Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig

A parallel goal-oriented adaptive finite element method for 2.5D electromagnetic modeling

(revised version: June 2011)

by

Kerry Key, and Jeffrey Ovall

Preprint no.: 40 2010

submitted to *Geophys. J. Int.*

A parallel goal-oriented adaptive finite element method for 2.5D electromagnetic modeling

Kerry Key[1] and Jeffrey Ovall[2]

[1] *Scripps Institution of Oceanography, La Jolla, CA, USA. E-mail: kkey@ucsd.edu*

[2] *University of Kentucky, Lexington, KY, USA. E-mail: jovall@ms.uky.edu*

SUMMARY

We present a parallel goal-oriented adaptive finite element method that can be used to rapidly compute highly accurate solutions for 2.5D controlled-source electromagnetic (CSEM) and 2D magnetotelluric (MT) modeling problems. We employ unstructured triangular grids that permit efficient discretization of complex modeling domains such as those containing topography, dipping layers and multiple scale structures. Iterative mesh refinement is guided by a goal-oriented error estimator that considers the relative error in the strike aligned fields and their spatial gradients, resulting in a more efficient mesh refinement than possible with a previous approach based on the absolute errors. Reliable error estimation is accomplished by a dual weighted residual method that is carried out via hierarchical basis computations. Our algorithm is parallelized over frequencies, wavenumbers, transmitters and receivers, where adaptive refinement is performed in parallel on subsets of these parameters. Mesh sharing allows an adapted mesh generated for a particular frequency and wavenumber to be shared with nearby frequencies and wavenumbers, thereby efficiently reducing the parallel load of the adaptive refinement calculations. We demonstrate the performance our algorithm on a large cluster computer through scaling tests for a complex model that includes strong seafloor topography variations and multiple thin stacked hydrocarbon reservoirs. In tests using up to eight-hundred

processors and a realistic suite of CSEM data parameters, our algorithm obtained run-times as short as a few seconds to tens of seconds.

Key words: electromagnetic theory, adaptive finite element modeling, unstructured grids, parallel computing, magnetotellurics, marine electromagnetics

1 INTRODUCTION

Numerical techniques employed to model electromagnetic (EM) geophysical problems commonly utilize a domain discretization based on a structured rectangular grid (e.g., Wannamaker et al. 1987; Mackie et al. 1993; Newman & Alumbaugh 1995). Such grids are appealing since they can be crafted without the need of specialized software and permit a convenient approximation of the spatial derivatives required by the traditional finite difference method. However, EM surveys are often acquired over complex terrains with large topographic or bathymetric variations that can impart a strong influence on the measured data (e.g., Wannamaker et al. 1986; Schwalenberg & Edwards 2004; Li & Constable 2007; Constable et al. 2009; Key & Constable 2011), yet including topography in rectangular modeling grids can be challenging. For example, the fine-scale discretization required to handle topography in the central region of a model leads to thin rows and columns of cells that extend laterally and vertically throughout the entire domain, often resulting in highly elongated cells (e.g., Constable et al. 2009, Figure 9). Such high aspect ratio cells can impart a large condition number to the resulting numerical linear system and often result in poor convergence for iterative Krylov solvers. Similar grid design problems arise for the discretization of known subsurface structures of highly variable shape and those that are inclined with respect to the structured grid.

The finely spaced rows and columns of cells required to represent these features will propagate throughout the entire model domain, resulting in a dense grid that may require more memory than is available and longer than desired run-times. Frequently, one must compromise between fine meshing to accurately represent known structures and coarse meshing to allow the problem to fit into memory or to be solvable in a reasonable time-frame. Certainly future improvements in computational power will allow these problems to be overcome by brute-force fine meshing of the entire model domain; however, for present day computers such profligate meshing schemes can only be implemented for relatively limited model domains.

A well-known solution to these discretization difficulties is to instead use an unstructured grid of triangular elements that can readily conform to highly complex model features. While unstructured grids have long been used in the applied math and engineering communities, they have only recently been applied to EM geophysical problems, as exemplified by finite element codes developed to handle

the topographic variations encountered in marine EM surveys (Key & Weiss 2006; Li & Key 2007; Franke et al. 2007; Li & Pek 2008). In part, these recent developments have been enabled by the widespread availability of robust computational geometry tools for creating constrained conforming Delaunay triangulations (e.g., Shewchuk 2002b), thereby lowering a significant software hurdle for the EM geophysicist interested in utilizing unstructured grids. In addition to the ease of handling topography, another key benefit of unstructured grids is the ability to efficiently discretize multiple-scale structures. For example, tiny grid elements required for near-surface features do not propagate into deeper regions when meshed with an unstructured grid. This model representation efficiency results in faster run times for a given model structure, as well as the ability to handle more complex structures for a given memory capacity.

Another concern for model discretization is the desired solution accuracy. The designer of structured grids must consider how fine the grid spacing should be in order to ensure a sufficiently accurate numerical computation. Typically, rules-of-thumb for grid density and the user's prior experience are used to design a static grid. However, such methods often offer little confidence that the grid is indeed accurate for a given set of model and data parameters. A solution to these cumbersome problems is to instead utilize a dynamic approach to grid generation, in which solution accuracy is enforced through the use of automated adaptive refinement methods. Adaptive methods seek to increase the solution accuracy by iteratively refining the grid, where each iteration consists of selecting a subset of elements for refinement based on an estimate of their contribution to the solution error, and then refining the grid by creating new smaller elements in these regions. Such methods offer great potential since they can be used to automatically compute an efficient and accurate model discretization, without requiring user intervention or even expert knowledge about a particular problem.

When the solution behavior is at least approximately known beforehand, grid refinement can be performed using a local a priori error estimator related to the element shape and size. However, given the complicated nature of electromagnetic coupling between the large conductivity variations encountered in geologic structures, a priori methods will only have limited effectiveness for EM geophysical applications. A posteriori methods based on either post-processing of the finite element solution or the computation of an auxiliary solution can overcome this difficulty (e.g., Ainsworth & Oden 2000). For example, gradient recovery methods involve post-processing of the finite element solution to obtain an improved gradient estimate, from which asymptotically exact error estimators can be obtained (e.g., Bank & Xu 2003). Other methods involve examining the residual of the finite element solution (e.g., Ainsworth & Oden 2000). In a relevant example, Franke et al. (2007) applied an error estimator based on the solution residual and the jump in tangential field components across element boundaries to create an adaptive method for magnetotelluric (MT) modeling.

The most commonly used error estimators seek to reduce the global error in the finite element solution. Specifically, the grid is refined anywhere the solution is estimated to be inaccurate. While such global error estimators can be effective for EM geophysics (e.g., Franke et al. 2007), a more efficient alternative approach can be found by noting that geophysical observations are typically made at only a relatively few discrete points, and therefore the solution only needs to be accurate at these locations. Such methods fall under the category of goal-oriented error estimation (cf. Becker & Rannacher 2001; Estep et al. 2005; Ovall 2006), where the goal is to obtain an accurate solution with respect to some functional measure—in our case the measure of interest restricts attention to some subdomain of the model, for example at the location of EM sensors. Goal-oriented error estimators are typically developed using a sensitivity functional that measures how the error in one portion of the model may corrupt the solution elsewhere, an effect referred to as pollution. The origins of estimating and controlling error in some quantity of interest in the context of finite element computations, as opposed to error in a global norm, can be traced at least back to a series of papers by Babuška and Miller (Babuška & Miller 1984a,b,c). Perhaps the earliest discussion of pollution effects for finite element problems may be found in (Nitsche & Schatz 1974), and another very good standard reference for understanding this phenomenon is given in Wahlbin (1991).

We have developed goal-oriented error estimators for 2D EM geophysics by applying the dual-error weighting approach (DEW, Ovall 2006) to the 2D magnetotelluric problem (Key & Weiss 2006) and the 2.5D controlled-source EM problem (Li & Key 2007). The DEW approach has proven useful for generating accurate EM responses for simple blocky models but suffers some setbacks as the complexity of the model increases. In particular, the DEW method uses gradient recovery computations that must be performed piecewise over regions of uniform conductivity, and these scale poorly as the complexity of the model increases. Furthermore, the recovered gradients can be highly inaccurate in uniform conductivity regions that are only a single element thick (as is often the case for thin geologic strata), and therefore can produce inaccurate error estimates in these regions.

In this work we investigate an alternative goal-oriented error estimator based on the dual weighted residual (DWR) method that was proposed (but not implemented) in Ovall (2006). A fundamental difference between the DWR and DEW methods is that for DWR the errors are determined through the weighting of residuals via hierarchical basis computations, rather than through the gradient recovery operations used by the DEW method. The hierarchical basis approach is advantageous since the underlying physics are applied throughout the entire model domain during the error estimate computations, rather than the piecewise application used for the gradient recovery operations. The dual approach is sourced using a new error functional that is related to the relative error in the strike aligned fields and

their spatial gradients, rather than the absolute error functional implemented in previous works (Key & Weiss 2006; Li & Key 2007).

We also consider the practical implementation of our algorithm for modern parallel computers, with the goal of being able to compute accurate 2D controlled-source electromagnetic (CSEM) responses in less than one minute for complicated models and a full suite of data parameters typical of offshore CSEM surveys. In addition to the straightforward embarrassingly parallel aspects of multi-frequency, multi-transmitter EM computations, we present a framework for optimized parallel computations by performing adaptive refinement on subsets of the full suite of data parameters. The adaptively refined grids generated for frequency and wavenumber subsets are then shared with nearby frequencies and wavenumbers, thus balancing computational speed with numerical accuracy.

2 PROBLEM STATEMENT

We begin by presenting the coupled differential equations that govern 2D electromagnetic geophysics. These well-studied equations (e.g., Stoyer & Greenfield 1976; Lee & Morrison 1985; Everett & Edwards 1992; Unsworth et al. 1993; Mitsuhata 2000) have recently been considered in the context of marine controlled-source EM exploration for oil and gas exploration on the continental shelves (Li & Key 2007; Abubakar et al. 2008; Kong et al. 2008). In our derivation below, we introduce a new compact form of these equations that allows for a concise statement of the physical problem and its finite element solution.

Consider the isotropic 2D electrical conductivity model $\sigma(y, z)$ with strike direction x and some imposed electric and magnetic sources $\mathbf{J^s}$ and $\mathbf{M^s}$. Assuming the time variation $e^{-i\omega t}$, the governing equations for the frequency domain electric field $\mathbf{E}$ and magnetic field $\mathbf{H}$ are

$$\nabla \times \mathbf{E} - i\omega\mu\mathbf{H} = \mathbf{M^s}, \tag{1}$$

$$\nabla \times \mathbf{H} - \sigma\mathbf{E} = \mathbf{J^s}, \tag{2}$$

where μ is the magnetic permeability and ω is the angular frequency. The quantity σ denotes the complex electric conductivity $\sigma = \hat{\sigma} - i\omega\epsilon$, where the dielectric permittivity ϵ is included for completeness. However, for low-frequency geophysical applications the imaginary term has a much smaller magnitude than the real conductivity $\hat{\sigma}$ and therefore is typically neglected (e.g., Ward & Hohmann 1988). Despite the 2D conductivity distribution, the sources $\mathbf{J^s}$ and $\mathbf{M^s}$ may contain 3D variations and therefore these equations describe 3D electromagnetic field variations. They can be transferred into 2D equations by Fourier transformation with respect to x, written as

$$\hat{\mathbf{F}}(k_x, y, z) = \int_{-\infty}^{\infty} \mathbf{F}(x, y, z) e^{-ik_x x} dx, \tag{3}$$

where k_x is the spatial wavenumber in the strike direction and a hat ($\hat{}$) denotes the quantity in the wavenumber domain (k_x, y, z). After Fourier transformation of (1) and (2), the coupled differential equations for the strike parallel components $\hat{E}_x$ and $\hat{H}_x$ are found to be

$$-\nabla \cdot (\sigma\lambda \nabla \hat{E}_x) - \nabla \cdot (ik_x\lambda R \nabla \hat{H}_x) + \sigma \hat{E}_x = f_1 \,, \tag{4}$$

$$-\nabla \cdot (ik_x\lambda R \nabla \hat{E}_x) - \nabla \cdot (i\omega\mu\lambda \nabla \hat{H}_x) + i\omega\mu \hat{H}_x = f_2 \,. \tag{5}$$

where

$$\begin{aligned}
\lambda^{-1} &= k_x^2 - i\omega\mu\sigma = k_x^2 - \omega^2\mu\epsilon - i\omega\mu\hat{\sigma}, && (6)\\
f_1 &= -\nabla \cdot (\sigma\lambda R \hat{\mathbf{M}}_t^s) + \nabla \cdot (ik_x\lambda \hat{\mathbf{J}}_t^s) - \hat{J}_x^s, && (7)\\
f_2 &= -\nabla \cdot (i\omega\mu\lambda R \hat{\mathbf{J}}_t^s) + \nabla \cdot (ik_x\lambda \hat{\mathbf{M}}_t^s) - \hat{M}_x^s, && (8)\\
R &= \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}. && (9)
\end{aligned}$$

$\hat{\mathbf{M}}_t^s$ and $\hat{\mathbf{J}}_t^s$ are the vector source currents transverse to $\hat{x}$ (i.e. $\hat{\mathbf{J}}_\mathbf{t}^\mathbf{s} = (\hat{J}_y^s, \hat{J}_z^s)$), and after the Fourier transformation the ∇ operator is only defined in the transverse plane (i.e. $\nabla = (\frac{\partial}{\partial y}, \frac{\partial}{\partial z})$). Noting that the rotation matrix R rotates vectors by $\pi/2$ counter-clockwise, we can express (4)-(5) in the compact form

$$-\nabla \cdot (A\nabla \mathbf{u}) + C\mathbf{u} = \mathbf{f} \text{ in } \Omega \qquad \mathbf{u} = \mathbf{0} \text{ on } \partial\Omega \tag{10}$$

$$A = \lambda \begin{pmatrix} \sigma I & ik_x R \\ ik_x R & i\omega\mu I \end{pmatrix} \qquad C = \begin{pmatrix} \sigma & 0 \\ 0 & i\omega\mu \end{pmatrix} \tag{11}$$

where $\mathbf{u} = (\hat{E}_x, \hat{H}_x)$, $\mathbf{f} = (f_1, f_2)$, and we interpret the gradient and divergence component-wise:

$$\begin{aligned}
\nabla \mathbf{u} &= (\nabla \hat{E}_x, \nabla \hat{H}_x), && (12)\\
\nabla \cdot A\nabla \mathbf{u} &= \nabla \cdot \mathbf{w} = (\nabla \cdot \mathbf{w}_1, \nabla \cdot \mathbf{w}_2) \,. && (13)
\end{aligned}$$

Here and elsewhere, $\Omega \subset \mathbb{R}^2$ is our bounded computational domain, and $I \in \mathbb{R}^{2\times 2}$ is the identity matrix. We point out that for (4) and (5) to make sense, it is necessary that $\lambda^{-1} \neq 0$, as λ appears in both the differential operators and the right-hand sides. We will henceforth assume that $\lambda^{-1} \neq 0$, which will always be true for the non-zero frequency and conductivity encountered for EM geophysical applications.

Once $\hat{E}_x$ and $\hat{H}_x$ have been obtained via the solution of (10), the transverse field components can be computed using

$$\begin{aligned}
\hat{\mathbf{E}}_t &= -ik_x\lambda \left(\nabla \hat{E}_x - R\hat{\mathbf{M}}_t^s\right) + i\omega\mu\lambda \left(R\nabla \hat{H}_x + \hat{\mathbf{J}}_t^s\right), && (14)\\
\hat{\mathbf{H}}_t &= -ik_x\lambda \left(\nabla \hat{H}_x - R\hat{\mathbf{J}}_t^s\right) + \sigma\lambda \left(R\nabla \hat{E}_x + \hat{\mathbf{M}}_t^s\right). && (15)
\end{aligned}$$

Therefore, the numerical 2D EM problem requires the accurate computation of both the strike aligned fields in (4) and (5) and their spatial gradients in (14) and (15). Finally, the 2D wavenumber domain fields can be converted into 3D spatial domain fields using the inverse Fourier transform

$$\mathbf{F}(x,y,z) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{\mathbf{F}}(k_x,y,z)e^{ik_x x}dk_x. \tag{16}$$

2.1 Finite Element Solution

For the finite element method, the coupled system of partial differential equations (10) is converted to the weak form and the 2D EM problem is restated as a coupled variational problem: Find $\mathbf{u} \in \mathcal{H} = [H_0^1(\Omega)]^2$ such that

$$\underbrace{\int_\Omega (A\nabla\mathbf{u}) : \nabla\overline{\mathbf{v}} + (C\mathbf{u})\cdot\overline{\mathbf{v}}\,dV}_{B(\mathbf{u},\mathbf{v})} = \underbrace{\int_\Omega \mathbf{f}\cdot\overline{\mathbf{v}}\,dV}_{F(\mathbf{v})} \text{ for all } \mathbf{v}\in\mathcal{H}\,. \tag{17}$$

Here, $H^1(\Omega)$ denotes the Sobolev space of (complex-valued) functions which are square-integrable and whose first-order weak derivatives are also square-integrable; $H_0^1(\Omega) \subset H^1(\Omega)$ consists of those functions which vanish on the boundary in the sense of trace (cf. Evans 2010; Adams & Fournier 2003). The notation $\overline{v}$ and $\overline{\mathbf{v}}$ used here and elsewhere denotes the complex conjugate of a scalar or the componentwise complex conjugate of a vector. The space $\mathcal{H}$ is equipped with the norm $\|\cdot\|_1$ and semi-norm $|\cdot|_1$ defined by

$$\|\mathbf{v}\|_1^2 = \int_\Omega |\nabla\mathbf{v}|^2 + |\mathbf{v}|^2\,dV \quad , \quad |\mathbf{v}|_1^2 = \int_\Omega |\nabla\mathbf{v}|^2\,dV\,.$$

Because of a Poincaré-Friedrichs inequality, the semi-norm $|\cdot|_1$ is actually a norm on $\mathcal{H}$, which is equivalent to the standard norm $\|\cdot\|_1$, i.e. there is a constant $C>0$ depending only on the domain Ω such that

$$|\mathbf{v}|_1 \le \|\mathbf{v}\|_1 \le C|\mathbf{v}|_1 \text{ for any } \mathbf{v}\in\mathcal{H}\,.$$

Setting $\mathbf{w} = (\mathbf{w}_1,\mathbf{w}_2) = A\nabla\mathbf{u}$, the double-dot notation should be understood as

$$(A\nabla\mathbf{u}) : \nabla\overline{\mathbf{v}} = \mathbf{w} : \nabla\overline{\mathbf{v}} = \mathbf{w}_1\cdot\nabla\bar{v}_1 + \mathbf{w}_2\cdot\nabla\bar{v}_2\,.$$

For our finite element discretization, we consider a family of triangulations $\mathcal{F} = \{\mathcal{T}_n\}_{n\in\mathbb{N}}$, also called meshes or grids, which is *conforming*—the intersection of any two triangles $\tau,\tau'\in\mathcal{T}_n$ (for any n) is either a common vertex, a common (complete) edge, or is empty; and *shape-regular*—all angles of all triangles in all triangulations are bounded away from 0 and π, with bounds independent of n. Such meshes can be highly adapted, and we make no assumptions about relationships between meshes, though one of our aims is that $\mathcal{T}_n$ will be better-suited for approximating $\mathbf{u}$ as n increases.

Given $\mathcal{T}_n$ we define the corresponding finite element space:

$$V_n = \{\mathbf{v} \in \mathcal{H} \cap [C(\Omega)]^2 : \ \mathbf{v}|_\tau \in [\mathbb{P}_1(\tau)]^2 \text{ for each } \tau \in \mathcal{T}_n\} \ . \tag{18}$$

Here and elsewhere, $\mathbb{P}_k(S)$ is the space of polynomials of total degree k or less on S, and $C(\Omega)$ denotes the set of globally continuous functions on Ω, meaning $\mathbf{v} \in V_n$ is continuous and both of its components are affine when restricted to a triangle. In other words, V_n refers to the space of piecewise-linear functions used for the finite element solution. A discrete version of (17) is: Find $\mathbf{u}_n \in V_n$ such that

$$B(\mathbf{u}_n, \mathbf{v}) = F(\mathbf{v}) \text{ for all } \mathbf{v} \in V_n \ . \tag{19}$$

The solution of (19) is equivalent to the solution of a $2N \times 2N$ sparse linear system, where N is the number of interior vertices in $\mathcal{T}_n$.

The technical details related to forming the linear systems associated with (19) and (21) (below) are postponed until Appendix A; a discussion of the well-posedness of (17), (19) and (21) is given in Appendix B.

3 HIERARCHICAL BASIS ERROR ESTIMATION

3.1 The Global Error Estimator

We wish to approximate the finite element error $\mathbf{u} - \mathbf{u}_n$ as a function in an auxiliary space, $\mathbf{u} - \mathbf{u}_n \approx \boldsymbol{\varepsilon}_n \in W_n$. Defining $\mathcal{V}_n$ to be the set of interior vertices of $\mathcal{T}_n$, we choose the auxiliary space to be

$$\begin{aligned} W_n = \{\mathbf{v} \in \mathcal{H} \cap [C(\Omega)]^2 : \ \mathbf{v}|_\tau \in [\mathbb{P}_2(\tau)]^2 \\ \text{for each } \tau \in \mathcal{T}_n \ , \ \mathbf{v}(z) = \mathbf{0} \text{ for each } z \in \mathcal{V}_n\} \ . \end{aligned} \tag{20}$$

Such spaces are said to consist of quadratic *edge-bump* functions, so-called because the standard basis for this space is naturally associated with the edges in the mesh, and such functions vanish at every vertex. The approximate error function is computed by "projecting" the error onto W_n:

$$B(\boldsymbol{\varepsilon}_n, \mathbf{v}) = F(\mathbf{v}) - B(\mathbf{u}_n, \mathbf{v}) \ (= B(\mathbf{u} - \mathbf{u}_n, \mathbf{v})) \quad \text{ for all } \mathbf{v} \in W_n \ . \tag{21}$$

We note that the space of functions $X_n = V_n \oplus W_n$, whose functions $\mathbf{x}$ which can be expressed uniquely as $\mathbf{x} = \mathbf{v} + \mathbf{w}$ where $\mathbf{v} \in V_n$ and $\mathbf{w} \in W_n$, are precisely those which are globally continuous and quadratic on each triangle. The natural bases for the spaces V_n and W_n are described in Appendix A. Together they form a so-called *hierarchical basis* for the space X_n—in contrast to the standard basis for X_n, in which every basis function has degree 2. The solution of (21) is equivalent to the solution of a $2M \times 2M$ linear system, where M is the number of interior edges in $\mathcal{T}_n$.

Hierarchical basis error estimation, as outlined in the previous paragraph, is well-known, and its properties are fairly well-understood, at least in the case of scalar equations (cf. Ovall 2007; Bank 1996; Ainsworth & Oden 2000). In the scalar case, a wealth of empirical data strongly suggests, and it can sometimes be proven (cf. Bank 1996) that there are reasonable constants $c_1, c_2 > 0$, independent of n, for which $c_1\|\boldsymbol{\varepsilon}_n\|_1 \leq \|\mathbf{u} - \mathbf{u}_n\|_1 \leq c_2\|\boldsymbol{\varepsilon}_n\|_1$ for n sufficiently large; also Ovall (2007) gives a partial explanation of why one often sees $\lim_{n\to\infty} \|\boldsymbol{\varepsilon}_n\|_1/\|\mathbf{u} - \mathbf{u}_n\|_1 = 1$ in practice, with ratios very near one even for modest mesh sizes. So we see that the global norm $\|\boldsymbol{\varepsilon}_n\|_1$ can be reliably used to determine if $\|\mathbf{u} - \mathbf{u}_n\|_1$ is below a given tolerance. Local norms on triangles τ, namely $\|\boldsymbol{\varepsilon}_n\|_{1,\tau}$, are used to decide which triangles should be refined to further decrease the error.

3.2 The Goal-Oriented Error Estimator

The global error estimator represented by (21) is useful when considering the accuracy of the finite element method over the entire model domain. However, for most practical geophysical applications, the finite element solution only needs to be locally accurate. For example, in EM geophysics an accurate solution is only required at discrete points in the model corresponding to the locations of the electric and magnetic sensors deployed for field surveys. In this case, the goal of obtaining accurate responses at the sensor locations can usually be accomplished with a much sparser mesh than required when a globally accurate solution is desired.

Consider a dual functional $G(\mathbf{u})$ that is some measure of the solution at the discrete locations of interest in the model domain. For example, $G(\mathbf{u})$ could be a measure of the absolute or relative size of $\mathbf{u}$ at the EM sensor locations, or could be a quantity derived from $\mathbf{u}$ such as the magnetotelluric impedance. The purpose of a goal-oriented error estimator is to reduce the size of the functional error $|G(\mathbf{u} - \mathbf{u}_n)|$. In other words, the goal is to reduce the error in the quantity of interest $G(\mathbf{u})$, rather than the global error in $\mathbf{u}$. The dual/adjoint problem associated with the functional G is: Find $\boldsymbol{w} \in \mathcal{H}$ such that

$$B(\mathbf{v}, \mathbf{w}) = G(\mathbf{v}) \text{ for all } \mathbf{v} \in \mathcal{H} \,. \tag{22}$$

We emphasize the reversal of the roles in the arguments between this problem and the primal problem (17), and note that, on the discrete level, the matrices associated with the computation of (approximate) solutions are conjugate-transposes (adjoints) of each other. The corresponding functionals of the primal and dual solutions $\mathbf{u}$ and $\mathbf{w}$ are related by

$$G(\mathbf{u}) = B(\mathbf{u}, \mathbf{w}) = F(\mathbf{w}). \tag{23}$$

The key relationship for the dual functional error $G(\mathbf{u} - \mathbf{u}_n)$ is then

$$G(\mathbf{u} - \mathbf{u}_n) = B(\mathbf{u} - \mathbf{u}_n, \mathbf{w}) = F(\mathbf{w}) - B(\mathbf{u}_n, \mathbf{w}). \tag{24}$$

The middle term shows that the dual solution $\mathbf{w}$ can be thought of as a sensitivity term for the error $\mathbf{u} - \mathbf{u}_n$, which serves to weight the error according to its influence on the dual functional $G(\mathbf{u})$. This idea has been exploited previously in the dual error-weighting approach (Ovall 2004, 2006) and has proven useful for EM geophysical applications (Key & Weiss 2006; Li & Key 2007; Li & Pek 2008). However, dual error-weighting requires a gradient recovery operator to approximate the gradient terms in $B(\mathbf{u} - \mathbf{u}_n, \mathbf{w})$, which for EM geophysics applications often suffers from a few shortcomings. In order to preserve the underlying physical boundary conditions, the gradient recovery operations must be performed piecewise over regions of uniform conductivity, and thus scale poorly as the complexity of the model increases. More importantly, the recovered gradients can be highly inaccurate in uniform conductivity regions that are only a single element thick (as is often the case when meshing thin geologic strata), and therefore often produce inaccurate error estimates in these regions. Here we consider an alternative approach based on the suggestion that the term on the right side of equation 24 could prove effective for a more robust goal-oriented error estimator (Ovall 2006). The approach described below can be considered an example of the broader class of dual-weighted-residual (DWR) methods (cf. Becker & Rannacher 2001), in which the (approximate) solution of a related dual/adjoint problem is used to "weight" either the strong or weak form of the residual of the primal/original problem. Those referring to (Ovall 2006) should note that, in that work, we call such methods dual-residual-weighting (DRW) methods—this just reflects a different naming convention.

We begin by solving the discrete dual problem for $\mathbf{w}_n \in V_n$

$$B(\mathbf{v}, \mathbf{w}_n) = G(\mathbf{v}) \qquad \text{for all} \quad \mathbf{v} \in V_n. \tag{25}$$

The discrete error in the dual solution, $\mathbf{w} - \mathbf{w}_n \approx \boldsymbol{\delta}_n \in W_n$, is then found in a similar manner to the discrete error in the primal problem (eq. 21)

$$B(\mathbf{v}, \boldsymbol{\delta}_n) = G(\mathbf{v}) - B(\mathbf{v}, \mathbf{w}_n) \text{ for all} \quad \mathbf{v} \in W_n. \tag{26}$$

An approximation to the error in the dual functional is then found by inserting $\mathbf{w} \approx \mathbf{w}_n + \boldsymbol{\delta}_n$ into (24), giving

$$G(\mathbf{u} - \mathbf{u}_n) \approx \mu = F(\mathbf{w}_n + \boldsymbol{\delta}_n) - B(\mathbf{u}_n, \mathbf{w}_n + \boldsymbol{\delta}_n) = F(\boldsymbol{\delta}_n) - B(\mathbf{u}_n, \boldsymbol{\delta}_n), \tag{27}$$

where the final equality in (27) is due to the identity $F(\mathbf{w}_n) = B(\mathbf{u}_n, \mathbf{w}_n)$. We also define the local element error indicator for triangle τ:

$$\mu_\tau = |F(\boldsymbol{\delta}_n)_\tau - B(\mathbf{u}_n, \boldsymbol{\delta}_n)_\tau|. \tag{28}$$

A critical part of this method is the choice of an appropriate dual functional $G(\mathbf{v})$, which in the following context is referred to as the error functional since it is a measure of the error in a specific quantity of interest. Since 2D EM problems typically require high accuracy in both the strike aligned fields and their spatial gradients, we find the following functional to be effective:

$$G \equiv G_*(\mathbf{v}) = a_0 \int_{\tau_s} \overline{\boldsymbol{\varepsilon}}_n \cdot \mathbf{v} \;\; dV + a_1 \int_{\tau_s} \nabla \overline{\boldsymbol{\varepsilon}}_n : \nabla \mathbf{v} \;\; dV, \tag{29}$$

where

$$a_0 = \frac{1}{\int_{\tau_s} \left(|\mathbf{u}_n + \boldsymbol{\varepsilon}_n|^2 + e_0^2 \right) dV}, \tag{30}$$

$$a_1 = \frac{1}{\int_{\tau_s} \left(|\nabla(\mathbf{u}_n + \boldsymbol{\varepsilon}_n)|^2 + e_1^2 \right) \;\; dV}, \tag{31}$$

and τ_s is the possibly discontinuous subdomain of elements containing the EM receivers. The terms a_0 and a_1 serve to weight the integrals so that they approximate the relative errors in $\mathbf{u}$ and $\nabla \mathbf{u}$ rather than their respective absolute errors. This relative weighting is appropriate for EM geophysics applications since the sources of relative error are typically more significant than the absolute errors such as sensor noise-floors. For example, uncertainties in sensor orientations, source-receiver offsets, sensor responses, and other data parameters often result in an estimated data relative error of order 1% regardless of the absolute field strength. Furthermore, the fields produced by controlled-source EM transmitters can exhibit several orders of magnitude variation across an array of EM sensors. If our error functional considered only the absolute error, the adaptive refinement method could get bogged down performing excessive refinement in regions with large field amplitudes while neglecting smaller amplitude regions, at least until the absolute errors in the large amplitude regions have been driven down to similar levels. This could result in catastrophic over-refinement that could possibly consume all available computational memory. The relative error functional given above will be far more parsimonious when guiding adaptive mesh refinement since it will operate somewhat independently of the absolute field amplitude. However, since real EM data at far source-receiver offsets is ultimately dominated by the sensor noise-floor, we do include absolute error cut-off values in our error functional through the constants e_0 and e_1, which also ensure the denominators do not equal zero. For example, e_0 could be set to the sensor noise-floor so that the dual functional will only be large when the error is significantly above the noise-floor. Such a cut-off would result in a more sparsely refined mesh than if, for example, e_0 was set to a value a few orders of magnitude smaller.

In summary, the DWR error-estimator aims to find a suitable approximation of the error functional $G(\mathbf{u} - \mathbf{u}_n)$ as follows. The primal solution $\mathbf{u}_n$ and its estimated global error $\boldsymbol{\varepsilon}_n$ are computed using (19) and (21). These quantities are then inserted into the error functional in (29). Eqs (25) and (26) are

then solved for the dual solution $\mathbf{w}_n$ and its estimated error $\boldsymbol{\delta}_n$. Finally, the local error indicator for each triangular element is found with (28).

4 ADAPTIVE REFINEMENT METHOD FOR NUMERICAL ACCURACY

Our adaptive refinement procedure starts with a coarse Delaunay triangulation of the model domain. We apply the goal-oriented error estimation method described above to produce the local error indicator μ_τ for each element. We then identify the fraction α of elements with the largest errors, marking them for refinement. After a new refined mesh has been created, and eqs. (19), (21), (25) and (26) have been solved on this mesh, μ_τ is computed and again the α elements with largest error estimates are marked for refinement. This process is repeated until the functional error is less than some user specified tolerance; for example, less than 1% relative error is typically desired. A downside to this approach is that computing μ_τ for each iteration requires the relatively expensive solutions of the finite element systems in eqs. (19), (21), (25) and (26). A shortcut around this inefficiency utilizes the special structure of the error estimator, which allows it to be forward projected onto the refined mesh, thereby yielding a reasonable approximation of μ_τ on the refined grid without requiring a new solution of the finite element systems (e.g., Bank & Xu 2003). In practice, we use a low-order quadrature integration to compute μ_τ on the refined elements. This method can be re-applied for a few refinement iterations before the approximation breaks down and a new computation of the error estimator is required. We find that five iterations of refining the 6% worst elements by approximately subdividing them in half works well, with the mesh vertex count doubling in size, after which the finite element systems for the error estimator need to be solved for the new mesh.

5 PARALLEL IMPLEMENTATION DETAILS

We have implemented this goal-oriented adaptive refinement methodology into a parallel Fortran code named MARE2DEM (Modeling with Adaptively Refined Elements for 2D EM) that currently supports both magnetotelluric and controlled-source EM problems. Parallel constructs were implemented using the Message Passing Interface standard. For CSEM modeling, zero Dirichlet conditions are applied to the model boundaries and the dipole sources are approximated using point delta functions, rather than the discrete pseudo-delta-function approach used in fixed-grid modeling approaches (e.g., Mitsuhata 2000; Abubakar et al. 2008). We find that about 30 discrete k_x values spaced logarithmically from 10^{-6} to 10^{-1} m^{-1} are required to obtain accurate responses for typical marine CSEM model parameters and transmission frequencies. By noting the symmetry of the field components, the Fourier integrals used to transform the fields back to the spatial domain in (16) can be converted to

into either sine or cosine transforms, depending on the transmitter orientation and the particular field component. These integrals can be performed efficiently using the digital filter method (e.g., Anderson 1979), applied with spline interpolation of the fields components computed at the discrete k_x values. The fields from arbitrarily oriented dipole sources are obtained by the vector superposition of the fields from a source oriented in the $\hat{x}$ direction and another in the $(\hat{y}, \hat{z})$ plane, therefore requiring at most one additional source computation. We use 401 point digital filters designed in a manner similar to Guptasarma & Singh (1997). For the MT method, the plane-wave source field is implemented through one-dimensional boundary conditions applied to the left and right sides of the model domain with a cosine taper of these solutions applied along the top and bottom boundaries (Wannamaker et al. 1987).

5.1 Mesh generation and refinement

The input conductivity model consists of segment bound polygonal regions with an assigned constant conductivity for each region. The model outer boundaries are placed at least several skin depths away from the region of interest so that anomalous fields generated in the center of the model are sufficiently attenuated by the model boundaries, in accordance with the imposed Dirichlet conditions. Mesh generation and subsequent refinements are accomplished by direct calls from Fortran to routines in the C-code Triangle, a program for robust and rapid generation of constrained, conforming Delaunay triangulations (Shewchuk 1996). Triangle uses a minimum interior angle constraint to generate high quality triangulations that are suitable for numerical accuracy and stability considerations (e.g., Shewchuk 2002a). This angle constraint presents the practical issue that the input model structure should not have narrow *slivers*, where two segments intersect with an angle smaller than the constraint. A typical minimum angle constraint in the range of $20 - 30^\circ$ serves to balance mesh quality with flexibility in the input model structure. Mesh refinement is performed in Triangle through the use of element area constraints. In MARE2DEM, elements that have been marked for refinement are given area constraints that are half their current area.

The forward projection of the error estimate from a coarse mesh onto a refined mesh requires knowledge of which parent element in the coarse mesh contains the centroid of each child element in the refined grid. Since refinement in Triangle results in a sequence of meshes containing a hierarchy of vertices rather than a hierarchy of triangles, identifying the parent element is more costly than for algorithms that refine simply through triangle bisection. Because the error needs to be projected onto all elements of the refined grid, computing brute-force in-triangle point location queries for each element in the refined grid would scale very poorly as the mesh size increases. Instead, MARE2DEM performs efficient point location queries by using a variant of the jump-and-walk method that uses a 2-d tree structure (e.g., Kennel 2004) for rapid location of the nearest vertex (the jump) before traversing

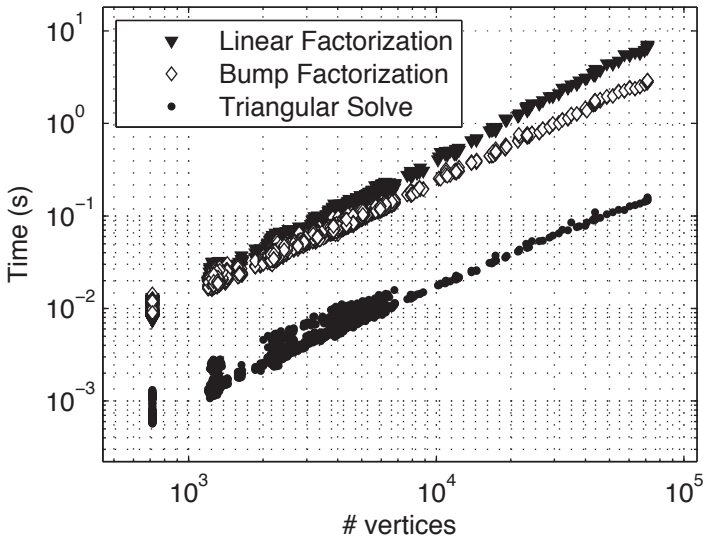

Figure 1. Scaling of the SuperLU solver for a typical adaptive refinement problem as a function of the number of mesh vertices. Triangles and diamonds show the time to compute the LU factorization for the sparse linear systems associated with the linear basis of the space V_n and the quadratic bump basis of the space W_n. Dots show the time to compute the triangular solves for a single right-hand-side vector. Computations were performed serially on a MacPro desktop computer containing two 2.93 GHz Quad-core Intel Xeon processors.

adjacent triangles to find which one contains the query point (the walk), resulting in a very favorable expected time dependence of $\Theta(n^{0.056})$ (Devroye et al. 2004).

5.2 Efficient solution of the linear systems

The four linear systems required for the error estimator may seem like an expensive burden that limits its practicality. However, the sparse matrices for these systems are identical between (19) and (25), and between (21) and (26); therefore, they can be solved efficiently using the LU decomposition (e.g., Golub & Van Loan 1996):

$$\mathbf{A}\mathbf{x} = \mathbf{L}\mathbf{U}\mathbf{x} = \mathbf{b} \tag{32}$$

where $\mathbf{L}$ and $\mathbf{U}$ are lower and upper triangular matrices. Once $\mathbf{L}$ and $\mathbf{U}$ have been computed, the solution for any right hand side vector $\mathbf{b}$ is found using forward and backward substitutions for the triangular systems

$$\mathbf{y} = \mathbf{L}^{-1}\mathbf{b}, \qquad \mathbf{x} = \mathbf{U}^{-1}\mathbf{y}. \tag{33}$$

MARE2DEM uses the super-nodal sparse matrix factorization routine SuperLU (Demmel et al. 1999). Fig. 1 shows the performance scaling of SuperLU for the LU factorization of the linear basis and bump basis systems, and for the triangular solves of (33). For a given mesh size, the factorizations take about

Table 1. Scaling coefficients for the forms shown in eq. (34) obtained by least squares fits to the SuperLU performance data shown in Fig. 1.

a:	0.000000650	b:	1.45
c:	0.00000226	d:	1.26
e:	0.000000638	f:	1.11

ten times longer than the triangular solution of a single right hand side vector. This means that additional source vectors can be added to the linear system with only a small increase in the computational time. Since the system matrices for the primal and dual problems are identical, MARE2DEM stores the LU factors computed for the primal problem and then reuses them for the dual problem (both for the linear and bump system matrices). The effective cost of the goal-oriented error estimator is therefore the additional time required to factorize the bump system matrix and the time for the triangular solutions of the three additional systems, totaling to roughly a factor of two increase over the time required for the primal problem alone. Additionally, as we show in the appendix, the coupling terms between $\hat{E}_x$ and $\hat{H}_x$ integrate to zero for the hierarchical basis system; hence MARE2DEM solves two smaller $M \times M$ systems rather than a single $2M \times 2M$ system when computing the hierarchical basis residuals. Since $k_x = 0$ for the magnetotelluric method, all the linear systems are decoupled and the primal and dual problems are solved rapidly as two uncoupled $N \times N$ systems rather than the $2N \times 2N$ coupled system required for the CSEM method.

Another efficiency of the LU decomposition is found for CSEM modeling. Multiple transmitter sources $\mathbf{J_s}$ and $\mathbf{M_s}$ in (4) and (5) result in multiple right hand side vectors in the linear systems that can be solved rapidly using the LU method described above. For mesh i with N_i vertices and n_s transmitters on the right hand sides of the linear systems, the time to solve the finite element systems and compute the error estimator is approximately

$$t_i = aN_i^b + 2cN_i^d + 4n_s eN_i^f, \tag{34}$$

where the first term on on the right represents LU factorization time for the $2N_i \times 2N_i$ sparse matrix used for both the primal and dual problems in (19) and (25), the second term is the LU factorization time for the two decoupled bump systems in (21) and (26), and the third term is the triangular solution time for the two linear and two bump systems for n_s sources. Table 1 lists values for the scaling coefficients obtained by least squares fits to the performance data shown in Fig. 1. Fig. 2 shows t_i as a function of the number of sources for meshes with 1000, 10,000 and 100,000 vertices. We can see that about 10-20 sources can be put on the right-hand side before the run-time doubles, and that further additional sources produce a run-time that is directly proportional to the number of sources. This

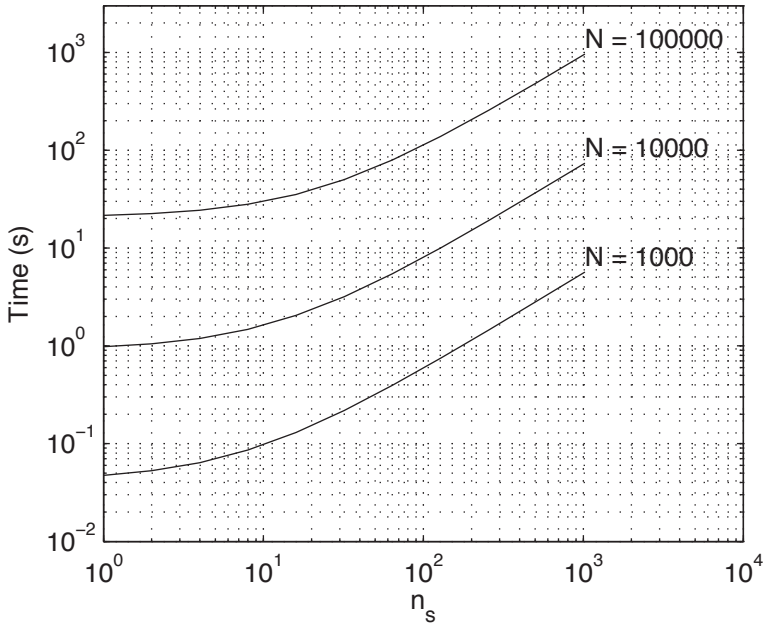

Figure 2. Time scaling of an adaptive refinement iteration as a function number of sources (n_s) in the right hand side of the linear system in eq. (34). Curves are shown for three different mesh sizes (N).

suggests that for problems with multiple transmitters, some efficiency can be gained lumping sources together into groups of about 10 for the adaptive refinement computations, rather than modeling each transmitter as an independent modeling task. A trade-off in this approach is that the resulting mesh will require more vertices (and hence memory and run-time) in order to accurately capture the electromagnetic fields produced by each transmitter. However, for the CSEM problems we have examined this is typically far offset by the time-efficiency gained from the multiple right hand side approach.

5.3 Parallelization

Many aspects of the 2.5D EM problem are embarrassingly parallel and readily allow for large savings in the computational run-time. Eqs. (4) and (5) show that independent computations are required for each wavenumber k_x and frequency ω, as well as for each source in $\mathbf{J_s}$ and $\mathbf{M_s}$. A typical 2D CSEM data set contains around 100 transmitter locations and 10 transmission frequencies, while about 30 wavenumbers are required for an accurate 2.5D computation. Thus, the fastest run-time would be obtained by running each of the 30,000 independent tasks on a cluster of the same size. Such massive clusters exist but represent the pinnacle of current technology and are therefore of limited availability; more typical clusters consist of a few hundred processors, while desktop machines routinely ship with 8 or more processing cores. Thus, for ordinary CSEM modeling problems there will be significantly

more parallel modeling tasks than available processors, although the number of tasks can be reduced by an order of magnitude through the transmitter grouping method mentioned in the previous section.

To demonstrate how MARE2DEM might scale on a large parallel system, we can create a model for the expected run-time, which is dominated by the solution of the linear systems. Assuming that the refinement iterations double the vertex count at each iteration (i.e. $N_i = 2^{i-1}N_1$), the total time to perform n refinement iterations is

$$T_n = \sum_{i=1}^{n} t_i = aN_1^b \frac{(2^{bn}-1)}{2^b-1} + 2cN_1^d \frac{(2^{dn}-1)}{2^d-1} + 4n_s eN_1^f \frac{(2^{fn}-1)}{2^f-1}. \tag{35}$$

If there are n_{tx} total number of transmitter sources and groups of n_s sources are used for LU factorization speedup, the total number of independent tasks is then $n_{tasks} = n_{tx}/n_s$. If we assume that each task requires roughly the same adaptive refinement time and n_{proc} is the number of parallel processors available, the expected total run-time required to complete all tasks at a given wavenumber and frequency is then

$$T_{total} = \frac{n_{tasks}}{n_{proc}} T_n. \tag{36}$$

As an example of the implications of this equation, Fig. 3 shows T_{total} as a function of the number of processors used, where the initial mesh contained $N_1 = 5000$ vertices and $n = 4$ and $n_{tx} = 1024$. While the fastest run-time is achieved by using a separate processor for each task (ie., $n_s = 1$), a reasonable run-time can be achieved with a factor of 16 fewer processors by grouping the sources together into sets of $n_s = 16$. On a single processor, grouping all sources together ($n_s = 1024$), while not very efficient, is still over 10 times faster than solving for each source separately ($n_s = 1$).

For the final component of the parallel implementation, we introduce the concepts of mesh sharing and receiver grouping. Traditional methods for EM modeling utilize a single fixed mesh that is designed to be suitable for the entire range of parameters (i.e. all transmitters, frequencies, receivers, wavenumbers), and therefore such a mesh is likely to be over-meshed for at least some of the data parameters, leading to an inefficiency in the run-time. Conversely, the adaptive finite element technique allows MARE2DEM to generate a unique mesh that is optimized for a given set of parameters, resulting in a very efficient use of mesh vertices. Since the EM solutions typically vary smoothly as a function of frequency and wavenumber, a mesh refined for a specific pair of these values is usually accurate for nearby values, as shown previously by examples in Key & Weiss (2006) and Li & Key (2007). Therefore, another efficiency is to share an adaptively refined mesh with neighboring frequencies and wavenumbers so that the refinement iterations only need to be computed once per mesh sharing group. Fig. 4 provides a visual interpretation of the mesh sharing concept, illustrating how various data parameters can be grouped together while the adaptive refinement iterations are performed only for the center parameter value. Since the refinement iterations require time T_n and the computations for

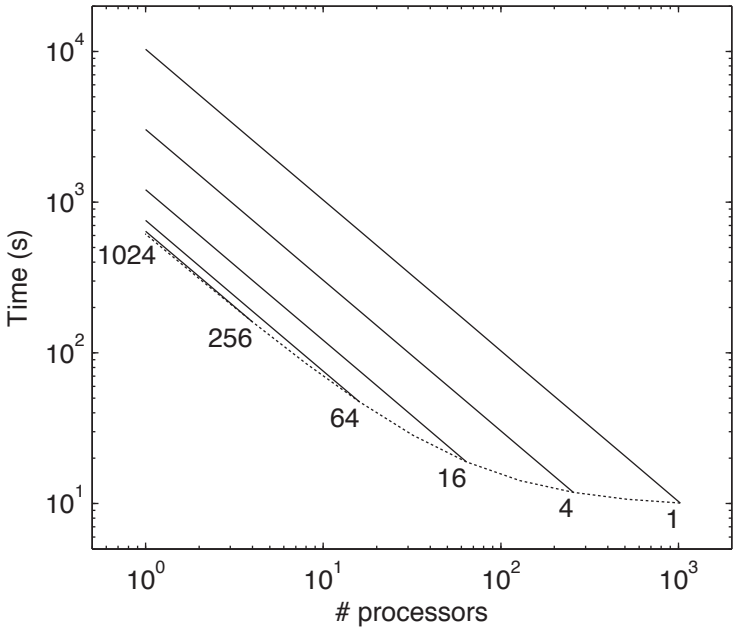

Figure 3. Expected performance T_{total} as a function of the number of processors utilized. The numbers next to each line indicate the groups of n_s sources. The lower limit of each line occurs when the number of processors available is equal to the number of tasks.

the nearby frequency-wavenumber pairs require time t_n, mesh sharing in the frequency-wavenumber space typical results in about 2-4 times faster run-times than for a full refinement sequence of every frequency-wavenumber pair.

Finally, the receivers can also be decomposed into small groups so that a given adaptive refinement task considers only a few receivers and therefore will run faster since a few receivers can be adaptively meshed with a more limited amount of mesh refinement than when all receivers are considered. We note that there can sometimes be a negative trade-off in performance for receiver grouping since each receiver group increases the number of parallel tasks to be performed, while the change in run-time for having fewer receivers in a given group may only be marginal. The best increase in performance will be found for very simple starting models with low mesh vertex counts to begin with, and always when the number of available processors is equal to or greater than the number of parallel tasks.

6 VALIDATION

Here we present examples that demonstrate the reliability and effectiveness of MARE2DEM for CSEM and MT modeling. The first example considers marine CSEM responses for the canonical 1D reservoir model, consisting of a 100 m thick, 100 ohm-m reservoir buried 1 km below the seafloor in 1 ohm-m sediments. The usefulness of the canonical 1D model resides in its well-studied solu-

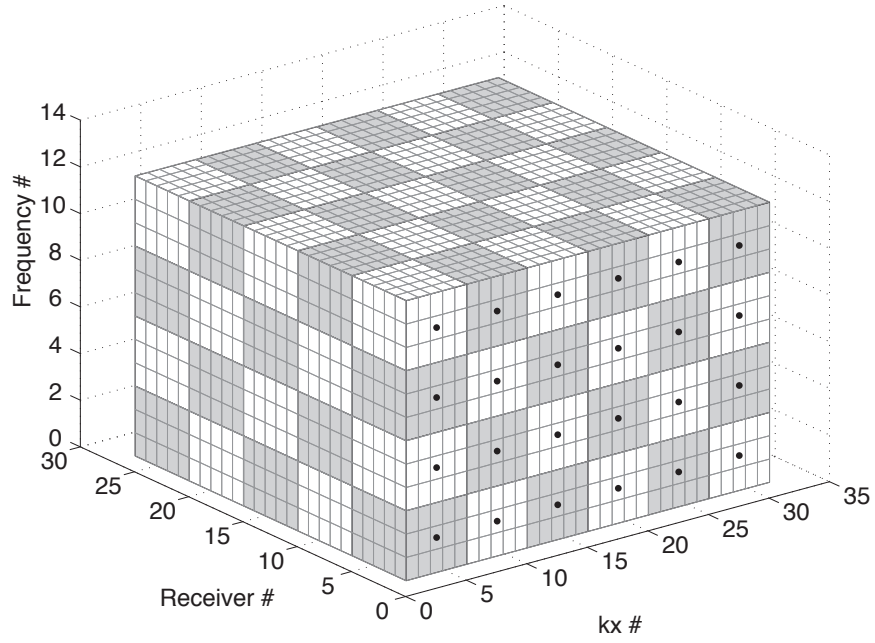

Figure 4. Example parallel decomposition of 2D CSEM modeling parameters into groups of independent tasks. Gray and white cubes denote parameters grouped together for mesh sharing. For a given group, the black dot denotes the frequency and wavenumber used for the adaptive refinement iterations that generate a mesh accurate for the remaining frequencies and wavenumbers of that group. In this example, a single mesh sharing group consists of five adjacent wavenumbers, three adjacent frequencies and five adjacent receivers. A fourth dimension can be added for group of adjacent EM transmitters. For MT modeling the decomposition will only depend on the frequencies and receivers since $k_x = 0$ for MT.

tions being readily available for validating the accuracy of numerical EM modeling techniques (e.g., Constable & Weiss 2006; Key 2009).

In our tests, the source consists of a 0.25 Hz horizontal electric dipole transmitter located 50 m above the seafloor while receivers are positioned along the seafloor every 500 m from 0 to 15 km range. Computations were performed on a MacPro Desktop containing two 2.93 GHz quad-core Intel CPUs, for a full capability of eight parallel processors. The MARE2DEM computations used thirty wavenumbers spaced logarithmically from 10^{-6} to 10^{-1} m^{-1}, with six continuous subsets of five wavenumbers used for mesh sharing. The mesh sharing subsets reached the 1% requested error tolerance within five to six adaptive refinement iterations. The starting mesh contained 4584 vertices, while the final meshes contained around 18,000-19,000 vertices, depending on the specific wavenumber.

The accuracy of the MARE2DEM solutions is measured by comparisons with the semi-analytical 1D responses from the open-source code Dipole1D (Key 2009). Fig. 5 shows that all field compo-

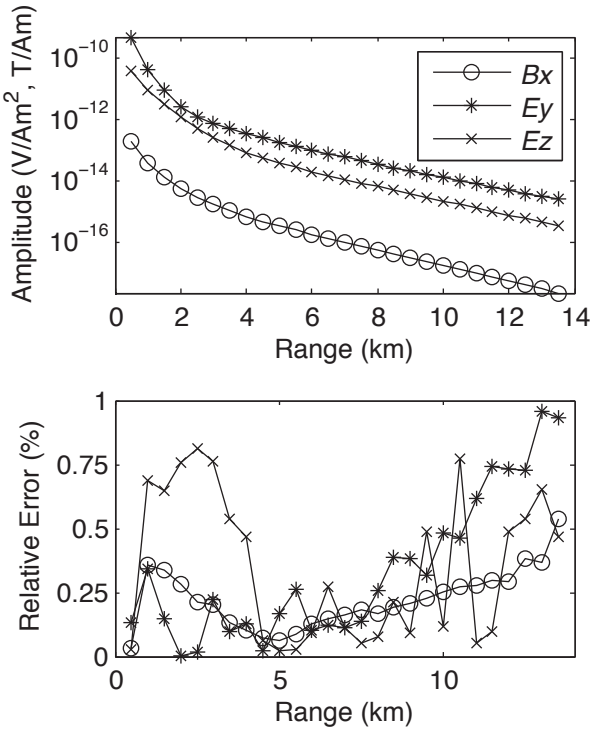

Figure 5. Validation of the magnetic and electric field responses computed by MARE2DEM for the canonical 1D reservoir model. Electric and magnetic field responses are shown in the top panel and the bottom panel shows their relative errors compared to solutions obtained with the Dipole1D modeling code.

nents for inline receivers are below the requested 1% error tolerance, demonstrating the validity of the MARE2DEM solutions for this model and data parameters. Fig. 6 extends this test to illustrate the code's effectiveness for a broad range of requested error tolerances. The true errors for all field components are slightly below the requested error tolerances, indicating that the goal-oriented error estimator is reliable for achieving a given tolerance. Furthermore, the increase in run-time that accompanies the significant reduction in the solution error is very modest—less than a factor of 10 despite the 1000 fold decrease in error. This demonstrates the efficiency in mesh refinement possible with goal-oriented error estimation.

Fig. 7 shows the accuracy of MARE2DEM for an example land MT problem consisting of the segmented slab model, for which semi-analytic solutions are available (Weaver et al. 1985, 1986). The segmented slab is composed of a 100 km thick slab that is divided into a 5 km wide, 1 ohm-m region in the middle, and is bound on the left and right by 10 and 50 ohm-m regions. The analytic solution has a perfect conductor underneath the slab, which we approximate in the finite element model by setting the basal resistivity to 10^{-8} ohm-m. MT responses at 100 s period were computed for 197 sites atop the slab at lateral positions between -100 to 100 km. For the MARE2DEM computations, the

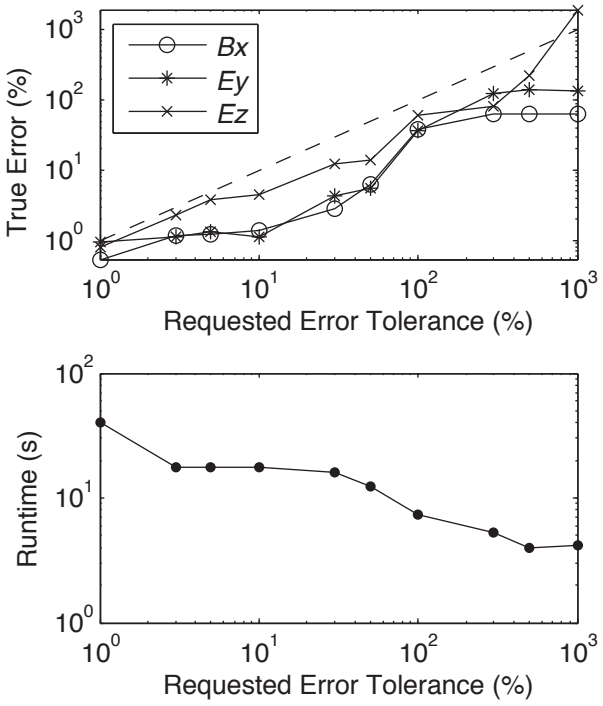

Figure 6. Effectiveness of the goal-oriented error estimator. The top plot shows the true error of the MARE2DEM solutions as a function of the requested error tolerance. The bottom plot shows the accompanying run-time required for the full adaptive refinement computations.

receivers were grouped into subsets of 20 receivers for each adaptive mesh refinement computation. On the MacPro computer described earlier, the entire process of adaptive refinement computations for all groups took about one second. Fig. 7 shows that the relative error in strike aligned electric field (E_x) is mostly below the 1% requested tolerance, although many of the receivers over the conductive middle segment have errors of up to 1.4%. Fig. 8 shows the coarse starting mesh consisting of 139 vertices and two examples of adaptively refined meshes for two different receiver groups that required around 3400 and 2600 vertices to meet the requested 1% error tolerance. The adaptively refined meshes are substantially denser around the particular receivers groups. They also show significant refinement in regions distal to the receivers, indicating where the mesh required refinement in order to mitigate pollution effects.

The fundamental datum of a magnetotelluric survey is the impedance tensor, which for 2D conductivity structures can be decomposed into the TE and TM mode impedances

$$Z_{\mathrm{TE}} = \frac{E_x}{H_y}, \qquad Z_{\mathrm{TM}} = \frac{E_y}{H_x}. \tag{37}$$

Fig. 9 shows the TE and TM mode impedance magnitudes and their relative error, which are generally

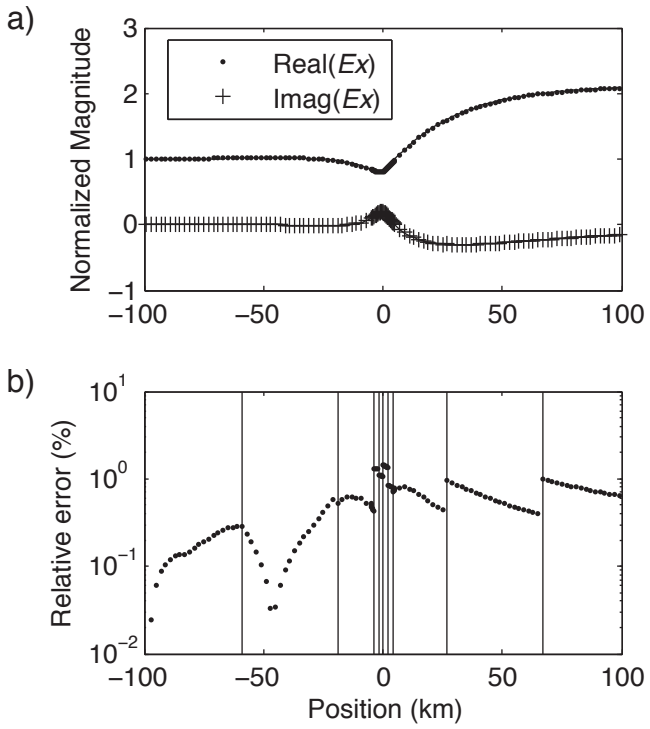

Figure 7. TE electric field response at 100 s period for the segmented slab model (top) and the relative error (bottom). Vertical lines delineate groups of receivers used for separate adaptive refinement computations.

less than 1%. The slight increase above 1% observed near 10 km position occurs where the responses have a strong lateral gradient. Part of the larger error here may be due to subtle pollution effects from the nearby conductive middle slab segment that is causing the lateral response gradient. This is a good example of the error estimator's reliability even when there are polluting structures nearby, although perfect reliability would have resulted in all responses with less than the target 1% error. An additional source of the increased error is due to the impedance being formed from the quotient of the electric and magnetic field components, which results in an upper bound for the impedance error that is larger than for the individual components, as can be easily shown through error propagation analysis.

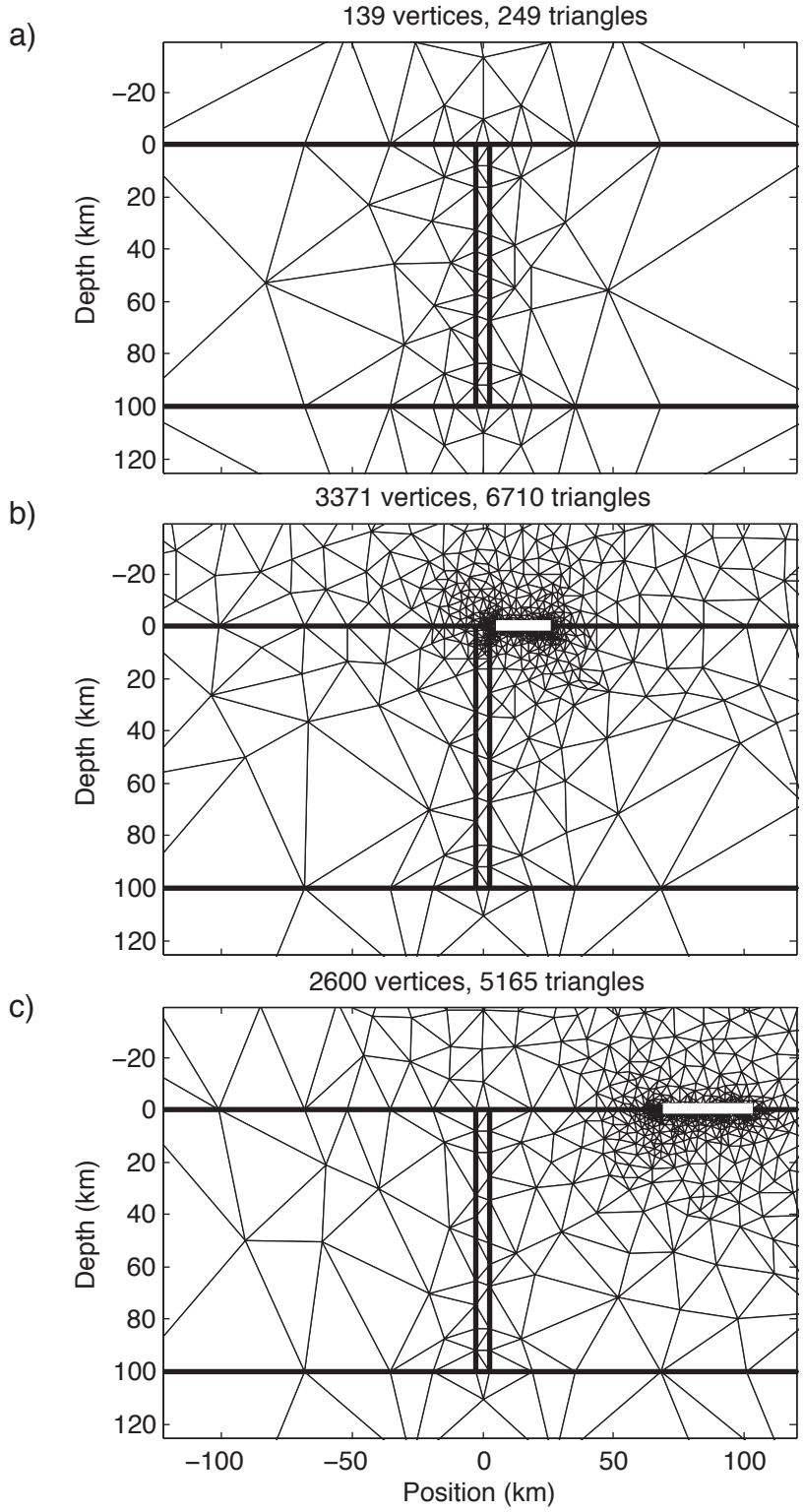

Figure 8. Adaptive mesh refinement for TE mode computations at 100 s period for the segmented slab model. Thick lines delineate the boundaries of the segmented slab while thin lines show the finite element mesh. The coarse starting mesh (top) was refined for various receiver groups, examples of which are shown in the middle and bottom plots. White bars indicate the locations of the receivers under consideration for each refined mesh.

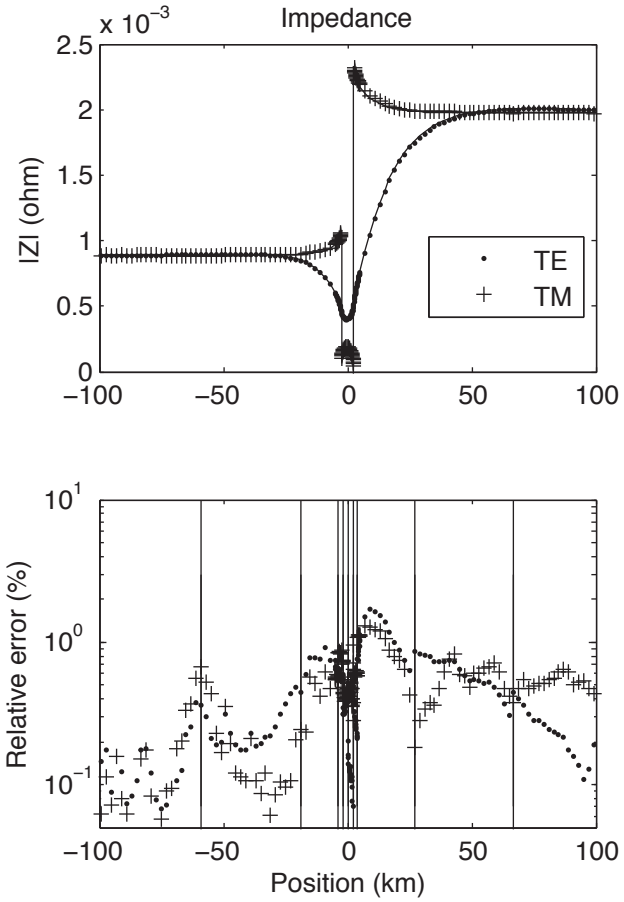

Figure 9. TE and TM mode impedance magnitudes for the segmented slab model computed using MARE2DEM (top) and their relative errors (bottom).Vertical lines delineate groups of receivers used for separate adaptive refinement computations.

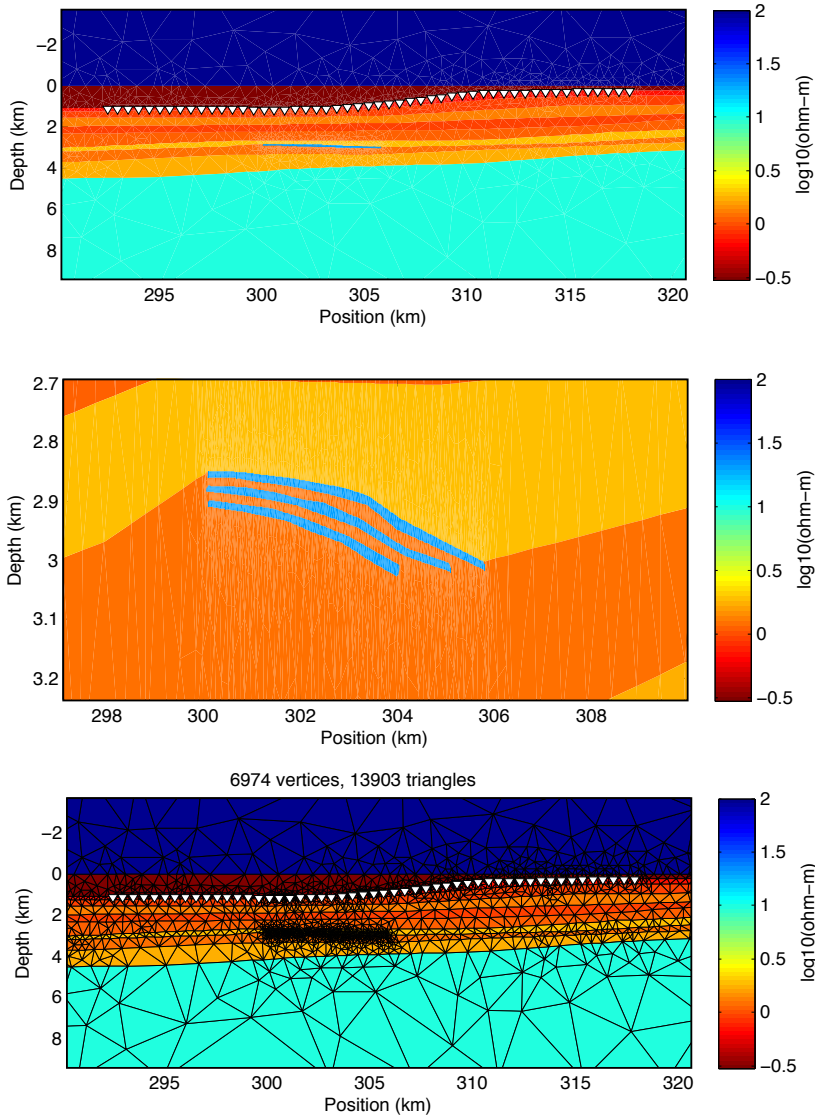

Figure 10. Complex marine conductivity model including typical features that are difficult to discretize on a rectangular grid: a large bathymetric slope, tilted regional strata and closely-spaced thin and dipping reservoir intervals. The three panels show the model (top), a close-up of the stacked reservoir layers (middle, note the vertical exaggeration) and the unstructured grid used as the starting mesh (bottom). Inverted white triangles show the location of the seafloor EM receivers. Only the central portion of the model is shown.

7 COMPLEX 2D EXAMPLE

The previous simple examples provided a validation of MARE2DEM and illustrated the reliability of the goal-oriented error estimator. Here we demonstrate how MARE2DEM performs for a significantly more complicated model and a much broader suite of data parameters. We designed a synthetic model that is representative of typical marine CSEM surveys for hydrocarbon exploration on the continental

Table 2. Scaling test parameters for the complex 2D model. Results from each run are shown in Fig. 11.

Run	# Frequencies	# Tx Grouped	# Total Tasks	# Refinement Tasks
a	12	1	18,360	1224
b	12	11	1800	120
c	12	51	360	24
d	1	1	1530	306
e	1	11	150	30
f	1	51	30	6

shelves (Fig. 10). The complex model contains many features that cannot be discretized efficiently on a structured rectangular grid: realistic seafloor topography, tilted geologic strata and closely-spaced thin resistive layers simulating a stacked reservoir sequence. The three dipping reservoir layers are only 10 m thick and have a resistivity of 20 ohm-m. All of these features are readily accommodated by the unstructured triangular grid using just under 7000 mesh vertices. The stratigraphic intervals might be known a priori from seismic exploration and exploratory wells. The purpose of conducting a CSEM survey would be to test whether a seismically imaged geologic trap structure is filled with resistive hydrocarbons or conductive seawater (e.g., Eidesmo et al. 2002), or to estimate the saturation of a reservoir as a function over time during production (e.g., Orange et al. 2009). The entire model domain is 200 km wide and 200 km high, vertically including 100 km of air and 100 km of seafloor; the reservoir is located at the middle of the domain horizontally.

To simulate a realistic marine CSEM profile, we modeled 51 receivers spaced every 500 m along the seafloor and 51 corresponding inline electric dipole transmitters deep-towed 50 m above the receivers. We modeled the CSEM responses at twelve frequencies spanning from 0.1 to 4.7 Hz. The particular frequencies are associated with the high amplitude harmonics of the doubly-symmetric waveform presented in Myer et al. (2011). The requested error tolerance was set to 1%. Computations were performed on the Triton Compute Cluster (TCC) maintained by the San Diego Supercomputer Center at the University of California, San Diego. The TCC system is composed of 256 nodes containing two quad-core Intel Nehalem 2.4 GHz CPUs, representing a parallel capacity of 2048 processors.

We performed a series of tests to examine the scaling of the MARE2DEM run-time as a function of the number of parallel processors. If each unique transmitter, receiver, wavenumber and frequency combination were considered separately for adaptive mesh refinement, the total number of tasks would be just under one-million. While this would yield the fastest run-time on a parallel system of the same size, this is far in excess of the number of available processors on the TCC system. Instead we use mesh sharing and parameter grouping to reduce the number of tasks significantly. We grouped all

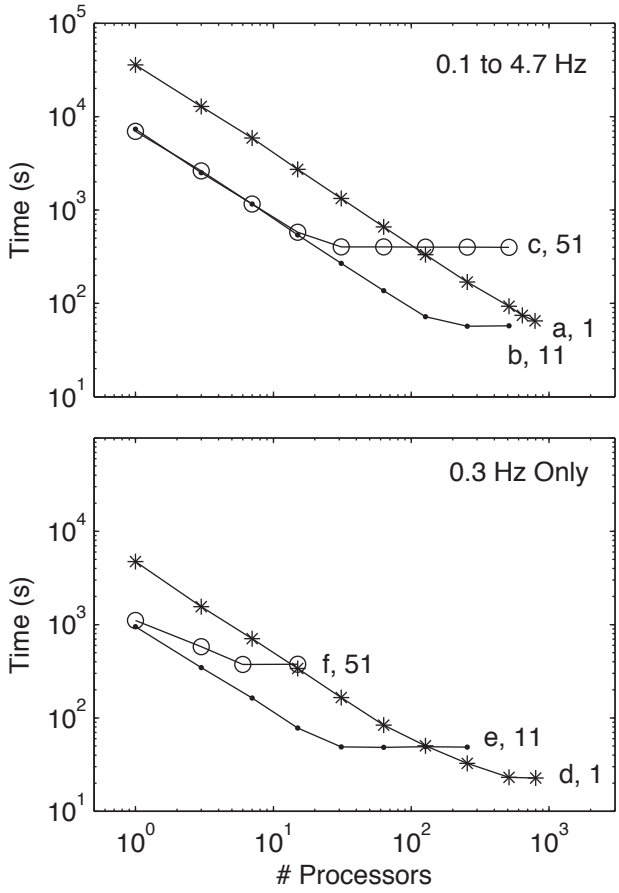

Figure 11. Run-time of MARE2DEM for the complex model as a function of the number of parallel processors used. The top panel shows the results for all twelve frequencies and the bottom panel shows results for only one frequency (0.3 Hz). The letters and numbers next to the curves correspond to specific transmitter grouping parameterizations specified in Table 2.

receivers together and then applied mesh sharing to groups of three consecutive frequencies and five consecutive wavenumbers. We then varied the number of transmitters grouped together into sets of 1, 11 and 51 transmitters, as shown in Table 2. We also ran a test of only a single frequency (0.3 Hz) to study how fast MARE2DEM performs for a reduced task. Depending on the number of frequencies under consideration and the number of transmitters grouped together, the number of total tasks varies from 30 (for one frequency and all 51 transmitters grouped together) to 18,360 (for 12 frequencies and each transmitter considered separately). Due to mesh sharing, the number of refinement tasks is much lower, spanning from six to 1224. We ran MARE2DEM for these parameterizations using from one to 800 parallel processors.

Fig. 11 shows the corresponding wall clock run-times for the scaling tests. All runs exhibit a linear decrease in run-time that is proportional to the number of processors, up until the number of

processors exceeds the number of refinement tasks, after which the run-time flattens out regardless of the number of processors used. The linear decrease in run-time is due to the embarrassingly parallel nature of the problem, where the low communications overhead allows for highly efficient parallel scaling. The flattening in run-time with additional processors occurs because the refinement tasks and mesh sharing tasks have been completed and no further speed-ups are possible with additional processors. For example, consider run (b), which required 120 refinement tasks and an additional 1680 computations when the final grids are shared with nearby parameters. The scaling tests show a linear decrease in run-time up until 120 processors were used, after which the increase in the number of processors does not further reduce the run-time, despite the additional 1680 mesh sharing tasks. This is simply explained by the variability of run-time required for each refinement task, which depends on the particular data parameters of the task. When a processor running a refinement task finishes, it then become available to compute any mesh sharing tasks. For run 'b', when the last (120th) adaptive refinement task completes, all other mesh sharing tasks have already been handled by the other nodes and the only remaining tasks are to compute its own mesh sharing computations. Therefore, no further speed improvements are possible from additional processors.

The scaling tests point out a few different ways that MARE2DEM can be used to model large 2D CSEM problems in as short a time as possible. For very large clusters containing hundreds of processors or more, the fastest run-times can be obtained by using small groups of transmitters, producing run-times of under a minute for typical CSEM survey parameters when using a few hundred nodes. Conversely, on smaller multi-core desktop computers that a typical working group might maintain (of order 10 processors), larger transmitter groups can be used to get run-times as short as a few minutes. In our tests here, the fastest run-time of only 23 seconds was obtained using 800 processors when only one frequency was modeled and each transmitter was grouped separately. We can predict that on a larger parallel system of at least 9600 processors, the full problem containing all twelve frequencies could be computed in as short as 23 s.

Fig. 12 shows an example of the error estimator μ_τ computed for the starting mesh (6974 vertices) and the final refined grid (26,065 vertices) for a particular transmitter location, frequency and wavenumber. The estimated error is in excess of the requested 1% tolerance over most of the domain in the starting mesh, while for the final mesh the error is less than the requested 1% tolerance. There is a significant amount of refinement around each EM receiver, as well as in the vicinity of the transmitter at 300 km position. Significant refinement is also observed in the seawater column and air layer proximal to the sea surface.

The final refined grids required a variable number of vertices, depending on the number of transmitters grouped and the particular frequency and wavenumber. The final grids for run (a) required be-

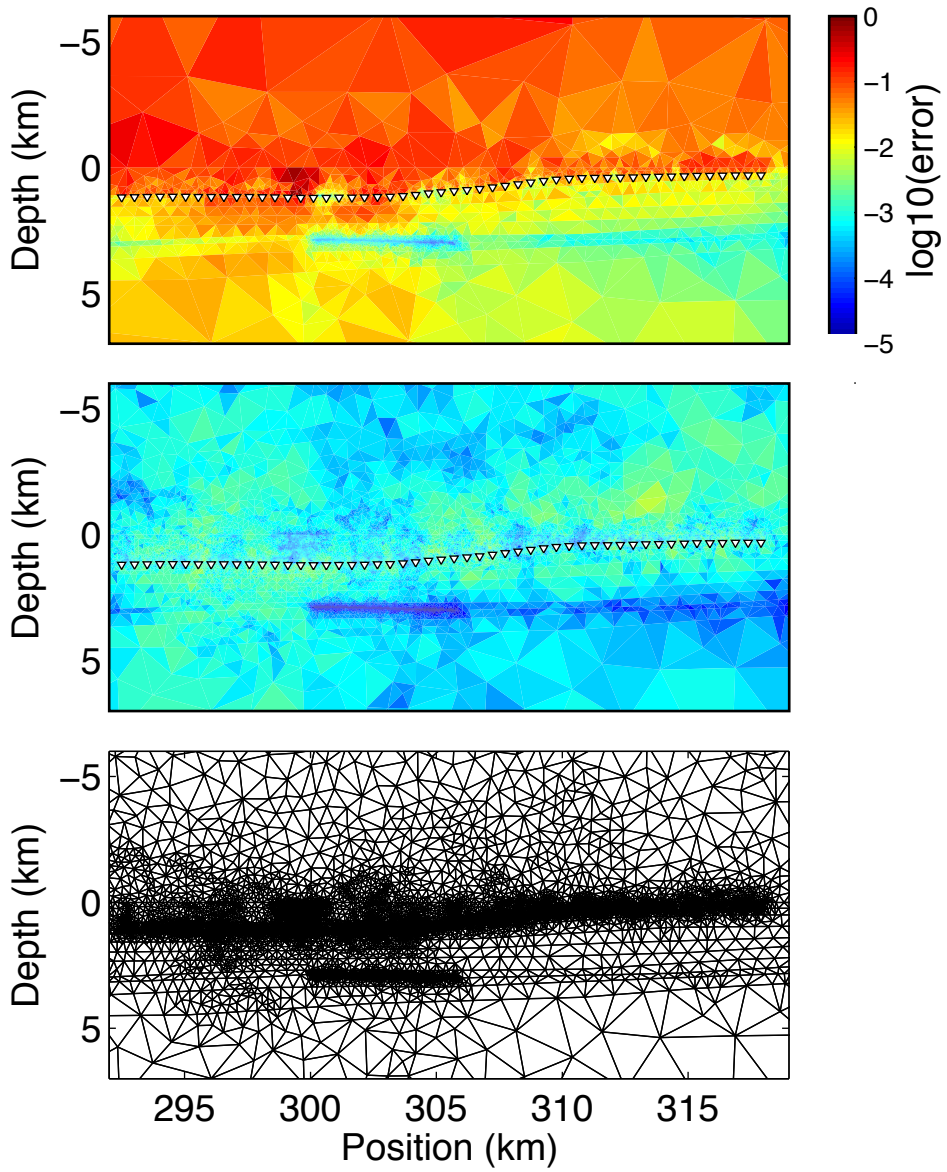

Figure 12. Estimated error μ_τ for the starting grid (top panel, 6974 vertices) and the final refined grid (middle, 26,065 vertices) computed for a transmitter at 300 km position, $k_x = .000016$ m^{-1} and $f = 0.3$ Hz. The bottom panel shows the mesh refinement for the final refined grid. Inverted white triangles show the location of the seafloor EM receivers. Only the central region of the modeling domain is shown.

tween 12,000 to 26,000 vertices, run (b) required 23,000-35,000 vertices, and run (c) required between 37,000-100,000 vertices. This scaling in mesh size is expected since the larger number of transmitters grouped together for runs (b) and (c) require more vertices to accurately capture the rapid decay of the EM fields around each additional transmitter.

Table 3 provides a representative example of the adaptive refinement performance when the starting mesh is refined for a single transmitter at 0.3 Hz and wavenumber 0.001 m^{-1}. Five iterations of

Table 3. Performance example of the MARE2DEM adaptive refinement for a single transmitter with $f = 0.3$ Hz and $k_x = 0.001$ m^{-1}. Grid size, maximum estimated element error μ_τ and run-time are listed for the refinement iterations (grids 1-4) and the final computation (grid 5).

Grid	# Vertices	max$\{\mu_\tau\}$ (%)	Run-time (s)	Cumulative (s)
1	6974	92.6	0.7	0.7
2	12464	23.2	1.4	2.1
3	22887	4.8	2.5	4.6
4	23509	1.5	2.6	7.2
5	23760	1	2.0	9.2

grid refinement were required to reach the target 1% error estimate, requiring a total run-time of 9.2 s. The adaptive refinement iterations dominated about 80% of the run-time, whereas the final grid required an additional 2.0 s, or roughly 20% additional time. When the final grid is shared with nearby frequencies and wavenumbers, only the final run-time of 2.0 s is required for these computations, which illustrates how mesh sharing can lead to significantly faster run-times. The table also documents the effectivity of the error estimator, as noted by the significant decrease in max$\{\mu_\tau\}$ with each iteration.

Fig. 13 shows an example of the inline electric field responses computed at 0.3 Hz. The responses for all 51 receivers and transmitters are shown for computations that included the three resistive stacked reservoir layers and for computations where the reservoir layers were removed. The responses show a significant amount of structure, particularly at amplitudes around 10^{-15} to 10^{-13} V/Am^2, where a large change in slope occurs with the arrival of the airwave. This typically occurs at large source-receiver offsets, where the responses become dominated by energy that has traveled along the air-sea interface (e.g., Constable & Weiss 2006).

Due to the strong attenuation of the fields with distance from the transmitter, the differences between the responses with and without the reservoir are difficult to identify when viewed on an absolute scale. Fig. 14 shows the relative differences between these responses in a midpoint-offset section, as shown for four of the twelve frequencies computed. These sections are generated by plotting the relative difference between the reservoir and no-reservoir responses horizontally along the transmitter-receiver midpoint, and vertically at the absolute transmitter-receiver offset. Since longer offsets have a broader and deeper sensitivity, these plots are sometimes referred to as anomaly pseudosections. We can now see that the peak signal from the reservoir occurs at transmitter-receiver offsets of about four to eight kilometers and is largely confined to source-receiver midpoints located over the reservoir. While the largest anomaly is observed at 1.1 Hz, most of the signal at this frequency is below the

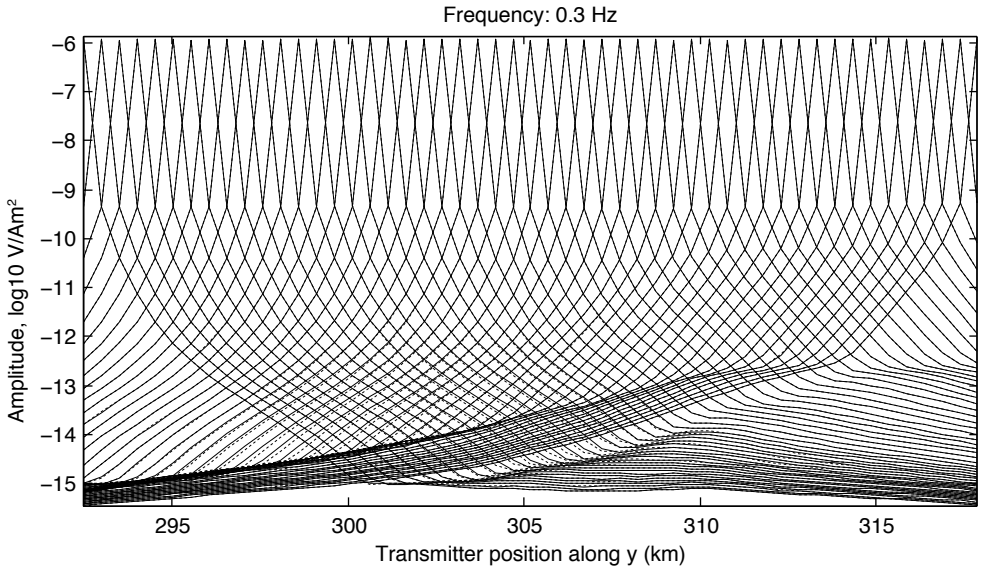

Figure 13. Horizontal electric field amplitude responses for the complex model with a 0.3 Hz inline electric field transmitter. Solid lines show the responses for the complex model in Fig. 10 and dashed lines show the responses for the same model after removal of the three stacked reservoir layers.

typical marine CSEM system noise-floor of about 10^{-15} V/Am^2. This is due to the rapid inductive attenuation of high frequency data with source-receiver offset. Conversely, the lower frequency of 0.3 Hz experiences less attenuation yet still contains a measurable 20-30% anomaly, suggesting that this frequency would be optimal for imaging the reservoir.

8 CONCLUSIONS

The parallel adaptive finite element algorithm presented here can be used for rapid and accurate simulations of 2D EM problems. The unstructured triangular grid readily incorporates complex structures with an efficient use of mesh vertices and without the discretization difficulties and cumbersome approximations experienced with rectangular grids. Our use of an automated adaptive refinement method based on a goal-oriented error estimator allows the MARE2DEM program to generate reliable EM responses, even for finite element novices who may have little or no experience with mesh design. The LU matrix factorization results in rapid solutions of the additional systems associated with the error estimator, and also allows for efficient computations when multiple transmitters are modeled. We showed that the parallelization over subsets of transmitters, receivers, wavenumbers and frequencies scales very efficiently for clusters of up to at least 1000 processors. Mesh sharing for nearby frequencies and wavenumbers can be used to reduce the parallel load and offers a method for fast solutions when only a smaller number of parallel processors are available.

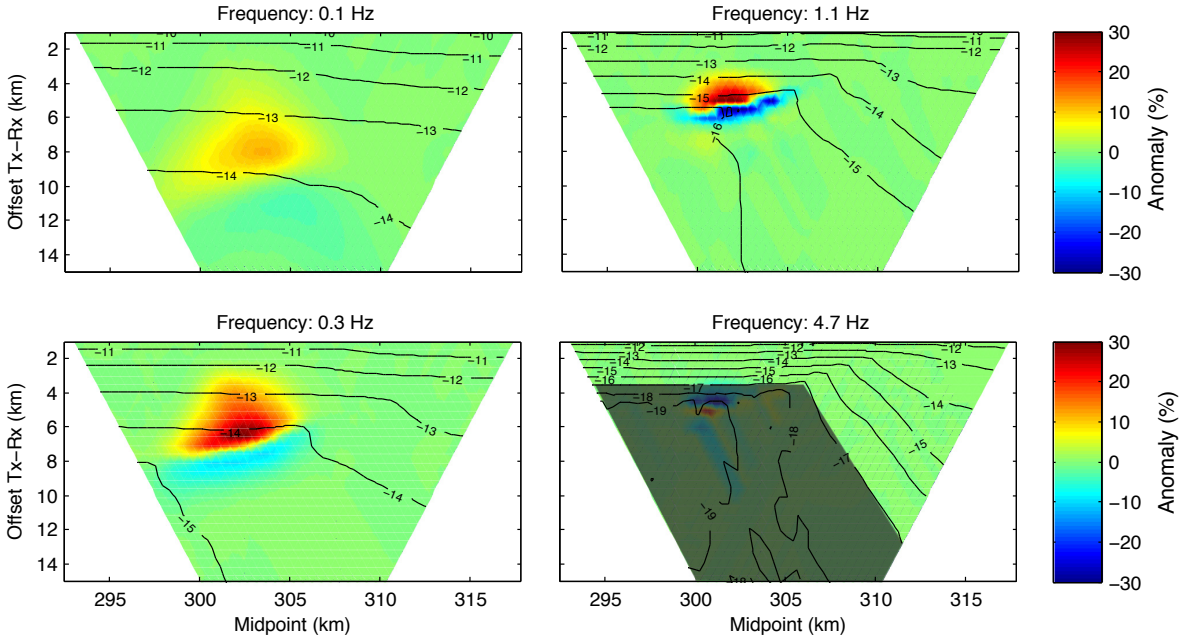

Figure 14. Electric field anomaly midpoint-offset sections for the complex model shown for four frequencies. The relative difference between the complex model response and the response without the reservoir layers is plotted horizontally at the source-receiver mid-point and vertically at the source-receiver offset. Since the responses for the sources and receivers are nearly reciprocal, only positive offsets are shown. Black contour lines show the absolute magnitude of the responses in units of $\log_{10}$ V/Am^2. The shaded area for the 4.7Hz solution covers the region where the solution is below the computational noise floor.

Our final example demonstrated how MARE2DEM can automatically compute EM responses for a realistic offshore model with realistic data parameters in run times of under one minute when run on a large cluster. Further speed improvements may be possible by incorporating a parallel algorithm for the LU factorizations, although the performance increase from this would be modest compared to the high efficiency of the independent parallel EM computations described here. However, if a significantly large parallel system is available, such that all embarrassingly parallel aspects of the problem are exploited and additional processors are still available, then parallel matrix factorizations may be a worthwhile consideration.

While the fast performance of MARE2DEM will be beneficial for the geophysicist interested in forward modeling highly detailed 2D problems, we anticipate that this algorithm will prove more useful as a kernel routine for non-linear inversions. Gradient based inversions for 2D EM have been around for some time (e.g., Unsworth & Oldenburg 1995), yet Bayesian methods have long been considered impractical since they can require tens of thousands of forward computations. The run-times of our complex test model show that Bayesian methods for realistic 2D EM problems are now tractable

on large clusters. In addition to inversion performance improvements from the parallel implementation, the automated adaptive meshing may lead to stability improvements, where an accurate mesh is automatically generated as model structure evolves throughout the inversion process, thereby insuring numerical accuracy is not compromised. An initial test of using MARE2DEM for a stochastic sharp boundary inversion for 2D MT problems is given in Chen et al. (2010). Finally, the adaptive methods we proposed here could be extended to finite element methods developed for three dimensional electromagnetic problems (e.g., Monk 2003).

ACKNOWLEDGMENTS

Kerry Key acknowledges funding support from BHP Billiton Petroleum, Woodside Petroleum and the Seafloor Electromagnetic Methods Consortium at Scripps Institution of Oceanography. Jeff Ovall thanks the Max Planck Institute for Mathematics, in Leipzig, Germany, for graciously hosting him while this manuscript was being completed. Thanks are also due to the University of Kentucky Mathematics Department, and the College of Arts and Sciences, for providing additional funding during the final stages of this project, and to Michael Holst for helpful conversation concerning Gårding inequalities. Jonathan Shewchuk and Matthew Kennel are acknowledged for making their codes Triangle and KDTREE2 freely available. The San Diego Supercomputer Center at UCSD provided access to the Triton Compute Cluster. G. Michael Hoversten and Jinsong Chen gave useful feedback on the parallel implementation of the MARE2DEM algorithm. This work benefited from the helpful suggestions of Christoph Schwarzbach and two anonymous reviewers.

REFERENCES

Abubakar, A., Habashy, T. M., Druskin, V. L., Knizhnerman, L., & Alumbaugh, D., 2008. 2.5D forward and inverse modeling for interpreting low-frequency electromagnetic measurements, *Geophysics*, **73**(4), F165–F177.

Adams, R. A. & Fournier, J. J. F., 2003. *Sobolev spaces*, vol. 140 of **Pure and Applied Mathematics (Amsterdam)**, Elsevier/Academic Press, Amsterdam, 2nd edn.

Ainsworth, M. & Oden, J. T., 2000. *A posteriori error estimation in finite element analysis*, Pure and Applied Mathematics (New York), Wiley-Interscience [John Wiley & Sons], New York.

Anderson, W. L., 1979. Numerical-integration of related Hankel transforms of orders 0 and 1 by adaptive digital filtering, *Geophysics*, **44**(7), 1287–1305.

Babuška, I. & Miller, A., 1984a. The post-processing approach in the finite-element method. 1. calculation of displacements, stresses and other higher derivatives of the displacements, *Internat. J. Numer. Methods Engrg.*, **20**(6), 1085–1109.

Babuška, I. & Miller, A., 1984b. The post-processing approach in the finite-element method. 2. the calculation of stress intensity factors, *Internat. J. Numer. Methods Engrg.*, **20**(6), 1111–1129.

Babuška, I. & Miller, A., 1984c. The post-processing approach in the finite-element method. 3. a posteriori error-estimates and adaptive mesh selection, *Internat. J. Numer. Methods Engrg.*, **20**(12), 2311–2324.

Bank, R. E., 1996. Hierarchical bases and the finite element method, in *Acta numerica, 1996*, vol. 5 of **Acta Numer.**, pp. 1–43, Cambridge Univ. Press, Cambridge.

Bank, R. E. & Xu, J. C., 2003. Asymptotically exact a posteriori error estimators, Part II: General unstructured grids, *SIAM Journal on Numerical Analysis*, **41**, 2313–2332.

Becker, R. & Rannacher, R., 2001. An optimal control approach to a posteriori error estimation in finite element methods, *Acta Numer.*, **10**, 1–102.

Chen, J., Hoversten, G.M., Key, K., & Nordquist, G., 2010. Stochastic inversion of 2D magnetotelluric data using sharp boundary parameterization, *SEG Technical Program Expanded Abstracts*, **29**(1), 609-613.

Constable, S. & Weiss, C. J., 2006. Mapping thin resistors and hydrocarbons with marine EM methods: Insights from 1D modeling, *Geophysics*, **71**(2), G43–G51.

Constable, S., Key, K., & Lewis, L., 2009. Mapping offshore sedimentary structure using electromagnetic methods and terrain effects in marine magnetotelluric data, *Geophysical Journal International*, **176**(2), 431–442.

Demmel, J. W., Eisenstat, S. C., Gilbert, J. R., Li, X. S., & Liu, J. W. H., 1999. A supernodal approach to sparse partial pivoting, *SIAM J. Matrix Analysis and Applications*, **20**(3), 720–755.

Devroye, L., Lemaire, C., & Moreau, J. M., 2004. Expected time analysis for delaunay point location, *Computational Geometry-Theory And Applications*, **29**(2), 61–89.

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.

Eisenberg, M. & Malvern, L., 1973. On finite element integration in natural co-ordinates, *International Journal for Numerical Methods in Engineering*, **7**(4), 574–575.

Estep, D., Holst, M., & Larson, M., 2005. Generalized Green's functions and the effective domain of influence, *SIAM J. Sci. Comput.*, **26**(4), 1314–1339 (electronic).

Evans, L. C., 2010. *Partial differential equations*, vol. 19 of **Graduate Studies in Mathematics**, American Mathematical Society, Providence, RI, 2nd edn.

Everett, M. & Edwards, R., 1992. Transient marine electromagnetics: the 2.5-D forward problem, *Geophysical Journal International*, **113**, 545–561.

Franke, A., Börner, R.U., & Spitzer, K., 2007. Adaptive unstructured grid finite element simulation of two-dimensional magnetotelluric fields for arbitrary surface and seafloor topography, *Geophysical Journal International*, **171**(1), 71–86.

Golub, G. H. & Van Loan, C. F., 1996. *Matrix computations*, Johns Hopkins University Press, Baltimore, 3rd edn.

Guptasarma, D. & Singh, B., 1997. New digital linear filters for Hankel J(0) and J(1) transforms, *Geophysical Prospecting*, **45**(5), 745–762.

Hackbusch, W., 2010. *Elliptic differential equations*, vol. 18 of **Springer Series in Computational Mathematics**, Springer-Verlag, Berlin, english edn., Theory and numerical treatment, Translated from the 1986 corrected German edition by Regine Fadiman and Patrick D. F. Ion.

Ihlenburg, F., 1998. *Finite element analysis of acoustic scattering*, vol. 132 of **Applied Mathematical Sciences**, Springer-Verlag, New York.

Kennel, M. B., 2004. KDTREE 2: Fortran 95 and C++ software to efficiently search for near neighbors in a multi-dimensional Euclidean space, *ArXiv*, p. physics/0408067v2.

Key, K., 2009. 1D inversion of multicomponent, multifrequency marine CSEM data: Methodology and synthetic studies for resolving thin resistive layers, *Geophysics*, **74**(2), F9–F20.

Key, K. & Constable, S., 2011. Coast effect distortion of marine magnetotelluric data: Insights from a pilot study offshore northeastern Japan, *Physics of the Earth and Planetary Interiors*, in press.

Key, K. & Weiss, C., 2006. Adaptive finite element modeling using unstructured grids: the 2D magnetotelluric example, *Geophysics*, **71**(6), G291–G299.

Kong, F. N., Johnstad, S. E., Røsten, T., & Westerdahl, H., 2008. A 2.5D finite-element-modeling difference method for marine CSEM modeling in stratified anisotropic media, *Geophysics*, **73**(1), F9–F19.

Lee, K. H. & Morrison, H. F., 1985. A numerical-solution for the electromagnetic scattering by a two-dimensional inhomogeneity, *Geophysics*, **50**(3), 466–472.

Li, Y. & Constable, S., 2007. 2D marine controlled-source electromagnetic modeling: Part 2 — the effect of bathymetry, *Geophysics*, **72**(2), WA63–WA71.

Li, Y. & Key, K., 2007. 2D marine controlled-source electromagnetic modeling: Part 1— An adaptive finite element algorithm, *Geophysics*, **72**(2), WA51–WA62.

Li, Y. & Pek, J., 2008. Adaptive finite element modelling of two-dimensional magnetotelluric fields in general

anisotropic media, *Geophysical Journal International*, **175**(3), 942–954.

Mackie, R. L., Madden, T. R., & Wannamaker, P. E., 1993. Three-dimensional magnetotelluric modeling using difference equations - Theory and comparisons to integral equation solutions, *Geophysics*, **58**(02), 215–226.

Mitsuhata, Y., 2000. 2-D electromagnetic modeling by finite-element method with a dipole source and topography, *Geophysics*, **65**(2), 465–475.

Monk, P., 2003. *Finite element methods for Maxwell's equations*, Numerical Mathematics and Scientific Computation, Oxford University Press, New York.

Myer, D., Constable, S., & Key, K., 2011. Broad-band waveforms and robust processing for marine CSEM surveys, *Geophysical Journal International*, **184**, 689–698.

Newman, G. A. & Alumbaugh, D. L., 1995. Frequency-domain modelling of airborne electromagnetic responses using staggered finite differences, *Geophys. Prosp.*, **43**(8), 1021–1042.

Nitsche, J. A. & Schatz, A. H., 1974. Interior estimates for Ritz-Galerkin methods, *Math. Comp.*, **28**, 937–958.

Orange, A., Key, K., & Constable, S., 2009. The feasibility of reservoir monitoring using time-lapse marine CSEM, *Geophysics*, **74**(2), F21–F29.

Ovall, J. S., 2004. *Duality-Based Adaptive Refinement for Elliptic PDEs*, Ph.D. thesis, University of California, San Diego, La Jolla, CA 92093, USA.

Ovall, J. S., 2006. Asymptotically exact functional error estimators based on superconvergent gradient recovery, *Numer. Math.*, **102**(3), 543–558.

Ovall, J. S., 2007. Function, gradient, and Hessian recovery using quadratic edge-bump functions, *SIAM J. Numer. Anal.*, **45**(3), 1064–1080 (electronic).

Schwalenberg, K. & Edwards, R. N., 2004. The effect of seafloor topography on magnetotelluric fields: an analytic formulation confirmed with numerical results, *Geophysical Journal International*, **159**, 607–621.

Shewchuk, J., 2002a. What is a good linear finite element? Interpolation, conditioning, anisotropy, and quality measures, *unpublished preprint*.

Shewchuk, J. R., 1996. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator, in *Applied Computational Geometry: Towards Geometric Engineering*, vol. 1148 of **Lecture Notes in Computer Science**, pp. 203–222, eds Lin, M. C. & Manocha, D., Springer-Verlag, From the First ACM Workshop on Applied Computational Geometry.

Shewchuk, J. R., 2002b. Delaunay refinement algorithms for triangular mesh generation, *Computational Geometry–Theory and Applications*, **22**, 21–74.

Stoyer, C. H. & Greenfield, R. J., 1976. Numerical-solutions of response of a 2-dimensional earth to an oscillating magnetic dipole source, *Geophysics*, **41**(3), 519–530.

Unsworth, M. & Oldenburg, D., 1995. Subspace inversion of electromagnetic data - application to mid-ocean-ridge exploration, *Geophysical Journal International*, **123**(1), 161–168.

Unsworth, M. J., Travis, B. J., & Chave, A. D., 1993. Electromagnetic induction by a finite electric dipole source over a 2-D earth, *Geophysics*, **58**(02), 198–214.

Wahlbin, L. B., 1991. Local behavior in finite element methods, in *Handbook of numerical analysis, Vol. II*,

Handb. Numer. Anal., II, pp. 353–522, North-Holland, Amsterdam.

Wannamaker, P. E., Stodt, J. A., & Rijo, L., 1986. Two-dimensional topographic responses in magnetotellurics modeled using finite elements, *Geophysics*, **51**(11), 2131–2144.

Wannamaker, P. E., Stodt, J. A., & Rijo, L., 1987. A stable finite-element solution for two-dimensional magnetotelluric modeling, *Geophysical Journal of the Royal Astronomical Society*, **88**, 277–296.

Ward, S. H. & Hohmann, G. W., 1988. Electromagnetic theory for geophysical applications, in *Electromagnetic methods in applied geophysics*, pp. 131–312, ed. Nabighian, M. N., Soc. Expl. Geophys., 01st edn.

Weaver, J., LeQuang, B., & Fischer, G., 1985. A comparison of analytic and numerical results for a two-dimensional control model in electromagnetic induction .1. B-polarization calculations, *Geophysical Journal of the Royal Astronomical Society*, **82**, 263–277.

Weaver, J., LeQuang, B., & Fischer, G., 1986. A comparison of analytical and numerical results for a 2-D control model in electromagnetic induction .2. E-polarization calculations, *Geophysical Journal of the Royal Astronomical Society*, **87**, 917–948.

Zhu, Y. & Cangellaris, A. C., 2006. *Multigrid Finite Element Methods for Electromagnetic Field Modeling*, John Wiley and Sons Ltd.

APPENDIX A: FINITE ELEMENT DETAILS

This brief appendix is intended as an aid for those interested in the finite element details required for the implementation and testing of our goal oriented adaptive method. Because the solution of the discrete dual/adjoint problems (25) and (26) involves the adjoints of the matrices needed for the solution of the discrete primal problems (19) and (21), we only describe those for the former.

Let $\mathcal{T}$ be a given triangulation, having vertices $\mathcal{V}$ and edges $\mathcal{E}$. For any vertex $z \in \mathcal{V}$ we define a function l_z, which is linear on every triangle, and which has value 1 at z and value 0 at every other vertex—in brief, $l_z(z') = \delta_{zz'}$ for all $z, z' \in \mathcal{V}$. From this relation, it is clear that l_z is identically 0 on any triangle which does not have z as a vertex. If τ is a triangle which has z as a vertex, and we number the vertices of τ counter-clockwise starting at z, then the formula for l_z *on* τ is given in eqs. (A.3)-(A.7)—the link is that $l_z = l_1$ on τ. Because of the key relation $l_z(z') = \delta_{zz'}$, any continuous, piecewise-linear scalar-function v on Ω can be described uniquely in terms of the basis $\{l_z\}$ by $v = \sum_{z\in\mathcal{V}} v_z l_z$, where $v_z = v(z)$. More specifically, the finite element solution $\mathbf{u}_n = (u_{n,1}, u_{n,2})$ of (19) can be expressed in the form

$$u_{n,1} = \sum_{z\in\mathcal{V}_I} u_{z,1} l_z \ , \quad u_{n,2} = \sum_{z\in\mathcal{V}_I} u_{z,2} l_z \ , \tag{A.1}$$

where $\mathcal{V}_I$ are the *interior* vertices in the triangulation—our function vanishes on the boundary. After choosing a global ordering of the interior vertices, the finite element solution is found by computing its coefficients $\{u_{z,1} : \ z \in \mathcal{V}_I\}$ and $\{u_{z,2} : \ z \in \mathcal{V}_I\}$ via solving the (2×2 block) linear system

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{pmatrix} \ , \tag{A.2}$$

where $\mathbf{u}_1$ is the vector of coefficients for $u_{n,1}$, $\mathbf{u}_2$ is the vector of coefficients for $u_{n,2}$, and

- the entries of A_{11} are of the form: $\int_\Omega \sigma(\lambda \nabla l_{z'} \cdot \nabla l_z + l_{z'} l_z)\, dV$,
- the entries of A_{22} are of the form: $\int_\Omega i\omega\mu(\lambda \nabla l_{z'} \cdot \nabla l_z + l_{z'} l_z)\, dV$,
- the entries of $A_{12} = A_{21}$ are of the form: $\int_\Omega ik_x\lambda\, (R\nabla l_{z'}) \cdot \nabla l_z \, dV$,
- the entries of $\mathbf{f}_k$ are of the form: $\int_\Omega f_k l_z \, dV$ for $k = 1, 2$.

If N is the number of interior vertices, we see that this system is $2N \times 2N$.

Similarly, for any edge $e \in \mathcal{E}$, let $z \in \mathcal{V}$ and $z' \in \mathcal{V}$ be its vertices. We define $q_e = 4 l_z l_{z'}$, and it is clear that q_e is only nonzero on one or (generally) two triangles having e as an edge. Again, if we pick one of these triangles and number its edges and vertices counter-clockwise as before, we have a formula for q_e (A.8) on that triangle. Our chosen basis for W_n is $\{q_e : \ e \in \mathcal{E}_I\}$, where $\mathcal{E}_I$ is the set of interior edges. The computation of $\boldsymbol{\varepsilon}_n$ in terms of this basis involves the solution of a $2M \times 2M$ system analogous to (A.2), where M is the number of interior edges. As we see below in (A.15),

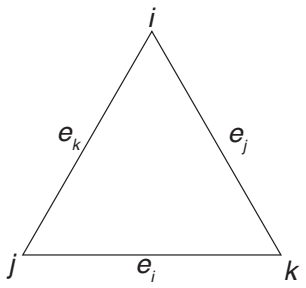

Figure A1. Finite element for triangle τ with vertices i, j, k and edges e_i, e_j, e_k opposite the vertices.

for this system the analogues of A_{12} and A_{21} are identically 0—i.e. the system naturally decouples. Though we do not explicitly give the righthand side for this system, it is easily derived from (21), and is constructed by computing integrals on triangles of the forms (A.16)-(A.18). Below we outline the rudimentary aspects of how the integrations are carried out for each element.

For a triangular element τ with vertices i, j, k and edges e_i, e_j, e_k opposite those vertices (Fig. A1), the linear basis function l_i associated with vertex i varies from value 1 at vertex i to zero at the other two vertices according to

$$l_i = \frac{1}{2|\tau|}(a_i + b_i y + c_i z), \tag{A.3}$$

where the area of the triangle $|\tau|$ is defined by

$$2|\tau| = \begin{vmatrix} 1 & 1 & 1 \\ y_i & y_j & y_k \\ z_i & z_j & z_k \end{vmatrix} = (a_i + b_i y_i + c_i z_i), \tag{A.4}$$

and

$$a_i = y_j z_k - y_k z_j, \tag{A.5}$$

$$b_i = z_j - z_k, \tag{A.6}$$

$$c_i = y_k - y_j, \tag{A.7}$$

with cyclic permutation of i, j, k. The quadratic bump basis function associated with edge i is

$$q_i = 4 l_j l_k, \tag{A.8}$$

again with cyclic permutation of i, j, k. The bump function q_i is zero at all three vertices and has a quadratic shape along edge e_i.

The details of forming the linear systems follow the standard approaches for the finite element method (e.g., Zhu & Cangellaris 2006), so here we only provide a few details that are relevant to our

implementation. Using the well known formula for polynomial integration on triangles (e.g., Eisenberg & Malvern 1973):

$$\int_\tau l_i^p l_j^q l_k^r d\tau = \frac{2|\tau| p! q! r!}{(p+q+r+2)!}, \tag{A.9}$$

we find the following formulas for the integrals required by the finite element equations. For the linear basis functions l_i, these are

$$\int_\tau \nabla l_i \cdot \nabla l_j = \frac{1}{4|\tau|}\left(b_i b_j + c_i c_j\right) \tag{A.10}$$

$$\int_\tau l_i l_j = \begin{cases} \frac{|\tau|}{6}, & i = j \\ \frac{|\tau|}{12}, & i \neq j \end{cases}, \tag{A.11}$$

$$\int_\tau (R\nabla l_i) \cdot \nabla l_j = \frac{1}{4|\tau|}\left(b_i c_j - b_j c_i\right). \tag{A.12}$$

Here, we recall the rotation matrix R described in Section 2. For the quadratic basis functions q_i, we have

$$\int_\tau \nabla q_i \cdot \nabla q_j = \begin{cases} \frac{2}{3|\tau|}\left(b_i^2 - b_j b_k + c_i^2 - c_j c_k\right), & i = j \\ \frac{2}{3|\tau|}\left(b_i b_j + c_i c_j\right), & i \neq j \end{cases}, \tag{A.13}$$

$$\int_\tau q_i q_j = \begin{cases} \frac{8}{45}|\tau|, & i = j \\ \frac{4}{45}|\tau|, & i \neq j \end{cases}, \tag{A.14}$$

$$\int_\tau (R\nabla q_i) \cdot \nabla q_j = 0\,. \tag{A.15}$$

This last integral is particularly useful since it eliminates the cross coupling between $\hat{E}_x$ and $\hat{H}_x$ in the linear systems using the quadratic bump basis, and hence each field can be solved separately using $M \times M$ sparse symmetric linear systems, rather than the larger $2M \times 2M$ systems required by the coupled problem. The right hand side of (21), (26) and (29) require the integral formulas

$$\int_\tau \nabla l_i \cdot \nabla q_j = -\frac{1}{3|\tau|}\left(b_i b_j + c_i c_j\right), \tag{A.16}$$

$$\int_\tau l_i q_j = \begin{cases} \frac{2|\tau|}{15}, & i \neq j \\ \frac{|\tau|}{15}, & i = j \end{cases}, \tag{A.17}$$

$$\int_\tau (R\nabla l_i) \cdot \nabla q_j = \frac{1}{3|\tau|}\left(b_j c_i - b_i c_j\right). \tag{A.18}$$

Of course, $(R\nabla q_j) \cdot \nabla l_i = -(R\nabla l_i) \cdot \nabla q_j$, so this last identity can be used with linear and quadratic functions reversed by changing sign.

APPENDIX B: WELL-POSEDNESS OF THE VARIATIONAL PROBLEMS

In low frequency EM geophysics applications it is legitimate to take $\epsilon = 0$, and we have done so in our numerical experiments. For the analysis below, we make this assumption as well. In this case, $\lambda^{-1} = k_x^2 - i\omega\mu\hat{\sigma}$, and the "diffusion matrix" A in our bilinear form B is given by $A = A_1 + iA_2$, where

$$A_1 = \frac{1}{k_x^4 + (\omega\mu\hat{\sigma})^2} \begin{pmatrix} k_x^2\hat{\sigma}I & ik_x^3 R \\ ik_x^3 R & -(\omega\mu)^2\hat{\sigma}I \end{pmatrix} \quad \text{and} \quad A_2 = \frac{1}{k_x^4 + (\omega\mu\hat{\sigma})^2} \begin{pmatrix} \omega\mu\hat{\sigma}^2 I & ik_x\omega\mu\hat{\sigma}R \\ ik_x\omega\mu\hat{\sigma}R & k_x^2\omega\mu I \end{pmatrix} .$$

We will first argue that the problems (17), (19) and (21) can have *at most* one solution, by showing that the only solution of "Find $\mathbf{u} \in \mathcal{H}$ such that $B(\mathbf{u}, \mathbf{v}) = 0$ for all $\mathbf{v} \in \mathcal{H}$" is $\mathbf{u} = 0$. In particular, we will show that $B(\mathbf{u}, \mathbf{u}) = 0$ only if $\mathbf{u} = \mathbf{0}$.

For $a, b, c \in \mathbb{R}$, the Hermitian matrix

$$M = \begin{pmatrix} aI & ibR \\ ibR & cI \end{pmatrix} \tag{B.1}$$

has eigenvalues

$$\lambda_1 = \lambda_2 = \frac{1}{2}\left(a + c - \sqrt{(a-c)^2 + 4b^2}\right) , \tag{B.2}$$

$$\lambda_3 = \lambda_4 = \frac{1}{2}\left(a + c + \sqrt{(a-c)^2 + 4b^2}\right) , \tag{B.3}$$

and corresponding eigenvectors

$$\boldsymbol{\psi}_1 = \{2ib, 0, 0, a - c + \sqrt{(a-c)^2 + 4b^2}\} , \tag{B.4}$$

$$\boldsymbol{\psi}_2 = \{0, -2ib, a - c + \sqrt{(a-c)^2 + 4b^2}, 0\} , \tag{B.5}$$

$$\boldsymbol{\psi}_3 = \{a - c + \sqrt{(a-c)^2 + 4b^2}, 0, 0, 2ib\} , \tag{B.6}$$

$$\boldsymbol{\psi}_4 = \{0, a - c + \sqrt{(a-c)^2 + 4b^2}, -2ib, 0\} . \tag{B.7}$$

In the case of A_2 above, $\lambda_1 = \lambda_2 = 0$, $\lambda_3 = \lambda_4 = \omega\mu(k_x^2 + \hat{\sigma}^2)/(k_x^4 + (\omega\mu\hat{\sigma})^2) > 0$, and its eigenvectors (after rescaling) are

$$\boldsymbol{\psi}_1 = \{ik_x, 0, 0, \hat{\sigma}\} \quad , \quad \boldsymbol{\psi}_2 = \{0, -ik_x, \hat{\sigma}, 0\} \quad , \quad \boldsymbol{\psi}_3 = \{\hat{\sigma}, 0, 0, ik_x\} \quad , \quad \boldsymbol{\psi}_4 = \{0, \hat{\sigma}, -ik_x, 0\} .$$

For $B(\mathbf{u}, \mathbf{u}) = 0$ to hold, it must be the case that

$$\Im[B(\mathbf{u}, \mathbf{u})] = \int_\Omega A_2\nabla\mathbf{u} : \nabla\overline{\mathbf{u}} + \omega\mu|u_2|^2 \, dV = 0 .$$

Since $A_2\nabla\mathbf{u} : \nabla\overline{\mathbf{u}} \geq 0$, this implies that $u_2 = 0$. So $\nabla\mathbf{u} = (\partial_y u_1, \partial_z u_1, 0, 0)$. The only way that $A_2\nabla\mathbf{u} : \nabla\overline{\mathbf{u}} = 0$ is if $\nabla\mathbf{u} \in \mathrm{span}(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2)$, but $\hat{\sigma} > 0$ (throughout the domain), so this implies that

$\nabla \mathbf{u} = \mathbf{0}$. Finally, the Dirichlet boundary conditions force $\mathbf{u} = \mathbf{0}$. Therefore, our variational problems can have at most one solution.

To see that they have precisely one solution, we will make an argument based on a Gårding inequality (cf. Hackbusch 2010; Ihlenburg 1998). It is clear that multiplying the eqs. (17), (19) and (21) by a non-zero constant $\alpha - i\beta$ has no effect on whether or not these problems are well-posed, and doing so will make the analysis cleaner. We define $\tilde{B}(\mathbf{v}, \mathbf{w}) = (\alpha - i\beta)B(\mathbf{v}, \mathbf{w})$ and $\tilde{F}(\mathbf{w}) = (\alpha - i\beta)F(\mathbf{w})$. Recalling our (physically realistic) assumption that $\lambda^{-1} \neq 0$, it is clear that both the sesquilinear form $\tilde{B}$ and the functional $\tilde{F}$ are *bounded*: there are constants $L, M > 0$ such that $|\tilde{F}(\mathbf{v})| \leq L\|\mathbf{v}\|_1$ and $|\tilde{B}(\mathbf{v}, \mathbf{w})| \leq M\|\mathbf{v}\|_1\|\mathbf{w}\|_1$ for all $\mathbf{v}, \mathbf{w} \in \mathcal{H}$. We will argue that $\tilde{B}$ satisfies a Gårding inequality as well: there are constants $m_0, m_1 > 0$ for which

$$\Im[\tilde{B}(\mathbf{v}, \mathbf{v})] + m_0\|\mathbf{v}\|_0^2 \geq m_1\|\mathbf{v}\|_1^2 \text{ for all } v \in \mathcal{H} \,. \tag{B.8}$$

Here, and below, $\|\mathbf{v}\|_0^2 = \int_\Omega |\mathbf{v}|^2 \, dV$ is the L^2-norm (squared). When $\tilde{B}$ satisfies a Gårding inequality, existence of a solution is equivalent to uniqueness of a solution—and we have already proven the latter of these. Therefore, establishing (B.8) will prove that our variational problems are well-posed.

For $\alpha, \beta \in \mathbb{R}$, we have $\tilde{A} = (\alpha - i\beta)A = \tilde{A}_1 + i\tilde{A}_2$, where

$$\tilde{A}_1 = \frac{1}{k_x^4 + (\omega\mu\hat{\sigma})^2} \begin{pmatrix} (\alpha k_x^2\hat{\sigma} + \beta\omega\mu\hat{\sigma}^2)I & i(\alpha k_x^3 + \beta k_x\omega\mu\hat{\sigma})R \\ i(\alpha k_x^3 + \beta k_x\omega\mu\hat{\sigma})R & (\beta k_x^2\omega\mu - \alpha(\omega\mu)^2\hat{\sigma})I \end{pmatrix} ,$$

$$\tilde{A}_2 = \frac{1}{k_x^4 + (\omega\mu\hat{\sigma})^2} \begin{pmatrix} (\alpha\omega\mu\hat{\sigma}^2 - \beta k_x^2\hat{\sigma})I & i(\alpha k_x\omega\mu\hat{\sigma} - \beta k_x^3)R \\ i(\alpha k_x\omega\mu\hat{\sigma} - \beta k_x^3)R & (\alpha k_x^2\omega\mu + \beta(\omega\mu)^2\hat{\sigma})I \end{pmatrix} .$$

It holds that

$$\Re[\tilde{B}(\mathbf{v}, \mathbf{v})] = \int_\Omega \tilde{A}_1\nabla\mathbf{v} : \nabla\overline{\mathbf{v}} + \alpha\hat{\sigma}|v_1|^2 + \beta\omega\mu|v_2|^2 \, dV \,,$$

$$\Im[\tilde{B}(\mathbf{v}, \mathbf{v})] = \int_\Omega \tilde{A}_2\nabla\mathbf{v} : \nabla\overline{\mathbf{v}} + \alpha\omega\mu|v_2|^2 - \beta\hat{\sigma}|v_1|^2 \, dV \,.$$

Referring back to (B.2)-(B.3), we see that the eigenvalues of $\tilde{A}_2$ will be strictly positive (and bounded away from 0) if $a + c > 0$ and $ac - b^2 > 0$. For $\tilde{A}_2$ we have

$$a + c = \frac{1}{k_x^4 + (\omega\mu\hat{\sigma})^2}\left(\alpha\omega\mu(k_x^2 + \hat{\sigma}^2) - \beta(k_x^2 - (\omega\mu)^2)\hat{\sigma}\right) \quad , \quad ac - b^2 = \frac{\beta(\alpha\omega\mu\hat{\sigma} - \beta k_x^2)}{k_x^4 + (\omega\mu\hat{\sigma})^2}$$

Taking $\beta = 1$, for example, we see that, if $\alpha > 0$ is large enough, both of these quantities are strictly positive—larger than some $\hat{m}_1 > 0$. This implies that

$$\Im[\tilde{B}(\mathbf{v}, \mathbf{v})] + \beta\hat{\sigma}_{\max}\|\mathbf{v}\|_0^2 \geq \hat{m}_1|\mathbf{v}|_1^2 \geq \frac{\hat{m}_1}{C}\|\mathbf{v}\|_1^2 \,,$$

where $\hat{\sigma}_{\max}$ is the largest value that $\hat{\sigma}$ attains on Ω and C is the optimal constant in the norm equivalence $|\mathbf{v}|_1 \leq \|\mathbf{v}\|_1 \leq C|\mathbf{v}|_1$. This completes the proof.