curl}, \Omega)$. For the discretization of this function space, we choose Nédélec elements of order zero (Bossavit 1998; Monk 2003; Šolín *et al.* 2004). The 12 shape functions—one for every edge of a hexahedron—of this finite element are continuous across cell boundaries in its tangential components, but discontinuous in its normal component. Since we use real-valued formulation (5), there are two degrees of freedom (DoF) per edge, resulting in a total of 24 DoFs per cell. For details on the definitions of basis functions and mappings between reference and real cells, the reader is referred to Brenner & Scott (1994), Šolín *et al.* (2004), Nam *et al.* (2007) and Demkowicz *et al.* (2008).

Replacing $\mathcal{H}_0(\text{curl}, \Omega)$ in (9) by its finite-dimensional approximation space $\mathbf{V}(\mathcal{K})$ yields the discrete formulation to find $\mathbf{E}^s_R, \mathbf{E}^s_I \in \mathbf{V}(\mathcal{K})$ such that

$$\langle \mu^{-1}\nabla \times \mathbf{E}^s_R, \nabla \times \varphi \rangle - \omega\langle \sigma\mathbf{E}^s_I, \varphi \rangle = \omega\langle (\sigma - \sigma^0)\mathbf{E}^0_I, \varphi \rangle$$

$$\langle \mu^{-1}\nabla \times \mathbf{E}^s_I, \nabla \times \psi \rangle + \omega\langle \sigma\mathbf{E}^s_R, \psi \rangle = -\omega\langle (\sigma - \sigma^0)\mathbf{E}^0_R, \psi \rangle \tag{10}$$

for all $\phi, \psi \in \mathbf{V}(\mathcal{K})$. Then, the solutions $\mathbf{E}^s_R, \mathbf{E}^s_I$ of (10) can be written as

$$\mathbf{E}^s_R := \sum_{i=1}^n \alpha_i\varphi_i \quad \text{and} \quad \mathbf{E}^s_I := \sum_{i=1}^n \beta_i\psi_i,$$

where $\{\varphi_i, \psi_j\}_{1 \le i, j \le n}$ is a basis of $\mathbf{V}(\mathcal{K})$. Inserting this into discrete formulation (10) yields

$$\sum_{i=1}^n \alpha_i\langle \mu^{-1}\nabla \times \varphi_i, \nabla \times \varphi_j \rangle - \omega\sum_{i=1}^n \beta_i\langle \sigma\psi_i, \varphi_j \rangle$$

$$= \langle \omega(\sigma - \sigma^0)\mathbf{E}^0_I, \varphi_j \rangle$$

$$\sum_{i=1}^n \beta_i\langle \mu^{-1}\nabla \times \psi_i, \nabla \times \psi_j \rangle + \omega\sum_{i=1}^n \alpha_i\langle \sigma\varphi_i, \psi_j \rangle$$

$$= -\langle \omega(\sigma - \sigma^0)\mathbf{E}^0_R, \psi_j \rangle \tag{11}$$

which can be written in matrix-vector notation as

$$\begin{pmatrix} \mathbf{C} & -\mathbf{M} \\ \mathbf{M} & \mathbf{C} \end{pmatrix}\begin{pmatrix} \bar{\mathbf{E}}^s_R \\ \bar{\mathbf{E}}^s_I \end{pmatrix} = \begin{pmatrix} \mathbf{f}_R \\ \mathbf{f}_I \end{pmatrix}, \tag{12}$$

where $\bar{\mathbf{E}}^s_R = (\alpha_1, \ldots, \alpha_n)^T$, $\bar{\mathbf{E}}^s_I = (\beta_1, \ldots, \beta_n)^T$ are blocks of the solution vector of this linear system of equations. Here, the sparse, symmetric, positive semi-definite matrices $\mathbf{C} \in \mathbb{R}^{n \times n}$ and $\mathbf{M} \in \mathbb{R}^{n \times n}$ are given by

$$[C]_{ij} = \langle \mu^{-1}\nabla \times \varphi_i, \nabla \times \varphi_j \rangle$$

and

$$[M]_{ij} = \omega\langle \sigma\psi_i, \varphi_j \rangle,$$

respectively, and $N := 2n$ stands for the total number of degrees of freedom with n being the number of edges of the mesh.

From the numerical point of view, it is beneficial to work with symmetric matrices. Multiplying the second row in eq. (12) by minus one, results in a symmetric indefinite system of linear equations

$$\underbrace{\begin{pmatrix} \mathbf{C} & -\mathbf{M} \\ -\mathbf{M} & -\mathbf{C} \end{pmatrix}}_{\mathbf{A}} \begin{pmatrix} \bar{\mathbf{E}}_R^s \\ \bar{\mathbf{E}}_I^s \end{pmatrix} = \begin{pmatrix} \mathbf{f}_R \\ -\mathbf{f}_I \end{pmatrix}. \tag{13}$$

In MT, we are highly interested in derived quantities such as the impedance tensor. To be able to calculate this tensor, system (13) must be solved two times for orthogonal source polarizations. Since we use the secondary field formulation and homogeneous Dirichlet boundary conditions, changing the polarization affects only the right-hand side of eq. (5). Hence, a costly recalculation of the system matrix entries in eq. (13) is avoided. Denoting horizontal components of the electric and magnetic fields for the two source polarizations by $E_{x1}, E_{y1}, H_{x1}, H_{y1}$ and $E_{x2}, E_{y2}, H_{x2}, H_{y2}$, we can write the impedance tensor $\mathbf{Z}$ as (Berdichevskii & Dmitriev 2008)

$$\begin{pmatrix} H_{x1} & H_{x2} \\ H_{y1} & H_{y2} \end{pmatrix} = \begin{pmatrix} Z_{xx} & Z_{xy} \\ Z_{yx} & Z_{yy} \end{pmatrix} \begin{pmatrix} E_{x1} & E_{x2} \\ E_{y1} & E_{y2} \end{pmatrix}, \tag{14}$$

from which individual components can be derived (Mackie & Madden 1993). Finally, apparent resistivity and phase for an impedance tensor component can be written as

$$\rho_a^{ij} = \frac{1}{\omega\mu}|Z_{ij}|^2 \tag{15}$$

and

$$\phi^{ij} = \arctan\left[\frac{\text{Im}(Z_{ij})}{\text{Re}(Z_{ij})}\right], \tag{16}$$

respectively.

2.3 *A posteriori* error estimator

Locally refined meshes can affect the condition number of the system matrix. To facilitate a proper testing of the presented methods on a series of locally refined meshes, we implemented a residual-based *a posteriori* error estimator (Beck *et al.* 2000; Chen *et al.* 2007; Bürg 2012) to estimate the accuracy of the computed numerical solution of (11) in comparison to the (unknown) analytical solution of (5). To identify those cells of $\mathcal{K}$ which need to be refined, the error estimator can be split into a sum of local contributions

$$\eta^2 := \sum_{K \in \mathcal{K}} \eta_K^2, \tag{17}$$

where

$$\eta_K^2 := \eta_{R,K}^2 + \eta_{J,K}^2 \tag{18}$$

for all $K \in \mathcal{K}$. Here, the residual-based term $\eta_{R,K}$ is defined as:

$$\eta_{R,K}^2 := h_K^2 \Big(\big\| \nabla \times \big(\mu^{-1}\nabla \times \mathbf{E}^s\big)$$
$$+ \, i\omega \big(\sigma\mathbf{E}^s + (\sigma - \sigma_0)\mathbf{E}^0\big) \big\|_{L^2(K)}^2$$
$$+ \big\| \nabla \cdot \big(\sigma\mathbf{E}^s + (\sigma - \sigma_0)\mathbf{E}^0\big) \big\|_{L^2(K)}^2 \Big), \tag{19}$$

and the jump-based term $\eta_{J,K}$ is given by

$$\eta_{J,K}^2 := \frac{1}{2} \sum_{e \in \mathcal{E}(K)} h_e \Big(\big\| \big[n_e \times \big(\mu^{-1}\nabla \times \mathbf{E}^s\big)\big] \big\|_{L^2(e)}^2$$
$$+ \big\| \big[n_e \cdot \big(\sigma\mathbf{E}^s + (\sigma - \sigma_0)\mathbf{E}^0\big)\big] \big\|_{L^2(e)}^2 \Big), \tag{20}$$

where $h_K := \text{diam}(K)$ is the diameter of hexahedra K, e is an interior face of K, that is, $e \cap \Gamma = \emptyset$, and $[\,\cdot\,]$ denotes the jump of the quantity across element boundaries. The residual-based term vanishes if both original eq. (4) and its divergence are exactly fulfilled. Note that the divergence of (4) represents the conservation law. The jump-based term vanishes if the tangential components of the magnetic field and the normal component of the current density are continuous. The continuity of these quantities is not guaranteed by the weak formulation (9) or the finite element solution of (11) itself, yet they approach zero when the finite element solution converges towards the analytical solution of (1). Thus, they turn out to be meaningful error indicators.

Since the error indicator η_K can be calculated for every cell $K \in \mathcal{K}$ and represents an estimation of the error of the finite element solution on this cell, it can be used for local adaptive mesh refinement. To this end, we use the fixed fraction strategy (Dörfler 1996) and select the smallest subset $\mathcal{M} \subseteq \mathcal{K}$ of cells, such that

$$\sum_{K \in \mathcal{M}} \eta_K^2 \geq \theta^2 \eta^2$$

for a predefined $0 < \theta \leq 1$. The selected cells are then refined through bisection in all three dimensions. The procedure can be repeated until η is sufficiently small. Note that the choice $\theta = 1$ corresponds to a global mesh refinement.

2.4 Hanging nodes

In contrast to tetrahedral meshes, it is extremely difficult to obtain a conforming hexahedral mesh (i.e. all cells share only a whole edge or face) during local refinement. To mitigate this problem, we permit non-conforming meshes with 1-irregular hanging nodes as shown in Fig. 1. To explain the concept of 1-irregular hanging nodes in a bit more detail, let us first consider the 2-D situation depicted in Fig. 2(a). As described in Section 2.2, each edge possesses one DoF and, thus, the edge hosting DoF x_2 coincides in part with the edges hosting DoFs x_0 and x_1, respectively. Since we want to approximate functions from the space $\mathcal{H}_0(\text{curl}, \Omega)$, the tangential component of the finite element solution of (11) has to be continuous across this edge. In this situation, this is not immediately the case, because the edge, on which DoF x_2 is located, is longer than the edges associated with DoFs x_0 and x_1. To resolve this discrepancy, one constrains DoFs x_0 and x_1 to be

$$x_0 = \frac{x_2}{2} \qquad \text{and} \qquad x_1 = \frac{x_2}{2}. \tag{21}$$

Now, let us consider the 3-D situation shown in Fig. 2(b). Here, we have in total twelve constrained DoFs: eight are located at the edges of the face at which the cells meet (indicated with red lines in Fig. 2b) and the other four are located on the edges which are inside of the face (indicated with blue lines in Fig. 2b). The eight DoFs on the outer edges are treated in exactly the same way as described in the 2-D case. The other four DoFs located in the interior of the face are set to the average values of the degrees of freedom associated with the two outer edges of the face which point into the same direction as the interior edge.

Enforcing these constraints is done in several steps. First, the entries of the rows and columns corresponding to the constrained DoFs are distributed to the corresponding unconstrained rows and columns. Similarly, the corresponding elements of the right-hand side vector are distributed to the unconstrained entries of this vector. After distribution, the constrained rows and columns of the system matrix and the corresponding entries of the right-hand side vector
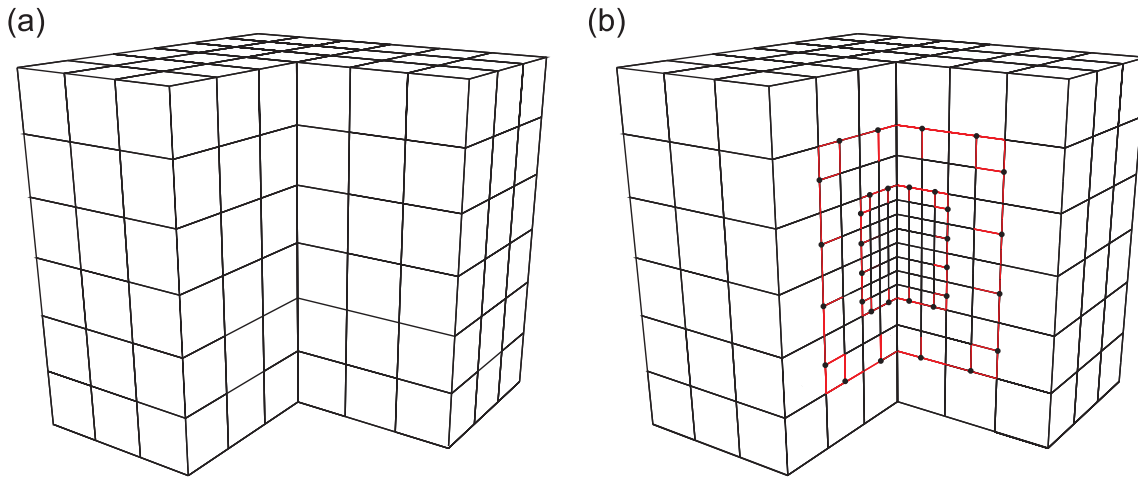
(a)

(b)

Figure 1. (a) Conforming regular hexahedral mesh. (b) Locally refined mesh with black dots indicating hanging nodes. Accordingly, edges coloured in red are associated with hanging nodes.
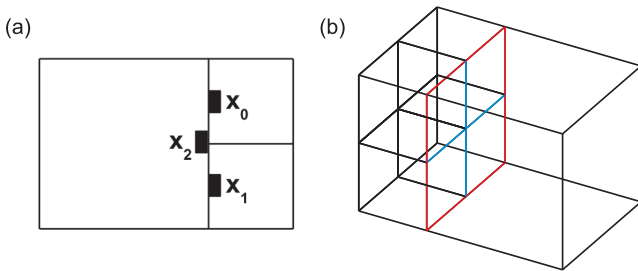
(a)   (b)

Figure 2. 1-irregular hanging nodes in 2-D (a) and 3-D (b).

are set to zero. To avoid a singular system matrix, however, entries on the main diagonal, which correspond to the constrained DoFs, are set to a positive value with magnitude similar to the other entries on the diagonal. This eliminates coupling between unconstrained and constrained DoFs and, thus, the elements of the solution vector corresponding to the constrained DoFs are zero, and have to be calculated by using the constraints (e.g. from eq. 21) once system (13) is solved. After this procedure, the finite element solution of the eq. (11) satisfies the continuity requirements of the space $\mathcal{H}_0(\text{curl}, \Omega)$ again. For more details about the concept of hanging nodes, we refer to Bangerth *et al.* (2007) and Bangerth & Kayser-Herold (2009).

3 LINEAR SOLVERS

In this section, we discuss the solution of the resulting system of linear eqs (13). A straightforward approach would be to use LU decomposition which is computationally very expensive and not scalable both in terms of time and memory (Operto *et al.* 2007; Brossier *et al.* 2010). Since we aim to solve large problems, we need to circumvent these drawbacks.

Therefore, we apply an iterative Krylov subspace technique, namely the flexible generalized minimal residual (FGMRES) method (Saad 2003). This method also works for indefinite matrices such as the one in eq. (13). For iterative solvers, the pre-conditioner is an extremely important component and has dramatic impact on its convergence rate. A good pre-conditioner has to be spectrally equivalent to the matrix it is applied to, that is, for some pre-conditioner **P**, the condition number of the product $\mathbf{P}^{-1}\mathbf{A}$ should have a con-

stant upper bound independent of the size of **A**. At the same time, construction and application of a pre-conditioner have to be cheaper than a direct factorization of the system matrix.

Here, we use a real, symmetric, and positive definite block-diagonal pre-conditioner

$$\mathbf{P}_{\mathbf{A}} := \begin{pmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{pmatrix}, \tag{22}$$

where $\mathbf{B} = \mathbf{C} + \mathbf{M}$. Chen *et al.* (2010) proved that, as long as matrices **C** and **M** are symmetric positive semi-definite and both conductivity and frequency are larger than zero, the matrix $\mathbf{P}_{\mathbf{A}}$ is spectrally equivalent to system matrix **A** from eq. (13) and the condition number of $\mathbf{P}_{\mathbf{A}}^{-1}\mathbf{A}$ has an upper bound of $\sqrt{2}$, independent of matrix size, frequency and conductivity contrasts. Therefore, by using this pre-conditioner, FGMRES should converge to a solution of (13) in a small number of iterations $N_{\text{iter}} \ll N$.

Because of its block-diagonal structure, a product of the inverse $\mathbf{P}_{\mathbf{A}}^{-1}$ with a vector entails the solution of two linear systems

$$\mathbf{Bx} = \mathbf{b}. \tag{23}$$

Since FGMRES requires one matrix-vector product per iteration, N_{iter} FGMRES iterations will incur the solution of $2N_{\text{iter}}$ problems (23). In the next two sections, we will discuss efficient methods to solve systems like (23).

3.1 Direct solver-based block pre-conditioner

To solve the linear system of eqs (23), we calculate the $\mathbf{LDL}^T$ decomposition of the symmetric matrix **B**, where **L** denotes some lower-triangular matrix and **D** is a diagonal matrix (Golub & van Loan 1996). Since pre-conditioner (22) is constructed only once, we also have to compute only one single decomposition. All other matrix–vector products will merely require inexpensive backward and forward substitutions. As long as the $\mathbf{P}_{\mathbf{A}}$-pre-conditioned FGMRES needs only a few iterations to converge, the time spent in backward and forward substitutions can be neglected and the overall cost of the solver is approximately equivalent to the time and memory required by the $\mathbf{LDL}^T$ factorization.

As a result, instead of applying an expensive **LU** factorization directly to the matrix in eq. (12), we use the FGMRES method pre-conditioned with the matrix $\mathbf{P}_{\mathbf{A}}$. This approach needs only one

single and much cheaper $\mathbf{LDL}^T$ decomposition of the matrix $\mathbf{B}$ which is half the size of $\mathbf{A}$. This results in a significant reduction of computational time and memory. Furthermore, there is also no need to recompute the factorization for subsequent right-hand side vectors; making this approach particularly suitable for inverse problems of medium sizes, where the necessity to get solutions for multiple right-hand sides exists (Egbert 2012; Grayver *et al.* 2013).

3.2 Auxiliary space block pre-conditioner

Even though the direct solver-based block pre-conditioner presented in the previous section is more efficient than applying a direct solver to $\mathbf{A}$ or to its complex analogue (see Section 3.3), its scalability is still quite poor. For larger problems with $N > 10^6$, we have to resort to a scalable and robust method to solve the block linear system of equations. Matrix $\mathbf{B}$ contains a discretization of the bilinear form involving the operator $\nabla \times$ and, therefore, all the difficulties discussed in Section 1 regarding the solution of the linear system of eqs (23) remain. Hiptmair & Xu (2007) address this issue and develop an efficient auxiliary space pre-conditioner for this type of problems.

For the sake of completeness, we give here a brief description of their pre-conditioner in simplified settings. A detailed derivation can be found in Hiptmair & Xu (2007), Kolev & Vassilevski (2009). In essence, auxiliary space pre-conditioning employs the fact that any vector $\mathbf{u}$ that belongs to the space $\mathbf{V}(\mathcal{K})$ (including our solution vector $\mathbf{x}$) possesses a stable decomposition

$$\mathbf{u} = \mathbf{\Pi}^{\mathrm{curl}}\mathbf{z} + \nabla p + \mathbf{v}, \tag{24}$$

where $\mathbf{z} \in Q(\mathcal{K})^3$, $p \in Q(\mathcal{K})$, and $\mathbf{v} \in \mathbf{V}(\mathcal{K})$. Here, the finite element space $Q(\mathcal{K})$ is a discrete version of the space

$$\mathcal{H}_0^1(\Omega) = \{q \in L^2(\Omega) \mid \nabla q \in L^2(\Omega)^3 \text{ and } q = 0 \text{ on } \Gamma\},$$

containing all scalar functions with square-integrable gradient and homogeneous Dirichlet boundary values. We assume that the basis functions for the space $Q(\mathcal{K})$ are given by nodal Lagrange elements and $\dim(Q(\mathcal{K})) = m$, where m denotes the number of mesh vertices or nodes. The $\mathcal{H}_0(\mathrm{curl})$-conforming interpolation operator $\mathbf{\Pi}^{\mathrm{curl}} : \mathcal{H}_0(\mathrm{curl}, \Omega) \to \mathbf{V}(\mathcal{K})$ is obtained by integrating tangential field components along the edges of the mesh

$$\mathbf{\Pi}^{\mathrm{curl}}\mathbf{z} = \sum_{l \in \mathcal{E}(\mathcal{K})} \left(\int_l \mathbf{z} \cdot \mathbf{t}_l \, ds \right) \phi_l,$$

where $\mathcal{E}(\mathcal{K})$ is the set of all interior edges of $\mathcal{K}$ and $\phi_l \in \mathbf{V}(\mathcal{K})$ is the Nédélec basis function associated with edge l.

Identity (24) is the discrete Helmholtz decomposition representing the vector $\mathbf{u}$ as the sum of divergence- and curl-free fields as well as some residual $\mathbf{v}$. This residual $\mathbf{v}$ is added to account for the discretization error and make (24) hold for finite element spaces $\mathbf{V}(\mathcal{K})$ and $Q(\mathcal{K})$ defined on mesh $\mathcal{K}$ (Bürg 2013; Hiptmair & Xu 2007).

Decomposition (24) suggests that an efficient pre-conditioner for $\mathbf{B}$ can be constructed via the application of multilevel Schwarz methods (Zhang 1992; Smith *et al.* 2004) and auxiliary space theory (Xu 1996). Suppose there exists a small number k of auxiliary spaces $\{\mathbf{W}_k(\mathcal{K})\}$ equipped with mapping operators $\mathbf{\Pi}_k : \mathbf{W}_k(\mathcal{K}) \to \mathbf{V}(\mathcal{K})$ and $\mathbf{\Pi}_k^T : \mathbf{V}(\mathcal{K}) \to \mathbf{W}_k(\mathcal{K})$ and auxiliary functions $\mathbf{u}_k \in \mathbf{W}_k(\mathcal{K})$ such that

$$\mathbf{u} = \sum_k \mathbf{\Pi}_k \mathbf{u}_k, \qquad \forall \mathbf{u} \in \mathbf{V}(\mathcal{K}).$$

Then, the optimal pre-conditioner for system (23) can be written as

$$\mathbf{P}_\mathbf{B}^{-1} = \sum_k \mathbf{\Pi}_k \mathbf{B}_k^{-1} \mathbf{\Pi}_k^T, \tag{25}$$

where $\mathbf{B}_k = \mathbf{\Pi}_k^T \mathbf{B} \mathbf{\Pi}_k$ is a set of variationally defined operators for all auxiliary spaces. By employing eqs (24) and (25), we introduce the following auxiliary space pre-conditioner for matrix $\mathbf{B}$:

$$\mathbf{P}_\mathbf{B}^{-1} = \mathbf{R}\mathbf{D}^{-1}\mathbf{R}^T + \mathbf{G}\mathbf{L}^{-1}\mathbf{G}^T + \mathbf{S}^{-1}, \tag{26}$$

where $\mathbf{D} = \mathbf{R}^T \mathbf{B} \mathbf{R}$ and $\mathbf{L} = \mathbf{G}^T \mathbf{B} \mathbf{G}$. Matrix $\mathbf{R} \in \mathbb{R}^{n \times 3m}$ represents the interpolation operator from the vector nodal space $Q(\mathcal{K})^3$ to the Nédélec edge space $\mathbf{V}(\mathcal{K})$ and matrix $\mathbf{G} \in \mathbb{R}^{n \times m}$ represents a discrete gradient operator that maps the gradient of the nodal space $Q(\mathcal{K})$ into $\mathbf{V}(\mathcal{K})$:

$$\mathbf{R} : Q(\mathcal{K})^3 \to \mathbf{V}(\mathcal{K})$$

$$\mathbf{G} : \nabla Q(\mathcal{K}) \to \mathbf{V}(\mathcal{K}).$$

Further, $\mathbf{S}$ is some symmetric positive definite matrix acting as the smoothing operator. Note, eq. (26) is equivalent to eq. (25) with mappings being defined as $\mathbf{\Pi}_1 = \mathbf{R}$, $\mathbf{\Pi}_2 = \mathbf{G}$, and $\mathbf{\Pi}_3 = \mathbf{I}$.

However, pre-conditioner (26) is not practically very useful yet, because it involves the computation of three inversions. The key step that makes it practically appealing is that these inverses can be replaced with spectrally equivalent analogues without deteriorating the key properties of $\mathbf{P}_\mathbf{B}^{-1}$. Discrete operators $\mathbf{D}$ and $\mathbf{L}$ represent second-order elliptic problems (Kolev & Vassilevski 2009) and, hence, algebraic multigrid methods can be used to calculate the products of $\mathbf{D}^{-1}$ and $\mathbf{L}^{-1}$ with a coefficient vector in an optimal way (Briggs *et al.* 2000; Falgout 2006). The matrix $\mathbf{S}$ in eq. (26) denotes a smoother for $\mathbf{B}$ and is associated with term $\mathbf{v}$ in decomposition (24), which originates from the discretization error and consists of local oscillatory components. Neglecting this term results in significant deterioration of the performance (Hiptmair & Xu 2007) and, thus, it has to be taken into account. On the other hand, it is well-known that the Jacobi and Gauss-Seidel iterative schemes can eliminate such components very efficiently (Baker *et al.* 2011). Therefore, reasonable choices for the matrix $\mathbf{S}$ are $\mathbf{S} = \mathrm{diag}(\mathbf{B})$, corresponding to the Jacobi iteration, and

$$\mathbf{S} = \mathbf{L}_\mathbf{B}^{-1} + \mathbf{L}_\mathbf{B}^{-T} - \mathbf{L}_\mathbf{B}^{-1}\mathbf{B}\mathbf{L}_\mathbf{B}^{-T},$$

where $\mathbf{L}_\mathbf{B}$ denotes the lower triangular part of $\mathbf{B}$, corresponding to the Gauss–Seidel iteration. Then, a practical application of the inverse $\mathbf{S}^{-1}$ of $\mathbf{S}$ consists only of performing a few Jacobi or Gauss–Seidel iterations.

Summarizing, we solve problem (23) by using the CG method with pre-conditioner (26). Therefore, we never have to compute matrix $\mathbf{P}_\mathbf{B}^{-1}$ explicitly, but rather calculate its product with a vector. As can be seen, this reduces the application of $\mathbf{P}_\mathbf{B}^{-1}$ to a solution of three linear systems. Since these systems are a part of the pre-conditioner, they are solved only approximately. In this case, the matrices $\mathbf{L}^{-1}$ and $\mathbf{D}^{-1}$ are replaced with a single V-cycle of algebraic multigrid, and the product $\mathbf{S}^{-1}\mathbf{u}$ is obtained by applying a few Gauss–Seidel iterations with the matrix $\mathbf{B}$ to vector $\mathbf{u}$.

3.3 Why not using the complex formulation?

We have derived discrete system (12) by using two coupled real-valued eqs (5) and employing the FEM. Eq. (12) is the real-valued formulation of the complex system

$$\mathbf{K}\bar{\mathbf{E}}^s = \mathbf{f}, \tag{27}$$

where $\mathbf{K} = \mathbf{C} + i\mathbf{M}$ is a complex symmetric non-Hermitian matrix of size $n \times n$. System (27) can be derived from eq. (4). Although the two systems are equivalent in theory, some implications arise if one aims at finding efficient numerical methods to solve them (Day & Heroux 2001; Benzi & Bertaccini 2008). In this section, we give a rationale behind our choice to work with real-valued formulations rather than its complex analogue and consider both computational and technical reasons for the two pre-conditioners presented in Sections 3.1 and 3.2.

In case of using a direct solver, the $\mathbf{LDL}^H$ decomposition can be applied directly to the matrix $\mathbf{K}$ from eq. (27). This would eliminate the necessity to use the FGMRES method in combination with pre-conditioner (22) which requires a single factorization of matrix $\mathbf{B}$ from eq. (23) and the solution of $2N_{\mathrm{iter}}$ systems using backward and forward substitutions with N_{iter} being the number of FGMRES iterations. Thanks to the high efficiency of the block-diagonal pre-conditioner (22), only a few iterations are required to converge to an accurate solution. Therefore, factorization of the complex matrix $\mathbf{K}$ can be more expensive than factorization of the real matrix $\mathbf{B}$ and $2N_{\mathrm{iter}}$ additional forward and backward substitutions. This is due to the fact that complex number arithmetic is more expensive than real number arithmetic, both in terms of floating-point operations and memory access. In comparison with the direct solver MUMPS, which we use in this work, the application of FGMRES with block-diagonal pre-conditioner is—on average—two times faster than the direct factorization of the complex matrix. The other advantage of this approach comes from the fact that the factorization of $\mathbf{B}$ requires twice less memory than the factorization of $\mathbf{K}$.

For the auxiliary space pre-conditioner, we are not aware of any documented evidence that this method is readily transferable to the complex case and whether its efficiency would be preserved. Such analysis goes beyond the scope of this paper and the following is speculation. Assuming one can derive it for the complex case, this would again eliminate the necessity to use block-diagonal pre-conditioner (22). The performance of the auxiliary space pre-conditioner depends largely on the solvers required for the second-order elliptic auxiliary problems (see eq. 26). Whereas scalable and highly optimized algebraic multigrid codes exist for real-valued problems (Henson & Yang 2002; Gee *et al.* 2006), the extension of these methods to the complex case is still an active area of research (MacLachlan & Oosterlee 2008). An alternative is to use geometric multigrid, but the scalable and robust implementation of such a method for non-conforming unstructured meshes poses a challenging task and would require a substantial amount of work, whereas the algebraic methods used in this work do not require any information about the mesh (Falgout 2006). Taking into account these technicalities, the implementation of the auxiliary space pre-conditioner for complex formulation (27) is not straightforward. Furthermore, considering that the complex arithmetics is more expensive than the real, the overall performance of this pre-conditioner for the complex formulation is difficult to predict.

4 NUMERICAL EXPERIMENTS

A competitive method needs to be able to handle high conductivity contrasts such as the one between air and land, or steel (e.g. pipelines or borehole casing) and normal rocks. To test the robustness of the method, we assign a small nonzero conductivity of 10^{-9} S m^{-1} to the air layer for all subsequent models. In all cases, FGMRES is stopped as soon as the norm of the relative residual drops below a pre-defined tolerance, that is,

$$\frac{||\mathbf{AE} - \mathbf{f}||}{||\mathbf{f}||} < tol,$$

where $tol = 10^{-8}$.

In addition, we need to set a stopping tolerance when solving the linear systems of eqs (23). Recall, matrix $\mathbf{P}_A^{-1}$ is only an approximation to $\mathbf{A}^{-1}$ and, thus, solving (23) to a very small tolerance is not necessary. On the other hand, if the solution is too rough, more outer FGMRES iterations will be performed. For the direct solver-based pre-conditioner discussed in Section 3.1, we readily obtain very small relative residuals, often close to machine precision, for the linear systems of eqs (23). Even though we do not need this level of accuracy, there is often no way to control this when using generic direct solvers. For the approach presented in Section 3.2, we use the CG method with auxiliary space pre-conditioner and can control the stopping conditions. We have found that a relative residual of 10^{-3} ensures fast convergence of the FGMRES method and a relatively quick application of $\mathbf{P}_A^{-1}$.

We have implemented the discrete finite element formulation presented in Section 2.2 using the open-source finite element library deal.II (Bangerth *et al.* 2007). The implementation is fully distributed, i.e. all structures and linear algebra objects are almost uniformly subdivided among a predefined number of MPI processes (Bangerth *et al.* 2011). For efficient mesh partitioning, deal.II makes use of p4est (Burstedde *et al.* 2011). To store vectors and matrices, we utilize PETSc (Balay *et al.* 2013) which also provides a unified interface to several distributed direct solvers. For this work, we use MUMPS (Amestoy *et al.* 2006) to construct the direct solver-based block pre-conditioner presented in Section 3.1. For auxiliary space pre-conditioning as presented in Section 3.2, we have implemented an interface to hypre's implementation (Kolev & Vassilevski 2009) and adapted it for our non-conforming hexahedral meshes. This implementation is based on highly scalable smoothers (Baker *et al.* 2011) and algebraic multigrid methods (Henson & Yang 2002).

All numerical tests were run on a distributed platform consisting of 32 interconnected nodes, each equipped with two eight-core AMD Opteron 2.2 GHz processors. Since the platform was installed in 2010, the run-times reported here are likely higher than we would obtain with more recent hardware.

4.1 Verification of the numerical scheme

We verify correctness of the developed method by using the COMMEMI 3D-1 and 3D-2 models (Fig. 3) and compare our results with those presented by Ren *et al.* (2013).

Figs 4 and 5 compare numerical solutions for the COMMEMI 3D-1 model along the dashed line depicted in Fig. 3(a) for frequencies of 0.1 and 10 Hz, respectively. We have used a locally refined mesh with 1.1 million degrees of freedom for this test. An excellent agreement between the numerical solutions is evident. For the majority of data points, the two solutions differ by less than 1 per cent. Fig. 6 depicts numerical solutions for the larger scale COMMEMI 3D-2 model extracted along the line $y = 40$ km (Fig. 3b). Again, both solutions show very good agreement. Fig. 7 shows the mesh which was used to calculate the numerical solutions for the COMMEMI 3D-2 model. For this problem, we refined the
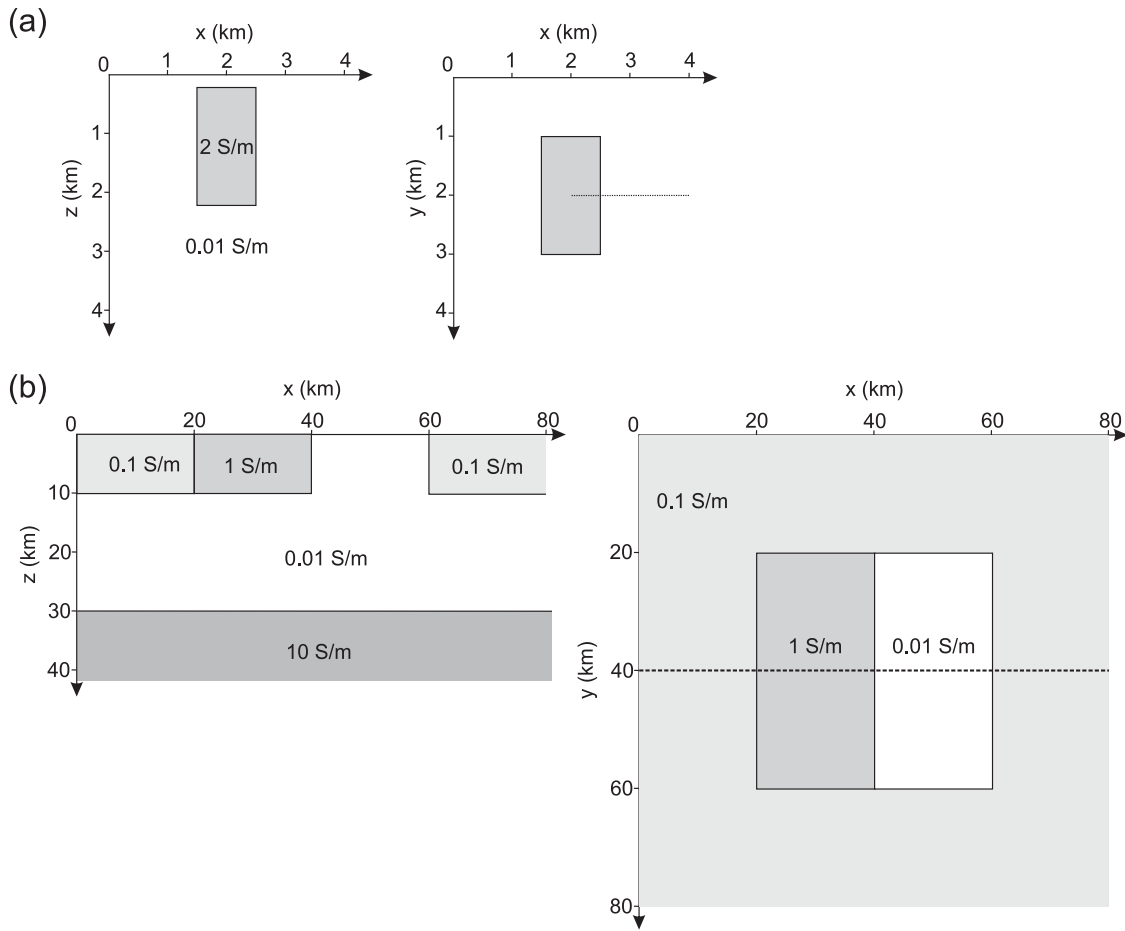
Figure 3. The COMMEMI 3D-1 (a) and 3D-2 (b) models (Zhdanov *et al.* 1997). The dotted lines indicate profiles along which apparent resistivity and phases are calculated.

mesh locally around the receiver points and ran several refinement cycles using the *a posteriori* error estimator presented in Section 2.3. The resulting mesh has 1.8 million degrees of freedom.

4.2 Robustness

In this section, we study the robustness of the solver. Particularly, we analyse how the auxiliary space block pre-conditioner presented in Section 3.2 performs with respect to frequency, size of the problem and number of MPI processes. We choose again the COMMEMI 3D-1 model for this experiment. In each subsequent test, we refine the model in one spatial dimension. This approximately doubles the number of DoFs. The problems are then run using a variable number of MPI processes such that there are about 200 000 DoFs per process. As the results for the two source polarizations are nearly identical, we present only one of them.

Table 1 reports the number of outer FGMRES iterations required to solve the original problem (13) and the average number of inner CG iterations spent in the solution of system (23) for a wide range of frequencies and variable problem sizes. First of all, we see that the number of FGMRES iterations is practically constant with respect to the problem size and always fulfills the condition $N_{\text{iter}} \ll N_{\text{DoFs}}$. This confirms that the condition number of the $\mathbf{P}_A^{-1}\mathbf{A}$ is independent from the size of system matrix. The number of inner CG iterations grows slowly with the size of the problem, but remains very small. This increase occurs because the auxiliary space pre-conditioner was optimized to deliver minimal run times, some-

times at the cost of slower convergence. It is possible to reduce $\bar{N}_{\text{iter}}^{\text{CG}}$ and keep the number of CG iterations constant by adjusting parameters of the auxiliary space pre-conditioner, but this results in considerably larger computation times. Consequently, we prefer a faster solver over optimal convergence.

Table 1 indicates that the number of FGMRES iterations slightly increases with frequency. Since the condition number of $\mathbf{P}_A^{-1}\mathbf{A}$ is bounded, it stops growing for frequencies higher than 10 Hz. Although not shown here, we tested the solver for frequencies up to 10^4 Hz and the number of iterations always stays below 20. Hence, changing the frequency over seven orders of magnitude results in a two-fold increase of FGMRES iterations. Thus, the solver proves to be robust and efficient. The slight increase for higher frequencies is partly compensated in terms of runtime by the fact that the inner CG solver requires less iterations to converge.

4.3 Adaptive mesh refinement

The ratio of volumes between the largest and smallest cells grows rapidly during adaptive mesh refinement. This can influence the condition number of the system matrix (Shewchuk 2002) and deteriorate convergence of the iterative solver. To investigate the influence of the adaptive mesh refinement on the solver, we apply the residual-based *a posteriori* error estimator from Section 2.3 on the COMMEMI 3D-1 model for a frequency of 0.1 Hz. We choose $\theta = 0.4$ (see Section 2.3 for a definition), that is, at every refinement step we extract and bisect a minimal subset of cells which
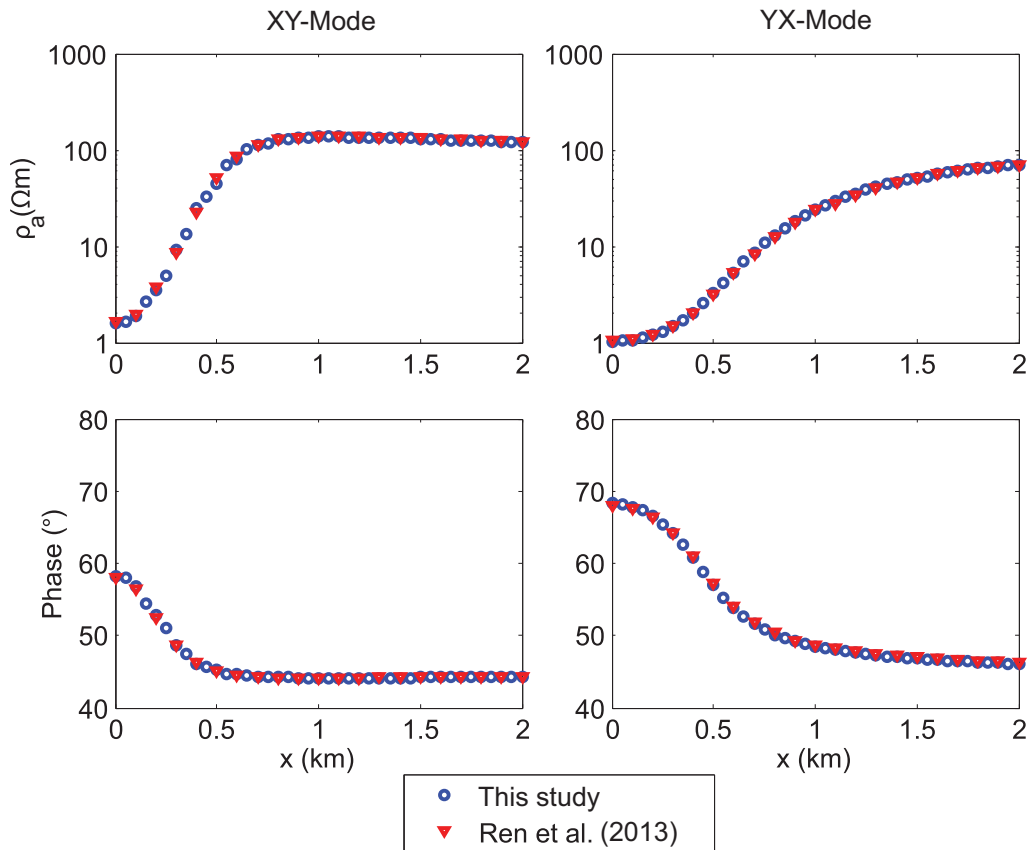
Figure 4. Apparent resistivities (ρ_a) and phases calculated for the COMMEMI 3D-1 model at a frequency of 0.1 Hz. The blue circles show the results of the new method, red triangles indicate the numerical solution of Ren *et al.* (2013).

account for about $0.4^2 \times 100 = 16$ per cent of the total error. As before, we measure the number of FGMRES iterations and average the number of inner CG iterations at every refinement cycle. Table 2 lists the number of iterations for 14 refinement cycles as well as the corresponding problem sizes. The number of FGMRES iterations stays constant which indicates the anticipated robust behaviour of the solver for locally refined meshes. For the inner CG solver, the number of iterations slightly increases for the last few refinement cycles. At that stage, the volume ratio between the largest and smallest hexahedra is larger than 10^{14} making the problem even more ill-conditioned. This explains the increase in iterations.

Fig. 8 shows the normalized estimated error for both uniform and adaptive refinement. During the first adaptive refinement steps the estimated error decreases much faster than for uniform refinement. This is due to the adaptive refinement strategy which first refines all the cells inside the single anomalous object of the COMMEMI 3D-1 model where the error is the largest. After the estimated error has decreased sufficiently the error reduction follows a more regular behaviour. For this particular model, adaptive local refinement and residual-based *a posteriori* error estimation requires 27-times fewer cells to achieve the same error as the uniform refinement strategy. Figs 9(a) and (b) show the normalized error indicators for the initial mesh and Figs 9(c) and (d) show the same quantity after twelve adaptive refinement steps. As anticipated, the largest errors are found near the regions of secondary sources or, equivalently, regions where $\sigma \neq \sigma^0$. After refinement, the error in the entire modelling domain is substantially reduced and exhibits a more uniform character.

4.4 Scalability

Next, we investigate scalability of our numerical scheme. Two types of tests are usually performed to assess scalability: (i) a model of fixed size is run on an increasing number of CPUs. If the run time decreases proportionally to the number of processes, the algorithm is said to be strongly scalable; (ii) both the size of the problem and the number of processes are increased in a way that the amount of work per process stays approximately constant. If also the run time stays (approximately) constant, the algorithm is called weakly scalable. Achieving good scalability requires not only the use of inherently scalable algorithms, but very often the implementation of algorithms has to be carefully tailored to a particular platform (Gropp 2005).

For geo-EM methods, the larger interest lies in achieving strong scalability which will considerably reduce run times by utilizing more resources. Fig. 10 shows results for the COMMEMI 3D-1 problem for a frequency of 0.1 Hz discretized with approximately 3.2 millions DoFs. For this experiment, we set the tolerance for the inner CG solver to 10^{-2}, because this resulted in minimal run times. The FGMRES solver required the same number of iterations as shown in Table 1. In addition to the solution time of system (13), we measured the time spent on system matrix and right-hand side vector assembly as well as the time required to compute the error estimator from Section 2.3. Commonly, these three components of the code constitute about 95 per cent of the total runtime. As anticipated, the solver represents the most expensive part followed by the error estimation and assembly routines. While assembly and
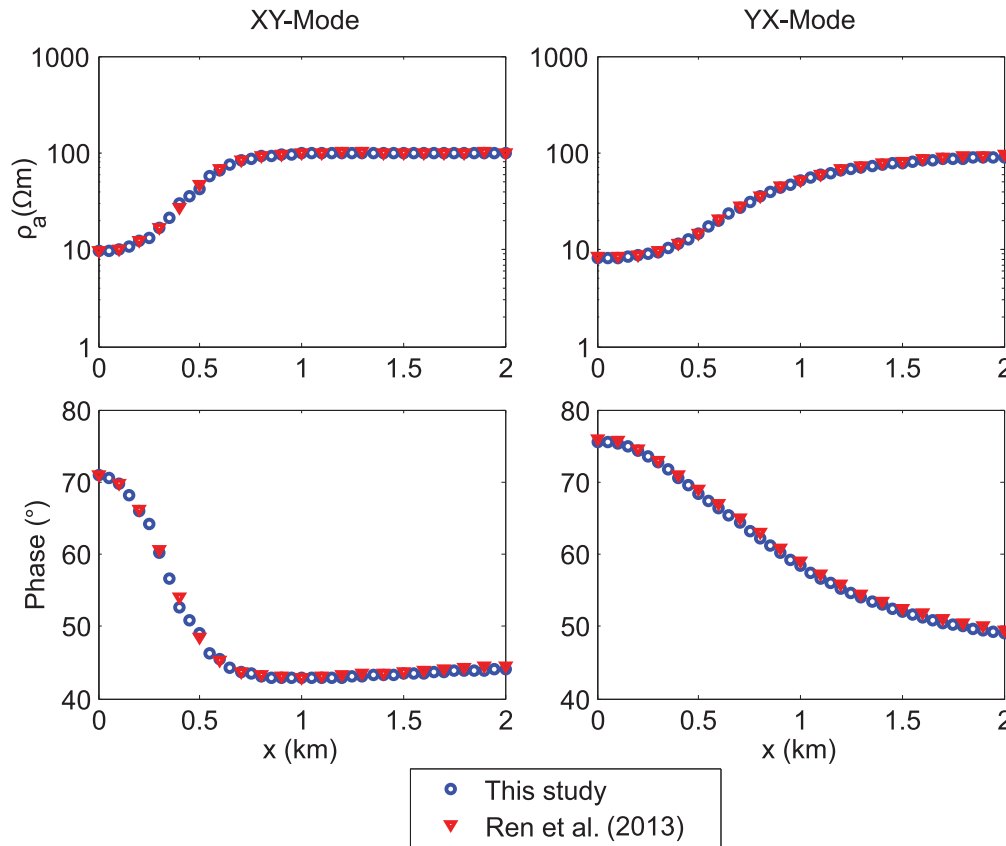
Figure 5. Apparent resistivities (ρ_a) and phases calculated for the COMMEMI 3D-1 model at a frequency of 10 Hz. The blue circles show the results of the new method, red triangles indicate the numerical solution of Ren *et al.* (2013).

error estimation scale optimally, the solver exhibits a slightly less efficient behaviour.

By increasing the number of processes, we subdivide our global problem into more local parts. This naturally increases the communication-to-computation ratio. We observe that once the local problem size drops below 10^5 DoFs (dashed vertical line in Fig. 10), the communication overhead becomes significant and scalability deteriorates. Consequently, we have to assign $>10^5$ DoFs per MPI process to preserve efficiency of the numerical scheme. Although the exact value of this threshold depends on the computational platform and the specific application, similar conclusions were inferred in other studies (Kolev & Vassilevski 2009; Kronbichler *et al.* 2012; Brown *et al.* 2013).

Note, we have shown run times for one source polarization. To calculate impedance tensor (14), we have to solve a second problem for the orthogonal polarization. Accordingly, the run times for the solver and the error estimator shown in Fig. 10 roughly double. In practice, however, we reuse many data and optimize the work-flow such that the solution for the second polarization comes at a smaller cost.

4.5 Comparison with other generic pre-conditioners

The solver presented in Section 3.2 consists of two nested loops. First, an outer FGMRES loop pre-conditioned with matrix (22). Second, an inner loop which involves the solution of two systems of type (23) by the CG solver with auxiliary space pre-conditioner (26). For brevity, we call this method FGMRES-PCG(AS) in this section. When considering the amount of work per iteration alone,

the FGMRES-PCG(AS) method seems unfavourable compared to different generic pre-conditioners, such as incomplete LU decomposition (ILU), successive over relaxation (SOR) or diagonal Jacobi scaling (Saad 2003). In contrast to FGMRES-PCG(AS), however, the number of iterations required by a Krylov method preconditioned with one of these generic techniques depends strongly on the frequency, conductivity contrasts and problem size (Haber & Ascher 2001; Ernst & Gander 2011; Um *et al.* 2013). Furthermore, widely used ILU(k) pre-conditioners (where k means the level of fill-in as defined in Benzi (2002)) have limited scalability and efficient parallel implementations are difficult to develop without deteriorating the convergence rate of the employed Krylov subspace method (Benzi 2002).

We compare the performance of FGMRES-PCG(AS) with the BiCGStab method pre-conditioned with ILU(2) and symmetric SOR. Most of the 3D MT modelling codes work with complex formulation (27) whose system matrix is symmetric non-Hermitian positive definite. In contrast, the matrix in our approach is indefinite and most of the generic pre-conditioners are either not applicable or much more expensive in this case. Therefore, the BiCGStab method is applied to complex eq. (27), whereas FGMRES-PCG(AS) is applied to its real equivalent in eq. (13). Table 3 shows the number of iterations and the computational time for a few combinations of solvers and preconditioners, and different problem sizes. The COMMEMI 3D-1 model with a frequency of 10 Hz were used for this test. Similar to the previous tests, the solvers were run until the relative residual dropped below 10^{-8} or the number of iterations exceeded 5000. Clearly, a single iteration of FGMRES-PCG(AS) is more
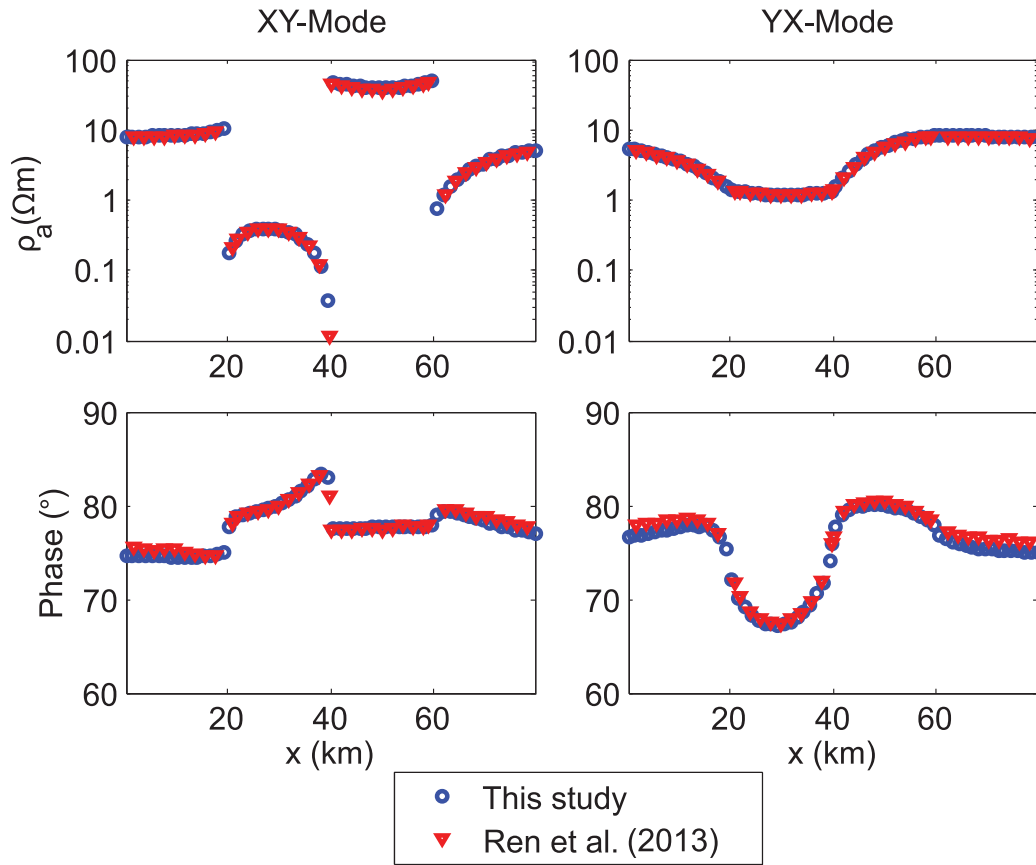
Figure 6. Apparent resistivities (ρ_a) and phases calculated for the COMMEMI 3D-2 model at a frequency of 0.001 Hz. The blue circles are solution obtained using the presented method and the red triangles are the numerical solution from Ren *et al.* (2013).

expensive than BiCGStab with either ILU or SOR pre-conditioning. Nevertheless, since FGMRES-PCG(AS) requires a nearly constant number of iterations independent of the problem size, it outperforms generic pre-conditioners already for problems of medium size.

Presently, most of the widely used 3-D MT modelling codes are based on finite-difference (FD) techniques and use generic pre-conditioners with divergence correction to solve arising systems of linear equations (e.g. Kelbert *et al.* 2014). Tests show that the generic pre-conditioners such as ILU usually exhibit better convergence and performance for system matrices which originate from FD methods when compared to FE methods. This is likely due to greater sparsity. Therefore, for problems of medium size, which do not necessitate the usage of unstructured meshes or adaptivity, FD methods and generic pre-conditioners are likely to be faster than FGMRES-PCG(AS). However, considering problems with 10^7–10^8 DoFs and the necessity to use distributed platforms, the generic pre-conditioners are not competitive due to poor scalability. In contrast, the scalability and performance of FGMRES-PCG(AS) are primarily restricted by the performance of the algebraic multigrid solvers (AMG) which are known to be scalable. Therefore, further progress in AMG will also enhance performance of the auxiliary space pre-conditioner.

4.6 Direct or iterative?

As discussed in Section 3.1, we can use MUMPS' $\mathbf{LDL}^T$ decomposition instead of the CG method with the auxiliary space pre-conditioner to solve the problems (23). Direct solvers are known to be very robust, but the auxiliary space pre-conditioner presented

in Section 3.2 proved to be robust as well. Moreover, in contrast to a direct solver, the auxiliary space pre-conditioner is scalable. For this reason, the only argument in favour of direct solver-based pre-conditioner (22) is its ability to solve for multiple right-hand sides (RHS) quickly. In this section, we present a detailed analysis of this scenario.

The total time required by the FGMRES solver with direct solver-based pre-conditioner (22) is $T_{\text{direct}}^0 = T_{\text{f}} + 2N_{\text{iter}}T_{\text{tri}} + T_{\text{fgmres}}$, where T_{f} is the time needed to factorize matrix $\mathbf{B}$, T_{tri} the time for triangular solves or, equivalently, forward and backward substitutions, T_{gmres} the time for internal FGMRES calculations and N_{iter} the number of FGMRES iterations. For subsequent right-hand side vectors, we reuse the already existing factorization and the expression becomes $T_{\text{direct}}^i = 2N_{\text{iter}}T_{\text{tri}} + T_{\text{fgmres}}$ for $i \in \{1, \ldots, N_{\text{RHS}}\}$. Since $T_{tri} \ll T_f$, this reduction in time is significant. For the approach presented in Section 3.2, the solution time is given by $T_{\text{asp}} = 2N_{\text{iter}}T_{CG} + T_{\text{fgmres}}$, where T_{CG} is the time required to solve a single system (23) using the CG method with the auxiliary space pre-conditioner. This means, if $T_{\text{tri}} < T_{\text{CG}}$, then there is a minimal number of right hand sides N_* such that the direct solver-based pre-conditioner outperforms the iterative one. A necessary condition for this is $N_* T_{\text{asp}} > T_{\text{f}}$. However, we know that T_{f} grows super-linearly (Pardo *et al.* 2012) whereas T_{asp} theoretically exhibits linear growth. Accordingly, N_* will increase with the size of the problem.

Fig. 11 shows the solution time for two different pre-conditioning strategies for the COMMEMI 3D-1 model for a frequency of 0.1 Hz and a comprehensive number of discretizations. It is evident that for a single RHS the auxiliary space pre-conditioner becomes faster for problems with more than about 400 000 unknowns. However,
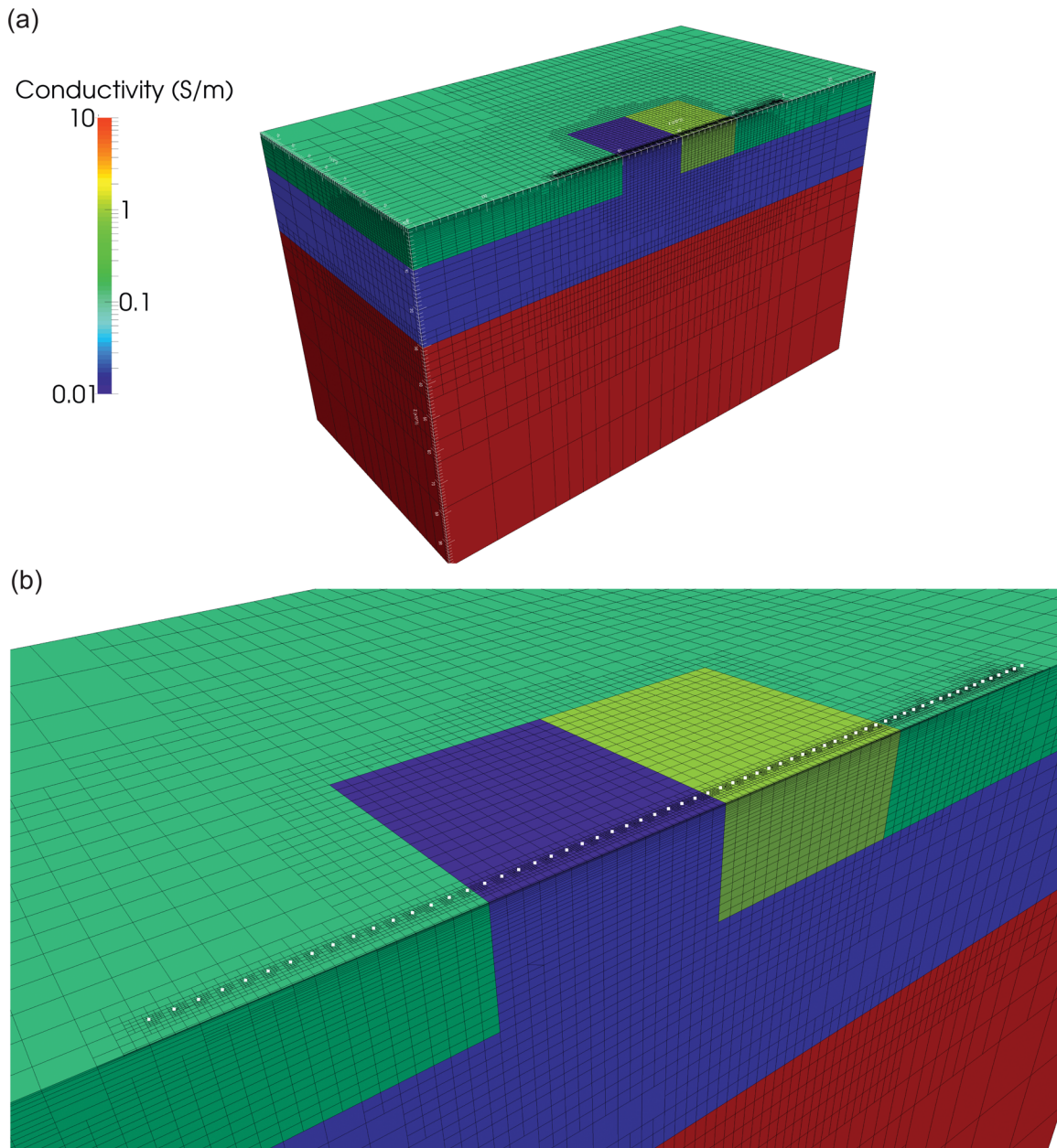
(a)

Conductivity (S/m)

(b)

Figure 7. (a) Section through the centre of the COMMEMI 3D-2 model showing the locally refined mesh which we have used to compute the numerical solution shown in Fig. 6. (b) Enlarged view of the central part. The white dots indicate receivers.

Table 1. Numerical results for COMMEMI 3D-1 model and variable frequencies. N_{iter} and $\bar{N}_{\text{iter}}^{\text{CG}}$ denote the number of outer FGMRES and the average number of inner CG iterations, respectively.

# MPI processes	# DoFs	Frequency (Hz)							
		0.001		0.1		10		1000	
		N_{iter}	$\bar{N}_{\text{iter}}^{\text{CG}}$	N_{iter}	$\bar{N}_{\text{iter}}^{\text{CG}}$	N_{iter}	$\bar{N}_{\text{iter}}^{\text{CG}}$	N_{iter}	$\bar{N}_{\text{iter}}^{\text{CG}}$
1	206 978	7	7	16	7	18	4	18	3
2	408 628	7	9	16	7	20	6	18	4
4	809 884	7	9	16	7	19	6	18	3
8	1 605 124	7	9	16	7	20	6	18	4
16	3 189 224	7	13	16	10	20	8	20	6
32	6 349 304	7	11	16	10	20	8	20	6
64	12 640 520	7	14	16	10	20	8	20	5
128	25 197 520	9	13	16	13	20	11	20	7

Table 2. Numerical results for COMMEMI 3D-1 model and 14 adaptive refinement steps. N_{iter} and $\bar{N}_{CG}$ denote the numbers of outer FGMRES iterations and the average number of inner CG iterations for the block pre-conditioner, respectively.

Refinement step	# DoFs	N_{iter}	$\bar{N}_{iter}^{CG}$
0	53 618	14	6
3	60 350	16	6
7	180 894	16	7
9	524 144	16	11
11	1 500 220	16	12
14	7 451 676	16	12
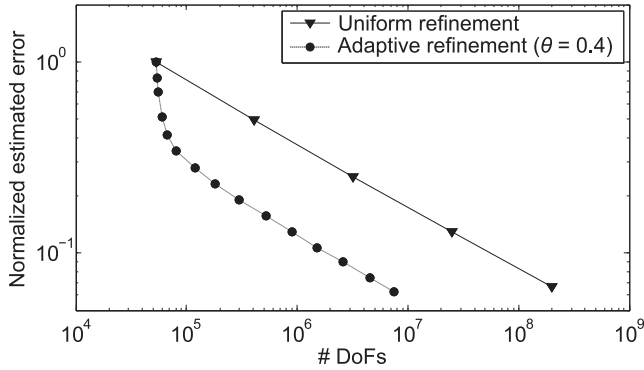
Figure 8. Normalized estimated errors for problems of variable size obtained by using uniform and adaptive mesh refinement for the COMMEMI 3D-1 model for a frequency of 0.1 Hz.

for multiple RHS vectors the situation is not that obvious. The green line indicates that for subsequent RHS vectors, where the existing factorization can be reused for the diagonal blocks in eq. (22), the total solution time is smaller than for the auxiliary space pre-conditioner. We calculated a log-linear fit for these curves and derived an estimate for the time complexity based on the slope of the
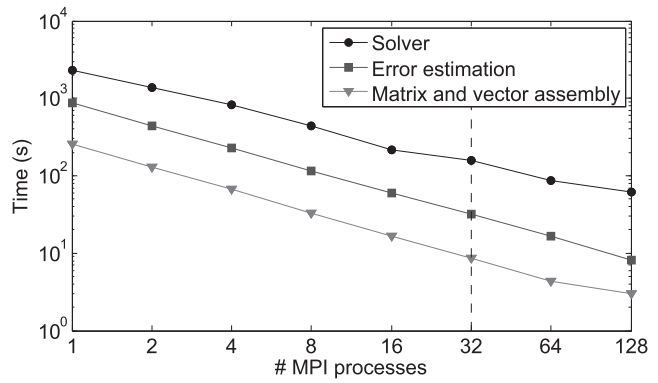

Figure 10. Results of the strong scaling experiment for the COMMEMI 3D-1 model with 3.2 millions DoFs. The vertical dashed line indicates the threshold of 10^5 DoFs per MPI process.

curve (values are reported in the legend of Fig. 11). As anticipated, the direct solver-based pre-conditioner exhibits quadratic growth (slope = 2.04). The green curve, indicating the time for forward and backward substitutions, grows at a rate of 1.45—much higher than linear. Finally, the auxiliary space pre-conditioner, shown by the blue curve, grows with slope 1.2. While this slope is larger than the optimal linear growth, it is considerably better than the other two.

We can determine N_* for a given problem size using these estimated quantities. The results in Fig. 12 show that even for relatively large problems, the direct solver can be faster if one needs to solve a problem for multiple RHS vectors. However, due to its lack of scalability and much better scaling of the auxiliary space pre-conditioner, this curve will shift upwards when more MPI processes are used. These considerations become particularly relevant in the context of inverse problems (Grayver *et al.* 2013).

So far, we have not considered the amount of memory used by the two pre-conditioning techniques. Fig. 13 closes this gap and shows
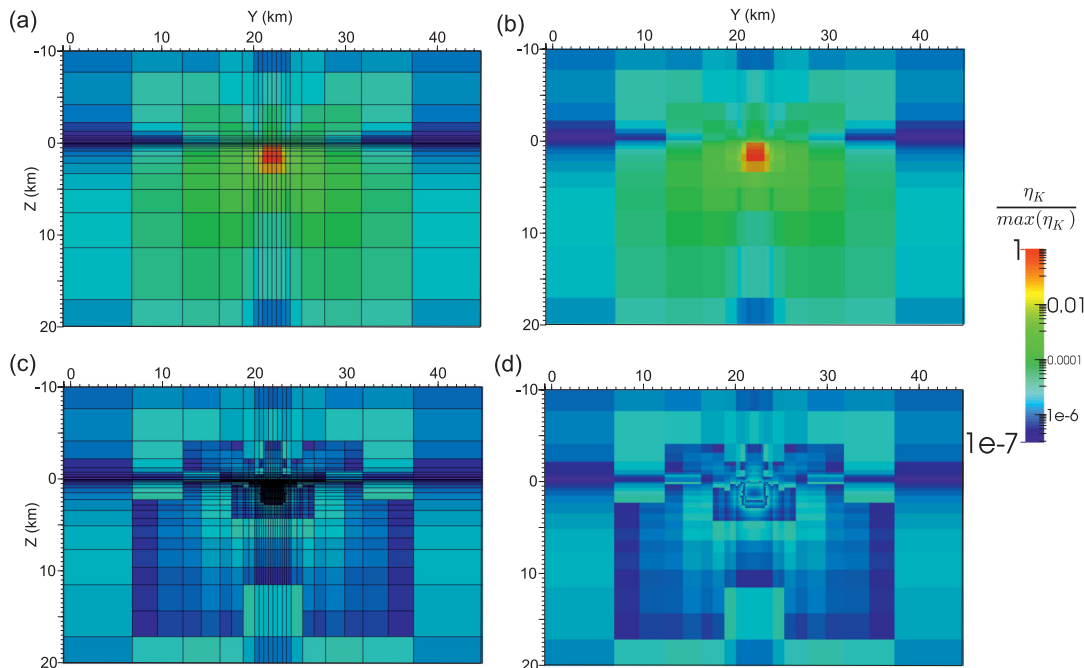

Figure 9. Normalized error indicator for the initial coarse grid of the COMMEMI 3D-1 model (a) with and (b) without grid lines. Plots (c) and (d) show the same quantity after 12 adaptive refinement steps. Shown is the section through the centre of the model.

Table 3. Iteration count and computational time for COMMEMI 3D-1 model and frequency of 10 Hz. The left column lists the number of complex DoFs which should be multiplied by two to get the respective real problem size for FGMRES-PCG(AS). The remaining three columns show the results for BiCGStab with two generic pre-conditioners and FGMRES-PCG(AS). All tests were run on a single AMD Opteron 6134 CPU core.

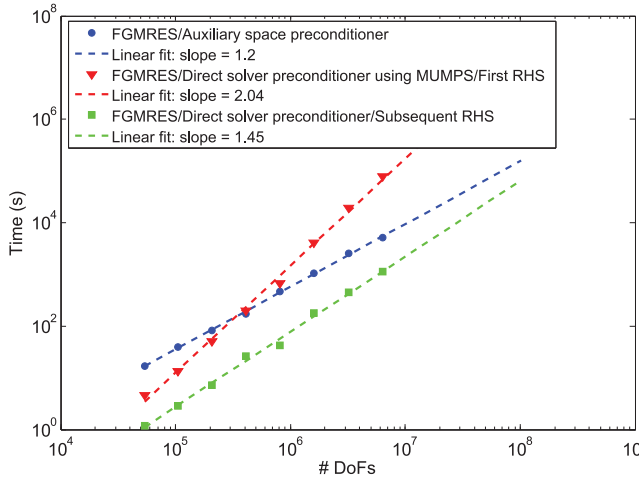| Number of complex DoFs	BiCGStab-SOR		BiCGStab-ILU(2)		FGMRES-PCG(AS)	
	N_{iter}	Time (s)	N_{iter}	Time (s)	N_{iter}	Time (s)
26 809	201	8	700	51	16	17
103 489	319	58	1322	601	18	82
204 314	639	240	2474	2695	18	174
802 562	1216	2400	4517	25 314	18	1039
1 594 612	2039	8911	>5000	–	18	2571

Figure 11. Solution time for the two pre-conditioner strategies—direct solver MUMPS (red) and CG method with auxiliary space pre-conditioning (blue)—for the block-diagonal systems in eq. (22) and various problem sizes. The green line shows the time needed to get the solution for subsequent right-hand side (RHS) vectors when direct solver is employed and the existing factorization is reused. All problems were run on a single CPU.
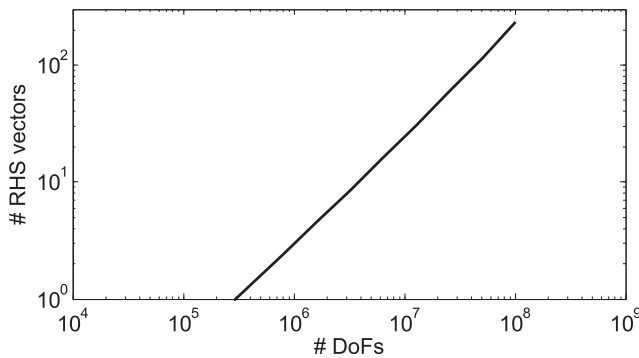

Figure 12. Number of RHS vectors for which the direct solver pre-conditioner outperforms the iterative auxiliary space pre-conditioner versus the problem size.

the total memory consumption for both approaches and its predicted slopes. Clearly, the memory complexity grows non-linearly with respect to the number of DoFs for the direct solver, whereas it exhibits linear growth for the auxiliary space pre-conditioner.

5 CONCLUSIONS

We have presented a robust and scalable numerical scheme to solve large-scale 3D geo-EM problems in frequency domain. We use the
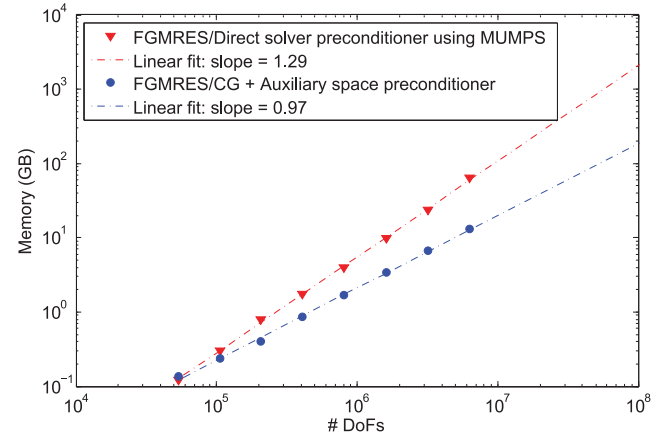

Figure 13. Total memory consumption of the numerical scheme using the two pre-conditioning techniques and their approximate slopes.

FEM to discretize the resulting weak formulation. The most challenging part of this scheme is the solution of a large system of linear equations. We reformulated the original complex-valued problem as two coupled real-valued equations and employed the FGMRES method with a very efficient block-diagonal pre-conditioner for the resulting system. This combination proved to be robust with respect to large conductivity contrasts, a wide frequency range and variable problem sizes with more than 10^8 degrees of freedom. The efficiency of the solver was confirmed for spatially stretched and locally refined meshes. We demonstrated this by introducing a residual-based *a posteriori* error estimator and an adaptive mesh refinement algorithm to create problem-adapted discretizations. In all cases, the solver required only a small number of iterations to converge. For models and settings considered, the performance of the solver did not deteriorate for low frequencies, which eliminates the necessity for divergence correction techniques.

A single application of the pre-conditioner entails the solution of two smaller real-valued symmetric systems of equations. Since these systems involve the operator $\nabla \times$, they are difficult to solve efficiently with generic iterative methods. Alternatively, they can be tackled by using either a direct solver or an iterative CG solver in combination with the recently developed auxiliary space pre-conditioner. The latter provides a scalable and robust approach to reformulate Maxwell's equations into a series of simpler problems which are solved using state of the art algebraic multigrid methods. Despite large amount of work per iteration required by this method, its robustness and scalability make it more suitable for large-scale problems than generic pre-conditioners such as ILU or SOR. On the other hand, the direct solver-based pre-conditioner is capable of delivering solutions for multiple right-hand side vectors at low

cost, since the factorization can be reused. We have presented a time analysis and an estimation for the number of right hand sides necessary to outperform the CG solver with the auxiliary space pre-conditioner. We showed that the direct solver-based pre-conditioner exhibits non-linear memory complexity, whereas the auxiliary space pre-conditioner scales linearly with respect to the number of degrees of freedom.

To utilize the performance offered by modern computational platforms, our code has been implemented in a fully distributed fashion. More importantly, the most expensive components of the code, such as solvers, matrix and vector assembly and error estimation, exhibit strong scalability as long as each MPI process possesses enough degrees of freedom to alleviate communication overburden. We have found this threshold to be approximately 10^5 DoFs. For further efficiency gain, we have also parallelized the code over frequencies.

REFERENCES

Adams, M.F., 2012. A low memory, highly concurrent multigrid algorithm, preprint (arXiv:1207.6720).

Amestoy, P.R., Guermouche, A., L'Excellent, J.-Y. & Pralet, S., 2006. Hybrid scheduling for the parallel solution of linear systems, *Parallel Comput.,* **32**(2), 136–156.

Arnold, D.N., Falk, R.S. & Winther, R., 2000. Multigrid in H(div) and H(curl), *Numer. Math.,* **85**(2), 197–217.

Baker, A.H., Falgout, R.D., Kolev, T.V. & Yang, U.M., 2011. Multigrid smoothers for ultraparallel computing, *SIAM J. Scient. Comput.,* **33**(5), 2864–2887.

Baker, A.H., Falgout, R.D., Kolev, T.V. & Yang, U.M., 2012. Scaling hypre's multigrid solvers to 100,000 cores, in *High-Performance Scientific Computing,* pp. 261–279, eds Berry, M.W., Gallivan, K.A., Gallopoulos, E., Grama, A., Philippe, B., Saad, Y. & Saied, F., Springer.

Balay, S. *et al.*, 2013. PETSc Users Manual Revision 3.4.

Bangerth, W. & Kayser-Herold, O., 2009. Data structures and requirements for *hp* finite element software, *ACM Trans. Math. Software,* **36**(4), doi:10.1145/1486525.1486529.

Bangerth, W., Hartmann, R. & Kanschat, G., 2007. deal.II—a general-purpose object-oriented finite element library, *ACM Trans. Math. Software,* **33**(4), doi:10.1145/1268776.1268779.

Bangerth, W., Burstedde, C., Heister, T. & Kronbichler, M., 2011. Algorithms and data structures for massively parallel generic adaptive finite element codes, *ACM Trans. Math. Software (TOMS),* **38**(2), 1–28.

Beck, R., Hiptmair, R., Hoppe, R.H. & Wohlmuth, B., 2000. Residual based a posteriori error estimators for eddy current computation, *Math. Modell. Numer. Anal.,* **34**(01), 159–182.

Benzi, M., 2002. Preconditioning techniques for large linear systems: a survey, *J. Comput. Phys.,* **182**(2), 418–477.

Benzi, M. & Bertaccini, D., 2008. Block preconditioning of real-valued iterative algorithms for complex linear systems, *IMA J. Numer. Anal.,* **28**(3), 598–618.

Berdichevskii, M.M.N. & Dmitriev, V.I., 2008. *Models and Methods of Magnetotellurics,* Springer.

Börner, R.-U., 2010. Numerical modelling in geo-electromagnetics: advances and challenges, *Surv. Geophys.,* **31**(2), 225–245.

Bossavit, A., 1998. *Comput. Electromagnet.,* Academic Press.

Brenner, S. & Scott, L., 1994. *The Mathematical Theory of Finite Element Methods,* Springer.

Briggs, W.L., Henson, V.E. & McCormick, S.F., 2000. *A Multigrid Tutorial,* 2nd edn, Vol. 72, Society for Industrial and Applied Mathematics.

Brossier, R., Etienne, V., Operto, S. & Virieux, J., 2010. Frequency-domain numerical modelling of visco-acoustic waves based on finite-difference and finite-element discontinuous Galerkin methods, in *Acoustic Waves,* pp. 125–158, ed. Dissanayake, D., Sciyo.

Brown, J., Smith, B. & Ahmadia, A., 2013. Achieving textbook multigrid efficiency for hydrostatic ice sheet flow, *SIAM J. Scient. Comput.,* **35**(2), B359–B375.

Bürg, M., 2012. A residual-based a posteriori error estimator for the *hp*-finite element method for Maxwell's equations, *Appl. Numer. Math.,* **62**(8), 922–940.

Bürg, M., 2013. Convergence of an automatic *hp*-adaptive finite element strategy for Maxwell's equations, *Appl. Numer. Math.,* **72**, 188–206.

Burstedde, C., Wilcox, L.C. & Ghattas, O., 2011. p4est: scalable algorithms for parallel adaptive mesh refinement on forests of octrees, *SIAM J. Scient. Comput.,* **33**(3), 1103–1133.

Chen, J., Chen, Z., Cui, T. & Zhang, L.-B., 2010. An adaptive finite element method for the eddy current model with circuit/field couplings, *SIAM J. Scient. Comput.,* **32**(2), 1020–1042.

Chen, Z., Wang, L. & Zheng, W., 2007. An adaptive multilevel method for time-harmonic Maxwell equations with singularities, *SIAM J. Scient. Comput.,* **29**(1), 118–138.

Day, D. & Heroux, M., 2001. Solving complex-valued linear systems via equivalent real formulations, *SIAM J. Scient. Comput.,* **23**(2), 480–498.

Demkowicz, L., Kurtz, J., Pardo, D., Paszyński, M., Rachowicz, W. & Zdunek, A., 2008. *Computing with hp-Adaptive Finite Elements,* Vol. 2, Chapman & Hall/CRC.

Dörfler, W., 1996. A convergent adaptive algorithm for Poisson's equation, *SIAM J. Numer. Anal.,* **33**, 1106–1124.

Egbert, G.D., 2012. Hybrid conjugate gradient-Occam algorithms for inversion of multifrequency and multitransmitter EM data, *Geophys. J. Int.,* **190**(1), 255–266.

Ernst, O.G. & Gander, M.J., 2011. Why it is difficult to solve Helmholtz problems with classical iterative methods, *Numer. Anal. Multiscale Probl.,* **83**, 325–361.

Falgout, R.D., 2006. An introduction to algebraic multigrid, *Comput. Sci. Eng.,* **8**(6), 24–33.

Farquharson, C.G. & Miensopust, M.P., 2011. Three-dimensional finite-element modelling of magnetotelluric data with a divergence correction, *J. appl. Geophys.,* **75**(4), 699–710.

Gee, M., Siefert, C., Hu, J., Tuminaro, R. & Sala, M., 2006. ML 5.0 smoothed aggregation user's guide, Tech. Rep. SAND2006-2649, Sandia National Laboratories.

Golub, G.H. & van Loan, C.F., 1996. *Matrix Computations,* 3rd edn, Johns Hopkins Univ. Press.

Gould, N.I.M., Scott, J.A. & Hu, Y., 2007. A numerical evaluation of sparse direct solvers for the solution of large sparse symmetric linear systems of equations, *ACM Trans. Math. Software,* **33**(2), doi:10.1145/1236463.1236465.

Grayver, A.V., Streich, R. & Ritter, O., 2013. Three-dimensional parallel distributed inversion of CSEM data using a direct forward solver, *Geophys. J. Int.,* **193**(3), 1432–1446.

Gropp, W., 2005. Issues in accurate and reliable use of parallel computing in numerical programs, in *Accuracy and Reliability in Scientific Computing,* pp. 253–263, ed. Einarsson, B., Cambridge Univ. Press.

Haber, E. & Ascher, U.M., 2001. Fast finite volume simulation of 3D electromagnetic problems with highly discontinuous coefficients, *SIAM J. Scient. Comput.,* **22**(6), 1943–1961.

Haber, E. & Heldmann, S., 2007. An octree multigrid method for quasi-static Maxwell's equations with highly discontinuous coefficients, *J. Comput. Phys.,* **223**(2), 783–796.

Henson, V.E. & Yang, U.M., 2002. BoomerAMG: a parallel algebraic multigrid solver and preconditioner, *Appl. Numer. Math.,* **41**(1), 155–177.

Hiptmair, R., 1998. Multigrid method for Maxwell's equations, *SIAM J. Numer. Anal.,* **36,** 204–225.

Hiptmair, R., 2002. Finite elements in computational electromagnetism, *Acta Numer.,* **11,** 237–339.

Hiptmair, R. & Xu, J., 2007. Nodal auxiliary space preconditioning in H (curl) and H (div) spaces, *SIAM J. Numer. Anal.,* **45**(6), 2483–2509.

Jin, J., 2002. *The Finite Element Method in Electromagnetics,* Wiley.

Kelbert, A., Meqbel, N., Egbert, G.D. & Tandon, K., 2014. ModEM: a modular system for inversion of electromagnetic geophysical data, *Comput. Geosci.,* **66,** 40–53

Kirk, B.S., Peterson, J.W., Stogner, R.H. & Carey, G.F., 2006. libmesh: a C++ library for parallel adaptive mesh refinement/coarsening simulations, *Eng. Comput.,* **22**(3–4), 237–254.

Kolev, T.V. & Vassilevski, P., 2009. Parallel auxiliary space AMG for H(curl) problems, *J. Comput. Math.,* **27**(5), 604–623.

Kronbichler, M., Heister, T. & Bangerth, W., 2012. High accuracy mantle convection simulation through modern numerical methods, *Geophys. J. Int.,* **191**(1), 12–29.

Li, X.S., 2005. An overview of SuperLU: algorithms, implementation, and user interface, *ACM Trans. Math. Software,* **31**(3), 302–325.

Logg, A. & Wells, G.N., 2010. Dolfin: automated finite element computing, *ACM Trans. Math. Software,* **37**(2), 20:1–20:28.

Mackie, R.L. & Madden, T.R., 1993. Three-dimensional magnetotelluric inversion using conjugate gradients, *Geophys. J. Int.,* **115**(1), 215–229.

MacLachlan, S.P. & Oosterlee, C.W., 2008. Algebraic multigrid solvers for complex-valued matrices, *SIAM J. Scient. Comput.,* **30**(3), 1548–1571.

Maurer, D. & Wieners, C., 2011. Parallel block decomposition method for distributed finite element matrices, *Parallel Comput.,* **37,** 742–758.

Monk, P., 2003. *Finite Element Methods for Maxwell's Equations,* Oxford Univ. Press

Mulder, W., 2006. A multigrid solver for 3D electromagnetic diffusion, *Geophys. Prospect.,* **54**(5), 633–649.

Nam, M.J., Kim, H.J., Song, Y., Lee, T.J., Son, J.-S. & Suh, J.H., 2007. 3D magnetotelluric modelling including surface topography, *Geophys. Prospect.,* **55**(2), 277–287.

Nechaev, O.V., Shurina, E.P. & Botchev, M.A., 2008. Multilevel iterative solvers for the edge finite element solution of the 3D Maxwell equation, *Comput. Math. Applicat.,* **55**(10), 2346–2362.

Newman, G.A. & Alumbaugh, D.L., 1995. Frequency-domain modelling of airborne electromagnetic responses using staggered finite differences, *Geophys. Prospect.,* **43,** 1021–1041.

Operto, S., Virieux, J., Amestoy, P., L'Excellent, J.-Y., Giraud, L. & Ali, H.B.H., 2007. 3D finite-difference frequency-domain modeling of visco-acoustic wave propagation using a massively parallel direct solver: a feasibility study, *Geophysics,* **72**(5), SM195–SM211.

Pardo, D., Paszynski, M., Collier, N., Alvarez, J., Dalcin, L. & Calo, V., 2012. A survey on direct solvers for Galerkin methods, *SeMA J.,* **1**(57), 107–134.

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, *Geophys. J. Int.,* **193**(2), 678–693.

Ren, Z., Kalscheuer, T., Greenhalgh, S. & Maurer, H., 2013. A goal-oriented adaptive finite-element approach for plane wave 3-D electromagnetic modelling, *Geophys. J. Int.,* **194**(2), 700–718.

Saad, Y., 2003. *Iterative Methods for Sparse Linear Systems,* 2nd edn, Society for Industrial and Applied Mathematics.

Sasaki, Y. & Meju, M.A., 2009. Useful characteristics of shallow and deep marine CSEM responses inferred from 3D finite-difference modeling, *Geophysics,* **74**(5), F67–F76.

Schenk, O. & Gärtner, K., 2004. Solving unsymmetric sparse systems of linear equations with PARDISO, *Future Generat. Comput. Syst.,* **20**(3), 475–487.

Schwarzbach, C., 2009. Stability of finite element solutions to Maxwell's equations in frequency domain, *PhD thesis,* TU Bergakademie Freiberg, Freiberg, Germany.

Schwarzbach, C., Börner, R.-U. & Spitzer, K., 2011. Three-dimensional adaptive higher order finite element simulation for geo-electromagnetics—a marine CSEM example, *Geophys. J. Int.,* **187**(1), 63–74.

Shewchuk, J., 2002. What is a good linear element? Interpolation, conditioning, and quality measures, in *Eleventh International Meshing Roundtable,* pp. 115–126, ed. Chrisochoides, N., Sandia National Laboratories.

Smith, B., Bjorstad, P. & Gropp, W., 2004. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations,* Cambridge Univ. Press.

Smith, T.J., 1996. Conservative modeling of 3-D electromagnetic fields. Part II: biconjugate gradient solution as an accelerator, *Geophysics,* **61**(5), 1319–1324.

Šolín, P., Segeth, K. & Doležel, I., 2004. *Higher-Order Finite Element Methods,* Chapman & Hall/CRC.

Streich, R., 2009. 3D finite-difference frequency-domain modeling of controlled-source electromagnetic data: direct solution and optimization for high accuracy, *Geophysics,* **74,** F95–F105.

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, *Geophys. J. Int.,* **193**(3), 1460–1473.

Xu, J., 1996. The auxiliary space method and optimal multigrid preconditioning techniques for unstructured grids, *Computing,* **56**(3), 215–235.

Zaglmayr, S., 2006. High order finite element methods for electromagnetic field computation, *PhD thesis,* Johannes Kepler University, Linz, Austria.

Zhang, X., 1992. Multilevel schwarz methods, *Numer. Math.,* **63**(1), 521–539.

Zhdanov, M., Varentsov, I.M., Weaver, J., Golubev, N. & Krylov, V., 1997. Methods for modelling electromagnetic fields results from COMMEMI—the international project on the comparison of modelling methods for electromagnetic induction, *J. appl. Geophys.,* **37**(3), 133–271.