Geophysical Journal International

Geophys. J. Int. (2016) **207,** 571–588
Advance Access publication 2016 August 1
GJI Geomagnetism, rock magnetism and palaeomagnetism

doi: 10.1093/gji/ggw290

MARE2DEM: a 2-D inversion code for controlled-source electromagnetic and magnetotelluric data

Kerry Key

Scripps Institution of Oceanography, UC San Diego, La Jolla, CA 92093-0225*, USA. E-mail: kkey@ucsd.edu*

Accepted 2016 July 28. Received 2016 July 27; in original form 2016 January 13

SUMMARY

This work presents MARE2DEM, a freely available code for 2-D anisotropic inversion of magnetotelluric (MT) data and frequency-domain controlled-source electromagnetic (CSEM) data from onshore and offshore surveys. MARE2DEM parametrizes the inverse model using a grid of arbitrarily shaped polygons, where unstructured triangular or quadrilateral grids are typically used due to their ease of construction. Unstructured grids provide significantly more geometric flexibility and parameter efficiency than the structured rectangular grids commonly used by most other inversion codes. Transmitter and receiver components located on topographic slopes can be tilted parallel to the boundary so that the simulated electromagnetic fields accurately reproduce the real survey geometry. The forward solution is implemented with a goal-oriented adaptive finite-element method that automatically generates and refines unstructured triangular element grids that conform to the inversion parameter grid, ensuring accurate responses as the model conductivity changes. This dual-grid approach is significantly more efficient than the conventional use of a single grid for both the forward and inverse meshes since the more detailed finite-element meshes required for accurate responses do not increase the memory requirements of the inverse problem. Forward solutions are computed in parallel with a highly efficient scaling by partitioning the data into smaller independent modeling tasks consisting of subsets of the input frequencies, transmitters and receivers. Non-linear inversion is carried out with a new Occam inversion approach that requires fewer forward calls. Dense matrix operations are optimized for memory and parallel scalability using the ScaLAPACK parallel library. Free parameters can be bounded using a new non-linear transformation that leaves the transformed parameters nearly the same as the original parameters within the bounds, thereby reducing non-linear smoothing effects. Data balancing normalization weights for the joint inversion of two or more data sets encourages the inversion to fit each data type equally well. A synthetic joint inversion of marine CSEM and MT data illustrates the algorithm's performance and parallel scaling on up to 480 processing cores. CSEM inversion of data from the Middle America Trench offshore Nicaragua demonstrates a real world application. The source code and MATLAB interface tools are freely available at http://mare2dem.ucsd.edu.

Key words: Numerical solutions; Inverse theory; Electromagnetic theory; Magnetotellurics; Marine electromagnetics.

1 INTRODUCTION

The industrial investment in marine controlled-source electromagnetic (CSEM) methods for offshore oil and gas exploration over the past 15 yr has driven innovations in data acquisition leading to a remarkable increase in the size of offshore data sets, with large modern surveys consisting of a few hundred kilometres of CSEM source tows and hundreds of EM receiver deployments that record the transmitted electromagnetic fields (e.g. Constable 2010; Myer *et al.* 2012; Pedersen & Hiner 2014). This large amount of data presents a significant computational challenge for inverse modeling (e.g. Commer *et al.* 2008). While some geometrically complex regions may require full 3-D inversion, the faster computational times possible with 2-D inversion make it an important practical tool for CSEM modeling and interpretation of real data, as well as for model feasibility and sensitivity studies.

The inline geometry of a typical CSEM survey is well suited for 2-D modeling, where the EM field generated by the dipole source is primarily measured by receivers deployed along the towline, making the data predominantly sensitive to subsurface conductivity variations in the region between the source and receivers (e.g. Weiss & Constable 2006; Constable 2010). Even when 3-D inversion is

needed due to complicated structures or the desire to model offline receiver data, 2-D inversions computed from data along parallel towlines can be combined to create a useful starting model for subsequent 3-D inversion, ideally leading to faster convergence of the 3-D inversion. While 1-D inversion can be used for even more rapid data analyses, it is prone to significant model biases since 1-D models are unable to reproduce the strong lateral edge effects from 2-D and 3-D structures, particularly when the source and receiver are on opposite sides of a lateral contact (e.g. Orange *et al.* 2009; Sasaki & Meju 2009). In the context of offshore hydrocarbon exploration, inline CSEM responses from thin, deeply buried 2-D reservoirs can often closely match those from 3-D reservoirs whereas the corresponding 1-D model responses are usually significantly different (e.g. Tehrani & Slob 2013). Furthermore, 1-D models cannot reproduce the effects of seafloor topography that are possible with 2-D modeling (Li & Constable 2007).

Regularized non-linear 2-D EM inversion for the magnetotelluric (MT) method and the significantly more computationally expensive CSEM method has been well developed for over the past two decades (e.g. deGroot Hedlin & Constable 1990; Unsworth & Oldenburg 1995; Lu *et al.* 1999; Rodi & Mackie 2001; Abubakar *et al.* 2008), yet it is useful to revisit this topic given recent advances in numerical modeling and parallel computing. In particular, three independent developments present the opportunity for an improved 2-D inversion toolkit.

The first advance is the application of unstructured triangular modeling grids in finite-element forward solvers for both MT and CSEM problems (e.g. Key & Weiss 2006; Franke *et al.* 2007; Li & Key 2007). These works showed that unstructured modeling grids are able to accurately and efficiently discretize complex 2-D geometries including topography, sloping structural boundaries and multiple-scale structures, all of which can be difficult and numerically expensive to incorporate into the structured rectangular modeling grids used by traditional finite-element and finite-difference modeling codes. By using unstructured grids for inversion parameters rather than the typical rectangular grids, these same benefits could be applied to inverse modeling. Unstructured grids have already been shown to be useful for 3-D inversion with complex seafloor topography (Schwarzbach & Haber 2013) and for 3-D forward modeling (Ren *et al.* 2013; Jahandari & Farquharson 2014).

The second advance is automatic adaptive mesh refinement for solving the forward problem (e.g. Key & Weiss 2006; Franke *et al.* 2007; Li & Key 2007; Key & Ovall 2011; Pardo *et al.* 2011). In this approach, the forward problem is solved on iteratively refined finite-element meshes until the solution meets a specified tolerance. Mesh refinement is guided by a goal-oriented error estimator that is used to select which elements should be refined, based on how the error in each element influences the accuracy of the EM responses at the receiver locations. Adaptive mesh refinement frees the user from the burden of having to design an accurate forward modeling grid, a task that can be cumbersome or uncertain for complicated models, or may be beyond a user's experience level.

In the context of inversion, adaptive mesh refinement could be performed during each step of the inversion to ensure the forward responses retain accuracy while the model conductivity parameters change, rather than relying on a single fixed mesh for all steps of the inversion, as is done in many existing codes. Since the adaptive refinement leads to a variable number of elements in the forward mesh, it can be decoupled from the static inversion parameter grid by using a dual-grid approach where the adaptively generated finite-element meshes conform to the parameter grid. Another reason to use a dual-grid approach is so that the tiny finite elements required around the transmitters and receivers for numerical accuracy of the forward responses do not increase the size of the inverse problem. A dual grid with a static finite-element mesh was shown to be beneficial in one of the first 2-D MT inversion codes (deGroot Hedlin & Constable 1990). An advance on this approach would be to include adaptivity, so that the user only needs to define the inverse parameter grid and the code would then automatically generate and refine the finite-element meshes on the fly. An earlier forward modeling code applied a similar approach where the model was defined by arbitrary polygonal regions of constant conductivity and all finite-element mesh generation and adaptive refinement was done on the fly (Key & Ovall 2011).

Finally, all modern computing platforms have parallel processing capabilities, ranging from a few processing cores on laptops, to a dozen or more cores on workstations, to hundreds or thousands of processing cores on commonly available networked cluster computers. Thus, a modern inversion code should leverage parallel computing algorithms for both the forward and inverse components. The EM forward problem is easily separated into many parallel calculations since each transmitter and frequency give rise to independent linear systems, resulting in what is known in the parallel computing community as an *embarrasingly parallel* problem where there is little to no need for intercommunication between parallel processes (e.g. Newman 2013). Since a typical data set may have 10 or more frequencies and hundreds of transmitters, the parallel speedup can be considerable when run on a cluster with as many processors as independent tasks (e.g. Key & Ovall 2011). Previous forward modeling work showed that additional parallel speedups are possible by decomposing the data into small subsets of receivers and transmitters and carrying out adaptive mesh refinement independently on each subset, resulting in smaller finite-element systems and thus faster forward solves for each reduced data subset (Key & Ovall 2011). Similar parallel speedups could be possible when computing the sensitivity terms in the rows of the Jacobian matrix required by inversion methods.

Parallel computing can also help solve problems that require more memory than is available on a single processor (e.g. Newman & Alumbaugh 1997; Newman 2013). For the non-linear Gauss–Newton inversion method considered in this work, the obvious candidates for distributed memory efficiencies include storing the dense matrices for the Jacobian and the approximate Hessian in parallel. Further, the dense matrix factorization required to solve the model update equations could be sped-up using parallel methods.

The aim of this work is to further describe these advances and others that are part of the new inversion code MARE2DEM (Modeling with Adaptively Refined Elements for 2-D EM), which is the successor to an earlier CSEM and MT forward code with the same name (Key & Ovall 2011). Although 2-D MT inversion codes have been available for a few decades, this is the first freely available 2-D code for anisotropic inversion of MT data and for isotropic or anisotropic inversion of CSEM data. The main novel features of MARE2DEM when compared to other 2-D EM inversion codes include the unstructured model parametrization, automatic adaptive mesh refinement, a highly scalable parallel data decomposition and parallel matrix inversion that allow for faster run-times and the solution of larger inverse problems when many processing cores are available, and the new fast Occam inversion method described below. Additional features include joint inversion of MT and CSEM data and a new non-linear transformation for bounding the range of permissible inverted resistivity values. Finally, all of the features described in this manuscript as well as interactive user-interface programs

for model construction and response plotting are freely available in software packages distributed at http://mare2dem.ucsd.edu.

2 TRIAXIALLY ANISOTROPIC FORWARD MODELING

A previous 2-D finite-element solution that includes triaxial anisotropy is given in Kong *et al.* (2008), while Li & Dai (2011) offer a solution for the more complicated case of tilted anisotropy. Here, the triaxially anisotropic formulation is given using the same notation as the isotropic equations in Key & Ovall (2011). This is followed by an overview of the goal-oriented adaptive finite-element solution to the forward problem and a brief presentation of the adjoint method used to compute the field component sensitivities required by the non-linear inversion.

2.1 Problem statement

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

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

$$\nabla \times \mathbf{H} - \bar{\bar{\sigma}}\mathbf{E} = \mathbf{J_s}. \tag{2}$$

The material electromagnetic properties are represented by the magnetic permeability μ and the symmetric second-rank electrical conductivity tensor $\bar{\bar{\sigma}}$, which in this work is restricted to be triaxial:

$$\bar{\bar{\sigma}} = \begin{bmatrix} \sigma_x & 0 & 0 \\ 0 & \sigma_y & 0 \\ 0 & 0 & \sigma_z \end{bmatrix}. \tag{3}$$

Each element of the tensor denotes the complex conductivity $\sigma = \hat{\sigma} - i\omega\epsilon$, where $\hat{\sigma}$ is the real component of electrical conductivity and the imaginary component depends on the dielectric permittivity ϵ. Since $\sigma \gg \omega\epsilon$ for most low-frequency geophysical applications, the imaginary term can usually be safely neglected in what is known as the quasi-static approximation (e.g. Ward & Hohmann 1987). The 3-D equations above are transferred into 2-D by Fourier transformation with respect to x:

$$\hat{\mathbf{F}}(k_x, y, z) = \int_{-\infty}^{\infty} \mathbf{F}(x, y, z) e^{-ik_x x} dx, \tag{4}$$

where k_x is the spatial wavenumber in the strike direction x and a hat ($\hat{}$) denotes the quantity in the wavenumber domain (k_x, y, z). After Fourier transformation and some tedious manipulations, the coupled differential equations for the strike parallel electromagnetic field components $\hat{E}_x$ and $\hat{H}_x$ can be found to be simply

$$-\nabla \cdot (A\nabla\mathbf{u}) + C\mathbf{u} = \mathbf{f} \quad \text{in } \Omega \qquad \mathbf{u} = \mathbf{0} \quad \text{on } \partial\Omega, \tag{5}$$

where Ω denotes the model domain with outer boundary $\partial\Omega$ and $\mathbf{u} = (\hat{E}_x, \hat{H}_x)$. The gradient and divergence are interpreted componentwise:

$$\nabla\mathbf{u} = (\nabla\hat{E}_x, \nabla\hat{H}_x),$$
$$\nabla \cdot A\nabla\mathbf{u} = \nabla \cdot \mathbf{w} = (\nabla \cdot \mathbf{w}_1, \nabla \cdot \mathbf{w}_2).$$

The coefficient matrices A and C are nearly the same as those given in Key & Ovall (2011), but here have been extended for triaxial anisotropy:

$$A = \begin{pmatrix} \lambda\sigma_t & ik_x\lambda R \\ ik_x R\lambda & i\omega\mu\lambda' \end{pmatrix}, \qquad C = \begin{pmatrix} \sigma_x & 0 \\ 0 & i\omega\mu \end{pmatrix},$$

where

$$R = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \qquad \sigma_t = \begin{pmatrix} \sigma_y & 0 \\ 0 & \sigma_z \end{pmatrix},$$

$$\lambda^{-1} = \begin{pmatrix} k_x^2 - i\omega\mu\sigma_y & 0 \\ 0 & k_x^2 - i\omega\mu\sigma_z \end{pmatrix}, \qquad \lambda' = R^T\lambda R. \tag{6}$$

The off-diagonal terms in matrix A couple the $\hat{E}_x$ and $\hat{H}_x$ fields for CSEM sources; however, for MT sources $k_x = 0$ since the 2-D fields are uniform along the strike direction x and hence (5) uncouples into two independent equations for the traverse electric and magnetic modes. The source term $\mathbf{f}$ is

$$\mathbf{f} = \nabla \cdot (AQ^T\mathbf{s}_t) - \mathbf{s}_x \tag{7}$$

where

$$Q = \begin{pmatrix} 0 & R \\ R & 0 \end{pmatrix}, \quad \mathbf{s}_t = (\hat{\mathbf{J}}_t^s, \hat{\mathbf{M}}_t^s), \qquad \mathbf{s}_x = (\hat{J}_x^s, \hat{M}_x^s). \tag{8}$$

The numerical solution of (5) provides the strike parallel fields $\hat{E}_x$ and $\hat{H}_x$. The transverse field vectors in the (y, z) plane ($\hat{\mathbf{E}}_t$ and $\hat{\mathbf{H}}_t$) can then be found with a post-processing step using the relation

$$\mathbf{u}_t = QA\nabla\mathbf{u} + Q^TAQ\,\mathbf{s}_t \tag{9}$$

where

$$\mathbf{u}_t = (\hat{\mathbf{E}}_t, \hat{\mathbf{H}}_t). \tag{10}$$

Finally, the 3-D spatial domain fields at position (x, y, z) are obtained with the inverse Fourier transform:

$$\mathbf{F}(x, y, z) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{\mathbf{F}}(k_x, y, z) e^{ik_x x} dk_x. \tag{11}$$

2.2 Solution using goal-oriented adaptive finite elements

By restating the strong form of the differential equations in (5) in their weak form, the coupled variational problem is: find $\mathbf{u} \in \mathcal{H} = [H_0^1(\Omega)]^2$ such that

$$\underbrace{\int_\Omega (A\nabla\mathbf{u}) : \nabla\bar{\mathbf{v}} + (C\mathbf{u}) \cdot \bar{\mathbf{v}}\, d\Omega}_{B(\mathbf{u},\mathbf{v})} = \underbrace{\int_\Omega \mathbf{f} \cdot \bar{\mathbf{v}}\, d\Omega}_{F(\mathbf{v})} \text{ for all } \mathbf{v} \in \mathcal{H}, \tag{12}$$

where the double-dot notation describes the relation

$$(A\nabla\mathbf{u}) : \nabla\bar{\mathbf{v}} = \mathbf{w} : \nabla\bar{\mathbf{v}} = \mathbf{w}_1 \cdot \nabla\bar{v}_1 + \mathbf{w}_2 \cdot \nabla\bar{v}_2, \tag{13}$$

$\mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2)$ and the $\bar{v}$ and $\bar{\mathbf{v}}$ notations denote complex conjugation. $H_0^1(\Omega)$ denotes the space of complex functions whose first-order weak derivatives are square integrable and which vanish on the boundary. The discrete version of (12) is:

$$B(\mathbf{u}_n, \mathbf{v}) = F(\mathbf{v}) \text{ for all } \mathbf{v} \in V_n, \tag{14}$$

where V_n is the space of piecewise linear functions used for the finite-element basis and $\mathbf{u}_n$ is a vector of the strike parallel electric and magnetic field coefficients at the element vertices. Further details of the discrete finite-element system are given in Appendix A.

Obtaining an accurate EM solution from the finite-element method depends on using a properly designed mesh. While in principle, the user of a finite-element code could *a priori* design a suitable mesh based on extensive prior experience and rules-of-thumb, a more reliable and convenient approach is to let the physics of the problem automatically guide an iterative adaptive mesh refinement scheme until the mesh produces an EM field solution with the desired numerical accuracy. This could be helpful for non-linear inversion where the conductivity within the model domain, and thus the coupling of the EM sources to structure and the spatial gradients of the EM fields, can change considerably during the inversion iterations.

The basic recipe for adaptive finite-element methods is to first compute the solution to the finite-element problem using a coarse triangulation mesh. Some post-processing or other method is used to estimate the error in the finite-element solution for each mesh element. The elements with the largest estimated error are marked for refinement, and the grid is subsequently refined in these regions. This sequence is repeated until the solution is estimated to be accurate to within a specified tolerance.

A variety of error estimation schemes have been proposed, but goal-oriented error estimators have received considerable attention since they seek to minimize the error in some linear functional of the finite-element solution that is related to a practical quantity of interest, rather than minimizing a global energy norm (Prudhomme & Oden 1999). For example, the goal for EM geophysics forward modeling is to produce accurate responses at the discrete locations of the receivers, and therefore the goal functional can be specified to be a functional form of the EM response. MARE2DEM uses a goal-oriented error estimator following the hierarchical basis method presented in Ovall (2006) and later applied to EM geophysics in Key & Ovall (2011). The error in the finite-element solution is approximated with $\mathbf{u} - \mathbf{u}_n \approx \boldsymbol{\varepsilon}_n \in W_n$, where W_n is an auxiliary space of quadratic *edge-bump* functions that vanish at every vertex but are continuous across the edges of the mesh. Together the spaces W_n and V_n form a hierarchical basis since the space of globally continuous quadratic functions $X_n = V_n \oplus W_n$. Additional details of W_n and V_n are given in Appendix A. Since CSEM responses can span several orders of magnitude due to the rapid decay of the EM fields, MARE2DEM uses a goal functional that approximates the relative error in the electric and magnetic field responses at the discrete locations of the EM receivers. Thus, the adaptive mesh refinement iterations seek to minimize the relative error in the EM responses. Further details are given in Appendix B.

2.3 Sensitivity calculations

The non-linear inversion method presented in the next section is guided by the partial derivatives of the electric and magnetic fields at each receiver with respect to the conductivity parameters of the inverse model; these quantities are usually referred to as the field sensitivities. While in principle these can be obtained directly by taking the partial derivative of (5) with respect to each parameter and then solving the corresponding finite-element system, this approach becomes inefficient when the number of model parameters n exceeds the total number of receiver components. A more efficient approach exploits the adjoint reciprocity of the problem (e.g. McGillivray *et al.* 1994, and references therein). If there are p total measured electric and magnetic field components, the adjoint reciprocity method requires only p additional solutions to the original finite-element system using p new right-hand side loading vectors ($\mathbf{f}$ in eq. A1), resulting in a large savings compared to the direct sensitivity solution when $n \gg p$. MARE2DEM exploits the direct matrix factorization used for the finite-element system by reusing the factorization when solving for the p adjoint sources, resulting in a significant time savings compared to using an iterative solver for both linear systems.

A concise derivation of the adjoint reciprocity formula for computing EM sensitivities is provided in McGillivray *et al.* (1994) and its application to various 2-D EM problems was previously presented in Farquharson & Oldenburg (1996) and Unsworth & Oldenburg (1995). Here, the formulation is extended for triaxial anisotropy. The sensitivity of a given electric or magnetic field component F to the anisotropic conductivity component σ_j can be found with the Fourier transform

$$\frac{\partial F}{\partial \sigma_j}(x, y, z) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{S}_j(k_x, y, z) e^{ik_x(x_r - x_s)} dk_x, \tag{15}$$

where the Fourier kernel is defined as

$$\hat{S}_j(k_x, y, z) = \int_{A_j} \hat{\mathbf{E}}^a(-k_x, y, z) \cdot \left(\frac{\partial \bar{\bar{\sigma}}}{\partial \sigma_j} \hat{\mathbf{E}}(k_x, y, z) \right) dA_j. \tag{16}$$

$\hat{\mathbf{E}}$ is the wavenumber domain electric field generated by a source at the true source location, whereas $\hat{\mathbf{E}}^a$ is an adjoint electric field generated by turning the particular receiver component F into an adjoint source. A_j refers to the cross-sectional area of the cell containing parameter σ_j and x_r and x_s are the along-strike positions of the receiver and true sources. The tensor derivative $\frac{\partial \bar{\bar{\sigma}}}{\partial \sigma_j}$ is zero except where it equals unity on the diagonal component (x, y or z) corresponding to the anisotropic orientation of $\partial \sigma_j$ and thereby acts to extract the corresponding parallel components of the electric fields $\hat{\mathbf{E}}^a$ and $\hat{\mathbf{E}}$. Thus, for anisotropic models, the sensitivity to a given conductivity direction only depends on the component of the electric field (and the associated electric current) parallel to that direction. Conversely, for isotropic models, $\frac{\partial \bar{\bar{\sigma}}}{\partial \sigma_j} = 1$ and the sensitivity depends on all three components of the electric field. In summary, eq. (16) shows that the sensitivity of a given field component to all model parameters can be found with only a single additional finite-element solution for the adjoint field $\hat{\mathbf{E}}^a$. For efficiency, MARE2DEM computes the adjoint solutions using the same adaptively refined mesh that is generated by the forward solution for $\hat{\mathbf{E}}$.

3 INVERSION FRAMEWORK

3.1 Fast Occam inversion

Regularized EM inversion (e.g. Parker 1994; Zhdanov 2002) typically seeks to minimize an unconstrained functional of the form

$$U = \|\mathbf{Rm}\|^2 + \|\mathbf{P}(\mathbf{m} - \mathbf{m}_*)\|^2 + \mu^{-1} \|\mathbf{W}(\mathbf{d} - \mathcal{F}(\mathbf{m}))\|^2 \tag{17}$$

where $\mathbf{m}$ is the n dimensional vector of model parameters with units $\log_{10}$(resistivity), the first term on the right-hand side measures the model's roughness, the second term measures the model's deviation from a prejudice $\mathbf{m}_*$ and the third term is the fit of model's forward response $\mathcal{F}(\mathbf{m})$ to the observed data vector $\mathbf{d}$ weighted by the data's inverse standard errors in the diagonal matrix $\mathbf{W}$. The prejudice term is optional, and its diagonal weighting matrix $\mathbf{P}$ is usually set to zero everywhere; if a user desires to bias a particular subset of the model parameters, the corresponding diagonal elements can be set to some positive weighting value and the desired resistivity can be inserted into $\mathbf{m}_*$. The Lagrange multiplier μ is used to balance the trade-off between the data fit and the model roughness and model prejudice, thereby stabilizing an otherwise ill-conditioned inverse

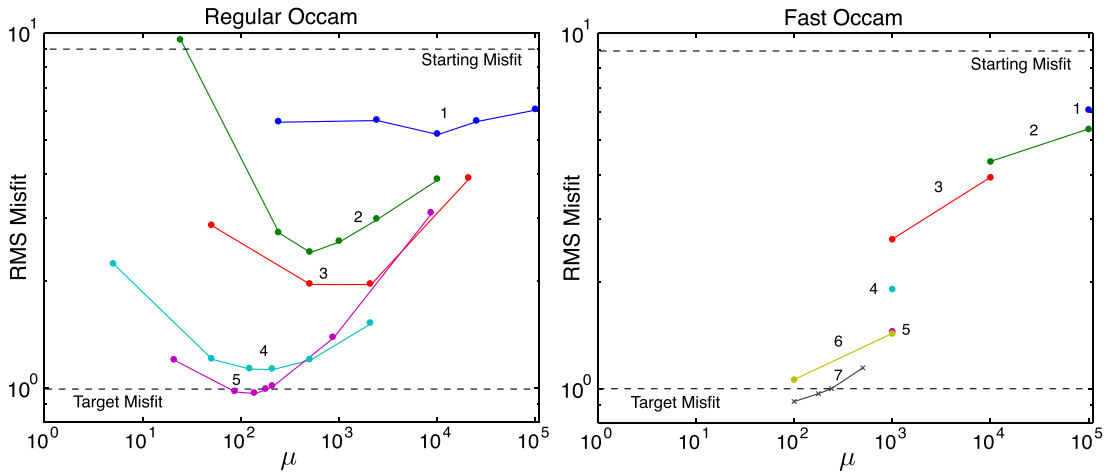

Figure 1. Convergence of the regular Occam and fast Occam algorithms. Coloured dots show the misfit for each forward call during the line search minimization of each Occam iteration. Numbers next to the coloured symbols and lines denote the iteration number. The fast Occam approach found an acceptable model at rms 1.0 using much fewer forward calls.

problem. The precise form of the roughness operator matrix $\mathbf{R}$ is discussed in a later section.

The Occam approach (Constable *et al.* 1987) is a variant of non-linear Gauss–Newton minimization that is widely known for its geophysical applications; it linearizes the functional about a starting model $\mathbf{m}_k$ and accomplishes the minimization iteratively using:

$$\mathbf{m}_{k+1} = \left[\mu\left(\mathbf{R}^T\mathbf{R} + \mathbf{P}^T\mathbf{P}\right) + (\mathbf{W}\mathbf{J}_k)^T\,\mathbf{W}\mathbf{J}_k\right]^{-1} \times \left[(\mathbf{W}\mathbf{J}_k)^T\,\mathbf{W}\hat{\mathbf{d}} + \mu\mathbf{P}^T\mathbf{P}\mathbf{m}_*\right] \tag{18}$$

where the modified data vector

$$\hat{\mathbf{d}} = \mathbf{d} - \mathcal{F}(\mathbf{m}_k) + \mathbf{J}_k\mathbf{m}_k. \tag{19}$$

The model Jacobian matrix $\mathbf{J}$ has entries

$$J_{ij} = \frac{\partial \mathcal{F}_i(\mathbf{m}_k)}{\partial m_j} = \frac{\partial \mathcal{F}_i(\mathbf{m}_k)}{\partial \log_{10}\rho_j} = -\frac{\ln(10)}{\rho_j}\frac{\partial \mathcal{F}_i(\mathbf{m}_k)}{\partial \sigma_j} \tag{20}$$

where sensitivities $\partial\mathcal{F}_i(\mathbf{m}_k)/\partial\sigma_j$ are efficiently computed using the method discussed in the previous section.

Phase 1 of the Occam algorithm consists of a line search over eq. (18) that seeks the value of μ that produces $\mathbf{m}_{k+1}$ with the lowest data misfit, which at least initially will be larger than the target misfit. In the original Occam implementation, this line search is carried out efficiently by first bracketing a minimum using a golden-section search, and then finding the minimum using Brent's method (Press *et al.* 2003). An illustrative example is shown on the left-hand side of Fig. 1. Each iteration starts off with a Jacobian calculation $\mathbf{J}(\mathbf{m}_k)$ in conjunction with a forward call $\mathcal{F}(\mathbf{m}_k)$ to determine the misfit of the starting model, then the line search is carried out. In the best case scenario where the minimum is bracketed and found immediately, the algorithm requires four forward model evaluations. In this particular example, up to six forward calls were needed for some iterations, and in other cases it is not uncommon to see up to 9 or 10 forward calls during an Occam iteration. While this dynamic search over μ gives Occam a robustness for finding better fitting models, the number of forward calls has been considered costly and has led to other inversion algorithms using a fixed μ value, which is only modified when a better fitting model cannot be found (see e.g. the discussions in Siripunvaraporn 2012; Egbert & Kelbert 2012).

Inspection of Fig. 1 shows that the minimum search during the first three iterations results in model misfits which, although at the local minimum, are still significantly larger than the target misfit. This motivates a minor change to the algorithm that makes it a hybrid that balances the speed of fixed μ algorithms and the robustness of the Occam's dynamically determined μ. The change, which here is referred to as the fast Occam algorithm, is as follows. If at any given step of the minimum search a model is found with a sufficient misfit reduction compared to the staring model's misfit, it is accepted and the iteration is terminated; otherwise the complete minimum search is carried out as normally done for the regular Occam algorithm. Through trial and error, a root-mean-squared (rms) misfit reduction threshold of about 15 per cent has been found to work well for typical MT and CSEM inversions. A comparison example is shown on the right-hand side of Fig. 1. Here, it took seven Occam iterations to reach the target misfit, but each iteration only required one or two forward calls, resulting in a significant time savings compared to the regular Occam approach.

For both the regular Occam and fast Occam approaches, if the minimum search fails to find a model with a lower misfit, a reduced model step is taken and the line search is carried out again using the model $\mathbf{m}'_{k+1}$:

$$\mathbf{m}'_{k+1} = \alpha\,\mathbf{m}_{k+1} + (1-\alpha)\mathbf{m}_k. \tag{21}$$

Initially, the step size $\alpha = 1$ and α is successively cut in half each time the line search fails to find a better fitting model.

Once a model with a misfit less than or equal to the target misfit has been found, phase 2 of the Occam algorithm begins, where it seeks the model at the target misfit that has the largest μ, and thus the smallest roughness norm. This is accomplished by bracketing μ about the target misfit and iteratively using interpolation to find the largest value of μ at the target misfit.

3.2 Distributed dense matrix operations

A computational difficulty that Occam and other Gauss–Newton style inversions encounter is dealing with the dense matrices in eq. (18) when the model size grows large. For n model parameters, $(\mathbf{W}\mathbf{J})^T\,\mathbf{W}\mathbf{J}$ requires n^2 values to be stored in memory. Furthermore, the dense matrix–matrix multiplication and subsequent matrix factorization for solving the linear system have operation counts that scale as $O(n^3)$. For big models, the time spent on the matrix operations can be orders of magnitude larger than the time spent computing the forward response and the model Jacobian. This has motivated the development of inversions using linear and non-linear

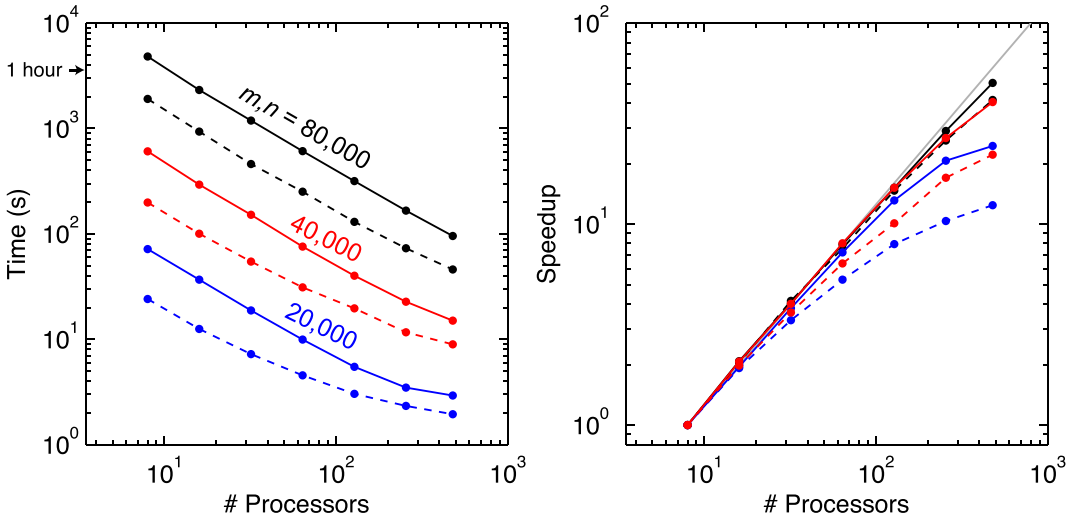

Figure 2. Runtime scaling (left) and relative speedup (right) of the ScaLAPACK routines used for the dense matrix operations in MARE2DEM. Solid lines show the time to compute the dense matrix product $(\mathbf{WJ})^T\,\mathbf{WJ}$ using the routine *PDSYRK*. Dashed lines show the time required to solve the linear system in eq. (18) using the Cholesky factorization routines *PDPOTRF* and *PDPOTRS*. Relative speedup is here defined to be T_8/T_p, where T_8 is the time taken on eight processors and T_p is the time required on p processors.

conjugate gradient methods that only require matrix-vector products between **J** (or its transpose) and a vector, which can be computed on-the-fly and do not require explicit storage of **J** in memory (e.g. Mackie & Madden 1993; Newman & Alumbaugh 2000; Newman & Boggs 2004). While these approaches are necessary for large-scale 3-D inversion due to their frugal use of memory (e.g. Commer & Newman 2008), a disadvantage is that convergence is much slower than Gauss–Newton style inversion, requiring significantly more forward calls (e.g. Siripunvaraporn 2012).

Although the present work only considers 2-D models, the number of parameters for long 2-D profiles can easily grow large enough that the above limitations become burdensome. For example, consider a marine CSEM towline that is 100 km long; if the inversion parameter grid has 25 m thick cells to 2 km beneath the seafloor and 50 m cell widths, the resulting grid will have 160 000 cells. For transversely isotropic conductivity inversion where each cell has two free parameters (i.e. vertical and horizontal conductivity), the inverse problem will have 320 000 total parameters. To support the inversion of such large data sets while still retaining the rapid convergence and robustness of the Occam inversion approach, MARE2DEM carries out the dense matrix operations in distributed form using the ScaLAPACK library (Blackford *et al.* 1997). ScaLAPACK contains a subset of LAPACK routines that have been redesigned for distributed memory parallel computers using the message passing interface (MPI) standard, making it highly portable for present-day computing environments ranging from multicore laptops to small- to large-scale cluster and cloud computing systems.

On a computer or cluster with p processors, a dense $m \times n$ matrix can be stored in distributed form using only mn/p values per processor. For example, if $m = n = 2 \times 10^5$ and $p = 100$ then $mn/p = 4 \times 10^8$. When stored as double precision variables, the 320 GB required to store the matrix on a single processor, which is near the memory capacity of current high-end cluster nodes, reduces to only 3.2 GB per processor in distributed form. For ScaLAPACK, the processors are mapped into a virtual 2-D grid of dimensions $p = r \times c$ where r is the number of virtual processor rows and c is the number of virtual processor columns. The matrix is distributed in 2-D block cyclic form where each processor has a local matrix of size $m/r \times n/c$. The block cyclic distribution has been shown to have significant advantages over simple contiguous block distribution, particularly for load balancing between processors and memory hierarchy within a single processor (Blackford *et al.* 1997).

Fig. 2 shows examples of using the ScaLAPACK routine *PDSYRK* to compute the dense matrix product $(\mathbf{WJ})^T\,\mathbf{WJ}$ and routines *PDPOTRF* and *PDPOTRS* to solve eq. (18) using Cholesky factorization. These computations were run on a medium-scale cluster with dual-socket, 8-core, 2.6GHz Intel Xeon E5-2670 processors connected with a quad data rate InfiniBand network. Test cases are shown for $m = n =$ 20 000, 40 000 and 80 000 when run on 8–480 processors. Consider the 80 000 case where the matrix–matrix multiplication took over 1 hr when carried out on eight processors, yet was reduced to only 95 s when run on 480 processors. A similar reduction is seen for the Cholesky factorization, which runs about 50 per cent faster than the matrix multiplication. To better illustrate the scaling of the algorithm, Fig. 2 also shows the relative speedup as a function of the number of processors; here the relative speedup is defined to be T_8/T_p where T_8 is the time taken on eight processors and T_p is the time required on p processors. An algorithm that scales with perfect efficiency will have a speedup that falls along the grey line where the speedup is $p/8$. As can be seen, both the matrix multiplication and Cholesky routines scale well for the large matrix case (80 000), reaching speedups of about 50 and 40 when run on 480 processors, which is close to the optimum speedup of 60. For lower number of processors ($\ll$100), the 20 000 and 40 000 cases have good speedups, but for larger number of processors the efficiency drops down as the work load per processor becomes insignificant compared to the interprocessor communication demands. However, the overall run-times for these cases are only a few seconds so that the less than ideal efficiency is not a concern.

3.3 Bounds on model parameters

The model update eq. (18) does not place any constraints on the range of values that a parameter can take, yet often there are geological reasons or ancillary data sets that suggest the conductivity will be within a certain range of values. When such inequality constraints are desired, they can be implemented simply by recasting the inverse problem using a non-linear transformation of the model

parameters so that the objective function and optimization algorithm remain essentially the same as the unconstrained problem (Box 1966). The model parameter m is bounded as

$$l < m(x) < u \tag{22}$$

while the transformed parameter x is unbounded

$$-\infty < x(m) < \infty. \tag{23}$$

With the transformed model vector $\mathbf{x}$, the model update is accomplished with a modified version of eq. (18):

$$\mathbf{x}_{k+1} = \left[\mu\left(\mathbf{R}^T\mathbf{R}\right) + \left(\mathbf{W}\mathbf{J}_k^*\right)^T \mathbf{W}\mathbf{J}_k^*\right]^{-1}\left[\left(\mathbf{W}\mathbf{J}_k^*\right)^T \mathbf{W}\hat{\mathbf{d}}\right] \tag{24}$$

where

$$\hat{\mathbf{d}} = \mathbf{d} - \mathcal{F}(\mathbf{m}(\mathbf{x}_k)) + \mathbf{J}_k^*\mathbf{x}_k, \tag{25}$$

$$J_{ij}^* = \frac{\partial m_j}{\partial x_j} J_{ij}, \tag{26}$$

and the model prejudice term has been omitted for brevity. In fact, most EM inversion codes already use such a transform approach by inverting for $\log_{10}\sigma$ or $\log_{10}\rho$ to avoid physically meaningless negative conductivities, as shown by (20). While an infinite number of transformations are possible for more restrictive double-sided inequality bounds, a commonly used one is the exponential transform (Zhdanov 2002; Habashy & Abubakar 2004; Commer & Newman 2008):

$$m = \frac{ue^{\tilde{x}} + l}{e^{\tilde{x}} + 1} \quad \text{where} \quad \tilde{x} = x - (u + l)/2. \tag{27}$$

In the variation shown here, the variable $\tilde{x}$ has been introduced to keep the symmetry about the midpoint of the bounds at $(u + l)/2$, which facilitates comparison with the bandpass transform introduced later. For large positive x, $m(x)$ asymptotes to u, while for large negative x, $m(x)$ asymptotes to l. The sensitivity transform is found by taking the derivative of this expression, yielding:

$$\frac{\partial m}{\partial x} = \frac{(u - l)e^{\tilde{x}}}{(1 + e^{\tilde{x}})^2}. \tag{28}$$

The transformed variable $x(m)$ can be found from (27):

$$x = \log(m - l) - \log(u - m) + (u + l)/2, \quad \text{where} \quad l < m < u. \tag{29}$$

Fig. 3 shows examples of this transformation and the sensitivity scaling for two different sets of bounds. For $l = -1 < m < u = 5$, the bound parameters and the transform variable x are fairly close to the original parameters. However, when the bounds are tighter, $l = -1 < m < u = 1$, the transformed parameters are quite different from the true parameters. This difference can also be seen in the sensitivity scaling curves shown in Fig. 3. Functionally, this difference does not affect the ability of the transform to bound the model parameters, and indeed the exponential transform has been shown to be useful in practice (e.g. Commer & Newman 2008). However, note that the roughness operator $\mathbf{R}$ in eq. (24) is operating on the transformed parameter and not the original parameter; this could have unintended consequences, for example, where the transformation significantly increases or decreases the spatial gradient in the transformed model, effectively resulting in a non-linear roughness operator that underpenalizes regions where the transform reduces the model gradient and overpenalizes regions where the gradient is increased.

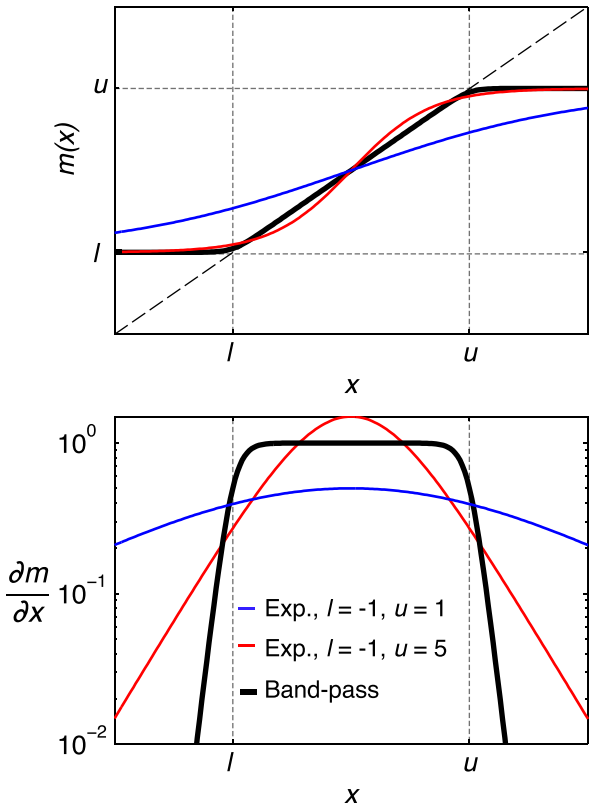

Figure 3. Non-linear transforms used to bound model parameters during inversion. The bound model parameter function $m(x)$ (top) and the sensitivity scaling $\frac{\partial m}{\partial x}$ (bottom) are shown for the exponential transform (red and blue) and the newly proposed bandpass transform (black).

This work proposes a new transformation that does not suffer from this shortcoming. Consider a flat sensitivity scaling with unit amplitude in the passband between b and a; this is ideal since it can be used to make the transformed parameters nearly identical to the original parameters within the range of the bounds. This suggests that a bandpass filter response equation is a good candidate for the transform's sensitivity scaling:

$$\frac{\partial m}{\partial x} = \frac{1 - e^{c(l-u)}}{(1 + e^{-c(x-l)})(1 + e^{c(x-u)})}, \tag{30}$$

where c is a constant that controls the decay of the scaling past the bounds. By setting c to be a function of the extent of the bounds, the shape of the transform shape can be made to be independent of the specific bounds. $c = 15/(u - l)$ has been found to work well in practice. Integrating this equation yields the expression for the bound model parameters:

$$m(x) = \frac{1}{c}\log\left(\frac{1 + e^{c(l-x)}}{1 + e^{c(u-x)}}\right) + u, \tag{31}$$

and solving for x yields

$$x(m) = \frac{1}{c}\log\left(\frac{e^{c(m-l)} - 1}{1 - e^{c(m-u)}}\right) + l. \tag{32}$$

Fig. 3 shows an example of this transform, which is here referred to as the bandpass transform. Between the bounds, the transformed parameters are identical to the original parameters, as desired, while the sensitivity scaling is flat between the passband with steep drop-offs beyond the bounds.

Both the exponential and bandpass transforms have been implemented in MARE2DEM so that each model parameter can have

its own unique bounds specified. For some data sets, the user will have *a priori* knowledge that can guide the use of a narrow range of parameter bounds in certain localized regions, for example, where nearby well logs provided independent constraints on conductivity. Narrow bounds could also be prescribed to test hypotheses about the range of permissible resistivity values that fit a given data set. Yet in most cases the inversion will be run without any bounds. However, experience has shown there to be a benefit from applying global bounds on all model parameters so that extreme values are excluded from the inversion. In particular, if the line search jumps to very low μ values, the Occam model update can produce rough models that have unrealistically high and low resistivity that usually produce poor misfit. Bounding all parameters to a plausible range (such as 0.1–100 000 ohm m for marine models) alleviates this situation. Another consequence is that unrealistically conductive regions can cause the adaptive mesh refinement scheme in the forward code to spend excessive effort refining the mesh to produce accurate responses in these regions; bounding all the model parameters to a plausible range avoids this computational time-wasting situation.

3.4 Roughness norm for an unstructured grid

The model roughness operator **R** stabilizes the inversion by providing a measure of the model variations so that its minimization will steer the inversion away from producing spurious structures. While a variety of norms are possible, the L_2 norm of the model gradient is a popular choice (e.g. Parker 1994; Zhdanov 2002):

$$\|R(m)\|^2 = \int_\Omega \nabla m \cdot \nabla m \; d\Omega. \tag{33}$$

Here, the gradient dot product is integrated over the model domain. On a structured grid, it is straightforward to form the discrete approximation of this integral using distance weighted first differences between neighbouring parameters in the vertical and horizontal directions. For unstructured grids, the situation is less clear. Schwarzbach & Haber (2013) proposed a primal-dual mixed finite-element formulation for discretizing this integral on unstructured tetrahedral meshes but this solution is cumbersome to implement and furthermore is limited to tetrahedral grids (and by extension triangular grids in 2-D).

Here, the following weighted sum of squares approach has been found to work well for both triangular and quadrilateral parameter grids:

$$\|\mathbf{Rm}\|^2 = \sum_{i=1}^{m} A_i \left[\sum_{j=1}^{N(i)} w_j \left(\frac{\Delta m_{ij}}{\Delta r_{ij}} \right)^2 \right] \tag{34}$$

where

$$\Delta m_{ij} = m_i - m_j, \tag{35}$$

$$\Delta r_{ij} = \sqrt{(y_i - y_j)^2 + (z_i - z_j)^2}, \tag{36}$$

$$w_j = \frac{A_j}{\sum_{k=1}^{N(i)} A_k}, \tag{37}$$

A_i is the area of parameter i and accounts for the integration over the parameter region and $N(i)$ is the set of all parameters sharing a vertex with parameter i. The term in brackets in eq. (34) approximates the two-norm of the gradient at parameter i using an area weighted average of differences between all parameters in a ring surrounding parameter i (see Fig. 4). The distance between parameters Δr_{ij} is

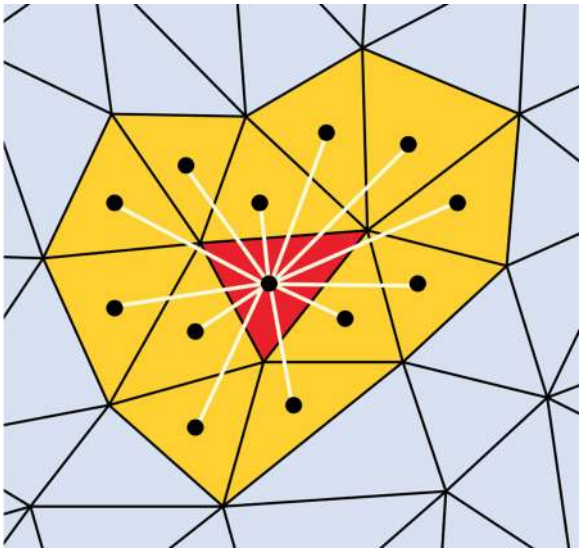

Figure 4. The roughness penalty for the central parameter (red triangle) is computed using a weighted sum of first differences with the ring of surrounding parameters (yellow triangles) according to eq. (34).

computed using the parameter centroids, as shown in Fig. 4. This form is easily implemented using sparse arrays so that computing $\mathbf{R}^T\mathbf{R}$ consumes an insignificant amount of time. A variant of the formulation above was also tested where $N(i)$ was reduced to the set of all parameters sharing an edge with parameter i but this produced rougher looking models.

A slight modification of the distance measure can be used to preferentially bias the roughness operator so that model smoothness is enhanced in either the horizontal or vertical directions:

$$\Delta r_{ij} = \sqrt{\left(\frac{y_i - y_j}{w_{hv}} \right)^2 + (z_i - z_j)^2}. \tag{38}$$

When the horizontal to vertical penalty weight $w_{hv} > 1$, the range is reduced in the horizontal direction, resulting in a larger horizontal gradient and corresponding horizontal roughness penalty; this biases the inversion towards enhanced horizontal smoothness. Conversely, when $w_{hv} < 1$, the horizontal distance is expanded, resulting in less horizontal smoothing and enhanced vertical smoothing.

An implicit assumption in the roughness formulation (34) is that the parameters are fairly small compared to the structures of interest so that they can be considered pixels in an image where the local weighted sum is a good approximation to the gradient. For inversion grids that have only a few large blocky parameters, the gradient approximation can breakdown; a better roughness penalty in this case is to use an unweighted norm of the first differences Δm_{ij} between all neighbouring parameters.

For triaxially anisotropic models, the roughness can be augmented to be

$$\|\mathbf{Rm}\|^2 \equiv \|\mathbf{Rm}_x\|^2 + \|\mathbf{Rm}_y\|^2 + \|\mathbf{Rm}_z\|^2 + \alpha\|\mathbf{m} - \mathbf{m}'\|^2 \tag{39}$$

where

$$\mathbf{m} = \begin{bmatrix} \mathbf{m}_x \\ \mathbf{m}_y \\ \mathbf{m}_z \end{bmatrix}, \qquad \mathbf{m}' = \begin{bmatrix} \mathbf{m}_y \\ \mathbf{m}_z \\ \mathbf{m}_x \end{bmatrix}. \tag{40}$$

The first three terms on the right side of eq. (39) measure the spatial roughness of each anisotropic component while the last term is a measures of the anisotropy. The anisotropy penalty weight α controls the size of the penalty against anisotropy and can be dialed

up or down at the user's discretion. Transversely isotopic models containing only two anisotropic components use a similar penalty formulation that omits the third component.

3.5 Joint inversion normalization weights

Joint inversion of MT and CSEM data offers the possibility of improved structural imaging for offshore exploration (e.g. Commer & Newman 2009; Wiik *et al.* 2013). The goal is to harness the resolving power of each data set to form a new image that is ideally better than the union of the independent inversion models. However, an issue for joint inversion is the ability to fit each data subset equally well when there is a large difference in the number of data for each subset. For example, consider the n dimensional data vector after partitioning it into MT and CSEM subsets with dimensions n_1 and n_2 according to

$$\mathbf{d} = \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{bmatrix}. \tag{41}$$

The misfit functional in (17) can then be expanded as

$$\begin{aligned} \|\mathbf{W}(\mathbf{d} - \mathcal{F}(\mathbf{m}))\|^2 &= \|\mathbf{W}_1(\mathbf{d}_1 - \mathcal{F}_1(\mathbf{m}))\|^2 + \|\mathbf{W}_2(\mathbf{d}_2 - \mathcal{F}_2(\mathbf{m}))\|^2 \\ &= \chi^2 = \chi_1^2 + \chi_2^2 \end{aligned} \tag{42}$$

Assuming the data uncertainties in $\mathbf{W}$ are well known, a good fitting model is characterized by $\chi^2 \approx n$ (Parker 1994). Ideally, joint inversion should produce a model that fits both data subsets equally well, so that $\chi_1^2 \approx n_1$ and $\chi_2^2 \approx n_2$. However, consider the case where one data set is significantly larger (i.e. $n_1 \gg n_2$); a model that fits both data sets well will have $\chi^2 \approx \chi_1^2 \gg \chi_2^2$. This implies that the fit to $\mathbf{d}_2$ can be negligible to the overall data fit, so long as the fit to $\mathbf{d}_2$ satisfies $\chi_2^2 \ll \chi_1^2$. In this situation, the joint inversion has no benefit over the solo inversion of $\mathbf{d}_1$. More precisely, this problem can be stated by defining the data density ratio $p = n_1/n_2$, where $p \gg 1$, and a misfit multiplier q that quantifies how much worse $\mathbf{d}_2$ is fit than its expected value of n_2. A model that fits the joint data set could satisfy the following relation:

$$\chi^2 = \chi_1^2 + \chi_2^2 = n_1 + qn_2 = pn_2 + qn_2. \tag{43}$$

Thus, when $p \gg q$ the fit to $\mathbf{d}_2$ can have negligible impact on the overall misfit. For example, when $p = 100$ and $q = 5$, the misfit for $\mathbf{d}_2$ can be five times worse than d_1 yet the overall χ^2 misfit is only increased by 5 per cent from the extremely poor fit to $\mathbf{d}_2$.

To alleviate this situation, the following normalized joint misfit functional is proposed

$$\|\alpha_1\mathbf{W}_1(\mathbf{d}_1 - \mathcal{F}_1(\mathbf{m}))\|^2 + \|\alpha_2\mathbf{W}_2(\mathbf{d}_2 - \mathcal{F}_2(\mathbf{m}))\|^2 = \frac{\chi_1^2}{n_1} + \frac{\chi_2^2}{n_2}, \tag{44}$$

where

$$\alpha_i = \sqrt{1/n_i} \tag{45}$$

is a data balancing weight that normalized each misfit functional so that smaller data subsets will have as much influence as larger subsets on the overall misfit budget. A model that fits both subsets well will then have

$$\frac{\chi_1^2}{n_1} + \frac{\chi_2^2}{n_2} \approx 1 + 1. \tag{46}$$

This approach is essentially the same as the weighting formula that Commer & Newman (2009) showed to be effective for joint MT and CSEM inversion. In that work, the weighting is only applied to $\mathbf{d}_2$ by specifying $\alpha_1 = 1$ and $\alpha_2 = \sqrt{n_1/n_2}$. The form presented here can be further generalized to N subsets with the data vector partitioned as

$$\mathbf{d} = \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ \cdots \\ \mathbf{d}_N \end{bmatrix} \tag{47}$$

and with the weighted joint misfit functional

$$\|\mathbf{W}(\mathbf{d} - \mathcal{F}(\mathbf{m}))\|^2 = \sum_{i=1}^{N} \|\alpha_i\mathbf{W}_i(\mathbf{d}_i - \mathcal{F}_i(\mathbf{m}))\|^2. \tag{48}$$

For example, this could be useful for joint inversion with three data subsets consisting of marine MT data, seafloor receiver CSEM data and towed-receiver CSEM data where the three subsets contains significantly different numbers of data. This approach may also be useful for joint inversion of MT and seismic data where the seismic data volume greatly exceeds the MT volume (e.g. Bennington *et al.* 2015). Another possible application is the solo inversion of CSEM data that has a large frequency bandwidth, where typically the higher frequencies have much fewer total number of data due to the more rapid attenuation of the high-frequency fields; the data balancing weights proposed here could be used to weight the data misfit functional so that the inversion is encouraged to fit each frequency equally well.

4 MARE2DEM INVERSION CODE

The techniques described above have been implemented in MARE2DEM, extending its functionality to include anisotropic inversion for 2-D CSEM and MT data. The source code is written in modern Fortran with parallel constructs implemented using the MPI standard. Parallel forward computations are carried out using the data decomposition scheme presented in Key & Ovall (2011), where the input transmitter and receiver arrays are sliced into smaller subsets that are each modeled in parallel using the goal-oriented adaptive finite element; this results in significantly faster solutions than possible when the adaptive method is applied to all transmitters and receivers on a single mesh. Additional parallelization is implemented across the requested data frequencies. A manager–worker framework controls the parallel forward computations, where the manager processor maintains a queue of all possible modeling tasks and assigns the next available task to the next available worker processor. Since the adaptive mesh refinement run-times can vary from task to task depending on the particular frequency, transmitter and receiver subsets, this dynamic task-queue-based approach offers much better load balancing among the processors than if the tasks were instead preassigned to each processor.

MARE2DEM is designed with the flexibility to meet a variety of 2-D EM modeling needs. Both electric and magnetic dipole sources have been implemented in addition to MT plane-wave sources. The receivers and transmitters can be located anywhere in the ground, sea or seafloor, allowing for the simulation of various survey geometries or investigating the physical behaviour of EM fields beneath the subsurface. The conductivity can be isotropic, transversely isotropic along one of the three axes or triaxially anisotropic, and each free parameter can be given its own unique bounds and prejudice values, if desired. In addition to the model roughness operator described above, the code can optionally use the minimum-gradient-support roughness norm (Portniaguine & Zhdanov 1999) when sharper

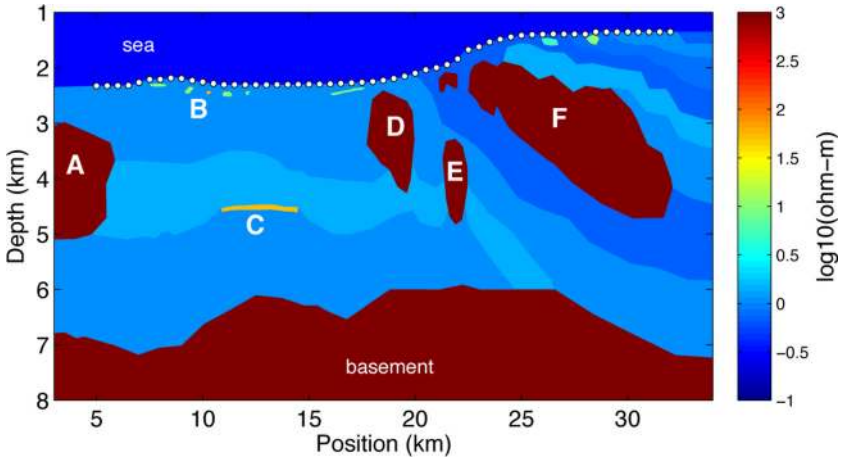

Figure 5. Offshore continental margin forward model used to demonstrate the performance of MARE2DEM for inversion. Labeled features include resistive salt bodies (A, D, E and F), shallow resistive gas hydrate accumulations (B and elsewhere along the seafloor) and a thin resistive hydrocarbon layer about 2.5 km beneath the seafloor (C). White dots show the locations of the 55 seafloor EM receivers. For the forward calculations, the polygonal model shown here is automatically meshed using unstructured triangular finite-element grids that exactly conform to the input node and line segments of the model.

structural boundaries are desired, yet this norm is more prone to getting stuck in local minima and should be applied with care.

An optional MATLAB user interface allows users to easily create forward models with complex surfaces and other polygonal features, and also has tools for rapidly creating pixelated inversion grids composed of either triangular or quadrilateral cells. The use of quadrilateral cells can be advantageous when the station spacing is much wider than the depth of investigation and hence the quadrilaterals can be made to be thin and very wide, offering a fine depth scale while limiting the number of free parameters between laterally adjacent stations. The MATLAB interface also allows users to decrease or entirely remove the inversion's roughness penalty along any of the input surfaces or parameter boundaries, which allows the inversion to recover sharp resistivity jumps (see e.g. Hoversten *et al.* 2015). A MATLAB tool for plotting the output inversion models allows seismic reflection images and well logs to be overlain on the resistivity. Other MATLAB tools allow for exploring the CSEM and MT data, the model responses and the normalized residuals.

The next two sections present synthetic and real data inversion examples that demonstrate the performance of MARE2DEM for large CSEM and MT problems. Other real data applications of the code include: anisotropic MT inversion to map mantle upwelling at a mid-ocean ridge (Key *et al.* 2013) and melt along the lithosphere–asthenosphere boundary (Naif *et al.* 2013), marine CSEM imaging of the Scarborough gas field (Myer *et al.* 2015), marine MT and CSEM for subbasalt exploration in the North Sea (Hoversten *et al.* 2015) and subsalt exploration in the Gulf of Mexico (Constable *et al.* 2015).

5 SYNTHETIC EXAMPLE: JOINT CSEM–MT INVERSION

Here, the inversion aspects of MARE2DEM are demonstrated on the isotropic forward model shown in Fig. 5. This model has seafloor topography and significant resistivity variations inspired by the salt tectonics dominated sedimentary margin in the northern Gulf of Mexico. There are a few salt bodies (features A, D, E and F) with resistivity 1000 times larger than the surrounding sedimentary strata, resistive gas hydrates directly beneath the seafloor (feature B), a thin resistive hydrocarbon reservoir (feature C) and a resistive low-porosity basement.

CSEM and MT forward responses were generated for 55 receivers positioned every 500 m along the seafloor. The receiver dipoles were tilted to be parallel to the sloping seafloor, with a maximum tilt angle of 20° along the steepest section. MT responses were computed for the TE and TM modes at 22 frequencies logarithmically spaced between 0.0001 and 0.1 Hz. For the CSEM data, 271 horizontal electric dipole transmitters were positioned every 100 m laterally with a 50 m tow height above the seafloor, as typical for real survey data. Inline CSEM electric field responses were computed at four frequencies: 0.1, 0.3, 1.0 and 3.0 Hz, providing a range of skin depths suitable for constraining shallow and deep features. To simulate real data, both the MT and CSEM amplitude and phase responses had 4 per cent random Gaussian noise added and the CSEM data were limited to offsets with amplitudes above 10^{-15} VA^{-1} m^{-2}. Since the forward calculations are more efficient for receivers than transmitters, the CSEM receiver and transmitter dipoles were interchanged via the electromagnetic reciprocity theorem (e.g. Parasnis 1988; Harrington 2001), resulting in 271 CSEM receivers and 55 transmitters. Finally, both the MT apparent resistivity and CSEM amplitude data were converted into their $\log_{10}$ forms since recent work has shown this form to be more robust for inversion and can lead to faster convergence (Wheelock *et al.* 2015). In total, the synthetic data set consisted of 51 180 CSEM data and 4840 MT data.

For inversion, a free parameter grid was designed by extending the seafloor topography profile 500 km laterally on each side to ensure the homogenous Dirichlet boundary conditions for the finite-element method are satisfied. A semi-rectangular region of interest was created beneath the seafloor, with its width slightly wider than the receiver profile and its bottom depth at about 8 km (Fig. 6). This region was then divided vertically into three subregions and each was filled with an unstructured grid of conforming triangular model parameters. The shallowest region immediately beneath the seafloor (region 1 in Fig. 6) was filled with the smallest triangles since this is where the data resolution is best, while the two deeper regions had increasingly larger parameters to reflect the decrease in resolution with distance from the transmitters and receivers. The outer region (labeled 4 in Fig. 6) was filled with triangles that grow increasingly

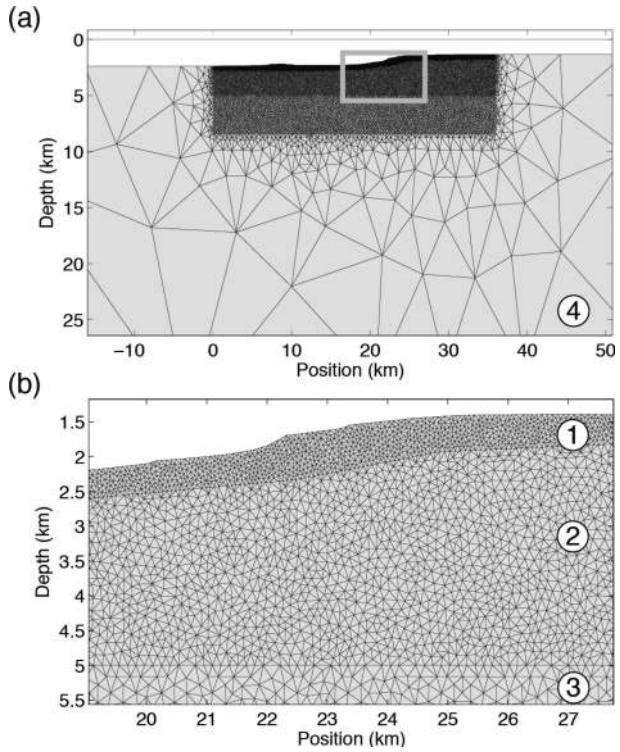

Figure 6. (a) Central portion of the unstructured inversion parameter grid. Panel (b) shows a closeup of the region in the grey rectangle near the seafloor on panel (a). Labels 1–3 show three levels of finer meshing within the region of interest at 0–35 km position and to about 8 km depth. Label 4 shows the outside region where the parameter grid is significantly coarser where resolution in the data is low.

larger with distance from the central region. This resulted in a total of 24 408 free parameters.

A few features of the parameter grid are worth noting. The unstructured triangles allow for a very efficient fine-scale parametrization along the seafloor boundary; this sloping boundary is also modeled at the full resolution of the seafloor topography profile and is not coarsely approximated using the stair-step approach required by inversion codes restricted by fixed rectangular grids. This also allows the receivers to be tilted parallel to the seafloor slope so that they are modeled with the same geometry as the real survey, whereas its not clear what tilt angle is best to use for the receivers when the model grid uses a stair-step approximation for the topography. Using three regions of successively coarser triangular parameters is a simple way to limit the total number of parameters while allowing for detailed resolution where required by the data. An improvement on this approach would be to create an automatic meshing algorithm that allows the triangle size to vary smoothly both in depth and lateral distance from the receivers and transmitters using some heuristic measures, or it could be based on an *a priori* approximation of the data sensitivity and potential model resolution for the expected conductivity. The coarse outer region of large triangles contains less than 4 per cent of the total number of parameters and thus is a very efficient padding of the parameter grid outside the region of interest. The automatic meshing tools included with the MATLAB graphical interface for MARE2DEM allows these types of parameter grids to be designed with only a minute or two of effort; likewise the graphical interface made it easy to draw the sedimentary strata and salt bodies of the original forward model shown in Fig. 5.

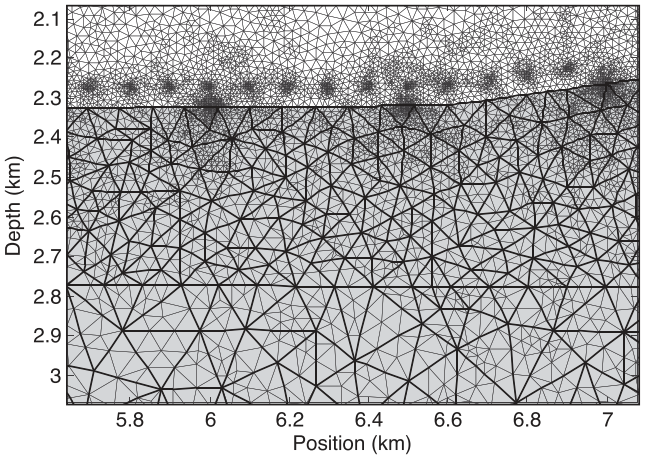

Figure 7. Closeup view showing a portion of the inversion parameter grid (thick lines) and an adaptively refined finite-element mesh (thin lines) created using the goal-oriented error estimator for a particular frequency and subset of the CSEM transmitters and receivers. Note the heavy refinement of the finite-element mesh around the transmitters (spaced every 100 m) and the receivers (spaced every 500 m) along the seafloor.

While users have to create the inversion parameter grid, MARE2DEM automatically generates the conforming finite-element meshes using goal-oriented adaptive refinement during the forward calculations. Each parallel modeling task generates a unique finite-element grid that ensures an accurate forward response for that particular subset of transmitters, receivers and frequency. Fig. 7 shows an example of an adaptively refined finite-element mesh. Typically, the adaptive refinement creates very small elements around the transmitters and receivers while the finite elements are larger further away. These meshes are generated on the fly in memory only and are deallocated from memory after the forward calculation for that subset is completed. During an Occam inversion iteration, new adaptively refined meshes are generated for the particular conductivity model of each forward call.

For this test, the parallel data decomposition divided the input CSEM transmitters and receivers into subsets of 5 consecutive transmitters, 10 consecutive receivers and 1 frequency, resulting in a total of 1232 parallel CSEM modeling tasks; however the input data array did not have all possible combinations of transmitters and receivers, in particular for long offsets where the responses were below the synthetic noise floor. Thus, only 658 of the tasks needed to be modeled. The MT data were decomposed into two subsets of receivers for each frequency, resulting in 44 parallel tasks. The total number of modeling tasks was thus 702. Since each task is completely independent, the fastest possible run-times for the forward computations would seemingly require 703 processors, where the extra processor accounts for the manager process. However, some tasks can be computed much faster than others depending on the particular subset of transmitters and receivers and the frequency; additionally, MT tasks usually consume only a fraction of the time required for CSEM tasks since they are only computed for the zero wavenumber and do not require the more expensive wavenumber domain Fourier transforms. Experience with marine CSEM data sets has shown that the forward calculations usually reach peak speed when the number of processors is somewhere between 50 and 100 per cent of the total number of tasks, but this is problem dependent.

Fig. 8 shows the scaling of the forward operator $\mathcal{F}(\mathbf{m})$ only and Jacobian matrix operator $\mathbf{J}(\mathbf{m})$ (which includes a concurrent $\mathcal{F}(\mathbf{m})$ calculation) as a function of the number of processors

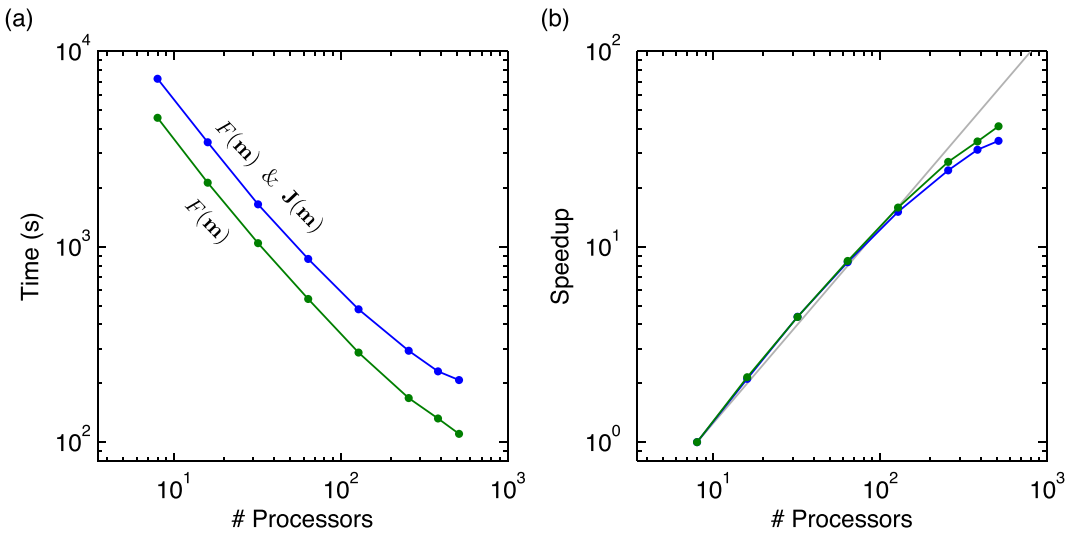

Figure 8. (a) Scaling of the forward operator $\mathcal{F}(\mathbf{m})$ and the combined Jacobian operator $\mathbf{J}(\mathbf{m})$ plus $\mathcal{F}(\mathbf{m})$ as a function of the number of processors. The parallel data decomposition divided the data into subsets of 5 consecutive transmitters and 10 consecutive receivers resulting in a total of 702 data subsets that could be modeled in parallel. (b) The speedup relative to the time taken when run on eight processors.

used. $\mathbf{J}(\mathbf{m})$ takes nearly twice as long to compute as $\mathcal{F}(\mathbf{m})$. Although MARE2DEM leverages the adjoint reciprocity approach to efficiently compute the wavenumber domain data sensitivities, each of these needs to be Fourier transformed to the spatial domain before its scaling and insertion into the Jacobian matrix; for n data and m parameters, mn Fourier transforms are required and this contributes greatly to the longer run-times for the Jacobian calculations. The overall speedup with increasing numbers of processors falls along the optimum scaling maximum until more than 100 processors are used and then begins to drop-off, showing that the manager–worker paradigm used by MARE2DEM scales very efficiently. Investigation of the speedup drop-off with larger number of processors shows that this is due to the variable run-time of each parallel modeling task, with the slowest tasks being responsible for the scaling drop-off as the number of processors approaches the total number of tasks. It is clear that the parallel implementation is worthwhile as the $\mathbf{J}(\mathbf{m})$ calls took 2 hr when using only eight processors, but scaled rapidly down to only 3.3 min when using 480 processors.

Fig. 9 shows the resulting inversion models obtained by the MT-only, CSEM-only and joint CSEM-MT data sets. All inversions began with a uniform 1 ohm m starting model. The MT-only inversion fit converged with a target rms misfit 1.0 after 14 iterations, requiring a total run-time of about 24 min when run on 64 processors. The joint CSEM-MT inversion converged at rms 1.0 after 18 iterations and took a total of 6 hr when run on 320 processors. Fig. 10 shows joint inversion rms misfit shown as a function of Occam iteration for the joint data set and the separate CSEM and MT subsets.

Although this example was primarily intended for demonstrating the performance of MARE2DEM, it also serves to illustrate the potential of marine MT and CSEM data for mapping resistive bodies. As expected, the MT inversion is able to recover an increase in resistivity for the larger salt bodies and the basement, but does not recover the thinner gas hydrates and oil reservoir. The resistivity of the salt and basement is underestimated by a factor of 10–30 due to the well-known saturation of MT responses from restive features imbedded in conductive backgrounds; the geometry of the salt bodies is also poorly recovered. The CSEM only inversion does a much better job recovering the salt body geometry and resistivity, in particular for features D and F. It also images the shallow gas hydrates near the seafloor and the deeper oil reservoir. The resistivity in the basement is higher than for the MT only inversion but is still about 10 times lower than the true value; additionally, the basement depth is poorly mapped, especially where it is within about 5 km from the lateral edges of the receiver array, suggesting that CSEM mapping of the depth to basement requires receiver arrays several kilometres wider than the region of interest. The joint inversion is moderately better than the CSEM-only inversion, with better recovery of salt body A, which extends outside the receiver array, and salt body E, which is a difficult target due to being laterally bounded by two much larger and shallower salt bodies . None of the inversions were able to recover the fine-scale resistivity changes due to the layering of the background sediments, but all of them obtained background resistivities that are fairly close to average sediment resistivity. Perhaps the best sediment recovery can be seen in the joint inversion around salt body F, where the inversion model hints that there is increased sediment resistivity above and decreased resistivity below. Overall, it appears that MT data have much less exploration value than CSEM data, but nonetheless the joint inversion is clearly better than the CSEM-only inversion, suggesting it is worthwhile to continue using EM receivers that can record natural source MT data simultaneously with the CSEM acquisition (e.g. Constable 2013).

6 REAL EXAMPLE: CSEM DATA FROM THE MIDDLE AMERICA TRENCH

Here, MARE2DEM is applied to CSEM data from the SERPENT project at the Middle America Trench offshore Nicaragua. The goals of this 2010 survey were to study the fluid content of the subducting plate and the forearc crust offshore Nicaragua using a combination of MT and CSEM data collected along a trench crossing profile. A brief summary of the main results is given here and further details can be found in the cited references. 1-D CSEM interpretation revealed a conductivity increase in the oceanic crust at the trench outer rise that is consistent with water in the pore spaces created by the abundant bending faults that form just prior to the subduction of the Cocos plate (Key *et al.* 2012). More rigorous 2-D inversion

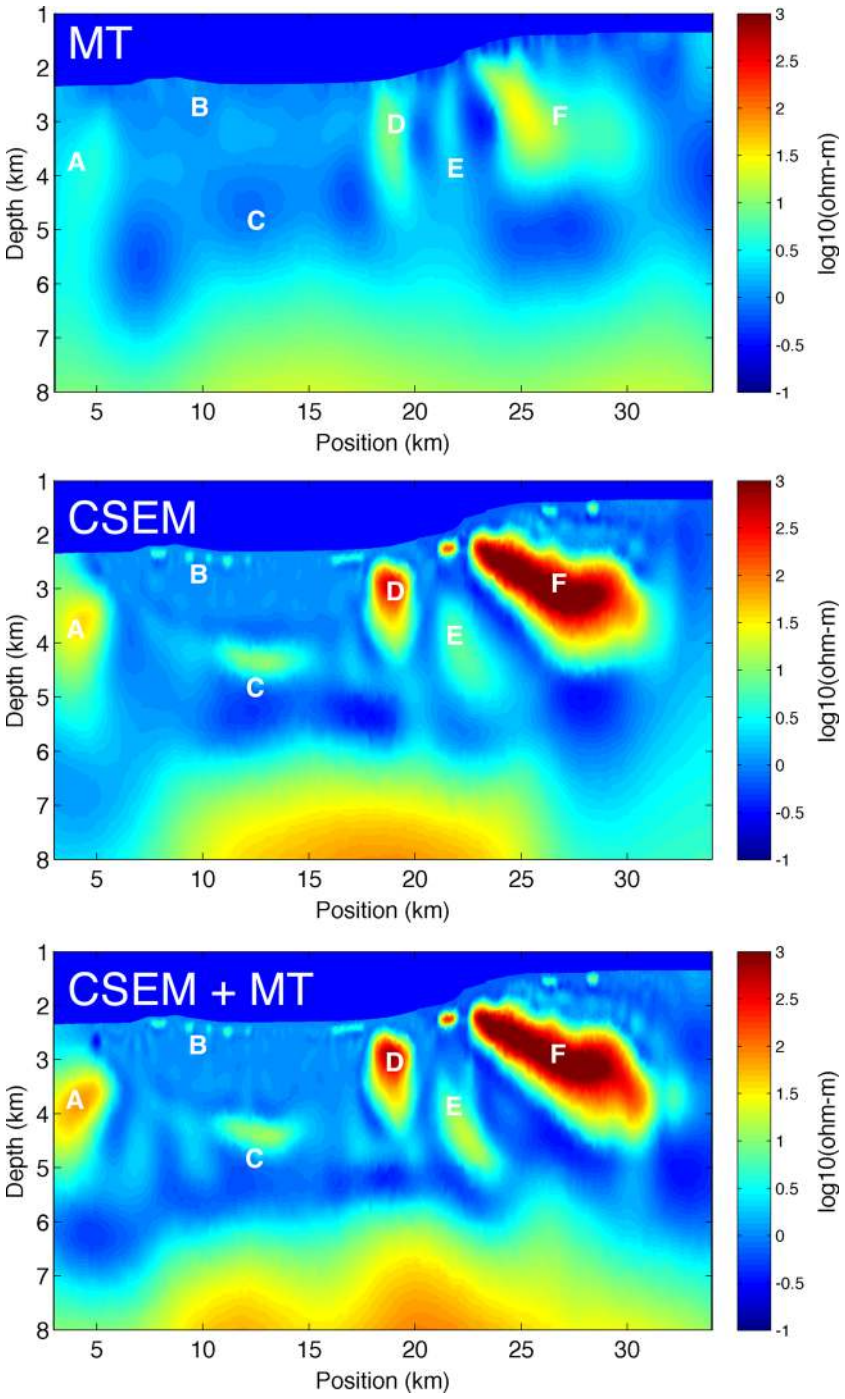

Figure 9. Converged inversion models for the MT only, CSEM only and joint MT and CSEM data sets.

with MARE2DEM imaged a rich fabric of conductive faults on the incoming oceanic plate, constraining a factor of two increase in the crustal bulk water content due to the porosity created in the fault damage zones (Naif *et al.* 2015). 2-D inversion also revealed a continuous band of conductive sediments being subducted to at least 30 km from the trench axis, suggesting a high water content along the shallow part of the plate interface, and also imaged an anomalous high conductivity zone extending from the plate boundary into the overlying forearc crust (Naif 2015).

As a demonstration, the entire trench crossing profile of CSEM data is re-inverted here. The data consist of the topography parallel component of the inline electric field recorded by 34 seafloor EM receivers. The CSEM transmissions were stacked into 60 s time bins, giving a transmitter spacing of about 80 m along the profile. Frequencies of 0.25, 0.75 and 1.75 Hz were selected for inversion since they correspond to the peak harmonics of the transmitter waveform. A total of 30 908 data were inverted using an error floor of only 2 per cent. Fig. 11 shows the 1 ohm m half-space starting model and the central portion of the inversion grid with 20 004 total free parameters. Here, a quadrilateral grid was used since given the relatively wide station spacing it was desired to have thin but wide parameters. Vertically thin parameters offer the possibility for resolving finely detailed structured beneath each station where resolution is highest, while horizontally wide parameters are useful for

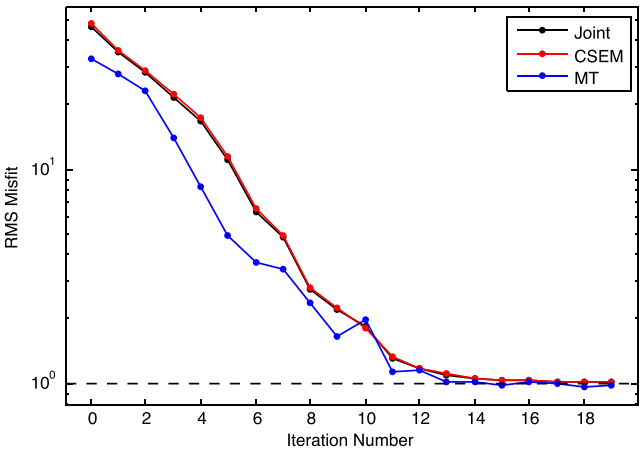

Figure 10. Joint inversion rms misfit shown as a function of Occam iteration for the joint misfit and the breakout for the CSEM and MT subsets.

keeping the total number of parameters from growing excessively large. The starting model has an rms misfit of 187.

Fig. 11 shows the final rms 1.0 converged model found after 32 Occam iterations. The model contains the previously mentioned features including conductive subvertical bending faults on the trench outer rise at positions of about 70–125 km and a band of wet sediments being subducted at 125–150 km position. Fig. 12 shows the data fit as a function of receiver position, source–receiver range and frequency. Overall these metrics show a good fit to the data since both the amplitude and phase are fit about equally well and there are no significant outliers nor large biases in the fit as function of any of these survey parameters.

7 CONCLUSIONS

This work demonstrated the benefits of using adaptive finite elements with unstructured grids for the 2-D inversion of CSEM and MT data. The most important practical advantage is the ease of building models and running inversions with this approach. Unstructured grids accurately represent seafloor topography and other *a priori* known geological boundaries in the inversion model. This removes any question of how to best simulate the data; the topography can be included as precisely as it is known and the electric field receiver dipoles can be set to be parallel to the local slope (or perpendicular to the slope for the nominally vertical component); this is in contrast to codes using structured rectangular grids where topography must be represented by a more subjective stair-step approximation. The unstructured parameter grid allows inversion parameters to be concentrated where needed while the outer regions of the model can be much more coarsely discretized, unlike structured grids where thin or narrow parameters in the region of interest extend out to the model sides. The goal-oriented adaptive mesh refinement method automatically computes accurate EM responses for a given model, freeing the user from the burden of designing an accurate forward modeling grid. Other features included in MARE2DEM such as anisotropy, the fast Occam method, bounds on model parameters and its highly scalable parallel implementation allow it to efficiently handle a diversity of EM modeling needs for both offshore and onshore EM exploration.

ACKNOWLEDGEMENTS

This work benefitted from helpful discussions with David Alumbaugh, Steven Constable, Mike Hoversten, David Myer, Samer

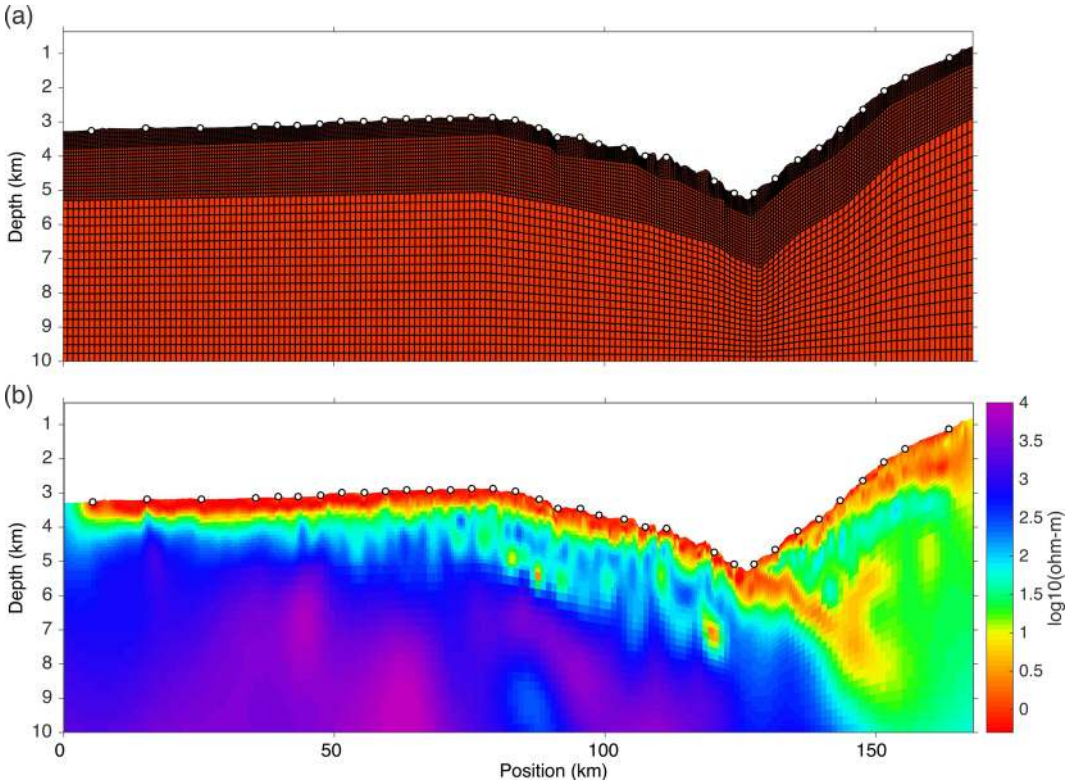

Figure 11. Real data inversion of marine CSEM data from the Middle America Trench offshore Nicaragua. (a) The multiscale quadrilateral grid of parameters with 1 ohm m starting values. Only the central region spanning the survey profile is shown. (b) Inversion model fitting the data to rms 1.0. Note the vertical exaggeration.

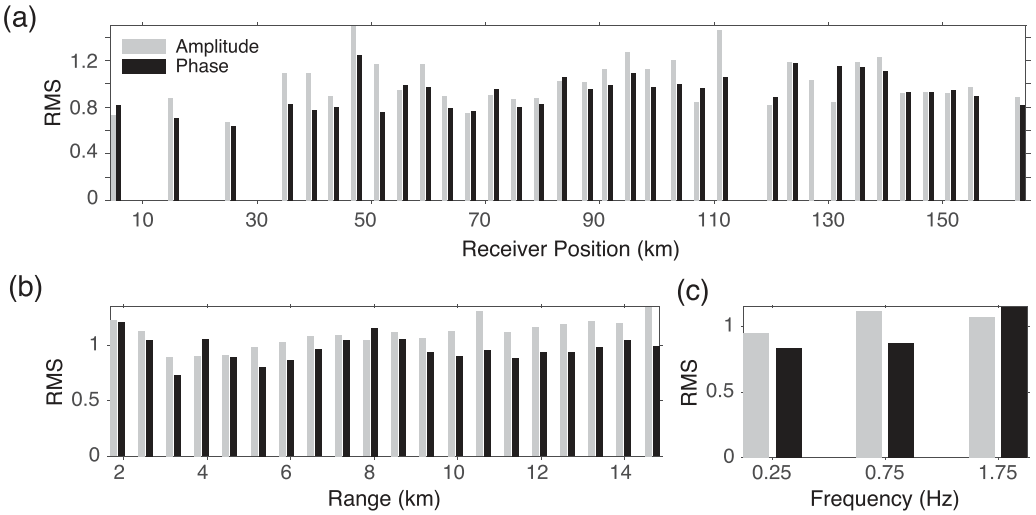

Figure 12. Breakdown of the rms 1.0 data fit. The misfit is shown separately for the amplitude and phase as a function of (a) receiver position, (b) source–receiver range and (c) frequency.

Naif and Jeff Ovall. David Myer is additionally thanked for his contributions to the MATLAB user interface software for MARE2DEM. The San Diego Supercomputer Center at UC San Diego provided technical support for the Triton Shared Computing Cluster. Editor Ute Weckman and two anonymous reviewers offered helpful suggestions. Funding support by the Seafloor Electromagnetic Methods Consortium made the free distribution of the MARE2DEM possible. National Science Foundation grants OCE-0841114 and OCE-0840894 supported the SERPENT project.

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.

Bennington, N.L., Zhang, H., Thurber, C.H. & Bedrosian, P.A., 2015. Joint inversion of seismic and magnetotelluric data in the Parkfield region of California using the normalized cross-gradient constraint, *Pure appl. Geophys.,* **172**(5), 1033–1052.

Blackford, L.S. *et al.*, 1997. *ScaLAPACK Users' Guide,* SIAM.

Box, M.J., 1966. A comparison of several current optimization methods, and the use of transformations in constrained problems, *Comput. J.,* **9**(1), 67–77.

Commer, M. & Newman, G.A., 2008. New advances in three-dimensional controlled-source electromagnetic inversion, *Geophys. J. Int.,* **172**(2), 513–535.

Commer, M. & Newman, G.A., 2009. Three-dimensional controlled-source electromagnetic and magnetotelluric joint inversion, *Geophys. J. Int.,* **178**(3), 1305–1316.

Commer, M., Newman, G.A., Caraone, J.J., Dickens, T.A., Green, K.E., Wahrmund, L.A., Willen, D.E. & Shiu, J., 2008. Massively parallel electrical-conductivity imaging of hydrocarbons using the IBM Blue Gene/L supercomputer, *IBM J. Res. Dev.,* **52,** 93–103.

Constable, S., 2010. Ten years of marine CSEM for hydrocarbon exploration, *Geophysics,* **75**(5), 75A67–75A81.

Constable, S., 2013. Review paper: instrumentation for marine magnetotelluric and controlled source electromagnetic sounding, *Geophys. Prospect.,* **61,** 505–532.

Constable, S., Orange, A. & Key, K., 2015. And the geophysicist replied: "Which model do you want?", *Geophysics,* **80**(3), E197–E212.

Constable, S.C., Parker, R.L. & Constable, C.G., 1987. Occam's inversion—a practical algorithm for generating smooth models from electromagnetic sounding data, *Geophysics,* **52**(03), 289–300.

deGroot Hedlin, C. & Constable, S., 1990. Occam's inversion to generate smooth two-dimensional models from magnetotelluric data, *Geophysics,* **55**(12), 1613–1624.

Egbert, G.D. & Kelbert, A., 2012. Computational recipes for electromagnetic inverse problems, *Geophys. J. Int.,* **189**(1), 251–267.

Eisenberg, M.A. & Malvern, L.E., 1973. On finite element integration in natural co-ordinates, *Int. J. Numer. Methods Eng.,* **7**(4), 574–575.

Farquharson, C.G. & Oldenburg, D.W., 1996. Approximate sensitivities for the electromagnetic inverse problem, *Geophys. J. Int.,* **126**(1), 235–252.

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, *Geophys. J. Int.,* **171**(1), 71–86.

Habashy, T.M. & Abubakar, A., 2004. A general framework for constraint minimization for the inversion of electromagnetic measurements, *Prog. Electromagn. Res.,* **46,** 265–312.

Harrington, R., 2001. *Time Harmonic Electromagnetic Fields,* IEEE Press.

Hoversten, G.M., Myer, D., Key, K., Alumbaugh, D., Hermann, O. & Hobbet, R., 2015. Field test of sub-basalt hydrocarbon exploration with marine controlled source electromagnetic and magnetotelluric data, *Geophys. Prospect.,* **63,** 1284–1310.

Jahandari, H. & Farquharson, C.G., 2014. A finite-volume solution to the geophysical electromagnetic forward problem using unstructured grids, *Geophysics,* **79**(6), E287–E302.

Key, K. & Ovall, J., 2011. A parallel goal-oriented adaptive finite element method for 2.5-D electromagnetic modelling, *Geophys. J. Int.,* **186**(1), 137–154.

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

Key, K., Constable, S., Matsuno, T., Evans, R.L. & Myer, D., 2012. Electromagnetic detection of plate hydration due to bending faults at the Middle America Trench, *Earth planet. Sci. Lett.,* **351–352,** 45–53.

Key, K., Constable, S., Liu, L. & Pommier, A., 2013. Electrical image of passive mantle upwelling beneath the northern East Pacific Rise, *Nature,* **495**(7442), 499–502.

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.

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

Li, Y. & Dai, S., 2011. Finite element modelling of marine controlled-source electromagnetic responses in two-dimensional dipping anisotropic conductivity structures, *Geophys. J. Int.,* **185,** 622–636.

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

Lu, X., Unsworth, M. & Booker, J., 1999. Rapid relaxation inversion of CSAMT data, *Geophys. J. Int.,* **138**(2), 381–392.

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

McGillivray, P.R., Oldenburg, D.W., Ellis, R.G. & Habashy, T.M., 1994. Calculation of sensitivities for the frequency-domain electromagnetic problem, *Geophys. J. Int.,* **116**(1), 1–4.

Myer, D., Constable, S., Key, K., Glinsky, M.E. & Liu, G., 2012. Marine CSEM of the Scarborough gas field, Part 1: Experimental design and data uncertainty, *Geophysics,* **77**(4), E281–E299.

Myer, D., Key, K. & Constable, S., 2015. Marine CSEM of the Scarborough gas field, Part 2: 2D inversion, *Geophysics,* **80**(3), E187–E196.

Naif, S., Key, K., Constable, S. & Evans, R.L., 2013. Melt-rich channel observed at the lithosphere-asthenosphere boundary, *Nature,* **495**(7441), 356–359.

Naif, S., Key, K., Constable, S. & Evans, R.L., 2015. Water-rich bending faults at the Middle America Trench, *Geochem. Geophys. Geosyst.,* **16**(8), 2582–2597.

Naif, S.N., 2015. Marine electromagnetic experiment across the Nicaragua Trench: imaging water-rich faults and melt-rich asthenosphere, *PhD thesis,* University of California San Diego.

Newman, G.A., 2013. A review of high-performance computational strategies for modeling and imaging of electromagnetic induction data, *Geophys. Surv.,* **35**(1), 85–100.

Newman, G.A. & Alumbaugh, D.L., 1997. Three-dimensional massively parallel electromagnetic inversion – 1. Theory, *Geophys. J. Int.,* **128**(2), 345–354.

Newman, G.A. & Alumbaugh, D.L., 2000. Three-dimensional magnetotelluric inversion using non-linear conjugate gradients, *Geophys. J. Int.,* **140**(2), 410–424.

Newman, G.A. & Boggs, P.T., 2004. Solution accelerators for large-scale three-dimensional electromagnetic inverse problems, *Inverse Probl.,* **20**(6), S151–S170.

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

Parasnis, D.S., 1988. Reciprocity theorems in geoelectric and geoelectromagnetic work, *Geoexploration,* **25**(3), 177–198.

Pardo, D., Nam, M.J., Torres-Verdín, C., Hoversten, M.G. & Garay, I., 2011. Simulation of marine controlled source electromagnetic measurements using a parallel fourier hp-finite element method, *Comput. Geosci.,* **15**(1), 53–67.

Parker, R.L., 1994. *Geophysical Inverse Theory,* Princeton Univ. Press.

Pedersen, H.T. & Hiner, M., 2014. Exploring channel reservoirs in a frontier area with regional CSEM as fluid indicator, in *SEG Technical Program Expanded Abstracts 2014,* pp. 798–802, Society of Exploration Geophysicists.

Portniaguine, O. & Zhdanov, M.S., 1999. Focusing geophysical inversion images, *Geophysics,* **64**(3), 874–887.

Press, W.H., Teukolsky, S.A., Vetterling, W.T. & Flannery, B.P., 2003. *Numerical Recipes in Fortran 77,* Vol. 1, Cambridge Univ. Press.

Prudhomme, S. & Oden, J.T., 1999. On goal-oriented error estimation for elliptic problems: application to the control of pointwise errors, *Comput. Methods Appl. Mech. Eng.,* **176**(1–4), 313–331.

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

Rodi, W. & Mackie, R.L., 2001. Nonlinear conjugate gradients algorithm for 2-D magnetotelluric inversion, *Geophysics,* **66**(1), 174–187.

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

Schwarzbach, C. & Haber, E., 2013. Finite element based inversion for time-harmonic electromagnetic problems, *Geophys. J. Int.,* **193**(2), 615–634.

Siripunvaraporn, W., 2012. Three-dimensional magnetotelluric inversion: an introductory guide for developers and users, *Geophys. Surv.,* **33**(1), 5–27.

Tehrani, A.M. & Slob, E., 2013. Applicability of 1D and 2.5D marine controlled source electromagnetic modelling, *Geophys. Prospect.,* **61**(s1), 602–613.

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

Ward, S.H. & Hohmann, G.W., 1987. Electromagnetic theory for geophysical applications, in *Electromagnetic Methods in Applied Geophysics,* pp. 131–312, ed. Nabighian, M.N., Society of Exploration Geophysicists Tulsa.

Weiss, C.J. & Constable, S., 2006. Mapping thin resistors in the marine environment, Part II: Modeling and analysis in 3D, *Geophysics,* **71**(6), G321–G332.

Wheelock, B., Constable, S. & Key, K., 2015. The advantages of logarithmically scaled data for electromagnetic inversion, *Geophys. J. Int.,* **201**(3), 1765–1780.

Wiik, T., Hokstad, K., Ursin, B. & Mütschard, L., 2013. Joint contrast source inversion of marine magnetotelluric and controlled-source electromagnetic data, *Geophysics,* **78**(6), E315–E327.

Zhdanov, M.S., 2002. *Inverse Theory and Applications in Geophysics,* Elsevier.

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

APPENDIX A: FINITE-ELEMENT DETAILS

For a triangulation $\mathcal{T}_n$ of the model domain with n vertices, eq. (14) can be written as a $2n \times 2n$ sparse linear system:

$$\mathbf{Bu} = \mathbf{f}. \tag{A1}$$

This can be expanded to

$$\begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{pmatrix}, \tag{A2}$$

where $\mathbf{u}_1$ and $\mathbf{u}_2$ are vectors of the strike parallel electric and magnetic field coefficients at the triangulation mesh vertices. The entries of the $n \times n$ submatrices in matrix $\mathbf{B}$ have the form:

$$B_{11}: \int_\Omega (\sigma_t \lambda \nabla v_i \cdot \nabla v_j + \sigma_x v_i v_j)\, d\Omega, \tag{A3}$$

$$B_{12} = -B_{21}^T: \int_\Omega ik_x (\lambda R \nabla v_i) \cdot \nabla v_j \, d\Omega, \tag{A4}$$

$$B_{22}: \int_\Omega i\omega\mu(\lambda' \nabla v_i \cdot \nabla v_j + v_i v_j)\, d\Omega. \tag{A5}$$

The entries of the source vector $\mathbf{f}$ have the form:

$$\mathbf{f}_1: \int_\Omega \left[-\left(ik_x \lambda \hat{\mathbf{J}}_t^s - \sigma_t \lambda R \hat{\mathbf{M}}_t^s \right) \cdot \nabla v_j - \hat{J}_x v_j \right] d\Omega, \tag{A6}$$

$$\mathbf{f}_2: \int_\Omega \left[-\left(-i\omega\mu R \lambda \hat{\mathbf{J}}_t^s - ik_x \lambda' \hat{\mathbf{M}}_t^s \right) \cdot \nabla v_j - \hat{M}_x v_j \right] d\Omega. \tag{A7}$$

The integrals of the basis function products in eqs (A3)–(A5) can be evaluated exactly with either numerical quadrature or formulae for polynomial integration; the latter are provided below. The discrete electric and magnetic sources $\mathbf{J_s}$ and $\mathbf{M_s}$ are implemented using vector delta functions. For example, a point electric dipole source

is represented as $\hat{\mathbf{J}}_s = (\hat{J}^s_x, \hat{\mathbf{J}}^s_t) = \delta(x - x_s)\hat{x} + \delta(y - y_s, z - z_s)\hat{\mathbf{t}}$, where $\hat{\mathbf{t}}$ defines a unit vector in the (y, z) plane. Here, the sources are decomposed into separate $\hat{x}$ and $\hat{\mathbf{t}}$ aligned sources. This decomposition produces electric and magnetic field components that are either symmetric or antisymmetric along the x axis, depending on the direction of the source direction and the field component. Thus, the Fourier transform integrals in eqs (11) and (16) can be converted into either cosine or sine transforms that only require the kernel function at positive k_x, with the significant benefit that the relatively expensive finite-element system only needs to be solved for positive k_x. Further details on the implementation of the transforms are discussed in Key & Ovall (2011).

In the finite-element method, the integral expressions (A3)–(A7) are broken up into a sum of integrals evaluated over each element (e.g. Zhu & Cangellaris 2006). Since the conductivity coefficients are restricted to be piecewise continuous across the elements, these integrals can be evaluated through direct integration of the polynomial basis (e.g. Eisenberg & Malvern 1973). For a given triangular element τ with corner vertices $(y_i, z_i, i = 1, 2, 3)$ and edges e_i opposite of vertex i, the area of the triangle $|\tau|$ is defined by

$$2|\tau| = (a_i + b_i y_i + c_i z_i), \tag{A8}$$

where

$$a_i = y_j z_k - y_k z_j, \tag{A9}$$

$$b_i = z_j - z_k, \tag{A10}$$

$$c_i = y_k - y_j, \tag{A11}$$

with cyclic permutation of i, j, k. For the linear space V_n, the basis vectors are $v_i = l_i$, where

$$l_i = \frac{1}{2|\tau|}(a_i + b_i y + c_i z). \tag{A12}$$

l_i varies from unity at vertex i to 0 at the vertices j and k. For the hierarchical basis of space W_n, the basis vectors are $v_i = q_i$, where the quadratic bump functions associated with edge i is

$$q_i = 4 l_j l_k, \tag{A13}$$

with cyclic permutation of i, j, k. The bump function q_i is zero at all three vertices and has a quadratic shape along the edge opposite of vertex i. Defining the diagonal coefficient matrix

$$D = \begin{pmatrix} d_1 & 0 \\ 0 & d_2 \end{pmatrix}, \tag{A14}$$

and using the linear basis functions $v_i = l_i$ and $v_j = l_j$, the integrals in expressions (A3)–(A5) for element τ can be shown to be

$$\int_\tau D\nabla l_i \cdot \nabla l_j = \frac{1}{4|\tau|}\left(d_1 b_i b_j + d_2 c_i c_j\right), \tag{A15}$$

$$\int_\tau l_i l_j = \begin{cases} \frac{|\tau|}{6}, & i = j \\ \frac{|\tau|}{12}, & i \neq j \end{cases}, \tag{A16}$$

$$\int_\tau (DR\nabla l_i) \cdot \nabla l_j = \frac{1}{4|\tau|}\left(d_2 b_i c_j - d_1 b_j c_i\right). \tag{A17}$$

For the quadratic basis functions $v_i = q_i$ and $v_j = q_j$, the integrals can be shown to be:

$$\int_\tau D\nabla q_i \cdot \nabla q_j = \begin{cases} \frac{2}{3|\tau|}\left(d_1\left(b_i^2 - b_j b_k\right) + d_2\left(c_i^2 - c_j c_k\right)\right), & i = j \\ \frac{2}{3|\tau|}\left(d_1 b_i b_j + d_2 c_i c_j\right), & i \neq j \end{cases}, \tag{A18}$$

$$\int_\tau q_i q_j = \begin{cases} \frac{8}{45}|\tau|, & i = j \\ \frac{4}{45}|\tau|, & i \neq j \end{cases}, \tag{A19}$$

$$\int_\tau DR\nabla q_i \cdot \nabla q_j = \begin{cases} (d_2 - d_1)\frac{1}{3|\tau|}\left(b_i c_i + b_j c_j + b_k c_k\right), & i = j \\ (d_2 - d_1)\frac{1}{3|\tau|}\left(b_i c_j + b_j c_i\right), & i \neq j \end{cases}. \tag{A20}$$

For the isotropic conductivity the last integral vanishes, allowing for the bump systems to be solved separately for $\hat{E}_x$ and $\hat{H}_x$. The right-hand sides of (B2), (B4) and (B5) require the integral formulae

$$\int_\tau D\nabla l_i \cdot \nabla q_j = -\frac{1}{3|\tau|}\left(d_1 b_i b_j + d_2 c_i c_j\right), \tag{A21}$$

$$\int_\tau l_i q_j = \begin{cases} \frac{2|\tau|}{15}, & i \neq j \\ \frac{|\tau|}{15}, & i = j \end{cases}, \tag{A22}$$

$$\int_\tau (DR\nabla l_i) \cdot \nabla q_j = \frac{1}{3|\tau|}\left(d_1 b_j c_i - d_2 b_i c_j\right). \tag{A23}$$

APPENDIX B: GOAL-ORIENTED ADAPTIVE MESH REFINEMENT DETAILS

Given a triangulation $\mathcal{T}_n$ and the spaces W_n and V_n described in Appendix A, the goal-oriented error estimator is obtained by solving a sequence of linear systems:

$$B(\mathbf{u}_n, \mathbf{v}) = F(\mathbf{v}) \qquad \text{for all } \mathbf{v} \in V_n, \tag{B1}$$

$$B(\boldsymbol{\varepsilon}_n, \mathbf{v}) = F(\mathbf{v}) - B(\mathbf{u}_n, \mathbf{v}) \qquad \text{for all } \mathbf{v} \in W_n, \tag{B2}$$

$$B(\mathbf{v}, \mathbf{w}_n) = G(\mathbf{v}) \qquad \text{for all } \mathbf{v} \in V_n, \tag{B3}$$

$$B(\mathbf{v}, \boldsymbol{\delta}_n) = G(\mathbf{v}) - B(\mathbf{v}, \mathbf{w}_n) \qquad \text{for all } \mathbf{v} \in W_n. \tag{B4}$$

Eq. (B1) is the primal problem; the error in its solution is approximated with the residual eq. (B2), which represents a $2m \times 2m$ linear system of edge bump functions associated with the m edges in $\mathcal{T}_n$. Eq. (B3) is a dual problem and the error in its solution is similarly approximated with the residual eq. (B4). The dual problem is linked to the primal problem by incorporating its error estimator $\boldsymbol{\varepsilon}_n$ in the goal function:

$$G(\mathbf{v}) \equiv a_0 \int_{\tau_s} \bar{\boldsymbol{\varepsilon}}_n \cdot \mathbf{v} \; dV + a_1 \int_{\tau_s} \nabla\bar{\boldsymbol{\varepsilon}}_n : \nabla\mathbf{v} \; dV, \tag{B5}$$

where

$$a_0 = \frac{1}{\int_{\tau_s}\left(|\mathbf{u}_n + \boldsymbol{\varepsilon}_n|^2 + e_0^2\right) dV}, \tag{B6}$$

$$a_1 = \frac{1}{\int_{\tau_s}\left(|\nabla(\mathbf{u}_n + \boldsymbol{\varepsilon}_n)|^2 + e_1^2\right) \; dV}, \tag{B7}$$

τ_s is the subdomain of elements containing the measurement locations and e_0 and e_1 are small constants to ensure the denominator is non-zero when u_n and ε_n are vanishingly small. Thus, it can be seen that the goal function is a relative measure of the solution error at each measurement location (i.e. at the positions of the EM receivers). After the dual problem's error approximation, $\boldsymbol{\delta}_n$ has been calculated with (B4), the goal-oriented error estimator (i.e. the error in the goal function) is then :

$$G(\mathbf{u} - \mathbf{u}_n) \approx F(\boldsymbol{\delta}_n) - B(\mathbf{u}_n, \boldsymbol{\delta}_n), \tag{B8}$$

and the local element error indicator for triangle τ is:

$$G_\tau = |F(\boldsymbol{\delta}_n)_\tau - B(\mathbf{u}_n, \boldsymbol{\delta}_n)_\tau|. \tag{B9}$$

Additionally, the solution to the primal problem can be improved by incorporating the estimated error using the expression $\mathbf{u} \approx \mathbf{u}_n + \boldsymbol{\varepsilon}_n$.

At first glance eqs (B2)–(B4) appear to be quite expensive since they require the solution of three additional linear systems. However, note that the primary and dual problems share the same system matrix due to symmetry. Since MARE2DEM uses a sparse direct factorization routine to solve the primary problem (B1), this factorization is reused for the dual problem (B3); thus solving the dual problem only requires a new sequence of forward and backward substitutions of the factored matrix, which can be done in a fraction of the time required to factor the matrix. The sparse $2\,\mathrm{m} \times 2\,\mathrm{m}$ systems for the primal error (B2) and dual error (B4) appear to be drastically more expensive since $m \approx 3n$. However, these systems are well conditioned due to the limited support of the edge bump functions and thus they can be solved efficiently using only a few iterations of the Gauss–Seidel method. In all cases, the system matrices are stored in compressed-sparse row format with less than about 20–30 non-zero terms per row, depending on the unstructured mesh topology, and thus do not require significant storage.